Departamento de Ciencias de la Educación, Lenguaje, Cultura y Artes, Ciencias Histórico-Jurídicas y Humanísticas y Lenguas Modernas

Facultad de Ciencias Jurídicas y Sociales

Universidad Rey Juan Carlos

Tesis Doctoral

ENTERPRISE SECURITY PATTERNS

Un nuevo tipo de Patrón de Seguridad

Doctorando: Santiago Moral García

Director de Tesis: David García Rosado

Co-Director de Tesis: Juan Manuel Vara Mesa

Co-Director de Tesis: Eduardo Fernández-Medina Patón

Departamento de Ciencias de la Educación, Lenguaje, Cultura y Artes, Ciencias Histórico-Jurídicas y Humanísticas y Lenguas Modernas

Facultad de Ciencias Jurídicas y Sociales

Universidad Rey Juan Carlos

Tesis Doctoral

ENTERPRISE SECURITY PATTERNS

Un nuevo tipo de Patrón de Seguridad

Doctorando: Santiago Moral García

Director de Tesis: David García Rosado

Co-Director de Tesis: Juan Manuel Vara Mesa

Co-Director de Tesis: Eduardo Fernández-Medina Patón

Universidad Rey Juan Carlos

El Dr. D. David García Rosado, de la Universidad de Castilla-La Mancha, el Dr. D. Juan Manuel Vara Mesa, de la Universidad Rey Juan Carlos, y el Dr. D. Eduardo Fernández-Medina Patón, de la Universidad de Castilla-La Mancha, director y co-directores, respectivamente, de la Tesis Doctoral: "Enterprise Security Patterns. Un nuevo tipo de Patrón de Seguridad" realizada por el doctorando Santiago Moral García,

HACEN CONSTAR QUE:

esta Tesis Doctoral reúne los requisitos para su defensa y aprobación

En Madrid, a 30 de marzo de 2017

Fdo.: David García Rosado | Fdo.: Juan Manuel Vara Mesa | Fdo.: Eduardo Fernández-Medina Patón

A Sara por apoyarme.

A mis padres, por ayudarme incondicionalmente.

A mi hermana, por aguantar largas conversaciones sobre estos temas.

A mis abuelos, siempre pensando cómo hacerme más feliz.

A mis tíos y prima, un ejemplo a seguir.

A Anais por su sonrisa.

Agradecimientos

El concepto de esta tesis no hubiera existido sin Santiago. Libre pensador y mejor padre. Él creó desde cero el Modelo Casandra y me convenció para embarcarme en este viaje.

No hubiera sido posible finalizar este viaje sin la ayuda consistente de Eduardo, David y Juancho. Han dirigido la tesis de forma excelente y son unos tutores excepcionales. Las universidades del mundo serían mucho mejor si tuvieran gente como ellos.

A Roberto, mi compañero de fatigas en esta gran aventura. Han sido muchas noches trabajando y muchas tardes disfrutando de su presencia. Los viajes juntos ya fueron la guinda del pastel.

A Eduardo Fernandez, por acogerme durante 4 meses en la Universidad Florida Atlantic. Sin él nunca hubiéramos llegado a conseguir el nivel de cohesión que tienen los patrones presentados.

Al grupo Kybele, al grupo GSyA, y en especial a Esperanza y Mario por creer y apoyar desde el principio en este proyecto.

No podía dejar de agradecer a Nekane, Idoia, Mario y Jesús por sus aportaciones profesionales. No han apoyado directamente, pero su influencia en el mundo de la seguridad hace que cada día tenga nuevos retos.

Finalmente, agradecer a Sara su paciencia y apoyo durante todos estos años. Sin su ayuda no hubiera sido posible estar escribiendo estas líneas.

“Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica:
La voluntad.”

Albert Einstein

“Lo que sabemos es una gota de agua; Lo que ignoramos es el océano.”

Isaac Newton

Contenido

Lista de Figuras

Lista de Tablas

Resumen

En los últimos años, la mayoría de las organizaciones, sin tener en cuenta su geografía o sector, han sufrido ataques intencionales contra sus activos de información. La mayoría de estos ataques son llevados a cabo por grupos de ciber-crimen organizado, cuyo objetivo principal es obtener o modificar datos confidenciales de las organizaciones.

Por esta razón, el principal objetivo de las organizaciones en términos de seguridad, es asegurar la continuidad de las operaciones de negocio y proteger sus activos de información. Con este propósito en mente, las organizaciones están buscando soporte en las arquitecturas de seguridad empresariales.

El objetivo de las arquitecturas de seguridad empresariales es proporcionar un diseño conceptual de las infraestructuras de seguridad. Este diseño enlaza todos los componentes incluidos en las arquitecturas como una unidad cohesiva con el objetivo de proteger los activos de la organización. Para hacer esto, las arquitecturas de seguridad empresariales determinan qué activos de información deben ser protegidos, desde qué tipo de ataques, y quién (personas) o qué (sistemas) tiene acceso a la información.

Debido al valor fundamental de los activos de información para las organizaciones, es necesario un enfoque sistemático a la hora de construir sistemas seguros. Las metodologías basadas en patrones proporcionan este enfoque. Dentro del ámbito de los patrones hay varios tipos de catálogos. En particular, los patrones de seguridad combinan un conocimiento significativo sobre seguridad con la estructura sistemática proporcionada por los patrones. Estos patrones proporcionan guías para apoyar la construcción y evaluación de mecanismos de

seguridad. El uso de los patrones de seguridad ayuda a incorporar los principios de seguridad a la hora de construir sistemas seguros, sin embargo, estos patrones tienen algunas limitaciones a la hora de construir grandes arquitecturas de seguridad.

Debido a estas limitaciones, en esta tesis doctoral hemos definido un nuevo tipo de patrón de seguridad, llamado Enterprise Security Patterns, para facilitar el diseño de arquitecturas de seguridad empresariales. Hemos adoptado ese nombre, porque el objetivo de estos patrones es proporcionar una estrategia basada en modelos para definir arquitecturas de seguridad en diferentes niveles de abstracción, incluyendo la implementación tecnológica. Estos patrones no intentan remplazar a los patrones de seguridad. Estos patrones utilizan e incorporan los patrones seguridad, proporcionando un patrón cohesivo que puede gestionar más riesgos o amenazas.

Enterprise Security Patterns combinan un amplio rango de elementos describiendo arquitecturas de seguridad empresariales que protegen activos de información en un contexto específico. Para ello, describimos el meta-modelo del patrón, un proceso de desarrollo de arquitecturas de seguridad empresariales basado en el enfoque de arquitecturas dirigidas por modelos, un nuevo lenguaje especifico del dominio para diseñar nuevas arquitecturas, un conjunto de transformaciones para aplicar transiciones entre los modelos de la solución y un proceso para facilitar la minería de este nuevo tipo de patrones. Además, mostramos un caso de estudio real realizado con una entidad financiera internacional.

A la hora de mitigar un problema de seguridad, las organizaciones podrían utilizar los Enterprise Security Patterns con el objetivo de seleccionar una estrategia de seguridad global, proporcionando a sus diseñadores un conjunto de guías de seguridad óptimas y validadas. Además, los ingenieros de seguridad podrían (i) gestionar por separado los elementos de seguridad incluidos en los distintos modelos de abstracción y (ii) realizar transformaciones entre los elementos. Este hecho facilitaría al diseñador en la selección y ajuste de las políticas de seguridad, mecanismos y tecnologías a la hora de construir nuevas arquitecturas.

Abstract

In recent years, the vast majority of organizations, regardless of their geographic location or industry, have suffered intentional attacks against their information systems. Most of these attacks are carried out by organized e-crime groups, whose main objective is to obtain or modify sensitive data from organizations.

For this reason, the main objective of organizations, in terms of security, is to ensure the continuity of business operations and protect their information assets. With these purposes in mind, organizations are seeking support from enterprise security architectures.

The objective of enterprise security architecture is to provide the conceptual design of the security infrastructure. This design links the components of security infrastructure as one cohesive unit in order to protect corporate information. To do this, the enterprise security architecture determine what information assets must be protected, from what types of attacks, and who (people) or what (system) has access to them.

Due to the fundamental value of information assets to enterprises, a systematic approach is required to build secure systems. Methodologies based on patterns provide this approach. Within the scope of patterns, we may find several catalogs. Security patterns join the extensive knowledge accumulated about security with the structure provided by patterns. These patterns provide the guidelines to support the construction and evaluation of security mechanisms. The use of security patterns helps to incorporate security principles when building secure systems. However, they have some limitations when building large security architectures.

Because of these limitations, in this thesis we have defined a new type of pattern, called Enterprise Security Pattern, to support the design of enterprise security architectures. We adopt this name, because the objective of these patterns is to provide a strategy based on models for defining enterprise security architectures in different levels of abstraction, including their technological implementation. These patterns are not intended to replace security patterns. They use and incorporate them in a more comprehensive pattern that can handle more threats.

An enterprise security pattern combines a wide range of items describing generic enterprise security architectures that protect a set of information assets in a specific context. To do this, we describe a precise pattern meta-model, a process to develop enterprise security architectures based on a Model-Driven Architecture (MDA) approach, a new domain specific language to design new architectures, a set of transformations to make transitions between solution models, and a process to facilitate Enterprise Security Pattern Mining. We also show a use case performed with a large financial institution.

When avoiding a security problem, organizations could use enterprise security patterns in order to select a global security strategy, providing their designers with an optimal and proven set of security guidelines. Using enterprise security patterns security engineers could, on the one hand, manage separately the security elements included in the different abstraction models, and on the other hand, perform automatic transformations between them. This fact would facilitate the designer in the selection and tailoring of security policies, patterns, mechanisms and technologies when they are building enterprise security architectures.

1. Introducción

1.1 Motivación

La tecnología está cada vez más presente en casi todos los aspectos de nuestra vida personal y profesional, mejorándola y generando cada vez más oportunidades para la innovación, aunque también dejando más espacio a las amenazas. Como cualquier otro activo de valor, la información también atrae la atención de los ciber-delincuentes, que buscan nuevas formas de apoderarse de ella y aprovecharla en su beneficio (McAfee Labs, 2016). En las últimas décadas los problemas relacionados con la seguridad de la información se han incrementado considerablemente en todos los tipos de compañías (financieras, tecnológicas, industriales, aeronáuticas, sanitarias, etc.), haciendo que su gestión sea extremadamente difícil y costosa.

La industria de la ciberseguridad prevé mover más de €175.000 millones en 2020, y las pérdidas que causan los ciberataques ascendieron sólo el año pasado hasta el entorno de los €350.000 millones, dadas las crecientes amenazas, cada vez más sofisticadas y peligrosas en Internet, según la Fundación Innovación Bankinter (Fundación Innovación Bankinter, 2016). Otro informe elaborado por Cybersecurity Ventures pronostica que el coste global en productos y servicios de ciberseguridad para hacer frente al ciber-crimen excederá el trillón de dólares en los próximos 5 años, del 2017 al 2021 (CyberSecurity Ventures, 2016). La Figura 1.1 muestra las pérdidas generadas por el ciber-crimen en comparación con el Producto Interior Bruto de cada uno de los países.

La mayoría de estos ataques se llevan a cabo por grupos de ciberdelincuencia organizada, cuyo principal objetivo en la mayoría de los casos, es obtener o modificar los datos sensibles de las organizaciones (Zhang et al., 2012; Ali Al et al., 2016). Los grupos de ciber-crimen no eligen datos aleatorios para atacar. La característica principal de estos datos es que pueden ser monetizados, es decir, pueden producir un beneficio económico para el atacante, que conlleva una pérdida económica para la organización atacada (IC3, 2015). Por esta razón, en los últimos años, el principal objetivo de las organizaciones, en términos de seguridad, es garantizar la continuidad de las operaciones de negocio y proteger las propiedades de seguridad de sus activos

de información, como son la confidencialidad, integridad, disponibilidad y auditabilidad de los datos (Harrop y Matteson, 2015; Shin et al., 2015; Cavelty y Mauer, 2016; Dua y Du, 2016).

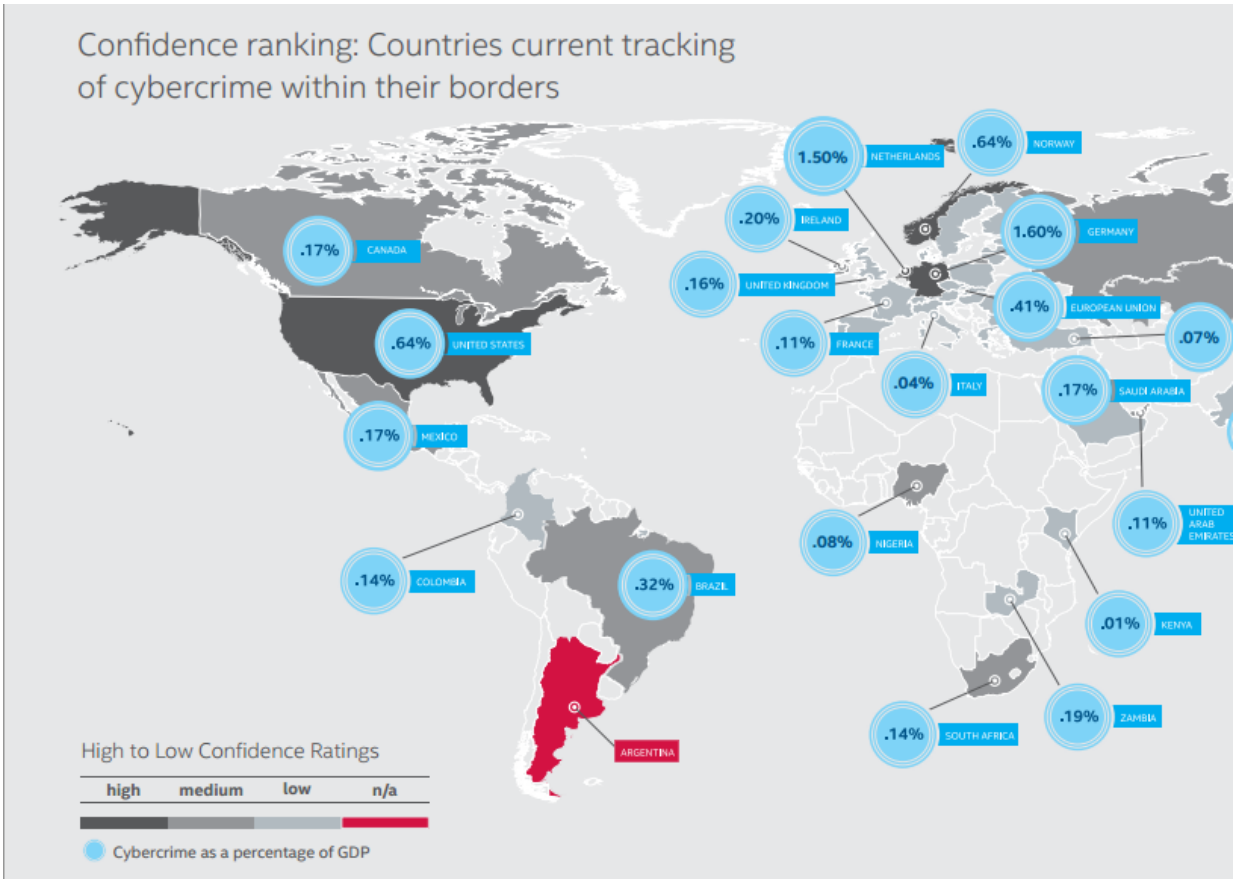


Figura 1.1. Ciber-crimen como porcentaje del Producto Interior Bruto

En cuanto a seguridad y análisis de riesgos, 2003 se considera un punto de inflexión que significa un antes y un después. Esto es así porque fue cuando se produjo el primer phishing en España (ElPais, http://elpais.com). Se empezaba a pensar que las metodologías de análisis de riesgos estudiadas hasta el momento tenían algunas deficiencias, ya que nadie en el mundo se había protegido ante este problema. Ninguna de las metodologías de análisis de riesgos más importantes hasta la fecha CRAMM (Yazar, 2002; CCTA, 2003), Octave (Alberts y Dorofee, 2002; Caralli et al., 2007), Magerit (MAP, 2012) fueron capaces de ayudar a decidir cuáles eran los siguientes pasos a dar. Nace en este momento un cierto escepticismo general hacia las metodologías de análisis de Gestión de Riesgos y hacía los Sistemas de Gestión de Seguridad de

la Información basados en la primera normativa internacional de seguridad, ISO/IEC 17799 (ISO, http://www.iso.org/). A partir de ese momento, surgen distintas iniciativas para realizar un mejor análisis y gestión de riesgos. Una de las más interesantes es la realizada por el departamento de Seguridad de la Información del Grupo BBVA empezó a trabajar en una metodología de análisis de riesgos, denominada Casandra (Rubio, 2016), basada en la Teoría de Juegos (Cox Jr, 2009; Apt y Grädel, 2011; Barron, 2013; Alpcan, 2014) y el equilibrio de Nash (Nash, 1950; Nash, 1951; Monsalve, 2003). La Figura 1.2 muestra los objetivos principales incluidos en la metodología Casandra y remarca el objetivo principal de esta Tesis Doctoral.

Como se puede observar en la Figura 1.2, la Metodología Casandra agrupa en 7 bloques los objetivos principales incluidos en la metodología:

- ✓ Lista priorizada de Incidentes.
- ✓ Lista priorizada de Activos de Información a proteger.
- ✓ Identificación de los Problemas de los que debe protegerse la organización.
- ✓ Lista priorizada de Procesos y Tareas a desarrollar.
- ✓ Lista de Arquitecturas de Seguridad que cumplen con los objetivos de protección.
- ✓ Lista de Eventos a monitorizar ligados a cada arquitectura de protección.
- ✓ Lista priorizada de Indicadores de Gestión asociados a la dedicación de los recursos y a los objetivos alcanzados.

El objetivo principal de esta Tesis Doctoral es desarrollar un meta-modelo arquitectónico que ayude a las organizaciones a diseñar arquitecturas de seguridad empresariales que protejan sus activos de información, a la hora de desarrollar nuevos sistemas de información y/o evolucionar los sistemas actuales.

El objetivo principal de las arquitecturas de seguridad empresariales es proporcionar el diseño conceptual de la infraestructura de seguridad del sistema, junto con los mecanismos de seguridad, las políticas y los procedimientos necesarios para proteger los activos de información

incluidos en el sistema (Arconati, 2002). Este diseño conceptual une los componentes de la infraestructura de seguridad en una unidad cohesiva con el fin de proteger la información corporativa. Para ello, la arquitectura de seguridad debe determinar cuáles son los activos de información que deben ser protegidos; de qué tipos de ataques; y quien (personas) o que (sistemas) tiene acceso a ellos.

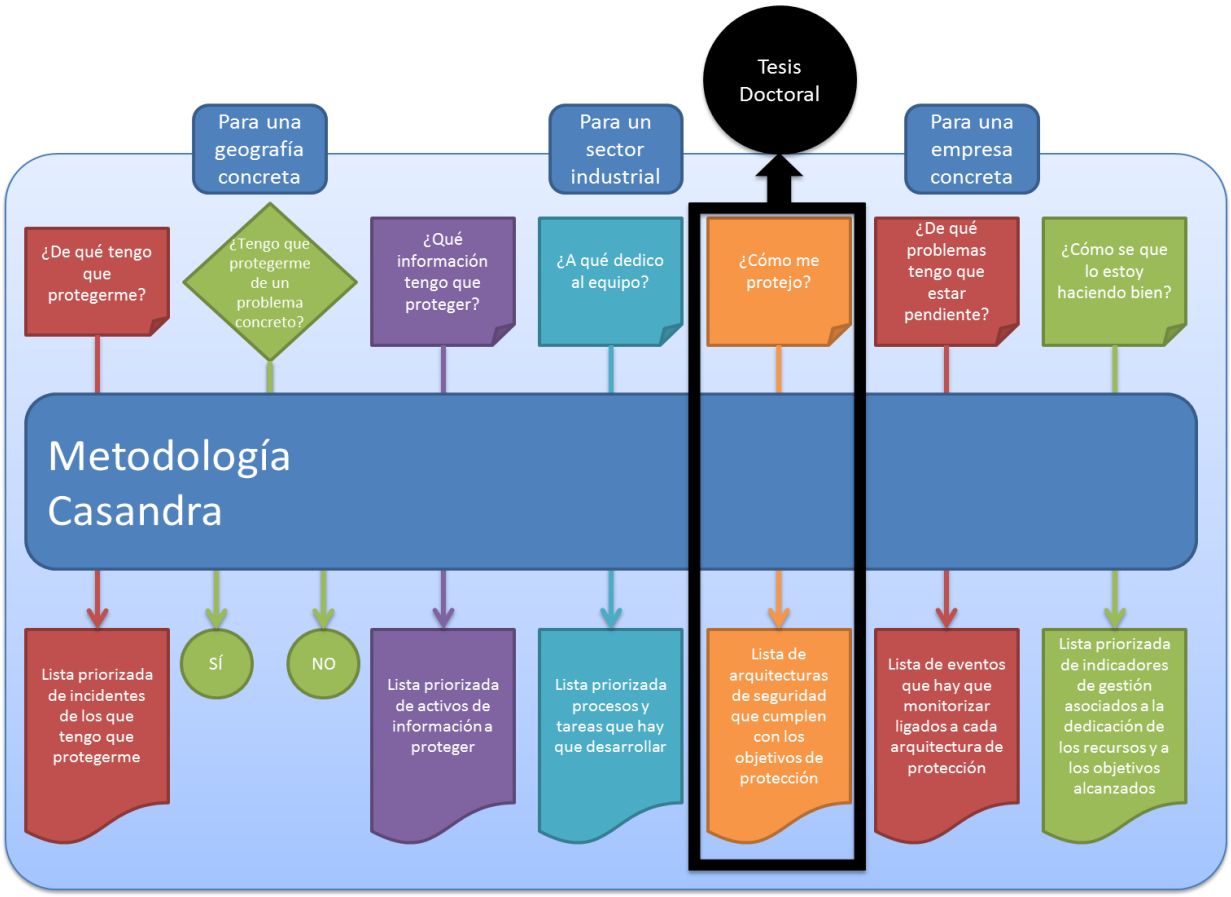


Figura 1.2. Metodología Casandra y Objetivo Principal de esta Tesis Doctoral

Debido al valor fundamental de los activos de información para las compañías, se requiere un enfoque estructurado y formal a la hora de construir sistemas seguros (Fernandez et al., 2006; Georg et al., 2009; Mouratidis, 2011; Schmidt et al., 2011). Como avalan los autores anteriores, las metodologías o modelos basados en patrones pueden proporcionar un enfoque sistemático. Por lo tanto, es recomendable el uso de modelos basados en patrones a la hora de implementar nuevos sistemas de información o evolucionar los sistemas existentes, ya que ayudan a construir

una arquitectura de seguridad de forma precisa. Los patrones de seguridad unen un amplio conocimiento acumulado acerca de la seguridad y la estructura proporcionada por los patrones (Schumacher et al., 2006). Estos patrones proporcionan las directrices para apoyar la construcción y evaluación de los mecanismos de seguridad (Fernandez et al., 2008). El uso de patrones de seguridad ayuda a incorporar los principios de seguridad en la construcción de sistemas seguros. Sin embargo, a lo largo de esta Tesis Doctoral mostramos que este tipo de patrones tienen algunas limitaciones a la hora de diseñar arquitecturas de seguridad empresariales.

Por otro lado, la Seguridad Dirigida por Modelos (*Model-Driven Security*, MDS) surgió hace más de una década como una especialización de la Ingeniería Dirigida por Modelos (*Model-Driven Engineering*, MDE) para el desarrollo de sistemas seguros. Como la MDE, los principios básicos de MDS son potenciar el rol de los modelos y aumentar el nivel de automatización, promoviendo la adopción de un enfoque conceptual (Bézivin, 2004). En el caso particular de la MDS, los modelos no están centrados en la descripción de la lógica de negocio de los sistemas, sino en capturar los requisitos de seguridad del sistema (Lucio et al., 2014). Las técnicas y herramientas basadas en MDS permiten considerar los aspectos de seguridad desde las primeras fases del desarrollo de un sistema, paliando al menos en parte el problema que resulta de la tendencia a ignorar la seguridad en el diseño de los sistemas.

MDS es una adaptación de la Arquitectura Dirigida por Modelos (MDA) (Truyen, 2006) que es el enfoque definido por el Object Management Group (OMG) para el desarrollo de software en el marco de MDE. MDA define tres puntos de vista de un sistema: (i) el Modelo Independiente de Computación (CIM), que es utilizado por los analistas de negocios, y se centra en el contexto y los requisitos del sistema sin tener en cuenta su estructura o procesamiento, (ii) el Modelo Independiente de Plataforma (PIM), que es utilizado por los arquitectos y diseñadores de software, y se centra en la capacidad operativa de un sistema fuera del contexto de una plataforma específica, y (iii) el Modelo Específico de la Plataforma (PSM), que es utilizado por los

desarrolladores y programadores de software, e incluye detalles relacionados con el sistema para una plataforma específica (Harmon, 2004).

Para cubrir el objetivo marcado, en esta Tesis Doctoral hemos definimos un nuevo enfoque de patrones para apoyar el diseño de arquitecturas de seguridad, denominado *Enterprise Security Pattern* (Moral-García et al., 2014). Ese concepto había sido utilizado anteriormente en (Romanosky, 2003), donde los autores utilizaron esta expresión para describir e identificar un conjunto de patrones existentes de seguridad centrados en el entorno empresarial. En esta Tesis hemos adoptado este nombre, ya que el objetivo de este nuevo meta-patrón es proporcionar una estrategia *top-down* basada en conceptos de Seguridad Dirigida por Modelos (MDS) para la definición de *Enterprise Security Architectures* (Arquitecturas de Seguridad Empresariales). Dentro de la estrategia *top-down*, la parte alta de la arquitectura correspondería a las necesidades de seguridad de un nuevo sistema de información plasmadas en un diagrama de contexto básico. La parte baja correspondería a la arquitectura de la solución tecnológica necesaria para proteger los activos de información incluidos en el sistema de información. Para ello, ponemos en práctica el paradigma de Arquitectura Dirigida por Modelos (MDA) incluido en MDS, a través de un proceso en el que los modelos de alto nivel, que representan el contexto del sistema, van siendo refinados hasta obtener el modelo que representa la Arquitectura de Seguridad Empresarial deseada. De esta manera, cada fase del proceso representa la definición de un nuevo modelo que añade mayor nivel de detalle al modelo anterior, incidiendo en los aspectos tecnológicos de la solución.

1.2 Patrones

Los patrones han tenido un gran auge durante las últimas décadas en el desarrollo de aplicaciones. En el ámbito de la Ingeniería del Software los patrones se aplicaron en primer lugar dentro de la comunidad orientada a objetos (Gamma E., 1995; Buschmann et al., 1996; Fowler, 1997).

Estos nuevos mecanismos tratan de buscar problemas comunes para abstraer la esencia del problema y de su solución, obteniendo soluciones fiables y probadas en marcos de trabajo. De esta manera, se consigue obtener un catálogo de pares problemas-soluciones donde cada problema tiene un nombre, un contexto y sus posibles soluciones. El simple hecho de documentar los problemas y darles un nombre permite a la comunidad de desarrollo dotarse de una jerga común, donde todos los que se refieren a un mismo problema usan el mismo nombre.

El uso de patrones constituye una ciencia de carácter estadístico, ya que los patrones por sí mismos y de modo aislado no resuelven todos los problemas posibles. Sin embargo, una colección de patrones cuidadosamente escogidos puede caracterizar un gran porcentaje, dependiendo del dominio y su amplitud, de los pares problema-solución para un dominio dado.

1.2.1 Origen

Los patrones dentro de la comunidad científica comenzaron a tomar fuerza a partir de la relevancia obtenida por el libro *Design Patterns: Elements of Reusable Object Oriented Software* (Gamma E., 1995).

Sin embargo, el origen de los patrones es anterior a este libro. El arquitecto Christopher Alexander describió el uso del término actual de patrón y escribió varios libros sobre construcción de edificaciones y planificación urbana: *Notes on the Synthesis of Form* (Alexander, 1964), *The Oregon Experiment* (Alexander, 1975), *A Pattern Language: Towns, Buildings, Constructions*

(Alexander et al., 1977) y *The Timeless Way of Building* (Alexander, 1979). Aunque los textos de Alexander tratan sobre arquitectura urbana, las ideas que en ellos se intentan captar pueden ser aplicadas a otras disciplinas, como la Ingeniería del Software.

1.2.2 Plantillas de Descripción

Existen multitud de plantillas de descripción de patrones, yendo desde un estilo narrativo (Alexander et al., 1977) hasta las descripciones basadas en plantillas formalizadas, como puede ser el caso de las plantillas presentadas en (Gamma E., 1995; Buschmann et al., 1996). El uso de plantillas está justificado, ya que dota de cierta estructura a una colección de patrones. Esta estructuración facilita su indexación y taxonomía, lo que ayuda a realizar búsquedas posteriores sobre la colección. El número de aspectos incluidos en la descripción del patrón puede variar dependiendo del dominio en el que se estén definiendo, ya que algunos aspectos pueden ser de gran interés en un dominio y pueden no ser relevantes en otro.

Los autores de (Gamma E., 1995), también conocidos como The Gang of Four, describen los siguientes aspectos como los elementos principales en la descripción de patrones. Esta plantilla de patrones es finalmente conocida en el mundo científico como GoF:

- ✓ **Nombre:** debe ser un nombre corto, significativo, que permita referir al patrón de manera no ambigua.
- ✓ **Problema:** un párrafo o frase que describa su intención. Se deben definir las metas y los objetivos que se persiguen dentro del contexto.
- ✓ **Contexto:** son las condiciones previas bajo las cuales el problema y la solución son recurrentes. El contexto se puede ver como la configuración inicial del sistema antes de que el patrón sea aplicado.
- ✓ **Fuerzas:** descripción de las virtudes relevantes y restricciones. Además, habría que describir cómo interaccionan o entran en conflicto entre sí y los objetivos que se persiguen.

- ✓ **Solución:** descripción de los pasos que nos permiten construir el producto necesitado, incluyendo las relaciones estáticas y las reglas dinámicas.
- ✓ **Ejemplos:** uno o varios ejemplos de aplicaciones del patrón que lo ilustran.
- ✓ **Contexto resultante:** descripción de la configuración del sistema una vez ha sido aplicado el patrón, incluyendo los problemas y nuevos patrones que pudieran aparecer dentro de ese nuevo contexto.
- ✓ **Fundamento:** justificación de los pasos o reglas descritas en el patrón, explicando cómo conseguir los requisitos deseados.
- ✓ **Patrones relacionados:** relaciones con otros patrones, en caso de que los hubiera.
- ✓ **Usos conocidos:** descripción de ocurrencias conocidas del patrón y sus aplicaciones dentro de sistemas existentes.

Los autores del libro *Pattern-oriented software architecture: A system of patterns* (Buschmann et al., 1996), también describen los aspectos clave en la descripción de patrones. Esta plantilla de patrones es conocida en el mundo científico como PoSA. Algunos de los aspectos incluidos en PoSA ya habían sido usados en la plantilla GoF.

- ✓ **Nombre**
- ✓ **Contexto**
- ✓ **Problema**
- ✓ **Solución**
- ✓ **Usos conocidos**
- ✓ **Patrones Relacionados**

A continuación, describimos otros de los aspectos que fueron incluidos en la plantilla de descripción PoSA, pero no habían sido definidos por GoF:

- ✓ **Ejemplo / Intención:** Un ejemplo del mundo real demostrando la existencia de un problema y la necesidad del patrón.
- ✓ **Estructura:** Una especificación detallada de los aspectos estructurales del patrón, usando la notación apropiada.
- ✓ **Dinámicas:** Escenarios típicos que describan el comportamiento del patrón a la hora de ejecutarlo.
- ✓ **Implementación:** Guías para la implementación satisfactoria del patrón en un mundo real.
- ✓ **Ejemplo Resuelto:** Descripción de cualquier elemento importante a resolver que no haya sido cubierto en la Solución, Estructura, Dinámicas o Implantación.
- ✓ **Consecuencias:** Los beneficios que tiene el patrón y cualquier posible responsabilidad.
- ✓ **Variantes:** Una breve descripción de las variantes o especializaciones de un patrón.

Otras plantillas de patrones también han sido descritas, como por ejemplo la plantilla mostrada en (Cuevas et al., 2008) que readapta la plantilla de GoF incluyendo algunas nuevas secciones o la mostrada en (Spanoudakis et al., 2007) basada en eventos de cálculo.

1.2.3 Lenguajes de Patrones

Un lenguaje de patrones está formado por un conjunto de patrones que comparten un mismo ámbito o contexto. Un lenguaje de patrones puede servir de base para un marco de trabajo, que es reusable y puede ser aplicado para resolver problemas dentro de ese ámbito o contexto.

La Figura 1.3 muestra los niveles de abstracción dentro del ámbito de sistemas de información en los que se clasifican los patrones (Díaz et al., 2005).

Figura 1.3. Niveles de Abstracción del Lenguaje de Patrones

A continuación, proporcionamos una breve explicación de cada uno de los niveles mostrados en la figura anterior:

- ✓ **Patrones arquitectónicos:** expresan estructuras de organización de sistemas de información. Proporcionan un conjunto de subsistemas predefinidos donde se especifican sus responsabilidades y se incluyen reglas y guías para organizar las relaciones entre los subsistemas.
- ✓ **Patrones de diseño:** proporcionan un esquema para refinar componentes o subsistemas de un sistema de información o las relaciones entre ellos. Este patrón describe una estructura recurrente de componentes que se comunican resolviendo un problema general de diseño en un contexto particular. Estos patrones suelen proporcionar ejemplos en un lenguaje de programación concreto, pero son independientes del lenguaje.
- ✓ **Patrones de programación, Idiomas:** son un patrón de bajo nivel, muy específico para un determinado lenguaje de programación. Este tipo de patrón describe cómo implementar aspectos particulares de componentes o sus relaciones entre ellos usando las características particulares de un lenguaje de programación dado. Un patrón de programación no puede ser reusado en otro lenguaje.

1.2.4 Características de un Patrón

Un patrón bien descrito debe proporcionar las siguientes características según Lea (Lea, 1999):

- ✓ **Encapsulación y abstracción:** cada patrón encapsula un problema bien definido y su solución en un dominio particular. Los patrones deben proporcionar fronteras claras que ayuden a separar claramente el espacio del problema del espacio de la solución.
- ✓ **Apertura y variabilidad:** los patrones deberían soportar extensiones o parametrización por parte de otros patrones, de modo que puedan colaborar para resolver problemas más complejos.
- ✓ **Generabilidad y composicionabilidad:** una vez aplicado un patrón, éste genera un contexto resultante que encaja con el contexto inicial de uno o más patrones en un lenguaje de patrones. Los siguientes patrones se aplican de manera incremental para alcanzar la solución completa.
- ✓ **Equilibrio:** cada patrón debe realizar algún tipo de balanceo entre las fuerzas y restricciones involucradas.

1.2.5 Patrones de Seguridad

El trabajo sobre patrones de seguridad ha evolucionado considerablemente en las últimas dos décadas. En 1997, Yoder y Barcalow escribieron el primer artículo publicado sobre patrones de seguridad (Yoder y Barcalow, 1997). Ellos incluyeron una variedad de patrones en diferentes aspectos de seguridad, utilizando la plantilla de Gang of Four (GoF) para describir los aspectos de seguridad y estructurar sus patrones.

Antes de ellos, al menos 2 artículos (Fernandez et al., 1994; Essmayr et al., 1996) habían mostrado modelos orientados a objetos sin llamarlos “patrones”. Un año más tarde, 1998, dos

contribuciones más sobre patrones de seguridad fueron publicadas: un lenguaje de patrón para criptografía (Braga et al., 1998), y un patrón para control de acceso (Neves y Garrido, 1998). Muchos otros patrones de seguridad aparecieron secuencialmente durante los siguientes años (Welch, 1999; Jaworski y Perrone, 2000; Fernandez y Pan, 2001; Kodituwakku et al., 2001; Fernandez, 2002; Schumacher, 2003), llegando a tener una sustancial colección.

Los patrones han sido probados satisfactoriamente en muchas áreas de desarrollo de software y parecen ser particularmente útiles para el desarrollo de sistemas seguros. En el trabajo de Schumacher et al. (Schumacher et al., 2006) se muestran las ventajas más importantes que los patrones de seguridad ofrecen en la construcción de sistemas seguros:

- ✓ Los patrones agrupan conocimiento básico de seguridad de forma estructurada y fácilmente entendible.
- ✓ La representación del patrón es familiar a los desarrolladores software e ingenieros de sistemas, una parte importante de su audiencia.
- ✓ Los patrones ya eran utilizados para capturar el conocimiento sobre la ingeniería de sistemas. El uso de los patrones para capturar el conocimiento de la seguridad ayuda a mejorar la integración de la seguridad en los sistemas de información de las compañías.

En esta tesis doctoral se ha realizado una revisión sistemática de patrones de seguridad (véase Sección 3.2) con el objetivo de analizar profundamente cómo de útiles son los patrones de seguridad actuales a la hora de tomar decisiones de diseño sobre problemas que ya han sido solucionados anteriormente.

Después de realizar esta revisión, hemos podido llegar a la conclusión que los patrones de seguridad son un método útil para presentar soluciones comunes, pero el conjunto de patrones que hemos estudiado no siguen una guía en común a la hora de documentar las soluciones. Debido a este hecho, es complicado llegar a obtener una clasificación homogénea de patrones de seguridad. Con esta revisión sistemática, también hemos podido concluir que los

patrones de seguridad proporcionan un conjunto de guías para apoyar la construcción y evaluación de mecanismos de seguridad, las cuales ayudan a incorporar los principios de seguridad a la hora de construir sistemas seguros. Sin embargo, hemos apreciado algunas limitaciones que nos han llevado a definir un nuevo meta-patrón de seguridad, llamado Enterprise Security Patterns, el cual utiliza los patrones de seguridad para facilitar el diseño de arquitecturas de seguridad de grandes sistemas de información (véase Capítulo 4).

1.3 Minería de Patrones

Una de las partes importantes de esta tesis doctoral es proporcionar una forma o camino de identificar soluciones probadas para problemas de seguridad concretos. El concepto de *Minería de Patrones* puede ser definido como el proceso o técnica de identificación de soluciones probadas (Schumacher, 2003). En esta sección mostramos el origen de la minería de los patrones y un apartado específico sobre minería de patrones de seguridad.

1.3.1 Origen de la Minería de Patrones

En 1997, Norman Kerth y Ward Cunningham consideraron tres enfoques generales a la hora de realizar minería de patrones (Kerth y Cunningham, 1997). El enfoque introspectivo, el enfoque basado en artefactos y el enfoque sociológico.

- ✓ **Enfoque Introspectivo:** Los investigadores analizan sistemas que ellos mismos han construido (o participado en su construcción) para intentar descubrir patrones basándose en su experiencia. Este enfoque permite descubrir nuevos patrones a través de experiencias individuales.
- ✓ **Enfoque basado en Artefactos:** Los investigadores analizan sistemas construidos por otras personas que intentan resolver un problema similar. Este enfoque nos permite encontrar puntos en común entre diferentes sistemas que intentan resolver el mismo problema.
- ✓ **Enfoque Sociológico:** Los investigadores entrevistan a las personas que han construido sistemas similares para conocer como resolvieron problemas particulares. Este enfoque nos permite encontrar problemas recurrentes en los sistemas analizados.

En 1998, Linda Rising y David DeLano dividen también en tres enfoques generales las actividades asociadas a la minería de patrones (Rising y Delano, 1998). Enfoque Individual, contribuciones *Second-hand* y Workshops de Minería de Patrones.

- ✓ **Enfoque Individual:** Los investigadores escriben un patrón desde su propia experiencia. Este enfoque es similar al enfoque introspectivo descrito en (Kerth y Cunningham, 1997).
- ✓ **Enfoque *Second-hand*:** Los investigadores obtienen información entrevistando a expertos en la materia a analizar o analizando sistemas construidos para resolver problemas similares. Este enfoque es la unión del enfoque sociológico y el enfoque basado en artefactos descrito en (Kerth y Cunningham, 1997).
- ✓ **Workshops de Minería de Patrones:** Un grupo de investigadores expertos en el tema a analizar trabajan juntos para discutir sobre los nuevos campos a investigar dentro de un ámbito específico.

Después de analizar los enfoques y categorías presentados en esta sección, llegamos a la conclusión que tanto el enfoque sociológico y basado en artefactos (Kerth y Cunningham, 1997), como los workshops de minería de patrones (Rising y Delano, 1998) podrían ser adaptados a la minería de patrones de seguridad.

1.3.2 Minería de Patrones de Seguridad

Como hemos observado en la sección anterior los enfoques para descubrir nuevos patrones pueden ser diferentes. Después de realizar una revisión sistemática sobre Minería de Patrones de Seguridad (véase Sección 3.3), hemos identificado que los autores de patrones de seguridad utilizan combinaciones de enfoques para llegar a documentar nuevos patrones, pero no hemos identificado ninguna metodología o enfoque específico de la minería de patrones de seguridad.

1.3.2.1 Estándares y Guías de Seguridad de la Información

Estándares de seguridad de la información suelen ser utilizados como fuentes informativas dentro de la minería de patrones de seguridad (Schumacher et al., 2006). A continuación, mostramos algunas de sus ventajas:

- ✓ El uso de estándares puede ser una fuente de inspiración. Los expertos normalmente conocen la solución a un problema dado. Sin embargo, su conocimiento está basado en la experiencia y obtener información precisa de entrevistas con ellos no es trabajo fácil. Los estándares de seguridad ayudan a expresar los aspectos más difíciles de la seguridad, permitiendo a los autores a concentrarse completamente en la solución.
- ✓ Los estándares de seguridad ayudan a lograr una terminología estandarizada en la descripción de los patrones de seguridad. Los estándares también ayudan a asegurar que los requisitos de seguridad son cumplidos por el patrón.
- ✓ Los estándares de seguridad son públicos, por lo tanto, el *retorno o feedback* recibido por los expertos en seguridad de la información ayuda a mejorar de forma rápida cualquier posible error o brecha. Las mejoras son fácilmente identificadas.

En el resto de esta sección, incluimos los estándares más relevantes y las más importantes guías y marcos de trabajo en el ámbito de la seguridad de la información.

1.3.2.1.1 ISO/IEC 27000-series

La serie de normas ISO/IEC 27000 (ISO/IEC 27000, http://www.iso.org) son estándares de seguridad publicados por la Organización Internacional para la Estandarización (ISO) y la Comisión Electrotécnica Internacional (IEC).

La serie contiene las mejores prácticas recomendadas en Seguridad de la información para desarrollar, implementar y mantener especificaciones para los Sistemas de Gestión de la Seguridad de la Información (SGSI).

- ✓ **ISO/IEC 27000:** Vocabulario estándar para el SGSI.
- ✓ **ISO/IEC 27001:** Norma que especifica los requisitos para la implantación del SGSI. Es la certificación que deben obtener las organizaciones para mostrar su madurez en la gestión de la seguridad de la información. Es la norma más importante de la familia. Adopta un enfoque de gestión de riesgos y promueve la mejora continua de los procesos.
- ✓ **ISO/IEC 27002:** Previamente BS 7799 Parte 1 y la norma ISO/IEC 17799. Es código de buenas prácticas para la gestión de seguridad de la información. Fue publicada en julio de 2005 como ISO 17799:2005 y recibió su nombre oficial ISO/IEC 27002:2005 el 1 de julio de 2007.
- ✓ **ISO/IEC 27003:** Directrices para la implementación de un SGSI. Es el soporte de la norma ISO/IEC 27001. Publicada el 1 de febrero del 2010.
- ✓ **ISO/IEC 27004:** Métricas para la gestión de seguridad de la información. Es la norma que proporciona recomendaciones de quién, cuándo y cómo realizar mediciones de seguridad de la información. Publicada el 7 de diciembre del 2009.
- ✓ **ISO/IEC 27005:** Gestión de riesgos en seguridad de la información. Es la norma que proporciona recomendaciones y lineamientos de métodos y técnicas de evaluación de riesgos de Seguridad en la Información, en soporte del proceso de gestión de riesgos de la norma ISO/IEC 27001. Es la más relacionada a la actual British Standard BS 7799 parte 3. Publicada en junio de 2008.
- ✓ **ISO/IEC 27006-2007:** Requisitos para la acreditación de las organizaciones que proporcionan la certificación de los sistemas de gestión de la seguridad de la información. Esta norma específica requisitos específicos para la certificación de SGSI y es usada en conjunto con la norma 17021-1, la norma genérica de acreditación.
- ✓ **ISO/IEC 27007:** Guía para auditar los SGSI.
- ✓ **ISO/IEC 27799-2008:** Guía para implementar ISO/IEC 27002 en la industria de la salud.

- ✓ **ISO/IEC 27035-2011:** Este standard hace foco en las actividades de detección, reporte y evaluación de incidentes de seguridad y sus vulnerabilidades.

1.3.2.1.2 Common Criteria (CC) – ISO/IEC 15408

Los Criterios Comunes para la Evaluación Tecnológica de la Seguridad de la Información, abreviado como *Common Criteria* o *CC*, son un estándar internacional (ISO/IEC 15408) para la certificación de la seguridad computacional (CC, http://www.commoncriteriaportal.org/). Los *Common Criteria* tienen su origen en 1990 y surgen como resultado de la armonización de los criterios sobre seguridad de productos software ya utilizados por diferentes países con el fin de que el resultado del proceso de evaluación pudiese ser aceptado en múltiples países.

Los *Common Criteria* permiten comparar los resultados entre evaluaciones de productos independientes. Para ello, se proporciona un conjunto común de requisitos funcionales para los productos de Tecnologías de la Información. Estos productos pueden ser hardware, software o firmware. El proceso de evaluación establece un nivel de confianza en el grado en el que el producto satisface la funcionalidad de seguridad de estos productos y ha superado las medidas de evaluación aplicadas.

Los *Common Criteria* definen los requisitos de seguridad acorde a 7 niveles de Garantía de Evaluación (*Evaluation Assurance Level*, *EAL*). La Tabla 1-1 muestra los Niveles de Garantía de Evaluación en comparación con el rigor del proceso que debe ser asignado al producto.

Nivel de Garantía de Evaluación	Rigor del Proceso para despliegue de un Producto
EAL 1	Funcionalmente Testeado
EAL 2	Estructuralmente Testeado
EAL 3	Metodológicamente Testeado y Revisado
EAL 4	Metodológicamente Diseñado, Testeado y Revisado
EAL 5	Semi-Formalmente Diseñado y Testeado
EAL 6	Semi-Formalmente Verificado, Diseñado y Testeado
EAL 7	Formalmente Diseñado y Testeado

Tabla 1-1. ISO 15408, Niveles de Garantía de Evaluación (EALs)

Los *Common Criteria* son útiles como guía para el desarrollo, evaluación o adquisición de productos de Tecnologías de Información que incluyan alguna función de seguridad. La lista de productos certificados se encuentra disponible en la web de Common Criteria (http://www.commoncriteriaportal.org/products\).

1.3.2.1.3 ISO/IEC 21827, SSE capability maturity model (SSE-CMM®)

El modelo de capacidad y madurez en la ingeniería de seguridad de sistemas es un modelo derivado del modelo de madurez del software CMM (CMM, 1995) y orientado hacia procesos. Describe las características esenciales de los procesos que deben existir en una organización para asegurar una buena seguridad en los sistemas.

Este es un modelo que pretende servir como:

- ✓ Herramienta para que las organizaciones evalúen las prácticas de ingeniería de seguridad y definan mejoras a las mismas.
- ✓ Mecanismo estándar para que las empresas puedan evaluar la capacidad de los proveedores de ingeniería de seguridad.
- ✓ Base para la creación de un mecanismo de evaluación y certificación.

SSE-CMM (norma ISO/IEC 21827) es utilizada para diseñar e implantar los procesos de seguridad, medir su capacidad y establecer planes de mejora en las áreas de proceso de interés para la organización. SSE-CMM tiene dos dimensiones, “dominio” y “capacidad”.

- ✓ **La dimensión del dominio**: Comprende las prácticas que de forma colectiva definen la ingeniería de la seguridad. A estas prácticas se las denomina “practicas base”. SSE-CMM contiene alrededor de 60 prácticas base de seguridad, organizadas en 11 áreas de proceso que cubren todas las áreas principales de la ingeniería de la seguridad.

- ✓ **La dimensión de la capacidad**: Comprende las prácticas que indican capacidad de gestión y de institucionalización del proceso. Se denominan "practicas genéricas", ya que se aplican en un amplio rango de dominios. Las prácticas genéricas se agrupan en áreas lógicas denominadas "características comunes", que están organizadas en "niveles de capacidad", los cuales representan el aumento de la capacidad de la organización.

1.3.2.1.4 NIST SP 800-53

El NIST (National Institute of Standards and Technology) es una agencia del Departamento de Comercio de los Estados Unidos que promueve la aprobación de estándares sobre diversos productos y servicios relacionados con la tecnología. Entre todas sus publicaciones, son particularmente relevantes las pertenecientes a la serie SP 800, destinados a la Seguridad de la Información. Esta serie se empezó a publicar en 1990 con la intención de proporcionar una identidad propia a las publicaciones en materia de seguridad, y contiene guías, pautas y propuestas tanto para el ámbito empresarial como industrial, gubernamental o académico.

La serie NIST SP 800 es un conjunto de documentos de libre descarga que se facilita desde el gobierno federal de los Estados Unidos, que describe las políticas de seguridad informática, procedimientos y directrices, que cubren hasta 256 salvaguardas y que las organiza en 18 categorías.

Algunos de los documentos más interesantes son:

- ✓ **SP 800-61:** directrices para detectar, analizar, priorizar y gestionar los incidentes de responder a ellas de forma eficiente y eficaz.
- ✓ **SP 800-50:** pautas para el diseño, desarrollo, implantación y evaluación de un programa de sensibilización y formación.
- ✓ **SP 800-116:** es el riesgo basado en la selección de los mecanismos de autenticación apropiados para gestionar el acceso físico.

- ✓ **SP 800-46:** prácticas para mitigar los riesgos asociados con las tecnologías utilizadas para el teletrabajo.
- ✓ **SP 800-122:** orientaciones para la protección de la confidencialidad de la información de identificación de personal con el apoyo de los sistemas de información.
- ✓ **SP 800-161:** una guía para identificar, evaluar, seleccionar e implantar la gestión de riesgos y controles para gestionar los riesgos e la cadena de suministro.
- ✓ **SP 800-92:** orientación sobre el desarrollo, implantación y mantenimiento de las prácticas de gestión de riesgos eficientes para apoyo.
- ✓ **SP 800-88:** recomendaciones para la implantación de un programa de saneamiento de los medios, teniendo en cuenta las técnicas y controles para la desinfección y eliminación de la información confidencial.
- ✓ **SP 800-83:** orientación sobre la prevención de ataques de este tipo y responder a los incidentes de malware.
- ✓ **SP 800-64:** descripción de las funciones de seguridad y responsabilidades clave necesarios en el desarrollo de los sistemas de información, y la información sobre la relación entre la seguridad de la información y el ciclo de vida del software de desarrollo.
- ✓ **SP 800-45:** proporciona prácticas de seguridad para el diseño, implantación y sistemas de correo electrónico de funcionamiento en las redes públicas y privadas de apoyo.
- ✓ **SP 800-44:** presenta las prácticas de seguridad para el diseño, implantación y operación de los servidores web de acceso público e infraestructura de la red relacionada.
- ✓ **SP 800-41:** proporciona una guía durante el desarrollo de las políticas y la selección de firewall, configuración, prueba, implantación y administración de servidores de seguridad.

- ✓ **SP 800-34:** proporciona información sobre el sistema de información de la planificación e contingencia y otros tipos de planes de seguridad y emergencia de contingencia.

1.3.2.1.5 COBIT 5 for Information Security

El estándar Cobit (Control Objectives for Information and related Technology) ofrece un conjunto de mejores prácticas para la gestión de los sistemas de información de las organizaciones.

COBIT 5 (ISACA, 2012) provee de un marco de trabajo integral que ayuda a las empresas a alcanzar sus objetivos para el gobierno y la gestión de las TI corporativas. Dicho de una manera sencilla, ayuda a las empresas a crear el valor óptimo desde la tecnología de la información (TI) manteniendo el equilibrio entre la generación de beneficios y la optimización de los niveles de riesgo y el uso de recursos. COBIT 5 permite que las TI se gobiernen y gestionen de un modo holístico para toda la empresa, abarcando al negocio completo de principio a fin y a las áreas funcionales de responsabilidad de TI, considerando los intereses relacionados con TI de los grupos de interés internos y externos.

El objetivo principal de Cobit consiste en proporcionar una guía a alto nivel sobre puntos en los que establecer controles internos con tal de:

- ✓ Asegurar el buen gobierno, protegiendo los intereses de los stakeholders (clientes, accionistas, empleados, etc.)
- ✓ Garantizar el cumplimiento normativo del sector al que pertenezca la organización.
- ✓ Mejorar la eficacia y eficiencia de los procesos y actividades de la organización.
- ✓ Garantizar la confidencialidad, integridad y disponibilidad de la información.

Utilizar COBIT 5 para Seguridad de la Información proporciona a la empresa una serie de capacidades relacionadas con la seguridad de la información que pueden resultar en beneficios como:

- ✓ Menor complejidad y mayor coste-beneficio debido a una mejorada y más fácil integración de estándares buenas prácticas y/o guías específicas del sector de seguridad de la información.
- ✓ Mayor satisfacción de usuario con la estructura y resultados de seguridad de la información.
- ✓ Mejor integración de la seguridad de la información en la empresa.
- ✓ Toma de decisiones de riesgo con conocimiento y conciencia del riesgo.
- ✓ Mejor prevención, detección y recuperación.
- ✓ Reducción (del impacto) de los incidentes de seguridad de la información.
- ✓ Soporte mejorado a la innovación y la competitividad.
- ✓ Mejor gestión de los costes relacionados con la función de seguridad de la información.
- ✓ Mayor conocimiento de la seguridad de la información.

Este modelo es uno de los más ampliamente utilizados para el gobierno de los sistemas de información, incluyendo la gestión de controles internos usados para satisfacer los requerimientos legales establecidos. Su jerarquía de principios, catalizadores y procesos provee una base sólida para la realización de auditorías informáticas en las organizaciones que lo implementan. El marco COBIT 5 se construye sobre 5 principios básicos clave para el gobierno y la gestión de las TI empresariales, como se puede ver en la Figura 1.4.

Figura 1.4. Principios de COBIT 5

1.3.2.1.6 CLASP

Comprehensive, Lightweight Application Security Process (CLASP) es un conjunto de componentes de proceso basados en roles y dirigidas por actividad guiados por mejores prácticas formalizadas. CLASP está diseñada para ayudar a los equipos de desarrollo de software a construir la seguridad en las primeras etapas del ciclo de vida de desarrollo existentes y nuevos software de una manera estructurada, repetible y medible (Graham, 2006).

El proceso CLASP es presentado a través de cinco perspectivas de alto nivel llamadas Vistas CLASP. Estas vistas permitirán que los usuarios CLASP comprendan rápidamente el proceso CLASP, incluyendo cómo los componentes del proceso CLASP interactúen y cómo aplicarlos a un ciclo de vida de desarrollo de software específico. Éstas son las vistas CLASP (ver Figura 1.5):

- ✓ **Vistas de Conceptos**. Esta vista proporciona una introducción de alto nivel a CLASP con una breve descripción de la interacción de las 5 vistas CLASP, las 7 mejores prácticas CLASP, la taxonomía CLASP, la relación de CLASP con las políticas de seguridad, y una secuencia de la aplicación de los componentes de proceso CLASP.
- ✓ **Vista basada en rol**. Esta vista contiene introducciones basados en roles del proceso.

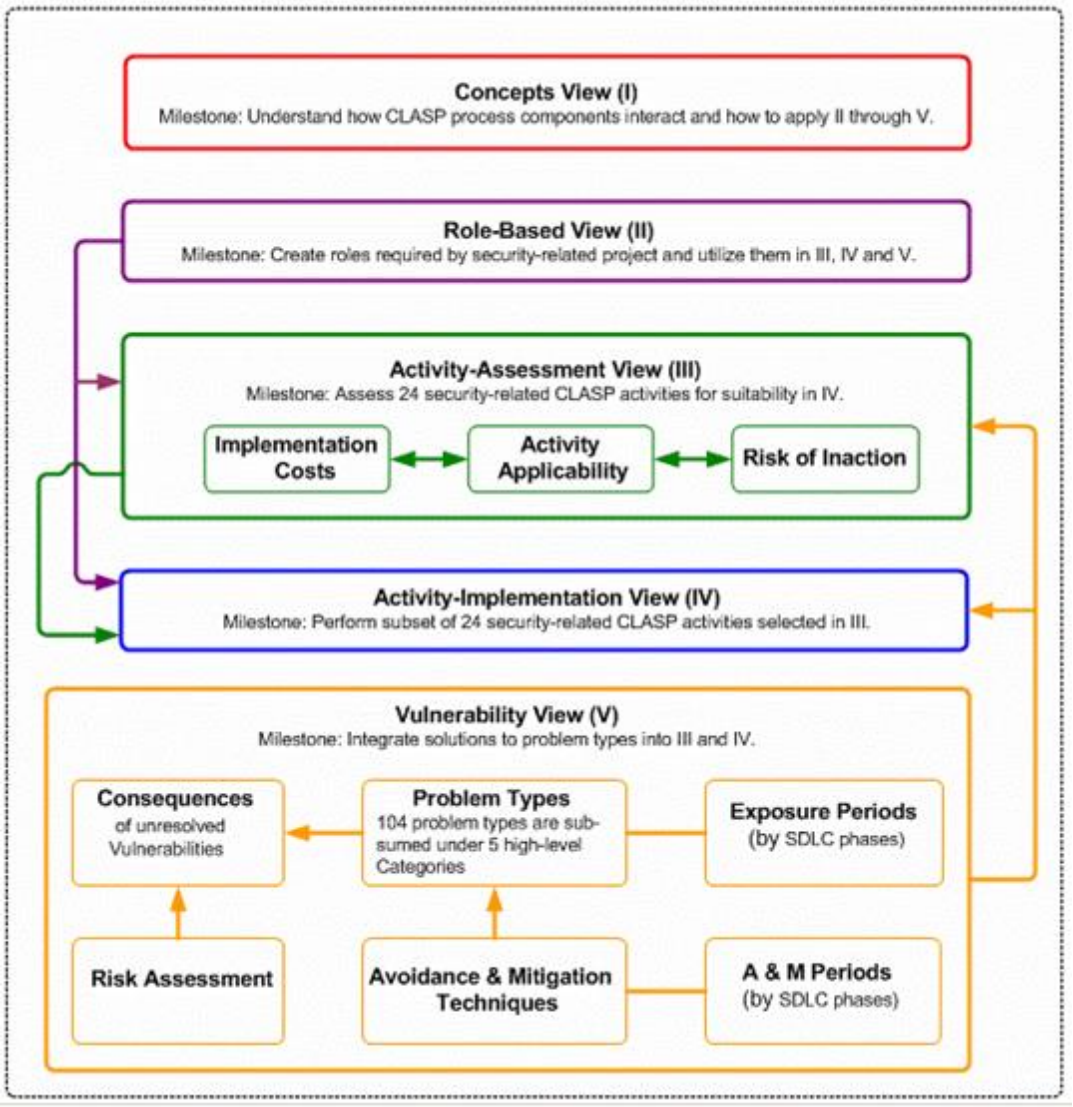


Figura 1.5. Vistas CLASP y sus interacciones

- ✓ **Vista de Evaluación de Actividad**. Esta vista ayuda a los gerentes de proyecto a evaluar la conveniencia de las 24 actividades CLASP y seleccionar un subconjunto de ellas. CLASP ofrece dos mapas ejemplos (heredada y nuevo) para ayudar a seleccionar las actividades correspondientes.
- ✓ **Vista de Implementación de Actividad**. Esta vista contiene las 24 actividades CLASP relacionadas con la seguridad que pueden integrarse en un proceso de desarrollo de software. La fase de actividades de SDLC se traduce dentro de software ejecutable

cualquier subconjunto de las 24 actividades relacionadas con la seguridad evaluadas y aceptadas en la evaluación de la actividad.

- ✓ **Vista de vulnerabilidad**. Esta vista contiene un catálogo de los 104 "tipos de problemas" subyacentes identificadas por CLASP que forman la base de las vulnerabilidades de seguridad en el código fuente de la aplicación. CLASP divide a los 104 tipos de problemas en cinco categorías de alto nivel. Un tipo de problema individual en sí mismo, a menudo no es una vulnerabilidad de seguridad, con frecuencia, es una combinación de problemas que conducen a una vulnerabilidad en el código fuente.

1.3.2.1.7 IT Baseline Protection Manual

El *IT Baseline Protection Manual* (BSI, 2000), elaborado por la Oficina Federal de Seguridad de la Información alemana, es una metodología que puede ser utilizada para identificar e implementar medidas o controles de seguridad de la información en una organización. Este manual contiene medidas de seguridad, consejos de aplicación y ayudas para numerosas configuraciones de tecnologías que normalmente se encuentran en los sistemas de información actuales. Este manual está destinado a ayudar en la solución de problemas comunes de seguridad, elevando el nivel de seguridad de los sistemas de información y simplificando la creación de políticas de seguridad.

Este manual ha sido creado de modo que pueda ser actualizado de forma continua y extendida. Es revisado cada seis meses para incorporar nuevas sugerencias de mejora, material adicional y reflejar los últimos desarrollos de sistemas de información.

1.3.2.2 Fuentes de Arquitecturas de Seguridad Empresariales

Además de los estándares de seguridad de información internacionales, hay varias compañías y organizaciones que ofrecen fuentes de seguridad de la información para arquitecturas empresariales. A continuación, mostramos algunas de las fuentes más relevantes.

1.3.2.2.1 National Institute of Standards and Technology

El Instituto Nacional de Normas y Tecnología (NIST por sus siglas en inglés, *National Institute of Standards and Technology*), llamada entre 1901 y 1988 Oficina Nacional de Normas (NBS por sus siglas del inglés *National Bureau of Standards*), es una agencia de la Administración de Tecnología del Departamento de Comercio de los Estados Unidos.

Figura 1.6. NIST Logo

La misión de este instituto es promover la innovación y la competencia industrial en Estados Unidos mediante avances en metrología, normas y tecnología de forma que mejoren la estabilidad económica y la calidad de vida.

Como parte de esta misión, los científicos e ingenieros del NIST continuamente refinan la ciencia de la medición (*metrología*) creando una ingeniería precisa y una manufacturación requerida para la mayoría de los avances tecnológicos actuales. También están directamente involucrados en el desarrollo y pruebas de normas hechos por el sector privado y agencias de gobierno. El progreso e innovación tecnológica de Estados Unidos dependen de las habilidades del NIST, especialmente si hablamos de cuatro áreas: biotecnología, nanotecnología, tecnologías de la información y fabricación avanzada.

Una de las divisiones del NIST es la División de Seguridad de la Computación (CSD por sus siglas en inglés, *Computer Security Division*), la cual publica mucha información sobre seguridad de la información (NIST, http://csrc.nist.gov/). El *website* de esta división (http://csrc.nist.gov/)

ofrece conjuntos de buenas prácticas de seguridad, guías de implementación, políticas de seguridad y otros documentos que pueden ayudar a la hora de escribir patrones de seguridad.

1.3.2.2.2 SANS Institute

El Instituto SANS (de sus siglas en inglés, SysAdmin Audit, Networking and Security) es una institución con ánimo de lucro fundada en 1989, con sede en Bethesda (Maryland, Estados Unidos) que agrupa a 165.000 profesionales de la seguridad informática (consultores, administradores de sistemas, universitarios, agencias gubernamentales, etc.)

Figura 1.7. SANS Institute Logo

Sus principales objetivos son:

- ✓ Reunir información sobre todo lo referente a seguridad informática (sistemas operativos, *routers*, *firewalls*, aplicaciones, *Intrusion Detection Systems*, etc.)
- ✓ Ofrecer capacitación y certificación en el ámbito de la seguridad informática

Igualmente, el SANS Institute es una universidad formativa en el ámbito de las tecnologías de seguridad. Es una referencia habitual en la prensa sobre temas de auditoría informática.

1.3.2.2.3 CERT

Un Equipo de Respuesta ante Emergencias Informáticas (CERT, del inglés *Computer Emergency Response Team*) es un centro de respuesta a incidentes de seguridad en tecnologías de la información. Se trata de un grupo de expertos responsable del desarrollo de medidas preventivas y reactivas ante incidencias de seguridad en los sistemas de información. Un CERT

estudia el estado de seguridad global de redes y ordenadores y proporciona servicios de respuesta ante incidentes a víctimas de ataques en la red, publica alertas relativas a amenazas y vulnerabilidades y ofrece información que ayude a mejorar la seguridad de estos sistemas. El primer CERT fue creado por la Universidad Carnegie Mellon (SEI, 1988) en respuesta al incidente del gusano Morris.

Figura 1.8. US-CERT Logo

También se puede utilizar el término CSIRT (Computer Security Incident Response Team, Equipo de Respuesta ante Incidencias de Seguridad) para referirse al mismo concepto. De hecho, el término CSIRT es el que se suele usar en Europa en lugar del término protegido CERT, que está registrado en EEUU por CERT Coordination Center (CERT/CC).

1.4 Ingeniería Dirigida por Modelos (MDE)

A lo largo de la historia de la Ingeniería del Software, los desarrolladores e investigadores han realizado grandes esfuerzos con el objetivo de simplificar el proceso de desarrollo de software a partir del aumento gradual del nivel de abstracción al que se diseña y se desarrolla. Así, por ejemplo, la programación en lenguaje ensamblador dio paso al empleo de lenguajes de programación como C o Fortran, que elevaban el nivel de abstracción al que se programaba y diseñaba. Esta tendencia ha continuado hasta nuestros días y así han surgido otros paradigmas como la orientación a objetos o la orientación a aspectos (Schmidt, 2006; Völter, 2011).

El último paso natural en esta tendencia de aumentar el nivel de abstracción al que se construye el software es la Ingeniería Dirigida por Modelos (MDE, *Model-Driven Engineering*) (Bézivin, 2004).

Con la llegada de MDE, el escenario en el que se construye el software ha cambiado drásticamente: los desarrolladores han pasado de centrarse en la codificación de la solución, a centrarse en el modelado del problema. Sin embargo, conviene mencionar que la idea de usar modelos a lo largo del proceso de desarrollo no es en absoluto nueva. Con anterioridad a la llegada de MDE, los modelos ya estaban presentes en tareas como la documentación o el diseño del sistema e incluso en algunos casos, a partir de estos modelos era posible generar la estructura o esqueleto del sistema. Un ejemplo de este escenario podría ser *Rational Rose* (IBM, https://www.ibm.com), que a partir de modelos UML es capaz de generar la especificación de las clases recogidas en dichos modelos, lo que podría utilizarse como punto de partida para implementar el resto del sistema. Sin embargo, una vez que cumplían su función, tradicionalmente limitada a las fases de análisis y diseño (de alto nivel), estos modelos eran descartados en fases sucesivas del proceso de desarrollo y en la mayoría de los casos no eran actualizados para reflejar cambios posteriores.

Los principales cambios o novedades que aporta MDE son: el rol principal que juegan los modelos en el proceso de desarrollo y el aumento del nivel de automatización del proceso en sí mismo. De hecho, una de las mejores formas de aprovechar al máximo las ventajas que ofrece MDE, en términos de desarrollos más rápidos y baratos, pasa por automatizar al máximo posible el proceso de desarrollo (Atkinson y Kuhne, 2003; Flore, 2003).

El impacto de MDE puede verse reflejado en nuevas metodologías de Desarrollo de Software Dirigido por Modelos (DSDM o en inglés: MDSD, *Model-Driven Software Development*) (Stahl et al., 2006; Völter, 2009). Estas metodologías se centran en potenciar el nivel de automatización en la construcción de software, proponiendo el modelado del sistema a diferentes niveles de abstracción, de forma que se construyan modelos cada vez más detallados hasta que estos puedan ser utilizados como planos de menor nivel para la generación de código o en algunos casos, constituyan el sistema ejecutable en sí mismo (Edwards et al., 2008; Fernández-Medina et al., 2009; Wu et al., 2009; Abdelhalim et al., 2010; Li et al., 2010; Dausend y Raiser, 2011).

En este contexto, uno de los puntos clave para aumentar el nivel de automatización es el empleo de transformaciones de modelo a modelo (M2M, *model-to-model*) y de transformaciones de modelo a código (M2T, *model-to-text*) que actúan como eslabones, uniendo cada paso del proceso (Gerber et al., 2002; Selic, 2003; Bézivin, 2004; Tratt, 2005). Así, las primeras se utilizan para convertir uno (o varios) modelos del sistema en otro modelo (u otros), mientras que las segundas se utilizan para la serialización de los modelos en el código que implementa el sistema.

Una de las interpretaciones más adoptadas de los principios de MDE es la realizada por el consorcio internacional Object Management Group (OMG): Arquitectura Dirigida por Modelos (MDA, *Model-Driven Ar*chitecture) (Rankel, 2002; Flore, 2003; Kleppe et al., 2003; OMG, 2003; Mellor et al., 2004; Guttman y Parodi, 2007; Selic, 2008; Watson, 2008). MDA define una arquitectura basada en tres niveles conceptuales o niveles de abstracción: CIM (*Computation Independent Model*), PIM (*Platform Independent Model*) y PSM (*Platform Specific Model*).

Utilizando técnicas MDA y tres niveles de abstracción, mejoramos los procesos de desarrollo de software, iniciándolos desde una perspectiva del negocio (ver Figura 1.9).

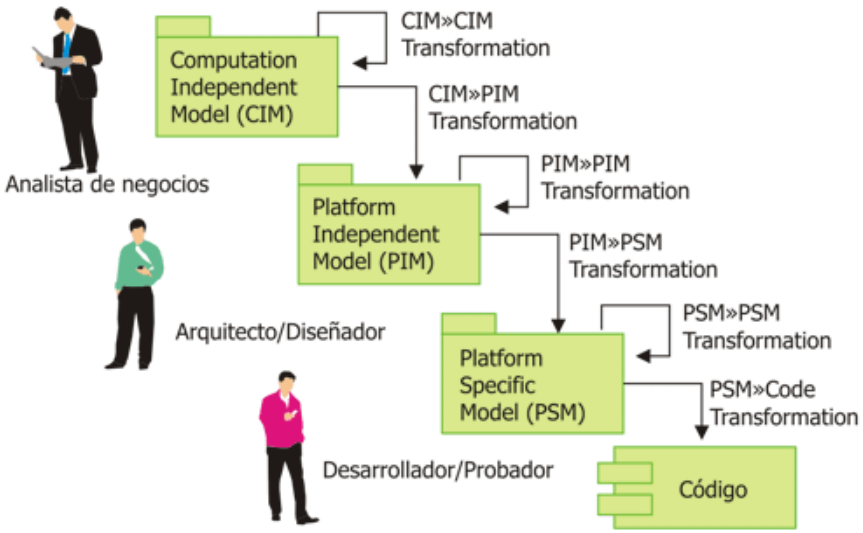


Figura 1.9. Enfoque MDA - Model Driven Architecture

Como puede ser observado en la figura anterior, los detalles del dominio de la aplicación se modelan mediante Modelos Independientes de Computación (nivel CIM) que sirven de conexión entre los expertos del negocio y los desarrolladores de sistemas. La funcionalidad y estructura del sistema, abstrayéndose de los detalles tecnológicos de la plataforma que soportará al sistema se modelan mediante Modelos Independientes de Plataforma (nivel PIM). Estos modelos podrán ser refinados tantas veces como sea necesario, con el fin de obtener una descripción del sistema con el nivel de claridad y abstracción óptimo. Para combinar estas especificaciones independientes de plataforma con los detalles específicos de la plataforma en la que se implementará el sistema final, se utilizan Modelos Específicos de Plataforma (nivel PSM). Partiendo de diferentes modelos PSM se puede obtener distintas implementaciones de un mismo sistema. Este nivel PSM, a su vez puede contener diferentes modelos que representan distintos grados de abstracción, pudiéndose agrupar en modelos que representan elementos generales a todas las plataformas y en modelos que representan elementos dependientes de una plataforma específica. Esta distinción da lugar a la consideración de otro nivel de abstracción,

denominado PDM (*Platform Dependent Models*) (Poole, 2001; Gervais, 2002; OMG, 2003). Adicionalmente, el código que implementa el sistema puede ser considerado otro modelo de más bajo nivel de abstracción (OMG, 2003).

1.5 Hipótesis y Objetivos

La principal hipótesis de esta tesis doctoral es:

Es posible sistematizar el diseño de arquitecturas de seguridad para desarrollar sistemas de información seguros.

En consonancia con la hipótesis anterior, el objetivo global queda definido como:

DEFINIR UN META-MODELO QUE SISTEMATICE EL DISEÑO DE ARQUITECTURAS DE SEGURIDAD PARA DESARROLLAR SISTEMAS DE INFORMACIÓN SEGUROS

En función de este objetivo principal se plantean los siguientes objetivos parciales:

1. Estudio del estado del arte de las propuestas existentes relacionadas con los patrones de seguridad, la minería de patrones de seguridad, así como marcos de trabajo y procesos, considerando las limitaciones y aportaciones de dichas propuestas.

2. Estudio del estado del arte de las propuestas existentes relacionadas con la Seguridad Dirigida por Modelos (Model-Driven Security, MDS), utilizando una taxonomía de evaluación para realizar el análisis.

3. Crear un meta-modelo para diseñar un nuevo tipo de patrón de seguridad, explicando los detalles y relaciones de los elementos del patrón con las arquitecturas de seguridad empresariales.

4. Crear un enfoque MDS para el modelado y transformación de las arquitecturas de seguridad de la información, adecuándose a distintos niveles de abstracción.

5. Definir un proceso reutilizable que ayude a realizar minería de Enterprise Security Patterns para descubrir nuevos patrones y poder ampliar el catálogo de este tipo de patrones.

6. Desarrollar una herramienta visual que ayude a los ingenieros de seguridad a la hora de diseñar y documentar los nuevos patrones.

7. Validación de la propuesta mediante su aplicación práctica en escenarios reales.

1.6 Marco de la Tesis

En esta sección se presenta el entorno en el que se ha desarrollado esta tesis doctoral. Primero se muestra la organización dentro de la cual el autor ha desarrollado este trabajo. Después se han presentan los proyectos de I+D que constituyen el contexto en el que se ha realizado la presente tesis doctoral.

1.6.1 Grupos de Investigación

1.6.1.1 GSyA - Grupo de Seguridad y Auditoría de la UCLM

El Grupo de Seguridad y Auditoría (GSyA, http://gsya.esi.uclm.es) de la Universidad de Castilla-La Mancha, tiene como principal objetivo investigar sobre diferentes aspectos relacionados con la seguridad y auditoría de los sistemas de información, contribuyendo a la mejora de la docencia en la Ingeniería Informática y a aportar soluciones a la industria del sector. El grupo GSyA está formado por profesores del Departamento de Tecnologías y Sistemas de Información de la Universidad de Castilla-La Mancha y tiene su sede en la Escuela Superior de Informática de Ciudad Real, donde cuenta con un Laboratorio de I+D.

1.6.1.2 Centro de Investigación para la Gestión Tecnológica del Riesgo (CIGTR)

La Fundación Universidad Rey Juan Carlos y el Grupo BBVA, crearon en febrero de 2010 el Centro de Investigación CIGRT con el objetivo de investigar, formar y difundir sobre la Gestión del Riesgo Tecnológico. En este centro se promueve la creación de un entorno ágil para desarrollar, difundir y probar nuevas tecnologías, con la participación de personal del mundo académico y del mundo de la empresarial. Entre sus objetivos también podemos destacar:

- ✓ Acoger proyectos de investigación y transferencia tecnológica.

- ✓ Ofrecer asesoramiento en innovación y desarrollo sobre proyectos de investigación.
- ✓ Promover la divulgación de los resultados obtenidos a través de cualquier medio oportuno (web, redes sociales, artículos, congresos).
- ✓ Impartir formación a través de eventos como los Cursos de Verano (Aranjuez).
- ✓ Colaboración con otras entidades que compartan objetivos similares.

1.6.1.3 Kybele

En los últimos años, el grupo Kybele (http://www.kybele.es/) de la Universidad Rey Juan Carlos (URJC), del cual también forma parte el doctorando, tiene como principal objetivo investigar sobre diferentes aspectos relacionados con la ingeniería del software, la gestión de la información y la ingeniería de servicios, contribuyendo a la mejora de la docencia en la Ingeniería Informática y a aportar soluciones a la industria del sector. El grupo Kybele está formado por profesores del Departamento de Ciencias de la educación, lenguaje, cultura y artes, ciencias histórica-jurídicas y humanísticas y lenguas modernas y tiene su sede en la Facultad de Ciencias Jurídicas y Sociales.

1.6.2 Proyectos de I+D

La realización de esta tesis doctoral se enmarca en los siguientes proyectos: MODEL-CAOS, MASAI, BUSINESS, SERENIDAD y SIGMA-CC.

- ✓ **MODEL-CAOS (TIN 2008-03582)**, financiado por el Ministerio de Educación y Ciencia, comenzó en el año 2009 y ha tenido una duración de tres años (hasta 2011).

  El objetivo principal de MODEL-CAOS ha sido la especificación de un marco para el desarrollo (semi-)automático de sistemas de información, centrándose en la utilización del paradigma de Orientación a Servicios. Este proyecto toma como base los trabajos realizados en proyectos anteriores, actualizándolos mediante la inclusión de las últimas tendencias en el desarrollo de sistemas de información: desarrollo

dirigido por modelos, orientación a servicios, etc., poniendo especial énfasis en el aspecto arquitectónico como elemento central que guía el proceso metodológico. Esta tesis se relaciona con los objetivos del proyecto aportando un meta-modelo de arquitecturas de seguridad para ayudar a los ingenieros de seguridad en el diseño sistemas de información.

- ✓ **MASAI (TIN-2011-22617)** cuyo objetivo principal es aplicar técnicas de Ingeniería de Modelos al Desarrollo Orientado a Servicios de Sistemas de Información, completando el entorno resultado del proyecto Model-CAOS e incluyendo aspectos de seguridad y gestión de equipos, siempre en el marco de la Ingeniería de Servicios. En particular la contribución de la presente Tesis al proyecto MaSaI pasa por la especificación de un método basado en patrones para incluir la seguridad en el desarrollo orientado a servicios de sistemas de información.

- ✓ **BUSINESS (PET2008-0136)**: Construcción de Sistemas de Información Seguros mediante un enfoque de desarrollo dirigido por modelos. Está financiado por el Ministerio de Ciencia e Innovación.

  Se centra en la definición de métodos de ingeniería que permitan el desarrollo de sistemas de información más seguros en los que los aspectos de seguridad, junto con los demás requisitos (funcionales y no funcionales), sean considerados desde etapas tempranas del desarrollo. El objetivo es definir un proceso basado en el desarrollo dirigido por modelos que integre la seguridad en el desarrollo de sistemas de información, permitiendo la integración de las necesidades de seguridad de los sistemas desde el principio, mejorando su integración con el resto de componentes del sistema y la calidad y robustez de las aplicaciones finales. El enfoque dirigido por modelos permitirá transformaciones de modelos de manera descendente y ascendente, permitiendo aplicar técnicas de ingeniería directa, reingeniería e ingeniería inversa para la construcción y el mantenimiento de sistemas de información seguros. Esta tesis está centrada en dos de los objetivos del proyecto, que sería la de definir modelos y componentes en los diferentes niveles de la

arquitectura de seguridad empresarial, a la vez que se definen transformaciones descendentes y ascendentes entre esos modelos en los distintos niveles. Para ello se utilizan las técnicas de especificación de transformaciones más estandarizadas como QVT.

- ✓ **SERENIDAD (PEII11-0327-7035):** avances en la SEguRidad y calidad dE la construcción De Almacenes de Datos basada en modelos. Está financiado por la Consejería de Educación, Ciencia y Cultura de la Junta de Comunidades de Castilla-La Mancha y Fondo Europeo de Desarrollo Regional FEDER.

  Este proyecto está enfocado sobre seguridad y calidad en la construcción de almacenes de datos mediante enfoques dirigidos por modelos, pretendiendo realizar avances en la integración de aspectos e indicadores de calidad y seguridad: a través de paradigmas híbridos de desarrollo que además de considerar los requisitos, consideran las fuentes de datos existentes; la aplicación de seguridad en los procesos de modernización de almacenes de datos; el estudio y aplicación de patrones de seguridad; o la creación de modelos de seguridad dinámicos. Esta tesis define y analiza una serie de patrones de seguridad, que a través de modelos y transformaciones entre ellos, nos permite realizar la integración de aspectos de seguridad desde el principio del proceso de desarrollo.

- ✓ **SIGMA-CC (TIN2012-36904):** Gobierno de la Seguridad y Migración Segura de Sistemas a la Computación en la Nube. Está financiado por el Ministerio de Economía y Competitividad y Fondo Europeo de Desarrollo Regional FEDER.

  Su principal objetivo es el de mejorar la forma en que se gobierna la seguridad de la información en entornos basados en el paradigma del Cloud Computing, prestando una atención especial al modo en que los sistemas y aplicaciones se migran a este nuevo entorno, para que éstos se adapten a los nuevos desafíos de seguridad de este paradigma. Esta tesis se relaciona con el proyecto en uno de los objetivos parciales que es la utilización y definición de patrones de seguridad, aportando una plantilla para su definición donde las características específicas de cada servicio o proveedor

pueden ser incorporadas fácilmente a través de la transformación de modelos arquitecturales.

1.7 Organización de la Tesis

El resto de la tesis doctoral se estructura en los siguientes capítulos:

Capítulo 2. **Método de Trabajo.** En este capítulo se presenta el método de trabajo que se ha adoptado para la consecución de los objetivos planteados.

Capítulo 3. Estado del arte. En este capítulo se realiza un estudio detallado de las principales áreas relacionadas con los Patrones de Seguridad y la Ingeniería Dirigida por Modelos. Para ello, se ha desarrollado:

1. Una Revisión Sistemática de Patrones de Seguridad (*Security Patterns*).
2. Una Revisión Sistemática de Minería de Patrones de Seguridad (*Security Pattern Mining*).
3. Un estudio detallado sobre la Seguridad dirigida por Modelos (MDS) y las Arquitecturas dirigidas por Modelos (MDA).

El principal objetivo de este estudio es adquirir un conocimiento base lo suficientemente profundo y maduro como para definir el meta-modelo de los *Enterprise Security Patterns*.

Capítulo 4. Enterprise Security Patterns. En este capítulo se desarrolla un nuevo meta-modelo de *Enterprise Security Patterns* con el objetivo de ayudar a las organizaciones en el diseño de arquitecturas de seguridad. Para presentar este meta-modelo, se describen los siguientes contenidos:

1. Una descripción de las arquitecturas de seguridad empresariales y los elementos incluidos en estas arquitecturas.

2. Una plantilla para documentar Enterprise Security Patterns, considerando los elementos incluidos en las arquitecturas de seguridad empresariales.

3. Un meta-modelo para diseñar Enterprise Security Patterns, explicando los detalles y relaciones de los elementos del patrón con las arquitecturas de seguridad empresariales.

4. Un proceso de modelado y transformación propuesto para soportar la definición de arquitecturas de seguridad empresariales, usando el paradigma MDS.

5. Un proceso reutilizable que ayude a realizar minería de Enterprise Security Patterns para descubrir nuevos patrones y poder ampliar el catálogo de este tipo de patrones.

Capítulo 5. Herramienta Prototipo. En este capítulo hemos desarrollado una nueva herramienta llamada C-SMArT (Casandra - Security Model-Driven Architecture Toolkit) que representa una característica adicional hacia la consideración de los Enterprise Security Patterns como marco completo para la especificación y diseño de arquitecturas de seguridad.

Capítulo 6. Caso de Estudio. En este capítulo se presenta un caso de estudio realizado junto con el Centro de Investigación para la Gestión Tecnológica del Riesgo (CIGTR) de la Fundación Universidad Rey Juan Carlos, en el que se documenta un nuevo *Enterprise Security Pattern* siguiendo la plantilla propuesta en el capítulo 4. Este nuevo patrón garantiza la confidencialidad de los datos utilizados en aplicaciones proporcionadas por otras compañías a través de Internet.

Capítulo 7. Conclusiones. En este capítulo se presentan las conclusiones finales alcanzadas durante la realización de la tesis doctoral, así como se muestran las publicaciones conseguidas que avalan la propuesta realizada. Además, se presentan las líneas de investigación futuras concernientes con los Enterprise Security Patterns.

Bibliografía. La lista de referencias bibliográficas citadas.

2. Método de Trabajo

2.1 Introducción

En algunas ramas de la investigación científica, por ejemplo, la medicina, la física o la biología, cuentan con un conjunto de reglas y descripciones en sus estrategias de investigación que no sólo incluyen una guía detallada para los investigadores, sino también una vista simplificada para el público (Shaw, 2002). En otras ramas como en Ciencias de la Computación, los sistemas de información y la Ingeniería del Software, el desarrollo de la investigación sólo data de cinco décadas (Glass et al., 2002) lo que las transforma en un campo sin demasiados antecedentes históricos (Lázaro y Marcos, 2005). No obstante, existe un esfuerzo por mejorar dicha situación y se pueden encontrar diversos métodos de investigación entre los que se incluyen análisis conceptuales, casos de estudio, análisis de datos, experimentación de campo, experimentación de laboratorio y simulación (Glass et al., 2004).

En este capítulo se presentan dos métodos de trabajo que se han utilizado para conseguir los objetivos planteados para esta tesis doctoral. Se ha considerado el uso del Método Investigación-Acción (*Action-Research*) por ser éste uno de los principales métodos de investigación cualitativa en el campo de los sistemas de información y en la Ingeniería del Software (Polo et al., 2002), y la propuesta de Barbara Kitchenham para la *Revisión Sistemática* de la literatura (Kitchenham, 2004).

2.2 Investigación-Acción

El término Investigación-Acción (IA) fue propuesto por primera vez en 1946 por el psicólogo social Kurt Lewin es su trabajo titulado *Action Research and Minority Problems* (Lewin, 1946).

En su primera definición, el método IA fue presentado como una forma de investigación que podía enlazar el enfoque experimental de las ciencias sociales con programas de acción social que respondieran a los problemas sociales principales de aquella época.

Lewin también argumentaba que se podían lograr de forma simultánea avances teóricos y cambios sociales. En las últimas décadas, los métodos de investigación cualitativa, y en especial IA, captaron la atención y aceptación de la comunidad científica vinculada a los sistemas de información y la Ingeniería del Software, desde que fue introducido por Wood-Harper (Wood-Harper, 1985).

En realidad el método IA no se refiere a un método de investigación concreto, sino a una clase de métodos que tienen en común las siguientes características (Baskerville, 1999):

- ✓ Orientación a la acción y al cambio.
- ✓ Focalización en un problema concreto.
- ✓ Un modelo de proceso que engloba etapas sistemáticas y normalmente iterativas.
- ✓ Colaboración entre los participantes.

2.2.1 Definiciones

Posiblemente por no ser un método concreto, existen diversas definiciones de IA. Algunas de las más relevantes son las siguientes:

- ✓ Para McTaggart (McTaggart, 1991) es *la forma que tienen los grupos de personas de preparar las condiciones necesarias para aprender de sus propias experiencias, y hacer estas experiencias accesibles a otros*.
- ✓ Para French y Bell (French y Bell, 1996) es *el proceso de recopilar de forma sistemática datos de la investigación acerca de un sistema actual en relación con algún objetivo, meta o necesidad de ese sistema; de alimentar de nuevo con esos datos al sistema; de emprender acciones por medio de variables alternativas seleccionadas dentro del sistema, basándose tanto en los datos como en las hipótesis; y de evaluar los resultados de las acciones, recopilando datos adicionales*.
- ✓ Para Wadsworth (Wadsworth, 1998) consiste en *la participación de todas las partes involucradas en la investigación, examinando la situación existente (que sienten como problemática), con los objetivos de cambiarla y mejorarla*.

Partiendo de las definiciones anteriores se puede deducir que el método IA tiene una doble finalidad: (i) generar un beneficio al "*cliente*" de la investigación, y al mismo tiempo, (ii) *generar "conocimiento de investigación" relevante*.

En el campo de los sistemas de información, el cliente de una investigación es normalmente una organización a la cual el investigador aporta un servicio de consultoría o de desarrollo de software, a cambio de acceder a datos de interés para la investigación y recibir financiación (Kock y Lau, 2001). En cualquier caso, el investigador que utiliza Investigación-Acción en Sistemas de Información (IA-SI) sirve a dos entidades diferentes: *el cliente de la investigación y la comunidad científica de sistemas de información*. Las necesidades de ambos suelen ser muy diferentes y, a veces, opuestas entre sí. Intentar satisfacer ambas demandas es el principal desafío que todos los investigadores de IA-SI tienen que enfrentar. Sirviendo tanto las necesidades de los practicantes, como las del conocimiento científico, se añaden nuevos elementos a la investigación que hace que sea más deseable.

2.2.2 Roles y modalidades en la Investigación - Acción

En un análisis más formal de los participantes incluidos en IA, (Wadsworth, 1998) identifica los siguientes cuatro tipos de roles en este método (en algunas ocasiones la misma persona o equipo puede desempeñar más de un rol):

- ✓ El **investigador**, el individuo o grupo que lleva a cabo de forma activa el proceso investigador.
- ✓ El **objeto investigado**, es decir, el problema a resolver.
- ✓ El **grupo crítico de referencia**, es aquél para quien se investiga desde la perspectiva de que tiene un problema que necesita ser resuelto y que también participa en el proceso de investigación (aunque menos activamente que el investigador). En este grupo hay tanto personas, que saben que están participando en la investigación, como otras que participan sin saberlo.
- ✓ El **beneficiario** o *stakeholder*, es aquél para quien se investiga en el sentido de que puede beneficiarse del resultado de la investigación, aunque no participa directamente en el proceso. Normalmente es el receptor de documentos, informes, etc. En este grupo, por ejemplo, caben tanto las empresas que se benefician de un nuevo método para resolver problemas en tecnologías de la información, como los técnicos que aplican dicha metodología.

Desde sus orígenes se han distinguido formas diferentes de aplicar Investigación- Acción. French y Bell (French y Bell, 1996) proponen cuatro variantes que dependen principalmente de las características del proyecto de investigación:

- ✓ **De diagnóstico**: el investigador se adentra en una situación problemática, la diagnostica y realiza recomendaciones al grupo crítico de referencia, pero sin que haya un control posterior de sus efectos.

- ✓ **Participativa**: el grupo crítico de referencia pone en práctica las recomendaciones realizadas por el investigador, compartiendo con él sus efectos y resultados.
- ✓ **Empírica**: el grupo crítico de referencia realiza un registro amplio y sistemático de sus acciones y sus efectos. Esta característica hace que esta variante sea difícilmente aplicable.
- ✓ **Experimental**: consiste en evaluar las diferentes opciones que existen para conseguir un objetivo. El principal inconveniente de esta variante reside en la dificultad de poder medir objetivamente las diversas opciones, ya que por lo general serán, o bien aplicadas en distintas organizaciones con distintas características que enturbian los resultados de la investigación, o bien en una sola organización, pero en distintos momentos, con lo que el entorno experimental habrá variado.

2.2.3 Etapas de la Investigación – Acción

Un proceso de investigación que emplea IA se compone de grupos de actividades organizadas formando un ciclo característico. Nancy Padak y Gary Padak (Padak y Padak, 1994) identifican los siguientes pasos, que deben seguirse en las investigaciones que utilicen este método. La Figura 2.1 muestra un diagrama conceptual de cómo se relacionan cada una de las etapas:

1. **Planificación:** identificar las cuestiones relevantes, que guiarán la investigación, que deben estar directamente relacionadas con el objeto que se está investigando y ser susceptibles de encontrarles respuesta. En esta actividad se buscan caminos alternativos, líneas a seguir o reforzar algo existente. El resultado es que se definen claramente otros problemas o situaciones a tratar. Algunos autores (Baskerville, 1997) distinguen entre diagnóstico (la identificación de los problemas iniciales) y planificación (la especificación de las acciones para resolver dichos problemas).

2. **Acción**: variación de la práctica, cuidadosa, deliberada y controlada. Se efectúa una simulación o prueba de la solución. Es el momento cuando el investigador interviene sobre la realidad.

3. **Observación**: recoger información, tomar datos, documentar lo que ocurre. Esta información puede proceder prácticamente de cualquier sitio (bibliografía, medidas, resultados de pruebas, observaciones, entrevistas, documentos, etc.). También se conoce como "evaluación".

4. **Reflexión**: compartir y analizar los resultados con el resto de interesados, de tal manera que se invite al planteamiento de nuevas cuestiones relevantes y, como añade Wadsworth en (Wadsworth, 1998), *profundizar en la materia que se está investigando para proporcionar conocimientos nuevos que puedan mejorar las prácticas, modificando éstas como parte del propio proceso investigador, para luego volver a investigar sobre estas prácticas una vez modificadas*. En algunas variantes de IA la reflexión no es considerada realmente como una etapa, sino como un proceso continuo que ocurre durante el tiempo de investigación.

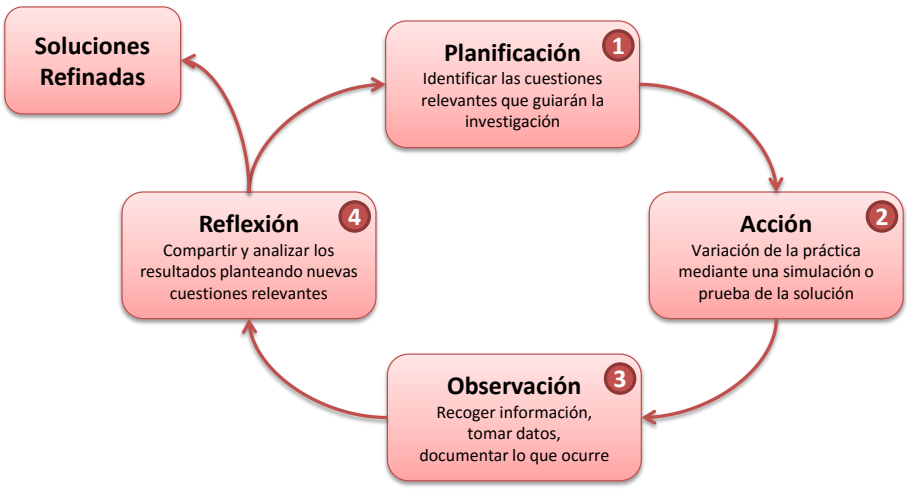


Figura 2.1. Carácter cíclico de Investigación-Acción

El carácter iterativo definido queda claramente expresado en la forma que se van obteniendo soluciones cada vez más refinadas al final de cada ciclo. Esto permite poner en marcha nuevas etapas que son llevadas a la práctica y comprobadas en el ciclo siguiente, tal como muestra la Figura 2.2. Este ciclo caracteriza al método IA como un proceso reflexivo de aprendizaje y búsqueda de soluciones. El carácter cíclico supone volver a evaluar o replantear las acciones a seguir ponderando diagnóstico y reflexión.

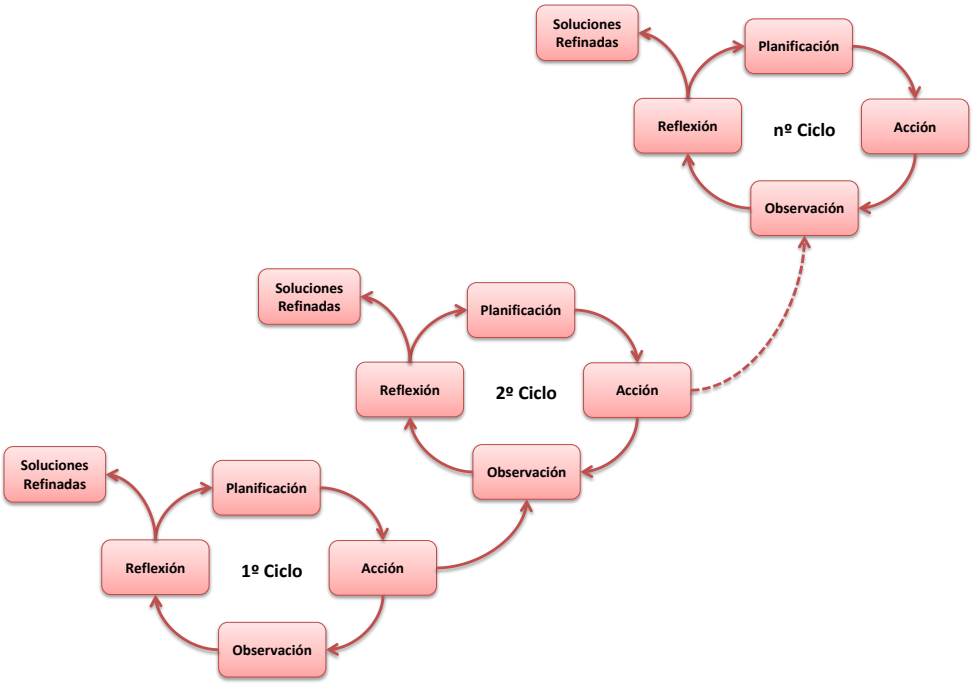


Figura 2.2. Carácter iterativo de Investigación-Acción

2.2.4 Problemas de la Investigación – Acción y alternativas

En los últimos años, IA ha sido reconocido como uno de los métodos de investigación (cualitativa) más potentes en el ámbito de los sistemas de información. Ahora bien, la comunidad de especialistas ha detectado diversos problemas en su aplicación que tienen tres causas fundamentales:

1. La falta de método con que los investigadores y practicantes utilizan y conciben IA-SI.

2. El contexto de consultoría utilizado, que impone una perspectiva demasiado restrictiva al implicar responsabilidades contractuales e intereses organizacionales que pueden ir en contra de lo propuesto por IA.

3. La ausencia de un modelo de proceso de investigación definido que indique los pasos a seguir en IA-SI.

Todo lo anterior puede tener como consecuencia una falta de rigurosidad del proceso de investigación. Para solventar estos problemas se han propuesto, entre otras, las siguientes alternativas:

- ✓ Conducir la investigación usando una perspectiva de gestión de proyectos.
- ✓ Incluir criterios de calidad especialmente concebidos.
- ✓ Analizar los factores que inciden en la formalización del proceso.
- ✓ Organizar el proceso con una estructura de proyecto.

Combinando varias de estas ideas, en (Estay y Pastor, 2000; Estay y Pastor, 2000) han propuesto *usar la gestión de proyectos para mejorar el rigor de un proyecto de IA-SI, lo cual se ha traducido en generar una estructura de proyecto que contenga los principales elementos de IA-SI*. Para lograr lo anterior, estos autores plantean la necesidad de adoptar prácticas de gestión, adecuadas a IA-SI, basadas en el PMBOK (Project Management Body of Knowledge), el modelo de gestión de proyectos más difundido a nivel internacional, propuesto por el Project Management Institute (PMI, 2000).

Para (Estay y Pastor, 2000) IA y proyecto son conceptos equivalentes, ya que ambos son experiencias de trabajo únicas con resultados finales igualmente únicos y, además, comparten la idea de intervención, es decir, ambos suponen una alteración voluntaria de la realidad. Aunque la intervención en IA produce alteraciones en una práctica de trabajo, también es una forma de obtener datos de la experiencia real que son necesarios para el proceso de investigación. Los mismos autores también han propuesto un modelo de madurez basado en CMM (SEI, 1995),

aplicando prácticas de gestión de proyectos de forma incremental con objeto de garantizar una mejora del rigor y calidad del uso de Investigación-Acción en sistemas de información (Estay y Pastor, 2001).

2.2.5 Ciclos en la Investigación - Acción

En el contexto de la investigación cualitativa en sistemas de información se puede considerar que existen dos realidades (científica/académica y práctica) que interactúan pero que se mueven en planos diferentes. IA-SI opera sobre esta realidad dual, que se concreta en dos tipos de ciclos de Investigación-Acción para dos tipos de proyectos:

1. **Ciclos orientados a resolver problemas** dentro de proyectos de sistemas de información. Estos proyectos están asociados al desarrollo de soluciones informáticas, es decir, proyectos de consultoría informática, de desarrollo de software, de implantación y/o mantenimiento de sistemas informáticos, etc. En este caso el investigador se encarga de resolver un problema e IA aparece como una herramienta más para el desarrollo de los sistemas de información.

2. **Ciclos orientados a investigar** dentro de proyectos de investigación. Estos proyectos son esfuerzos intencionados buscando un resultado. En este caso, IA nos ofrece un método de trabajo y una justificación para acercarnos a una determinada realidad con fines de probar una teoría o hipótesis.

Por otro lado, la estructura de proyecto de IA-SI propuesta por (Estay y Pastor, 2000) define dos ciclos característicos:

1. **Ciclo orientado a construir una solución** para generar nuevo conocimiento útil a practicantes y mejorar su práctica. El investigador se conecta con la realidad mediante una intervención. La investigación se utiliza para construir modelos, teorías o

conocimiento de manera informada e influida por la realidad. En este ciclo es el interés por resolver un problema lo que origina el interés por la investigación.

2. **Ciclo orientado a gestionar la investigación** para producir nuevo conocimiento a la disciplina de sistemas de información y mejorar la práctica de los investigadores. En este ciclo es el interés por la investigación el que origina interés por resolver ciertos problemas.

En resumen, IA-SI puede analizarse desde dos dimensiones complementarias (Figura 2.3):

- ✓ Una dimensión *vertical* en función del tipo de proyecto.
- ✓ Una dimensión *horizontal* en función del bi-ciclo típico de la estructura de un proyecto de IA-SI.

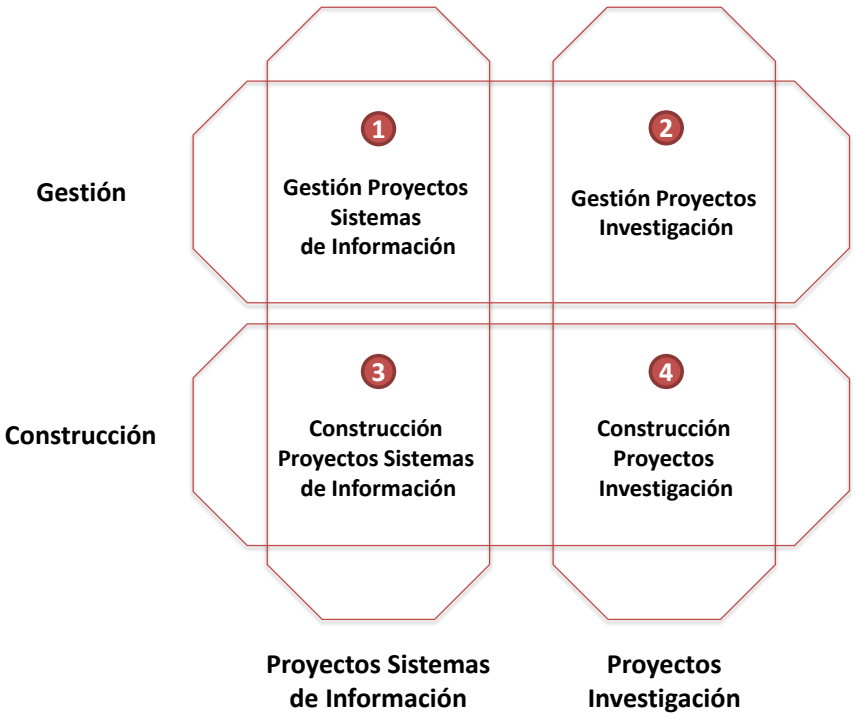


Figura 2.3. Dos dimensiones de Investigación-Acción en Sistemas de Información

En (Lau, 1997) se puede encontrar un resumen del uso de IA-SI, así como diversos ejemplos comentados, que han sido publicados por diferentes autores, y que hacen referencia a

la construcción y desarrollo de sistemas de información, y más especialmente, al análisis, diseño, desarrollo e implementación de software y a los procesos asociados.

En (Baskerville, 1999) se hace una introducción al uso de IA-SI indicando 10 formas de utilización y cuatro características que determinan dicha forma de uso:

- ✓ El modelo de proceso: iterativo, reflexivo o lineal.
- ✓ La estructura: rigurosa o fluida.
- ✓ El rol del investigador: colaborador, facilitador o experto.
- ✓ Los objetivos principales: desarrollo organizacional, diseño de sistemas, conocimiento científico o entrenamiento.

En (Baskerville y Wood-Harper, 1996) señalan siete estrategias básicas para llevar a cabo IA-SI:

- ✓ Utilizar el “paradigma del cambio”.
- ✓ Establecer un acuerdo o contrato formal de investigación.
- ✓ Proporcionar un marco de trabajo teórico.
- ✓ Planificar los métodos de captura de datos.
- ✓ Mantener la colaboración y el aprendizaje recíproco entre investigador y grupo crítico de referencia.
- ✓ Incentivar las iteraciones del ciclo típico.
- ✓ Buscar la generalización de las soluciones.

2.3 Revisiones Sistemáticas de la Literatura

La gran mayoría de las investigaciones, tanto en el mundo académico como en el profesional, deberían considerar una revisión de la literatura o estado del arte del tema que se desea abordar, aunque éste no sea el objetivo final de la investigación. Un estado del arte constituye la base para la formulación de investigaciones de mayor calado y es fundamental para explicar las aportaciones al conocimiento actual que se pretenden alcanzar con la investigación.

Una **Revisión Sistemática** puede ser definida como una buena forma de evaluar e interpretar el estado del arte de una investigación, área o fenómeno de interés, utilizando una metodología confiable, rigurosa y auditable.

Dado que las disciplinas de computación tienen una trayectoria reciente en comparación con otras disciplinas de la ciencia, no existen metodologías propias que guíen el desarrollo de revisiones sistemáticas de la literatura en éstas. Por este motivo, Kitchenham propuso un método para realizar revisiones sistemáticas (Kitchenham, 2004) basado en pautas desarrolladas para la investigación médica, adaptado para ser usado por equipos investigadores asociados al ámbito de la Ingeniería del Software. La Tabla 2-1 muestra las tres etapas fundamentales según la versión más reciente de la metodología (Kitchenham, 2007).

Estas etapas a su vez se encuentran dividas en otras etapas que detallan la forma en que se deben desarrollar. Para el caso particular de esta tesis doctoral, en que la supervisión está a cargo de un tutor o guía y que tiene un tiempo límite para su realización, se adaptó la propuesta considerando los pilares principales de la misma (ver Tabla 2-2).

Etapa 1: Planificación de la revisión
✓ Identificación de la necesidad de revisión
✓ Comisionado de la revisión
✓ Especificación de la cuestión de investigación
✓ Definición de un protocolo de revisión
✓ Evaluación del protocolo de revisión
Etapa 2: Desarrollo de la revisión
✓ Identificación de la investigación
✓ Selección de los estudios primarios
✓ Evaluación de la calidad del estudio
✓ Extracción y seguimiento de datos
✓ Síntesis de datos
Etapa 3: Publicación de los resultados de la revisión
✓ Especificación de los mecanismos de diseminación
✓ Formateado del informe principal
✓ Evaluación del informe

Tabla 2-1. Metodología de Revisiones Sistemáticas de Kitchenham

Etapa 1: Planificación de la revisión
✓ Identificación de la necesidad de revisión
✓ Especificación de la cuestión de investigación
✓ Definición de un protocolo de revisión
✓ Evaluación del protocolo de revisión
Etapa 2: Desarrollo de la revisión
✓ Búsqueda de los estudios primarios
✓ Selección de los estudios primarios
✓ Extracción y gestión de datos
✓ Síntesis de datos
Etapa 3: Publicación de los resultados de la revisión

Tabla 2-2. Adaptación para esta tesis doctoral de la metodología de Kitchenham

Las etapas del método usado en esta tesis doctoral para llevar a cabo la revisión de la literatura se describen a continuación de forma detallada.

2.3.1 Etapa 1: Planificación de la revisión

Esta etapa tiene como propósito específico definir los parámetros más importantes que serán tenidos en cuenta cuando se lleve a cabo la revisión. Se debe establecer las razones que justifican llevarla a cabo, la manera en que se hará la búsqueda de trabajos y la forma en que éstos serán revisados. Finalmente, se evaluará la planificación realizada. Esta etapa ha sido dividida en las siguientes sub-etapas:

2.3.1.1 Identificación de la necesidad de la revisión

La necesidad de una revisión sistemática surge de la necesidad de un investigador por recopilar, de manera rigurosa e imparcial, toda la información existente sobre algún fenómeno de interés. El objetivo de dicha recopilación es iniciar otras actividades de investigación futuras.

Antes de emprender una revisión sistemática, el investigador se debe asegurar de que ésta es necesaria. En particular, es recomendable identificar y analizar cualquier revisión sistemática existente acerca del fenómeno de interés con un criterio de evaluación apropiado. Con este fin, resulta conveniente utilizar listas de verificación que contengan preguntas como las siguientes:

- ✓ ¿Cuáles son los objetivos de la revisión?
- ✓ ¿Qué fuentes fueron buscadas para identificar estudios primarios?
- ✓ ¿Qué criterios se incluyeron o excluyeron y cómo fueron aplicados?
- ✓ ¿Cómo fueron extraídos los datos de los estudios preliminares?
- ✓ ¿Cómo fueron sintetizados los datos?
- ✓ ¿Cómo se diferencian los estudios investigados?

- ✓ ¿Cómo fueron combinados los datos?
- ✓ ¿Era razonable combinar los estudios?

Las razones más frecuentes que justifican la necesidad de una revisión sistemática son (Kitchenham, 2004):

- ✓ Resumir la evidencia existente concerniente a una tecnología.
- ✓ Identificar algún vacío en la investigación con el objeto de sugerir áreas para investigaciones futuras.

Proveer un marco de trabajo y/o los antecedentes necesarios con el objeto de posicionar nuevas actividades de investigación.

Junto con lo anterior, se deben identificar claramente los recursos con los que se cuenta al iniciar la revisión (por ejemplo: Internet, revistas electrónicas de acceso público o restringido, actas de congreso, etc.), ya que esto puede variar conforme se avance en la investigación. Es posible que se pueda contar con nuevos recursos, como por ejemplo: acceso a investigaciones recientes, una nueva suscripción o adquisición de libros.

2.3.1.2 Definición de un protocolo de búsqueda

En esta sub-etapa se deben definir las normas que seguirá la investigación respecto del proceso de búsqueda en las fuentes de información definidas en la sub-etapa anterior. El protocolo de búsqueda debe contener una definición de los términos que se buscarán, las combinaciones de éstos, la estrategia de búsqueda empleada dependiendo de la fuente y la manera en que se registrarán los resultados.

En relación con la estrategia de búsqueda, es importante establecer la manera en que se va a proceder respecto de cada fuente empleada. Un caso particular es Internet, que está jugando un rol cada vez más importante en cuanto a la accesibilidad de la literatura científica. Debido a que la información disponible en Internet es muy abundante, es necesario establecer criterios

para hacer filtros que permitan obtener sólo aquella información que sea realmente útil. Se recomienda hacer un registro de los resultados de las búsquedas. Esto puede servir para justificar la necesidad de investigar en algún área específica, o para demostrar, cuantitativamente, que los trabajos en una determinada área son escasos, o que son muy heterogéneos, o simplemente para demostrar la rigurosidad con que se ha realizado el proceso de búsqueda.

Finalmente, es importante tener en cuenta que el proceso de búsqueda es perfectible, por lo que el protocolo puede y debe ser mejorado durante el desarrollo de la búsqueda, por ejemplo, se pueden incorporar otros términos de búsqueda o realizar otras combinaciones de los términos usados.

2.3.1.3 Definición de un protocolo de revisión

La definición de un protocolo de revisión implica especificar las normas de revisión, los criterios de exclusión e inclusión, la estrategia de extracción de datos y finalmente la estrategia de síntesis. Estos criterios forman parte del protocolo y deberán estar definidos antes de emprender la revisión sistemática. Contar con un protocolo predefinido contribuye a evitar los prejuicios del investigador. Con esto se desea evitar, en la medida de lo posible, que la selección de los estudios individuales pueda estar guiada por las expectativas del investigador.

Una parte importante del protocolo de revisión es contar con una definición de la forma en que se hará la revisión de manuscritos. Dado que un alto número de los estudios que serán revisados se encuentran en formato de artículo científico, se ha tomado como referencia para este protocolo de revisión la estructura de artículo científico propuesta por (Srba, 2008): Resumen, Introducción, Trabajos relacionados, Preliminares, Cuerpo del Artículo, Conclusión, Referencias.

Al igual que el protocolo anterior, éste puede ser perfeccionado durante el desarrollo de la revisión.

2.3.1.4 Evaluación de la planificación

Esta etapa consiste en hacer una valoración de la planificación. Ya que esta propuesta se enmarca en el contexto del desarrollo de una tesis doctoral la evaluación de la planificación tendrá que hacerla el tutor o guía de la tesis.

2.3.2 Etapa 2: Desarrollo de la revisión

En esta etapa se lleva a cabo la revisión propiamente dicha. Su desarrollo está guiado por la planificación de la revisión; no obstante, por ser éste un proceso flexible, es posible incluir cambios que mejoren su desempeño. A continuación, se definen las sub-etapas que contempla el desarrollo de la revisión.

2.3.2.1 Búsqueda de estudios primarios

La búsqueda de estudios primarios se debe realizar en base al protocolo de búsqueda que fue definido para ello. Los estudios que se consideren potencialmente útiles se deberán dejar accesibles para la siguiente etapa, ya sea en formato electrónico y/o impreso. También es posible dejar registrado el lugar dónde se puedan ubicar para que, cuando corresponda, se proceda a su selección.

2.3.2.2 Selección de estudios primarios

La selección de los estudios se debe hacer en base al protocolo de revisión definido. Este proceso será guiado por los criterios de inclusión y exclusión definidos anteriormente. Aunque estos criterios dependen de los intereses de la investigación es recomendable dejar un registro de los motivos de exclusión.

2.3.2.3 Extracción y gestión de datos

En esta sub-etapa se extrae la información de interés en los estudios, ya sea en forma de resúmenes, ideas o partes de los documentos. Esta extracción se debe realizar en base al protocolo de revisión definido anteriormente. Adicionalmente se debe registrar la información necesaria para gestión, lo cual incluye el título del documento, autor o autores, fecha de publicación, lugar de publicación, ubicación física u otro tipo de información que los investigadores consideren pertinente. Se recomienda usar herramientas, como por ejemplo EndNote (http://www.endnote.com), que permitan mantener y gestionar los datos más relevantes de los estudios revisados. Este tipo de herramientas, junto con el registro de la información de los documentos, permite realizar búsquedas, ordenamiento, en general, hacer una adecuada gestión de la información.

2.3.2.4 Síntesis de datos

En esta sub-etapa, al igual que en las anteriores, se debe aplicar el protocolo definido en la revisión. Consiste en registrar la información extraída de los estudios primarios siguiendo alguna estrategia definida. Los datos pueden ser sintetizados considerando, por ejemplo, el enfoque que se le desea dar a la presentación del estado del arte o la identificación del o los fenómenos de interés.

2.3.3 Etapa 3: Publicación de los resultados de la revisión

Esta etapa se corresponde a la utilización de los resultados una vez que disponemos de ellos. Estos resultados pasan a formar parte, por ejemplo, de una tesis doctoral, y adicionalmente pueden ser comunicados a través de la publicación de un artículo o un informe técnico.

2.4 Aplicación de los Métodos de Trabajo en esta Tesis Doctoral

Tanto la revisión sistemática de la literatura como el método Investigación–Acción han sido usados para el desarrollo de esta tesis doctoral. A continuación, mostramos como han sido aplicados estos métodos de trabajo.

2.4.1 Investigación-Acción

El método Investigación-Acción ha sido aplicado en su variante participativa, es decir, aquella en la que el grupo crítico de referencia pone en práctica las recomendaciones realizadas por el investigador, compartiendo con él sus efectos y resultados. Para lo cual se han considerado los siguientes participantes:

Investigador: en este caso corresponde a los grupos de investigación Kybele y GSyA. El primer grupo está formado por profesores del Departamento de Ciencias de la educación, lenguaje, cultura y artes, ciencias histórica-jurídicas y humanísticas y lenguas modernas de la Universidad Rey Juan Carlos; el segundo grupo está formado por profesores del Departamento de Tecnologías y Sistemas de Información de la Universidad de Castilla-La Mancha. El autor de este trabajo es miembro colaborador de ambos grupos.

Objeto investigado: es decir, el problema a resolver. En este caso se refiere a los patrones de seguridad y la minería de patrones de seguridad.

Grupo crítico de referencia (GCR): aquel para el cual se investiga y además participa en el proceso de investigación. En este caso está compuesto por los participantes en los proyectos MASAI, SERENIDAD, FEDER, SIGMA-CC y GEODAS en los que colaboran diversas universidades

nacionales. Por todo ello, el grupo crítico de referencia ha estado constituido por representantes de universidades españolas.

Beneficiarios: serán todas aquellas organizaciones que se puedan ver potencialmente beneficiadas por los resultados del trabajo, es decir, todas aquellas organizaciones que estén interesadas en la utilización de patrones de seguridad dentro de su metodología de desarrollo de software. En concreto, el grupo BBVA, organización en la que se han aplicado y validado los Enterprise Security Patterns (véase Capítulo 6), se ha visto enormemente beneficiada con la adopción de este tipo de patrones siéndole garantizada, normalizada y facilitada la tarea de integrar el aspecto de las arquitecturas de seguridad en fases iniciales de su metodología de desarrollo de software.

En la Figura 2.4, se muestran los participantes y sus relaciones en la aplicación del método Investigación-acción dentro de nuestro trabajo.

La aplicación de IA-SI más evidente es cuando una organización humana interactúa con sistemas de información. De hecho, Investigación-Acción es una las pocas aproximaciones válidas para estudiar los efectos de alteraciones específicas en metodologías de desarrollo y mantenimiento de sistemas en organizaciones humanas (Baskerville y Wood-Harper, 1996). Por tanto, la definición de un Artefacto Software desde el punto de vista de su modelado es un dominio adecuado para la aplicación de Investigación-Acción. Ello se demuestra con los resultados conseguidos en la aplicación del método:

1. 1) El investigador propuso un caso de estudio para diseñar arquitecturas de seguridad que fue aceptado por el GCR.

2. 2) El investigador trabajó activamente para que los beneficios fueran mutuos, científicos para el investigador y prácticos para el GCR.

3. El conocimiento obtenido pudo ser aplicado rápidamente.

4. La investigación se desarrolló en un proceso típico cíclico e iterativo combinando teoría y práctica.

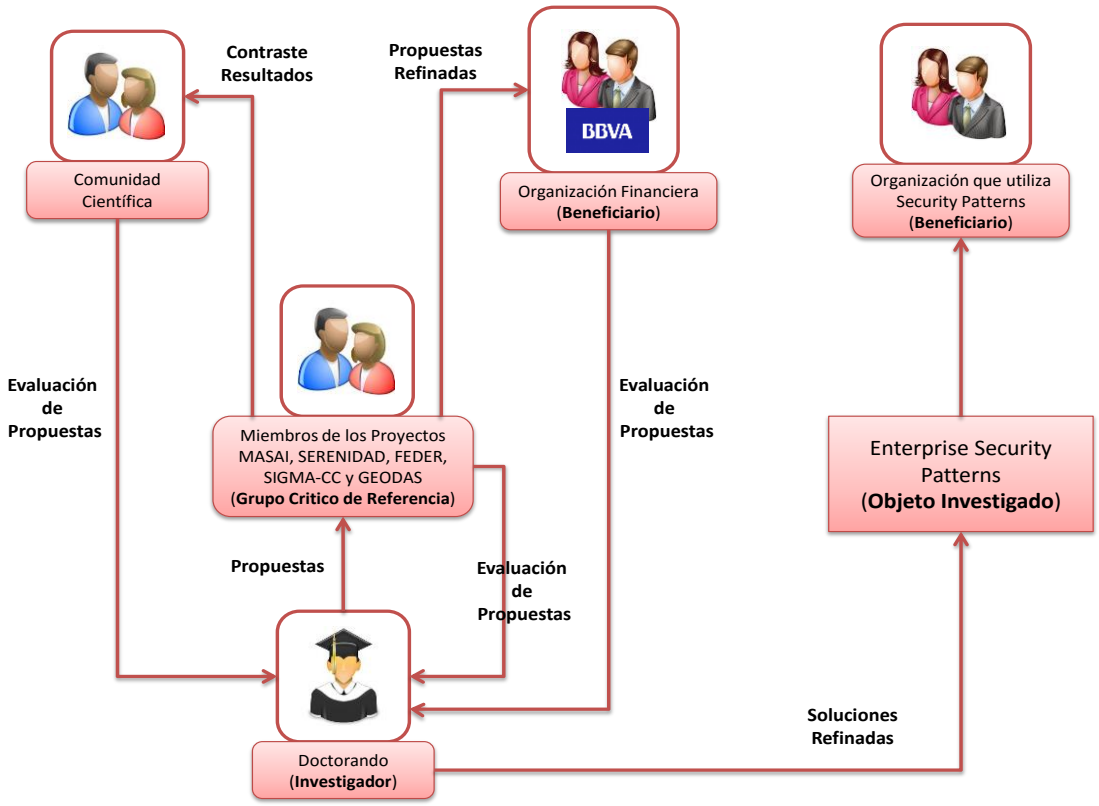


Figura 2.4. Participantes en la Aplicación de Investigación-Acción

La puesta en marcha de Investigación-Acción durante el proceso investigador de este trabajo ha supuesto una realimentación entre el investigador y el GCR. Se podrían identificar tres grupos de ciclos básicos en la aplicación de investigación en acción para nuestro proceso de desarrollo que nos han permitido ir obteniendo soluciones cada vez más refinadas generadas de forma participativa. Podemos resumir este proceso en los siguientes ciclos:

Ciclo general inicial: investigadores y grupo crítico de referencia definieron la problemática general del diseño de arquitecturas de seguridad empresariales y en particular, se establecieron objetivos y requisitos generales, para la definición de un conjunto de guías que facilitara el trabajo de los ingenieros de seguridad, a la hora de diseñar arquitecturas de seguridad (planificación). Se procedió a la búsqueda de toda la información de posible interés al respecto

(acción). Su análisis posterior (observación) permitió descubrir que los patrones de seguridad definidos por la comunidad científica hasta la fecha no cubrían todos los aspectos necesarios y debían tenerse en cuenta múltiples aspectos de naturaleza diferente. El razonamiento y puesta en común (reflexión) entre los investigadores y el GCR permitió detectar que la solución podría consistir en crear un nuevo tipo de patrón de seguridad que abordara de forma integrada las arquitecturas dirigidas por modelos (MDA) y los patrones de seguridad, ya que se detectó en la fase de acción que no existían propuestas relacionadas con la integración y aplicación práctica de los cinco pilares de las arquitecturas de seguridad: activos de información, contexto, amenazas, *stakeholders* y tecnologías de seguridad.

Ciclos generales intermedios: A partir del ciclo general se identificaron los siguientes grupos cíclicos intermedios:

1. Ciclo Conceptual: ¿qué elementos son necesarios para definir arquitecturas de seguridad basándonos en patrones?

2. Ciclo Técnico: ¿qué herramientas, estándares y tecnologías software son útiles para definir de forma integrada los requisitos de los elementos relacionados con una arquitectura de seguridad?

Ciclos específicos finales: A partir del momento en que las respuestas anteriores quedaron claras, tanto para los investigadores como para el grupo crítico de referencia, se procedió a realizar ciclos específicos de Investigación-Acción para cada uno de los dos componentes principales.

Los ciclos anteriores significan que para el desarrollo de la propuesta en que se basa esta tesis doctoral, se ha utilizado Investigación-Acción con una estructura de proyecto multi-cíclica.

A la hora de realizar los diversos ciclos aparecieron dependencias, sobre todo en el ciclo técnico, ya que las tecnologías de seguridad utilizadas en la arquitectura de la organización

financiera dependían de diferentes responsables, siendo necesarias un número determinado de iteraciones en el ciclo conceptual.

Por lo tanto, en nuestra opinión el resultado de esta investigación homogeniza y facilita a los ingenieros el diseño de arquitecturas de seguridad de una organización, incluyendo los Enterprise Security Patterns desde primeras fases del ciclo de vida de desarrollo de software.

2.4.2 Revisiones Sistemáticas

Las revisiones sistemáticas de la literatura han sido utilizadas para establecer una sólida base en cuanto a los trabajos relacionados con el objeto investigado, es decir, los patrones de seguridad y la minería de patrones de seguridad. Puesto que la investigación tiene un carácter dinámico, este método se ha aplicado a lo largo de todo el desarrollo de esta tesis doctoral. Esto ha implicado que conforme se construye la base teórica, se hacen propuestas y se contrastan y evalúan los resultados. También se han tenido que replantear búsquedas, modificar criterios de selección y restricciones con el objetivo de acomodar las revisiones a los objetivos propuestos por el Grupo BBVA. Estas revisiones han sido aplicadas principalmente en las primeras etapas de la tesis doctoral.

Los resultados de la revisión sistemática de la literatura se encuentran en el Estado del Arte (ver Capítulo 3) y han sido difundidos como parte de las publicaciones que han permitido contrastar los resultados de esta tesis doctoral (ver Capítulo 7).

3. Estado del Arte

3.1 Introducción

En este capítulo se van a mostrar las fases que han sido realizadas y los resultados obtenidos de dos Revisiones de Sistemáticas:

- ✓ Patrones de Seguridad.
- ✓ Minería de Patrones de Seguridad.

Para desarrollar ambas Revisiones Sistemáticas de forma estructurada y organizada se ha seguido la propuesta presentada en los trabajos de B. Kitchenham (Kitchenham, 2004; Brereton et al., 2007; Kitchenham, 2007). Para el caso particular de esta tesis doctoral, en que la supervisión está a cargo de un tutor o guía y que tiene un tiempo límite para su realización, se adaptó la propuesta considerando los pilares principales de la misma (*véase Tabla 2-1*, *Capítulo* 2).

También se ha realizado una evaluación de las propuestas existentes relacionadas con la Seguridad Dirigida por Modelos (*Model-Driven Security*, MDS), con el objetivo de valorar si alguna de las propuestas nos podría ayudar a la hora de diseñar arquitecturas de seguridad de grandes sistemas de información. No se ha realizado una revisión sistemática sobre este campo debido a que (Nguyen et al., 2013) ya habían realizado este trabajo anteriormente.

3.2 Revisión Sistemática de Patrones de Seguridad

La utilización de patrones de seguridad como guía para desarrollar un sistema seguro es una práctica bastante extendida (Fernandez et al., 2009; Maña et al., 2009). De hecho, en los últimos años el número de patrones publicados ha crecido de manera considerable (Rosado et al., 2006; Schumacher et al., 2006; Yskout et al., 2006). Sin embargo, existe una gran variedad y diversidad en las pautas de descripción de cada una de las propuestas (Schumacher, 2003; Anwar

et al., 2006; Fernandez et al., 2009), incluso, en varias ocasiones, se han propuesto varios patrones diferentes que dan respuesta al mismo conjunto de requisitos o problemas de seguridad (Fernandez et al., 2008; Sarmah et al., 2008).

A continuación, se muestran detalladamente las etapas incluidas en la planificación y desarrollo de la revisión realizada, incluyendo un conjunto de resultados y conclusiones al respecto.

3.2.1 Planificación de la Revisión

Esta etapa tiene como propósito específico definir los parámetros más importantes que han sido tenidos en cuenta a la hora de llevar a cabo la revisión. Se han establecido las razones que justifican llevarla a cabo, la manera en que la búsqueda de trabajos ha sido realizada y la forma en que éstos han sido revisados.

3.2.1.1 Identificación de la Necesidad de Revisión

El objetivo principal del desarrollo de esta tesis doctoral, desde sus primeras definiciones, ha sido encontrar método para sistematizar el diseño de arquitecturas de seguridad, basándonos en los patrones de seguridad. Por este hecho, desde las primeras iteraciones que se han mantenido con el Centro de Investigación de Gestión Tecnológica del Riesgo, ha sido identificada la necesidad de conocer el estado del arte de los patrones de seguridad como requisito base en los hitos marcados para el despliegue del proyecto.

El desarrollo de esta revisión sistemática ha sido clave en la evolución de la tesis, ya que hemos podido verificar el trabajo que otros grupos están realizando en relación a patrones de seguridad y analizar en profundidad las necesidades que tiene una gran compañía financiera en materia de patrones de seguridad.

3.2.1.2 Especificación de la Cuestión de Investigación

La Pregunta de Investigación que ha dirigido esta Revisión Sistemática es la siguiente:

¿Qué iniciativas dentro de la comunidad científica se han llevado a cabo para describir y/o desarrollar sistemas seguros usando patrones de seguridad?

Figura 3.1. Palabras clave y conceptos relacionados de la Revisión Sistemática

La Figura 3.1 nos muestra las palabras clave, los sinónimos y los conceptos relacionados que constituyen esta cuestión y que serán usados durante la Revisión Sistemática. En el centro de la figura se muestran los conceptos principales en los que se ha centrado esta Revisión Sistemática: *patrones y anti-patrones relativos a la seguridad*. Alrededor del núcleo central se han representado los requisitos de seguridad más relevantes que debe cumplir un Sistema de Información: *Confidencialidad, Integridad, Identificación, Auditabilidad, Privacidad, Seguridad en capas, Control de Acceso y Disponibilidad*. Este conjunto de aspectos ha sido seleccionado con el fin de realizar un estudio fiable y completo en el campo de los Patrones de Seguridad.

3.2.1.3 Definición del Protocolo de Revisión

Todas las publicaciones encontradas en las fuentes de búsqueda elegidas que cumplan con la cadena de búsqueda definida estarán incluidas dentro del grupo de población que será analizado.

Una vez se identifique el grupo de población inicial se aplicarán los criterios de inclusión y exclusión de estudios para obtener un grupo de población más acotado y valioso, denominado estudios primarios.

Una vez se identifiquen los estudios primarios, se organizarán, clasificarán y sintetizarán, con el fin de que otros investigadores puedan obtener información actualizada en el ámbito de los patrones de seguridad. Para ello, se mostrarán los resultados de esta revisión de forma resumida en una tabla.

Por último, se obtendrá una visión de la situación actual, que nos permitirá detectar las carencias existentes en relación a los patrones de seguridad, conocer las distintas formas de describirlos y descubrir nuevas necesidades en este campo de estudio.

3.2.2 Desarrollo de la Revisión

En esta etapa se ha llevado a cabo la revisión propiamente dicha. Para ello, se ha realizado la búsqueda y selección de los denominados estudios primarios o grupo de población primario. Finalmente, se ha extraído la información de interés de cada uno de los estudios y se han sintetizado los resultados y conclusiones obtenidas.

3.2.2.1 Búsqueda de Estudios Primarios

En primera instancia, se ha realizado una búsqueda en un conocido motor de búsquedas de Internet para tener una idea inicial y poder evaluar el volumen de los estudios que existe sobre

patrones de seguridad. Esta evaluación inicial también nos ha ayudado a comprobar que NO existe actualmente una revisión sistemática en este campo de investigación.

En segundo lugar, se han definido los criterios que han sido usados para realizar la selección de los estudios. Se ha decidido que las búsquedas se realizarán sobre bibliotecas digitales, debido a que contienen gran cantidad de documentación. Estas bibliotecas poseen en general motores de búsqueda avanzados que permiten refinar mejor la selección. Estos motores permiten la búsqueda lógica sobre las distintas secciones de los artículos (título, palabras clave, autores, resumen, contenido, etc.).

Después de consultar a los expertos en esta área, se ha decidido realizar las búsquedas de estudios en la biblioteca digital de la Universidad Rey Juan Carlos. Esta biblioteca digital está asociada institucionalmente con una web llamada 'Consorcio Madroño' (http://www.consorciomadrono.es/), que agrupa las fuentes de ciencias más importantes. Tras hacer una selección de la lista total de fuentes a las que se ha tenido acceso, se ha decidido ejecutar la búsqueda sobre las siguientes fuentes:

- ✓ Biblioteca digital ACM
- ✓ Biblioteca digital IEEE
- ✓ SpringerLink
- ✓ Science@Direct

Además de estas fuentes, también han sido consultados procedimientos de conferencias sugeridas por expertos en la Seguridad de la Información. Por último, cabe destacar que las fuentes deben ser de habla inglesa, ya que la investigación de calidad se suele realizar en foros de habla inglesa.

3.2.2.2 Selección de los Estudios

Después de definir las fuentes de búsqueda, ha sido necesario describir el proceso y los criterios para la evaluación y selección de los estudios. Para obtener la cadena de búsqueda combinamos las palabras clave seleccionadas con conectores lógicos AND y OR. En la Tabla 3-1 se muestra la cadena de búsqueda general que se ha elegido para realizar la selección de los estudios sobre las distintas fuentes. Esta cadena de búsqueda se ha adaptado para cada uno de los motores de búsqueda específicos.

((**TITULO** = PATTERN **OR** ANTIPATTERN) **AND** (**ABSTRACT** = AUDITABILITY **OR** SECURITY **OR** CONFIDENTIALITY **OR** INTEGRITY **OR** IDENTIFICATION **OR** PRIVACY **OR** AUTHENTICATION **OR** ACCESS CONTROL **OR** SECURITY LAYERED))

Tabla 3-1. Cadena de Búsqueda de la Revisión Sistemática de Patrones de Seguridad

Todos los trabajos en los que aparece la palabra '*Pattern*' (Patrón) o '*Antipattern*' (Anti-Patrón) en su título, y a su vez aparece en el resumen algunas de las palabras o expresiones mostradas anteriormente, serán seleccionados para el estudio. Los criterios de inclusión y exclusión son aplicados después de realizar la selección de estudios inicial.

Los criterios de inclusión usados para filtrar la selección de estudios inicial son los siguientes:

- ✓ Análisis de los títulos, palabras clave, resúmenes y conclusiones de los estudios para encontrar que estudios usan los patrones o anti-patrones de seguridad.
- ✓ Los patrones o anti-patrones de seguridad deben ser usados para solucionar deficiencias de seguridad.
- ✓ Estudios que exponen nuevas metodologías de seguridad.
- ✓ Estudios que realizan una clasificación de patrones de seguridad.

Los criterios de exclusión usados para filtrar la selección de estudios inicial son los siguientes:

- ✓ Los patrones o anti-patrones expuestos en el estudio no están relacionados con la seguridad de la información.
- ✓ En el estudio analizado no se estructuran soluciones de seguridad en forma de patrón.

Una vez se han filtrado los estudios de la selección inicial, se ha obtenido un segundo conjunto de estudios que cumplen todos los requisitos expuestos anteriormente. Este nuevo conjunto de trabajos es analizado en profundidad, leyendo el artículo entero en la mayoría de los casos. En el resto de casos se ha leído como mínimo el resumen, la introducción y la conclusión.

Una vez realizado el análisis de los estudios en profundidad, se han elegido los estudios basados en patrones de seguridad que investiguen en el campo del diseño y la arquitectura de sistemas seguros, ya que se pretenden identificar las propuestas más interesantes para esta Revisión Sistemática.

3.2.2.3 Extracción y Gestión de los Datos

Una vez ejecutada la cadena de búsqueda en las diferentes fuentes electrónicas, se ha obtenido un total de 965 estudios. Después de analizar y filtrar los trabajos con los criterios de inclusión se han obtenido 80 estudios relevantes. Finalmente, aplicando los criterios de exclusión, se ha obtenido un conjunto de 32 estudios primarios. Para gestionar los trabajos recopilados e incluir las referencias de Revisión Sistemática, se ha utilizado una herramienta de gestión bibliográfica denominada EndNote (http://www.endnote.com). Esta herramienta ha permitido gestionar las referencias y estudios de una manera eficaz y rápida.

En la Figura 3.2 se muestran las fuentes en las que se ha realizado la búsqueda y el número de estudios que se han encontrado en cada una de ellas.

	Estudios				
Fuentes	**Encontrados**	**No Repetidos**	**Relevantes**	**Primarios**	**%**
IEEE Digital Library	400	400	32	**12**	37.50%
SpringerLink	242	241	18	**8**	25.00%
ACM Digital Library	128	127	18	**10**	31.25%
Science@Direct	202	197	12	**2**	6.25%
Total	972	965	80	**32**	**100.00%**

Figura 3.2. Estadística de Estudios Obtenidos en el Desarrollo de la Revisión Sistemática

La columna "Relevantes" muestra el número de estudios en cada una de las fuentes después de filtrar los estudios con los criterios de inclusión. La columna "Primarios" muestra el número de estudios definitivo en cada una de las fuentes. Esta columna es resultado de filtrar los estudios de la columna "Relevantes" con los criterios de exclusión.

El porcentaje de estudios que encajan completamente dentro de los criterios de inclusión y exclusión previamente definidos se muestran en la Figura 3.3, distribuidos por fuentes.

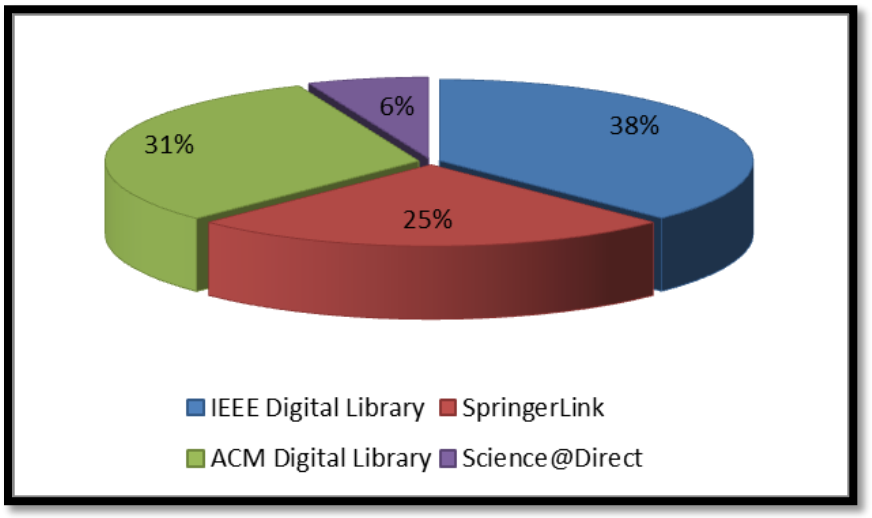


Figura 3.3. Porcentaje de Estudios Obtenidos por Fuente

A continuación, mostramos la información extraída de los estudios primarios seleccionados. Esta información contiene distintas técnicas, marcos de trabajo, metodologías, procesos y algunas iniciativas que establecen los patrones de seguridad para proteger sistemas de información frente ataques ejecutados de manera intencional.

Dada la gran variedad de estudios que se han encontrado, se ha decidido clasificarlos en tres grupos de propuestas distintas. El primer grupo muestra un conjunto de trabajos que describen nuevos patrones seguridad o desarrollan métodos para describir los patrones, el segundo grupo está compuesto por estudios que usan los patrones de seguridad para construir sistemas de información y el tercer grupo muestra un conjunto de trabajos que realizan clasificaciones de patrones de seguridad. Para cada una de las iniciativas seleccionadas se va a mostrar un breve resumen describiendo las características principales que posee.

3.2.2.3.1 Estudios que describen Patrones de Seguridad

En este apartado se va a realizar una síntesis de las propuestas centradas en el descubrimiento de nuevos patrones de seguridad. Debido a la diversidad de soluciones que se proponen, se van a agrupar los trabajos según la problemática que solucionan.

3.2.2.3.1.1 Patrones de Seguridad para comunicaciones seguras

Se han agrupado los artículos relacionados con soluciones de seguridad enfocadas al ámbito de las comunicaciones entre los distintos sistemas y el envío y recepción de mensajes que realizan.

- ✓ *(Fernandez et al., 2007) "Security Patterns for Voice over IP Networks"*

Los autores presentan cuatro patrones de seguridad que pueden ser utilizados para el diseño seguro de los sistemas de VoIP, ya que describen mecanismos que pueden controlar muchos de los posibles ataques. Los patrones presentados son: 1) "Secure VoIP call", oculta el significado de los mensajes mediante la realización de encriptación de llamadas en un entorno VoIP; 2) "Network Segmentation", que realiza la separación de los servicios de voz y datos para contrarrestar posibles ataques contra la VLAN de voz por un atacante en la VLAN de datos; 3) "VoIP Tunneling", que proporciona una forma de garantizar la confidencialidad y la integridad de las llamadas en telefonía IP por la encapsulación de datos desde un protocolo dentro del flujo de protocolos de otros; y 4) "Signed Authenticated Call", que realiza tanto la autenticación de

dispositivo como de usuario antes de decidir el acceso a los servicios VoIP. Los autores utilizan modelos UML y reflejan aspectos de la infraestructura de los sistemas, incluyendo casos de uso y arquitecturas. Además, consideran posibles ataques de seguridad y relacionan éstos con las soluciones propuestas. Por último, este enfoque brinda un entorno de trabajo para ayudar a los desarrolladores a aplicar la seguridad en sus sistemas.

- ✓ *(Chavhan y Chhabria, 2009) "Multiple design patterns for voice over IP security"*

Los autores proponen tres patrones de diseño para las implementaciones de VoIP en relación con problemas de seguridad específicos de estos sistemas. En el primer patrón, "Secure Traversal of Firewalls or NATs for VoIP", se muestra una técnica de cifrado para paquetes de voz. En el segundo patrón, "Detecting and Mitigating DDoS Attacks Targeting VoIP", se presenta la generación de claves y en el último patrón, "Securing VoIP against Eavesdropping", se presenta una técnica para descifrar paquetes de voz. También desarrollan un módulo de IPsec para VoIP en entornos Cliente/Servidor. Por último, exponen la teoría de que la seguridad en sistemas VoIP puede ser mejorada usando patrones de diseño.

- ✓ *(Fernandez y Ortega-Arjona, 2009) "The Secure Pipes and Filters Pattern"*

Los autores presentan un patrón (Secure Pipes and Filters) para reforzar el canal de comunicaciones entre sistemas, que es una versión más segura de los patrones originales. Este modelo añade una serie de mecanismos de seguridad para proporcionar funciones de seguridad añadidas al patrón original. Este tipo de patrones pueden ayudar para agregar controles de seguridad en la fase de procesado. Combinado con otros patrones similares, da al diseñador una serie de posibilidades cuando se construye el middleware de un sistema complejo.

3.2.2.3.1.2 Patrones de Seguridad para un control de accesos e identificación seguros

En este apartado se han incluido trabajos que muestran patrones para aumentar la seguridad en los sistemas de información, reforzándolos con mecanismos efectivos de autorización, autenticación y control de acceso.

- ✓ *(Delessy et al., 2007) "A Pattern Language for Identity Management"*

Los autores proponen un lenguaje de patrones para el sistema de gestión de identidades. Este lenguaje se compone de tres patrones, el patrón "Circulo de Confianza", que representa a una federación de proveedores de servicios que comparten relaciones de confianza, el patrón "Proveedor de Identidad", que centraliza la administración de la seguridad a un dominio concreto, y el patrón de la "Federación de Identidades", que permite la propagación de los atributos de un usuario entre dominios de seguridad diferentes. El lenguaje de patrones para la gestión de las identidades se basa en SAML, un lenguaje de marcado con aserciones de seguridad.

- ✓ *(Cuevas et al., 2008) "Security Patterns for Capturing Encryption-Based Access Control to Sensor Data"*

Los autores describen una solución para el control de acceso, extremo a extremo, a los datos generados por sensores inalámbricos. Se usan los patrones de seguridad para definir un modelo abstracto de control de acceso basado en criptografía para los datos de los sensores. Para ello, proponen tres patrones de seguridad: "Sensor Encryption Pattern" para el cifrado de datos de los sensores, "Data Decryption Pattern" para el descifrado de los datos y "Grant-Based Access Control Patter" basado en el control de acceso.

- ✓ *(Morrison y Fernandez, 2006) "The credentials pattern"*

Los autores proponen un patrón de seguridad para ser usado en sistemas distribuidos. Este patrón describe el uso de la identificación de la información para definir la autenticación y control de acceso. Sus ventajas son que la información de autenticación y autorización se registra de forma uniforme, persistente y portable, y que las credenciales de una autoridad de confianza pueden considerarse prueba de identidad y de autorización. También tiene algunas desventajas como que la entidad que emite las credenciales debe ser de confianza y que las credenciales deben ser hechas a prueba de falsificaciones.

- ✓ *(Fernandez y Pernul, 2006) "Patterns for session-based access control"*

Los autores describen varios patrones para mostrar el efecto de las sesiones en modelos de control de acceso. Los autores muestran un patrón que controla el acceso de las diferentes sesiones (Access Session), describiendo cómo una sesión puede limitar el derecho de un usuario. Algunas de las ventajas son que se le pueden otorgar sólo los derechos necesarios para cada contexto de acuerdo a su función y se puede invocar en una sesión sólo aquellos contextos que son necesarios para una tarea determinada, e incluso se pueden excluir combinaciones de contextos que podrían dar lugar a posibles violaciones de acceso o conflictos de intereses. Una posible desventaja es que podría ser ineficiente si queremos aplicar el control de acceso para abrir muchas sesiones para realizar actividades complejas. Además, se utilizan dos patrones más, que combinados con el patrón anterior, se puede construir un patrón específico de control de acceso. Por último, esta propuesta muestra un entorno de trabajo real basado en el conjunto de patrones descrito.

3.2.2.3.1.3 Patrones de Seguridad para garantizar la privacidad

En este apartado se han incluido los trabajos que describen patrones de seguridad que proponen soluciones al problema de la privacidad. Este aspecto es muy relevante en los intercambios de datos personales entre los usuarios y sistemas. Aquí se exponen varias soluciones de seguridad para preservar la privacidad.

- ✓ *(Lobato et al., 2009) "Patterns to Support the Development of Privacy Policies"*

Los autores presentan un conjunto de patrones (privacy policy definition, visible privacy policy, privacy policy issues, notification of risks and changes, proof of security, personal information objectives, user control) para la estandarización del desarrollo de políticas de privacidad con el fin de ser utilizado en sitios web. Estos patrones consideran principalmente aspectos relacionados con la seguridad, la información del usuario y la privacidad. El objetivo de este trabajo es asegurar los sitios web, ya que los usuarios necesitan acceder proporcionando información personal y esperan que los sistemas les proporcionen integridad, seguridad y

privacidad. Por otra parte, los autores muestran un ejemplo de una política de privacidad en la que se combinan todos los patrones descritos en el documento.

- ✓ *(Schumacher, 2003) "Example Security Patterns and Annotations"*

El autor presenta dos patrones de seguridad: uno para la manipulación de cookies (Handling Cookies), que protege la identidad de los usuarios cuando tienen acceso a un sitio web y otro (Pseudonymous E-Mail), que permite a los usuarios utilizar un servicio de correo electrónico sin revelar su propia identidad.

- ✓ *(Romanosky et al., 2006) "Privacy patterns for online interactions"*

Los autores enfocan su trabajo en patrones de privacidad existentes. Para reforzar este escenario se describen tres patrones más, "Informed Consent for Web-based Transactions", Masked Online Traffic" y "Minimal Information Asymmetry". Estos patrones se basan en las transacciones vía web, el enmascaramiento de tráfico en línea y la asimetría mínima de la información, respectivamente. Estas nuevas pautas pueden ayudar a resolver el problema de privacidad, pretendiendo que las organizaciones online, los diseñadores de páginas web y los usuarios puedan utilizar información personal sin ningún problema de seguridad.

3.2.2.3.1.4 Patrones de Ataque y de Mal Uso

En este apartado se han incluido los trabajos que describen un tipo de patrones de seguridad: los patrones de ataque y los patrones de mal uso. En este tipo de patrones los autores se colocan en el lado del atacante, y describen paso a paso todos los elementos del ataque, exponiendo las pautas de seguridad que neutralizan dichos ataques.

- ✓ *(Fernandez et al., 2009) "Modeling Misuse Patterns"*

Los autores proponen un patrón de uso indebido. Por un lado, este modelo describe desde el punto de vista del atacante, cómo realizar un ataque a un sistema y define el contexto del ataque. Por otra parte, se analizan las maneras de detener el ataque aplicando los patrones de seguridad. También se propone cómo trazar un ataque una vez que ha ocurrido para recopilar

y observar los datos apropiados para realizar un análisis forense posterior. Además, se presenta un modelo que caracteriza la estructura de este tipo de patrón.

- ✓ *(Fernandez et al., 2007) "Attack Patterns: A New Forensic and Design Tool"*

De manera similar al anterior trabajo, los autores presentan un patrón de ataque, que proporciona una descripción específica de los objetivos del ataque y pasos de éste. Este patrón describe cómo se lleva a cabo un ataque y cómo trazarlo una vez ha ocurrido, enumerando los patrones de seguridad que se pueden utilizar para mitigar el ataque. Por otra parte, se presenta un ataque de denegación de servicio en las redes del tipo VoIP para demostrar el valor del patrón realizado.

- ✓ *(Hashizume et al., 2013) Three Misuse Patterns for Cloud Computing*

En este trabajo los autores presentan algunos ataques en forma de patrones de uso indebido o mal uso, donde un patrón de mal uso describe cómo un ataque se lleva a cabo desde el punto de vista del atacante. Especialmente, se describen tres patrones de mal uso de recursos: "Resource Usage Monitoring Inference", "Malicious Virtual Machine Creation" y "Malicious Virtual Machine Migration Process". Estos patrones muestran que el Cloud también es vulnerable a los ataques contra algunas de sus funciones, como imágenes de máquinas virtuales compartidas, la migración entre máquinas virtuales, o el aislamiento. Identificar sólo amenazas no es suficiente, hay que entender cómo se lleva a cabo un ataque. Estos patrones de uso indebido pueden ayudar a los diseñadores a entender cómo se lleva a cabo un ataque y qué componentes del sistema se utilizaron y comprometida durante un ataque en un entorno Cloud.

- ✓ *(Encina et al., 2014) A misuse pattern for DoS in federated inter-clouds*

Los autores presentan un patrón de mal uso de un ataque genérico de denegación de servicio para sistemas federados Inter-Cloud. Usan un patrón llamado "federated inter-cloud pattern" y lo aplican a un contexto de Denegación de Servicios. Un mal uso de este tipo de denegación de servicio intenta interrumpir la disponibilidad del sistema Inter-Cloud haciendo

muchas peticiones de recursos o interrumpiendo el seguimiento de los acuerdos de cumplimiento entre consumidores y proveedores de servicios.

- ✓ *(Fernandez et al., 2012) A Misuse Pattern for Retrieving Data from a Database Using SQL Injection*

En este trabajo se presenta un patrón de uso indebido para recuperar datos de una base de datos mediante la inyección de SQL, que describe las características esenciales de este tipo de ataque. Los autores describen los componentes del sistema en el que el ataque se realiza mediante diagramas de clases y dinámicos del ataque usando diagramas de secuencia. Los autores describen la plantilla para definir los patrones de mal uso y describen el patrón "Retrieving Data from a Data-Base using SQL Injection" a partir de dicha plantilla. El trabajo demuestra que es posible evitar este mal uso con técnicas relativamente sencillas y que puede ser utilizado también para rastrear la fuente del acceso ilegal.

3.2.2.3.1.5 Patrones de Seguridad para mejorar la relación de confianza

En este apartado se han incluido los trabajos que proponen patrones de seguridad para reforzar las relaciones de confianza entre el usuario y los sistemas o entre dos usuarios, para tratar de cumplir los requisitos fundamentales de seguridad.

- ✓ *(Fischer et al., 2009) "A Pattern for Secure Graphical User Interface Systems"*

Los autores presentan un patrón de seguridad para desarrollar una interfaz gráfica de usuario segura llamado "Secure GUI System Pattern". Este patrón puede ayudar a reforzar los sistemas de interfaz gráfica de usuario y evaluar su uso en ámbitos diferentes. Además, se muestra cómo analizar los requisitos de seguridad para fomentar la confianza, preservando al mismo tiempo la flexibilidad que demandan las interfaces gráficas de usuario.

- ✓ *(Sorniotti et al., 2009) "A Security Pattern for Untraceable Secret Handshakes"*

Los autores describen un patrón para reforzar las relaciones de confianza entre usuarios. Este protocolo permite que dos usuarios puedan verificar mutuamente las propiedades del otro sin revelar su identidad. Está formado por tres patrones: el primero llamado "Property Certification Pattern" para la certificación de las características; otro llamado "Secure Match Pattern" para garantizar la concordancia de los datos; y el último llamado "Mutual Key PoK Pattern" para comprobar mutuamente las claves partiendo de un conocimiento de ellas.

3.2.2.3.1.6 Otros Patrones de Seguridad para construir sistemas seguros

En este apartado se muestran varias propuestas para construir sistemas seguros utilizando patrones de seguridad, tanto de diseño como arquitectónicos.

- ✓ *(Fernandez et al., 2008) "The Secure Three-Tier Architecture Pattern"*

Los autores proponen un patrón para asegurar las arquitecturas basadas en tres capas. Este patrón puede ser aplicado a sistemas distribuidos y enfocado a la ejecución de aplicaciones complejas y heterogéneas. Hay distintos debates sobre las propiedades del patrón arquitectónico tres capas, así como varios patrones desarrollados (Aarsten et al., 1996; Vogel, 2001), pero ninguno de éstos considera la seguridad.

- ✓ *(Fernandez et al., 2006) "Even more patterns for secure operating systems"*

Los autores describen patrones de seguridad para la representación de los procesos y subprocesos de los sistemas operativos. Como los sistemas operativos son muy críticos, los autores proponen varios patrones de seguridad para resolver los problemas de seguridad. Estos patrones son: "Secure Process/Thread" para la representación de los procesos y subprocesos, haciendo hincapié en sus aspectos de seguridad; "Virtual Address Space Structure Selection" considera la selección de la estructura de espacio de direcciones virtuales; y "Administrator Hierarchy" para controlar el poder de los administradores, una fuente común de problemas de seguridad.

- ✓ *(Spanoudakis et al., 2007) "Towards security monitoring patterns"*

Los autores introducen una serie de patrones para monitorizar las propiedades de seguridad básicas. Con ellos se puede comprobar, en tiempo de ejecución, la robustez de los requisitos generales de seguridad de un sistema. Para conseguir esto, se introducen patrones que cubren los requisitos principales de seguridad como es la confidencialidad, la integridad y disponibilidad.

- ✓ *(Fernandez et al., 2011) "Two security patterns: Least Privilege and Security Logger and Auditor"*

Los autores presentan dos patrones de seguridad que describen aspectos fundamentales: un patrón llamado "Least Privilege" con el propósito de minimizar abusos por parte de los usuarios o los trabajadores de una institución, otorgando a los usuarios o los procesos de ejecución de un sistema, los derechos que necesitan para desempeñar sus funciones y no más. Un segundo patrón presentado es el llamado "Security Logger and Auditor" con el objetivo de hacer un seguimiento de las acciones del usuario con el fin de determinar quién hizo qué y cuándo. Para ello se registran todas las acciones sensibles a la seguridad realizadas por los usuarios y proporcionar acceso controlado a los registros con fines de auditoría.

3.2.2.3.2 Estudios que utilizan Patrones de Seguridad

En este apartado se van a resumir las iniciativas seleccionadas que desarrollen sistemas usando patrones de seguridad.

- ✓ *(Fernández, 2007) "Security Patterns and Secure Systems Design"*

El autor muestra la anatomía y el uso de los patrones de seguridad en la construcción de sistemas seguros. Para ello, realiza una clasificación de un conjunto de patrones de seguridad, con el fin de aplicarlos a través de un método de desarrollo de sistemas seguros basado en una arquitectura jerárquica. Este método utiliza capas para definir el alcance de cada uno de los mecanismos de seguridad.

- ✓ *(Fernandez y Yuan, 2007) "Securing analysis patterns"*

Los autores muestran cómo se pueden utilizar los patrones de seguridad para desarrollar aplicaciones seguras, utilizando el modelo conceptual de la aplicación e instanciando distintos patrones de seguridad. Para construir el modelo conceptual de forma sistemática se describe una metodología basada en SAP. Esta metodología proporciona un análisis de las vulnerabilidades del sistema, permitiendo la detección de amenazas en cada una de las etapas del ciclo de vida del software.

- ✓ *(Fernandez et al., 2009) "On building secure SCADA systems using security patterns"*

Los autores utilizan los patrones de seguridad para proponer un método que permite construir un sistema SCADA seguro. Este estudio muestra un nuevo enfoque, ya que utiliza medidas importantes para resolver los problemas de seguridad, estudiando la arquitectura de un sistema SCADA genérico y analizando las amenazas potenciales que puede tener el sistema en contra.

- ✓ *(Schnjakin et al., 2009) "A pattern-driven security advisor for service-oriented architectures"*

Los autores presentan una arquitectura y una implementación para apoyar la creación de Servicios Web seguros. Este enfoque simplifica la generación de políticas de seguridad en SOA y facilita la configuración de los módulos de seguridad para los sistemas basados en servicios utilizando patrones. Este sistema puede ser utilizado para resolver los protocolos concretos y los mecanismos de seguridad a nivel técnico.

- ✓ *(Delessy y Fernandez, 2008) "A pattern-driven security process for SOA applications"*

Los autores muestran un método para construir aplicaciones SOA seguras utilizando un desarrollo dirigido por modelos y estándares de seguridad. El método de desarrollo está dividido en dos capas diferenciadas, una capa superior que engloba las soluciones en un nivel de

abstracción más alto y una capa inferior que describe las soluciones de seguridad de una manera más concreta.

- ✓ *(Maña et al., 2009) "Development of Applications Based on Security Patterns"*

Los autores presentan SERENITY, un proyecto de investigación de seguridad financiado por la Unión Europea, que separa el desarrollo de soluciones de seguridad y el desarrollo de software seguro, coincidiendo en que ambos desarrollos deben estar basados en el uso de patrones de seguridad. Por otro lado, se muestra una interfaz de programación de aplicaciones Java, especialmente diseñada para desarrollar aplicaciones dentro del framework de los patrones de seguridad.

- ✓ *(Horvath y Dörges, 2008) "From security patterns to implementation using petri nets"*

Los autores presentan un método para construir modelos de patrones de seguridad, en un nivel de abstracción alto, utilizando redes Petri. Realizan un refinamiento gradual e intuitivo de las redes de Petri y luego permite la creación de una implementación de red de Petri, modelo impulsado por la ingeniería de software dirigida por modelos (MDSE) y seguridad dirigida por modelo (MDS). Los autores argumentan que, utilizando estas redes, resuelven las deficiencias que tienen los patrones: se describen de manera informal, no son adecuados para construir arquitecturas complejas, etc.

- ✓ *(Muñoz-Arteaga et al., 2009) "A methodology for designing information security feedback based on User Interface Patterns"*

Los autores proponen una metodología para la construcción de sistemas interactivos seguros utilizando patrones de seguridad y ayudando a diseñar una retroalimentación de la seguridad de la información adecuada. Esta metodología está orientada a facilitar cómo los aspectos de seguridad se transmiten a los usuarios finales. La aportación consiste en un conjunto de patrones de diseño para el diseño usable de una retroalimentación de seguridad de la información que combina el concepto de patrones de interfaz de usuario y los criterios de HCI-S (Security Human–Computer Interaction). Los autores crean un modelo básico para ejemplificar

la presentación de la retroalimentación de seguridad de la información para el usuario final cuando se detecta una amenaza.

- ✓ *(Li et al., 2004) "Research and Implementation of Single Sign-On Mechanism for ASP Pattern"*

Los autores proponen un método para diseñar e implementar sistemas basados en Single Sign-On, un procedimiento de autenticación que habilita al usuario para acceder a varios sistemas con una sola instancia de identificación. Para desarrollar la construcción del sistema se basan en cuatro módulos: el almacenamiento de la información de seguridad, un servidor de seguridad de dominio, un centro de autenticación y las herramientas de configuración. Además, se analizan características de seguridad que pueden evitar varios tipos de ataque.

- ✓ *(Bouaziz y Coulette, 2012) Applying Security Patterns for Component Based Applications Using UML Profile*

La aportación de este trabajo está enfocada en la introducción de los temas de seguridad en las aplicaciones basadas en componentes. Una solución basada en la ingeniería dirigida por modelos (MDE) y los patrones de seguridad parece prometedora para resolver el problema de la gestión de propiedades no funcionales en las fases de desarrollo basados en componentes. Así, los autores proponen un proceso de construcción de perfiles UML estructurado sobre la base de patrones de seguridad para intentar reducir la limitación que existe en cuanto a la existencia de guías apropiadas para dirigir al desarrollador en el contexto de los patrones de seguridad. Se proporciona una ilustración del proceso propuesto utilizando el patrón de "replicación activa". Un caso de estudio del sistema GPS es presentado para demostrar la aplicación del perfil UML generado usando el proceso propuesto.

3.2.2.3.3 Estudios que clasifican Patrones de Seguridad

En este apartado se presenta un conjunto de trabajos que muestran varias propuestas que organizan o clasifican los patrones de seguridad.

- ✓ *(Sarmah et al., 2008) "Security Pattern Lattice: A Formal Model to Organize Security Patterns"*

Los autores proponen una clasificación de patrones de seguridad basada en un modelo formal que se usa para organizarlos, llamado "trust-based security model". La clasificación está formada por patrones enumerados en el *Common Criteria for Information Technology Security Evaluation* [62].

- ✓ *(Fernandez et al., 2008) "Classifying Security Patterns", (Fernandez et al., 2008) "Patterns and Pattern Diagrams for Access Control" y (Washizaki et al., 2009) "Improving the Classification of Security Patterns"*

Con este conjunto de propuestas, los autores tienen como objetivo facilitar la labor de los diseñadores de la seguridad de la información, a la vez que intentan organizar las investigaciones actuales sobre el campo de los patrones de seguridad.

En su primer trabajo *(Fernandez et al., 2008)* se presentan tres posibilidades para clasificar patrones: usando como referencia los objetivos funcionales de los patrones, usando como referencia las capas arquitecturales de un sistema y buscando similitudes lingüísticas en su descripción. Finalmente, se muestra una forma de utilizar estas clasificaciones, donde un diseñador puede realizar una selección de patrones para usarlos en su sistema.

Su investigación sigue evolucionando y en su segunda propuesta *(Fernandez et al., 2008)* proponen una nueva representación de patrones de seguridad llamada "Pattern Graph", en la que se realiza una descripción de los patrones más precisa, describiendo cómo se relacionan entre ellos y ayudando a clasificarlos. En esta propuesta también se muestra un patrón llamado "Dimension Graph", con el fin de mejorar, aún más, la precisión de la descripción de los patrones. Finalmente, validan su trabajo mostrando una tabla con una clasificación de patrones de seguridad.

Para complementar los estudios anteriores, se realiza un tercer trabajo *(Washizaki et al., 2009)* en el que se muestran las relaciones entre los distintos patrones, haciendo uso de diagramas de patrones. La idea principal de esta iniciativa es proporcionar a los diseñadores una herramienta de navegación, basada en un catálogo para seleccionar los patrones de seguridad más adecuados a la hora de desarrollar un sistema.

- ✓ *(Alvi y Zulkernine, 2012) A Comparative Study of Software Security Pattern Classifications*

Este trabajo examina diversos esquemas de clasificación de patrones de seguridad y hace una comparación de dichas clasificaciones teniendo en cuenta diversos atributos. El resultado es de gran ayuda para la selección de un esquema de clasificación adecuada basada en los atributos deseables de clasificación y las métricas de calidad. Son muchos los atributos que se tienen en cuenta a la hora de hacer esta clasificación como son producto y proceso, propósito, abstracción, ciclo de vida y dominio del problema, aplicabilidad, contexto de seguridad, fases de desarrollo y objetivos de seguridad, usuarios, ontologías, patrones de ataques, dominio de aplicación, etc.

3.2.3 Síntesis de Datos

Como se ha mostrado en secciones anteriores, existe una gran variedad de propuestas basadas en patrones de seguridad aplicables a los sistemas de información. En esta sección se van a mostrar los resultados obtenidos divididos en cuatro secciones que pueden ayudarnos a conocer mejor las características comunes de las propuestas analizadas. Las cuatro secciones que dividen los resultados de esta Revisión Sistemática son los siguientes:

- ✓ Definición de Plantillas para Patrones de Seguridad
- ✓ Aplicabilidad de los Patrones de Seguridad en Sistemas Complejos
- ✓ Desarrollos Software usando Patrones de Seguridad
- ✓ Clasificaciones de Patrones de Seguridad

En cada una de las secciones se mostrará una discusión en relación a los resultados obtenidos y se propondrán una serie de mejoras.

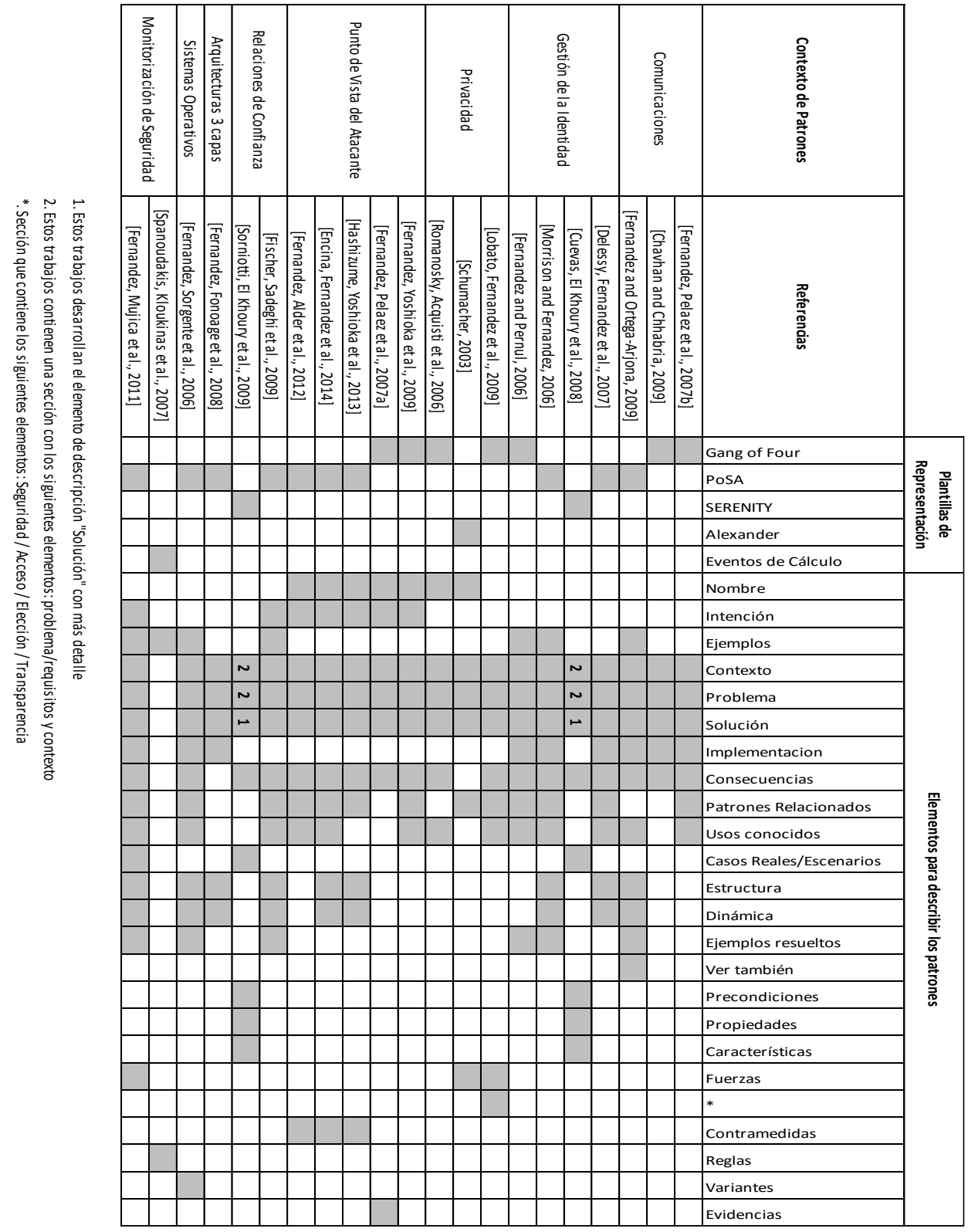
3.2.3.1 Definición de Plantillas para Patrones de Seguridad

En esta sección se van a analizar los criterios de descripción utilizados en las propuestas analizadas dentro de la sección 3.2.2.3.1 Estudios que describen Patrones de Seguridad.

En la Figura 3.4 las filas muestran las referencias de los estudios analizados que describen patrones de seguridad. En el eje vertical se muestran las plantillas que han sido utilizadas para describir las propuestas (ver sección 1.2.2), detallando los elementos utilizados en la descripción de cada uno de los patrones de seguridad.

Como se puede extraer de la Figura 3.4 y más concretamente de las columnas que representan las plantillas utilizadas en la descripción de patrones, no existe una plantilla estándar que sirva de guía para la descripción de patrones de seguridad que pueda ser utilizada por los expertos en esta materia. Esta situación provoca una variabilidad significativa en relación a los elementos que componen un patrón de seguridad. Como se observa en las columnas que se refieren a los elementos utilizados en las descripciones de patrones cada autor describe los patrones siguiendo sus propias directrices, aunque existen algunos elementos comunes de las diferentes plantillas (Schumacher, 2003; Romanosky et al., 2006; Cuevas et al., 2008). Incluso utilizando la misma plantilla, algunos autores no repiten en su totalidad todos los elementos de ésta, probablemente fruto de la evolución en sus propuestas y tratando de mejorar la usabilidad de los patrones, optan por añadir nuevos elementos (Morrison y Fernandez, 2006; Delessy et al., 2007; Lobato et al., 2009).

Los elementos más utilizados en las descripciones de patrones de seguridad son la tripleta “Contexto”, “Problema” y “Solución” propuesta en (Alexander et al., 1977). Esto demuestra que existe una necesidad por parte de los investigadores de definir una serie de conceptos básicos a la hora de documentar un patrón descubierto, pero al no existir una plantilla



1. Estos trabajos desarrollan el elemento de descripción "Solución" con más detalle
2. Estos trabajos contienen una sección con los siguientes elementos: problema/requisitos y contexto

*. Sección que contiene los siguientes elementos: Seguridad / Acceso / Elección / Transparencia

Figura 3.4. Características Descriptivas de las Propuestas que Describen Patrones de Seguridad

estándar para los patrones de seguridad, se genera una diversidad muy destacada en las distintas descripciones. Debido a esta diversidad no se puede realizar un catálogo homogéneo de patrones de seguridad, ya que es difícil de unificar toda la literatura existente sobre éstos. Este hecho también provoca que los diseñadores de sistemas de información tengan cada vez más dificultad a la hora de seleccionar los patrones más apropiados para unos determinados requisitos de seguridad dados (Weiss y Mouratidis, 2008). Localizado este problema se detecta la necesidad de diseñar un conjunto de pautas que recojan una serie de características esenciales para la descripción de los patrones de seguridad, con el fin de conseguir un catálogo homogéneo. La principal aportación de este catálogo sería conseguir soluciones equivalentes entre distintos diseñadores de sistemas de información seguros.

3.2.3.2 Aplicabilidad de los Patrones de Seguridad

En esta sección se va a determinar si el estado del arte de los patrones de seguridad actuales satisface las necesidades reales de la Ingeniería de Seguridad de la Información.

Para ello, hemos analizado si las propuestas incluidas dentro de la sección 3.2.2.3.1 Estudios que describen Patrones de Seguridad, cumplen con un conjunto de cuestiones que hemos tomado como base para realizar este análisis:

1. En relación al entorno en el que se han creado los nuevos patrones.

   - ✓ ¿Cuántas propuestas están basadas en patrones derivados de soluciones de casos reales?

2. En relación a los incrementos drásticos que pueden ocasionar los patrones expuestos dentro del sistemas de información en el que sean utilizados:

   - ✓ ¿Cuántas propuestas consideran si el uso de los patrones expuestos aumentan drásticamente la memoria consumida, la potencia de procesamiento necesario, la capacidad de almacenamiento y la frecuencia de actualización de la tecnología asociada a los sistemas involucrados?

3. En relación a las consideraciones que debe tener un ingeniero de seguridad a la hora de utilizar estos patrones en el desarrollo de un nuevo sistema:

   - ✓ ¿Cuántas propuestas consideran el aumento de la complejidad de la administración de la solución ante la posibilidad de su uso masivo?
   - ✓ ¿Cuántas propuestas analizan la complejidad del uso del patrón desde el punto de vista del usuario final, el administrador de sistemas y el administrador de la seguridad?
   - ✓ ¿Cuántas propuestas han considerado los aspectos de trazabilidad del patrón?

En la Figura 3.5 las filas muestran las referencias de los trabajos analizados. En el eje vertical se muestran las cuestiones que se han tomado como base para realizar el análisis de aplicabilidad. Cada una de las celdas ofrece las respuestas obtenidas después de analizar cada una de las propuestas, indicando si se consideran los aspectos cuestionados totalmente (S), parcialmente (P) o no se consideran (N).

Como se puede observar en la figura anterior los resultados obtenidos en el análisis son bastante significativos. La mayoría de las propuestas analizadas están basadas en casos de laboratorio. Partiendo de la base de que los patrones por definición son un mecanismo validado, en la realización del análisis sólo cinco propuestas (Chavhan y Chhabria, 2009; Lobato et al., 2009; Fernandez et al., 2012; Hashizume et al., 2013; Encina et al., 2014) están basadas en casos reales. En relación a las cuestiones relacionadas con los incrementos drásticos de parámetros, sólo una propuesta (Cuevas et al., 2008) tiene en cuenta el posible aumento de la memoria consumida en el sistema involucrado. De hecho, la mayoría de las propuestas no detallan el posible aumento de la capacidad de procesamiento consumida por el sistema involucrado, cuando se emplea la solución planteada, y sólo dos de las propuestas (Cuevas et al., 2008; Chavhan y Chhabria, 2009) abordan directamente la posibilidad de una disminución del rendimiento. Además, sólo una propuesta

Contexto de Patrones	Referencias	Propuestas basadas en casos reales	Incrementos drásticos				Las propuestas consideran			
			Memoria	Capacidad de Proceso	Almacenamiento	Frecuencia de Actualización	Gestión de la complejidad por uso masivo	Complejidad de uso para el Usuario Final	Complejidad de uso para el Administrador Sistema / Seguridad	Trazabilidad / Auditabilidad
Comunicaciones	[Fernandez, Pelaez et al., 2007b]	P	N	P	P	N	N	P	P	N
	[Chavhan and Chhabria, 2009]	S	N	S	N	N	N	P	P	N
	[Fernandez and Ortega-Arjona, 2009]	N	N	P	N	N	N	P	N	N
Gestión de la Identidad	[Delessy, Fernandez et al., 2007]	N	N	P	N	N	N	P	N	N
	[Cuevas, El Khoury et al., 2008]	N	S	S	S	N	N	N	S	P
	[Morrison and Fernandez, 2006]	N	N	N	N	N	N	P	N	N
	[Fernandez and Pernul, 2006]	N	N	P	N	N	N	P	P	N
Privacidad	[Lobato, Fernandez et al., 2009]	S	N	N	N	N	N	S	N	N
	[Schumacher, 2003]	N	N	P	N	N	N	S	N	N
Punto de Vista del Atacante	[Romanosky, Acquisti et al., 2006]	N	N	P	N	N	N	S	N	N
	[Fernandez, Yoshioka et al., 2009]	P	N	N	N	N	N	N	N	S
	[Fernandez, Pelaez et al., 2007a]	P	N	N	N	N	N	N	N	S
	[Hashizume, Yoshioka et al., 2013]	S	N	N	N	N	N	P	P	N
	[Encina, Fernandez et al., 2014]	S	N	N	N	N	N	S	N	P
	[Fernandez, Alder et al., 2012]	S	N	N	N	N	N	P	N	N
Relaciones de Confianza	[Fischer, Sadeghi et al., 2009]	N	N	N	N	N	N	P	P	N
	[Sorniotti, El Khoury et al., 2009]	P	N	N	N	N	N	N	N	N
Arquitecturas 3 capas	[Fernandez, Fonoage et al., 2008]	N	N	N	N	N	N	P	N	N
Sistemas Operativos	[Fernandez, Sorgente et al., 2006]	N	N	P	N	N	N	N	P	S
Monitorización de la Seguridad	[Spanoudakis, Kloukinas et al., 2007]	P	N	N	N	N	N	N	N	P
	[Fernandez, Mujica et al., 2011]	S	N	N	N	N	N	P	S	P

Figura 3.5. Características de aplicabilidad de las propuestas que describen patrones de seguridad

(Cuevas et al., 2008) tiene en cuenta en su descripción el posible aumento del almacenamiento. En cuanto al incremento de la frecuencia de actualización de las tecnologías asociadas al sistema de información y la complejidad del uso a gran escala de la solución, ninguna de las propuestas analizadas describe nada al respecto. Por otro lado, pocos trabajos (Schumacher, 2003; Romanosky et al., 2006; Lobato et al., 2009) discuten explícitamente sobre la complejidad que el uso de la solución significaría para el usuario final, es decir, el número de pasos que debe dar el usuario para poder utilizar el servicio, y algunos de ellos (Fernandez et al., 2008; Fischer et al., 2009) sólo mencionan que la solución debe ser transparente para el usuario final. En relación a la cuestión relacionada con la complejidad de la utilización de la solución para el administrador del sistema o para el administrador de seguridad, sólo se ha encontrado una propuesta (Cuevas et al., 2008) que haga referencia a este aspecto. Por último, la mayoría de las propuestas no tienen en cuenta la trazabilidad o auditabilidad de la solución planteada.

Por lo tanto, un ingeniero de seguridad tendría gran dificultad a la hora de utilizar estos patrones en el desarrollo de sus sistemas de información, ya que la mayoría de ellos no ofrecen respuestas a las preguntas relacionadas con los incrementos drásticos de los parámetros informáticos más comunes (memoria, capacidad de proceso, almacenamiento, y frecuencia de actualización) y a las preguntas relacionadas con las consideraciones que debe tener en cuenta un ingeniero de seguridad a la hora de desarrollar un sistemas de información (gestión de la complejidad por uso masivo, complejidad de uso para el usuario final, complejidad de uso para el administrador, trazabilidad de la solución).

En la Figura 3.6 se muestra un resumen del análisis realizado en un gráfico estadístico. En esta figura se pueden observar el número de propuestas que han respondido afirmativamente, negativamente y parcialmente a cada una de las preguntas.

Uno de los objetivos de este análisis es determinar si existe algún tipo de correlación entre los patrones basados en casos reales y los que han obtenido respuestas afirmativas en las demás cuestiones. El otro objetivo es determinar si existe algún tipo de correlación entre los patrones

que no están basados en casos reales y los que han respondido negativamente a las demás cuestiones.

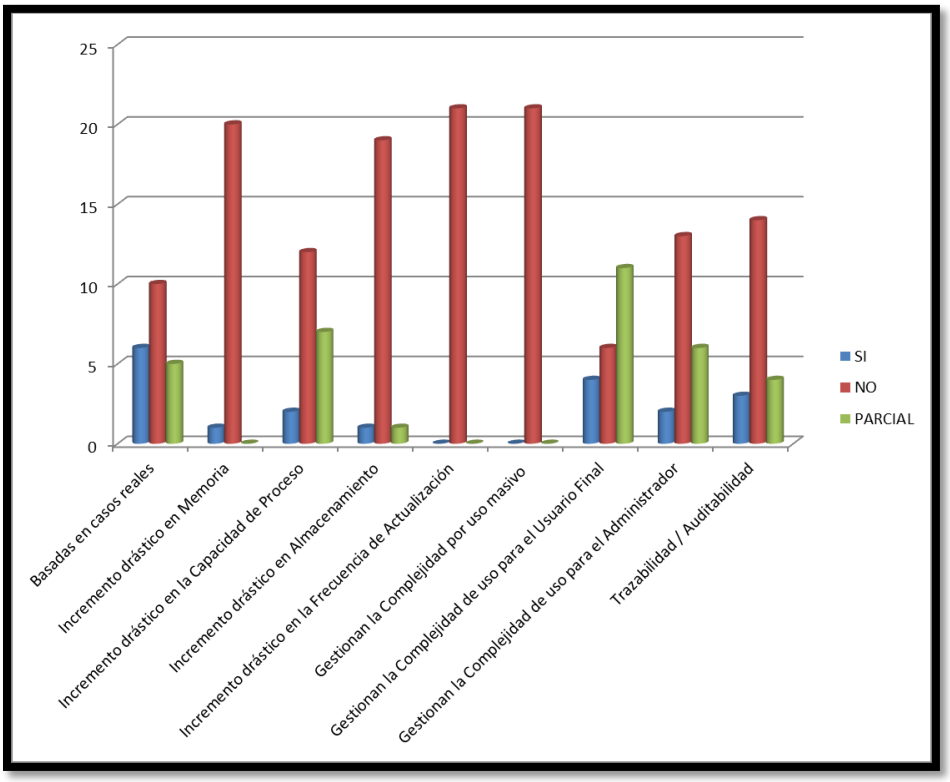


Figura 3.6. Número de propuestas en relación a las cuestiones planteadas

Existe una correlación entre los patrones que están basados en casos teóricos o casos de laboratorio y los que han respondido negativamente a las cuestiones planteadas en el análisis, por lo que queda demostrado que los patrones que han sido creados como patrones reales, basándose en soluciones implantadas, son mucho más útiles que los que se basan en análisis teóricos o en casos de laboratorio.

Cuando se hace un análisis de un caso real es muy difícil no haber tenido en cuenta alguna de las cuestiones del análisis. Es decir, cuando se realiza un diseño de ingeniería de seguridad es prácticamente imposible no haber respondido en el mismo algunas de las siguientes cuestiones:

1. En relación a los incrementos drásticos que pueden ocasionar los patrones expuestos dentro del sistemas de información en el que son utilizados:

   - ✓ ¿Cómo va a impactar el diseño de seguridad sobre los parámetros básicos de la planificación y explotación de los sistemas de información: memoria, capacidad de proceso y almacenamiento?

   - ✓ ¿Van a aparecer necesidades especiales como el aumento de frecuencia en la aplicación de parches?

2. En relación a las consideraciones que debe tener un ingeniero de seguridad a la hora de utilizar estos patrones en el desarrollo de un nuevo sistema:

   - ✓ Si el sistema se comienza a utilizar de forma masiva, ¿cómo y por qué se van incrementar las personas necesarias para la administración de la seguridad y del propio sistema de información?

   - ✓ ¿Cómo se van a solventar las necesidades de registro de la actividad del propio sistema de seguridad y si fuera requerido del sistema de información?

   - ✓ ¿Cómo se van a gestionar los usuarios involucrados?

   - ✓ ¿De qué y cómo se van a hacer las copias de seguridad de los sistemas involucrados?

Dada esta correlación parece obvio que nos encontramos ante dos tipos concretos de patrones, los patrones teóricos o basados en casos de laboratorio y los patrones empíricos, que describen en profundidad las preguntas a las que se enfrenta un ingeniero de seguridad a la hora de desarrollar un nuevo Sistema de Información.

Los patrones teóricos o basados en casos de laboratorio, a partir de ahora los denominaremos patrones teóricos, tienen escasa aceptación en el diseño de seguridad de los sistemas de información dentro de las organizaciones. De todas formas, estos patrones son en sí mismo un ejercicio muy interesante que puede permitir:

- ✓ Documentar la forma de razonamiento y de análisis de los expertos en seguridad. Dados problemas teóricos se puede hacer el análisis de cómo deberían resolverse.
- ✓ Derivado del punto anterior, puede ser un elemento central en la creación de una ontología de soluciones e Ingeniería de Seguridad de la Información, que ayude a guiar a los ingenieros de seguridad en la toma de decisiones en el análisis y diseño de sus sistemas de información.
- ✓ Puede ser un elemento muy útil en la formación de administradores, ingenieros y consultores de Seguridad de la Información.
- ✓ Son un ejemplo a probar en instalaciones reales. Puede tomarse como referencia un patrón teórico concreto y utilizarlo para intentar resolver un problema real. Este patrón teórico puede pasar en ese momento a ser un patrón empírico.

Como elemento concluyente de esta discusión, para que un patrón empírico pueda tener una aplicabilidad directa en un análisis y/o diseño de un Sistema de Información debería incorporar:

- ✓ Los detalles de la aplicación real de la que ha sido extraído, incluyendo volumetría del entorno donde ha sido aplicado: número de usuarios que acceden, picos de consumos de memoria y capacidad de proceso, y número de sistemas (físicos y lógicos) involucrados, tanto los que se están protegiendo como los que se han empleado en la protección.
- ✓ Las razones por las que se debe desplegar la solución de seguridad descrita en el patrón. Básicamente debería responderse a la pregunta ¿De qué me estoy protegiendo?
- ✓ El nivel de seguridad al que debe aplicarse el patrón. Aunque no es operativo hacer una clasificación de activos en cada patrón, es fundamental indicar el tipo de información, desde el punto de vista de riesgos, al que debe aplicarse el patrón.

- ✓ Los riesgos que se están asumiendo al emplear el patrón en cuestión, con el fin de poder tomar las contra-acciones oportunas en caso de que sea necesario tomarlas. Por ejemplo, si estoy protegiendo un sistema con usuario y contraseña estoy expuesto a que un usuario le preste su contraseña a otro o que puedan copiársela utilizando un troyano.
- ✓ Las respuestas a las cuestiones que se han planteado en el análisis realizado, es decir, la gestión de usuarios de los sistemas de información y de los sistemas de seguridad involucrados, las necesidades de almacenamiento, los aspectos básicos de cómo se realizan las copias de seguridad, el método empleado para registrar la actividad de los propios sistemas de seguridad y las necesidades especiales de monitorización o administración.

3.2.3.3 Desarrollos Software usando Patrones de Seguridad

En esta apartado se van a analizar los trabajos analizados en la sección 3.2.2.3.2 Estudios que utilizan Patrones de Seguridad.

En la Figura 3.7 se muestra de forma resumida el ámbito en el que se utilizan los patrones de seguridad. Como se puede observar en la figura, la mayoría de los trabajos analizados están enfocados a utilizar patrones de seguridad para desarrollar aplicaciones software de manera segura. En concreto, el uso de patrones de seguridad es más extenso en el campo de los Servicios Web (Web Services) o en Arquitecturas Orientadas a Servicios (Service-oriented Architecture). También existen trabajos en los que se utilizan los patrones de seguridad para ayudar a introducir seguridad en el ámbito de la Ingeniería de Sistemas.

Después de realizar este análisis se detecta una carencia significativa en el uso de patrones de seguridad para construir infraestructuras seguras, entendiendo por infraestructura toda la arquitectura del perímetro de seguridad de una organización. Las propuestas que más se aproximan a estas necesidades son (Li et al., 2004; Fernandez et al., 2009). Esta carencia provoca que los ingenieros de sistemas de información no tengan a su disposición trabajos relevantes que

les sirvan de ejemplo, en los que se utilicen patrones de seguridad para solucionar problemas relacionados con las arquitecturas de seguridad. Esta situación provoca que los ingenieros de seguridad tengan que reevaluar todos los problemas que encuentren en sus desarrollos. Esta falta de reusabilidad conlleva una pérdida de tiempo y un aumento de coste al afrontar nuevos problemas, ya que las personas encargadas de este trabajo tienen que empezar prácticamente desde cero, aunque anteriormente hayan solventado un problema similar. Por este motivo, se cree conveniente la creación de un catálogo de problemas relacionados con las arquitecturas de seguridad, en el cuál se expongan las correspondientes soluciones aportadas en las que se han utilizado patrones de seguridad. De esta manera se podrían fomentar estas soluciones entre los diferentes ingenieros, sin necesidad de que tengan conocimientos avanzados de seguridad.

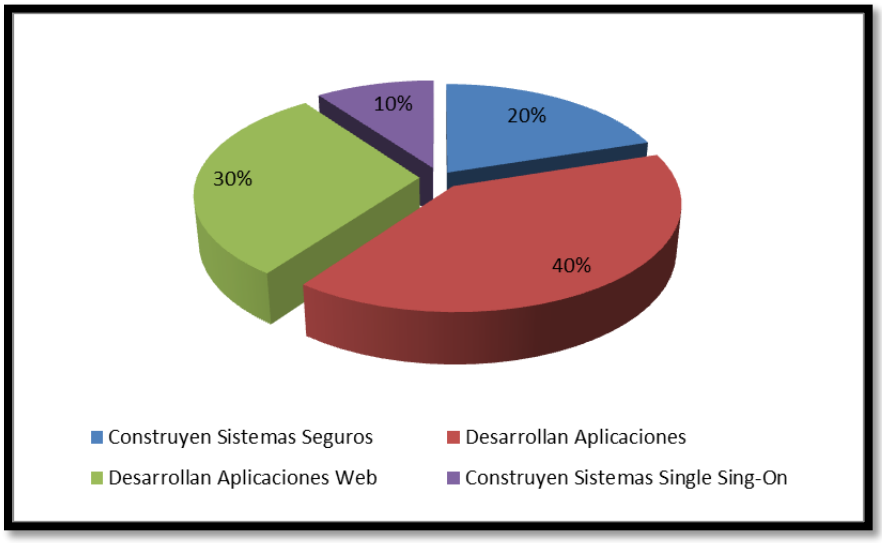


Figura 3.7. Campos en los que son usados los patrones de seguridad

Por otro lado, los trabajos en los que se utilizan patrones de seguridad para diseñar sistemas seguros no exponen una metodología clara y fácil de utilizar que guíe a los ingenieros de SI a la hora de utilizar este tipo de patrones. Esta carencia conlleva que los ingenieros de SI no tengan una guía para afrontar problemas de seguridad en este ámbito. Una de las principales consecuencias asociadas a esta carencia es que cada uno de los ingenieros siga sus propias pautas a la hora de solucionar problemas de seguridad. Este hecho puede desencadenar los siguientes sucesos:

- ✓ Los ingenieros de Sistemas eviten la utilización de los patrones de seguridad al desconocer los mecanismos de utilización de éstos, para evitar riesgos y errores de mal uso.
- ✓ Pérdida de tiempo e ineficacia por parte de los ingenieros de SI a la hora de afrontar el análisis y el diseño de SI, al no tener un mecanismo seguro y fiable para solucionar los problemas de seguridad.

Por otro lado, los diseñadores que se deciden a utilizar los patrones de seguridad desconociendo las pautas que hay que seguir a la hora de aplicarlos, acaban solucionando los problemas que les surgen de una manera subjetiva, desembocando en una gran diversidad de criterios y de soluciones diferentes entre sí.

Si se desarrollara una metodología que solventara problemas de seguridad utilizando patrones de seguridad, se obtendrían los siguientes beneficios:

- ✓ Ayudar al ingeniero de sistemas de seguridad a solventar los problemas en este ámbito de una manera rápida y eficaz.
- ✓ Obtener soluciones homogéneas de seguridad entre los distintos ingenieros de SI.

Obtener soluciones usables y fáciles de exportar para que cualquiera que tenga un problema de seguridad en SI, ya sea experto o no, pueda solventarlo siguiendo las pautas establecidas en la metodología.

3.2.3.4 Clasificaciones de Patrones de Seguridad

En esta sección se van a analizar las propuestas sintetizadas en el apartado 3.2.2.3.3 relacionadas con la clasificación de patrones de seguridad existentes. Finalmente, se mostrará una discusión en relación a los resultados obtenidos y se propondrán una serie de sugerencias y/o mejoras.

En la Figura 3.8 las columnas muestran las referencias de las propuestas que clasifican u organizan patrones de seguridad. Las filas muestran los elementos utilizados por los autores para realizar cada una de las clasificaciones u organizaciones.

	Elementos para clasificar y/u organizar patrones de seguridad				
Referencias	Elementos de Confianza	Contexto de Seguridad	Diagramas de Patrón	Grafo de Dimensión	Grafo de Patrón
[Fernandez, Washizaki et al., 2008]			■		
[Sarmah, Hazarika et al., 2008]	■	■			
[Alvi and Zulkernine, 2012]		■			
[Fernandez, Pernul et al., 2008]			■		
[Washizaki, Fernandez et al., 2009]				■	■

Figura 3.8. Elementos para Clasificar Patrones de Seguridad

Como se puede observar en la Figura 3.8 existen diferentes campos para realizar las clasificaciones u organizaciones en las distintas propuestas. Este hecho puede ser provocado en gran medida por la heterogeneidad de modos de formalizar los patrones de seguridad, que los diferentes autores aportan y los beneficios que estos autores creen que pueden tener los patrones de seguridad en el desarrollo de un sistema de información seguro.

Por otro lado, ninguna de las propuestas tiene en cuenta, a la hora de realizar la clasificación u organización de patrones de seguridad, los aspectos mostrados en la Figura 3.4. Características Descriptivas de las Propuestas que Describen Patrones de Seguridad (sección 3.2.3.1) en la que se muestra cómo debería ser un patrón de seguridad para resolver problemas complejos en entornos reales. Al no existir una clasificación homogénea de patrones de seguridad que agrupe los aspectos relacionados con el ciclo de vida y mantenimiento de un sistema de información seguro en un entorno real, es difícil seleccionar o descartar alguno de estos patrones cuando se aborda un problema en este ámbito.

Algunas de las consecuencias derivadas de este problema son que los ingenieros o arquitectos de sistemas de información pueden desconfiar de la utilización de patrones de seguridad en el desarrollo de un sistema de información reales, como se ha reflexionado en la sección anterior, ya que estas clasificaciones no tienen en cuenta las dificultades a las que se tiene que enfrentar un ingeniero en las etapas de análisis y diseño. Además, este problema también puede provocar la pérdida de efectividad en el trabajo diario de un ingeniero a la hora de aportar una solución a un problema dado.

Una clasificación u organización de patrones de seguridad sería más útil, si tuviese en cuenta aspectos relacionados con el desarrollo de sistemas reales, tales como, complejidad de uso, coste de mantenimiento, etc., ya que quedarían bien identificados cada uno de estos aspectos y sería más óptima la utilización de este tipo de clasificaciones. En definitiva, se cree necesario el desarrollo una clasificación de patrones de seguridad, que contemple los usos y costumbres de gestión que habría que realizar cotidianamente en el sistema, es decir:

- ✓ La forma de actualización de la tecnología asociada al sistema.
- ✓ Las copias de Seguridad de los activos del sistema.
- ✓ La monitorización de la capacidad de máquinas.
- ✓ La monitorización de la seguridad de las partes críticas del sistema.

Con todo esto, se pretende guiar a los ingenieros de seguridad a la hora de afrontar problemas en sistemas reales utilizando patrones de seguridad y su vez, tener un mecanismo mediante el cual se pueda seleccionar un patrón de seguridad adecuado atendiendo a una problemática específica planteada.

3.2.4 Conclusiones

Actualmente el número de problemas recurrentes relacionados con la seguridad de la información es elevado para todas las organizaciones. Después de realizar esta revisión

sistemática, podemos llegar a la conclusión que los patrones de seguridad son un método útil para presentar soluciones comunes, pero el conjunto de patrones que hemos estudiado no siguen una guía un común a la hora de documentar las soluciones, por lo tanto, es altamente complicado llegar a obtener una clasificación de patrones de seguridad.

También podemos concluir que los patrones de seguridad estudiados proporcionan un conjunto de guías para apoyar la construcción y evaluación de mecanismos de seguridad, las cuales ayudan a incorporar los principios de seguridad a la hora de construir sistemas seguros. Sin embargo, hemos podido apreciar que los patrones de seguridad actuales tienen algunas limitaciones:

- ✓ Los patrones de seguridad actuales son pequeñas unidades de defensa, es decir, cada uno de ellos solo pueden gestionar una amenaza. Considerando el número de amenazas que los sistemas de información tienen actualmente, los diseñadores deben aplicar un amplio conjunto de patrones de seguridad a la hora de construir sistemas seguros.
- ✓ Existen diferentes versiones del mismo patrón para cada uno de los niveles arquitecturales. Por ejemplo, es posible encontrar un patrón de Control de Acceso Abstracto, otro patrón de Control de Acceso para Sistemas Distribuidos, otro patrón de Control de Acceso para Servicios Web, etc. A la hora de construir un sistema seguro se van a necesitar un amplio conjunto de patrones de seguridad y este hecho incrementa la complejidad que tienen los ingenieros de seguridad cuando intentan seleccionar un patrón.
- ✓ Varias instanciaciones de un mismo patrón pueden tener aspectos comunes pero el diseñador tiene que encontrarlas. Por ejemplo, dentro de un sistema, el diseñador puede necesitar un patrón de control de acceso para restringir el acceso a los sistemas virtualizados, y a su vez, puede necesitar otro patrón de control de acceso para restringir el acceso al mainframe. En este caso, el diseñador está usando dos

instanciaciones del mismo patrón, pero es posible que algunos aspectos de las instanciaciones sean comunes.

Debido a estas limitaciones, en el siguiente capítulo (ver Capítulo 4) se ha definido un nuevo meta-patrón de seguridad, llamado Enterprise Security Pattern, para facilitar el diseño de arquitecturas de seguridad empresariales. El objetivo de este nuevo meta-patrón es proporcionar una estrategia *top-down* para definir arquitecturas de seguridad empresariales modeladas en diferentes niveles de abstracción. Las organizaciones pueden utilizar este nuevo tipo de patrones con el fin de seleccionar una estrategia de seguridad global, estandarizando los tipos de arquitecturas de seguridad dentro de la empresa y proporcionando a sus diseñadores un conjunto de directrices de seguridad óptimo y probado.

3.3 Revisión Sistemática de Minería de Patrones de Seguridad

La Minería de Patrones es una disciplina orientada al descubrimiento e identificación de nuevos patrones (Rising y Delano, 1998). Aunque diferentes dominios (programación, orientación a objetos, etc.) tienen una gran gama de patrones, el proceso de minería es fundamental para la revisión de los patrones actuales y el descubrimiento de patrones futuros. Por este motivo, en esta sección presentamos una Revisión Sistemática orientada a la Minería de los Patrones de Seguridad.

A continuación, se muestran detalladamente las etapas incluidas en la planificación y desarrollo de la revisión realizada, incluyendo un conjunto de resultados y conclusiones al respecto.

3.3.1 Planificación de la Revisión

Esta etapa tiene como propósito específico definir los parámetros más importantes que han sido tenidos en cuenta a la hora de llevar a cabo la revisión. Se han establecido las razones que justifican llevarla a cabo, la manera en que la búsqueda de trabajos ha sido realizada y la forma en que éstos han sido revisados.

3.3.1.1 Identificación de la Necesidad de Revisión

Una de las partes fundamentales en el descubrimiento de nuevos patrones es el proceso de minería de patrones utilizado. En relación a esta tesis, hemos realizado la revisión sistemática de minería de patrones de seguridad con el objetivo de conocer los estudios más importantes que muestran fuentes, frameworks, métodos y metodologías que nos ayuden a descubrir e identificar problemas recurrentes relacionados con los sistemas y arquitecturas de seguridad.

3.3.1.2 Especificación de la Cuestión de Investigación

La Pregunta de Investigación que ha dirigido esta Revisión Sistemática es la siguiente:

¿Qué iniciativas dentro de la comunidad científica se han llevado a cabo para descubrir e identificar soluciones para problemas de seguridad recurrentes a través de la Minería de Patrones de Seguridad?

La Figura 3.9 nos muestra las palabras clave, los sinónimos y los conceptos relacionados que constituyen esta cuestión y que serán usados durante la Revisión Sistemática

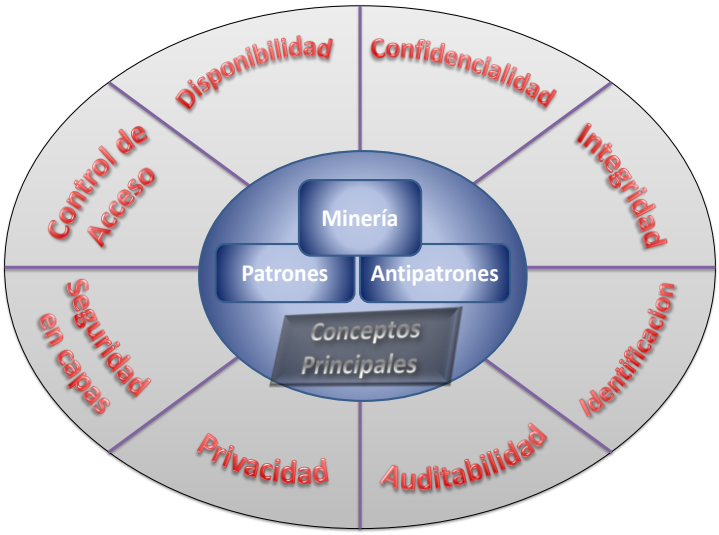


Figura 3.9. Palabras clave y conceptos relacionados de la Revisión Sistemática

En el centro de la figura se muestran los conceptos principales en los que se ha centrado esta Revisión Sistemática: *Minería de patrones y anti-patrones*. Alrededor del núcleo central se han representado los requisitos de seguridad más relevantes que debe cumplir un Sistema de Información: *Confidencialidad, Integridad, Identificación, Auditabilidad, Privacidad, Seguridad en capas, Control de Acceso y Disponibilidad*. Este conjunto de aspectos ha sido seleccionado con el fin de realizar un estudio fiable y completo en el campo de la Minería de Patrones de Seguridad.

3.3.1.3 Definición del Protocolo de Revisión

Todas las publicaciones encontradas en las fuentes de búsqueda elegidas que cumplan con la cadena de búsqueda definida estarán incluidas dentro del grupo de población que será analizado.

Una vez identificado el grupo de población inicial se aplicarán los criterios de inclusión y exclusión de estudios para obtener un grupo de población más acotado y valioso, denominado estudios primarios.

Una vez se identifiquen los estudios primarios, se organizarán, clasificarán y sintetizarán, con el fin de que otros investigadores puedan obtener información actualizada en el ámbito de la minería de los patrones de seguridad.

Por último, se obtendrá una visión de la situación actual, que nos permitirá detectar las carencias existentes en relación a la minería de patrones de seguridad y descubrir nuevas necesidades en este campo de estudio.

3.3.2 Desarrollo de la Revisión

En esta etapa se ha llevado a cabo la revisión propiamente dicha. Para ello, se ha realizado la búsqueda y selección de los denominados estudios primarios o grupo de población primario. Finalmente, se ha extraído la información de interés de cada uno de los estudios y se han sintetizado los resultados y conclusiones obtenidas.

3.3.2.1 Búsqueda de Estudios Primarios

En primera instancia, se ha realizado una búsqueda en un conocido motor de búsquedas de Internet para tener una idea inicial y poder evaluar el volumen de los estudios que existe sobre minería de patrones de seguridad. Esta evaluación inicial también nos ha ayudado a comprobar que NO existía ninguna revisión sistemática en este campo de investigación.

En segundo lugar, se han definido los criterios que han sido usados para realizar la selección de los estudios. Se ha decidido que las búsquedas se realizarán sobre bibliotecas digitales, debido a que contienen gran cantidad de documentación. Estas bibliotecas poseen en general motores de búsqueda avanzados que permiten refinar mejor la selección. Estos motores permiten la búsqueda lógica sobre las distintas secciones de los artículos (título, palabras clave, autores, resumen, contenido, etc.).

Después de consultar a los expertos en esta área, se ha decidido realizar las búsquedas de estudios en la biblioteca digital de la Universidad Rey Juan Carlos. Esta biblioteca digital está asociada institucionalmente con una web llamada 'Madroño', que agrupa las fuentes de ciencias más importantes (Madroño, 1999). Tras hacer una selección de la lista total de fuentes a las que se ha tenido acceso, se ha decidido ejecutar la búsqueda sobre las siguientes fuentes:

- ✓ Biblioteca digital ACM
- ✓ Biblioteca digital IEEE
- ✓ SpringerLink
- ✓ Science@Direct

Además de estas fuentes, también han sido consultados procedimientos de conferencias sugeridas por expertos en la materia. Por último, cabe destacar que las fuentes deben ser de habla inglesa, ya que la investigación de calidad se suele realizar en foros de habla inglesa.

3.3.2.2 Selección de los Estudios

Después de definir las fuentes de búsqueda, ha sido necesario describir el proceso y los criterios para la evaluación y selección de los estudios. Para obtener la cadena de búsqueda combinamos las palabras clave seleccionadas con conectores lógicos AND y OR. En la Tabla 3-2 se muestra la cadena de búsqueda general que se ha elegido para realizar la selección de los estudios sobre las distintas fuentes. Esta cadena de búsqueda se ha adaptado para cada uno de los motores de búsqueda específicos.

(((**TITULO** = MINING) **AND** (**TITULO** = PATTERN **OR** ANTIPATTERN)) **AND** (**ABSTRACT** = AUDITABILITY **OR** SECURITY **OR** CONFIDENTIALITY **OR** INTEGRITY **OR** IDENTIFICATION **OR** PRIVACY **OR** AUTHENTICATION **OR** ACCESS CONTROL **OR** SECURITY LAYERED))

Tabla 3-2. Cadena de Búsqueda de la Revisión Sistemática de Minería de Patrones de Seguridad

Todos los trabajos en los que aparece la palabra 'Mining' (Minería) y la palabra 'Pattern' (Patrón) o 'Antipattern' (Anti-Patrón) en su título, y a su vez aparece en el resumen algunas de las palabras o expresiones mostradas anteriormente, serán seleccionados para el estudio.

Una vez se ha definido la cadena de búsqueda hemos definido los criterios de inclusión y exclusión aplicados a la hora de realizar la selección de estudios inicial.

Los criterios de inclusión usados para filtrar la selección de estudios inicial son los siguientes:

- ✓ Todos los estudios obtenidos después de ejecutar la cadena de búsqueda en las fuentes seleccionadas serán incluidos en esta revisión.

Los criterios de exclusión usados para filtrar la selección de estudios inicial son los siguientes:

- ✓ Los estudios encontrados que NO estén dentro de la disciplina de los Sistemas de Información serán excluidos.
- ✓ Los estudios duplicados serán excluidos.
- ✓ Los estudios que NO contengan fuentes, frameworks, métodos o metodologías para descubrir e identificar soluciones para resolver problemas recurrentes de seguridad serán excluidos.

Estos criterios nos han ayudado a definir el procedimiento para evaluar y obtener el grupo de población o estudios primarios. Este procedimiento está formado por 4 pasos:

1. Todos los estudios que cumplen los criterios de inclusión serán incluidos en la revisión.

2. Del conjunto de estudios inicial, se leerá título y resumen para aplicar los criterios de exclusión definidos.

3. Del conjunto de estudios final, se leerá el artículo completo para saber si realmente los estudios cumplen todos los criterios.

3.3.2.3 Extracción y Gestión de los Datos

Una vez ejecutada la cadena de búsqueda en las diferentes fuentes electrónicas, se ha obtenido un total de 142 estudios. Después de analizar y filtrar los trabajos con los criterios de exclusión se han obtenido 8 estudios relevantes. Finalmente, después de leer los artículos completos, se ha obtenido un conjunto de 2 estudios primarios.

En la Figura 3.10 se muestran el número de estudios que se han encontrado en cada una de las fases del procedimiento de búsqueda. La columna “Relevantes” muestra el número de estudios en cada una de las fuentes después de filtrar los estudios con los criterios de exclusión. La columna “Primarios” muestra el número de estudios definitivo.

	Estudios				
Fuentes	**Encontrados**	**No Repetidos**	**Relevantes**	**Primarios**	**%**
IEEE Digital Library	57	51	3	**1**	50.00%
ACM Digital Library	46	14	2	**0**	0.00%
SpringerLink	32	30	2	**1**	50.00%
Science@Direct	7	7	1	**0**	0.00%
Total	**142**	**102**	**8**	**2**	**100.00%**

Figura 3.10. Estadística de Estudios Obtenidos en el Desarrollo de la Revisión Sistemática

A continuación, mostramos la información extraída de los estudios primarios seleccionados. Para cada una de las iniciativas seleccionadas se va a mostrar un breve resumen describiendo las características principales que posee.

- ✓ *(Ryoo et al., 2010) "A methodology for mining security tactics from security patterns"*

En este artículo los autores presentan una metodología para descubrir e identificar tácticas de seguridad desde patrones de seguridad bien conocidos. El objetivo principal de esta metodología es examinar patrones de seguridad existentes para verificar si algunos de esos patrones satisfacen las condiciones necesarias para calificarlos como una táctica de seguridad. Estas condiciones incluyen: *atomicidad, fuerzas, problema, completitud, balance entre fuerzas*. Después de definir la metodología, se muestra un ejemplo para demostrar como los criterios descritos pueden ser usados para descubrir tácticas de seguridad en patrones de seguridad existentes. Los autores esperan que esta investigación ayude a producir nuevas arquitecturas de seguridad partiendo de la minería de tácticas de seguridad.

- ✓ *(Schumacher, 2003) "Mining Security Patterns"*

En la sección 7.5 de este libro, Schumacher muestra diferentes enfoques para realizar minería de patrones de seguridad. Básicamente, se están considerando dos formas o métodos para realizar la minería (i) aprender de errores típicos e intentar descubrir como esos errores podrían haber sido prevenidos, y (ii) obtener información directamente desde estándares de seguridad. En esta sección, Schumacher también introduce una amplia lista de fuentes de información que podrían ayudar a descubrir nuevos patrones de seguridad. Esta lista incluye proveedores de seguridad de la información (CERT, grupos de hackers, compañías de seguridad, grupos de noticias, listas de distribución, artículos y libros) y estándares de seguridad (*Common Criteria, IT Baseline Protection Manual, ISO 17779*).

3.3.3 Síntesis de Datos

Después de ejecutar la revisión sistemática se han analizado y sintetizado los estudios obtenidos en la fase anterior.

En relación al estudio (Ryoo et al., 2010) hemos observado que las tácticas de seguridad pueden ser descritas en una jerarquía basada en las siguientes tres categorías: *la resistencia de ataques, la detección de ataques y la recuperación después de un ataque*. Entendemos que esta categorización podría ser útil en etapas muy tempranas del ciclo de vida del desarrollo, donde estas tácticas podrían ayudar a la hora de realizar un análisis de requisitos y un análisis funcional del nuevo sistema de información, pero a su vez hemos observado que las tácticas son menos útiles en etapas más avanzadas dentro del ciclo de vida, ya que no guían a los ingenieros de seguridad en el diseño y despliegue de nuevas tecnologías y sistemas.

En relación a la seccion 7.5 “Mining Security Patterns” del libro (Schumacher, 2003) hemos observado que tanto las técnicas como las fuentes de información mostradas son muy útiles para llevar a cabo una minería de patrones de seguridad, pero a su vez no son suficientes a la hora de descubrir, diseñar y documentar nuevos patrones de seguridad. Esto es debido a que los patrones de seguridad orientados al diseño de arquitecturas seguridad incluyen distintos niveles de abstracción en su solución y realizan un análisis cualitativo de los aspectos tecnológicos más importantes en relación a la solución que fue propuesta en (Moral-García et al., 2011).

3.3.4 Conclusiones

En esta sección se van a mostrar las conclusiones obtenidas tras la realización de la revisión sistemática de Minería de Patrones de Seguridad.

Las tácticas de seguridad podrían ser útiles en fases tempranas del ciclo de vida del desarrollo, pero son menos útiles en etapas posteriores ya que no consideran algunos de los elementos básicos asociados al diseño y construcción de arquitecturas seguras:

- ✓ ¿Qué activos de información necesitan ser protegidos?
- ✓ ¿Cuál es el contexto asociado al sistema de información?
- ✓ ¿Cuáles son las amenazas a las que va a estar expuesta la arquitectura?
- ✓ ¿Quién va a desarrollar y mantener el sistema de información?
- ✓ ¿Cuáles son las tecnologías de seguridad que necesitan ser desplegadas?

Los sistemas de información proporcionados en (Schumacher, 2003) podrían ser útiles a la hora de descubrir y documentar soluciones de patrones de seguridad con cierto nivel de abstracción, pero no es común encontrar documentos de seguridad de la información públicos que incluyan soluciones orientadas a productos tecnológicos y detallen los aspectos tecnológicos más importantes de la solución.

Tras realizar esta revisión sistemática, hemos llegado a la conclusión que los estudios analizados no cumplen los requisitos básicos a la hora de descubrir, diseñar y documentar patrones de seguridad orientados en el diseño de arquitecturas o sistemas seguros.

También es fácil apreciar que el número de estudios primarios es demasiado bajo. En esta tesis doctoral ha sido presentada una propuesta de minería de patrones de seguridad (sección 4.6), la cual ha sido utilizada para descubrir y documentar los *Enterprise Security Patterns* propuestos en el *capítulo 6. Caso de Estudio*. De todas formas, creemos necesario que los grupos de seguridad de la información deberían valorar esta falta de enfoques, con el objetivo de crear una metodología de minería de patrones de seguridad que pudiera ser referencia en este ámbito.

3.4 Revisión de Seguridad Dirigida por Modelos (MDS)

La Seguridad Dirigida por Modelos (*Model-Driven Security*, MDS) surgió hace más de una década como una especialización de la Ingeniería Dirigida por Modelos (*Model-Driven Engineering*, MDE) para el desarrollo de sistemas seguros. Como la MDE, los principios básicos de MDS son potenciar el rol de los modelos y aumentar el nivel de automatización, promoviendo la adopción de un enfoque conceptual (Bézivin, 2004). En el caso particular de la MDS, los modelos no están centrados en la descripción de la lógica de negocio de los sistemas, sino en capturar los requisitos de seguridad del sistema (Lucio et al., 2014). Las técnicas y herramientas basadas en MDS permiten considerar los aspectos de seguridad desde las primeras fases del desarrollo de un sistema, paliando al menos en parte el problema que resulta de la tendencia a ignorar la seguridad en el diseño de los sistemas.

En este caso, no hemos realizado una revisión sistemática como en el caso de los patrones de seguridad y la minería de patrones de seguridad, dado que ya existían trabajos recientes que habían abordado la tarea de elaborar una revisión sistemática del estado del arte en MDS campo (Jensen y Jaatun, 2011; Nguyen et al., 2013).

Así, en esta sección utilizaremos una taxonomía para evaluar los enfoques MDS existentes más destacados, para analizar si alguna de esas propuestas nos podría ayudar a la hora diseñar y documentar arquitecturas de seguridad de grandes sistemas de información.

3.4.1 Taxonomía de Evaluación

Los conceptos que componen esta taxonomía son un subconjunto de los que aparecen en las siguientes propuestas (Khwaja y Urban, 2002; Kasal et al., 2011; Nguyen et al., 2013). En nuestro caso, hemos seleccionado aquellos conceptos que nos permitían evaluar si las propuestas existentes podrían ser útiles a la hora de diseñar arquitecturas de seguridad. En la Tabla 3-3 describimos y resumimos cada una de las entradas de la taxonomía.

Entrada	Descripción
Dominios de Aplicación	¿Es una propuesta de propósito general o para un dominio específico?
Propiedades de Seguridad	¿En qué propiedades de seguridad está centrada la propuesta? ¿Son ellas expresables en un meta-modelo?
Enfoque de Modelado	¿Qué paradigma(s) de Modelado utiliza (MDA, AOM, DSM)? ¿Qué lenguaje(s) de Modelado utiliza (UML, DSLs, Lenguajes Formales)?
Transformaciones de Modelo	¿La propuesta utiliza transformaciones M2M y/o M2T? ¿Qué motor de transformaciones utiliza?
Herramienta	¿La propuesta dispone de herramientas de soporte? ¿Qué funcionalidad proporcionan?
Uso de Patones	¿La propuesta utiliza patrones a la hora de diseñar las soluciones? ¿Qué tipo de patrones utilizan?

Tabla 3-3. Entradas Taxonomía Model-Driven Security

A continuación, mostramos una breve descripción de cada una de las entradas incluidas en la tabla anterior:

Dominios de Aplicación. Esta entrada permite identificar los dominios donde podría ser aplicada la propuesta: *aplicaciones web, sistemas de comercio online, sistemas embebidos, sistemas distribuidos*, etc. A menudo, las propuestas relacionadas con la MDS son ideadas para utilizarse en un dominio concreto con el objetivo de mejorar aspectos de seguridad específicos del dominio, aunque algunas son más generalistas. Esta entrada nos va a ayudar a comparar el ámbito de aplicación de las propuestas.

Propiedades de Seguridad. La agencia Europea de Comunicaciones y Seguridad de la Información define la seguridad como "*Las capacidades de los sistemas de información para resistir acciones ilegales o accidentes maliciosos que comprometen la disponibilidad, autorización, autenticidad, integridad o confidencialidad de los datos almacenados o transferidos con un cierto nivel de confianza*". Algunas propuestas de MDS únicamente se centran en alguna de las propiedades de la seguridad, mientras que otras gestionan más de un aspecto. Esta entrada

nos va a ayudar a comparar que propiedades de seguridad podrían ser reforzadas en un sistema aplicando la propuesta objeto de estudio.

Enfoque de Modelado. Como ya dijimos anteriormente, MDS es una especialización de MDE, por lo tanto, parece necesario distinguir las propuestas MDS de acuerdo al *paradigma* y *lenguaje de modelado* que adoptan.

El *paradigma de modelado* hace referencia a la forma de poner en práctica los principios de la MDE. Así, los cuatro paradigmas más conocidos son las Arquitecturas Dirigidas por modelos (MDA), que distingue distintos niveles de abstracción y apuesta por elaborar diferentes modelos para cada nivel hasta llegar a modelos con un nivel de detalle que permitan ser utilizados como guía para la implementación o, directamente, generar el código que implementa el sistema; el Modelado Orientado a Aspectos (AOM), que aboga por modelar por separado los diferentes aspectos del sistema (en este caso, propiedades de seguridad), y combinar luego al información de esos modelos en un único modelo integrador, aplicando un conjunto de reglas de modelado; el Modelado Especifico de Dominio (DSM), que promueve el uso de modelos que manejen abstracciones próximas al dominio de interés; y finalmente el Modelado Multi-Paradigma (MPM), cuyo objetivo es ayudar a combinar e integrar el conjunto heterogéneo de modelos (distintos niveles de abstracción y distintos aspectos del sistema) que conforman el desarrollo de un sistema.

En cuanto al lenguaje de modelado, a la hora de trabajar con modelos, podemos optar por utilizar soluciones ya existentes, y en este caso el lenguaje por excelencia es UML, aunque también podríamos utilizar lenguajes de especificación formal, como Maude (Clavel et al., 1999) o Z (Potter et al., 1990), o podemos optar por definir un nuevo lenguaje, que permita abstraer más fielmente los elementos que componen el dominio de interés. En este segundo caso existen fundamentalmente dos tendencias: extender UML definiendo un perfil, o definir un lenguaje específico de dominio (DSL) (Mernik et al., 2005). Acorde a (Nguyen et al., 2013) UML y los perfiles UML son la opción más adoptada por las propuestas existentes en el contexto de la MDS, aunque algunos trabajos hacen uso de DSLs. Por otro lado, los lenguajes de especificación formal no son

demasiado utilizados, probablemente por su curva de aprendizaje, la dificultad de integrarlos con otras herramientas, y porque a menudo requieren una mayor especialización.

Transformaciones de Modelos. En cualquier propuesta de MDE (y por tanto MDS), las transformaciones de modelos juegan un papel clave, pues son la forma de implementar cualquier procesamiento de la información recogida en los modelos manejados en la propuesta. Así, permiten automatizar procesos que de llevarse a cabo de manera manual resultarían tediosos o repetitivos y propensos a errores. Por ejemplo, en el contexto de una propuesta MDA permitirían ir refinando los modelos de mayor nivel de abstracción hasta llegar a modelos con un nivel de detalle suficiente para ser utilizadas como entrada para la generación de código.

Con esta entrada pretendemos identificar la forma en que la propuesta objeto de estudio hace uso de las transformaciones de modelos. Así, la propuesta podría utilizar transformaciones modelo a modelo (MTM), por ejemplo, para automatizar el refinamiento de los de alto nivel, o transformaciones modelo a texto (M2T) para generar artefactos textuales a partir de los modelos, como código fuente, casos de prueba, informes, etc.

Herramienta de Apoyo. Como ya hemos indicado, un aspecto clave de la MDE es aumentar el nivel de automatización, proporcionando herramientas que permitan reducir el nivel de complejidad accidental del trabajo con modelos y aumenten la productividad del equipo. Además, automatizar las tareas manuales propensas a errores redunda en una mejora de la calidad de la solución producida. Las herramientas de apoyo pueden cubrir muchas de las fases de un proceso de desarrollo MDS. Esto incluye editores gráficos, IDEs para el desarrollo de transformaciones, validadores, gestores de trazabilidad, generadores de código, etc. Por todo ello, el objetivo de esta entrada es identificar si la propuesta en cuestión es soportada por algún tipo de herramienta, y en tal caso, analizar la funcionalidad que dicha herramienta proporciona.

Uso de Patrones. Los patrones han sido probados satisfactoriamente en muchas áreas de desarrollo de software y parecen ser particularmente útiles para el desarrollo de sistemas seguros. Como ya se ha comentado anteriormente, los patrones agrupan conocimiento de forma

estructurada y fácilmente entendible. Por lo tanto, el uso de patrones podría ayudar a la hora de diseñar de forma estructurada las soluciones MDE y/o MDS.

3.4.2 Desarrollo de la Revisión

En esta sección utilizamos la taxonomía que acabamos de presentar (ver sección 3.4.1) para evaluar los 5 enfoques o propuestas MDS más relevantes de los encontrados en la literatura. A la hora de determinar la relevancia de las propuestas nos hemos basado en si estaban o no incluidas en las revisiones del estado del arte de MDS publicadas (Jensen y Jaatun, 2011; Kasal et al., 2011; Nguyen et al., 2013) y en su popularidad. Como indicador de popularidad hemos utilizado el número de citas de la publicación más referenciada de entre aquellas que componen la propuesta. La Tabla 3-4 muestra los resultados obtenidos.

Propuesta	Numero de Citas
SecureUML	649
UMLsec	549
SECTET	54
SecureMDD	30
ModelSec	11

Tabla 3-4. Numero de Citas por Propuesta

A continuación, se muestra el resultado de la evaluación de los 5 enfoques seleccionados: *UMLsec, SecureUML, Sectet, ModelSec y SecureMDD*.

3.4.2.1 UMLsec

UMLsec (Jürjens, 2001; Jürjens, 2002; Jürjens, 2004; Jürjens, 2005; Best et al., 2007; Hatebur et al., 2011) es una extensión de UML para el análisis de sistemas seguros. La idea es anotar mediante el uso de estereotipos los modelos UML del sistema para especificar los requisitos de seguridad. La Figura 3.11 muestra la visión general del enfoque UMLsec. La idea

subyacente es que, si podemos modelar el comportamiento del atacante, las diferentes propiedades de seguridad, como la confidencialidad o el control de acceso, se pueden comprobar de forma automática confrontando el modelo del atacante con los modelos del sistema anotados o enriquecidos con los estereotipos incluidos en el perfil definido por UMLSec.

Dominios de Aplicación. UMLsec puede utilizarse para modelar y analizar propiedades de seguridad en distintos dominios, incluyendo aplicaciones web (Houmb y Jürjens, 2003), sistemas embebidos (Jürjens, 2007) y sistemas distribuidos.

Propiedades de Seguridad. UMLsec trata un gran número de requisitos de seguridad: *confidencialidad, integridad, autenticidad, autorización, flujo de información seguro y no repudio*.

Enfoque de Modelado. Tal como ya hemos dicho, UMLsec extiende UML proporcionando mecanismos de anotación para añadir información de seguridad a los diagramas UML que permiten modelar las distintas perspectivas del sistema (ver Figura 3.11).

Transformaciones de Modelos. UMLsec no propone ni define transformaciones de modelos que apoyen el proceso de desarrollo.

Herramienta de Apoyo. El conjunto de herramientas que proporciona UMLsec tiene por objetivo proporcionar capacidades de análisis, informando de si el sistema cumple con los requisitos de seguridad que el usuario ha especificado utilizando los estereotipos de que dispone, a partir del parseo de los documentos XMI que almacenan los modelos UML extendidos. Es posible aplicar el enfoque a sistemas existentes mediante ingeniería inversa del código (Best et al., 2007).

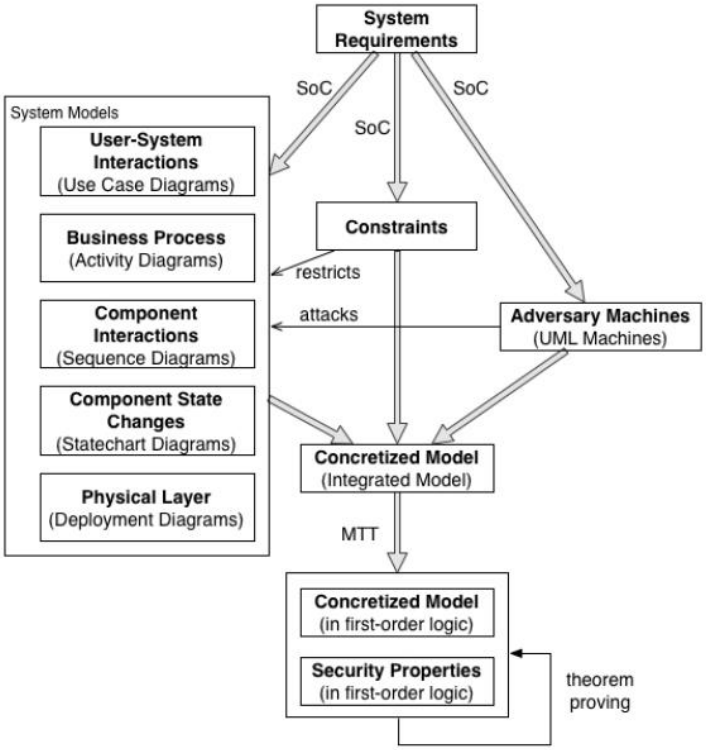


Figura 3.11. Vista General de UMLsec

3.4.2.2 SecureUML

En 2002, Basin et al. (Lodderstedt et al., 2002; Basin et al., 2003; Basin et al., 2006; Basin et al., 2011) proponen SecureUML, un lenguaje de modelado que define un vocabulario para anotar modelos UML con información sobre control de accesos. Más concretamente, SecureUML permite definir políticas de control de acceso basado en roles (*Role-Based Access Control*, RBAC) y algunas otras restricciones de autorización.

La Figura 3.12 muestra una representación gráfica del enfoque SecureUML. Para combinar los requisitos del sistema recogidos en los modelos de negocio, con los requisitos de seguridad, recogidos en los modelos de control de acceso, se define un nuevo modelo integrador utilizando el perfil que define la propuesta (dialecto). Este modelo permite especializar los

diferentes elementos del lenguaje de modelado del sistema, con aspectos propios de la seguridad.

Dominios de Aplicación. SecureUML es un perfil UML y por tanto es un lenguaje de propósito general, que puede aplicarse a cualquier dominio.

Propiedades de Seguridad. Como ya se ha señalado, se centra en control de acceso o autenticación y de hecho, todos los ejemplos de uso que encontramos en la literatura se centran en este aspecto.

Enfoque de Modelado. SecureUML es en esencia un perfil UML.

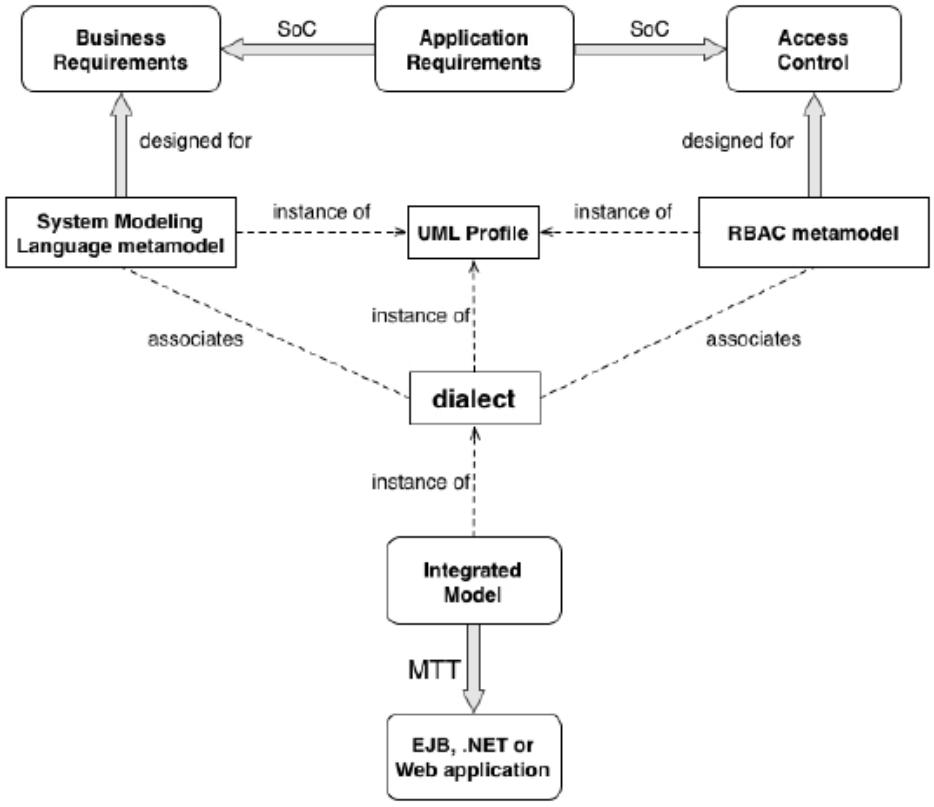


Figura 3.12. Vista General de SecureUML

Transformaciones de Modelos. La Figura 3.12 muestra cómo pueden generarse sistemas Enterprise Java Beans (EJB) y .NET a través de modelos extendidos con SecureUML. Para ello, el generador de código enlaza los conceptos del lenguaje de seguridad, con los mecanismos de seguridad que proporcionan EJB y .NET. No obstante, a pesar de que la generación de código puede ser vista como una transformación M2T, no se implementa con un lenguaje de transformaciones.

Herramienta de Apoyo. Los autores proporcionan un prototipo para generar aplicaciones EJBs, implementado utilizando el generador basado en plantillas integrado en ArcStyler (GmbH, 2005).

3.4.2.3 SECTET

SECTET (Alam et al., 2006; Hafner et al., 2006; Alam et al., 2007; Breu et al., 2007; Hafner et al., 2008) es un marco MDS para apoyar el diseño, implementación y gestión de *workflows* seguros en los que participan varias organizaciones, generalmente en entornos *peer-to-peer*, es decir, descentralizados.

Dominios de Aplicación. La propuesta se ha utilizado en diferentes dominios, como el gobierno electrónico, la salud o la educación (Breu et al., 2005; Hafner et al., 2005)

Propiedades de Seguridad. SECTET gestiona dos tipos de políticas de seguridad:

- ✓ ***Políticas Básicas de Seguridad*:** integridad, confidencialidad y no repudio.
- ✓ ***Políticas Avanzadas de Seguridad*:** Control de Acceso Basado en Roles (RBAC) estático y dinámico.

Enfoque de Modelado. SECTET define SECTET-DSL, un lenguaje basado en UML que a su vez está compuesto de dos sub-lenguajes: (i) SECTET-UML, un perfil UML para aspectos de seguridad estáticos en el modelado del sistema y (ii) SECTET-PL, un lenguaje basado en OCL para

modelar requisitos de seguridad dinámicos. SECTET-UML se utiliza para modelar los requerimientos del negocio y los requisitos de seguridad estática. Tal y como se muestra en la Figura 3.13, el enfoque de modelado que propone SECTET articula el modelado de los requisitos de seguridad en base a dos dimensiones ortogonales: la vista del Workflow y la vista de la Interfaz. Los modelos de la primera representan el intercambio de mensajes entre los diferentes participantes, centrándose en los requisitos de seguridad como confidencialidad, integridad y protocolos de no-repudio. Por otro lado, en la Vista de la Interfaz, se modela cada participante como un nodo que ofrece servicios sujetos a ciertas restricciones de seguridad.

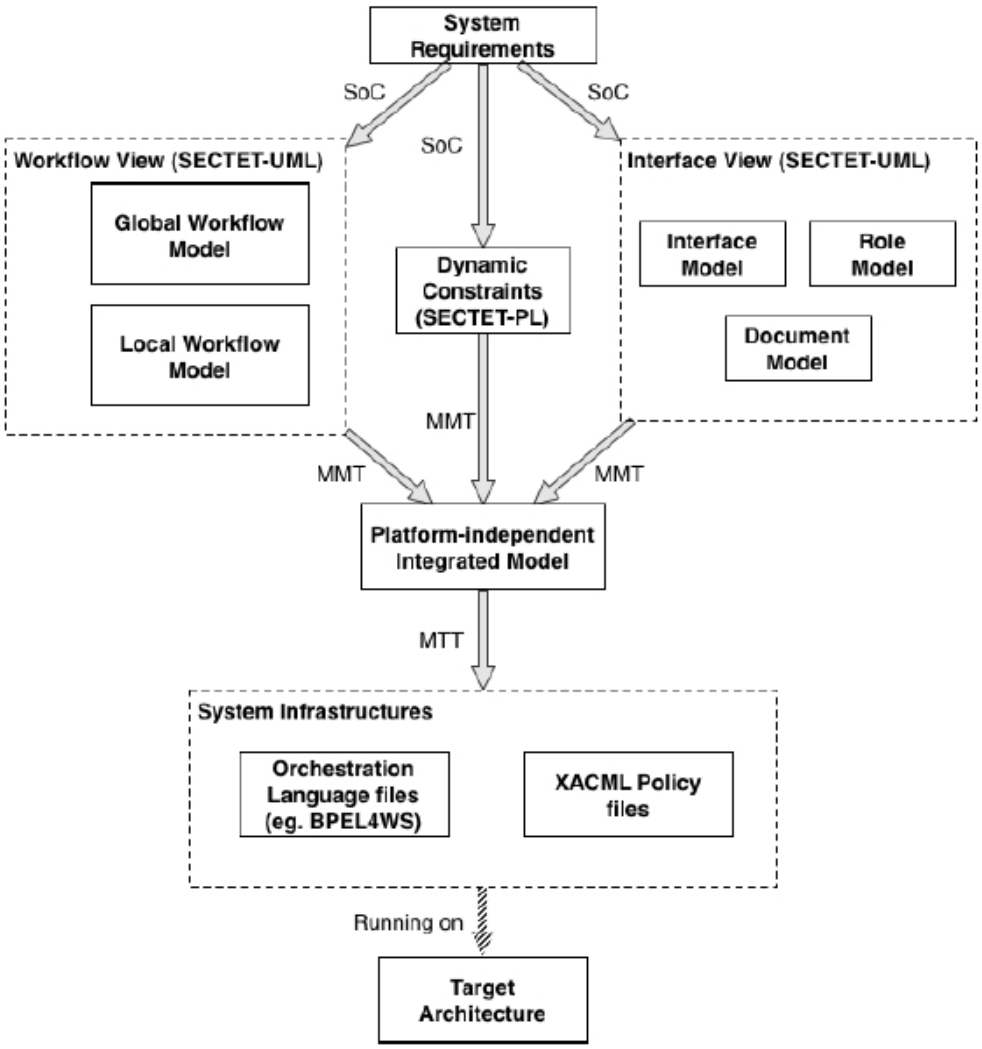


Figura 3.13. Vista General de la propuesta de SECTET

Transformaciones de Modelos. La Figura 3.14 ilustra el uso de transformaciones en SECTET. Una transformación M2M implementada con QVTo (OMG, 2011) transforma los modelos de Roles y Accesos del modelo de Dominio en un metamodelo XACML, que es luego poblado con elementos del dominio concreto en el que trabajemos para generar diferentes modelos utilizando EMF (Steinberg et al., 2008). Finalmente, una transformación M2T codificada con Xpand (http://www.eclipse.org/modeling/m2t/?project=xpand) serializa estos modelos en ficheros de políticas XACML.

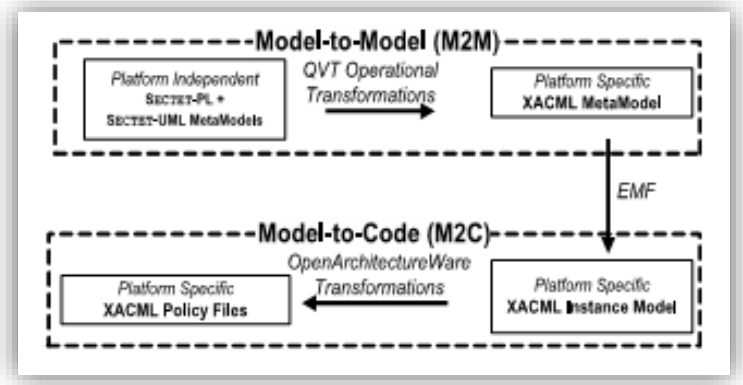


Figura 3.14. Modelo de Transformaciones en SECTET (Alam et al., 2007)

En la metodología SECTET los requisitos del sistema son concretados en tres puntos de vista distintos (ver Figura 3.13): la vista del flujo de trabajo (*workflow view*), las restricciones dinámicas (*dynamic constraints*) y la vista de la interfaz (*interface view*). En este nivel la construcción de los modelos se apoya en software como (MagicDraw, http://www.nomagic.com), que permite exportar los modelos UML a archivos XMI.

Herramienta de Apoyo. Los autores proporcionan un prototipo que integra las transformaciones anteriores.

3.4.2.4 ModelSec

ModelSec (Sanchez et al., 2009) es una propuesta para la generación de artefactos de seguridad (como reglas que implementan políticas de seguridad) a partir de modelos que recogen los requisitos de seguridad del sistema.

Dominios de Aplicación. En (Sanchez et al., 2009) los autores utilizan una aplicación Web para la gestión de datos sanitarios como caso de estudio adaptado de (Fernández-Medina y Piattini, 2005). El objetivo del ejemplo es el diseño de una base de datos segura donde los autores muestran como ModelSec gestiona el control de acceso y el código de seguridad de la base de datos. Aunque los autores explican el enfoque usando una base de datos, ModelSec no está restringido a un dominio de aplicación particular.

Propiedades de Seguridad. El punto de partidad de ModelSec es la definición de modelos de requisitos de seguridad. Estos modelos abarcan múltiples problemas de seguridad en forma integrada, incluyendo la privacidad, la integridad, el control de acceso, la autenticación, la disponibilidad, el no repudio y la auditoría.

Enfoque de Modelado. La Figura 3.15 representa la visión global de ModelSec. Como puede observarse, los requisitos de seguridad se recogen en modelos diferentes de los modelos utilizados para recoger el resto de requisitos del sistema. Para definir estos modelos los autores proporcionan un DSL para la elaboración de modelos de requisitos, que también incluye abstracciones para modelar los requisitos de seguridad. Los requisitos de seguridad recogidos en el modelo de seguridad se sincronizan con los artefactos recogidos en los modelos conceptuales (típicamente diagramas de clases UML) que sirven para elaborar el modelo de dominio. El modelo de requisitos de seguridad se utiliza para generar un modelo (parcial) de diseño de seguridad que debe ser completado por el diseñador con información adicional que a su vez da lugar a un modelo de implementación de seguridad que se utiliza como entrada para la generación del código que implementa los requisitos de seguridad. Puede darse el caso de que sea necesario

sincronizar manualmente este código con el que implementa el sistema, dado que los modelos de diseño del sistema y los de seguridad evolucionan por separado.

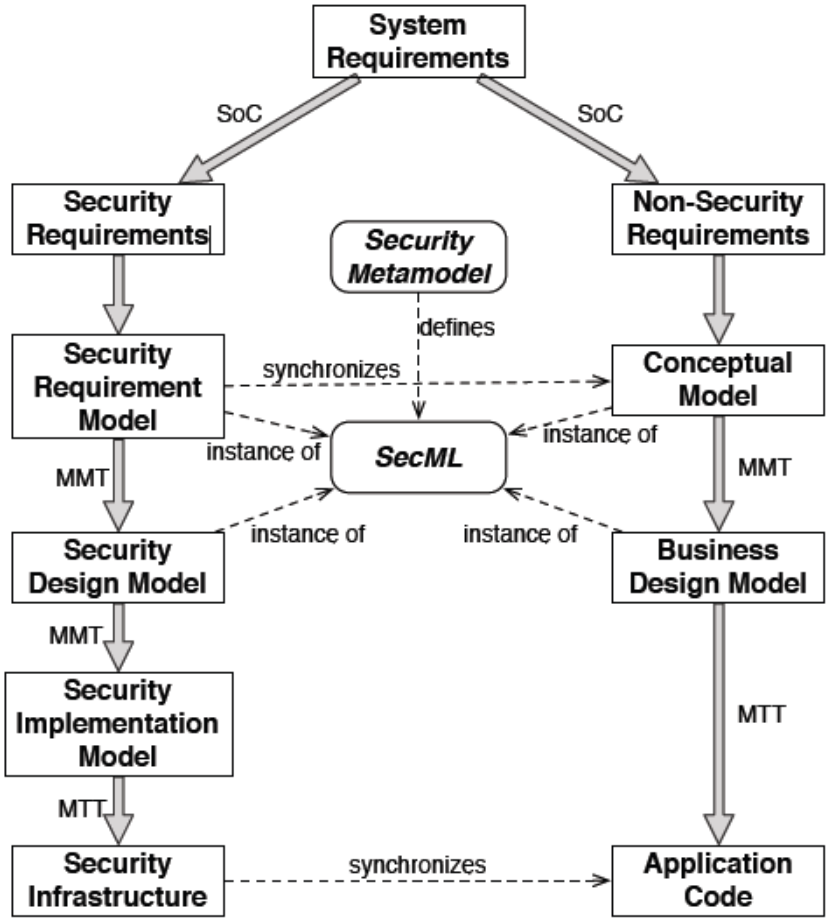


Figura 3.15. Vista General de ModelSec

Transformaciones de Modelos. El proceso descrito pone de manifiesto que ModelSec utiliza varias transformaciones de modelos. En particular, transformaciones M2M transforman los modelos de análisis en modelos de diseño, y estos, en modelos de implementación. Así mismo, transformaciones M2T transforman los modelos de implementación y diseño en infraestructuras de seguridad y el código que implementa el sistema.

Herramienta de Apoyo. Las transformaciones M2M están escritas con (Cuadrado et al., 2006) RubyTL, un DSL para el desarrollo de transformaciones construido como una extensión del lenguaje de programación Ruby. Por otro lado, las transformaciones M2T se codifican utilizando MOFScript (Oldevik, 2006), un lenguaje imperativo para el desarrollo de transformaciones de modelo a texto.

3.4.2.5 SecureMDD

SecureMDD (Moebius et al., 2009; Moebius et al., 2009; Moebius et al., 2009; Moebius et al., 2010; Moebius et al., 2012) es otro enfoque basado en UML cuyo objetivo es facilitar el desarrollo seguro de aplicaciones críticas de seguridad basadas en protocolos criptográficos.

Dominios de Aplicación. Los autores han trabajado en aplicaciones de diferentes tamaños, partiendo desde aplicaciones pequeñas y simples, que ayudaron a mejorar su enfoque, para acabar proponiendo una metodología de desarrollo para el manejo de aplicaciones con tamaño industrial.

Propiedades de Seguridad. SecureMDD se centra en las propiedades de seguridad habituales, como la integridad de los datos y la confidencialidad. En sentido estricto, SecureMDD no utiliza un lenguaje específico para especificar las propiedades de seguridad, si no que se especifican utilizando un lenguaje lógico llamado *Dynamic Logic*.

Enfoque de Modelado. SecureMDD define un perfil UML que extiende los diagramas de actividad, añadiendo la posibilidad de recoger indicaciones sobre el procesamiento de los mensajes, p.e.: cambio de estado y encriptación. Para ello, los autores definen un metamodelo llamado MEL que contiene las abstracciones necesarias. Este modelo extendido se utiliza para generar, por un lado, modelos de diseño de la aplicación (en la mayoría de los ejemplos modelos Java (Card)) y un modelo formal que plasma la información recogida en las expresiones MEL del modelo UML extendido. Estos son utilizados a su vez para generar código Java (Card) y reglas AST, de forma que KIV, un intérprete para verificación interactiva de propiedades, desarrollado por los autores, puede comprobar las propiedades de seguridad de la aplicación generada.

La Figura 3.16 representa el enfoque general para SecureMDD. El proceso comienza con la creación de un modelo UML para los requisitos funcionales de la aplicación. Los diagramas de clases describen las diferentes entidades de aplicación (*terminales*, *tarjetas*, *usuarios*, *atacantes* y *la infraestructura de comunicación*), mientras que los diagramas de secuencia y actividades modelan el comportamiento del sistema y las interacciones entre las entidades.

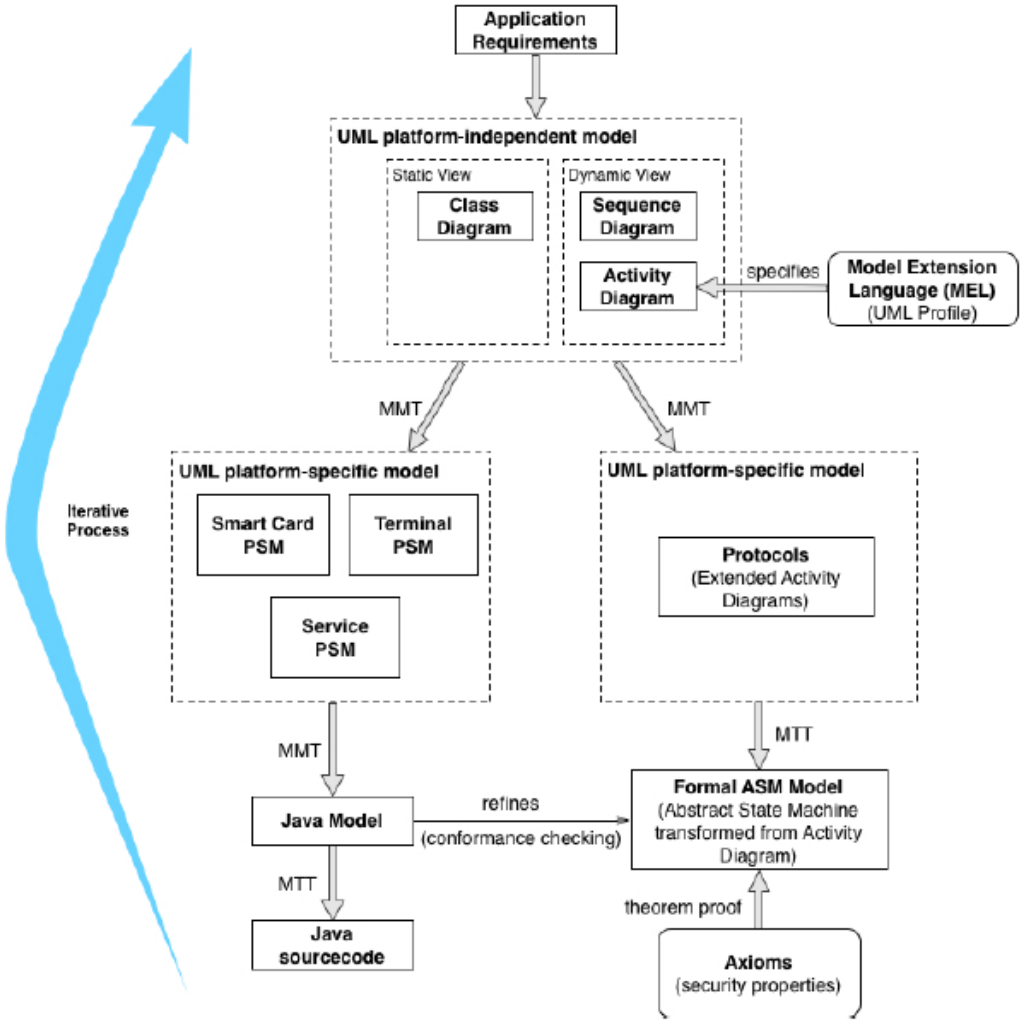


Figura 3.16. Vista General de SecureMDD

Transformaciones de Modelos. Como puede observarse en la Figura 3.16, SecureMDD utiliza transformaciones MTM para transformar los modelos independientes de la plataforma en

dos modelos diferentes: un modelo Java centrado en la ejecución y un modelo de Máquina de Estados Abstracta (ASM) utilizado con fines de verificación, y transformaciones M2T para soportar la generación de código.

Herramienta de Apoyo. Los autores de SecureMDD no describen específicamente una herramienta de apoyo en su enfoque. Sin embargo, varios documentos proporcionan información sobre los artefactos de modelado, las cadenas de transformación y el procedimiento de verificación, en particular (Moebius et al., 2009; Moebius et al., 2012). Si podemos decir que el lenguaje utilizado para codificar las transformaciones M2M es QVT, mientras que Xpand es el utilizado para las transformaciones M2T.

3.4.3 Síntesis de Datos

En esta sección se presentan las principales conclusiones extraídas del estudio de las propuestas más relevantes en el área de la MDS que existen en la actualidad. Para ello, en primer lugar, ofrecemos una vista general de estas propuestas, apoyándonos en la taxonomía definida en la Sección 3.4.1, y a continuación las analizamos en relación con el objetivo principal de esta tesis doctoral: *sistematizar el diseño de arquitecturas de seguridad*.

3.4.3.1 Análisis de las Propuestas MDS existentes

A continuación, resumimos el análisis de las propuestas más relevantes en el área de la MDS, centrándonos en las propiedades que componen la taxonomía definida en la Sección 3.4.1. La Tabla 3-5 resume los resultados del análisis: las columnas corresponden a cada una de las entradas de la taxonomía y las filas a cada uno de los enfoques MDS seleccionados.

Propuesta	Dominios	Propiedades	Enfoque de Modelado		Transformaciones	
			Paradigma	Lenguaje	M2M	M2T
UMLSec	Aplicaciones web, Sistemas Embebidos, Sistemas Distribuidos	Confidencialidad, Integridad, Autenticidad, Autorización, No Repudio	MPM	Perfiles UML	-	Compilador
SecureUML	Aplicaciones web	Control de Acceso	MDA + DSM	Perfiles UML	-	Compilador
SECTET	Gobierno electrónico, Sanidad electrónica, Educación electrónica	Confidencialidad, Integridad, No Repudio, Control de Acceso	MDA + DSM	Perfiles UML	QVT	XPAND
ModelSec	Aplicaciones web, Bases de Datos	Privacidad, Integridad, Autenticación, Disponibilidad, No Repudio, Control de Acceso	MDA + DSM	SecML (nuevo DSL)	RubyTL	MOF-Script
SecureMDD	Tarjetas Inteligentes y Aplicaciones de Servicio	Criptografía (Integridad, Confidencialidad, Secreto)	MDA + DSM	Perfiles UML	QVT	XPAND

Tabla 3-5. Análisis de las Propuestas MDS Existentes

Dominios de Aplicación. La mayoría de las propuestas analizadas están diseñados para la creación de aplicaciones Web, excepto SecureMDD, que se centra en el desarrollo de tarjetas inteligentes y aplicaciones basadas en Servicios. UMLSec por su parte es el enfoque más general.

Propiedades de Seguridad. La mayoría de los enfoques tratan con diferentes propiedades de seguridad, siendo la confidencialidad y la integridad las propiedades más comunes. SecureUML es el único que restringe sus objetivos al control de acceso.

Enfoque de Modelado. La mayoría de las propuestas aplican los principios del paradigma MDA y el DSM, a excepción de UMLsec, que aplica los principios del MPM. Más interesante

resulta el hecho únicamente ModelSec utiliza un lenguaje que no está basado en UML. El resto de propuestas se basan en la definición y uso de perfiles UML.

Transformaciones de Modelos. UMLSec y SecureUML no utilizan transformaciones de modelos y se limitan al uso de compiladores ad-hoc para soportar la generación de código. SECTET y SecureMDD utilizan QVT y Xpand para las transformaciones M2M y M2T respectivamente. Finalmente, ModelSec utiliza RubyTL para las transformaciones M2M y MOF-Script para las transformaciones M2T.

Herramienta de Apoyo. Aunque la mayoría de las propuestas se benefician de los recientes avances en el contexto de la MDE en términos de herramientas de apoyo para la edición de modelos y el desarrollo de transformaciones, el soporte tecnológico ofrecido por estas propuestas es sólo parcial y limitado a ciertas etapas del proceso de desarrollo.

3.4.3.2 Aplicación de las Propuestas MDS existentes al Desarrollo de Arquitecturas de Seguridad

A continuación, analizamos las propuestas MDS más relevantes en relación con el objetivo de esta Tesis Doctoral. Para ello, la Tabla 3-6 resume las principales conclusiones del análisis, que se estructura en torno a las propiedades de la taxonomía utilizada para guiar el estudio de las propuestas. El objetivo de esta sección es comprobar si alguna de las propuestas existentes podría ayudarnos a la hora de construir y documentar arquitecturas de seguridad de grandes sistemas de información.

Dominios de Aplicación. Las arquitecturas de seguridad están compuestas principalmente por tecnologías de seguridad (firewalls, sistemas de detección de intrusos, sistemas de control de acceso, etc.). Ninguna de las propuestas analizadas incluye las arquitecturas de seguridad dentro de sus dominios de aplicación. Este hecho es fundamental a la hora de seleccionar alguna de las propuestas analizadas. Ninguna de las propuestas está centrada en el dominio de aplicación en el que estamos trabajando.

Propuesta	Dominios	Propiedades	Enfoque de Modelado	Uso de Patrones
UMLSec	No	Si	Parcial	No
SecureUML	No	No	Parcial	No
SECTET	No	Si	Parcial	No
ModelSec	No	Si	Si	No
SecureMDD	No	Parcial	Parcial	No

Tabla 3-6. Propuestas MDS para el Desarrollo de Arquitecturas de Seguridad

Propiedades de Seguridad. El principal objetivo de las arquitecturas de seguridad es mantener la confidencialidad e integridad de los activos de información protegidos por la arquitectura. Cualquiera de los enfoques que incluya estas propiedades cumpliría con estos requisitos. En este caso, todos los enfoques, excepto SecureUML, que se limita a ofrecer soluciones para el control de acceso, podrían ser utilizados. Conviene aclarar que SecureMDD aparece como *Parcial*, porque incluye la criptografía como medio para proteger la confidencialidad e integridad de los datos, pero la criptografía es solo uno de los medios para ejecutarlo.

Enfoque de Modelado. Tal y como ya hemos defendido a lo largo de esta Tesis, la separación entre niveles de abstracción propuesta por MDA parece la mejor opción a la hora de desarrollar arquitecturas de seguridad. En este sentido, la mayoría de las propuestas analizadas sería de aplicación puesto que apuestan por esta separación. En cambio, en cuanto al lenguaje de modelado la mayoría de las propuestas apuesta por la utilización de perfiles UML, una opción que se ha mostrado mucho menos conveniente en la práctica (Vara y Marcos, 2012), por lo que en la presente Tesis se ha optado por la utilización de DSLs.

Uso de Patrones. Los patrones proporcionan soluciones probadas y homologadas a la hora de dar soluciones a problemas recurrentes. En este sentido, ninguna de las propuestas analizadas utiliza patrones a la hora de diseñar las soluciones MDS.

3.4.4 Conclusiones

En esta sección se ha realizado un estudio y evaluación de las propuestas más relevantes existentes hasta la fecha en el área de la MDS, con el objetivo de analizar si alguna de ellas podría resultar de utilidad a la hora construir y documentar arquitecturas de seguridad empresariales. Las conclusiones más relevantes extraídas de este análisis son las siguientes:

- ✓ Existe un consenso generalizado en cuanto a la utilidad de los modelos en el desarrollo de sistemas seguros, que ha dado lugar a la aparición de diferentes propuestas que aplican los principios de la MDE en el contexto de la seguridad.
- ✓ En general, cada una de estas propuestas están respaldadas por varias publicaciones en foros internacionales de prestigio, lo que avala la idea de aprovechar el potencial de la MDE para lidiar con los problemas relacionados con la gestión de la seguridad.
- ✓ La mayor parte de las propuestas analizadas adoptan la separación de niveles de abstracción y se basan en el uso de perfiles UML, una mala práctica completamente desaconsejada por la comunidad MDE en los últimos años, que ha comprobado la ineficacia de los perfiles. Por el contrario, la tendencia generalizada a la hora de construir una solución tecnológica basada en los principios de la MDE es la definición de DSLs, acompañados del correspondiente conjunto de herramientas de soporte.
- ✓ Las soluciones proporcionadas en este tipo de enfoques no consideran la utilización de patrones de seguridad a la hora de diseñar sus soluciones. La comunidad científica avala la utilización de estos patrones como herramienta para homogeneizar y documentar las nuevas soluciones de seguridad encontradas. Debido a esto, uno de

los objetivos de esta Tesis es la utilización de patrones en la definición de las arquitecturas de seguridad.

- ✓ Probablemente, la observación más importante resultado de este estudio se refiere al ámbito de aplicación de las propuestas analizadas. Ninguna de ellas fue diseñada para construir y documentar arquitecturas de seguridad empresariales. Podríamos decir que su objetivo es soportar la generación de código que implemente ciertas restricciones o propiedades de seguridad, mientras que el objetivo de esta Tesis es de más alto nivel, pues se centra en la definición de arquitecturas (visión microscópica vs macroscópica: diseño de bajo nivel vs diseño de alto nivel o arquitectónico). En cierto modo, serán los elementos que compondrán la selección de tecnologías que definirá la arquitectura resultado de la aplicación de la propuesta de la presente Tesis Doctoral, los que internamente podrían aplicar las soluciones ofrecidas por las propuestas existentes en el área de la MDS para implementar la funcionalidad que cada uno de ellos proporciona.

Por todo ello, la presente Tesis Doctoral aborda la definición de una nueva propuesta que adopta los principios de la MDS para construir y documentar nuevas arquitecturas de seguridad.

4. Enterprise Security Patterns

4.1 Introducción

Como hemos mostrado en el Capítulo Introductorio, el objetivo principal de esta Tesis Doctoral es desarrollar un meta-modelo que ayude a las organizaciones a diseñar arquitecturas de seguridad, que protejan sus activos de información a la hora de desarrollar nuevos sistemas de información.

Existen muchos problemas recurrentes en la construcción de sistemas de información, y los patrones tratan de capturar una solución homogénea (Alexander et al., 1977). Los patrones de seguridad proporcionan las directrices para apoyar la construcción y evaluación de los mecanismos de seguridad (Fernandez et al., 2008), y su utilización ayuda a incorporar los principios de seguridad en la construcción de sistemas seguros. Sin embargo, como ya hemos mostrado en el estado del arte de esta tesis doctoral, los patrones de seguridad tienen algunas limitaciones:

- ✓ Son pequeñas unidades de defensa (Pelaez et al., 2009). Sólo pueden manejar una (o unas pocas) amenazas.
- ✓ Hay diferentes versiones de un mismo patrón para cada nivel arquitectónico. La construcción de sistemas seguros necesita un amplio conjunto de patrones de seguridad, y este hecho aumenta la complejidad cuando un diseñador está tratando de seleccionar un patrón de seguridad.
- ✓ Varias instancias de un mismo patrón pueden tener aspectos comunes pero el diseñador tiene que encontrarlos. Este hecho puede dar lugar a redundancias de diseño innecesarias.

Debido a estas limitaciones, en (Moral-García et al., 2010; Moral-García et al., 2011) definimos un nuevo enfoque de patrones de seguridad para apoyar el diseño de arquitecturas. Posteriormente, en (Moral-García et al., 2014) acabamos de definir el concepto de *Enterprise Security Pattern*. Ese concepto había sido utilizado anteriormente en (Romanosky, 2003), donde

los autores utilizaron esta expresión para describir e identificar un conjunto de patrones existentes de seguridad centrados en el entorno empresarial. Nosotros adoptamos este nombre, ya que el objetivo de este nuevo patrón es proporcionar una estrategia *top-down* basada en conceptos de Seguridad Dirigida por Modelos (MDS) para la definición de Arquitecturas de Seguridad Empresariales.

Dentro de la estrategia *top-down*, la parte *top* corresponde a las necesidades de un nuevo sistema de información plasmadas en un diagrama de contexto básico. La parte *down* correspondería a la arquitectura de la solución tecnológica necesaria para proteger los activos de información incluidos en el sistema de información. Para ello, ponemos en práctica las ideas de la Arquitectura Dirigida por Modelos (MDA), a través de un proceso en el que los modelos de alto nivel que representan el contexto del sistema van siendo refinados hasta obtener el modelo que representa la Arquitectura de Seguridad Empresarial deseada, de manera que cada fase del proceso representa la definición de un nuevo modelo que añade mayor nivel de detalle al modelo anterior, incidiendo en los aspectos tecnológicos de la solución.

Los Enterprise Security Patterns no están diseñados para sustituir a los patrones de seguridad. Estos patrones utilizan e incorporan los patrones de seguridad en un patrón más amplio para manejar más amenazas y proteger un conjunto de activos de información en un contexto específico. El objetivo principal de estos patrones es proporcionar una solución tecnológica segura y documentada para un contexto dado. Por este motivo, las organizaciones pueden utilizar estos patrones con el fin de seleccionar una estrategia de seguridad global, estandarizando los tipos de arquitecturas de seguridad dentro de la empresa y proporcionando a sus diseñadores un conjunto de directrices de seguridad óptimo y probado.

A continuación, se muestra un breve resumen de las secciones incluidas en este Capítulo:

La **Sección 4.2** muestra una descripción de las **Arquitecturas de Seguridad Empresariales** y los elementos incluidos en estas arquitecturas. El objetivo principal de esta sección es

proporcionar una visión general de los elementos a tener en cuenta a la hora de diseñar este tipo de arquitecturas.

La **Sección 4.3** muestra una **Plantilla para Documentar Enterprise Security Patterns**. El objetivo principal de esta sección es proporcionar la lista de elementos principales que deben ser incluidos en cada nuevo patrón, ayudando a homogeneizar la forma en la que son documentados.

La **Sección 4.4** muestra el **Meta-modelo de los Enterprise Security Patterns** explicando los detalles y relaciones de los elementos incluidos en el patrón. Cada uno de los diagramas muestra un elemento básico de las Arquitecturas de Seguridad Empresariales, y describe como los Enterprise Security Patterns están relacionados con este elemento.

La **Sección 4.5** describe el **Proceso de Modelado** de las **Arquitecturas de Seguridad Empresariales** incluidas en la Solución de los patrones. Para ello, La sección 4.5.1 proporciona una vista general del proceso, mientras que la sección 4.5.2 detalla las decisiones que guían los refinamientos que implica cada paso del proceso de modelado propuesto.

La **Sección 4.6** introduce la arquitectura de un nuevo marco de trabajo o *framework* para facilitar la **Minería de Enterprise Security Patterns**. El objetivo principal de este nuevo *framework* es crear un entorno que ayude a los investigadores a descubrir, documentar y clasificar este tipo de patrones.

4.2 Arquitecturas de Seguridad Empresariales

Un subconjunto de las arquitecturas software son las arquitecturas que proporcionan seguridad para los sistemas de información de una compañía o institución. Conceptualmente en la última década aparece el concepto de Arquitectura de Seguridad Empresarial (*Enterprise Security Architecture*). Al igual que en la construcción convencional de arquitecturas de edificios o en la construcción de arquitecturas software, a la hora de construir este tipo de arquitecturas, los arquitectos de seguridad empresarial deben tener en cuenta una serie elementos y conceptos (Sherwood et al., 2009):

- ✓ El objetivo que se quiere lograr con la construcción de la arquitectura.
- ✓ El entorno en la que la arquitectura va a ser construida y usada.
- ✓ Los riesgos que la arquitectura puede sufrir en ese entorno.
- ✓ Las capacidades técnicas necesitadas para construir y operar la arquitectura.
- ✓ ¿Quién (o Qué) construirá, usará y mantendrá la arquitectura?

Figura 4.1. Elementos de las Arquitecturas de Seguridad Empresariales

La Figura 4.1 muestra los elementos de las Arquitecturas de Seguridad Empresariales asociados con cada una de las consideraciones que los arquitectos o diseñadores de seguridad tienen que tener en cuenta a la hora de construir estas arquitecturas. A continuación, mostramos una descripción de cada uno de los elementos mostrados en la figura anterior.

4.2.1 Activos de Información (*Information Assets*)

Los activos de información pueden ser definidos como elementos de información almacenados en los sistemas de una organización. Cuando nosotros nos referimos a los sistemas de una organización, estamos incluyendo todos los tipos de soportes electrónicos donde puede residir o ser transportada información, es decir, ordenadores portátiles, teléfonos móviles, servidores físicos, servidores virtuales, pasarelas de acceso a Internet, etc.

Dependiendo del tamaño o la industria a la que pertenece una organización la lista y catalogación de activos de información puede variar significativamente. Si el número de activos es muy alto, la complejidad y coste de protegerlos puede incrementar significativamente.

4.2.2 Contexto (*Context*)

Uno de los puntos más importantes a la hora de analizar el entorno de una nueva arquitectura de seguridad es la definición del contexto de negocio. El contexto de negocio es el resultado de los factores internos y externos en los que va a ser construida esa arquitectura. Conociendo el contexto de negocio de una arquitectura es posible analizar los factores críticos del negocio y el riesgo que podría aceptar esa arquitectura.

Las políticas de seguridad son los elementos conceptuales que nos van a ayudar a la hora de incluir la seguridad en un contexto de negocio. Las políticas son directivas de gestión que indican una predeterminada acción, o un camino para gestionar un problema o situación (Wood, 2000). Sin políticas sería mucho más difícil construir sistemas seguros, ya que no sabríamos lo que deberíamos proteger ni cuanto esfuerzo deberíamos incluir en esta protección (Fernandez,

2013). Un sistema utiliza una combinación de políticas de seguridad acorde a su objetivo y su entorno. A la hora de construir sistemas seguros, los arquitectos tienen que considerar diferentes políticas de seguridad, es decir, políticas de confidencialidad, políticas de integridad, políticas de disponibilidad, políticas de auditabilidad, etc.

La arquitectura resultante es una combinación funcional del proceso y la tecnología para lograr el objetivo de la arquitectura dentro del contexto dado. La arquitectura de seguridad debe proporcionar cumplimiento legal y regulatorio para ese contexto (IBM, 2007), además de asegurar que el riesgo aceptado es el mismo que el analizado.

4.2.3 Amenazas (*Threats*)

Las amenazas que pueden ser encontradas en un sistema de información pueden ser de distinta índole, es decir, amenazas que afecten a la confidencialidad, integridad, disponibilidad, etc. A la hora de proteger los activos de información de un sistema, un arquitecto de seguridad debería considerar los métodos y caminos que los grupos de crimen organizado y empleados desleales utilizan para romper o esquivar las defensas de seguridad más comunes.

Dependiendo de la criticidad o sensibilidad de los activos de información a proteger y las amenazas asociadas con esos activos, las medidas de seguridad incluidas en la arquitectura de seguridad empresarial pueden variar significativamente. Por este motivo, es muy importante que las organizaciones tengan una clasificación de activos de información, con el objetivo de facilitar las decisiones de diseño y poder tener un mejor entendimiento de los riesgos que están siendo aceptados.

4.2.4 Tecnologías de Seguridad (*Security Technologies*)

Las tecnologías de seguridad asisten en (i) la protección o mitigación de los efectos de métodos de ataque usados contra activos de información, (ii) la evaluación del daño provocado por los ataques, o (iii) la gestión de la respuesta ante este tipo de ataques.

Una de las características de los sistemas de información es que ellos podrían operar y funciones sin tecnologías de seguridad. Las tecnologías de seguridad son herramientas que proporcionan el nivel de seguridad deseado. Ellos no incluyen o añaden funcionalidad al contexto del negocio, a no ser que el propio negocio sea la seguridad de los sistemas.

4.2.5 Stakeholders y Sistemas

Un *Stakeholder* puede ser definido como un individuo, un equipo, o una organización con un interés o una preocupación en un sistema de información (Lankhorst, 2009). A la hora de construir sistemas seguros, los arquitectos deberían tener en cuenta los intereses y las preocupaciones de los *stakeholders* que construirán, usarán, y mantendrán las tecnologías de seguridad involucradas en la arquitectura de seguridad empresarial.

Considerando que los *stakeholders* a menudo tienen diferente nivel de conocimiento sobre los sistemas a construir, el arquitecto debería ser capaz de explicar las ventajas y las desventajas de las decisiones de diseño asociadas a un nuevo desarrollo. En la mayoría de los casos los stakeholders a tener en cuenta son los siguientes: administrador de sistemas, administrador de seguridad, administrador de logs, el equipo de desarrolladores de seguridad, los usuarios técnicos y los usuarios finales. Cabe destacar que, dependiendo de la metodología utilizada en la construcción de sistemas, esta lista de stakeholders podría variar.

Además, los arquitectos deberían tener en cuenta los sistemas y aplicaciones que interactúan y utilizan las tecnologías de seguridad implantadas. El objetivo de estas consideraciones es evitar problemas de acoplamiento y funcionalidad entre las aplicaciones y las tecnologías de seguridad en etapas posteriores del ciclo de vida de desarrollo (fase de implementación o fase de pruebas).

4.3 Plantilla para Documentar Enterprise Security Patterns

Como hemos comentado en la sección anterior (ver Sección 4.2) las arquitecturas de seguridad empresariales están principalmente formadas por 5 elementos: *activos de información, contexto, amenazas, stakeholders y tecnologías de seguridad*.

Dado que el objetivo principal de los Enterprise Security Patterns es facilitar la construcción de este tipo de arquitecturas, hemos creído conveniente hacer una relación formal entre las arquitecturas de seguridad y los elementos usados para documentar los Enterprise Security Patterns. De esta manera, a la hora de utilizar los Enterprise Security Patterns dentro de una metodología de seguridad, estaremos incorporando los conceptos principales de las arquitecturas de seguridad dentro del proceso de desarrollo.

La Figura 4.2 presenta un meta-modelo UML (*Unified Modeling Language*) que relaciona los elementos definidos para documentar los Enterprise Security Patterns (rectángulo blanco con *) y los elementos de una arquitectura de seguridad empresarial (rectángulo sombreado). Como podemos ver en la Figura 4.2, todos los elementos de nuestro patrón están relacionados con algún elemento de las arquitecturas de seguridad. El objetivo de esta relación es definir la plantilla para documentar los Enterprise Security Patterns incluyendo los principios de seguridad, aspectos, consideraciones y problemas asociados a las arquitecturas de seguridad empresariales. A continuación, mostramos en detalle las relaciones entre los elementos incluidos en el meta-modelo anterior:

- ✓ La intención principal (*Intent*) de los Enterprise Security Patterns es proteger Activos de Información (*Information Assets*) dentro de arquitecturas de seguridad. Estos activos normalmente tienen asociados unas determinadas Amenazas (*Threats*) en un Contexto específico (*Context*).

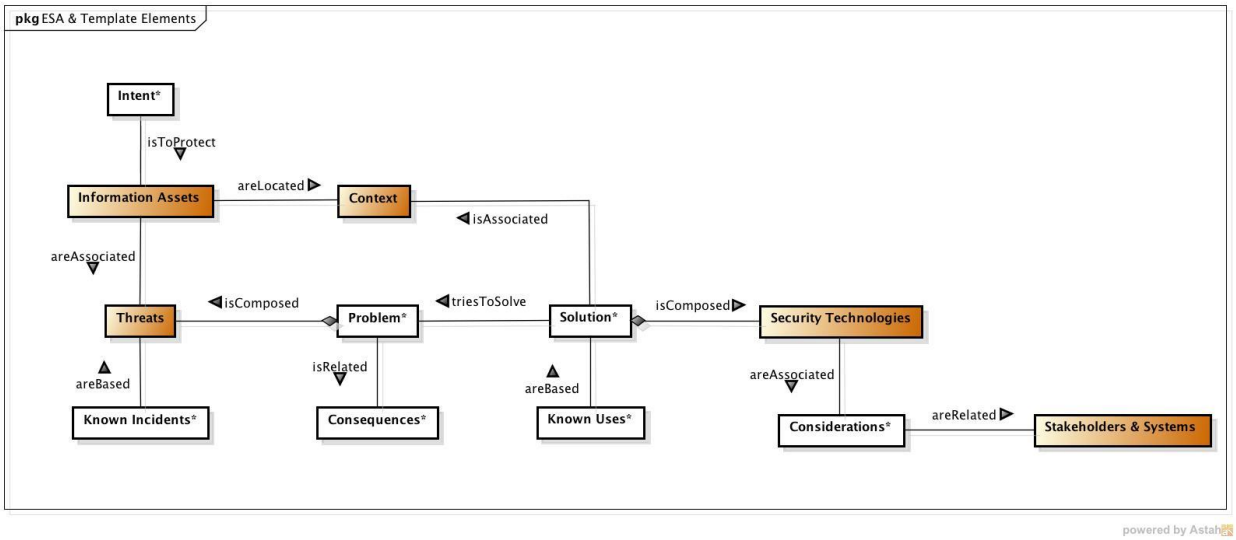


Figura 4.2. Relación entre Elementos de Arquitecturas de Seguridad y Enterprise Security Patterns

- ✓ Las Amenazas (*Threats*) asociadas a los activos de información son descritas en el Problema (*Problem*) del patrón con el objetivo de analizar y mitigar sus Consecuencias (*Consequences*). Casos de Incidentes Conocidos (*Known Incidents*) son también descritos con el objetivo de documentar incidentes reales asociados a esas Amenazas (*Threats*).
- ✓ La solución (*Solution*) de los Enterprise Security Patterns intenta resolver los problemas (*Problem*) asociados a las Amenazas (*Threats*). Una solución (*Solution*) está asociada a un Contexto específico (*Context*) y está compuesta por un conjunto de Tecnologías de Seguridad (*Security Technologies*). Casos de Uso Conocidos (*Known Uses*) son también descritos con el objetivo de documentar casos de éxito donde ha sido desplegado el patrón.
- ✓ Las Consideraciones (*Considerations*) de los Enterprise Security Patterns estan asociadas a las Tecnologias de Seguridad utilizadas (Security Technologies) y tienen en cuenta los Stakeholders y Sistemas involucrados en la arquitectura.

A la hora de seleccionar los elementos de la plantilla para documentar los Enterprise Security Patterns, no solo hemos tenido en cuenta los elementos definidos en las arquitecturas de seguridad empresariales. Esta plantilla principalmente incluye secciones de las plantillas

proporcionadas por (Gamma E., 1995; Buschmann et al., 1996), y algunas nuevas secciones que consideramos necesarias en el diseño de arquitecturas de seguridad, tales como la Intención *(Intent)*, los Incidentes Conocidos (*Known Incidents*) y las Consideraciones *(Considerations)*.

A continuación, se describen todos los elementos incluidos en la plantilla para documentar este nuevo tipo de patrones:

- ✓ **Nombre (*Name*):** El nombre del patrón debe representar el problema que se está tratando de resolver. Este nombre debe ser único dentro del ámbito de este tipo de patrones.
- ✓ **Intención (*Intent*):** En esta sección se ofrece una breve descripción de la finalidad prevista del patrón.
- ✓ **Contexto (*Context*):** Esta sección describe el entorno genérico, bajo el cual, el patrón debería ser aplicado. El contexto debe incluir (i) el tipo de activos de información a proteger (datos, aplicaciones, código fuente y/o configuraciones), (ii) los dominios de seguridad donde el activo está almacenado y va a ser transferido, y (iii) las características principales de quien (clientes, empleados y/o técnicos) o que (sistemas) accederán a los activos. El contexto debe ser especificado utilizando el diagrama de contexto mostrado en la Sección 4.4.2.
- ✓ **Problema (*Problem*):** Esta sección describe la situación que ha llevado a la necesidad de aplicar una serie de mecanismos de seguridad, incluyendo las amenazas que causan la situación y las fuerzas que guían la solución. El problema debe también considerar los activos de información, ya que van a influir en los mecanismos de seguridad incluidos en la solución.
- ✓ **Incidentes Conocidos (*Known Incidents*):** En esta sección se describen casos reales de incidentes de seguridad relacionados con el problema. Estos incidentes se pueden encontrar en sitios web especializados, como (OSF, http://datalossdb.org/), que

recogen este tipo de eventos y especifican cuando ocurrieron, cómo ocurrieron y cuál fue su impacto.

- ✓ **Solución (*Solution*):** En este apartado se describe cómo la arquitectura de seguridad empresarial podría manejar las amenazas asociadas a los activos de información que el patrón pretende proteger. La solución debe ser expresada en cuatro modelos diferentes: el Modelo Independiente de la Computación (CIM), el Modelo Independiente de la Plataforma (PIM), el Modelo Específico de la Plataforma (PSM) y el Modelo Dependiente de Producto (PDM). Cada uno de estos modelos se debe especificar utilizando el modelo de diagrama mostrado en la Sección 4.4.4 Meta-modelo de Tecnologías de Seguridad.
- ✓ **Consideraciones (*Considerations*):** En esta sección se presenta un análisis cualitativo del Modelo Dependiente de Producto presentado en la solución. Este análisis muestra (i) la sobrecarga de rendimiento del patrón, (ii) el costo de su instalación, (iii) la complejidad de su expansión masiva, y (iv) la complejidad de las partes interesadas o *stakeholders* que van a construir, utilizar y mantener las tecnologías implicadas en la solución. Este análisis también debe mostrar si la arquitectura de seguridad necesitaría medidas complementarias para alcanzar su objetivo, es decir, el riesgo residual de la solución.
- ✓ **Consecuencias (*Consequences*):** Esta sección debe mostrar las ventajas e inconvenientes de la solución en relación a las fuerzas que se encuentran en el problema. Las consecuencias también deben discutir qué amenazas son prevenidas, y que amenazas no lo son, a la hora de implementar la solución en un sistema real. La enumeración de las consecuencias debe coincidir con las fuerzas del problema, pero puede haber consecuencias que no correspondan con ninguna fuerza.
- ✓ **Usos conocidos (*Known Uses*):** En esta sección se describen las arquitecturas de seguridad empresariales existentes, donde se ha utilizado la solución aportada en el

patrón. Para soluciones donde el patrón aún no se ha implementado, los contextos específicos en los que el patrón podría desplegarse son suficientes.

- ✓ **Patrones relacionados (*See Also*):** Esta sección ofrece referencias a los Enterprise Security Patterns que resuelven problemas similares, consideran contextos similares, o complementen este patrón.

4.4 Meta-modelo de Enterprise Security Patterns

Como ya hemos descrito en la sección anterior de este capítulo, los Enterprise Security Patterns combinan una amplia gama de elementos a la hora de describir arquitecturas de seguridad empresariales. Para ello, los Enterprise Security Patterns combinan en un patrón cohesivo, todos los elementos incluidos en las arquitecturas de seguridad de la empresa: (i) los activos de información que deben protegerse, (ii) el contexto en el que se encuentran estos activos, (iii) las amenazas asociadas con los activos, (iv) las políticas de seguridad, patrones, mecanismos y tecnologías que se utilizan para detener estas amenazas, y (v) las partes interesadas y los sistemas involucrados en la solución.

A continuación, vamos a explicar los detalles y relaciones de los elementos de los Enterprise Security Patterns, utilizando diagramas UML. Para ello, cada uno de los diagramas mostrará un elemento básico de las arquitecturas de seguridad y describirá como los Enterprise Security Patterns están relacionados con este elemento.

4.4.1 Meta-modelo de Activos de Información

A la hora de construir arquitecturas de seguridad, las organizaciones deben utilizar una clasificación activos de información, con el fin de facilitar el trabajo del ingeniero de seguridad. Los activos de información deben ser clasificados en grupos, de acuerdo con su historial de sensibilidad o criticidad, lo que indica la importancia que dichos activos tienen para la organización y el nivel de protección que se debe aplicar para protegerlos de amenazas y ataques. El proceso de creación de grupos de activos de información ayuda a las organizaciones a desarrollar un inventario de sus activos y describirlos con detalle suficiente para transmitir su valor. Este valor puede depender de varios aspectos o factores. Por esta razón, al clasificar los activos, las organizaciones deben buscar el apoyo de una metodología de análisis de riesgos.

La identificación de los activos y su registro de sensibilidad facilitarán el establecimiento de políticas coste-efectividad a la hora de preservar esos activos. Por ejemplo, el folleto con los nuevos productos de las organizaciones, necesitarán políticas de seguridad relacionadas con su integridad y disponibilidad. Sin embargo, la información de la organización en relación con la compra de un producto de su competidor necesitará políticas de seguridad adicionales relacionadas con su confidencialidad.

Como podemos observar en la Figura 4.3, los activos de información de las organizaciones se pueden clasificar en tres grandes grupos: los datos, las aplicaciones, y el código y las configuraciones. La Tabla 4-1 muestra algunos ejemplos de los activos de información de cada grupo.

Grupos	Activos de Información	
Datos	Clientes	Nombre Número de Cuenta …
	Empleados	Dirección Física Puesto …
	Sistemas	Contraseñas Certificados …
	Información Organizacional	Presupuesto Planes de Negocio …
Aplicaciones	Clientes	Gestion de Elementos Compra
	Empleados	Correo Electrónico Nómina …
Código y Configuraciones	Sistemas	Sistemas Operativos Servidores Web …

Tabla 4-1. Grupos de Activos de Información

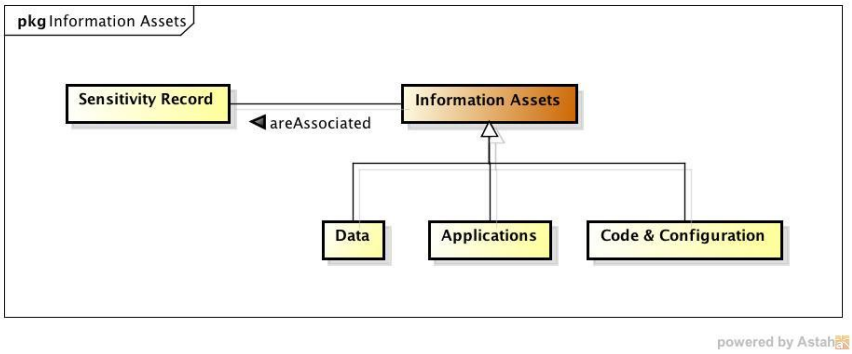


Figura 4.3. Meta-modelo UML de Activos de Información

4.4.2 Meta-modelo de Contexto

Teniendo en cuenta los elementos incluidos en el contexto de las Arquitecturas de Seguridad Empresariales, aquí hemos definido un modelo de dominios de seguridad (*Security Realms*) y un registro de sensibilidad (*Sensitivity Record*) utilizados por los Enterprise Security Patterns. La Figura 4.4 presenta un meta-modelo de UML que incluye estos elementos y las relaciones entre ellos.

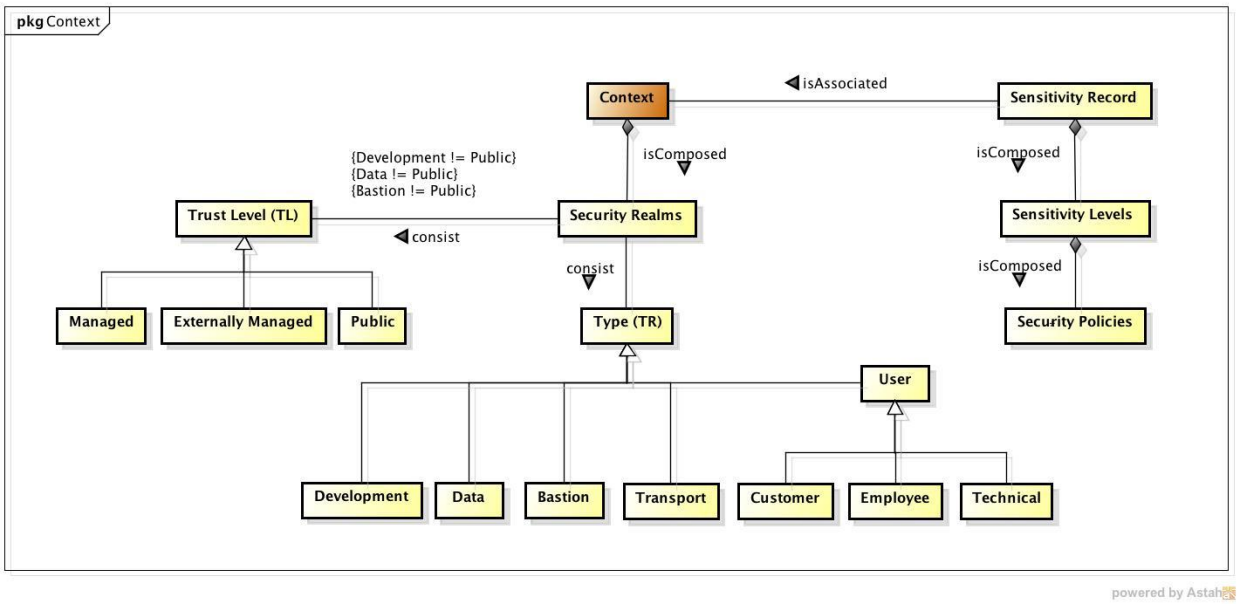


Figura 4.4. Meta-modelo UML del Contexto

4.4.2.1 Modelo de Dominios de Seguridad

Los dominios de seguridad (o *Security Realms*) se pueden definir como entidades lógicas y discretas que dividen una red empresarial. El objetivo principal de estos dominios es estandarizar la seguridad empresarial con el fin de reducir el costo, la demora de los usuarios, y los gastos administrativos de los procedimientos de seguridad redundantes. La principal característica de los dominios de seguridad es que cada uno de ellos tiene las mismas políticas de seguridad en común. Por lo tanto, la red de la empresa puede estar compuesta por un conjunto de dominios de seguridad (o sub-redes) y para cada uno de ellos se pueden definir diferentes políticas de seguridad. Como muestra la Figura 4.4, los dominios de seguridad definidos en esta propuesta están compuestos por un Nivel de Confianza (*Trust Level*) y una Categoría o Tipo Dominio (*Type*).

El Nivel de Confianza (NC) muestra principalmente quien maneja o gestiona el dominio. Como mostramos a continuación en la descripción de los distintos niveles, dependiendo de quién es responsable de la seguridad del dominio, las políticas a aplicar podrían cambiar:

- ✓ **Gestionado (*Managed, M*):** Si el dominio es administrado por el departamento de seguridad de la organización; tenemos la capacidad para diseñar e implementar mecanismos de seguridad dentro del dominio.
- ✓ **Externamente Gestionado (*Externally Managed, EM*):** Si el dominio es administrado por otra organización o entidad; podemos suponer que este reino tiene niveles de seguridad razonables, pero no tenemos la capacidad para inspeccionarlos o auditarlos.
- ✓ **Público (*Public, P*):** Si el dominio no es manejado por cualquier organización; no podemos estar seguros de que este dominio tenga niveles de seguridad adecuados. No tenemos ni la capacidad de diseñar ni implementar mecanismos de seguridad dentro del dominio.

A continuación, detallamos el conjunto de categorías o Tipos de Dominios (TD) que hemos considerado en nuestra propuesta. Estos tipos se basan en la clasificación que se encuentra en (Arconati, 2002):

- ✓ **Cliente (*Customer, C*):** consiste en un cliente o un grupo de clientes con el mismo propósito. Los clientes suelen tener permisos para leer y modificar sus propios datos. La lectura y modificaciones que puedan presentarse en los datos se realizan generalmente a través de aplicaciones específicas. Un ejemplo de este dominio podría ser un cliente accediendo a la página web principal desde un dispositivo móvil.
- ✓ **Empleado (*Employee, E*):** consta de un empleado o grupo de empleados con el mismo propósito. Los empleados tienden a tener permisos para leer y modificar sus datos y los datos de sus clientes. Al igual que los clientes, la lectura y los cambios que se hacen en los datos se realizan generalmente a través de aplicaciones específicas. Un ejemplo de este dominio podría ser el personal de un banco trabajando desde la oficina.
- ✓ **Usuario Técnico (*Technical User, TU*):** consiste en un usuario técnico o grupo de usuarios técnicos con el mismo propósito. Los usuarios técnicos tienden a tener permisos para leer y modificar las aplicaciones, el código y las configuraciones de los sistemas. La lectura y los cambios que se hacen en los activos de información de la organización por lo general se realizan directamente sobre el activo. Un ejemplo de este dominio podría ser un equipo de desarrolladores trabajando dentro de la organización o trabajando desde las oficinas de un tercero.
- ✓ **Desarrollo (*Development, De*):** consiste en un grupo de aplicaciones que están en desarrollo. Estas aplicaciones son accesibles únicamente por los usuarios técnicos. Dentro de este ámbito no hay datos. Un ejemplo de este dominio podría ser un servidor de aplicaciones utilizado para llevar a cabo el desarrollo de un nuevo sistema.
- ✓ **Producción o Datos (*Data, Da*):** consiste en un grupo de aplicaciones y datos que están siendo utilizados por los clientes, los empleados y los usuarios técnicos. Estas

aplicaciones y datos están en continua operación. Un ejemplo de este dominio podría ser el centro de datos de una compañía.

- ✓ **DMZ o Bastión (*Bastion, B*):** consiste en un grupo de tecnologías que se utilizan para separar los dominios públicos de los dominios administrados o gestionados por un tercero. Un ejemplo de este dominio podría ser el portal de acceso a la banca online.
- ✓ **Transporte (*Transport, T*):** consiste en las secciones de la red de la empresa utilizadas para proporcionar conectividad entre dominios. Un ejemplo de este dominio podría ser el sistema de enrutamiento que existen entre el dispositivo de un cliente y el portal de acceso de la banca online.

La clasificación de Dominios de Seguridad (DS) definida en esta propuesta se podría definir como Tipo de Dominio (TD) x Nivel de Confianza (NC). La Tabla 4-2 muestra las posibles relaciones entre los tipos de dominios y los niveles de confianza.

		Nivel de Confianza (NC)		
		Gestionado	Externamente Gestionado	Público
Tipo de Dominio (TD)	Cliente	✓	✓	✓
	Empleado	✓	✓	✓
	Usuario Técnico	✓	✓	✓
	Desarrollo	✓	✓	-
	Datos	✓	✓	-
	Bastión	✓	✓	-
	Transporte	✓	✓	✓

Tabla 4-2. Clasificación de Dominios de Seguridad (DS)

Como se puede observar en la Tabla 4-2, hay tres tipos de dominios (*bastión, desarrollo y datos*) que deben ser gestionados por alguna organización, dado que no hemos encontrado ningún caso en el que este tipo de dominios puedan ser encontrados en el Nivel de Confianza Público. Los dominios de seguridad definidos pueden ajustarse para adaptarse a diferentes tipos

de aplicaciones o entornos empresariales. Lo que importa dentro de este modelo, es la utilización de una clasificación de este tipo que ayude a dividir la red empresarial en distintos dominios.

4.4.2.2 Modelo de Registro de Sensibilidad

Una característica común de todos los activos de información es que tienen que ser almacenados y pueden ser transportados a través de los dominios de seguridad. En términos de seguridad, los niveles de sensibilidad que se deben aplicar a los dominios incluidos en un contexto específico, forman el *registro de sensibilidad o huella* de un activo de información. Por consiguiente, los niveles de sensibilidad pueden ser utilizados para determinar la criticidad de los activos de información.

Dentro del modelo definido, el nivel de sensibilidad o huella se define aplicando un conjunto de políticas de seguridad a cado uno de los dominios incluidos en el contexto. Las políticas de seguridad aplicadas en cada dominio pueden variar, pero tenemos que preservar todos los atributos de seguridad requeridos por los activos (*confidencialidad, integridad, disponibilidad y auditabilidad*) a la hora de manipularlos o transferirlos. A continuación, mostramos un conjunto de políticas de seguridad asociadas a la confidencialidad, que los Enterprise Security Patterns utilizarán a la hora de definir el nivel de sensibilidad de un activo de información. Este conjunto de políticas de seguridad se compone de:

- ✓ **Canal Seguro (CS):** es un modo de transferencia de información que resistente la manipulación o escucha de los mensajes enviados a través del canal.
- ✓ **Canal Claro (CC):** es un modo de transferencia de información que no garantiza la manipulación o escucha de los mensajes enviados a través del canal.
- ✓ **Canal Bloqueado (CB):** en esta política, la transferencia de la información está bloqueada. No es posible enviar información a través del canal.
- ✓ **Almacenamiento Seguro (AS):** es un modo de almacenamiento de información que resiste la manipulación o lectura de los activos almacenados.

- ✓ **Almacenamiento Claro (AC):** Es un modo de almacenamiento de información que no garantiza la manipulación o lectura de los activos almacenados.
- ✓ **Almacenamiento Bloqueado (AB):** en esta política, el almacenamiento de la información está bloqueada. No es posible almacenar información.

A la hora de establecer el registro de sensibilidad o huella de un activo de información en un contexto, los ingenieros de seguridad deben obtener el nivel de sensibilidad de ese activo para todos los dominios de seguridad incluidos en el contexto. Para obtener el nivel de sensibilidad de ese activo en cada dominio de seguridad, los ingenieros de seguridad deben responder a cuatro preguntas relacionadas con los siguientes aspectos de seguridad: *autorización de acceso, encriptación, y autorización de almacenamiento*. Las cuatro preguntas son las siguientes:

1. ¿Puede el Activo de Información A ser transportado a través del Dominio de Seguridad D?
2. Si es así, ¿debería A ser transportado de forma segura?
3. ¿Puede el Activo de Información A almacenarse en el Dominio de Seguridad D?
4. Si es así, ¿debería A ser almacenado de forma segura?

De acuerdo con las respuestas a estas preguntas, las políticas de seguridad serán asignadas, como se muestra en la Tabla 4-3. La primera columna indica el Nivel de Sensibilidad (NS) proporcionado por cada conjunto de políticas de seguridad, donde 1 es el más bajo y 6 el más alto.

Como se puede observar en la Tabla 4-3, la combinación de respuestas nos ha proporcionado siete niveles de sensibilidad diferentes, pero uno de ellos no es aplicable (*la fila sombreada*) porque no es habitual encontrar una política de seguridad donde el activo de información tenga que ser almacenado de forma de segura y pueda ser transportado de manera clara.

		Combinación de Respuestas			
NS	**Políticas de Seguridad**	**1**	**2**	**3**	**4**
4	Canal Seguro (CS) y Almacenamiento Seguro (AS)	*Yes*	*Yes*	*Yes*	*Yes*
3	Canal Seguro (CS) y Almacenamiento Claro (AC)	*Yes*	*Yes*	*Yes*	*No*
5	Canal Seguro (CS) y Almacenamiento Bloqueado (AB)	*Yes*	*Yes*	*No*	-
-	Canal Claro (CC) y Almacenamiento Seguro (AS)	*Yes*	*No*	*Yes*	*Yes*
1	Canal Claro (CC) y Almacenamiento Claro (AC)	*Yes*	*No*	*Yes*	*No*
2	Canal Claro (CC) y Almacenamiento Bloqueado (AB)	*Yes*	*No*	*No*	-
6	Canal Bloqueado (CB)	*No*	-	-	-

Tabla 4-3. Políticas de Seguridad del Registro de Sensibilidad

La salida del registro de sensibilidad o huella es un conjunto de números que representan el nivel de sensibilidad de un activo de información para cada uno de los dominios de seguridad incluidos en el contexto. Este conjunto de números ayudará a los ingenieros de seguridad en el diseño de las soluciones de los Enterprise Security Patterns, indicando las políticas de confidencialidad que se aplicarán a la hora de proteger los activos en cada uno de los dominios. Los Enterprise Security Patterns utilizarán políticas de seguridad adicionales, tales como políticas de integridad, disponibilidad o auditabilidad, en siguientes fases del modelo, aunque el objetivo principal de estos patrones es proteger la confidencialidad de los activos.

Antes de utilizar Enterprise Security Patterns, los ingenieros de seguridad de las organizaciones deben crear perfiles de activos de información y responder a las cuatro preguntas anteriores para cada uno de los dominios incluidos en el contexto. Varias organizaciones podrían aplicar diferentes registros de sensibilidad para el mismo activo. Por ejemplo, a la hora de clasificar los números de cuenta de clientes, una organización de la industria alimentaria podría decidir la aplicación de niveles bajos o medios de sensibilidad en todos sus dominios de seguridad. Sin embargo, una organización bancaria podría decidir la aplicación de niveles altos o

muy altos de sensibilidad. Debido a esto, los Enterprise Security Patterns no tratan de proteger los activos de información individuales. Tienen la intención de proteger los activos de información con el mismo nivel de sensibilidad en un contexto particular.

4.4.3 Meta-modelo de Amenazas

La Figura 4.5 muestra un meta-modelo de UML que muestra las relaciones de las Amenazas (*Threats*). Como podemos ver en la figura, el problema que los Enterprise Security Patterns intenta resolver considera las amenazas asociadas a los activos de información, y las fuerzas que permiten la posibilidad que esas amenazas se hagan materiales. La enumeración de las consecuencias debe coincidir con las fuerzas del problema, pero puede haber consecuencias que no corresponden a ninguna fuerza.

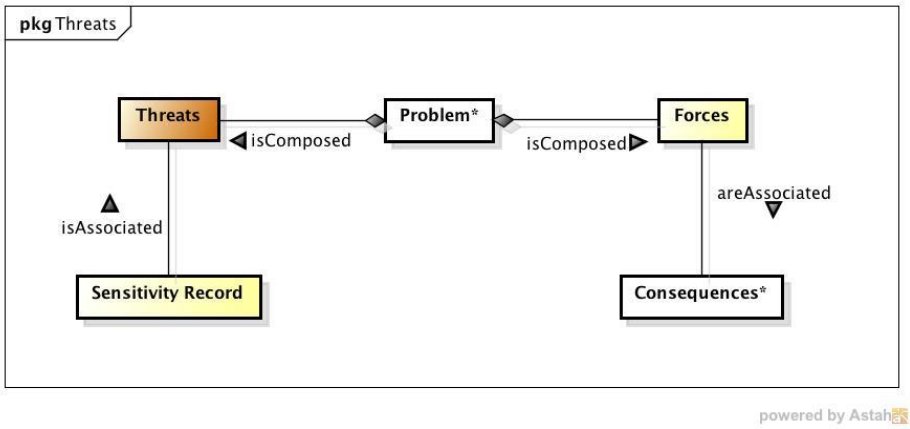


Figura 4.5. Meta-modelo UML de Amenazas

Los diseñadores de seguridad podrían verificar si es necesario proteger un activo de información, utilizando la sensibilidad del registro de los activos de información y las amenazas asociadas a estos activos. Por ejemplo, si un diseñador conociera que un Activo de información (A) es susceptible a un ataque de acceso por Fuerza Bruta (FB) en un Dominio de seguridad (D), FB podría violar la confidencialidad de A usando los canales de acceso al activo. Si el registro de la sensibilidad de A no requiere asegurar los canales de acceso en D, el diseñador no necesita para proteger A de FB. Sin embargo, el diseñador de seguridad debe proteger el activo, si las

amenazas asociadas violan las propiedades de seguridad requeridos por su registro de sensibilidad.

4.4.4 Meta-modelo de Tecnologías de Seguridad

La Arquitectura Dirigida por Modelos (MDA) (Truyen, 2006) es el enfoque definido por el Object Management Group (OMG) para el desarrollo de software en el marco Ingeniería Dirigido por Modelos (MDE). MDA define tres puntos de vista de un sistema: (i) el Modelo Independiente de Computación (CIM), que es utilizado por los analistas de negocios, y se centra en el contexto y los requisitos del sistema sin tener en cuenta su estructura o procesamiento, (ii) el Modelo Independiente de Plataforma (PIM), que es utilizado por los arquitectos y diseñadores de software, y se centra en la capacidad operativa de un sistema fuera del contexto de una plataforma específica, y (iii) el Modelo Específico de la Plataforma (PSM), que es utilizado por los desarrolladores y programadores de software, e incluye detalles relacionados con el sistema para una plataforma específica (Harmon, 2004).

A la hora de describir la solución de los Enterprise Security Patterns, nos hemos basado en el paradigma MDA, adaptando la arquitectura con el contexto de seguridad y al entorno empresarial, donde la parte tecnológica tiene una mayor importancia. Como podemos observar en el meta-modelo de UML para el elemento de la solución (Figura 4.6), hemos definido cuatro puntos de vista, tres de ellos incluidos en el paradigma MDA (CIM, PIM y PSM) y un cuarto punto de vista llamado Modelo Dependiente de Producto (PDM), centrado en el entorno tecnológico, es decir, las tecnologías de seguridad proporcionadas por la solución.

Esta arquitectura no sólo propone un conjunto de modelos que representan el sistema en diferentes niveles de abstracción, sino también un ciclo de vida de desarrollo de software (Meservy y Fenstermacher, 2005) con el que: (i) captura los requisitos en un CIM, (ii) crea uno o más PIM, (iii) transforma el PIM en uno o más PSM, incluyendo reglas específicas de la plataforma y (iv) transforma el PSM en uno o más PDM, incluyendo productos tecnológicos existentes en la industria de la seguridad de información.

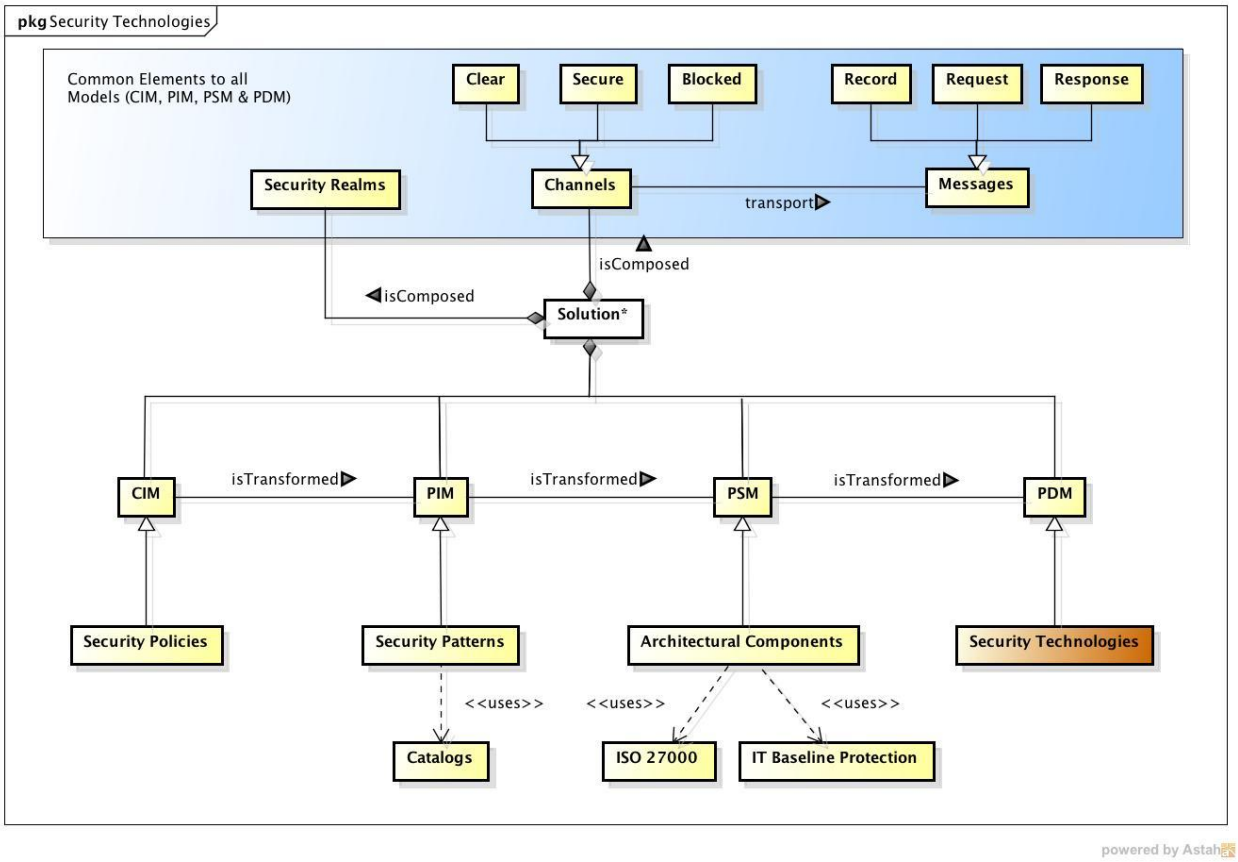


Figura 4.6. Meta-Modelo UML de las Tecnologías de Seguridad

A continuación, se muestra una breve descripción de los cuatro modelos incluidos en la solución de los Enterprise Security Patterns:

Modelo Independiente de la Computación (CIM): Este modelo proporciona una descripción de las políticas de seguridad que el sistema debe cumplir independientemente de sus características funcionales y tecnológicas. Las políticas de seguridad definidas en el registro de la sensibilidad de los activos de información (véase la sección 4.4.2.2) se deben aplicar a los dominios de seguridad incluidos en el contexto. A la hora de construir o diseñar sistemas de seguridad, el CIM podría ayudarnos a definir los requisitos de seguridad de los sistemas a proteger.

Modelo Independiente de la Plataforma (PIM): Este modelo ofrece una descripción conceptual de los mecanismos de seguridad que se deben incorporar en el sistema y las

relaciones que existen entre ellos, con independencia de sus características tecnológicas y el detalle de su implementación. Un mismo CIM podría ser instanciado N veces en este modelo, ya que una política de seguridad puede corresponder a diferentes patrones de seguridad. Una buena guía que puede ser utilizada como base para seleccionar los patrones de seguridad necesarias son las directrices elaboradas en (Schumacher et al., 2006) y (Fernandez, 2013). A la hora de construir o diseñar sistemas de seguridad, el PIM podría ayudarnos a utilizar los Enterprise Security Patterns en las etapas de análisis de las metodologías de seguridad.

Modelo Específico de la Plataforma (PSM): Este modelo define los componentes arquitectónicos incluidos en la arquitectura de seguridad empresarial, independientemente de la tecnología utilizada para resolver el problema. El PSM debería tener en cuenta la forma de organizar los mecanismos de seguridad dentro de la arquitectura. Un mismo PIM podría ser instanciado N veces en este modelo, ya que un mecanismo de seguridad puede ser incluido en diferentes componentes arquitectónicos. Los patrones de seguridad descritos en el PIM se incluyen dentro de los componentes de seguridad arquitectónicos. Dos buenas guías que pueden ser usadas como base para seleccionar los componentes arquitectónicos son la ISO / IEC-27000-series (ISO, http://www.iso.org) y el Manual *IT Baseline Protection* (BSI, 2000). El PSM podría ayudarnos a utilizar los Enterprise Security Patterns en las etapas de diseño de las metodologías de seguridad.

Modelo Dependiente de Producto (PDM): Este modelo define los componentes tecnológicos incluidos en la arquitectura de seguridad. Un mismo PSM podría ser instanciado N veces en este modelo, ya que un mismo componente arquitectónico puede corresponder a diferentes productos tecnológicos. Los productos tecnológicos deben ser productos de buena reputación elaborados por fabricantes conocidos en la industria de la seguridad. La solución final podría variar significativamente dependiendo de las tecnologías utilizadas.

Como también podemos ver en la Figura 4.6, los cuatro modelos de la solución tienen algunos elementos en común. Como hemos mostrado anteriormente en la introducción de este capítulo, la solución de estos patrones trata de resolver un problema en un contexto específico.

Esto significa que los cuatro modelos construyen su solución basándose en el mismo conjunto de dominios de seguridad, y tienen en cuenta el conjunto de amenazas incluido en el problema del patrón. Además, todos los modelos incluidos en la solución utilizan un conjunto de canales de comunicación. Estos canales conectan (i) los usuarios, (ii) los componentes del modelo, y (iii) los activos de información. Cada uno de los canales tiene un emisor y un receptor. Los tipos de canales que podemos encontrar dentro de las soluciones de los Enterprise Security Patterns fueron definidos anteriormente en el meta-modelo del Contexto (ver sección 4.4.2): *Canal claro, Canal seguro y Canal bloqueado*. Con el fin de mostrar una representación lógica del tipo de mensaje que podían transportar, a continuación, hemos definido los tres tipos de mensajes que pueden ser enviados a través de los canales:

- ✓ **Mensaje de Solicitud (*Request Message*):** Un emisor transmite una solicitud a un receptor a través del canal.
- ✓ **Mensaje de Respuesta (Response Message):** Un receptor responde a una petición enviada por un emisor a través del canal.
- ✓ **Mensaje de Registro (Record Message):** Un emisor transmite información relevante a un receptor a través del canal. El receptor debe registrar esta información.

4.4.5 Meta-modelo de Stakeholders y Sistemas

La Figura 4.7 muestra un meta-modelo UML con las relaciones y elementos asociados a los *Stakeholders* y Sistemas. Como podemos ver en esta figura, los Enterprise Security Patterns presentan un análisis cualitativo (o conjunto de consideraciones) del Modelo Dependiente de Producto de la solución. Este conjunto de consideraciones está dividido en dos partes:

1. Un análisis cualitativo de los sistemas o tecnologías implicadas en la solución. Este análisis está relacionado con:

    a. La sobrecarga del sistema en el rendimiento de la solución en relación a almacenamiento, memoria primaria, procesador y ancho de banda.

b. El costo de instalación de la solución.

c. La complejidad de la expansión masiva de la solución.

d. El riesgo residual de la solución, es decir, si se necesitarían medidas complementarias en la arquitectura de seguridad para alcanzar el objetivo.

2. El análisis cualitativo de la complejidad de la solución para los siguientes grupos de *stakeholders*: el administrador de Seguridad, el administrador del sistema de logs, el usuario final y el administrador del sistema.

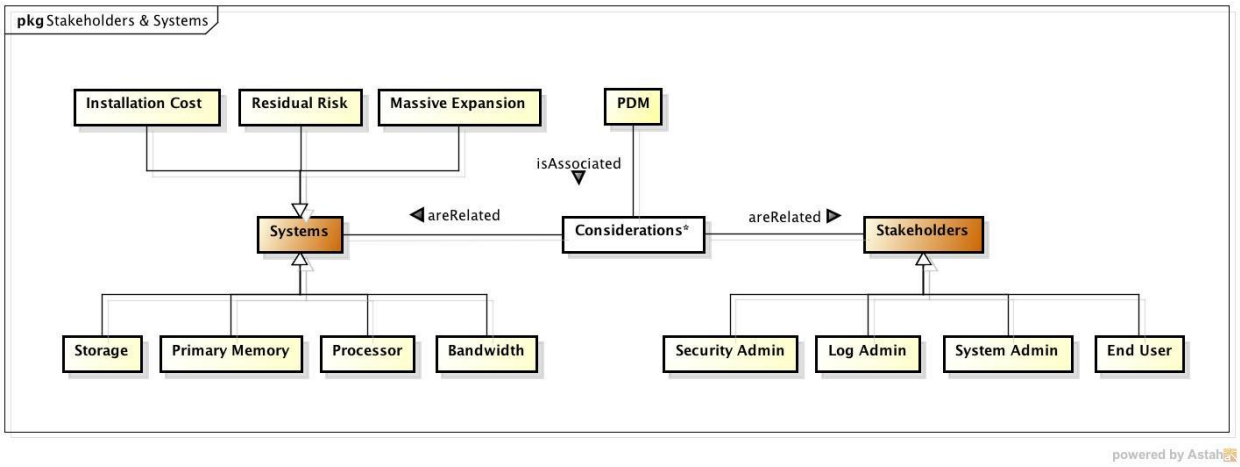


Figura 4.7. Meta-modelo UML de Stakeholders y Sistemas

Al llevar a cabo los análisis, se debe considerar si el despliegue de la solución altera cualitativamente cada uno de los aspectos mencionados anteriormente de forma Nula (0), Baja (1), Medio (2) o Alto (3).

Como hemos visto en esta sección, las consideraciones proporcionados por estos patrones pueden ayudar a la hora de llevar a cabo un análisis de costos, el rendimiento del sistema y la complejidad de la solución tecnológica. Por esta razón, los Enterprise Security Patterns también podrían utilizarse en las etapas iniciales de las metodologías de seguridad como una forma de estimar los costos y esfuerzos.

4.5 Desarrollo de Arquitecturas de Seguridad Empresariales dirigida por Modelos

En esta sección se describe el proceso de modelado propuesto para soportar la definición de arquitecturas de seguridad empresariales basándose en el concepto de *Enterprise Security Pattern* y en los diferentes meta-modelos presentados en las secciones anteriores.

Para ello, la sección 4.5.1 proporciona una vista general del proceso, la sección 4.5.2 muestra un Lenguaje Especifico del Dominio (DSL) para diseñar arquitecturas de seguridad empresariales, y la sección 4.5.3 detalla las decisiones que guían los refinamientos que implica cada paso del proceso de modelado propuesto.

4.5.1 Proceso de modelado

La Figura 4.8 proporciona una vista general del proceso de modelado para la definición de arquitecturas empresariales de seguridad que se propone en la presente tesis doctoral y que se basa en la instanciación de los diferentes meta-modelos presentados en la sección 4.4.

El primer paso consiste en la definición del **Modelo de Contexto**, en el cual se identifican los diferentes Dominios de Seguridad o partes en las que se divide la red que es objeto de estudio, especificando el Nivel de Confianza del Dominio, es decir quién lo gestiona, y el Tipo del Dominio, esto es, si se trata de un usuario (Cliente, Empleado o Técnico) o un Sistema (Bastión, Desarrollo, Datos, Transporte). En este modelo se identifican igualmente los Activos de Información a proteger y las posibles interacciones (por ejemplo, accesos) entre sistemas o usuarios y Activos de Información.

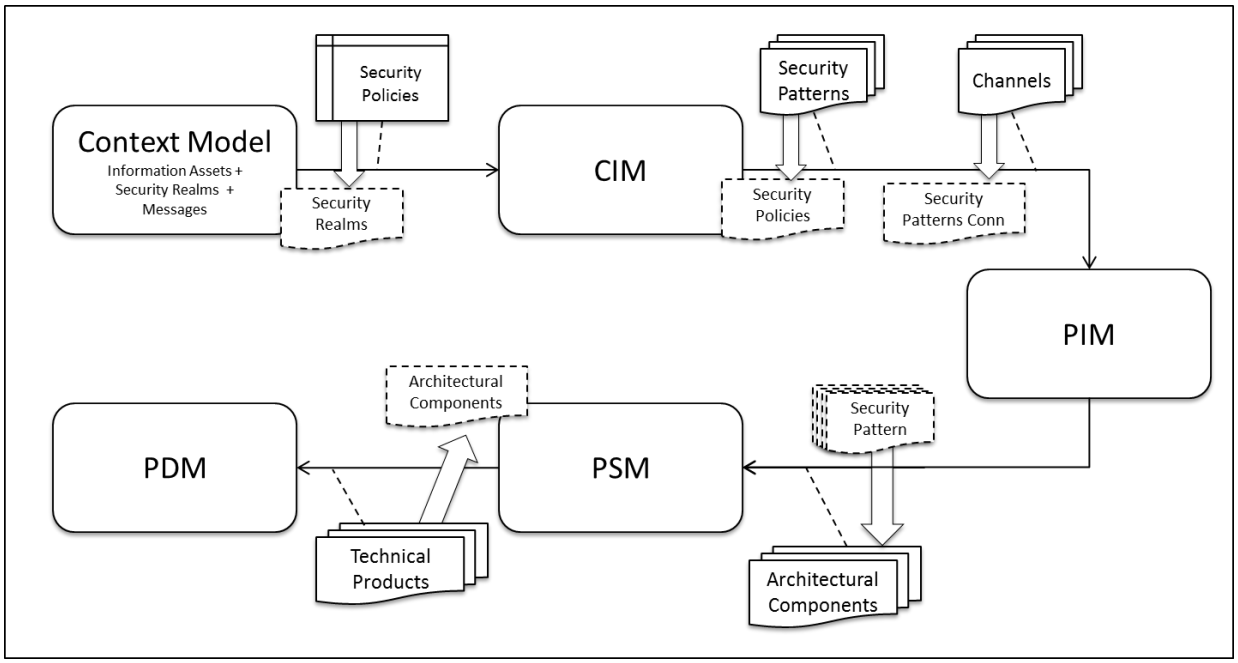


Figura 4.8. Proceso de Modelado de Arquitecturas Empresariales de Seguridad

Una vez que hemos delimitado el ámbito del sistema, el primer paso hacia la solución consiste en obtener los diferentes Registros de Sensibilidad, una dupla de Políticas de Seguridad, que definen el comportamiento que debe regir el transporte y almacenamiento de los Activos de Información en el dominio. Cada dupla se corresponde con un Nivel de Sensibilidad. Por tanto, la identificación de las Políticas de Seguridad que aplican al Dominio, permite asignarle un Nivel de Sensibilidad. El resultado de asignar las Políticas de Seguridad, identificando los Niveles de Sensibilidad de cada Dominio incluido en el Modelo de contexto da lugar al **CIM** (**Computer Independent Model**). Nótese que, de acuerdo a la terminología utilizada por la OMG, hemos dado en denominar CIM a este primer modelo de la solución, pues no es más que una evolución del modelo de contexto en el que, abstrayéndonos por completo de detalles tecnológicos, incluimos los requisitos de seguridad que debe cumplir el sistema (las Políticas de Seguridad que deben aplicar a cada dominio).

A continuación, utilizamos las guías para la identificación de patrones definidas por (Schumacher et al., 2006) y (Fernandez, 2013), para identificar qué Patrones de Seguridad corresponden a cada Política de Seguridad. La idea subyacente es asegurar que cada intercambio

de mensajes o trasiego de información, esté asegurado por la aplicación de uno o varios patrones de seguridad. El resultado de aplicar los patrones seleccionados sobre el CIM es el **PIM** (**Platform Independent Model**), que proporciona una descripción conceptual de los mecanismos de seguridad que debemos incorporar en el sistema, abstrayéndose por completo de detalles tecnológicos y aspectos de implementación. Conviene resaltar que la correspondencia entre Políticas y Patrones de Seguridad no es única, por lo que a partir de un mismo CIM podemos obtener diferentes PIM en función de la selección de patrones que hagamos.

El siguiente paso en el proceso propuesto consiste en encapsular los diferentes Patrones de Seguridad incluidos en el PIM en Componentes Arquitectónicos. Como resultado obtenemos el **PSM** (**Platform Specific Model**) que detalla aún más la solución que estamos diseñando, sin llegar al nivel de detalle propio de la implementación, ya que estos Componentes Arquitectónicos son de naturaleza genérica y admiten diferentes implementaciones. De la misma forma que un CIM podía dar lugar a diferentes PIM en función de las decisiones de diseño escogidas (correspondencias entre Políticas y Patrones de Seguridad), un PIM puede dar lugar a diferentes PSMs en función de qué Componentes Arquitectónicos se utilicen para encapsular los diferentes patrones. Tanto la ISO 27000 (ISO, http://www.iso.org), como la IT Baseline Protection Manual (BSI, 2000), podrían utilizarse para guiar este proceso de selección.

El último conjunto de decisiones de diseño consiste en realizar una selección de productos comerciales que implementen cada Componente Arquitectónico, obteniendo así el **PDM (Product Dependent Model),** un modelo completo y detallado de la solución arquitectónica diseñada, lista para ser traducida directamente a una solución tecnológica. Nótese nuevamente que este PDM no es único, cada PSM puede dar lugar a diferentes PDMs en función de la selección de productos comerciales que realicemos.

4.5.2 Lenguaje Específico del Dominio (DSL)

En esta sección, se muestra un Lenguaje Especifico del Dominio o DSL (*Domain Specific Language*) que ha sido desarrollado para diseñar Arquitecturas de Seguridad Empresariales.

Para ello, hemos tenido en cuenta todos los elementos de diseño incluidos en el meta-modelo de los Enterprise Security Patterns:

- ✓ Activos de Información
- ✓ Dominios de Seguridad
- ✓ Políticas de Seguridad
- ✓ Canales de Comunicación y los Tipos de Mensajes.
- ✓ Componentes Específicos del Modelo:
  - *Modelo PIM* >> Patrones de Seguridad
  - *Modelo PSM* >> Elementos Arquitectónicos
  - *Modelo PSM* >> Productos Tecnológicos.

4.5.2.1 Activos de Información

Los elementos incluidos en el meta-modelo de Activos de Información mostrado en la sección anterior son: (i) los Datos, (ii) las Aplicaciones y (iii) el Código y las Configuraciones. La Tabla 4-4 muestra la representación gráfica de estos elementos dentro del Lenguaje Específico del Dominio definido.

	Elementos	Iconos
Activos de Información	*Datos*	
	Aplicaciones	
	Código and Configuraciones	

Tabla 4-4. Elementos DSL. Activos de Información

4.5.2.2 Dominios de Seguridad

Los elementos incluidos en el meta-modelo del Contexto asociados a los Dominios de Seguridad son: (i) Los Tipos de Dominios de Seguridad y (ii) su Nivel de Confianza.

La Tabla 4-5 muestra la representación gráfica de los tipos de Dominios de Seguridad dentro del Lenguaje Específico del Dominio definido.

	Elementos	Iconos
Dominios de Seguridad	*Cliente*	
	Empleado	
	Usuario Técnico	
	Datos (Producción)	
	Desarrollo	
	Bastión (DMZ)	
	Transporte	

Tabla 4-5. Elementos DSL. Tipos de Dominios de Seguridad

La Tabla 4-6 muestra la representación gráfica de los Niveles de Confianza de los Dominios de Seguridad dentro del Lenguaje Específico del Dominio definido.

Como ya se ha explicado anteriormente, cada uno de los dominios de seguridad estará compuesto por un tipo de dominio y por un nivel de confianza.

	Elementos	Iconos
Nivel de Confianza	*Gestionado*	
	Externamente Gestionado	
	Publico	

Tabla 4-6. Elementos DSL. Niveles de Confianza de los Dominios de Seguridad

4.5.2.3 Políticas de Seguridad

Dentro del meta-modelo del Contexto mostrado en la sección anterior, también se han definido las Políticas de Seguridad que serán aplicadas dentro del modelo CIM.

La Tabla 4-7 muestra la representación gráfica de las Políticas de Seguridad que serán utilizadas dentro del Lenguaje Específico del Dominio definido. La columna NS muestra el Nivel de Sensibilidad asociado a cada una de las políticas.

NS	Políticas de Seguridad	Iconos
4	Canal Seguro (CS) y Almacenamiento Seguro (AS)	
3	Canal Seguro (CS) y Almacenamiento Claro (AC)	
5	Canal Seguro (CS) y Almacenamiento Bloqueado (AB)	
1	Canal Claro (CC) y Almacenamiento Claro (AC)	
2	Canal Claro (CC) y Almacenamiento Bloqueado (AB)	
6	Canal Bloqueado (CB)	

Tabla 4-7. Elementos DSL. Políticas de Seguridad

4.5.2.4 Canales de Comunicación y Tipos de Mensajes

Los elementos comunes incluidos en el meta-modelo de Tecnologías de Seguridad mostrado en la sección anterior son los siguientes: (i) Los Canales de Comunicación y (ii) los Tipos de Mensajes que pueden ser enviados a través de esos canales.

La Tabla 4-8 muestra la representación gráfica de los Tipos de Canales de Comunicación que pueden ser usados dentro del Lenguaje Específico del Dominio definido.

	Elementos	Iconos
Canal de Comunicación	*Claro*	
	Seguro	
	Bloqueado	

Tabla 4-8. Elementos DSL. Canales de Comunicación

La Tabla 4-9 muestra la representación gráfica de los Tipos de Mensajes que pueden ser usados dentro del Lenguaje Específico del Dominio definido.

	Elementos	Iconos
Mensajes	*Solicitud o Respuesta*	
	Solicitud y Respuesta	
	Registro	

Tabla 4-9. Elementos DSL. Tipos de Mensajes

Cada una de las flechas incluidas en los Diagramas de Arquitectura representará el tipo de Canal de Comunicación y el tipo de Mensaje que están enviando. Como ya hemos mostrado anteriormente, los Canales de Comunicación Bloqueados no tendrán capacidad para enviar mensajes.

4.5.2.5 Componentes Específicos del Modelo

Los elementos específicos incluidos en el meta-modelo de Tecnologías de Seguridad mostrado en la sección anterior son los siguientes: (i) Patrones de Seguridad (PIM), (ii) elementos Arquitectónicos (PSM) y (iii) Productos Tecnológicos (PDM).

La Tabla 4-10 muestra la representación gráfica de los componentes específicos del modelo usados dentro del Lenguaje Específico del Dominio definido.

	Elementos	Iconos
Componentes Específicos del Modelo	*(PIM) Patrones de Seguridad*	
	(PSM) Elementos Arquitectónicos	
	(PDM) Productos Tecnológicos	

Tabla 4-10. Elementos DSL. Componentes Específicos del Modelo

4.5.3 Transiciones entre modelos

En esta sección se muestran las decisiones que guían los refinamientos que implican cada paso del proceso de modelado propuesto en la sección anterior.

A continuación, se detallan las guías de modelado asociadas a las transiciones entre (i) el Modelo de Contexto y el CIM, (ii) el CIM y el PIM, y (iii) el PIM y el PSM.

4.5.3.1 Del Contexto al CIM

La Tabla 4-11 proporciona una descripción de las decisiones de diseño asociadas a las transiciones entre los modelos de Contexto y los CIM. Un Identificador (ID) ha sido asignado a cada decisión con el objetivo de facilitar la aplicación de cada una de las decisiones.

ID	Decisiones de Diseño
CON-CIM-1	Entre un Dominio Público (NO Gestionado) y un Dominio Gestionado siempre hay una red Bastión (DMZ).
CON-CIM-2	Siempre que la información a proteger este fuera de la organización (Dominio Público o Gestionado Externamente), la Autenticación se realiza dentro de los sistemas de la Organización.

Tabla 4-11. Decisiones de Diseño entre el Modelo de Contexto y el CIM

4.5.3.1.1 Decisión de Diseño CON-CIM-1

La Figura 4.9 proporciona un ejemplo para la aplicación de la decisión de diseño CON-CIM-1. La parte izquierda de la imagen muestra un contexto en el cual un empleado (*dominio P-E*) accede desde una red pública (*dominio P-T*) a su correo corporativo, el cual está alojado en una empresa externa (*dominio EM-Da*). La primera flecha (1) simula la solicitud inicial del empleado para acceder al correo electrónico. La segunda flecha (2) simula el envío del formulario para proporcionar las credenciales de acceso al correo. La tercera flecha (3) simula el envío de las credenciales por parte del empleado. Así la empresa externa puede validar si el empleado es quien dice ser.

La parte derecha de la imagen muestra la transición entre el Modelo de Contexto y el CIM, aplicando la decisión de diseño CON-CIM-1. Se puede observar como aparece un dominio Bastión (*EM-B*) entre el empleado y el centro de datos de la empresa externa.

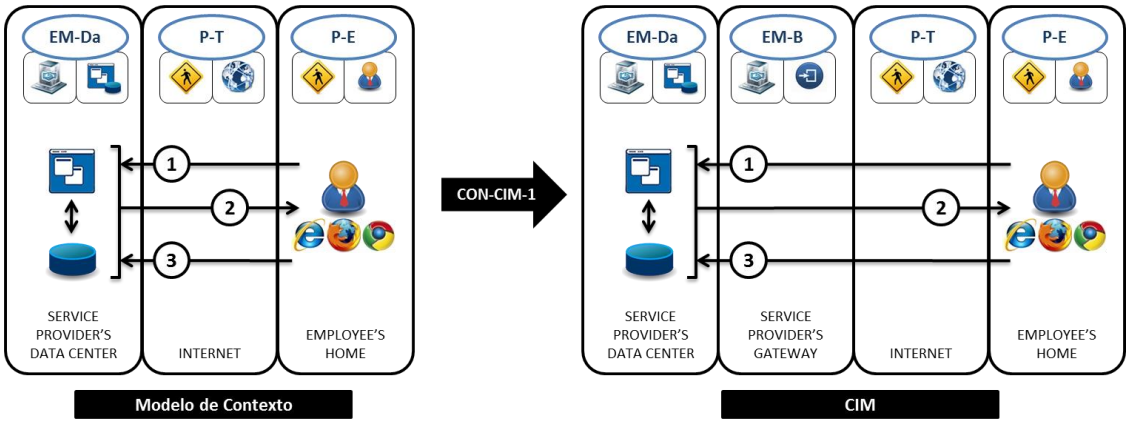


Figura 4.9. Aplicación de la Decisión de Diseño CON-CIM-1

4.5.3.1.2 Decisión de Diseño CON-CIM-2

El diagrama CIM mostrado en la imagen anterior no es el diagrama CIM final asociado al modelo de contexto, dado que, en este caso particular, también aplica la decisión de diseño CON-CIM-2 (*Siempre que la información a proteger este fuera de la organización, la Autenticación se realiza dentro de los sistemas de la Organización*).

La Figura 4.10 proporciona un ejemplo para la aplicación de la decisión de diseño CON-CIM-1 y CON-CIM-2, conjuntamente.

Como se puede comprobar en la figura anterior, además de aparecer un dominio Bastión en la empresa externa (*dominio EM-B*), el nuevo CIM incluye un dominio de Datos gestionado (*dominio M-Da*) y un dominio Bastión gestionado (*dominio M-B*).

A la hora de aplicar las decisiones de diseño se han modificado algunos de los mensajes intercambiados. La primera flecha (1) y la segunda flecha (2) simulan la solicitud inicial del empleado para acceder al correo electrónico y el envío del formulario para proporcionar las credenciales de acceso al correo, al igual que lo hacían en el Modelo del Contexto. La tercera flecha (3) simula el envío de las credenciales por parte del empleado hacia los sistemas de autenticación alojados dentro de la organización. La cuarta flecha (4) simula el envío de las credenciales de acceso desde la organización hasta la empresa externa. En este caso, la

organización valida que el empleado es quien dice ser y la empresa externa otorga el acceso al empleado validado.

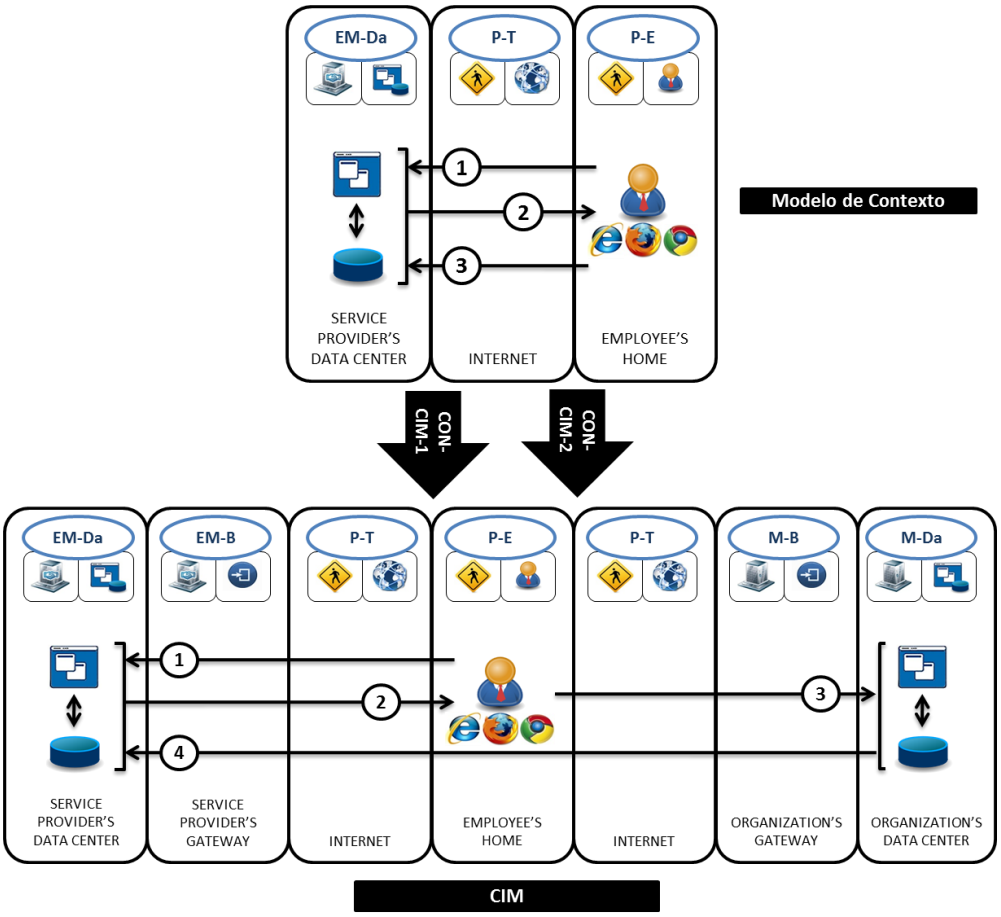


Figura 4.10. Aplicación de las Decisiones de Diseño CON-CIM-1 y CON-CIM-2

4.5.3.2 Del CIM al PIM

La Tabla 4-12 proporciona una descripción de las decisiones de diseño asociadas a las transiciones entre los modelos CIM y los PIM. Un Identificador (ID) ha sido asignado a cada decisión con el objetivo de facilitar la aplicación de cada una de las decisiones.

ID	Decisiones de Diseño
CIM-PIM-1	En el Destino de las Comunicaciones Seguras debe aparecer un patrón *Secure Channel* en el último dominio con sensibilidad 5, 4, o 3 en la dirección del mensaje.
CIM-PIM-2	Todo acceso desde un dominio público (No Gestionado) hacia la red Bastión implica un proceso de Autorización (tripleta de *Identificación*, *Autenticación* y *Control de Acceso*).
CIM-PIM-3	Los patrones de *Autenticación* siempre deben estar en dominios Gestionados que no sean el dominio de Bastión, Transporte o Usuarios.
CIM-PIM-4	Si la información no puede quedar alojada en el Dominio del Usuario es necesario incluir un *Motor de Virtualización* y la tripleta de Autorización (*Identificación, Autenticación y Control de Acceso*) dentro del Dominio de Datos de la organización.
CIM-PIM-5	Todo acceso a la información y/o aplicación que aloja la información necesita un proceso de Autorización (tripleta de *Identificación, Autenticación* y *Control de Acceso*).
CIM-PIM-6	Todo Dominio Gestionado o externamente Gestionado debe tener un sistema de Registro de Actividad de TODOS los elementos de seguridad.

Tabla 4-12. Decisiones de Diseño entre el CIM y el PIM

La Figura 4.11 proporciona un diagrama CIM que nos va a ayudar a explicar la aplicación de cada una de las decisiones de diseño mostradas en la tabla anterior. Este diagrama CIM está basado en un modelo de Contexto donde un empleado accede desde una empresa externa (*dominio EM-TU*) a datos productivos de una organización (*dominio M-Da*) pasando por un dominio público (*dominio P-T*) y la red bastión de la organización (*dominio M-B*).

El registro de sensibilidad proporcionado en el diagrama nos indica que los datos deben viajar cifrados fuera del dominio de datos de la organización (*dominio M-Da*) y no pueden quedar almacenados en ningún dominio que no sea el dominio de datos (*dominio M-Da*).

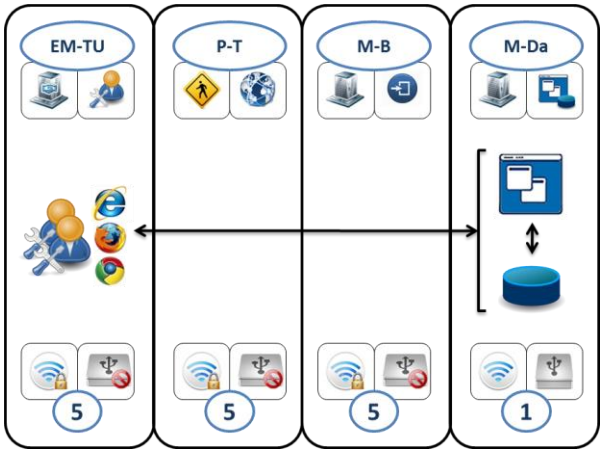


Figura 4.11. Diagrama CIM para explicación de las transiciones entre CIM y PIM

4.5.3.2.1 Decisión de Diseño CIM-PIM-1

La Figura 4.12 proporciona un ejemplo para la aplicación de la decisión de diseño CIM-PIM-1: *en el Destino de las Comunicaciones Seguras debe aparecer un patrón Secure Channel en el último dominio con sensibilidad 5, 4, o 3 en la dirección del mensaje*.

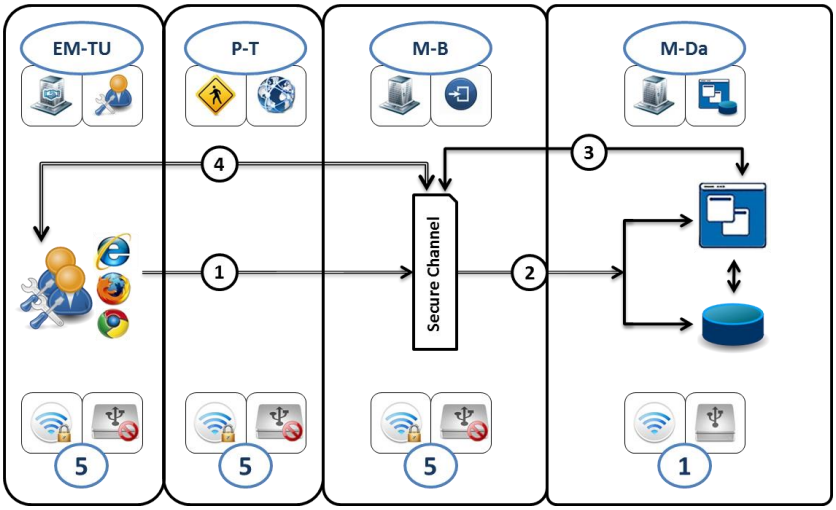


Figura 4.12. Aplicación de la Decisión de Diseño CIM-PIM-1

A continuación, se muestra la descripción de los mensajes intercambiados:

- ✓ La primera flecha (1) simula la solicitud de acceso al dato de forma cifrada.

- ✓ La segunda flecha (2) simula la solicitud de acceso al dato en claro. En este momento el patrón *Secure Channel* ya ha descifrado el canal de comunicación.
- ✓ La tercera flecha (3) simula el envío de información sin cifrar. Esta información llega al patrón *Secure Channel* para ser cifrada antes de su envío fuera de los dominios de la organización.
- ✓ La cuarta flecha (4) simula el envío de la información mediante un canal cifrado. El navegador del empleado tiene la capacidad de descifrar la información.

4.5.3.2.2 Decisiones de Diseño CIM-PIM-2 y CIM-PIM-3

Siguiendo el ejemplo anterior, la Figura 4.13 proporciona un ejemplo para la aplicación de la decisión de diseño CIM-PIM-2: *Todo acceso desde un dominio público (No Gestionado) hacia la red Bastión implica un proceso de Autorización (tripleta de Identificación, Autenticación y Control de Acceso)*; y CIM-PIM-3: *Los patrones de Autenticación siempre deben estar en dominios Gestionados que no sean el dominio de Bastión, Transporte o Usuarios.*

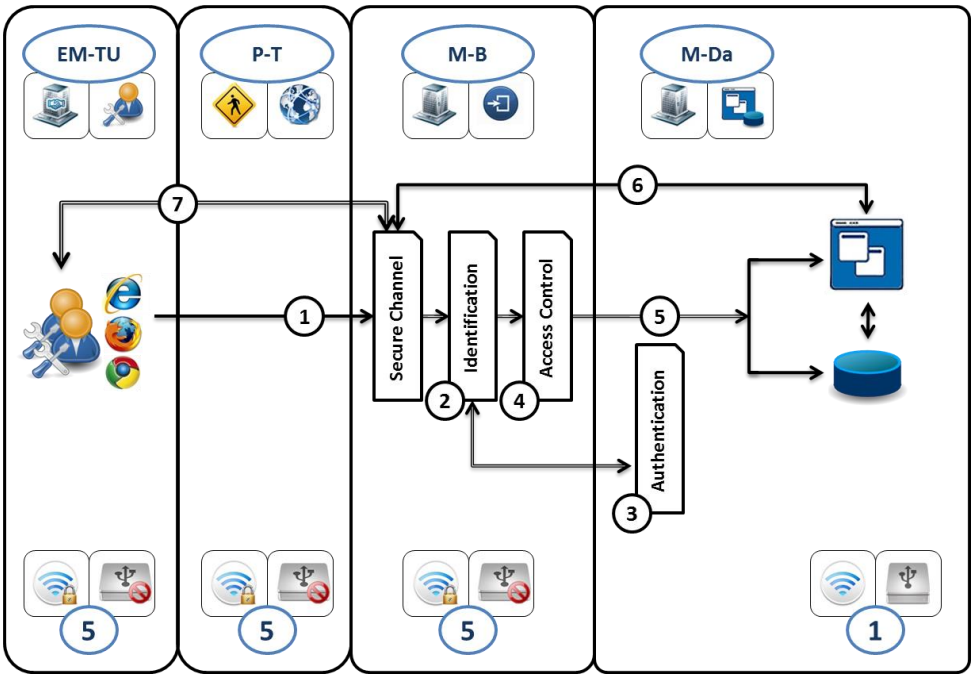


Figura 4.13. Aplicación de las Decisiones de Diseño CIM-PIM-2 y CIM-PIM-3

A continuación, se muestra la descripción de los mensajes intercambiados:

- ✓ La primera flecha (1) simula la solicitud de acceso al dato de forma cifrada.
- ✓ La segunda (2), tercera (3) y cuarta (4) flecha simulan la Identificación, Autenticación y Control de Acceso del empleado a la red interna de la organización. En este momento el patrón *Secure Channel* ya ha descifrado el canal de comunicación.
- ✓ La quinta flecha (5) simula la solicitud de acceso al dato en claro.
- ✓ La sexta flecha (6) simula el envío de información sin cifrar. Esta información llega al patrón *Secure Channel* para ser cifrada antes de su envío fuera de los dominios de la organización.
- ✓ La cuarta flecha (7) simula el envío de la información mediante un canal cifrado. El navegador del empleado tiene la capacidad de descifrar la información.

4.5.3.2.3 Decisión de Diseño CIM-PIM-4

Siguiendo el ejemplo anterior, la Figura 4.14 proporciona un ejemplo para la aplicación de la decisión de diseño CIM-PIM-4: *Si la información no puede quedar alojada en el Dominio del Usuario es necesario incluir un Motor de Virtualización y la tripleta de Autorización (Identificación, Autenticación y Control de Acceso) dentro del Dominio de Datos de la organización*.

A continuación, se muestra la descripción de los mensajes intercambiados:

- ✓ La primera flecha (1) simula la solicitud de acceso al dato de forma cifrada.
- ✓ La segunda (2), tercera (3) y cuarta (4) flecha simulan la Identificación, Autenticación y Control de Acceso del empleado a la red interna de la organización. En este momento el patrón *Secure Channel* ya ha descifrado el canal de comunicación.
- ✓ La quinta (5), sexta (6) y séptima (7) flecha simulan la Identificación, Autenticación y Control de Acceso del empleado en el motor de virtualización.
- ✓ La octava flecha (8) simula la solicitud de acceso al dato en claro.

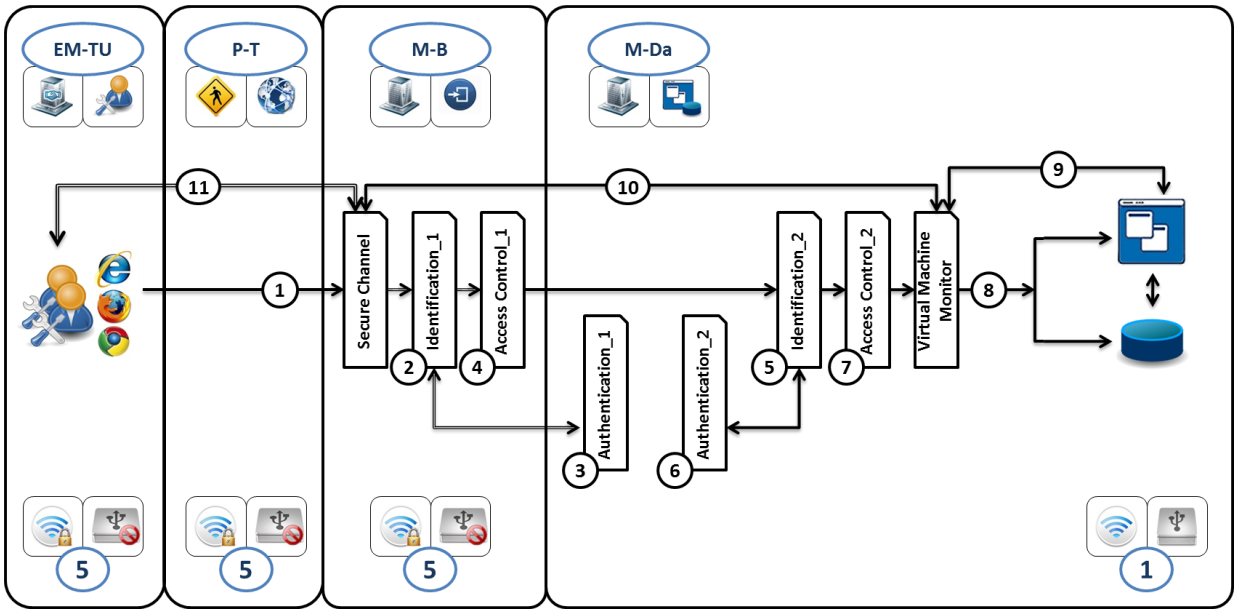


Figura 4.14. Aplicación de la Decisión de Diseño CIM-PIM-4

- ✓ La novena flecha (9) simula el envío de información sin cifrar. Esta información llega al patrón *Virtual Machine Monitor* para ser virtualizada antes de su envío fuera de los dominios de la organización.
- ✓ La décima flecha (10) simula el envío de la información virtualizada. Esta información llega al patrón *Secure Channel* para ser cifrada antes de su envío fuera de los dominios de la organización. En este momento, la información esta virtualizada y cifrada.
- ✓ La undécima flecha (11) simula el envío de la información mediante un canal cifrado. El navegador del empleado tiene la capacidad de descifrar la información. El empleado no será capaz de almacenar esta información en local debido a la virtualización.

4.5.3.2.4 Decisión de Diseño CIM-PIM-5

Siguiendo el ejemplo anterior, la Figura 4.15 proporciona un ejemplo para la aplicación de la decisión de diseño CIM-PIM-5: *Todo acceso a la información y/o aplicación que aloja la*

información necesita un proceso de Autorización (tripleta de Identificación, Autenticación y Control de Acceso).

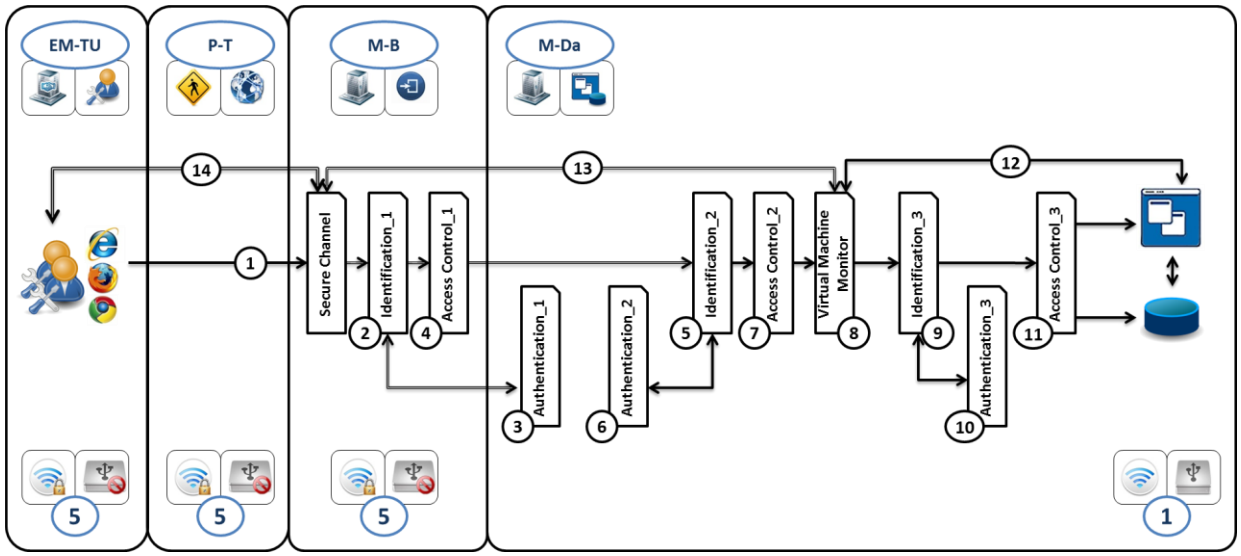


Figura 4.15. Aplicación de la Decisión de Diseño CIM-PIM-5

- ✓ La primera flecha (1) simula la solicitud de acceso al dato de forma cifrada.
- ✓ La segunda (2), tercera (3) y cuarta (4) flecha simulan la Identificación, Autenticación y Control de Acceso del empleado a la red interna de la organización. En este momento el patrón *Secure Channel* ya ha descifrado el canal de comunicación.
- ✓ La quinta (5), sexta (6) y séptima (7) flecha simulan la Identificación, Autenticación y Control de Acceso del empleado en el motor de virtualización.
- ✓ La octava flecha (8) simula la solicitud de acceso al dato en claro.
- ✓ La novena (9), decima (10) y undécima (11) flecha simulan la Identificación, Autenticación y Control de Acceso del empleado en la información y/o aplicación solicitada.
- ✓ La duodécima flecha (12) simula el envío de información sin cifrar. Esta información llega al patrón *Virtual Machine Monitor* para ser virtualizada antes de su envío fuera de los dominios de la organización.

- ✓ La decimotercera flecha (13) simula el envío de la información virtualizada. Esta información llega al patrón *Secure Channel* para ser cifrada antes de su envío fuera de los dominios de la organización. En este momento, la información esta virtualizada y cifrada.
- ✓ La decimocuarta flecha (14) simula el envío de la información mediante un canal cifrado. El navegador del empleado tiene la capacidad de descifrar la información. El empleado no será capaz de almacenar esta información en local debido a la virtualización.

4.5.3.2.5 Decisión de Diseño CIM-PIM-6

Siguiendo el ejemplo anterior, la Figura 4.16 proporciona un ejemplo para la aplicación de la decisión de diseño CIM-PIM-6: *Todo Dominio Gestionado o externamente Gestionado debe tener un sistema de Registro de Actividad de TODOS los elementos de seguridad*.

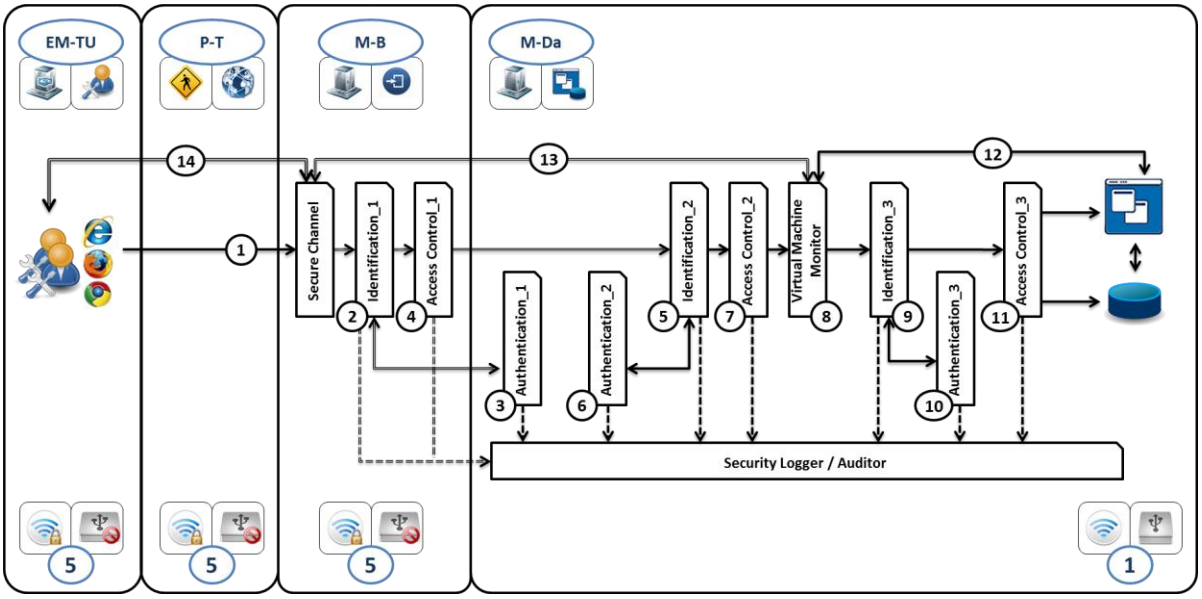


Figura 4.16. Aplicación de la Decisión de Diseño CIM-PIM-6

En este caso los mensajes intercambiados son exactamente iguales a la figura anterior. A cambio se puede observar que todos los patrones están interconectados con el patrón *Security Logger / Auditor*.

4.5.3.3 Del PIM al PSM

La Tabla 4-13 proporciona una descripción de las decisiones de diseño asociadas a las transiciones entre los modelos PIM y los PSM. Un Identificador (ID) ha sido asignado a cada decisión con el objetivo de facilitar la aplicación de cada una de las decisiones.

ID	Decisiones de Diseño
PIM-PSM-1	Un elemento arquitectónico nunca puede pertenecer a dos dominios distintos.
PIM-PSM-2	Un patrón *Security Logger* se corresponde con un *Sistema de Registro de Logs*.
PIM-PSM-3	Si encontramos en el dominio Bastión, (i) Un patrón de *Identificación*, (ii) Un patrón *Secure Channel* + un patrón de *Identificación*, o (iii) un patrón de *Secure Channel* + un patrón de *Identificación* + un patrón *Control de Acceso*, la transformación PSM podría ser un *Web Server*, un *Proxy Inverso* o una *VPN*. Si el dato no puede quedar almacenado en el dominio desde el que está accediendo el usuario, la transformación debe ser una VPN.
PIM-PSM-4	Si encontramos un patrón *Identificación* + un patrón *Control de Acceso* + un patrón de Máquina Virtual, la transformación PSM se corresponde con un Sistema de Virtualización.
PIM-PSM-5	Si encontramos *los datos* junto alguno de los elementos de la tripleta de Autorización (patrón *Identificación*, patrón *Control de Acceso* y/o patrón de Autenticación), la transformación PSM se corresponde con un Servidor de Datos. Si encontramos los datos junto a un patrón de Almacenamiento Seguro, la transformación PSM corresponde con un Servidor de Datos Desasociados.
PIM-PSM-6	Si encontramos *las aplicaciones* junto alguno de los elementos de la tripleta de Autorización (patrón *Identificación*, patrón *Control de Acceso* y/o patrón de Autenticación), la transformación PSM se corresponde con un Servidor de Aplicaciones.
PIM-PSM-7	Un patrón de *Autenticación* se corresponde con (i) un *Servidor de Usuario y Contraseña*, (ii) un *Servidor de Doble Factor*, (iii) un *Dispositivo de Biométricos*, o (iv) un conjunto de los tres anteriores. La transformación va a depender de la robustez que se quiera ofrecer en el proceso de autenticación.

Tabla 4-13. Decisiones de Diseño entre el PIM y el PSM

Con el objetivo de explicar la aplicación de cada una de las decisiones de diseño mostradas en la Tabla 4-13, vamos a utilizar el ejemplo mostrado en la sección anterior partiendo desde la Figura 4.16. Dado que la decisión de diseño PIM-PSM-1: *Un elemento arquitectónico nunca puede pertenecer a dos dominios distintos*, no tiene aplicación (*es una restricción*), empezaremos aplicando la segunda decisión de diseño.

4.5.3.3.1 Decisión de Diseño PIM-PSM-2

La Figura 4.17 proporciona un ejemplo para la aplicación de la decisión de diseño PIM-PSM-2: *Un patrón Security Logger se corresponde con un Sistema de Registro de Logs*.

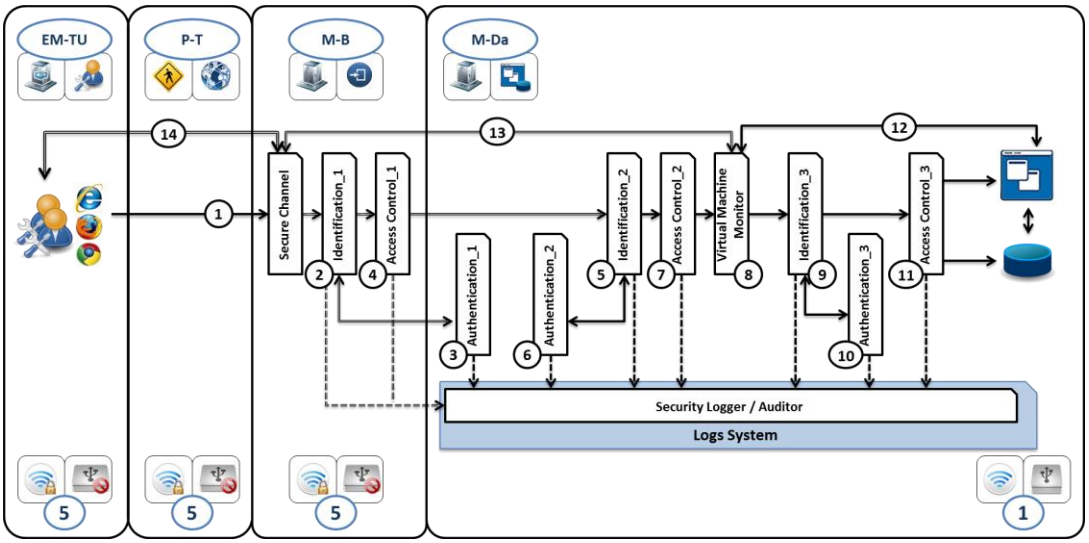


Figura 4.17. Aplicación de la Decisión de Diseño PIM-PSM-2

4.5.3.3.2 Decisión de Diseño PIM-PSM-3

La Figura 4.18 proporciona un ejemplo para la aplicación de la decisión de diseño PIM-PSM-3: *Si encontramos en el dominio Bastión, (i) Un patrón Secure Channel + un patrón de Identificación, o (ii) un patrón de Secure Channel + un patrón de Identificación + un patrón Control de Acceso, la transformación PSM podría ser un Proxy Inverso o una VPN. Si el dato no puede quedar almacenado en el dominio desde el que está accediendo el usuario, la transformación debe ser una VPN.*

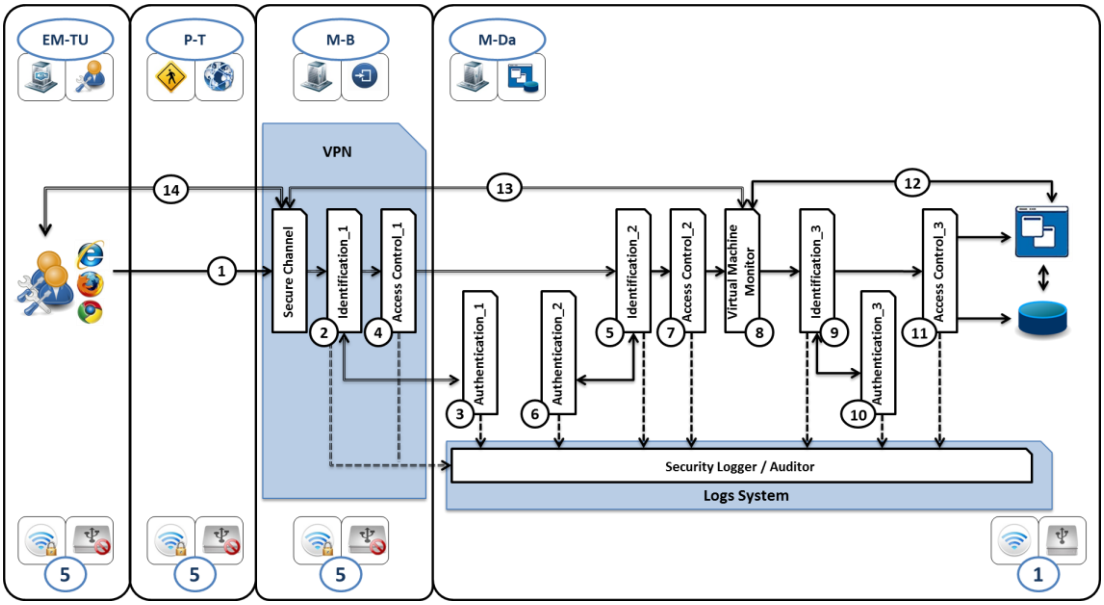


Figura 4.18. Aplicación de la Decisión de Diseño PIM-PSM-3

En este caso en particular el dato transferido no puede quedar almacenado en el dominio del usuario, por lo tanto, la transformación PSM se corresponde con una *Virtual Private Network* (VPN).

4.5.3.3.3 Decisión de Diseño PIM-PSM-4

La Figura 4.19 proporciona un ejemplo para la aplicación de la decisión de diseño PIM-PSM-4: *Si encontramos un patrón Identificación + un patrón Control de Acceso + un patrón de Máquina Virtual, la transformación PSM se corresponde con un Sistema de Virtualización*.

4.5.3.3.4 Decisiones de Diseño PIM-PSM-5 y PIM-PSM-6

La Figura 4.20 proporciona un ejemplo para la aplicación de la decisión de diseño PIM-PSM-5: *Si encontramos los datos junto a la tripleta de Autorización (patrón Identificación + patrón Control de Acceso + patrón de Autenticación), la transformación PSM se corresponde con un Servidor de Datos*, y PIM-PSM-6: *Si encontramos las aplicaciones junto a la tripleta de Autorización (patrón Identificación + patrón Control de Acceso + patrón de Autenticación), la transformación PSM se corresponde con un Servidor de Aplicaciones*.

En este caso en particular, tanto los datos como las aplicaciones se encuentran junto a la tripleta de Autorización. Por este motivo, la transformación PSM se corresponde con un *Servidor de Datos y Aplicaciones*.

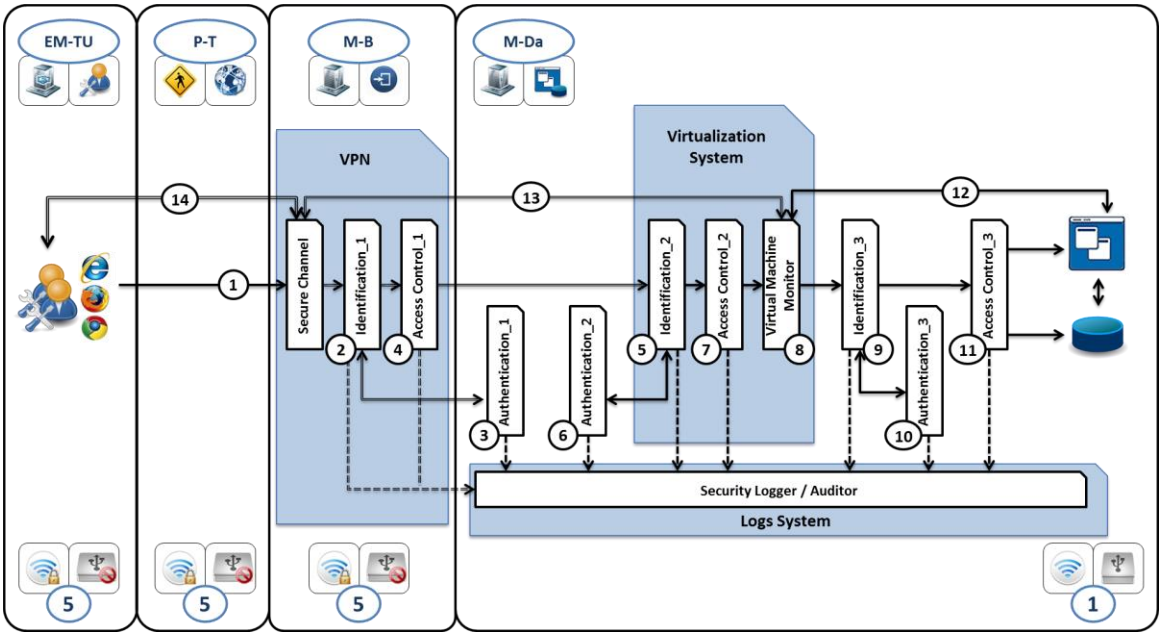


Figura 4.19. Aplicación de la Decisión de Diseño PIM-PSM-4

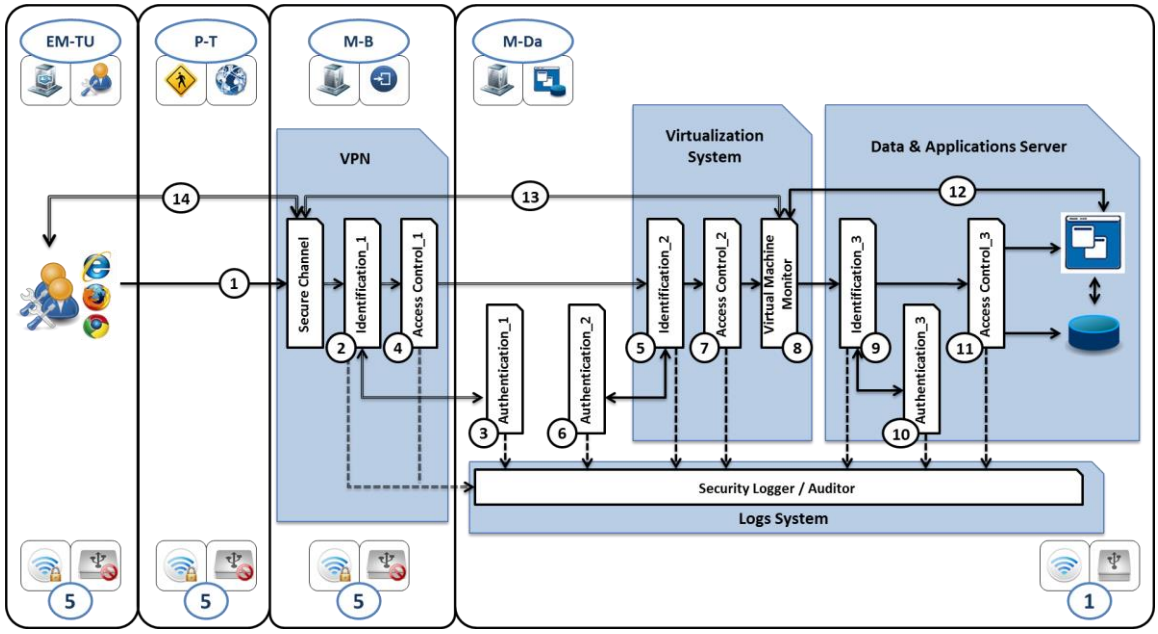


Figura 4.20. Aplicación de las Decisiones de Diseño PIM-PSM-5 y PIM-PSM-6

4.5.3.3.5 Decisión de Diseño PIM-PSM-7

La Figura 4.21 proporciona un ejemplo para la aplicación de la decisión de diseño PIM-PSM-7: *Un patrón de Autenticación se corresponde con (i) un Servidor de Usuario y Contraseña, (ii) un Servidor de Doble Factor, (iii) un Dispositivo de Biométricos, o (iv) un conjunto de los tres anteriores.*

En este caso en particular, el primer patrón de *Autenticación* ha sido transformado en un *Servidor de Doble Factor* (más robusto) y el segundo patrón de *Autenticación* ha sido transformado en un *Servidor de Usuario y Contraseña* (menos robusto).

Como ya se ha comentado anteriormente, la transformación PSM va a depender de la robustez que se quiera ofrecer en el proceso de Autenticación. Para este ejemplo, se ha decidido dar una mayor robustez al proceso de Autorización asociado al acceso desde una red externa, y una menor robustez al proceso de Autorización asociado al acceso del Sistema de Virtualización. La decisión de diseño dependerá del diseñador o arquitecto.

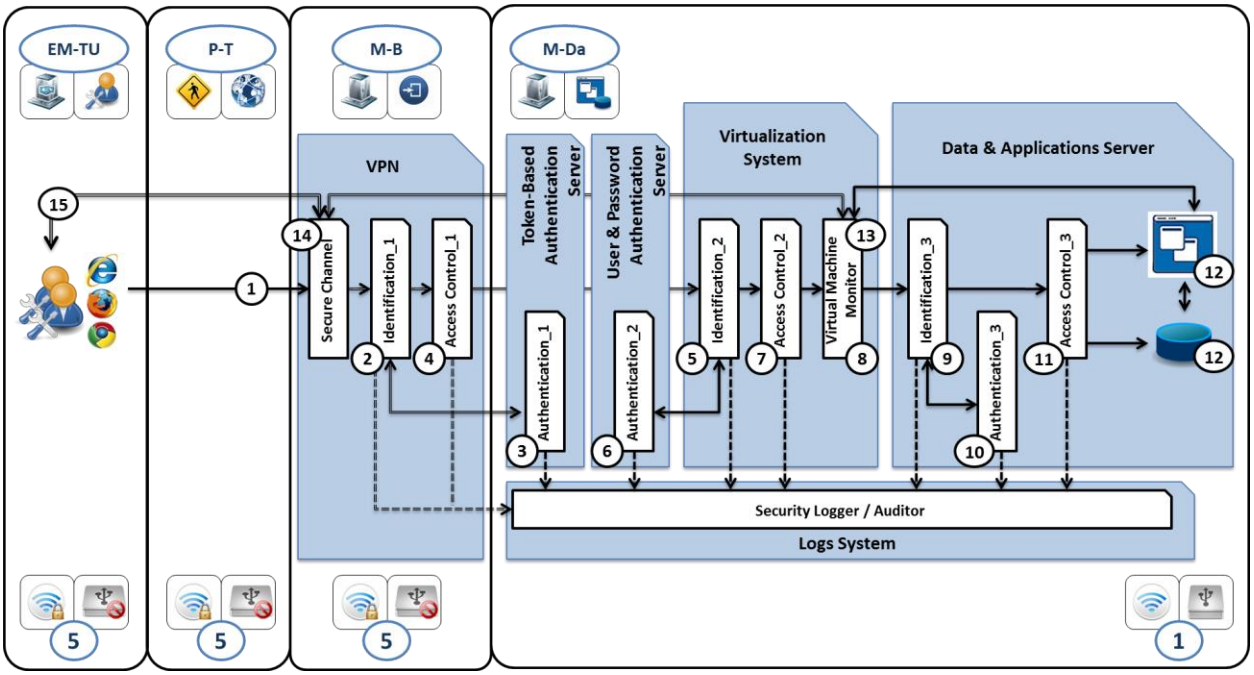


Figura 4.21. Aplicación de la Decisión de Diseño PIM-PSM-7

4.6 Minería de Enterprise Security Patterns

Una de las principales conclusiones extraídas de la revisión sistemática de Minería de Patrones de Seguridad realizada en la Sección 3.3, es que los estudios analizados no cumplen los requisitos básicos a la hora de descubrir y documentar patrones de seguridad orientados en el diseño de arquitecturas o sistemas seguros.

En esta sección hemos definido un proceso para facilitar la minería de Enterprise Security Patterns. El objetivo principal de este nuevo proceso es crear un entorno que ayude a los investigadores a descubrir y documentar nuevos patrones. La Figura 4.22 muestra a la derecha los elementos principales del patrón incluidos en el proceso de minería y a la izquierda los 4 pasos definidos dentro de la minería de Enterprise Security Patterns.

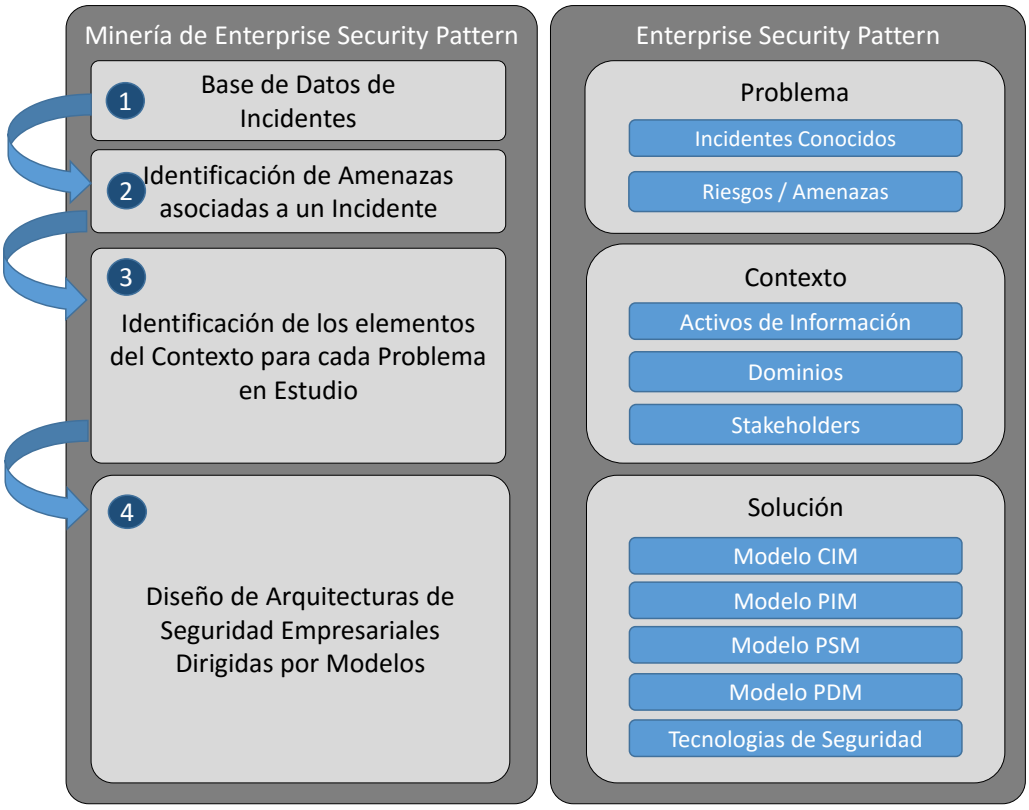


Figura 4.22. Proceso de Minería de Enterprise Security Patterns

Como se puede observar en la Figura 4.22 los elementos principales de los Enterprise Security Patterns elegidos para facilitar la minería de estos patrones son la tripleta *Problema*, *Contexto* y *Solución*. Esto es debido a que el objetivo principal de este tipo de patrones es solucionar problemas concretos en contextos concretos.

Para facilitar la minería de Enterprise Security Patterns hemos definido 4 pasos que pueden ayudar a los investigadores a encontrar nuevos patrones:

1. Base de Datos de Incidentes (BDI) para seleccionar nuevos problemas a solucionar.
2. Identificación de Riesgos o Amenazas asociadas a un Incidente.
3. Identificación de los elementos del Contexto para cada Problema en estudio.
4. Diseño de la Arquitectura de Seguridad Empresarial que soluciona el problema en estudio dentro del contexto dado.

A continuación, mostramos una descripción de cada uno de los elementos incluidos en el proceso de minería.

4.6.1 Base de Datos de Incidentes (BDI)

Como ya habíamos mostrado en la Introducción de esta tesis doctoral, uno de los principales objetivos de la metodología Casandra (Rubio, 2016) es ayudar a las organizaciones a saber de qué tienen que protegerse. Para ello, Casandra (i) describe los distintos tipos de incidentes, (ii) crea un Algebra de Venn para clasificar los incidentes y (iii) define los elementos incluidos en la Base de Datos de Incidentes (BDI).

4.6.1.1 Tipos de Incidentes

Si atendemos al origen intencional de los incidentes (y por tanto de las amenazas incluidas en los Enterprise Security Patterns) podemos clasificar los incidentes en dos grandes grupos. Por un lado, tenemos los incidentes que se materializan por accidente (*accidentales*), y por el otro

lado, los incidentes que se instancian sólo por la acción deliberada de una persona o personas (intencionales).

Dado que un incidente es la materialización de una amenaza y que una amenaza es un incidente en potencia, sólo podemos tener datos de frecuencia e impacto de las amenazas que al menos se hayan materializado una vez. De una amenaza que nunca se haya instanciado no podríamos conocer sus efectos reales. Dada esta relación entre amenazas e incidentes, en la clasificación desarrollada en la metodología Casandra se describen realmente los incidentes, quedando las amenazas tipificadas en función de los incidentes en los que se pueden materializar.

Siguiendo esta descripción y atendiendo a la capacidad de disponer de actuarios de incidentes podemos clasificarlos en dos grandes grupos:

1. **Accidentales o Incidentes sujetos a comportamiento estadístico:** Es posible tipificarlos y obtener una frecuencia de su ocurrencia y un impacto medio. No hay nadie interesado en que suceda un incidente en particular. Este grupo debe gestionarse utilizando Metodologías de Análisis de Riesgo basadas en frecuencia e impacto de los incidentes.

2. **Intencionales o Incidentes no sujetos a comportamiento estadístico:** Por su propia característica no están sujetos a comportamiento estadístico, ya que su ocurrencia está basada en el riesgo/beneficio del atacante más que en las medidas de protección propias. La propuesta que hace la Metodología Casandra es que se gestionen utilizando la Teoría de Juegos para modelarlos.

Dada la naturaleza de los incidentes y el riesgo implícito en la intencionalidad de los incidentes, dentro de la minería de Enterprise Security Patterns vamos a priorizar la definición y clasificación de patrones que solucionen problemas o amenazas intencionales, aunque también serán atendidos aquellos problemas que tengan índole accidental.

4.6.1.2 Algebra de Venn de Incidentes (AVI)

Dentro de la Metodología Casandra se define un modelo de agrupación de los incidentes en conjuntos y subconjuntos que se denomina Algebra de Venn de Incidentes (AVI).

Cada incidente de seguridad debe ser ubicado dentro de una serie de subconjuntos (públicos y privados) Cuanto más adentro esté dentro de la estructura AVI mayor probabilidad de ocurrencia tendrá y mayor será nuestra priorización, en la definición de un nuevo Enterprise Security Pattern que ayude a las organizaciones a protegerse de ese problema o amenaza.

4.6.1.2.1 Conjuntos Públicos

Dentro de los conjuntos públicos van a existir varios tipos de subconjuntos (ver Figura 4.23 y Figura 4.24). La calidad de la información que tenga un AVI concreta vendrá determinada por el tiempo y el esfuerzo que se haya dedicado a su construcción. Un AVI que tenga un histórico de cinco años de incidentes para un sector industrial concreto y especializado en áreas geográficas específicas se confiere como una herramienta que va a permitir fijar con mucha precisión la estrategia de seguridad de una empresa de ese sector en ese entorno geográfico.

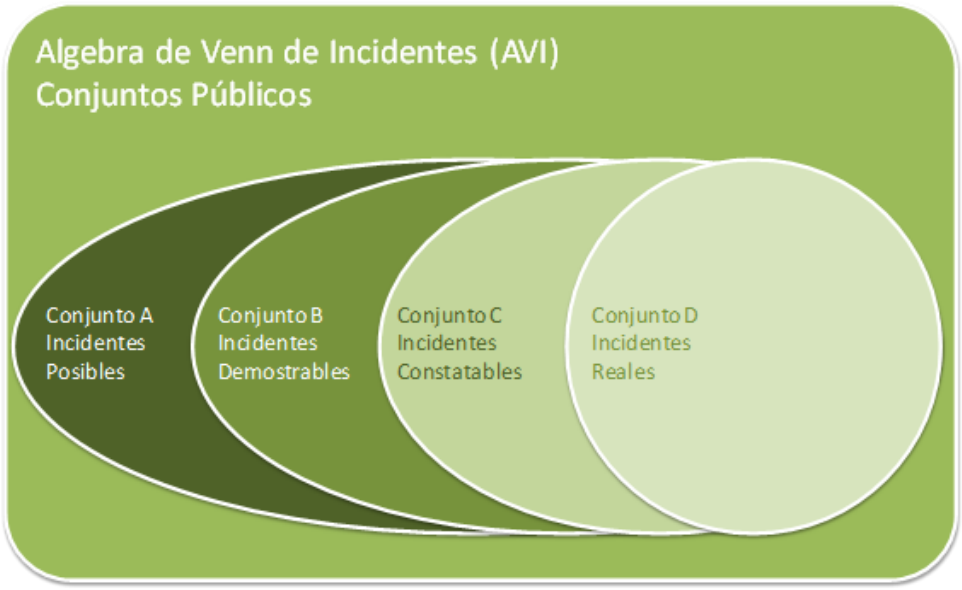


Figura 4.23. AVI. Conjuntos Públicos

Conjunto A: Conjunto de todos los incidentes posibles. Es el conjunto de mayor tamaño, aquí se ubican todos los incidentes de seguridad posibles.

Conjunto B: Subconjunto del Conjunto A de los incidentes demostrables. En este subconjunto se encuentran todos aquellos incidentes de los que podría hacerse una demostración, independientemente de que realmente se haya hecho. En este subconjunto se incluyen tipos de ataques que no puedan demostrarse por razón de la complejidad o el coste de la demostración, pero que a ojos de los propietarios del AVI puedan darse como demostrables.

Conjunto C: Subconjunto del conjunto B de los incidentes constatables. En este subconjunto se encuentran todos los incidentes de los que tenemos demostración empírica. De estos incidentes debemos tener los datos específicos de su suceso, o tener referencia de qué laboratorio, Universidad, empresa o entidad dice tener demostración empírica de los mismos.

Conjunto D: Subconjunto del Subconjunto C de los incidentes que han sucedido en la realidad. En este subconjunto se encuentran los incidentes que realmente han sucedido en la realidad y que han tenido algún tipo de impacto para una víctima. De estos incidentes tenemos que tener recogidos los datos de la fecha, el impacto y en general la descripción de los mismos.

Para un conocimiento general de la realidad con estos cuatro conjuntos sería suficiente. Para que el AVI sea un elemento realmente útil en la toma de decisiones, es necesario particularizarlo para un sector industrial concreto y en una geografía concreta.

Conjunto E: Subconjunto del conjunto D de los incidentes que han sucedido para la industria en análisis. En este subconjunto se encuentran los incidentes que han sucedido al menos una vez dentro del sector industrial en análisis.

Conjunto F: Subconjunto del conjunto E de los incidentes que han sucedido para la región geográfica en análisis. En este subconjunto se encuentran los incidentes que han sucedido al menos una vez en el sector industrial en análisis y en la zona geográfica analizada.

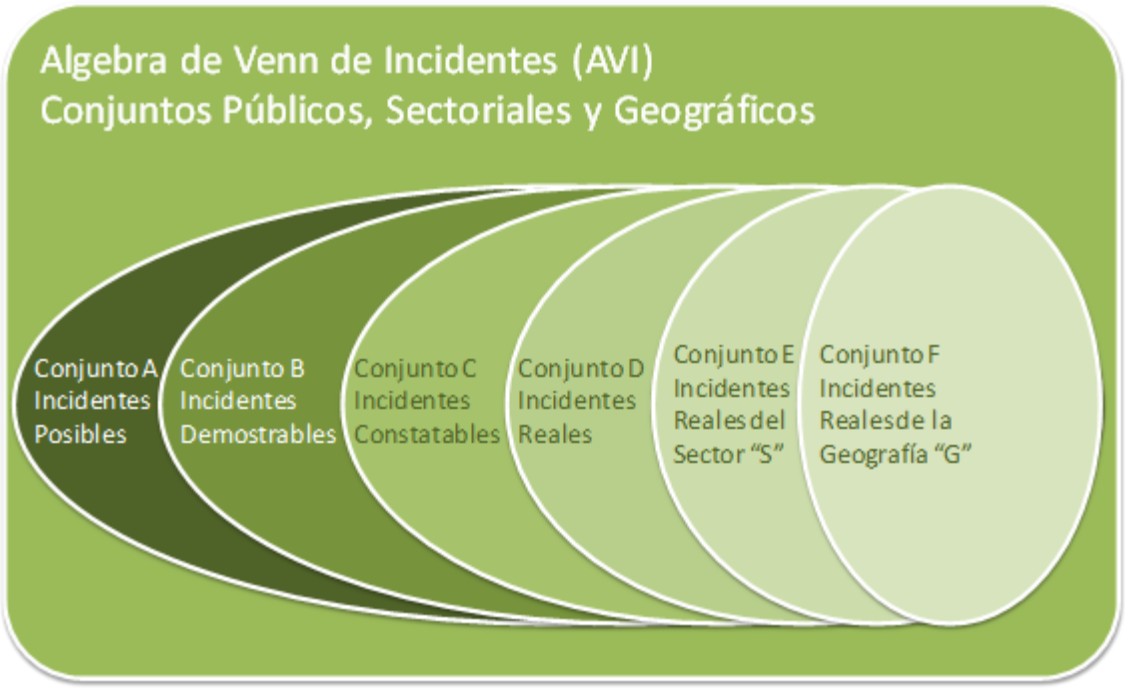


Figura 4.24. AVI. Conjuntos públicos, sectoriales y geográficos

También podríamos contar con el Subconjunto E' prima como conjunto del Subconjunto D de los incidentes que han sucedido al menos una vez en la zona geográfica en análisis. Por simplicidad no vamos a incluir este subconjunto en la representación del modelo y en la descripción de su uso, pero de la misma forma que se interpreta que las zonas geográficas son un subconjunto del sector industrial puede hacerse todo el desarrollo justo al revés, cuando el interés concreto sea conocer la situación general de una zona geográfica con todos los sectores industriales incluidos.

4.6.1.2.2 Conjuntos Privados

Junto a la información que podemos obtener de fuentes públicas, la Metodología Casandra incorpora la información que cada empresa tenga de sus propios incidentes (ver Figura 4.25).

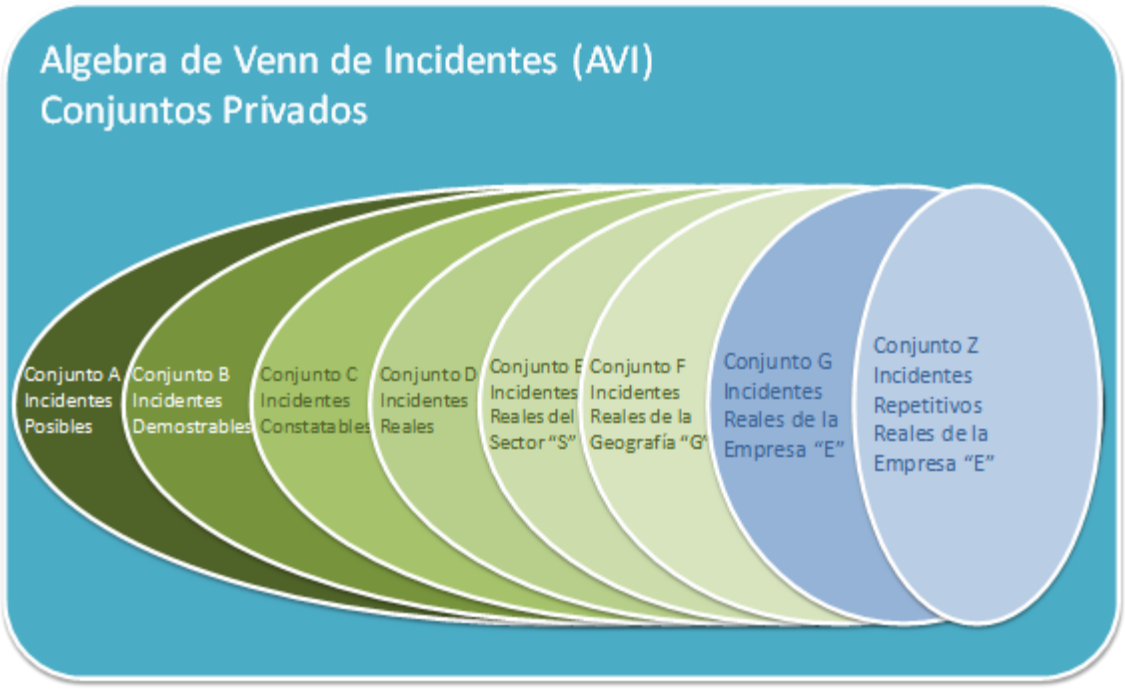


Figura 4.25. AVI. Conjuntos Privados

Cuánto más rica sea esta información, más se podrá ajustar la estrategia de riesgos a la compañía en análisis.

Conjunto G: Subconjunto del conjunto F y del conjunto E de los incidentes que han sucedido al menos una vez en la empresa en análisis. El subconjunto G lo tendremos presente siempre que estemos haciendo el AVI de una empresa en concreto. Podrá ser un subconjunto del subconjunto de incidentes de un sector industrial, de una zona geográfica o de ambos.

Como último subconjunto siempre tendremos el subconjunto de incidentes repetitivos. Cuando un incidente se presenta de forma recurrente, se ubicará dentro del Subconjunto Z. Lo denominamos con esta letra por ser siempre el último de los subconjuntos.

4.6.1.3 Ubicación del incidente en el Modelo AVI

Para ubicar correctamente un incidente necesitamos saber su tipo. Para ello, tenemos en la BDI un campo que nos indicará si el incidente es del tipo:

- ✓ **Hipotético:** Consideramos en este conjunto (Conjunto A del Modelo AVI) aquellos incidentes que quedan en el campo de lo teórico. No existe demostración empírica del incidente.
- ✓ **Demostrable:** Se corresponde con el Conjunto B del Modelo AVI. Son los incidentes de los que si bien no existe ninguna demostración empírica el propietario del AVI puede decidir que con la inversión y el tiempo suficiente podría pasar al conjunto de los incidentes demostrados.
- ✓ **Demostrado:** Se corresponde con el Conjunto C del Modelo AVI. Son los incidentes de los que existe demostración empírica, aunque no exista ningún caso real de incidente documentado donde se haya utilizado este ataque.
- ✓ **Real:** Se corresponde con el Conjunto D del Modelo AVI. Son los incidentes de los que existe constancia pública de su ocurrencia.
- ✓ **Interno:** Se corresponde con el Conjunto G del Modelo AVI. Son los incidentes reales que no han salido a la luz pública pero que el propietario del Modelo AVI conoce y le sirven para enriquecer su modelo de decisión. Es recomendable que en la implementación física de los nombres de las empresas e instituciones asociadas a este tipo de incidente no aparezcan los nombres reales y se sustituyan por un código que no forme parte del al BDI.

Junto a la información en cuanto al tipo de incidente debemos incorporar la ubicación empresarial y geográfica en la que se ha producido.

- ✓ **País:** Debe recoger el código del país en el que se ha producido el incidente.
- ✓ **Sector:** Debe recoger el sector empresarial en el que se ha producido el incidente.
- ✓ **Empresa:** Debe recoger el nombre (o el código por discreción) en el que se ha producido el incidente.

Dado que el ataque puede darse simultáneamente desde varios países, sobre ciudadanos de distintos países y afectando múltiples empresas de múltiples geografías, debemos considerar

que estos campos pueden contener bien el código del país donde radica la empresa o institución que acumula los daños principales, o un código del área geográfica que debe estar descrito en un diccionario adjunto de geografías que forma parte de la BDI.

Por ejemplo, si un tipo de incidente se está dando específicamente en distintos países de Sudamérica podemos crear el acrónimo “LATAM” de “LATino AMerica” e incluirlo en el diccionario de geografías.

Algo similar puede suceder con los sectores industriales. Es posible que haya un incidente que incluya a más de un sector industrial. Al igual que con las geografías, el diccionario adjunto de sectores industriales, que forma parte de la BDI, pude incluir códigos que agrupen distintos sectores industriales. Igual sucede con el diccionario homólogo para empresas.

Aunque gran parte de incidentes puedan necesitar una riqueza informativa mayor cara a hacer un análisis estadístico más fino, incluyendo múltiples ubicaciones geográficas en el ataque, el objeto de este campo es servir de base para la toma de decisión por un analista de riesgo que quiere determinar si un tipo de ataque es susceptible de darse para una industria concreta y/o en una geografía concreta.

No obstante, con el objetivo de alcanzar un nivel muy alto y exhaustivo de la calidad de la información alojada en la BDI, la implementación física debe recoger la capacidad de crear por parte del usuario listas de países, sectores industriales y empresas que sean unión de listas y/o elementos ya definidos.

En el caso de que la empresa esté informada, es decir que estemos ante un incidente real, debemos contar con campo que indique la frecuencia del incidente en esa empresa:

- ✓ **Puntual:** Estamos ante un incidente que no se había producido anteriormente en la empresa.
- ✓ **Repetitivo:** Estamos ante un incidente que se ha producido más veces en la empresa en la que se ha identificado.

Junto con los datos de ubicación del incidente en el Modelo AVI es fundamental recoger toda la información que se requiere para determinar el impacto real de cada incidente, a efectos de tener la mejor capacidad posible de priorización de las acciones de las medidas de protección. Para tener completo el Modelo AVI hay que identificar también los siguientes valores para cada incidente:

- ✓ **El activo o conjunto de activos que han sido atacados:** A tal efecto debe definirse un diccionario de activos de información atacados. Al igual que sucedía con las empresas, sectores y geografías, el usuario debe poder definir nuevos conjuntos de activos como unión de activos definidos anteriormente.
- ✓ **Principio de Seguridad atacado:** Confidencialidad, Integridad y/o Disponibilidad.
- ✓ **Tipo de ataque desde el punto de vista de su intencionalidad:** Se catalogan como accidentales o intencionales.
- ✓ **Impacto:** Para considerar el impacto hay que identificar tres tipos de impactos posibles:
  - **Impactos de imagen:** Suelen ser muy difíciles de valorar. Se va a proponer la creación de una escala exponencial cualitativa referenciada a casos reales.
  - **Robo de activos de Información:** Pueden medirse por el volumen de activos que hayan sido sustraídos o alterados.
  - **Monetización de los activos:** Valor económico conseguido por el atacante tras la venta o utilización de los activos afectados por el incidente.

Como resumen de los campos involucrados en la ubicación de cada incidente presentamos la Tabla 4-14. Con estos datos queda recogida toda la información requerida para instrumentar la ubicación de cada incidente en el Modelo AVI.

Elemento	Valores posibles
Código del Incidente	• Secuencial numérico
Verosimilitud del Incidente	• Hipotético • Demostrable • Demostrado • Real • Interno
País	• Código del País o países
Sector Industrial	• Código del Sector o Sectores
Empresa o Institución	• Código de la empresa o empresas
Repetición	• Puntual • Repetitivo
Activo o Conjunto de Activos	• Código del Activo
Principio de Seguridad Atacado	• Confidencialidad • Integridad • Disponibilidad
Tipo de Ataque	• Accidentales • Intencionales
Impacto	• Imagen • Fuga de Información • Monetización de activos

Tabla 4-14. Elementos y valores posibles de los incidentes

4.6.1.4 Proceso de Identificación de incidentes candidatos

La Figura 4.26 muestra el proceso de Identificación de nuevos incidentes que son candidatos para realizar Minería de Enterprise Security Patterns.

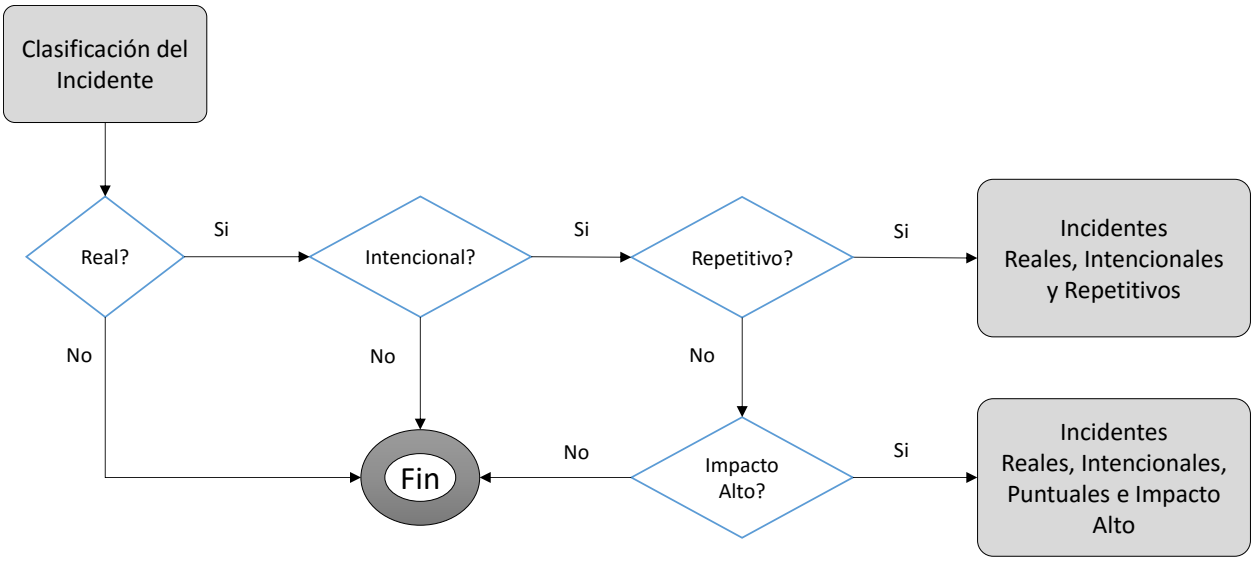


Figura 4.26. Proceso de Identificación de Nuevos Incidentes Candidatos

Como muestra la figura anterior, para cada uno de los incidentes que aparecen publicados en los espacios especializados hay que responder una serie de preguntas con el objetivo de focalizar y priorizar la minería de estos patrones. A continuación, la lista de preguntas:

1. ¿Es un incidente real?
    a. En caso negativo acabamos el proceso, ya que no vamos a priorizar la Minería de Enterprise Security Patterns sobre incidentes que no sean reales.
    b. En caso afirmativo pasamos a la siguiente pregunta.
2. ¿Es un incidente intencional?
    a. En caso negativo acabamos el proceso, ya que no vamos a priorizar la Minería de Enterprise Security Patterns sobre incidentes que no sean intencionales.
    b. En caso afirmativo pasamos a la siguiente pregunta.

3. ¿Es un incidente repetitivo?

    a. En caso afirmativo categorizamos el incidente dentro de los *incidentes Reales, Intencionales y Repetitivos*. Este grupo de incidentes va a ser uno de los dos grupos que se van a priorizar a la hora de hacer minería. Dentro de este grupo también se evaluará el impacto del incidente.

    b. En caso negativo pasamos a la siguiente pregunta.

4. ¿Es un incidente de impacto alto?

    a. En caso negativo acabamos el proceso, ya que no vamos a priorizar la Minería de Enterprise Security Patterns sobre incidentes que no sean repetitivos o tengan alto impacto.

    b. En caso afirmativo categorizamos el incidente dentro de los *incidentes Reales, Intencionales, Puntuales e Impacto Alto*. Este grupo de incidentes va a ser el segundo de los dos grupos que se van a priorizar a la hora de hacer minería.

Como conclusión de este proceso, los grupos de incidentes categorizados como "*Reales, Intencionales y Repetitivos*" y "*Reales, Intencionales, Puntuales e Impacto Alto*" son los incidentes candidatos a seguir el resto de puntos incluidos en el proceso de Minería de Enterprise Security Patterns.

4.6.2 Identificación de las Amenazas

Una vez ya hemos decidido el nuevo incidente que queremos incluir dentro del proceso de minería, debemos identificar los riesgos y amenazas que una organización enfrenta ante ese incidente.

Para identificar los distintos tipos de amenazas incluidos en un incidente o problema, debemos identificar los actores o usuarios involucrados en el marco del problema para ese

entorno. Principalmente, los actores que vamos a analizar son los incluidos en el meta-modelo del Contexto de los Enterprise Security Patterns (ver Sección 4.4.2), más un cuarto actor que por defecto va a generar incidentes intencionales (Atacante):

- ✓ Cliente
- ✓ Empleado
- ✓ Usuario Técnico
- ✓ Atacante

Con el objetivo de ayudar en la identificación de las amenazas y sus contramedidas a continuación vamos a mostrar (i) un Catálogo de Amenazas y (ii) un Catálogo de Contramedidas. Dentro de ambos catálogos no están todas las amenazas o contramedidas que se pueden encontrar en todos los incidentes. Según se vayan documentando más incidentes estos catálogos pueden seguir creciendo en tamaño y contenido.

4.6.2.1 Catálogo de Amenazas

Como ya hemos comentado en la Base de datos de incidentes, las dos categorías de amenazas existentes son las intencionales y las accidentales. Dado que el proceso de minería de Enterprise Security Patterns va a priorizar los incidentes intencionales, el catálogo de amenazas incluido en esta sección está enfocado a ese tipo de incidentes. A continuación, mostramos un listado de posibles amenazas intencionales:

- ✓ Manipulación indebida de configuración o manipulación indebida de configuración de infraestructura.
- ✓ Suplantación de identidad.
- ✓ Abuso de privilegios.
- ✓ Acceso / uso no autorizado.
- ✓ Difusión o distribución no autorizada y/o dañina.

- ✓ Denegación de servicios.
- ✓ Re-encaminamiento.
- ✓ Análisis de tráfico / Escucha o interceptación de información.
- ✓ Repudio.
- ✓ Modificación o alteración no autorizada de información (corrupción de datos).
- ✓ Manipulación de código fuente.
- ✓ Robo de datos o recursos técnicos.
- ✓ Extorsión, sabotaje o fraude.
- ✓ Destrucción/agresión material o inmaterial.

Según la tipología de los elementos incluidos en el incidente, se puede realizar una categorización más ajustada de las amenazas que podemos encontrar en cada uno de los elementos que pueden ser identificados (sistemas de información, personas e información).

4.6.2.1.1 Sistemas de información

A continuación, mostramos un catálogo de amenazas vinculadas principalmente a los sistemas de información:

- ✓ **Deficiencia en la validación de las entradas de datos, ejecución de comandos o deficiencias de programación**. Por ejemplo: entradas no validadas utilizadas para generar *queries* en bases de datos, fallo en la validación de entrada de cookies, parámetros de *queries*, cabeceras HTTP, bases de datos o recursos de red, desbordamientos de buffer, etc.
- ✓ **Deficiencias en la Autenticación**. Por ejemplo: utilización de contraseñas débiles, almacenamiento de credenciales en texto claro en archivos de configuración.

- ✓ **Deficiencias en la Autorización o Gestión de sesiones**. Por ejemplo: Inadecuada segregación de funciones, fallo en el bloqueo de acceso a recursos en función de identidades de aplicaciones, predicción de credenciales o sesiones, etc.
- ✓ **Débil gestión de la configuración.** Por ejemplo: Utilización de interfaces de administración inseguros, Utilización almacenamiento de configuración inseguro, des configuración de aplicaciones o servidores, etc.
- ✓ **Ataques en cliente**. Por ejemplo: Denegación de servicios, Phishing, Cross Site Scripting.
- ✓ **Revelación de información.** Por ejemplo: Fuga de información, Robo de información, Indexación de directorio, Protección de capa de transporte insuficiente.
- ✓ **Ataques lógicos / Deficiencias en la Gestión de excepciones**. Por ejemplo: Abuso de funcionalidad, Denegación de Servicios, Fallo en el manejo de excepciones estructurado.

4.6.2.1.2 Personas

A continuación, mostramos un catálogo de amenazas vinculadas principalmente a las personas:

- ✓ **Ingeniería Social.** Por Ejemplo: Llamadas de teléfono para obtener usuarios y contraseñas, datos confidenciales de clientes, etc.
- ✓ **Robo de Datos.** Por ejemplo: documentos impresos encima de las mesas, etc.
- ✓ **Robo de Dispositivos Físicos.** Por ejemplo: documentos almacenados en un USB, documentos almacenados en ordenadores portátiles, etc.

4.6.2.1.3 Información

A continuación, mostramos un catálogo de amenazas vinculadas principalmente a la información:

- ✓ **Incumplimiento de requisitos legales.** Por ejemplo: LOPD, marco regulatorio, etc.
- ✓ **Acceso a datos sensibles.** Por ejemplo: Acceso en almacenamiento o en memoria (swap).
- ✓ **Ocultación de datos.** Por ejemplo: imágenes, texto, vídeo, etc.
- ✓ **Escuchas (Sniffing).** Por ejemplo: Monitorización del tráfico de red, análisis de puertos, inyección de caracteres en conexiones para la emulación de comandos.
- ✓ **Fragmentación de paquetes en red (Fragmentación IP).** Por ejemplo: Burla de técnicas básicas de inspección de datagramas IP para el envío de información o la utilización de información no autorizada.
- ✓ **Pérdida de integridad o confidencialidad de información.** Por ejemplo: Modificación / alteración de información, Pérdida o cracking de claves de cifrado, ataques de modificación / alteración de certificados digitales.

4.6.2.2 Catálogo de Contramedidas

Es posible clasificar salvaguardas o contramedidas atendiendo a:

- ✓ El aspecto de seguridad en el que se centran.
- ✓ La estrategia que utilizan parar detener las amenazas.

A continuación, se describen las siguientes estrategias de control (prevención, detección, mitigación y corrección) para minimizar la probabilidad de éxito de ataques y/o de objetivos clave:

4.6.2.2.1 Prevención

- ✓ Control de acceso lógico y físico o estructural.
- ✓ Entrenamiento del personal o educación de usuarios de red a través de cursos de formación.

- ✓ Implementación de planes de concienciación.
- ✓ Asignación de responsabilidades.
- ✓ Carteles, advertencias, mensajes de información.
- ✓ Cifrado de información.
- ✓ Utilización de firewalls de protección de accesos a redes y subredes.
- ✓ Organización de claves de acceso.
- ✓ Validación de entradas: longitud, rango, formato y tipo.
- ✓ Utilización de políticas de contraseñas fuertes y forzar el cumplimiento de la política a través de reglas en los sistemas de cómputo.
- ✓ Utilización de mecanismos de autenticación que no requieran el envío de credenciales en texto claro.
- ✓ Cifrado de canales de comunicación para securizar tokens de autenticación.
- ✓ Aplicación de Políticas de segregación de funciones.
- ✓ Aseguramiento del canal de comunicación durante las sesiones abiertas.
- ✓ Cambio de contraseñas de forma periódica.
- ✓ Reducción de *timeouts* de sesión.
- ✓ Utilización de autenticación y autorización fuerte en interfaces de administración.
- ✓ Anulación de configuración predeterminada o por defecto.
- ✓ Comprobación periódica de copias de respaldo.
- ✓ Filtrado de comandos en aplicaciones para evitar determinados ataques como el XSS (evitar comandos como SCRIPT, OBJECT, APPLET, EMBED o FORM).
- ✓ Envío de mensajes intimidatorios cuando se detecta que un usuario de aplicación intenta un posible ataque.

4.6.2.2.2 Detección

- ✓ Registro de logs y bitácoras.
- ✓ Alarmas tempranas.
- ✓ Reporte de excepciones.
- ✓ Monitorización de actividades o de vigilancia, de movimiento, de metales, etc.
- ✓ Auditoría periódica.
- ✓ Sistemas de detección de intrusos: Establecimiento de reglas para identificar comportamiento malicioso.

4.6.2.2.3 Mitigación

- ✓ Actualización de software o cambio de versión.
- ✓ Utilización de versiones recientes de navegadores (nuevos filtros de seguridad introducidos).
- ✓ Sistemas de prevención de intrusos o filtrado de paquetes.
- ✓ Utilización de antivirus, antispyware o antimalware.
- ✓ Puesta en práctica de programación segura (Utilización de try/catch para el manejo de excepciones).
- ✓ Definición de máximo número de conexiones o capacidad de hosts para evitar ataques de tipo DoS.
- ✓ Reemplazo de funciones inseguras para evitar ataques de tipo buffer overflow.
- ✓ Autenticación de extremos de comunicación para evitar ataques de IP spoofing.
- ✓ Utilización de cuentas con privilegios restringidos (el menor número necesario).
- ✓ Anulación del almacenamiento de información sensible en espacio Web o secretos en software.

- ✓ Revisión de código fuente.

4.6.2.2.4 Corrección

- ✓ Establecimiento de sitios redundantes.
- ✓ Respaldo de datos o realización de copias de seguridad.
- ✓ Remediación a través de parches de seguridad específicos sobre aplicaciones,
- ✓ Respaldo de fuentes de potencia o baterías.
- ✓ Establecimiento y ejecución del Plan de Gestión de incidentes.
- ✓ Establecimiento y ejecución del Plan de Recuperación ante desastres o emergencias de seguridad.

4.6.3 Identificación del Contexto

Para poder identificar el Contexto del patrón debemos conocer (i) los activos de información a proteger, (ii) los dominios de seguridad desde donde se accede al dato y por donde el dato es transmitido, y (iii) los *stakeholders* o usuarios que están accediendo activo de información.

Como podemos ver en la Sección 4.6.1 Bases de Datos de Incidentes (BDI), dentro de la categorización de los incidentes están incluidos los activos de información afectados. Para este caso en particular, dentro de la identificación del Contexto solo tenemos que instanciar esa característica del incidente para documentar los activos de información involucrados en el contexto del patrón.

Los stakeholders o usuarios también suelen ser fáciles de identificar, ya que solo tenemos 3 tipos de usuarios (clientes, empleados y usuarios técnicos) y en la gran mayoría de los casos los usuarios que utilizan los sistemas están dentro de los grupos de clientes o empleados.

La identificación de los dominios de seguridad si puede ser un poco más tediosa, sobre todo si el incidente es dado en un sistema de información complejo o si el sistema de información está alojado en distintos proveedores. Para este caso particular, se puede utilizar el enfoque de Artefactos (*Artifacts*) y/o enfoque Sociológico dentro del modelo de minería de patrones propuesto por (Kerth y Cunningham, 1997).

En el enfoque de Artefactos, los investigadores analizan sistemas construidos por otras personas que intentan resolver un problema en un contexto similar, permitiendo encontrar puntos en común entre diferentes sistemas. En este enfoque, los investigadores deben buscar un conjunto de documentos públicos dentro de fuentes de información públicas. El objetivo inicial es crear una lista de documentos públicos que ayuden a solucionar el mismo problema de seguridad. Después de crear la lista inicial de documentos, los investigadores deben analizar las soluciones de seguridad proporcionadas para encontrar puntos en común entre diferentes soluciones analizadas, con el fin de obtener un contexto único para al problema dado.

En el enfoque Sociológico, los investigadores entrevistan a las personas que han construido sistemas similares para conocer como resolvieron problemas particulares, permitiendo encontrar contextos similares en los sistemas analizados. Los investigadores deben preguntar a las personas que habían diseñado las arquitecturas de seguridad cómo resolvieron problemas particulares, a fin de encontrar contextos recurrentes en los diseños analizados.

4.6.4 Diseño de la Solución

Para realizar el diseño de la solución dentro del proceso de Minería de Enterprise Security Patterns, una vez ya han sido identificadas las amenazas y el contexto del patrón, solo hay que seguir las pautas definidas en el proceso de Modelado de Arquitecturas de Seguridad Empresariales dirigidas por modelos.

Con el objetivo de no repetir la información presentada en el proceso de modelado, aquí dejamos una referencia a la sección donde ya ha sido presentado (ver Sección 4.5.1).

5. Prototipo Tecnológico

5.1 Introducción

Tal y como ha quedado patente hasta ahora, los modelos juegan un papel clave en la propuesta metodológica de la presente Tesis Doctoral. Por tanto, como con cualquier otra propuesta que aplique los principios de la Ingeniería dirigida por Modelos (Selic, 2003), resulta prácticamente imprescindible acompañar la propuesta metodológica de una solución tecnológica. Este capítulo presenta la solución tecnológica que acompaña la propuesta metodológica de la presente Tesis Doctoral: C-SMArT (*Casandra - Security Model-driven Architecture Toolkit*).

Dicha solución proporciona un entorno de modelado que facilita la creación y gestión de modelos elaborados con los Lenguajes Específicos de Dominio (DSL) presentados en el capítulo anterior, y contribuye decisivamente a la consideración de los Enterprise Security Patterns como marco completo para la especificación y diseño de arquitecturas de seguridad empresariales.

El hecho de disponer de una solución tecnológica que acompañe la propuesta metodológica facilitará la adopción de esta, ya que la solución tecnológica actuará como implementación de referencia de la propuesta (Curran, 2003). Disponer de una implementación de referencia permite mostrar, no solo que la aplicación de la propuesta es factible, sino cómo debe ser aplicada. No en vano, la utilidad de cualquier propuesta metodológica que no venga acompañada de una implementación de referencia es cuando menos discutible. Además, el proceso de desarrollo de la solución tecnológica constituye en sí mismo una validación adicional de la propuesta, pues la implementación obliga a reconsiderar y revisar las diferentes decisiones de diseño incluidas en la propuesta. Por ejemplo, la simple implementación de los meta-modelos de cada uno de los DSLs implicó revisar y refinar la especificación de dichos lenguajes.

Conceptualmente, la decisión de construir un entorno de modelado que soporte la propuesta resulta en una serie de ventajas:

- ✓ Soporte para la creación, edición y gestión de modelos de forma intuitiva y eficiente. No en vano, la que ha sido reconocida como la principal ventaja de la Ingeniería Dirigida por Modelos históricamente es el soporte que proporciona para documentar arquitecturas software (Whittle et al., 2014). La solución que se presenta en este capítulo es un perfecto ejemplo de la utilidad de los entornos de modelado para este fin, proporcionando el soporte ideal pasa la consulta y visualización de las decisiones de diseño de alto nivel contenidas en los modelos elaborados con C-SMArT.
- ✓ La utilización de un entorno de modelado proporciona además algunos mecanismos para la verificación de la coherencia y consistencia de los modelos elaborados. Aún sin integrar mecanismos de validación adicionales, el simple hecho de que el meta-modelos de un DSL define las reglas sintácticas que debe cumplir cualquier modelo elaborado con dicho DSL, impone una serie de normas y restricciones que la herramienta obligará al usuario a seguir cuando la utilice para elaborar un modelo (Selic, 2012). Además, el entorno que se presenta tiene una naturaleza extensible, sería relativamente sencillo y técnicamente poco complejo implementar e integrar mecanismos de validación de modelos adicionales.
- ✓ Finalmente, y relacionada con la anterior, la extensibilidad del entorno de modelado que se presenta facilita enormemente la interoperabilidad con otras propuestas en el campo de la Ingeniería dirigida por Modelos. Sería factible, por ejemplo, conectar las decisiones de diseño incluidas en una arquitectura diseñada con C-SMArT, con las herramientas de *reporting* que proporcionasen los dispositivos desplegados en el entorno real, de manera que podría enriquecerse la vista de alto nivel proporcionada por los modelos C-SMArT con señales visuales soluciones que alertasen de posibles problemas o informasen del correcto funcionamiento de la solución. Una vez que se dispone de un primer prototipo como el que se presenta en este capítulo, las posibilidades de interconexión con otros sistemas o herramientas son casi infinitas.

Las siguientes subsecciones describen el proceso de desarrollo seguido para construir el entorno de modelado propuesto e ilustran algunas decisiones de diseño relevantes.

5.2 Proceso de Desarrollo

La estrategia utilizada para construir la herramienta C-Smart se basa en las directrices definidas en M2DAT (Vara, 2009), que propone un proceso de desarrollo genérico pata construcción de herramientas que apliquen los principios de la MDE.

Así, el desarrollo del entorno de modelado que da soporte a la propuesta C-Smart se divide en dos etapas. En primer lugar, se establece la arquitectura de la solución, es decir, su diseño conceptual. Posteriormente, dicha arquitectura se refina en un diseño técnico, donde se especifican que tecnologías, lenguajes y plataformas se utilizarán para implementar la arquitectura conceptual diseñada en el paso anterior. Conviene mencionar que el proceso de desarrollo de la herramienta sigue un enfoque iterativo e incremental, por lo que ha sido posible refinar el entorno de modelado mientras se iba refinando la definición de los DSLs propuesto por C-Smart.

5.2.1 Diseño Conceptual

Conceptualmente, la arquitectura de C-SMArT presenta un alto nivel de modularidad. Tal y como se puede apreciar en la Figura 5.1, atendiendo a la dimensión vertical de la arquitectura, la herramienta se ha concebido como un conjunto de 5 componentes que permiten trabajar de manera independiente con cada uno de los 5 DSL soportados.

Por otro lado, atendiendo a la dimensión horizontal, la arquitectura de la herramienta se ha definido siguiendo el tradicional enfoque por capas (Parnas, 1972), distinguiendo la capa de la interfaz o presentación, la de la lógica de negocio y finalmente la de persistencia. En el contexto particular de la MDE, esta *separation of concerns* permite distinguir la presentación de cada modelo del propio modelo en sí mismo (Kulkarni y Reddy, 2003) y como veremos más adelante facilita la gestión de modelos. Más adelante se mostrará cómo la tecnología utilizada para la construcción de la herramienta soporta de forma semi-automática esta separación y cómo se ha integrado por tanto en C-SMArT.

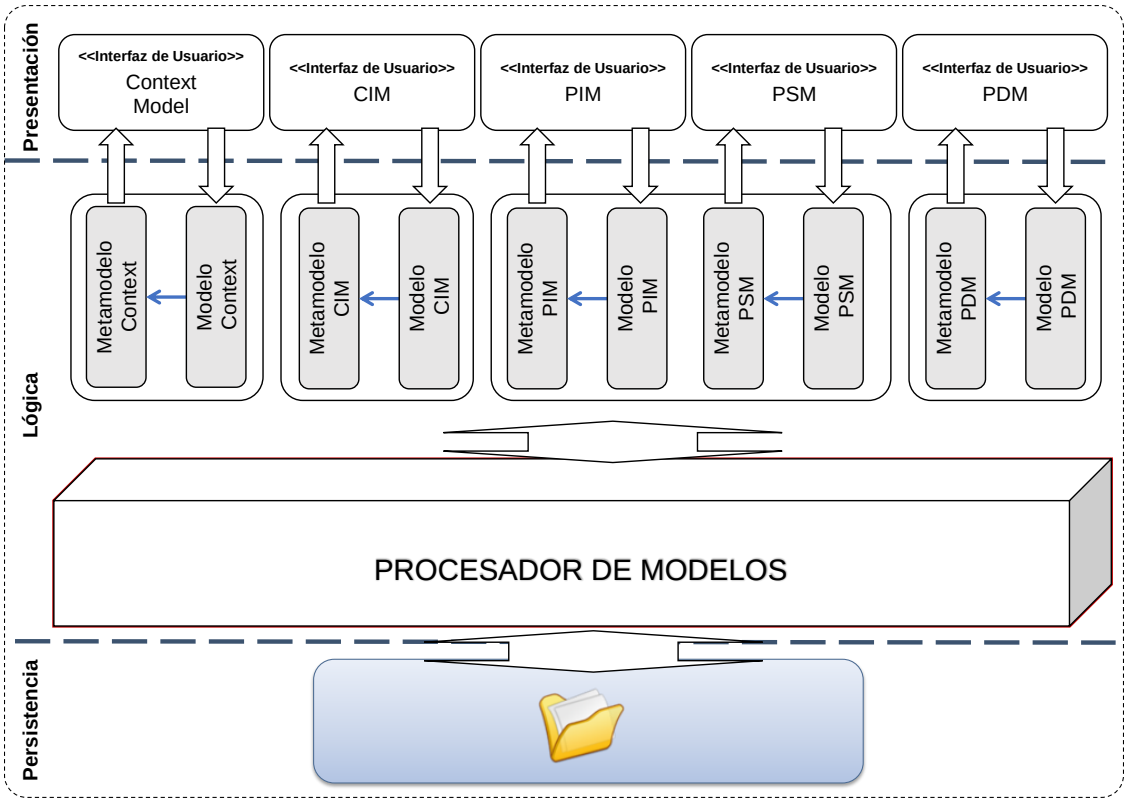


Figura 5.1. Arquitectura Conceptual del Toolkit C-SMArT

La capa de Presentación incluye los editores para trabajar con cada tipo de modelo para el diseño de Arquitecturas de Seguridad Empresariales propuesto por C-SMArT: el Modelo del Contexto, el Modelo CIM, el Modelo PIM, el Modelo PSM y el Modelo PDM propuestos por C-SMArT. En consecuencia, la interfaz de C-SMArT incluirá los controles y paneles que permitirán editar, crear y administrar los diferentes modelos, mientras que los modelos en sí serán gestionados por la lógica de la aplicación.

La capa de la Lógica será por tanto la encargada de gestionar los modelos elaborados con C-SMArT. Nos referimos en este caso a la sintaxis abstracta de los modelos, es decir, la información relativa a qué elementos contiene y qué relaciones hay entre ellos. La información relativa a su representación, es decir, que símbolo gráfico se utiliza para representar cada elemento y en qué parte de la pantalla o del panel de dibujo se encuentra, es lo que se denomina sintaxis concreta. Nótese que esta información es en realidad accesoria desde el punto de vista del procesamiento del modelo. En efecto, cuando se analice una arquitectura de seguridad

empresarial representada con uno de los modelos soportados por C-SMArT, lo relevante será identificar qué elementos arquitectónicos se han incluido en el modelo y con qué otros elementos arquitectónicos se relacionan. La forma exacta en que esos elementos se hayan representado gráficamente en el modelo no es tan importante.

Para soportar esta separación entre sintaxis abstracta y concreta, en esta capa se incluyen implementaciones del metamodelo de cada DSL y un conjunto de parsers que, a partir de la información recogida de la interfaz, son capaces de generar modelos conformes al metamodelo en cuestión. Además, dado que es en esta capa donde está la información relevante sobre cada modelo (la sintaxis abstracta), será esta capa intermedia el punto donde integrar soporte para cualquier tarea adicional que se quiera realizar con los modelos producidos con la herramienta. Por ejemplo, sería el lugar donde integrar transformaciones que permitiesen derivar un PIM de un CIM, o herramientas de *weaving* que permitiesen identificar las relaciones entre un PDM y un PSM, generadores de código o de informes que documentasen los modelos, nuevos validadores que comprobasen un particular conjunto de reglas o restricciones, etc. Siguiendo la idea recogida en (Völter, 2008), nos referiremos al módulo encargado de semi-automatizar todas estas tareas relacionadas con la gestión de modelos como procesador de modelos.

Finalmente, la capa de Persistencia de la herramienta C-SMArT consiste en un tradicional sistema de ficheros que incorpora políticas habituales de control de versiones. El uso de sistemas de persistencia más complejos, como bases de datos o similare, han sido finalmente descartados, ya que, por el momento, sólo traerían complejidad adicional al desarrollo de la herramienta.

5.2.2 Diseño Técnico

Una vez que la arquitectura conceptual de la herramienta y sus características estructurales han sido definidas, el siguiente paso lógico es seleccionar el enfoque y las tecnologías que se van utilizar para implementar cada componente.

Al respecto, conviene destacar que el procesador de modelos del prototipo presentado en esta Tesis Doctoral no soporta ninguna funcionalidad adicional, aunque como se ha mencionado varias veces, ha sido diseñado y construido para ser extendido, de manera que incorporar o integrar nuevas funcionalidades de procesamiento de modelos sea relativamente sencillo e inmediato. Dicho de otro modo, en su estado actual, de C-SMArT ofrece soporte para elaborar y gestionar cada uno de los modelos de arquitecturas de seguridad empresariales definidos en la propuesta metodológica, pero no implementa ninguna relación entre ellos.

La Figura 5.2 muestra los principales componentes incluidos en la arquitectura técnica actual y lo que podría ser una evolución de la herramienta.

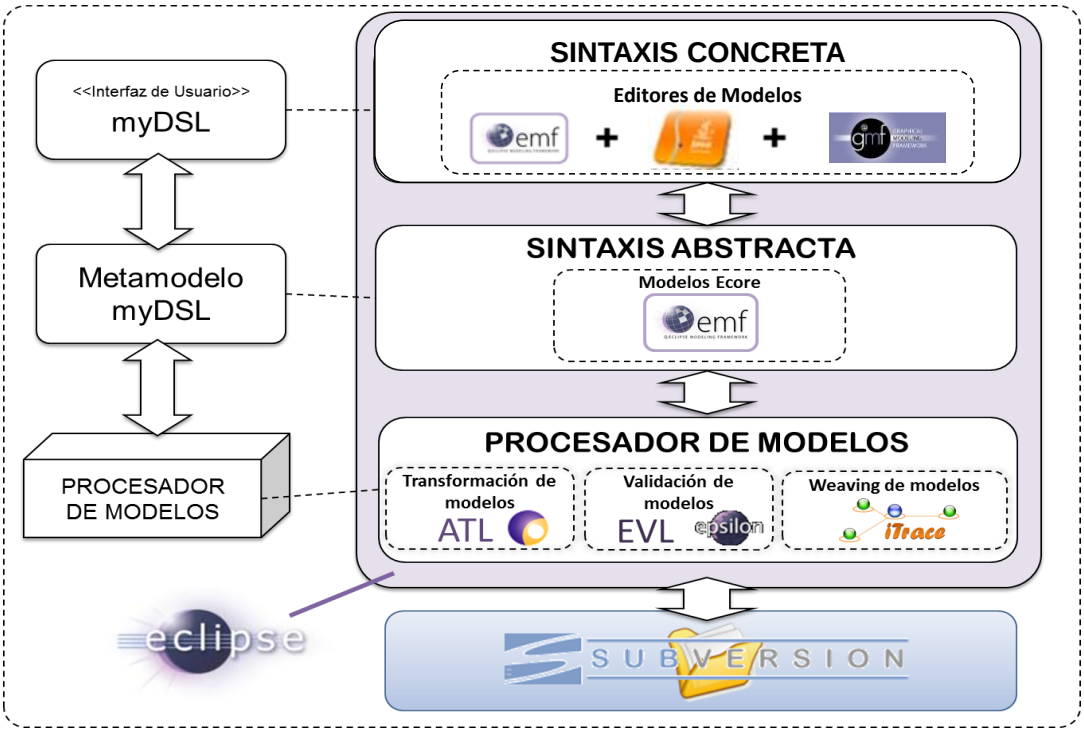


Figura 5.2. Arquitectura Técnica del Toolkit C-SMArT

Como se puede observar, Eclipse es la plataforma sobre la que se construye la herramienta. En particular, la parte central de la figura que ilustra el diseño técnico de la herramienta revela cómo se implementa el mencionado concepto de *separation of concerns* sobre Eclipse. Para ello se utilizan las facilidades proporcionadas por EMF (Steinberg et al., 2008) para, a partir del meta-modelo de cada DSL, generar automáticamente el código que implementa

una serie de *parsers* que se encargan de persistir los modelos elaborados por el usuario con la herramienta. En realidad, la información contenida en los ficheros que serializan estos modelos es simplemente la referente a la sintaxis abstracta del modelo en cuestión.

Por otro lado, para dar soporte a la representación gráfica de los modelos se utilizan las funcionalidades que proporciona EMF y GMF (Gronback, 2009) para el desarrollo de 2 editores gráficos para cada DSL. El primero proporciona una funcionalidad básica: permite editar modelos utilizando una representación en forma de árbol y suele utilizarse para tareas de depuración. El segundo es un diagramador que utiliza una representación tipo UML (nodos y flechas junto a iconos gráficos). Así, podemos decir que este componente es el que se encarga de dar soporte tecnológico para implementar la sintaxis concreta de cada uno de los DSL soportados por la herramienta.

Finalmente, a partir del meta-modelo de cada DSL, EMF también genera una API reflexiva, que permite acceder programáticamente a cualquiera de los modelos elaborados con el DSL en cuestión. De manera que la forma de integrar nuevas funcionalidades para el procesamiento de modelos en el prototipo actual pasa simplemente por hacer uso de esta API para acceder a los modelos elaborados con la herramienta. Haciendo uso de la API, la sintaxis abstracta de cualquier modelo elaborado con C-SMArT podría ser utilizada como entrada o salida de cualquier tarea que automatizase algún tipo de procesamiento, desde la simple generación de informes a complejas transformaciones. Así, aunque este primer prototipo no incluye soporte para estas funcionalidades, dado que si ofrece el punto de conexión para integrarlas en la herramienta, a continuación se apuntan algunos ejemplos de tareas para las que se podría implementar soporte en el procesador de modelos, tal y como ilustra la parte inferior de la figura:

- ✓ El refinamiento sucesivo de modelos que implica el diseño de una arquitectura de seguridad empresarial de acuerdo a la propuesta de esta Tesis Doctoral, podría semi-automatizarse por medio de transformaciones de modelos (Sendall y Kozaczynski, 2003). Por ejemplo, las reglas o guías de transformación presentadas en la sección

4.5.3 Transformaciones o transiciones entre Modelos pueden ser desarrolladas haciendo uso del lenguaje ATL (Jouault et al., 2008).

- ✓ Para recoger las diferentes decisiones de diseño que guían el paso de un modelo a otro, podrían utilizarse modelos de *weaving* o de relaciones (Santiago et al., 2013). Por ejemplo, para pasar del modelo de contexto al modelo CIM propuestos en la presente tesis doctoral, debe especificarse qué *Security Policies* son asignadas a cada elemento *Realm* presente en el modelo de contexto. Esta asignación puede realizarse mediante modelos de relaciones *iTrace*, poniendo en práctica la idea que se recogía en (Santiago et al., 2013).
- ✓ Finalmente, podrían implementarse diferentes tipos de validación para cada modelo de la arquitectura de seguridad diseñada o incluso reglas que afectasen a varios modelos a la vez, utilizando soluciones como EVL (Epsilon Validation Language) (Kolovos et al., 2009), que soporta funcionalidades como definición de restricciones relacionadas, interacción con el usuario o especificación de nivel de severidad del error.

En cuanto a la persistencia física de los modelos, en esta versión de la herramienta no se realiza ningún aporte científico reseñable, ya que se construye utilizando un sistema de control de versiones tradicional. En particular, se utiliza *Subclipse*, una implementación de *Subversion* (Pilato et al., 2008) para Eclipse. Si en algún momento, el número de modelos a gestionar o el tamaño de los mismos, implicase problemas de escalabilidad, podría optarse por cualquiera de las soluciones para gestionar la persistencia de grandes modelos (Espinazo y Garcia, 2014; Daniel et al., 2016).

5.3 Detalles Técnicos

En esta sección se destacan brevemente algunas observaciones reseñables respecto a la implementación de la propuesta metodológica de la presente Tesis Doctoral en la herramienta C-SMArT, que, como se ha mostrado, proporciona varios DSLs y el entorno de modelado necesario para utilizarlos.

Estas observaciones se estructuran en dos grandes bloques: decisiones referentes a la especificación de la sintaxis abstracta de los diferentes DSL construidos, detalles de implementación relativos al desarrollo de los editores gráficos (sintaxis concreta).

5.3.1 Sintaxis Abstracta

Esta sección ilustra las decisiones de alto nivel más reseñables en cuanto a la definición de la sintaxis abstracta de los DSL incluidos en C-SMArT.

La primera y más importante decisión tiene que ver con el hecho de que, aunque en esta memoria se viene hablando de 5 DSL diferentes, en realidad se han implementado como un único DSL con 5 sintaxis concretas diferentes.

En efecto, en el proceso de modelado descrito en la sección 4.5.1 se propone la elaboración de 5 modelos con diferentes niveles de abstracción. A priori, la elaboración de 5 modelos distintos sugeriría el desarrollo de 5 DSL diferentes para soportar la elaboración de cada modelo. No obstante, del análisis de los elementos de modelado que se manejan en cada nivel de abstracción, se puede observar que, elevar el nivel de detalle en cada paso del proceso implica simplemente añadir nuevos elementos de modelado al conjunto de elementos de modelado que ya incluía el nivel anterior. No se modifican ni eliminan en cambio elementos del conjunto. Es decir, se puede contemplar el CIM como un modelo de contexto ampliado, el PIM como un CIM ampliado, y así sucesivamente.

En este escenario, la decisión fue definir un único meta-modelo que incluyese todos los elementos de modelado necesarios para elaborar cualquiera de los 5 modelos contenidos en la propuesta metodológica, tal y como ilustra la vista parcial del meta-modelo *Ecore* que se muestra en la Figura 5.3.

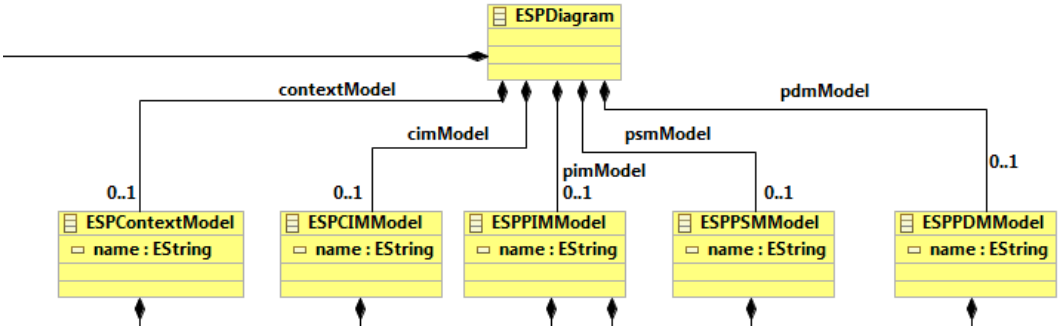


Figura 5.3. Vista parcial del meta-modelo C-SMArT: meta-clases de los modelos

Así, los elementos de modelado propios de cada nivel de abstracción (Contexto, CIM, PIM, PSM y PDM) se incluyen en el meta-modelo como componentes de la meta-clase que representa cada modelo (*ESPContexModel*, *ESPCIMModel*, etc.). Este diseño permite elaborar cualquiera de los modelos implicados en el proceso de modelado definido en la presente Tesis Doctoral con un mismo DSL.

Por otro lado, a partir de este meta-modelo único se construyeron 5 editores gráficos diferentes: uno para cada modelo contemplado en la propuesta metodológica. Puede hablarse así de un único DSL con 5 sintaxis concretas diferentes, o 5 DSL diferentes.

Conviene aclarar que en realidad se trata de 10 editores gráficos, 2 por cada modelo de la propuesta: uno basado en EMF tipo árbol y un diagramador de tipo cajas y flechas, para cada modelo.

La alternativa habría sido definir 5 meta-modelos diferentes, que tendrían gran cantidad de elementos comunes: cada meta-modelo incluiría los elementos del modelo de menor nivel de abstracción. Evidentemente, ese diseño tendría un enorme impacto en términos de refinamiento y evolución: si se decidiese introducir algún cambio en alguno de los meta-modelos (modificar,

añadir o eliminar elementos de modelado), habría que replicar ese cambio en el meta-modelo de cada uno de los DSL utilizados para elaborar los modelos de menor nivel de abstracción, que por tanto incluyen al primero. Por ejemplo, un cambio sobre el meta-modelo del DSL para elaborar modelos de contexto debería ser replicado sobre los meta-modelos de los otros 4 DSL.

No obstante, si en un futuro fuera preciso hacer explícita esta diferenciación, esto es, disponer de 5 DSL diferentes, uno para cada modelo o nivel de abstracción, el coste y la complejidad de la implementación serían mínimos y asumibles.

Por un lado, para definir la sintaxis abstracta de cada DSL se partiría del meta-modelo común que define la del actual y se generarían 5 versiones en las que únicamente habría que eliminar elementos.

Por otro lado, para que los modelos elaborados con el DSL actual pudieran seguir gestionándose con los nuevos DSL, bastaría con desarrollar 4 transformaciones de modelos que mapearían los modelos elaborados con los DSL originales primeros a cada uno de los 4 DSL intermedios (Contexto, CIM, PIM y PSM). Estás transformaciones serían además extremadamente sencillas, ya que las reglas de correspondencia son en todos los casos 1:1 porque los elementos del meta-modelo origen y los meta-modelos destino serían exactamente los mismos.

Por otro lado, la decisión de utilizar GMF como marco para la construcción de los diagramadores o editores gráficos de la herramienta tiene alguna implicación menor sobre la definición de la sintaxis abstracta. En realidad, para no añadir complejidad accidental al desarrollo de un diagramador con GMF, es recomendable seguir ciertas buenas prácticas.

La más destacable es probablemente el hecho de que conviene que cualquier meta-modelo que vaya ser utilizado como punto de partida para la construcción de un diagramador incluya como nodo raíz una meta-clase que, cuando se instancie en un modelo concreto, permita representar el propio diagrama.

Así, tal y como muestra la Figura 5.4. El meta-modelo C-SMArT incluye la meta-clase **ESPDiagram**, que es el nodo raíz del resto del meta-modelo.

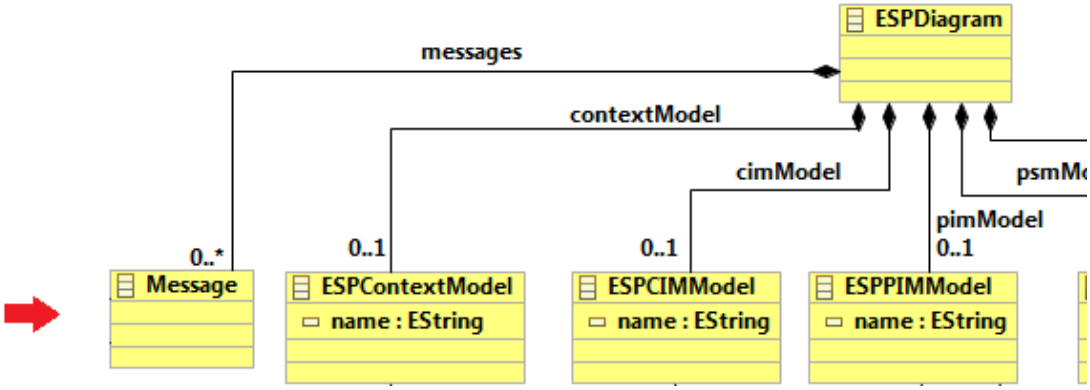


Figura 5.4. Vista parcial del meta-modelo C-SMArT: meta-clase para representar el diagrama

De esta manera, cualquier modelo C-SMArT será un diagrama que contendrá un objeto de tipo *ESPContextModel* / *ESPCIMModel* / *ESPSMModel* / *ESPIMModel* / *ESPDMModel*) y una sería de objetos de tipo *Message*. De manera que el objeto que representa el modelo contendrá a su vez el resto de objetos del modelo.

5.3.2 Sintaxis Concreta

En esta sección se resumen algunas de las consideraciones reseñables que dirigieron la construcción de los diagramadores incluidos en la herramienta que proporciona soporte tecnológico a la propuesta metodológica de la presente Tesis Doctoral. En general, se trata de funcionalidades que resultan particularmente complejas de soportar cuando se utiliza GMF, y para las que por tanto fue necesario un esfuerzo extra en términos de diseño e implementación.

En primer lugar, se pretendió maximizar la usabilidad del diagramador, de manera que su funcionamiento resultase intuitivo para usuarios/diseñadores no familiarizados con el entorno Eclipse.

Por ejemplo, en el entorno Eclipse, es habitual que cualquier personalización de un objeto de dibujo, o la invocación de cualquier acción sobre cualquier otro tipo de objeto, como un

fichero o carpeta, pase por la utilización del panel de propiedades, asociado a la opción *Show Properties* del menú contextual (véase el cuadro de propiedades mostrado en la parte derecha de la Figura 5.5). Para evitar que el usuario tenga que invocar el menú contextual y luego localizar la opción más conveniente para la acción que desea ejecutar, se ha modificado este comportamiento habitual. Así, un doble clic sobre cualquier objeto de dibujo del diagrama, mostrará automáticamente el cuadro de propiedades de dicho objeto, resultando en un comportamiento mucho más intuitivo.

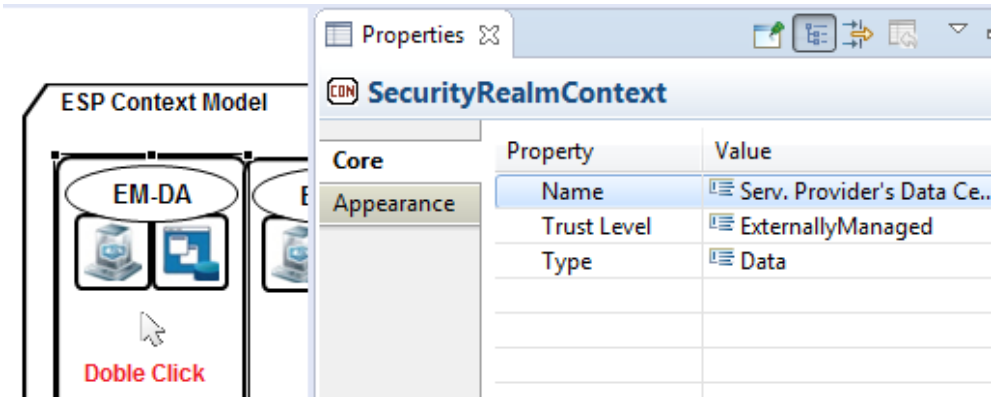


Figura 5.5. Cuadro de propiedades del objeto *SecurityRealmContext*

Desde el punto de vista del desarrollo con GMF, una de las características más destacables del diagramador es el comportamiento dinámico de algunos objetos de dibujo, no soportado de forma nativa por GMF, lo que ha obligado a implementar una serie de manejadores de eventos para integrar en el editor esas capacidades de gestión de eventos que GMF no proporciona. El resultado es ofrecer figuras con apariencia dinámica, capaces de responder en tiempo real a las modificaciones que el usuario realice sobre el modelo subyacente.

Volviendo al ejemplo de la Figura 5.5, cuando el usuario utiliza el cuadro de propiedades para cambiar el valor de la propiedad *TrustLevel* o la propiedad *Type* del objeto *SecurityRealmContext*, la imagen mostrada dentro del recuadro interior izquierdo (respectivamente derecho) cambia automáticamente, mostrándose en su lugar la figura asociada al nuevo valor asignado a la propiedad.

Soportar esta funcionalidad para los diferentes elementos de modelado ha supuesto codificar un gran número de métodos auxiliares para navegar por los diferentes *Canvas* que representan cada elemento del modelo y transmitir eventos (como el cambio de valor de una propiedad) de unos a otros.

Otro aspecto muy destacable del diagramador es que soporta figuras con varios niveles de profundidad o anidamiento, frente al comportamiento nativo de GMF, que sólo soporta 1 nivel de profundidad. Para ello ha sido preciso modificar las relaciones de mapeo que GMF identifica en el modelo *.gmfmap*, indicando explícitamente que, en el caso de ciertos elementos, la posibilidad de anidar objetos, hacía referencia a múltiples anidamientos, y no sólo a un único nivel.

Por ejemplo, tal y como muestra la parte izquierda de la Figura 5.6, el meta-modelo del DSL recoge que un objeto del tipo *TechnologicalProduct* puede estar compuesto a su vez por uno o varios objetos de tipo *TechnologicalProduct*. Conceptualmente, deberían permitirse por tanto anidamientos múltiples, como el que se muestra en la parte derecha de la figura. Sin embargo, GMF no soporta este comportamiento por defecto, y es preciso refinar el proceso de mapeo que implementa GMF para que las figuras del diagramador (que deban hacerlo) soporten efectivamente el anidamiento múltiple.

Por último, otra de las funcionalidades destacadas del diagramador, no soportada de forma nativa por GMF, es el redimensionamiento automático de figuras.

Todos los objetos de dibujo en GMF se implementan internamente como un objeto del tipo *DefaultSizeNodeFigure* (que permanece oculto), y como objeto hijo de este objeto raíz se representa la figura especificada por el usuario. Por ejemplo, la parte izquierda de la Figura 5.7 muestra el objeto del tipo *DefaultSizeNodeFigure* que GMF utiliza para manejar internamente un objeto de tipo *SecurityRealm* que el editor muestra como un rectángulo negro redondeado.

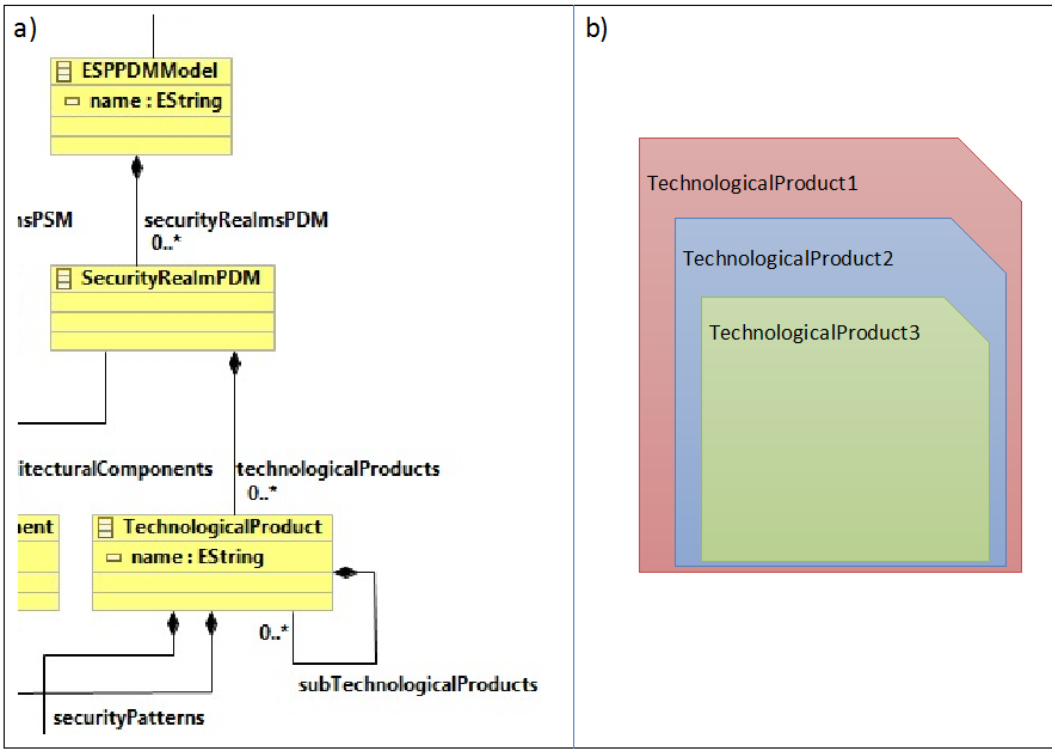


Figura 5.6. a) Vista parcial del meta-modelo C-SMArT; b) Anidamiento de 3 objetos de tipo *TechnologicalProduct*

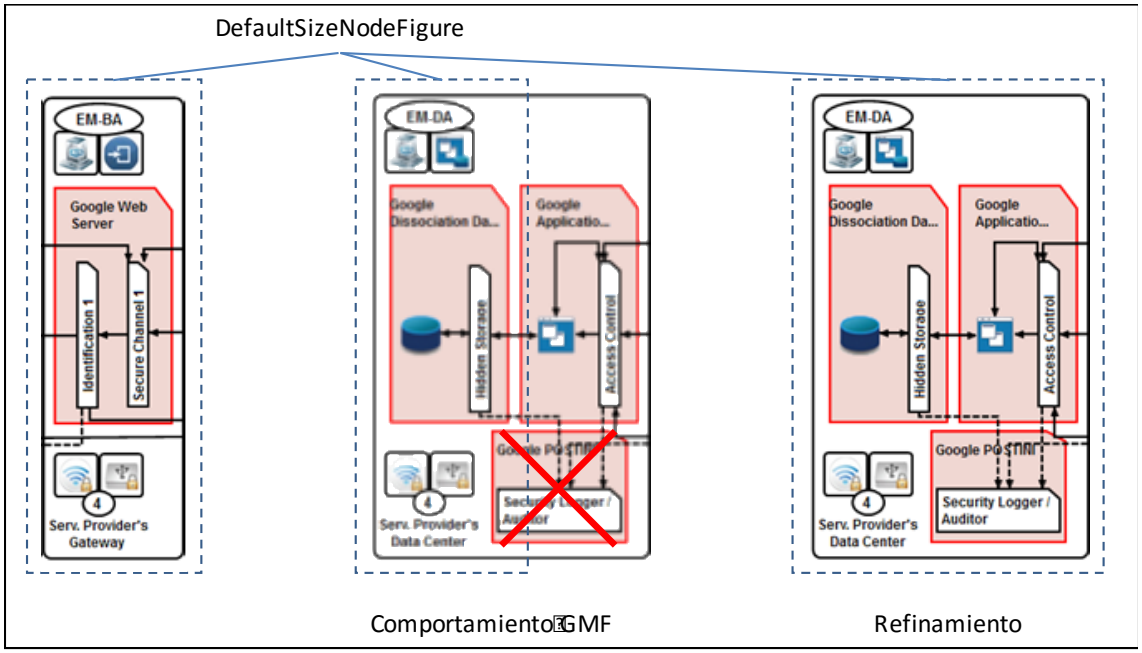


Figura 5.7. Soportando el redimensionamiento automático de figuras en GMF

Si el usuario decide redimensionar un objeto *SecurityRealm*, redimensionando dicho rectángulo redondeado, el comportamiento por defecto de GMF no es capaz de reflejar internamente ese redimensionamiento, con lo que el objeto *DefaultSizeNodeFigure* seguiría teniendo exactamente las mismas dimensiones iniciales (*Preferred Size*), tal y como muestra la parte central de la figura. En tal caso, no se podría añadir el objeto de tipo *TechnicalProduct* (Google POSTINI) inferior.

Por el contrario, se ha refinado el código generado por GMF para que el redimensionamiento de la figura visible se transmita automáticamente al objeto interno que maneja GMF. Así, el resultado del redimensionamiento en el diagramador desarrollado es el que se muestra en la parte derecha de la figura: el objeto interno *DefaultSizeNodeFigure* ha sido redimensionado y podemos, por ejemplo, añadir nuevos objetos, como el objeto Google POSTINI, al objeto de tipo *SecurityRealm*.

6. Caso de Estudio

En este capítulo se presenta la aplicación de los *Enterprise Security Patterns* siguiendo el método de Investigación-Acción que se detalla en el capítulo 2. En primer lugar, se realiza una breve descripción de la Organización en la cual se ha aplicado el caso de estudio. A continuación, se presenta el problema que se pretende solucionar y la aplicación de un nuevo *Enterprise Security Pattern* definido para solucionar el problema. Finalmente, se muestran unas reflexiones sobre las lecciones aprendidas del caso de estudio realizado.

6.1 Descripción de la Organización

Como muestra el capítulo 2, la organización del caso de estudio ha sido formada por el doctorando, el beneficiario (Grupo BBVA) y el grupo critico de referencia, es decir, el Grupo de Seguridad y Auditoria (GSyA) de la Universidad de Castilla La Mancha, y el grupo Kybele de la Universidad Rey Juan Carlos.

El Grupo BBVA se hizo eco de las actividades que se estaban realizando en esta tesis doctoral, y a través del Centro de Investigación para la Gestion Tecnológica del Riesgo (CIGTR), nos propuso trabajar en conjunto para poner a prueba los *Enterprise Security Patterns* y buscar una solución a un problema general que el grupo BBVA necesitaba analizar.

Toda relación entre el doctorando y el Grupo BBVA fue avalada y consensuada con el CIGRT y la Fundación Universidad Rey Juan Carlos, por lo que el propio centro de investigación podría también considerarse como uno de los beneficiarios del caso de estudio.

6.2 Descripción del Problema

En las últimas décadas, las arquitecturas de correo electrónico y aplicaciones de colaboración estaban alojadas normalmente dentro del entorno productivo de las organizaciones. Los empleados tenían capacidad de acceder a estas aplicaciones desde fuera de la oficina y esto provocaba una serie de riesgos que las organizaciones tenían que gestionar o asumir. A continuación, mostramos algunos de los riesgos relacionados con la confidencialidad de los datos en estas aplicaciones y como las organizaciones previenen cada uno de ellos:

- ✓ Un atacante podría interceptar los datos de los empleados mientras viajaban por Internet. Para prevenir este riesgo, las organizaciones tienen la necesidad de cifrar la comunicación entre el empleado y el centro de datos.
- ✓ Un atacante podría robar las credenciales de acceso de un empleado y suplantar su identidad. Para prevenir este riesgo, las organizaciones tienen la necesidad de asegurar que el empleado es quien dice ser. Normalmente se utilizan mecanismos de autenticación avanzados como puede ser el doble factor.
- ✓ Un atacante puede tomar ventaja de una vulnerabilidad y acceder a los datos del empleado. Para prevenir este riesgo, las organizaciones necesitan un proceso de gestión de parches continuo, teniendo frecuencias de parcheo muy bajas en los sistemas expuestos a Internet.
- ✓ Un usuario técnico encargado del mantenimiento de los sistemas puede acceder a la información de los empleados. Para prevenir este riesgo, las organizaciones necesitan mantener cifrada la información almacenada y monitorizar la actividad de usuarios privilegiados en la red interna.

Con el nacimiento del *Cloud Computing* y el *Software as a Service* (SaaS), la forma de acceso a los aplicativos cambia significativamente. Los riesgos que las organizaciones tienen que gestionar en este nuevo contexto son similares a los entornos clásicos, pero la forma de

gestionarlos es diferente, ya que se introducen nuevos actores y parte de la seguridad reside en los sistemas que está proveyendo un tercero.

El Centro de Investigación CIGTR liderado por el grupo BBVA nos invitó a realizar un caso de estudio para ayudar a solucionar este problema que se les estaba presentando. BBVA estaba mirando opciones para migrar el correo electrónico de todos sus empleados a Google u Office 365 y querían estandarizar la arquitectura de seguridad para todos los bancos del grupo. En la siguiente sección mostramos el Enterprise Security Pattern que fue diseñado junto con los ingenieros de seguridad del grupo BBVA para solucionar este problema.

6.3 Enterprise Security Pattern: Secure Software as a Service

En esta sección se documenta un nuevo Enterprise Security Pattern que podría ser usado por las organizaciones a la hora de proteger la confidencialidad de los activos de información alojados en aplicaciones online externalizadas, como por ejemplo el correo electrónico, las herramientas colaborativas, las herramientas de gestión de proyectos, etc. A continuación, mostramos cada uno de los elementos incluidos en el patrón (*véase Sección 4.3 Plantilla para Documentar Enterprise Security Patterns*).

6.3.1 Contexto

Los empleados de una organización necesitan acceder desde su casa (*Dominio Público de Empleados*, *P-E*) a las aplicaciones online alojadas en el centro de datos de una compañía externa (*Dominio Externamente Gestionado de Datos*, *EM-D*) utilizando su conexión de Internet (*Dominio Público de Transporte*, *P-T*). La siguiente figura muestra el diagrama del contexto configurado para esta primera aproximación.

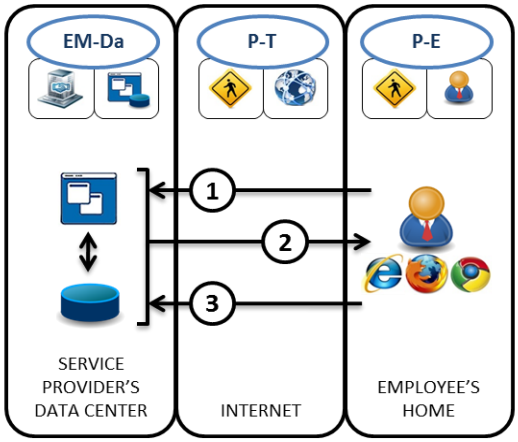


Figura 6.1. Diagrama de Contexto

6.3.2 Problema

El problema que intenta solucionar este patrón ha sido documentado en la sección 0, dentro de la descripción del problema del caso de estudio. Como se comentaba anteriormente, BBVA estaba mirando opciones para migrar el correo electrónico de todos sus empleados a Google u Office 365 y querían entender los riesgos a los que se enfrentaban antes de tomar una decisión sobre la iniciativa.

6.3.3 Incidentes Conocidos

Actualmente existen muchos incidentes causados por robos de identidad o suplantación de identidad. El principal objetivo de estos robos es obtener información relevante que pueda ser monetizada o que cause un impacto reputacional sobre la persona o la compañía atacada.

Uno de los incidentes más notorios fue cuando un grupo de hackers robo los credenciales de acceso de la cuenta de Twitter de Fox News y publicaron que el presidente Barack Obama había fallecido (GIZMODO, http://gizmodo.com).

En Diciembre del 2006, Julian Assange fundo WikiLeaks (Assange, https://wikileaks.org/) una organización mediática internacional sin ánimo de lucro, que publica a través de su sitio web informes anónimos y documentos filtrados con contenido sensible en materia de interés público, preservando el anonimato de sus fuentes. WikiLeaks ha realizado una gran diversidad de publicaciones relacionadas con los servicios de Inteligencia de las primeras potencias mundiales, la economía mundial, política internacional e información confidencial sobre importantes organizaciones.

Todas las compañías están expuestas a este tipo de robos. Utilizando el Enterprise Security Pattern que estamos definiendo en esta sección las compañías podrían evitar que los hackers robaran información del correo electrónico de los empleados, incluso si roban sus credenciales. Esto es debido a que la solución proporciona una fuerte validación para asegurar que el empleado es quien dice ser.

6.3.4 Solución

A continuación, se muestra una descripción detallada de los cuatro modelos incluidos en la solución de este nuevo Enterprise Security Patterns:

- ✓ Modelo Independiente de la Computación (CIM)
- ✓ Modelo Independiente de la Plataforma (PIM)
- ✓ Modelo Específico de la Plataforma (PSM)
- ✓ Modelo Dependiente de Producto (PDM)

6.3.4.1 Modelo Independiente de la Computación (CIM)

Este modelo proporciona una descripción de las políticas de seguridad que el sistema debe cumplir independientemente de sus características funcionales y tecnológicas. Para definir el modelo CIM hemos aplicado las decisiones de diseño sobre el contexto actual (Tabla 6-1) y posteriormente hemos incluido el registro de sensibilidad asociado a cada uno de los dominios definidos (Tabla 6-2).

ID	Decisiones de Diseño Contexto – CIM
CON-CIM-1	Entre un Dominio Público (NO Gestionado) y un Dominio Gestionado siempre hay una red Bastión (DMZ).
CON-CIM-2	Siempre que la información a proteger este fuera de la organización (Dominio Público o Gestionado Externamente), la Autenticación se realiza dentro de los sistemas de la Organización.

Tabla 6-1. Decisiones de Diseño entre el Modelo de Contexto y el CIM

Aplicación de las Decisiones de Diseño

La Figura 6.2 muestra la aplicación de la decisión de diseño CON-CIM-1. La parte izquierda de la imagen muestra el contexto definido en la sección anterior. La primera flecha (1) simula la solicitud inicial del empleado para acceder al correo electrónico. La segunda flecha (2) simula el envío del formulario para proporcionar las credenciales de acceso al correo. La tercera flecha (3) simula el envío de las credenciales por parte del empleado. Así la empresa externa puede validar si el empleado es quien dice ser.

La parte derecha de la imagen muestra la transición entre el Modelo de Contexto y el CIM, aplicando la decisión de diseño CON-CIM-1. Se puede observar como aparece un dominio Bastión (*EM-B*) entre el empleado y el centro de datos de la empresa externa.

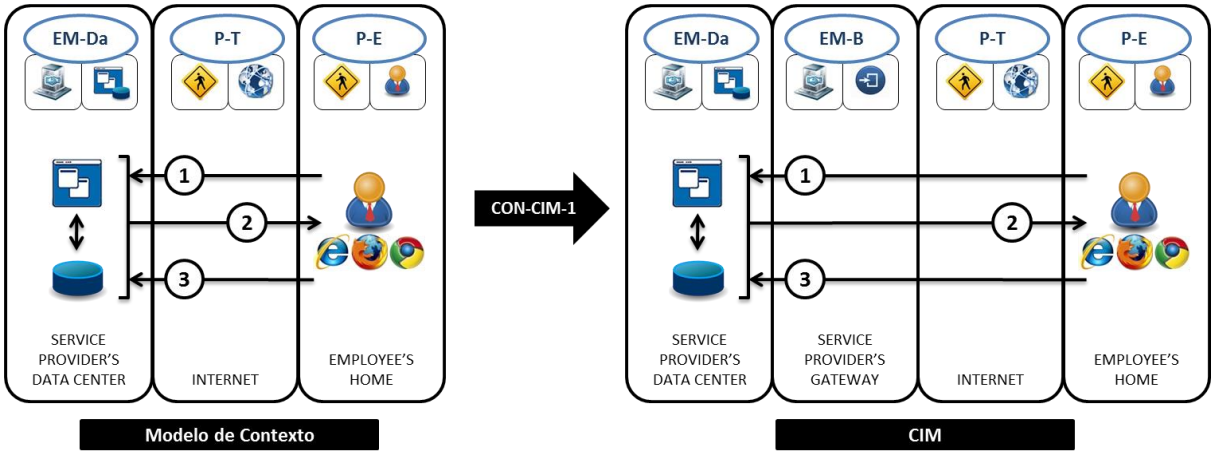


Figura 6.2. Modelo CIM - Aplicación de la Decisión de Diseño CON-CIM-1

El diagrama CIM mostrado en la imagen anterior no es el diagrama CIM final asociado al modelo de contexto. En este caso particular, también aplica la decisión de diseño CON-CIM-2 (*Siempre que la información a proteger este fuera de la organización, la Autenticación se realiza dentro de los sistemas de la Organización*).

La Figura 6.3 muestra la aplicación de la decisión de diseño CON-CIM-1 y CON-CIM-2, conjuntamente.

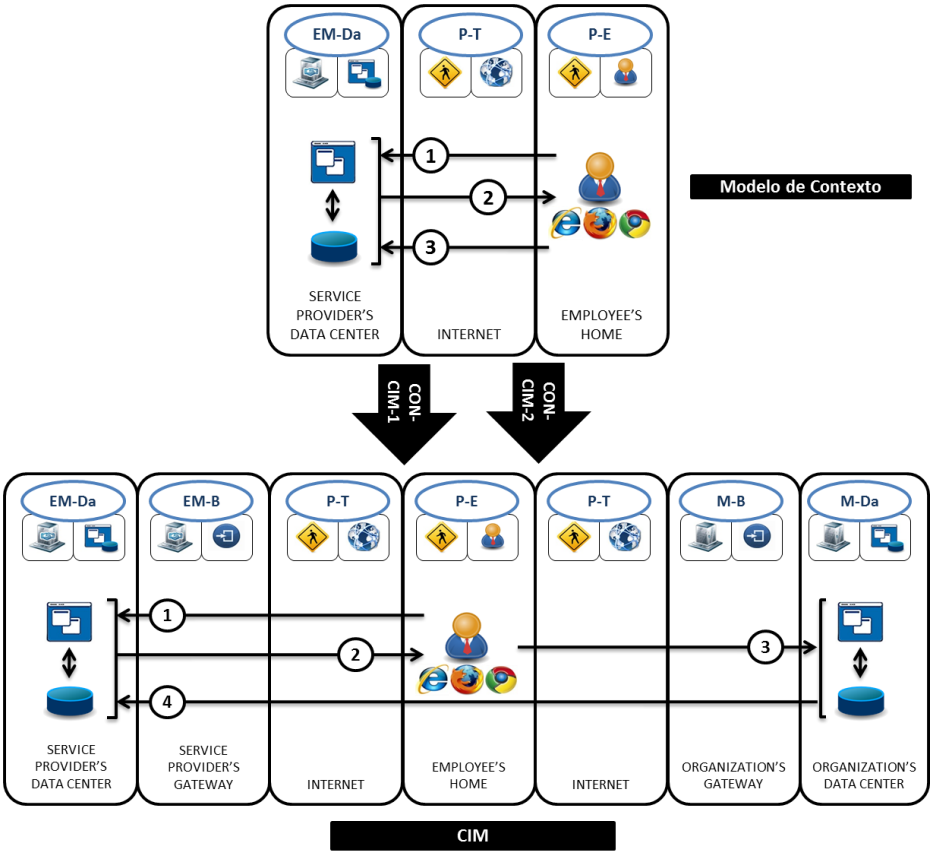


Figura 6.3. Modelo CIM - Aplicación de las Decisiones de Diseño CON-CIM-1 y CON-CIM-2

Como se puede comprobar en la figura anterior, además de aparecer un dominio Bastión en la empresa externa (*dominio EM-B*), el nuevo CIM incluye un dominio de Datos gestionado (*dominio M-Da*) y un dominio Bastión gestionado (*dominio M-B*).

A la hora de aplicar las decisiones de diseño se han modificado algunos de los mensajes intercambiados. La primera flecha (1) y la segunda flecha (2) simulan la solicitud inicial del empleado para acceder al correo electrónico y el envío del formulario para proporcionar las credenciales de acceso al correo, al igual que lo hacían en el Modelo del Contexto. La tercera flecha (3) simula el envío de las credenciales por parte del empleado hacia los sistemas de autenticación alojados dentro de la organización. La cuarta flecha (4) simula el envío de las credenciales de acceso desde la organización hasta la empresa externa. En este caso, la organización valida que el empleado es quien dice ser y la empresa externa otorga el acceso al empleado validado.

Registro de Sensibilidad

La siguiente tabla muestra el nivel de sensibilidad (NL) y las políticas de seguridad de cada uno de los dominios incluidos en el contexto de la solución.

NL	Dominio	Políticas de Seguridad
4	Externamente Gestionados – Datos (*EM-Da*)	Canal Seguro (*CS*) y Almacenamiento Seguro (*AS*)
4	Externamente Gestionado – Bastión (*EM-B*)	Canal Seguro (*CS*) y Almacenamiento Seguro (*AS*)
4	Publico – Transporte (*P-T*)	Canal Seguro (*CS*) y Almacenamiento Seguro (*AS*)
3	Publico – Empleado (*P-E*)	Canal Seguro (*CS*) y Almacenamiento Claro (*AC*)
4	Publico – Transporte (*P-T*)	Canal Seguro (*CS*) y Almacenamiento Seguro (*AS*)
4	Gestionado – Bastión (*M-B*)	Canal Seguro (*CS*) y Almacenamiento Seguro (*AS*)
1	Gestionado – Datos (*M-Da*)	Canal Claro (*CC*) y Almacenamiento Claro (*AC*)

Tabla 6-2. Registro de Sensibilidad del Contexto

La Figura 6.4 muestra el diagrama del modelo CIM después de aplicar las decisiones de diseño sobre el modelo de Contexto e incluir el registro de sensibilidad. Este modelo será utilizado para aplicar las decisiones de diseño en el modelo PIM.

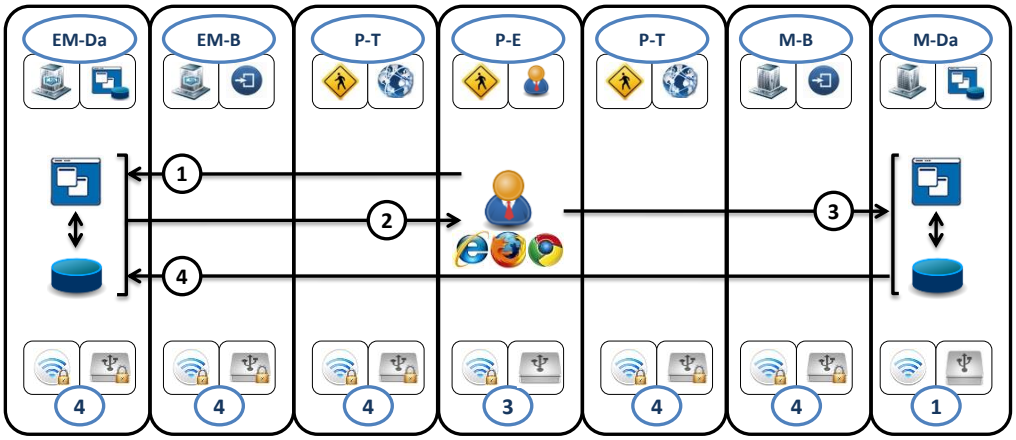


Figura 6.4. Modelo Independiente de la Computación (CIM)

6.3.4.2 Modelo Independiente de la Plataforma (PIM)

El modelo PIM ofrece una descripción conceptual de los mecanismos de seguridad que se deben incorporar en el sistema y las relaciones que existen entre ellos, con independencia de sus características tecnológicas y el detalle de su implementación. Una buena guía que puede ser utilizada como base para seleccionar los patrones de seguridad son las directrices elaboradas en (Schumacher et al., 2006) y (Fernandez, 2013).

Para crear el modelo PIM hemos aplicado las decisiones de diseño sobre el modelo CIM documentado en la sección anterior. A continuación, mostramos la Tabla 6-3 con las decisiones de diseño que aplican a este modelo.

ID	Decisiones de Diseño CIM – PIM
CIM-PIM-1	En el Destino de las Comunicaciones Seguras debe aparecer un patrón *Secure Channel* en el último dominio con sensibilidad 5, 4, o 3 en la dirección del mensaje.
CIM-PIM-2	Todo acceso desde un dominio público (No Gestionado) hacia la red Bastión implica un proceso de Autorización (tripleta de *Identificación*, *Autenticación* y *Control de Acceso*).
CIM-PIM-3	Los patrones de *Autenticación* siempre deben estar en dominios Gestionados que no sean el dominio de Bastión, Transporte o Usuarios.
CIM-PIM-4	Si la información no puede quedar alojada en el Dominio del Usuario es necesario incluir un *Motor de Virtualización* y la tripleta de Autorización (*Identificación, Autenticación y Control de Acceso*) dentro del Dominio de Datos de la organización.
CIM-PIM-5	Todo acceso a la información y/o aplicación que aloja la información necesita un proceso de Autorización (tripleta de *Identificación, Autenticación* y *Control de Acceso*).
CIM-PIM-6	Todo Dominio Gestionado o externamente Gestionado debe tener un sistema de Registro de Actividad de TODOS los elementos de seguridad.

Tabla 6-3. Decisiones de Diseño entre el Modelo CIM y PIM

Analizando la primera decisión de diseño (*En el Destino de las Comunicaciones Seguras debe aparecer un patrón Secure Channel en el último dominio con sensibilidad 5, 4, o 3 en la dirección del mensaje*) nos damos cuenta que necesitamos incluir un patrón Secure Channel en el dominio del empleado y otro en el dominio de Bastión de la organización.

Esto es porque en la dirección del primer mensaje no hay dominios que permitan ninguna comunicación no segura, es decir, no hay ningún dominio con nivel de sensibilidad 1 o 2. En la dirección del tercer mensaje el último dominio que permite comunicación segura es el Bastion de la organización. La Figura 6.5 muestra la inclusión del patrón *Secure Channel* dentro del modelo PIM.

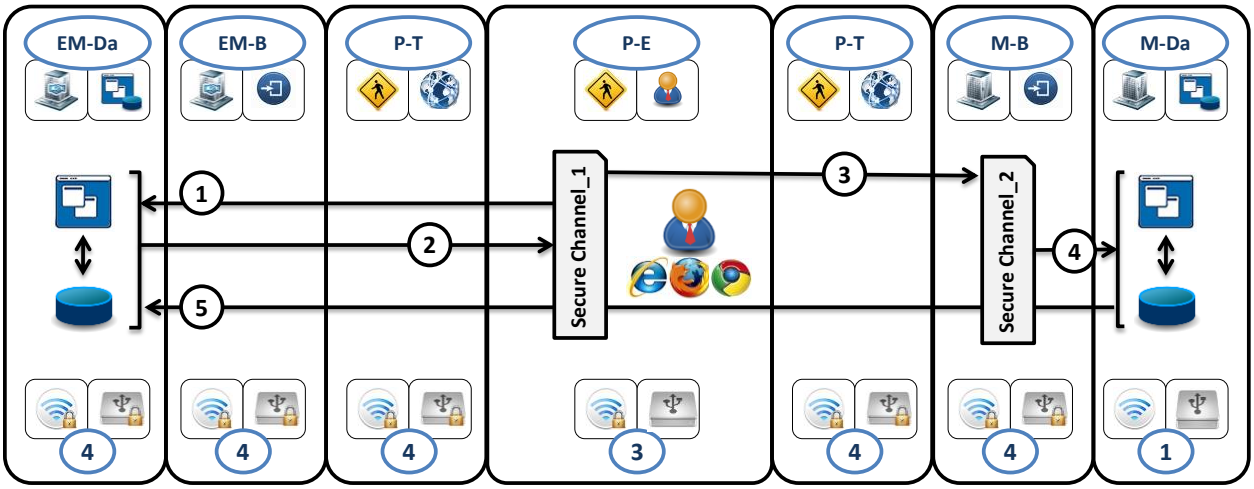


Figura 6.5. Modelo PIM - Aplicación de las Decisiones de Diseño 1

Analizando la decisión de diseño CIM-PIM-2 (*Todo acceso desde un dominio público hacia la red Bastión implica un proceso de Autorización, es decir la tripleta de Identificación, Autenticación y Control de Acceso*) vemos la necesidad de incluir un patrón de seguridad de Identificación en el dominio Bastión de la empresa proveedora del servicio y en el dominio Bastión de la Compañía.

Antes de posicionar en el modelo el patrón de seguridad de *Authentication* evaluamos la decisión de diseño CIM-PIM-3 (Los patrones de Autenticación siempre deben estar en dominios Gestionados que no sean el dominio de Bastión, Transporte o Usuarios) y CON-CIM-2 (Siempre que la información a proteger este fuera de la organización, la Autenticación se realiza dentro de los sistemas de la Organización). Basándonos en estas dos decisiones el patrón de *Authentication* tiene que estar alojado dentro del dominio de datos de la organización.

Al contrario, el patrón de seguridad de Access Control debe estar lo más cerca posible de los aplicativos o datos a acceder. Por lo que hemos incluido el patrón de Access Control dentro del dominio de datos de la empresa proveedora del servicio.

La Figura 6.6 muestra el modelo PIM con las decisiones de diseño aplicadas hasta el momento.

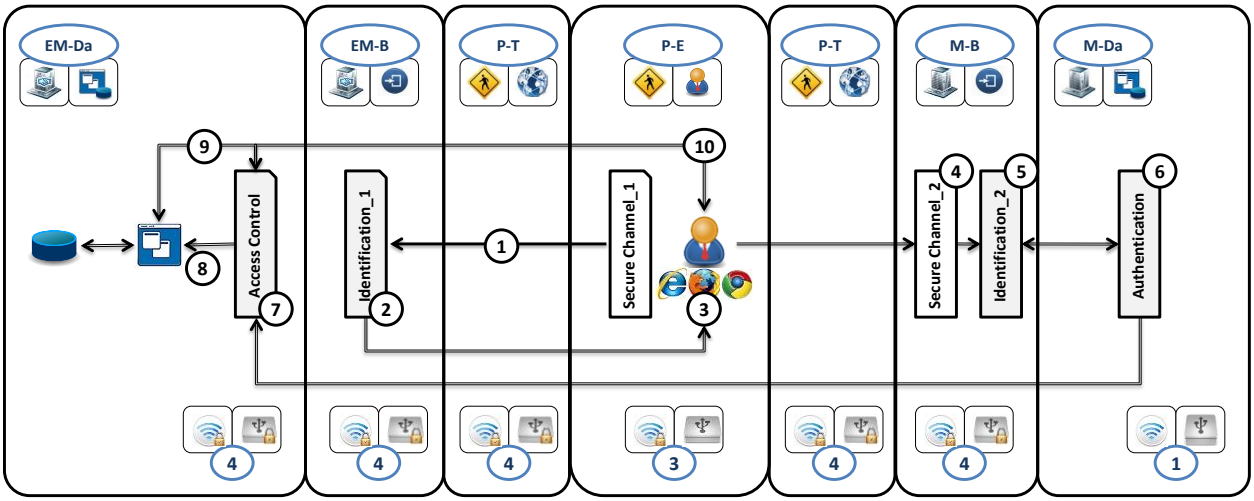


Figura 6.6. Modelo PIM - Aplicación de las Decisiones de Diseño 2

La primera flecha (1) y la segunda flecha (2) simulan la solicitud inicial del empleado para acceder al correo electrónico y el envío del formulario para proporcionar las credenciales de acceso al correo. El tercer ítem (3) simula la inclusión de las credenciales dentro del formulario de autenticación. La cuarta flecha (4) simula el envío seguro de las credenciales por parte del empleado. La quinta flecha (5) simula la validación de las credenciales dentro de los sistemas de autenticación de la organización. Si los credenciales no fueran correctos el flujo de conexión terminaría en este punto. La sexta flecha (6) simula la notificación a la empresa proveedora de que las credenciales introducidas han sido autenticados con éxito. El séptimo ítem (7) simula la revisión de acceso de las credenciales proporcionadas al recurso solicitado. Si las credenciales no tuvieran acceso el flujo de conexión terminaría en este punto. La octava flecha (8) simula la notificación del control de acceso al aplicativo indicando que el usuario tiene permiso de acceso al recurso solicitado. La novena (9) y la décima flecha (10) simulan la conexión entre el empleado y los aplicativos alojados en el centro de datos del proveedor de servicios.

Analizando el resto de decisiones de diseño entre el CIM y el PIM, vemos que también tenemos aplicar la decisión de diseño CIM-PIM-6 (*Todo Dominio Gestionado o externamente Gestionado debe tener un sistema de Registro de Actividad de TODOS los elementos de seguridad*). La Figura 6.7 muestra en gris la inclusión de dos patrones *Security Logger / Auditor*, uno para el dominio gestionado y otro para el dominio externamente gestionado. También se han incluido mensajes de registro de logs entre cada uno de los patrones y el patrón *Security Logger*.

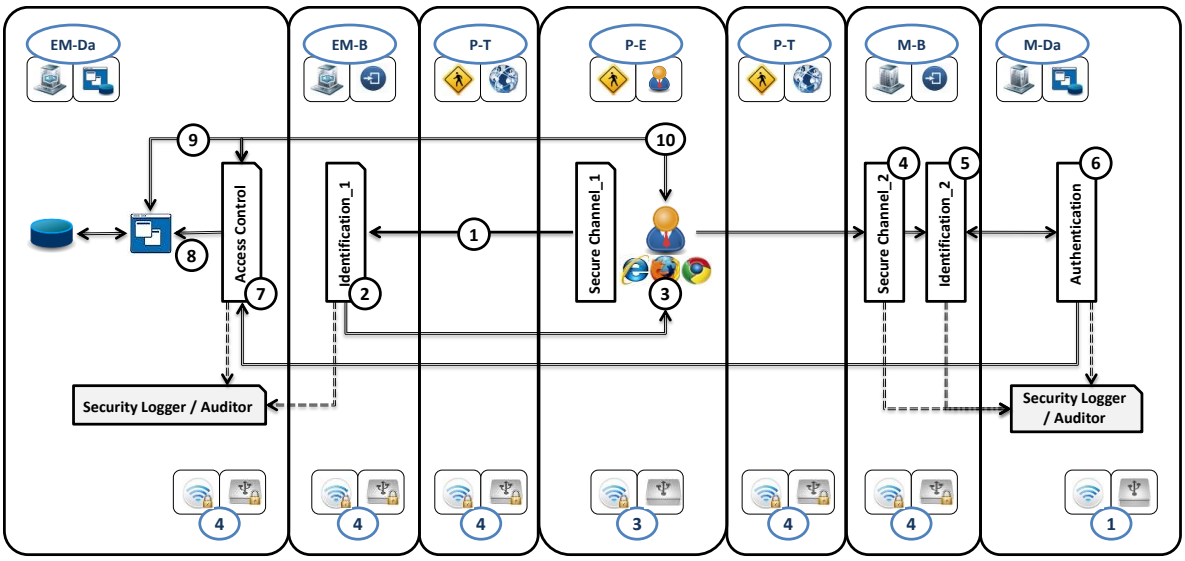


Figura 6.7. Modelo PIM - Aplicación de las Decisiones de Diseño 3

Una vez hemos acabado de revisar las decisiones de diseño, pasamos a revisar que se cumplan las políticas de seguridad incluidas en cada uno de los dominios. Podemos observar que todos los dominios donde la comunicación debe ser segura tienen canales de comunicación cifrados. A la hora de revisar los dominios donde el almacenamiento debe ser cifrado vemos que todos los dominios son de transito de información a excepción del dominio de Datos Externamente Gestionado (EM-Da). En este dominio los datos deben ser almacenados de forma segura. En la Figura 6.8 hemos añadido un patrón *Secure / Hidden Storage* entre la aplicación y los datos con el objetivo de mantener la seguridad de los datos almacenados en el centro de datos del proveedor de servicio.

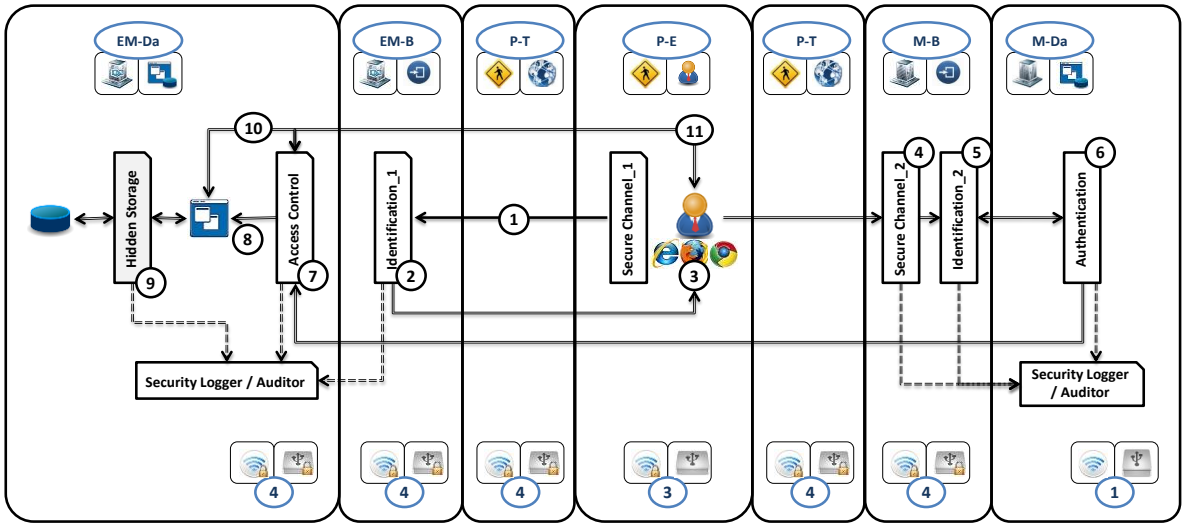


Figura 6.8. Modelo Independiente de la Plataforma (PIM)

6.3.4.3 Modelo Específico de la Plataforma (PSM)

Este modelo define los componentes arquitectónicos incluidos en la arquitectura de seguridad empresarial, independientemente de la tecnología utilizada para resolver el problema. El PSM debe tener en cuenta la forma de organizar los mecanismos de seguridad dentro de la arquitectura. Los patrones de seguridad descritos en el PIM se incluyen dentro de los componentes de seguridad arquitectónicos. Dos buenas guías que pueden ser usadas como base para seleccionar los componentes arquitectónicos son la ISO / IEC-27000-series (ISO, http://www.iso.org) y el Manual *IT Baseline Protection* (BSI, 2000). Para definir el modelo PSM hemos aplicado las decisiones de diseño sobre el modelo PIM documentado en la sección anterior. A continuación, mostramos la **Tabla 6-4** con las decisiones de diseño que aplican a este modelo.

ID	Decisiones de Diseño PIM – PSM
PIM-PSM-1	Un elemento arquitectónico nunca puede pertenecer a dos dominios distintos.
PIM-PSM-2	Un patrón *Security Logger* se corresponde con un *Sistema de Registro de Logs*.
PIM-PSM-3	Si encontramos en el dominio Bastión, (i) Un patrón de *Identificación*, (ii) Un patrón *Secure Channel* + un patrón de *Identificación*, o (iii) un patrón de *Secure Channel* + un patrón de *Identificación* + un patrón *Control de Acceso*, la transformación PSM podría ser un *Web Server*, un *Proxy Inverso* o una *VPN*. Si el dato no puede quedar almacenado en el dominio desde el que está accediendo el usuario, la transformación debe ser una VPN.
PIM-PSM-4	Si encontramos un patrón *Identificación* + un patrón *Control de Acceso* + un patrón de Máquina Virtual, la transformación PSM se corresponde con un Sistema de Virtualización.
PIM-PSM-5	Si encontramos *los datos* junto alguno de los elementos de la tripleta de Autorización (patrón *Identificación*, patrón *Control de Acceso* y/o patrón de Autenticación), la transformación PSM se corresponde con un Servidor de Datos. Si encontramos los datos junto a un patrón de *Almacenamiento Seguro*, la transformación PSM corresponde con un Servidor de Datos Desasociados.
PIM-PSM-6	Si encontramos *las aplicaciones* junto alguno de los elementos de la tripleta de Autorización (patrón *Identificación*, patrón *Control de Acceso* y/o patrón de Autenticación), la transformación PSM se corresponde con un Servidor de Aplicaciones.
PIM-PSM-7	Un patrón de *Autenticación* se corresponde con (i) un *Servidor de Usuario y Contraseña*, (ii) un *Servidor de Doble Factor*, (iii) un *Dispositivo de Biométricos*, o (iv) un conjunto de los tres anteriores. La transformación va a depender de la robustez que se quiera ofrecer en el proceso de autenticación.

Tabla 6-4. Decisiones de Diseño entre el Modelo PIM y el PSM

Analizando las decisiones de diseño vemos que PIM-PSM-2 (Un patrón *Security Logger* se corresponde con un *Sistema de Registro de Logs*) aplica a nuestro entorno. La Figura 6.9 muestra como los patrones *Security Logger / Audit* son embebidos dentro de un Sistema de Registro de Logs (*Logs System*). En este caso en particular la transformación es uno a uno, pero se pueden dar casos donde más de un patrón sea embebido en el mismo componente arquitectónico.

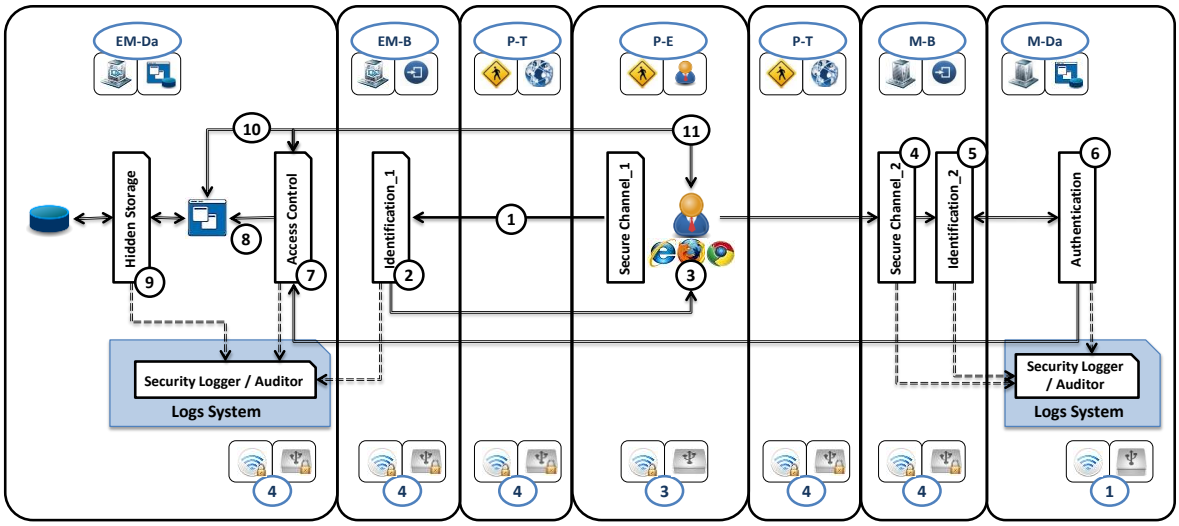


Figura 6.9. Modelo PSM - Aplicación de las Decisiones de Diseño 1

Si seguimos analizando las decisiones de diseño podemos observar que PIM-PSM-3 también aplica a nuestro entorno. Podemos observar un patrón *Identification* en el dominio Bastión del proveedor y un patrón *Secure Channel* + patrón *Identification* en el dominio Bastión de la compañía. En este caso en particular, la transformación PSM podría ser un *Web Server*, un *Proxy Inverso* o una *Virtual Private Network (VPN)*. La decisión de que componente elegir en este caso va a depender del nivel de seguridad que queremos aplicar en cada caso. Inversamente cada uno de los componentes es más caro de mantener, por lo que es importante elegir el nivel de seguridad adecuado sin necesidad de exceder las medidas necesarias.

La VPN debería siempre utilizarse en aquellos casos donde el dominio del empleado tuviera bloqueado el alojamiento de los datos accedidos. Para este caso en particular el empleado podría almacenar los datos por lo que no vemos una estricta necesidad a la hora de incluir este elemento.

En el lado del Bastión de la compañía debemos asegurar que el servicio de Identificación solo acceda a al servicio de Autenticación. En estos casos es recomendable utilizar Proxy inverso (*Reverse Proxy*), ya que es una herramienta más potente desde el punto de vista de seguridad y nos va a permitir configurar las conexiones entrantes con mayor granularidad. En el lado del Bastión del proveedor sería suficiente con un Servidor Web (*Web Server*) que recoja las

solicitudes de los clientes y las reenvíe para ser analizados en el centro de datos de la compañía. La Figura 6.10 muestra las decisiones de diseño tomadas para PIM-PSM-3.

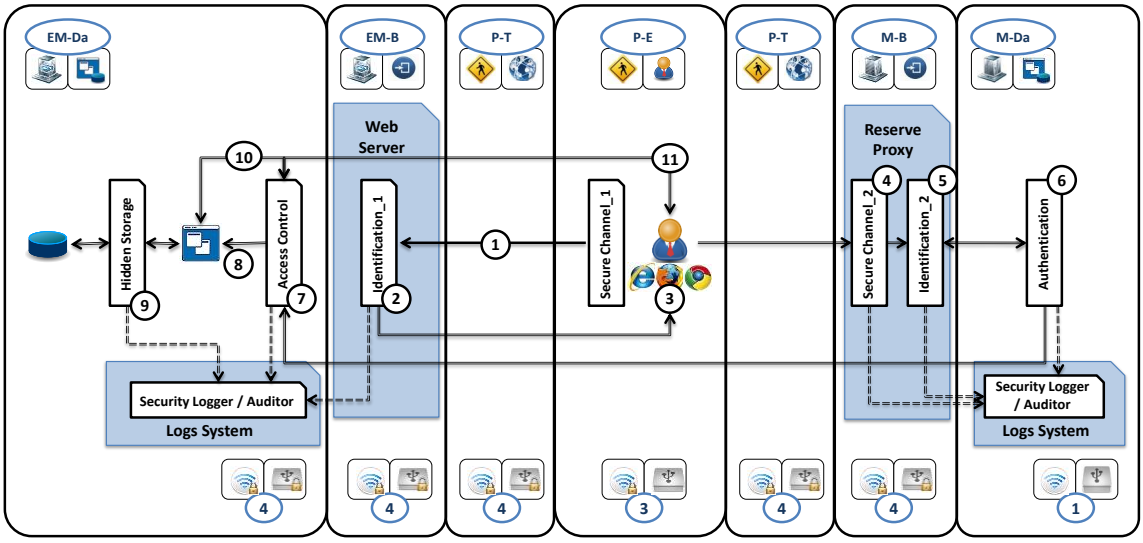


Figura 6.10. Modelo PSM - Aplicación de las Decisiones de Diseño 2

Si analizamos las decisiones de diseño asociadas al patrón de Autenticación vemos que PIM-PSM-7 hace referencia a éste: *Un patrón de Autenticación se corresponde con (i) un Servidor de Usuario y Contraseña, (ii) un Servidor de Doble Factor, (iii) un Dispositivo de Biométricos, o (iv) un conjunto de los tres anteriores.*

Esta aplicación es accedida desde cualquier parte del mundo, por lo tanto, el sistema de autenticación no puede estar basado en usuario y contraseña, ya que no seríamos capaces de evitar incidentes conocidos que han sido mostrados en la sección anterior. Para estos casos donde la aplicación está muy expuesta a Internet, debemos optar por mecanismos de autenticación fuertes que nos permitan asegurar que nadie está suplantando la identidad de algunos de nuestros usuarios.

Existen diferentes formas de autenticación basada en doble factor (Biometricos, Token, Reconocimiento Contextual, etc.). En este caso en particular hemos elegido una autenticación basada en *token*, ya que es bastante extendido su uso, pero podríamos haber elegido algún otro tipo de autenticación robusta.

La Figura 6.11 muestra las decisiones de diseño tomadas para PIM-PSM-7.

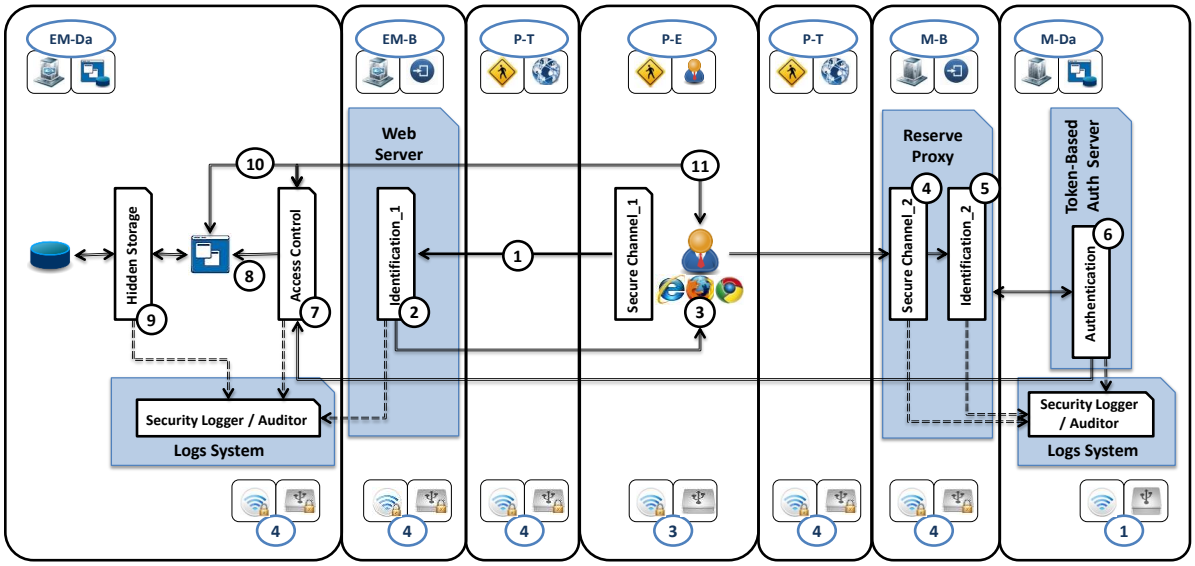


Figura 6.11. Modelo PSM - Aplicación de las Decisiones de Diseño 3

Si seguimos revisando las decisiones de diseño, podemos observar que tenemos una decisión asociada a los datos PIM-PSM-5. En este caso en particular encontramos los datos junto a un patrón de Almacenamiento Seguro (*Hidden Storage*). No hay una tripleta de Autenticación junto al dato. Por lo tanto, la transformación PSM corresponde con un Servidor de Datos Desasociados.

También podemos observar que hay una decisión de diseño asociada a las aplicaciones PIM-PSM-6. Si encontramos las aplicaciones junto alguno de los elementos de la tripleta de Autorización (patrón Identificación, patrón Control de Acceso y/o patrón de Autenticación), la transformación PSM se corresponde con un Servidor de Aplicaciones. La Figura 6.12 muestra las decisiones de diseño tomadas para PIM-PSM-5 y PIM-PSM-6.

Después de terminar de revisar todas las decisiones de diseño, podemos observar que el patrón de *Secure Channel* que está dentro del dominio del empleado sigue sin ser transformado.

Como ya decíamos anteriormente, el número de decisiones de diseño incluidas en esta tesis doctoral es un ejemplo inicial de las transiciones que hemos logrado recopilar en nuestro

estudio. Nuevas decisiones de diseño van a seguir apareciendo mientras se vayan construyendo más Enterprise Security Patterns.

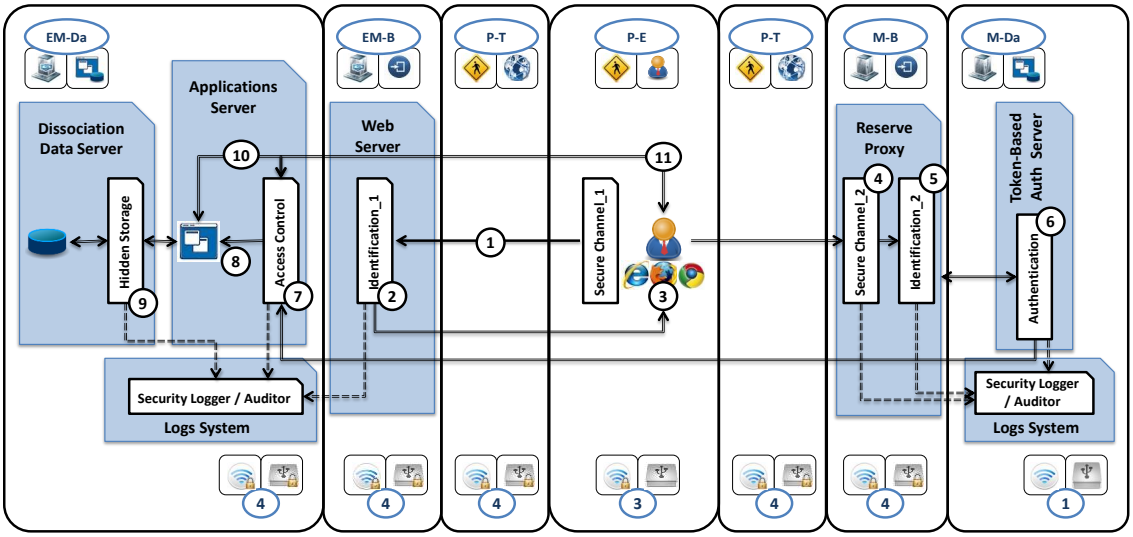


Figura 6.12. Modelo PSM - Aplicación de las Decisiones de Diseño 4

Dada esta nueva situación podríamos crear la Decisión de Diseño PIM-PSM-8: Si encontramos en el dominio de Empleado un patrón de Secure Channel, la transformación PSM podría ser un Navegador Web o una VPN. Si el dato no puede quedar almacenado en el dominio desde el que está accediendo el usuario, la transformación debería ser una VPN. En este caso en particular el dato puede quedar almacenado en el dominio del empleado por lo que la transformación PSM es un navegador Web (*Web Browser*).

La Figura 6.13 muestra el modelo Especifico de la Plataforma (PSM) después de aplicar las decisiones de diseño sobre el modelo PIM.

6.3.4.4 Modelo Dependiente de Producto (PDM)

Este modelo define los componentes tecnológicos incluidos en la arquitectura de seguridad. Un mismo PSM podría ser instanciado N veces en este modelo, ya que un mismo componente arquitectónico puede corresponder a diferentes productos tecnológicos. Los productos tecnológicos deben ser productos de buena reputación elaborados por fabricantes

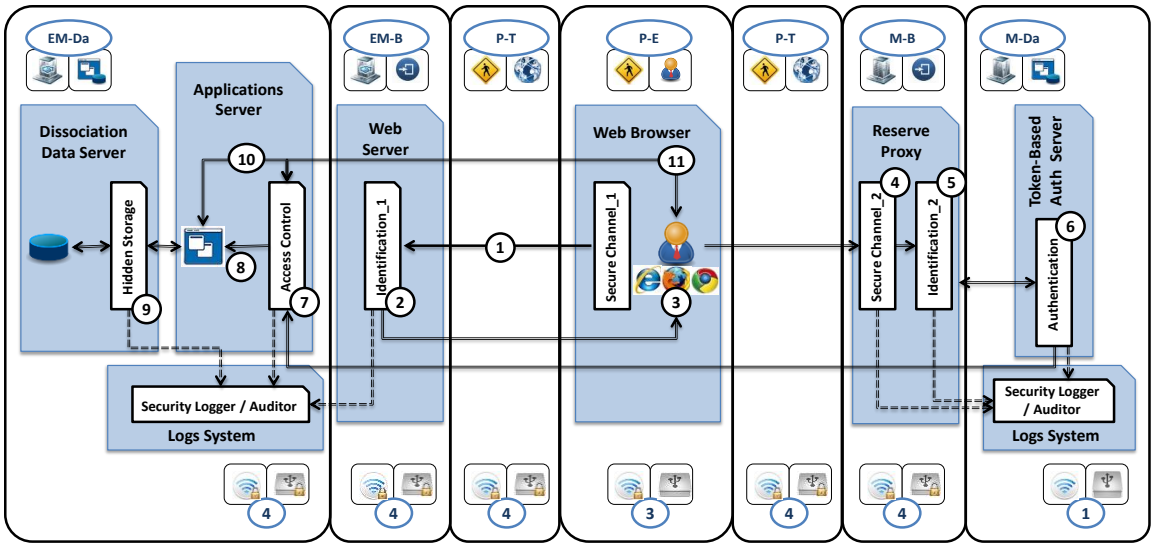


Figura 6.13. Modelo Especifico de la Plataforma (PSM)

conocidos en la industria de la seguridad. La solución final podría variar significativamente dependiendo de las tecnologías utilizadas.

Dado que el patrón ha sido creado para ayudar a analizar el riesgo de llevarse el correo de una compañía a la nube hemos elegido como empresa externa a Google. La Figura 6.14 muestra las transformaciones PDM que han sido ejecutadas para cada uno de los elementos arquitectónicos incluidos en el modelo PSM. A continuación, la lista de productos incluidos en la solución final:

- ✓ **Dissociation Data Server:** Google Dissociation Data Server
- ✓ **Application Server:** Google Application Server
- ✓ **Web Server:** Google Web Server
- ✓ **Web Browser:** Chrome Browser
- ✓ **Reverse Proxy:** IBM WebSEAL
- ✓ **Token-Based Authentication Server:** RSA Server
- ✓ **Logs System:** Splunk

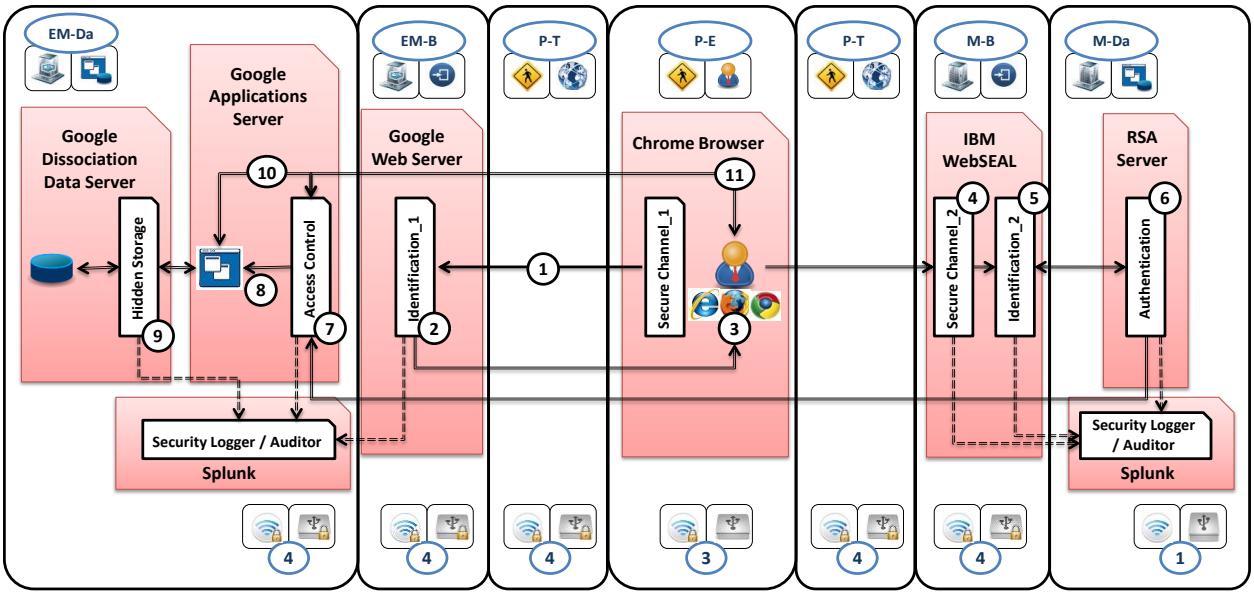


Figura 6.14. Modelo Dependiente del Producto (PDM)

6.3.5 Consideraciones

Las consideraciones mostradas para este Enterprise Security Pattern dependen principalmente de los elementos arquitectónicos incluidos en el modelo PSM y principalmente de las tecnologías seleccionadas en el modelo PDM. Si alguien tiene en mente cambiar las tecnologías incluidas en este patrón, debe considerar que otro conjunto de tecnologías podría cambiar este análisis. La Tabla 6-5 muestra los resultados del análisis para cada uno de los aspectos relevantes incluidos en las consideraciones del patrón.

Como podemos ver en la Tabla 6-5, al implementar la solución del patrón el rendimiento de la arquitectura de seguridad de la empresa no aumenta, incluso podría disminuir ya que parte de la infraestructura sería subcontratada.

El administrador de seguridad y el administrador de logs tienen que trabajar en algunos casos fuera de la organización. Este hecho puede significar un pequeño aumento de personal en el equipo de seguridad.

	Aspectos a Considerar	Análisis
Rendimiento	*Almacenamiento*	*0*
	Memoria	*0*
	Procesador	*0*
	Ancho de Banda	*0*
Complejidad	*Administrador de Seguridad*	*1*
	Administrador de Logs	*1*
	Usuario Final	*0*
	Expansión Masiva	*0*
	Administrador de Sistemas	*0*
	Coste de Instalación	*0*
	Riesgo Residual	*0*

Tabla 6-5. Consideraciones asociadas a la solución del patrón

El costo de instalación no aumenta, incluso podría disminuir debido a que la infraestructura, el personal, la implementación y el mantenimiento están subcontratados. Además, una vez desplegada la solución, el riesgo residual es mínimo. Esto significa que la solución no necesita medidas complementarias para alcanzar su objetivo inicial.

6.3.6 Consecuencias

Como ya mostramos anteriormente en la Descripción del Problema (ver sección 6.2), los riesgos encontrados están relacionadas con la confidencialidad de los activos. A continuación, se

analizan los mecanismos de seguridad incluidos en la solución del patrón, con el fin de prevenir o reducir el riesgo de las amenazas identificadas:

- ✓ Un atacante podría interceptar los datos de los empleados mientras viajaban por Internet. Para prevenir este riesgo, las comunicaciones entre el navegador web del empleado, el servidor web del proveedor de servicios y el proxy inverso de la organización utilizan canales seguros.
- ✓ Un atacante podría robar las credenciales de acceso de un empleado y suplantar su identidad. Para prevenir este riesgo, la solución proporcionada en este patrón utiliza mecanismos de autenticación avanzados como es el caso del doble factor basado en *token o* credenciales dinámicas. Aunque un atacante conociera el usuario y contraseña de un empleado, también debería tener en su posesión el hardware que proporciona el *token* o credencial dinámico.
- ✓ Un atacante puede tomar ventaja de una vulnerabilidad y acceder a los datos del empleado. Para prevenir este riesgo, las organizaciones necesitan un proceso de gestión de parches continuo, teniendo frecuencias de parcheo muy bajas en los sistemas expuestos a Internet.
- ✓ Un usuario técnico encargado del mantenimiento de los sistemas puede acceder a la información de los empleados. Para prevenir este riesgo, la solución proporcionada incluye un servidor de datos desasociados que no permite visualizar la información a los administradores, y toda la actividad de los usuarios privilegiados es monitorizada a través del sistema de logs.

Este patrón también sería aplicable en un contexto en el que los empleados accedan a sus aplicaciones externalizadas desde dentro de la organización (Dominio Gestionado de Empleados, *M-E*), en lugar de acceder desde su hogar.

6.3.7 Usos Conocidos

Google es uno de los proveedores de aplicaciones externalizadas que ofrecen las medidas de seguridad y arquitectura incluidas en la solución de este patrón.

Uno de sus productos más populares es G Suite desarrollado por Google Cloud. Actualmente, Google tiene millones de empresas utilizando sus aplicaciones (Google, https://google.com). En esa amplia lista podemos encontrar desde pequeñas empresas a empresas incluidas dentro de Fortune 500 (fortune.com, 2017) .

6.4 Lecciones Aprendidas

A continuación, mostramos algunas de las principales lecciones aprendidas durante el caso de estudio realizado:

- ✓ Una de las primeras observaciones obtenidas en el estudio, ha sido que la utilización de MDS en el proceso de desarrollo de nuevas arquitecturas de seguridad, ha sido muy bien acogido dentro del departamento de seguridad de la información del grupo BBVA. Hasta ahora los ingenieros de seguridad tomaban notas en lenguaje natural sobre las necesidades del negocio y basándose en esas notas plasmaban y construían nuevas arquitecturas tecnológicas.

  La inclusión de distintos niveles de abstracción dentro del proceso de construcción facilita bastante la labor de los ingenieros y sistematiza el resultado del análisis entre dos ingenieros distintos, es decir, es más fácil que dos ingenieros lleguen a soluciones similares usando MDS que si no lo utilizaran.

- ✓ Las decisiones de diseño incluidas en la tesis doctoral no son completas y pueden necesitar ser modificadas por factores externos. A la vez que se vayan descubriendo nuevos patrones es posible que se vayan generando nuevas decisiones de diseño.

  También hemos comprobado que factores externos, como pueden ser nuevos incidentes o nuevos riesgos podrían modificar decisiones de diseño existentes o generar nuevas decisiones. Para mitigar estos nuevos riesgos, es necesario un proceso sistemático de revisión de Enterprise Security Patterns cada vez que se identifiquen nuevos incidentes de seguridad que estén relacionados con la solución del patrón.

- ✓ Las consideraciones de los Enterprise Security Patterns dependen totalmente de las tecnologías de seguridad seleccionadas en la solución del patrón. Por ello, dependiendo de las tecnologías escogidas las consideraciones a tener en cuenta por los ingenieros pueden ser muy diferentes.

Para agilizar la construcción de nuevos Enterprise Security Patterns y facilitar el uso de estos patrones, vemos la necesidad de realizar un catálogo de tecnologías de seguridad, asociando las consideraciones de complejidad y rendimiento que el ingeniero deberá tener en cuenta si elige utilizar esa tecnología en particular.

- ✓ Aunque en este caso de estudio no hemos validado el proceso de minería de Enterprise de Security Patterns, debido a que la empresa beneficiaria ya tenía un problema a resolver, vemos la necesidad de validar el proceso de minería de Enterprise Security Patterns presentado en el capítulo 4.

  En el tiempo que hemos estado junto los ingenieros de seguridad del grupo BBVA, para la mayoría de los nuevos proyectos que recibían no tenían una arquitectura de seguridad validada que pudieran reutilizar para dar una solución. Prevemos que el trabajo de minería alrededor a estos patrones va a ser muy significativo hasta que no se tenga un buen catálogo.

7. Conclusiones

En este capítulo se hace una exposición de las principales conclusiones obtenidas tras la realización de esta tesis doctoral. El capítulo está organizado de la siguiente manera:

- ✓ En la **Sección 7.1** se presenta de forma detallada la consecución de cada uno de los objetivos enunciados en esta tesis expuestos en la Sección 1.5 en la que se presenta la hipótesis y los objetivos de ésta.
- ✓ En la **Sección 7.2**, se especifican las aportaciones más relevantes de este trabajo.
- ✓ En la **Sección 7.3** se muestra una relación de las diferentes publicaciones realizadas en congresos y revistas científicas durante el periodo de investigación, las cuales sirven como contraste de resultados.
- ✓ En la **Sección 7.4** se enumeran las líneas de trabajo futuras en las que se puede continuar esta propuesta de tesis doctoral.

7.1 Análisis de la Consecución de Objetivos

Para realizar un análisis detallado de la consecución de objetivos hay que hacer referencia a la Sección 1.5 de esta tesis doctoral, donde se han expuesto los objetivos parciales que se pretendían cumplir para satisfacer el objetivo global en el marco de la tesis. En esta sección se expone tanto el objetivo global de este trabajo, como los objetivos parciales, realizando una discusión sobre su grado de consecución. La Tabla 7-1 muestra el objetivo inicial de esta tesis doctoral, tal y como se expuso en la Sección 1.5.

DEFINIR UN META-MODELO QUE SISTEMATICE EL DISEÑO DE ARQUITECTURAS DE SEGURIDAD PARA DESARROLLAR SISTEMAS DE INFORMACIÓN SEGUROS

Tabla 7-1. Objetivo Inicial de la Tesis Doctoral

Para satisfacer este objetivo global, es necesario satisfacer una serie de objetivos parciales. A continuación, se presenta una valoración de la consecución de los diferentes objetivos parciales para demostrar el cumplimiento del objetivo inicial.

- **Objetivo 1:** Estudio del estado del arte de las propuestas existentes relacionadas con los patrones de seguridad, la minería de patrones de seguridad, así como marcos de trabajo y procesos, considerando las limitaciones y aportaciones de dichas propuestas.

  En el Capítulo 3, se han realizado dos revisiones sistemáticas de la literatura existente relacionada con los Patrones de Seguridad y la Minería de Patrones de seguridad. Se han analizado en profundidad 32 estudios primarios relacionados con patrones de seguridad y 2 estudios primarios relacionados con minería de patrones de seguridad. En ambas revisiones sistemáticas se han aportado conclusiones detalladas del análisis realizado en cada uno de los campos de trabajo.

- **Objetivo 2:** Estudio del estado del arte de las propuestas existentes relacionadas con la Seguridad Dirigida por Modelos (*Model-Driven Security*, MDS), utilizando una taxonomía de evaluación para realizar el análisis.

  En el Capítulo 3, se ha realizado una revisión de la literatura existente relacionada con Metodologías de Seguridad dirigidas por Modelos. Se han analizado en profundidad 5 metodologías de seguridad dirigidas por modelos (SecureUML, UMLSec, SECTEC, SEcure MDD y ModelSec). Para realizar la revisión se ha utilizado una taxonomía de evaluación basada en las siguientes 6 entradas: (i) Dominios de Aplicación, (ii) Propiedades de Seguridad, (iii) Enfoque de Modelado, (iv) Transformaciones de Modelo, (v) Herramienta y (vi) Uso de Patones. Al igual que en las revisiones sistemáticas, se han aportado conclusiones detalladas del análisis realizado en este campo de trabajo.

- **Objetivo 3:** Crear un meta-modelo para diseñar un nuevo tipo de patrón de seguridad, explicando los detalles y las relaciones de los elementos del patrón con las arquitecturas de seguridad empresariales.

Se considera que este objetivo ha sido alcanzado debido a que en el Capítulo 4 se ha definido un nuevo meta-modelo para documentar el diseño de arquitecturas de seguridad teniendo en cuenta 4 niveles de abstracción, desde un nivel de abstracción independiente de la computación hasta un nivel de abstracción dependiente del producto tecnológico.

- **Objetivo 4:** Crear un enfoque MDS para el modelado y transformación de las arquitecturas de seguridad de la información, adecuándose a distintos niveles de abstracción.

  Se considera que este objetivo ha sido alcanzado ya que en la sección 4.5 se ha definido un proceso de modelado de arquitecturas de seguridad empresariales teniendo en cuenta los 4 niveles de abstracción incluidos en las arquitecturas MDA. Además de la definición del proceso, en esa misma sección se ha definido un nuevo Lenguaje Especifico del Dominio para documentar las nuevas arquitecturas y un conjunto de transiciones para facilitar la transformación entre cada uno de los modelos de abstracción.

- **Objetivo 5:** Definir un proceso reutilizable que ayude a realizar minería de Enterprise Security Patterns para descubrir nuevos patrones y poder ampliar el catálogo de este tipo de patrones.

  Este objetivo ha sido alcanzado en su totalidad dado que en la sección 4.6 se ha introducido un nuevo proceso para facilitar la minería de Enterprise Security Patterns. Este nuevo proceso ha sido diseñado para crear un entorno que ayude a los investigadores a descubrir y documentar este nuevo tipo de patrones. Para la definición de este nuevo proceso, se han tenido en cuenta los 3 elementos principales de los Enterprise Security Patterns: el Problema, el Contexto y la Solución.

- **Objetivo 6:** Desarrollar una herramienta visual que ayude a los ingenieros de seguridad a la hora de diseñar y documentar los nuevos patrones.

Este objetivo se considera satisfecho con la herramienta presentada en el Capítulo 5. Esta herramienta representa una característica adicional hacia la consideración de los *Enterprise Security Patterns* como marco completo para la especificación y diseño de arquitecturas de seguridad.

- **Objetivo 7:** Validación de la propuesta mediante su aplicación práctica en escenarios reales.

  Este objetivo ha sido alcanzado en su totalidad dado que además del caso de uso real planteado en el Capítulo 6, los Enterprise Security Patterns han sido implantados dentro de la metodología de proyectos de seguridad en una gran corporación y apoyándose en ellos se han llevado a cabo más de 1000 proyectos en los últimos años.

Por todo esto, creemos que el objetivo principal de esta tesis doctoral ha sido conseguido, al haber cumplido los objetivos parciales y haber definido un el meta-modelo de un nuevo patrón de seguridad que ayuda a documentar y homogeneizar las arquitecturas de seguridad.

Como conclusión se puede afirmar que la hipótesis de partida de esta tesis doctoral **"Es posible sistematizar el diseño de arquitecturas de seguridad para desarrollar sistemas de información seguros"** es verdadera y que se ha resuelto exitosamente con la consecución de los objetivos propuestos.

7.2 Aportaciones de la Tesis Doctoral

A lo largo de esta tesis doctoral se han realizado diversas aportaciones que se resumen a continuación:

1. En el capítulo 3 se han realizado dos Revisiones Sistemáticas, una de Patrones de Seguridad y otra de Minería de Patrones de Seguridad. En la primera revisión se han analizado diversas propuestas haciendo foco en tres conceptos principales, (i) cómo eran definidos los patrones en cada una de ellas, (ii) si los patrones eran aplicables o no en arquitecturas de seguridad complejos y (iii) como eran clasificados los patrones. En la segunda revisión sistemática hemos llegado a la concusión que las propuestas actuales de Minería de Patrones de Seguridad podrían ser útiles en fases tempranas del ciclo de vida del desarrollo, pero son menos útiles en etapas posteriores ya que no consideran algunos de los elementos básicos asociados al diseño y construcción de arquitecturas seguras. En este capítulo también se ha realizado una evaluación de las propuestas existentes relacionadas con la Seguridad Dirigida por Modelos (*Model-Driven Security*, MDS), con el objetivo de valorar si alguna de las propuestas nos podría ayudar a la hora de diseñar arquitecturas de seguridad de grandes sistemas de información. No se ha realizado una revisión sistemática sobre este campo debido a que (Nguyen et al., 2013) ya habían realizado este trabajo anteriormente.

2. En el Capítulo 4 se ha definido un nuevo tipo de patrón de seguridad llamado *Enterprise Security Patterns*. Este nuevo patrón no está diseñado para sustituir a los patrones de seguridad existentes. Enterprise Security Patterns utilizan e incorporan los patrones de seguridad en un patrón más amplio para manejar más amenazas y proteger un conjunto de activos de información en un contexto específico. El objetivo principal de estos patrones es proporcionar una solución tecnológica segura y documentada para un contexto dado. A continuación, se muestran las aportaciones más importantes dentro de este capítulo:

- ✓ La descripción de las Arquitecturas de Seguridad Empresariales y los elementos incluidos en estas arquitecturas, con el objetivo de proporcionar una visión general de los elementos a tener en cuenta a la hora de diseñar este tipo de arquitecturas.
- ✓ La definición de una nueva Plantilla para documentar Enterprise Security Patterns. Esta plantilla muestra una relación formal entre los elementos incluidos en las Arquitecturas de Seguridad Empresariales y los elementos usados para documentar los Enterprise Security Patterns. De esta manera hemos incorporado los conceptos principales de las arquitecturas de seguridad dentro de los Enterprise Security Patterns.
- ✓ La definición del Meta-modelo completo de los Enterprise Security Patterns explicando los detalles y relaciones de los elementos incluidos en el patrón. Cada uno de los elementos básicos de las Arquitecturas de Seguridad Empresariales están representados en un diagrama, describiendo como los Enterprise Security Patterns están relacionados con ese elemento.
- ✓ La descripción del Proceso de Modelado que hemos propuesto para soportar la definición de Arquitecturas de Seguridad Empresariales, basándose en el concepto de Enterprise Security Pattern y en los diferentes meta-modelos presentados en las secciones anteriores. Para ello, se ha proporcionado una vista general del proceso completo, un lenguaje específico del dominio para representar las soluciones de los nuevos patrones y las decisiones de diseño que guían los refinamientos en cada paso del proceso.
- ✓ La definición de un nuevo marco de trabajo o *framework* para facilitar la Minería de Enterprise Security Patterns. El objetivo principal de este nuevo *framework* es crear un entorno que ayude a los investigadores a descubrir, documentar y clasificar este tipo de patrones.

3. En el capítulo 5 se ha implementado una nueva herramienta llamada C-SMArT (*Casandra - Security Model-driven Architecture Toolkit*) asociada al Lenguaje Especifico del Dominio (DSL) definido para representar las soluciones incluidas en los nuevos patrones. Esta herramienta representa una característica adicional hacia la consideración de los *Enterprise Security Patterns* como marco completo para la especificación y diseño de arquitecturas de seguridad.

4. En el capítulo 6 se ha validado el meta-modelo de Enterprise Security Patterns diseñado en el capítulo 4 con un caso de estudio real. Este estudio ha sido realizado junto al departamento de seguridad de la información de una gran organización internacional.

7.3 Contraste de Resultados

Los diferentes resultados y propuestas fruto de esta tesis doctoral han sido publicados en diversos foros científicos. En la Tabla 7-2 se muestra un resumen de las publicaciones obtenidas, que están enmarcadas dentro del ámbito de la tesis doctoral.

Tipo de Revista / Congreso	TOTAL
Revistas Internacionales	1
Revistas Internacionales JCR	2
Congresos Internacionales	4
Congresos Nacionales	1
TOTAL	**8**

Tabla 7-2. Publicaciones Ligadas a la Tesis Doctoral

En la Tabla 7-3 se muestra una relación de todas las publicaciones generadas fruto de esta tesis doctoral. En ella se clasificarán las publicaciones en base a su ámbito general y la temática específica. Se utilizará además un código para cada una de ellas (Año y Siglas) para ser detalladas en las tablas posteriores.

Ámbito	Temática específica	Publicación
Estado del Arte de Patrones de Seguridad	Homogeneidad de Patrones de Seguridad	***2010 – RECSI***
	Aplicabilidad de Patrones de Seguridad	***2010 - IS***
Nuevo enfoque para los Patrones de Seguridad	Propuesta nueva plantilla de Patrones de Seguridad	***2010 – PATTERNS***
	Caso de estudio de nueva plantilla de Patrones de Seguridad	***2012 - JAS***
Nuevo enfoque para los Patrones de Seguridad Empresarial (*Enterprise Security Patterns*)	Propuesta nueva plantilla de Patrones de Seguridad Empresarial	***2012 - WOSIS***
		2014 – SCN (JCR)
	Caso de estudio de nueva plantilla de Patrones de Seguridad Empresarial	***2013 – CS&I (JCR)***
Metodología	General	***2011 – WOSIS***

Tabla 7-3. Listado de Publicaciones por Ámbito y Temática Específica

7.3.1 Revistas

En esta sección se exponen las diferentes publicaciones realizadas que dan soporte al trabajo propuesto en esta tesis doctoral en revistas.

7.3.1.1 Internacionales

En relación a revistas científicas internacionales se han publicado los siguientes trabajos:

Código	Publicación
2012 - JAS	**Santiago Moral-García**, Roberto Ortiz, Santiago Moral-Rubio, Javier Garzás y Eduardo Fernández-Medina, A New Pattern Template to Support the Design of Security Architectures: A Case Study. Revista Internacional: **International Journal on Advances in Security**. Páginas, inicial: 173 final: 184. 2012.
2014 – SCN (JCR)	**Santiago Moral-García**, Santiago Moral-Rubio, David G. Rosado, Eduardo B. Fernandez y Eduardo Fernández-Medina, Enterprise Security Pattern: A new Type of Security Pattern. Revista Internacional: **International Journal on Security and Communication Networks**. Páginas, inicial: 1670 final: 1690. 2014. (Revista indexada con JCR = 0,720 en 2014)
2014 - CS&I (JCR)	**Santiago Moral-García**, Santiago Moral-Rubio, Eduardo B. Fernandez y Eduardo Fernández-Medina, Enterprise Security Pattern: A Model Driven Architecture Instance. Revista Internacional: **International Journal on Computer Standards and Interfaces**. Páginas, inicial: 748 final: 758. 2014. (Revista indexada con JCR = 0,879 en 2014)

Tabla 7-4. Relación de Publicaciones en Revistas Internacionales

7.3.2 Congresos

En esta sección se presentan los diferentes trabajos publicados en congresos científicos que han servido para contrastar los resultados de la tesis doctoral que aquí se propone.

7.3.2.1 Internacionales

En lo que respecta a congresos internacionales, se pueden consultar las siguientes propuestas:

Código	Publicación
2010 – IS	Roberto Ortiz, **Santiago Moral-García**, Santiago Moral-Rubio, Belén Vela, Javier Garzás y Eduardo Fernández-Medina, Applicability of Security Patterns. Congreso Internacional: The 5th International Symposium on Information Security (IS 2010), OTM 2010. Páginas, inicial: 672 final: 684. 2010.
2010 – PATTERNS	**Santiago Moral-García**, Roberto Ortiz, Santiago Moral-Rubio, Belén Vela, Javier Garzás y Eduardo Fernández-Medina, A New Pattern Template to Support the Design of Security Architectures. (Artículo premiado como uno de los tres mejores artículos de la conferencia y seleccionado para su publicación en IARIA Journals). Congreso Internacional: The 2nd International Conferences on Pervasive Patterns and Applications, Computation World 2010. Páginas, inicial: 66 final: 71. 2010.
2011 - WOSIS	**Santiago Moral-García**, Santiago Moral-Rubio y Eduardo Fernández-Medina Security, Pattern Mining: Systematic Review and Proposal. Congreso Internacional: The 8th International Workshop on Security in Information Systems. 2011.
2012 - WOSIS	**Santiago Moral-García**, Santiago Moral-Rubio, Eduardo B. Fernandez, Eduardo Fernández-Medina, A New Enterprise Security Pattern: Secure Software as a Service (SaaS). Congreso Internacional: The 9th International Workshop on Security in Information Systems. 2012.

Tabla 7-5. Relación de Publicaciones en Congresos Internacionales

7.3.2.2 Nacionales

En relación a congresos nacionales, se ha presentado el siguiente trabajo:

Código	Publicación
2010 – RECSI	**Santiago Moral-García**, Roberto Ortiz, Belén Vela, Javier Garzás y Eduardo Fernández-Medina, Patrones de Seguridad: ¿Homogéneos, validados y útiles? Congreso Nacional: XI Reunión Española sobre Criptología y Seguridad de la Información. 2010.

Tabla 7-6. Relación de Publicaciones en Congresos Nacionales

7.4 Líneas de Trabajo Futuras

Además de los aspectos tratados en los capítulos anteriores, hay otros aspectos que consideramos que serían interesantes explorar en el futuro en otras líneas de investigación asociadas a los Enterprise Security Patterns.

7.4.1 Automatización Asistida de las transformaciones entre modelos

Como ya se ha comentado anteriormente, cada uno de los modelos de la solución de los patrones (CIM, PIM, PSM y PDM) tiene una relación 1 a N con su siguiente modelo. Esto quiere decir que la solución A de un Enterprise Security Pattern podría proporcionar el mismo nivel de protección que la solución B utilizando distintas tecnologías de seguridad o componentes arquitecturales.

En esta tesis hemos creado un prototipo tecnológico que ayuda a los ingenieros a diseñar todos los modelos de los Enterprise Security Patterns sin ayuda tecnológica en la hora de aplicar las transformaciones entre cada uno de los modelos. Para poder desarrollar una herramienta que proporcione una automatización asistida dentro del proceso de desarrollo de nuevas soluciones, sería necesario mantener los siguientes 3 catálogos y crear las relaciones entre cada uno de los elementos incluidos en ellos.

- ✓ Catálogo de Patrones de Seguridad
- ✓ Catálogo de Componentes Arquitecturales
- ✓ Catálogo de Tecnologías de Seguridad

Cuando hablamos de automatización asistida nos estamos refiriendo a un proceso de desarrollo en el que el sistema vaya realizando preguntas al usuario con el fin de ir afinando la construcción de cada uno de los modelos. En este momento, no creemos que sea factible la

automatización por completo del proceso de desarrollo de estos nuevos patrones, pero también podría ser una posible línea de trabajo futura.

7.4.2 Repositorio Público de Enterprise Security Patterns

Como ya se ha comentado durante la tesis doctoral, una de las complejidades a las que se enfrentan los arquitectos de seguridad, es la elección de los mecanismos de seguridad apropiados a la hora de construir nuevos sistemas de información. En esta tesis doctoral se facilita un nuevo meta-modelo para ayudar en la construcción de estas nuevas arquitecturas, pero sería de gran utilidad la creación de una comunidad donde se publiquen todos los Enterprise Security Patterns descubiertos hasta el momento.

Dentro de este repositorio público sería interesante que los usuarios pudieran retroalimentar y enriquecer la documentación proporcionada en cada uno de los patrones. Si en algún momento empieza a crecer el número de patrones descubiertos, va a existir una nueva necesidad alrededor de la categorización de los Enterprise Security Patterns, dado que los ingenieros van a necesitar realizar búsquedas específicas para encontrar el tipo de patrón que necesitan.

7.4.3 Repositorio de Incidentes para Asistir la Minería de Enterprise Security Patterns

Como ya hemos mostrado a la hora de definir el proceso de Minería de Enterprise Security Patterns, la base de datos de incidentes es un elemento esencial a la hora de priorizar sobre que nuevos patrones debería estar trabajando la comunidad científica.

Por este motivo, creemos que sería de gran utilidad la creación de una comunidad donde se publicaran y categorizaran todos los incidentes de seguridad conocidos en los medios públicos. La comunidad científica alrededor de los Enterprise Security Patterns debería estar revisando todos los incidentes reales de carácter intencional para proporcionar a la comunidad un patrón

que ya soluciona el problema o empezar a trabajar en la minería de un nuevo patrón que ayude a proporcionar una solución.

Las empresas privadas de diversos sectores también podrían proporcionar incidentes de seguridad privados con el objetivo de priorizar la minería en incidentes reales que están afectando a empresas de sectores específicos.

8. Bibliografía

Aarsten, A., D. Brugali, et al. (1996). "Patterns of three-tier client server architectures." Proceedings of the 1996 Pattern Languages of Programs (PLoP) Conference, Monticello, IL, September 1996.

Abdelhalim, I., J. Sharp, et al. (2010). "Formal Verification of Tokeneer Behaviours Modelled in fUML Using CSP." Formal Methods and Software Engineering 6447: 371-387.

Alam, M., R. Breu, et al. (2007). "Model-Driven Security Engineering for Trust Management in SECTET." Journal of Software 2(1): 47-60.

Alam, M., M. Hafner, et al. (2006). A Framework for Modeling Restricted Delegation in Service Oriented Architecture. 3rd International Conference on Trust, Privacy, and Security in Digital Business, TrustBus 2006, Springer-Verlag.

Alberts, C. J. y A. Dorofee (2002). Managing Information Security Risks: The Octave Approach, Addison-Wesley Longman Publishing Co., Inc.

Alexander, C. (1964). "Notes on the Synthesis of Form." Harvard University Press.

Alexander, C. (1975). "The Oregon Experiment." Oxford University Press.

Alexander, C. (1979). "The Timeless Way of Building." Oxford University Press.

Alexander, C., S. Ishikawa, et al. (1977). "A Pattern Language: Towns, Buildings, Constructions." Oxford University Press.

Ali Al, M., H. A. Ahmed, et al. (2016). "Cyber Terrorism Taxonomies: Definition, Targets, Patterns, Risk Factors, and Mitigation Strategies." International Journal of Cyber Warfare and Terrorism (IJCWT) 6(1): 1-12.

Alpcan, T. (2014). "Game Theory for Security."

Alvi, A. K. y M. Zulkernine (2012). A comparative study of software security pattern classifications. Availability, Reliability and Security (ARES), 2012 Seventh International Conference on, IEEE.

Anwar, Z., W. Yurcik, et al. (2006). Multiple design patterns for voice over IP (VoIP) security. Performance, Computing, and Communications Conference, 2006. IPCCC 2006. 25th IEEE International.

Apt, K. R. y E. Grädel (2011). Lectures in game theory for computer scientists, Cambridge University Press.

Arconati, N. (2002). One Approach to Enterprise Security Architecture, SANS Institute. SANS Security Essentials GSEC version 1.3.

Assange, J., from https://wikileaks.org/.

Atkinson, C. y T. Kuhne (2003). "Model-Driven Development: A Metamodeling Foundation " IEEE Software 20 (5): 36-41.

Barron, E. N. (2013). Game theory: an introduction, John Wiley & Sons.

Basin, D., M. Clavel, et al. (2011). A Decade of Model-Driven Security. 16th ACM Symposium on Access Control Models and Technologies, SACMAT 2011, Innsbruck, Austria, ACM.

Basin, D., J. Doser, et al. (2003). Model driven security for process-oriented systems. ACM Symposium on Access Control Models and Technologies, Como, Italy, ACM Press.

Basin, D., J. Doser, et al. (2006). "Model Driven Security: from UML Models to Access Control Infrastructures." ACM Transactions on Software Engineering and Methodology 15(1): 39-91.

Baskerville, R. (1997). "Distinguishing Action Research From Participative Case Studies." Journal of Systems and Information Technology 1: 25-45.

Baskerville, R. (1999). "Investigating Information Systems with Action Research." Communications of the Association for Information Systems 2(19).

Baskerville, R. y T. Wood-Harper (1996). "A Critical Perspective on Action Research as a Method for Information Systems Research." Journal of Information Technology 11: 235-246.

Best, B., J. Jürjens, et al. (2007). Model-Based Security Engineering of Distributed Information Systems Using UMLSec. 29th International Conference on Software Engineering - ICSE: 581-590.

Bézivin, J. (2004). "In search of a basic principle for model driven engineering " Novatica Journal, Special Issue 5 (2): 21-24.

Bouaziz, R. y B. Coulette (2012). Applying Security Patterns for Component Based Applications Using UML profile. Computational Science and Engineering (CSE), 2012 IEEE 15th International Conference on, IEEE.

Braga, A. M., C. M. F. Rubira, et al. (1998). Tropyc: A Pattern Language for Cryptographic Software. in the 5th Pattern Language of Programming Conference (PLoP'98).

Brereton, P., B. Kitchenham, et al. (2007). "Lessons from applying the systematic literature review process within the software engineering domain." J. Syst. Software: 571-583.

Breu, R., M. Hafner, et al. (2005). Model driven security for inter-organizational workflows in e-government. TCGOV.

Breu, R., G. Popp, et al. (2007). "Model Based Development of Access Policies." International Journal on Software Tools for Technology Transfer 9: 457-470.

BSI (2000). IT Baseline Protection Manual, Federal Agency for Security in Information Technology, Germany.

Buschmann, F., R. Meunier, et al. (1996). Pattern-oriented software architecture: A system of patterns, Wiley.

Caralli, R. A., J. F. Stevens, et al. (2007). Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process, Carnegie Mellon University.

Cavelty, M. D. y V. Mauer (2016). Power and Security in the Information Age: Investigating the Role of the State in Cyberspace, Taylor & Francis.

CC. "Common Criteria for Information Technology Evaluation." from http://www.commoncriteriaportal.org/.

CCTA (2003). CCTA Risk Análisis and Management Method (CRAMM), Versión 5.0. C. C. a. T. Agency.

Chavhan, N. A. y S. A. Chhabria (2009). Multiple design patterns for voice over IP security. Proceedings of the International Conference on Advances in Computing, Communication and Control. Mumbai, India, ACM.

Clavel, M., F. Durán, et al. (1999). The Maude System. 10th International Conference on Rewriting Techniques and Applications.

Cox Jr, L. A. T. (2009). "Game theory and risk analysis." Risk Analysis 29(8): 1062-1068.

Cuadrado, J. S., J. G. Molina, et al. (2006). RubyTL: A Practical, Extensible Transformation Language Model Driven Architecture - Foundations and Applications Springer Berlin / Heidelberg

Cuevas, A., P. El Khoury, et al. (2008). Security Patterns for Capturing Encryption-Based Access Control to Sensor Data. SECURWARE '08. Second International Conference on Emerging Security Information, Systems and Technologies.

Curran, J. (2003). "Conformance Testing: An Industry Perspective." Java Conformance Testing Sun Microsystems.

CyberSecurity Ventures (2016). Hackerpocalypse: A Cybercrime Revelation.

Daniel, G., G. Sunyé, et al. (2016). NeoEMF: a Multi-database Model Persistence Framework for Very Large Models. In Proceedings of the MoDELS 2016 Demo and Poster Sessions co-located with ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2016).

Dausend, M. y F. Raiser (2011). Model Transformation using Constraint Handling Rules as a basis for Model Interpretation Proceedings of the Eighth International Workshop on Constraint Handling Rules, El Cairo, Egypt

Delessy, N., E. B. Fernandez, et al. (2007). A Pattern Language for Identity Management. ICCGI 2007. International Multi-Conference on Computing in the Global Information Technology.

Delessy, N. A. y E. B. Fernandez (2008). A pattern-driven security process for SOA applications. Proceedings of the 2008 ACM symposium on Applied computing. Fortaleza, Ceara, Brazil, ACM.

Díaz, M. P., S. Montero, et al. (2005). Patrones e interfaz de usuario. Ingeniería de la web y patrones de diseño. P. E. S.A.: 309-329.

Dua, S. y X. Du (2016). Data Mining and Machine Learning in Cybersecurity, CRC Press.

Edwards, G., C. Seo, et al. (2008). "Model Interpreter Frameworks: A foundation for the analysis of Domain-specific software architectures " Journal of Universal Computer Science 14 (8): 1182-1206.

ElPais. "La industria informática y la banca unen sus fuerzas para frenar el 'phishing'." from http://elpais.com.

Encina, O., E. B. Fernandez, et al. (2014). "A misuse pattern for Denial-of-Service in federated Inter-Clouds." Procs. of AsianPLoP (Pattern Languages of Programs).

Espinazo, J. y J. Garcia (2014). "Querying large models efficiently." Information Software Technology 56 (6): 586-622.

Essmayr, W., G. Pernul, et al. (1996). Access Controls by Object-Oriented Concepts. Proceedings of IFIP WG 11.3 Eleventh International Conference on Database Security.

Estay, C. y J. Pastor (2000). Improving Action Research in Information Systems with Project management. 2000 Americas Conference on Information Systems. Long Beach, California: 1558-1561.

Estay, C. y J. Pastor (2000). Towards a project structure for Action-Research in Information Systems. 10th Annual Business and Information Technology Conference (BIT). Manchester (United Kingdom).

Estay, C. y J. Pastor (2001). Un Modelo de Madurez para Investigación-Acción en Sistemas de Información. Jornadas de Ingeniería del Software y Bases de Datos. (JISBD). Ciudad Real. España: 265-281.

Fernández-Medina, E., J. Jurjens, et al. (2009). "Model-Driven Development for secure information systems." Information and Software Technology 51 (5): 809-814.

Fernández-Medina, E. y M. Piattini (2005). "Designing Secure Databases." Information and Software Technology 47: 463–477.

Fernandez, E., J. Pelaez, et al. (2007). Attack Patterns: A New Forensic and Design Tool. Advances in Digital Forensics III: 345-357.

Fernandez, E., G. Pernul, et al. (2008). Patterns and Pattern Diagrams for Access Control. Trust, Privacy and Security in Digital Business: 38-47.

Fernandez, E. B. (2002). Patterns for Operating Systems Access Control. in Proceedings of Pattern Language of Programming (PLoP'02).

Fernandez, E. B. (2013). Security patterns in practice: Building secure architectures using software patterns, Wiley.

Fernández, E. B. (2007). "Security patterns and secure systems design." ACM Southeast Regional Conference : 510 [DBLP:conf/ACMse/Fernandez07].

Fernandez, E. B., E. Alder, et al. (2012). A Misuse Pattern for Retrieving Data from a Database Using SQL Injection. BioMedical Computing (BioMedCom), 2012 ASE/IEEE International Conference on, IEEE.

Fernandez, E. B., M. Fonoage, et al. (2008). The Secure Three-Tier Architecture Pattern. CISIS 2008. International Conference on Complex, Intelligent and Software Intensive Systems.

Fernandez, E. B., M. M. Larrondo-Petrie, et al. (2006). A Methodology to Develop Secure Systems Using Patterns. Integrating security and software engineering: Advances and future vision, IGI Global: 107-126.

Fernandez, E. B., S. Mujica, et al. (2011). Two security patterns: Least Privilege and Security Logger/Auditor. Asian PLoP.

Fernandez, E. B. y J. L. Ortega-Arjona (2009). The Secure Pipes and Filters Pattern. DEXA '09. 20th International Workshop on Database and Expert Systems Application.

Fernandez, E. B. y R. Pan (2001). A Pattern Language for Security Models. in Proceedings of Pattern Language of Programming (PLoP'01)

Fernandez, E. B., J. C. Pelaez, et al. (2007). Security Patterns for Voice over IP Networks. ICCGI 2007. International Multi-Conference on Computing in the Global Information Technology.

Fernandez, E. B. y G. Pernul (2006). Patterns for session-based access control. Proceedings of the 2006 conference on Pattern languages of programs. Portland, Oregon, ACM.

Fernandez, E. B., T. Sorgente, et al. (2006). Even more patterns for secure operating systems. Proceedings of the 2006 conference on Pattern languages of programs. Portland, Oregon, ACM.

Fernandez, E. B., H. Washizaki, et al. (2008). Classifying Security Patterns. Asia-Pacific Web Conference: 342-347.

Fernandez, E. B., J. Wu, et al. (1994). User Group Structures in Object-Oriented Databases. Proceedings of the 8th Annual IFIP W.G.11.3 Working Conference on Database Security, Germany.

Fernandez, E. B., J. Wu, et al. (2009). On building secure SCADA systems using security patterns. Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies. Oak Ridge, Tennessee, ACM.

Fernandez, E. B., N. Yoshioka, et al. (2009). Modeling Misuse Patterns. ARES '09. International Conference on Availability, Reliability and Security.

Fernandez, E. B. y X. Yuan (2007). Securing analysis patterns. Proceedings of the 45th annual southeast regional conference. Winston-Salem, North Carolina, ACM.

Fischer, T., A. R. Sadeghi, et al. (2009). A Pattern for Secure Graphical User Interface Systems. DEXA '09. 20th International Workshop on Database and Expert Systems Application.

Flore, F. (2003). "MDA: The proof is in automating transformations between models." OptimalJ White Paper 1-4.

fortune.com. from http://beta.fortune.com/global500.

Fowler, M. (1997). "Analysis Patterns: Reusable Object Models." Adisson Wesley.

French, W. L. y C. H. Bell, Eds. (1996). Organizational Development: Behavioral Science Interventions for Organization Improvement. London, Prentice Hall.

Fundación Innovación Bankinter (2016). Ciberseguridad, un desafío mundial.

Gamma E., H., R., Johnson, R., Vlissides J. (1995). "Design Patterns: Elements of Reusable Object Oriented Software." Addison Wesley.

Georg, G., I. Ray, et al. (2009). "An aspect-oriented methodology for designing secure applications." Information and Software Technology 51(5): 846-864.

Gerber, A., M. Lawley, et al. (2002). "Transformation: The Missing Link of MDA." Graph Transformation 2505: 90-105.

Gervais, M. P. (2002). Towards an MDA-oriented methodology Computer Software and Applications Conference, COMPSAC 2002.

GIZMODO. "Fox News' Twitter Account Hacked." from http://gizmodo.com/5817870/fox-news-twitter-account-hacked-claims-barack-obama-is-dead.

Glass, R. L., I. Vessey, et al. (2002). "Research in software engineering: an analysis of the literature." Information & Software Technology 44(8): 491-506.

Glass, R. L., I. Vessey, et al. (2004). "An analysis of research in computing disciplines." Communications of the ACM 47(6): 89-94.

GmbH, I. O. S. (2005). ArcStyler, The leading platform for Model Driven Architecture (MDA).

Google. "Google Suite." from https://gsuite.google.com/.

Graham, D. (2006). Introduction to the CLASP Process, OWASP CLASP Project http://www.owasp.org/index.php/Category: OWASP_CLASP_Project.

Gronback, R. C. (2009). Eclipse modeling project: a domain-specific language (DSL) toolkit.

Guttman, M. y J. Parodi (2007). Real-life MDA: solving business problems with model driven architecture, Morgan Kaufmann.

Hafner, M., M. Breu, et al. (2005). Modelling inter-organizational workflow security in a peer-to-peer environment. IEEE International Conference on Web Services (ICWS'05), Washington, DC, USA, IEEE Computer Society.

Hafner, M., R. Breu, et al. (2006). "SECTET: An Extensible Framework for the realization of Secure inter-organizational Workflows." Internet Research 16(5): 491-506.

Hafner, M., M. Memon, et al. (2008). Modeling and Enforcing Advanced Access Control Policies in Healthcare Systems with SECTET. Models in Software Engineering, Springer-Verlag. 5002 of Lecture Notes in Computer Science: 132-144.

Harmon, P. (2004). The OMG's model driven architecture and BPM. Newsletter of Business Process Trends. 2 (5).

Harrop, W. y A. Matteson (2015). Cyber Resilience: A Review of Critical National Infrastructure and Cyber-Security Protection Measures Applied in the UK and USA. Current and Emerging Trends in Cyber Operations: Policy, Strategy and Practice. F. Lemieux. London, Palgrave Macmillan UK: 149-166.

Hashizume, K., N. Yoshioka, et al. (2013). "Three misuse patterns for Cloud Computing." Security engineering for Cloud Computing: approaches and Tools: 36-53.

Hatebur, D., M. Heisel, et al. (2011). Systematic Development of UMLsec Design Models Based on Security Requirements. Fundamental Approaches to Software Engineering, Springer-Verlag. 6603 of Lecture Notes in Computer Science: 232–246.

Horvath, V. y T. Dörges (2008). From security patterns to implementation using petri nets. Proceedings of the fourth international workshop on Software engineering for secure systems. Leipzig, Germany, ACM.

Houmb, S. y J. Jürjens (2003). Developing Secure Networked Web-Based Systems Using Model-Based Risk Assessment and UMLsec. 10th Asia-Pacific Software Engineering Conference, APSEC 2003, IEEE Computer Society.

IBM (2007). "Enterprise Security Architecture. Using IBM Tivoli Security Solutions." International Technical Support Organization.

IBM. "Rational Rose Product line. ." from http://www-03.ibm.com/software/products/en/ratirosefami.

IC3 (2015). 2015 Internet Crime Report, Internet Crime Complaint Center.

ISACA (2012). COBIT 5 for Information Security.

ISO. "International Organization for Standardization." from http://www.iso.org.

ISO. "ISO/IEC 17799:2005." from http://www.iso.org.

ISO/IEC 27000. "International Organizacion for Standardization (ISO) and the International Electrotechnical Commission 27000:2014." from http://www.iso.org.

Jaworski, J. y P. J. Perrone (2000). Java Security Handbook, SAMS.

Jensen, J. y M. G. Jaatun (2011). Security in Model Driven Development: A Survey. 6th International Conference on Availability, Reliability and Security, ARES 2011, IEEE Computer Society.

Jouault, F., F. Allilaire, et al. (2008). "ATL: A model transformation tool." Science of computer programming 72(1): 31-39.

Jürjens, J. (2001). Towards Development of Secure Systems Using UMLsec. Fundamental Approaches to Software Engineering, Springer-Verlag. volume 2029 of Lecture Notes in Computer Science: 187-200.

Jürjens, J. (2002). UMLsec: Extending UML for Secure Systems Development. UML 2002 - The Unified Modeling Language, Springer-Verlag. 2460 of Lecture Notes in Computer Science: 412–425.

Jürjens, J. (2004). Model-Based Security Engineering with UML. 4th International School on Foundations of Security Analysis and Design, FOSAD 2004, Springer-Verlag.

Jürjens, J. (2005). Secure Systems Development with UML, Springer-Verlag.

Jürjens, J. (2007). Developing Secure Embedded Systems: Pitfalls and How to Avoid Them. 29th International Conference on Software Engineering, ICSE 2007, Minneapolis, MN, USA, IEEE Computer Society.

Kasal, K., J. Heurix, et al. (2011). Model-Driven Development Meets Security: An Evaluation of Current Approaches. Proceedings of the 44th Hawaii International Conference on System Sciences, HICSS-44 2011, IEEE Computer Society.

Kerth, N. L. y W. Cunningham (1997). "Using Patterns to Improve Our Architectural Vision." IEEE Software 23: 53-59.

Khwaja, A. y J. Urban (2002). "A Synthesis of Evaluation Criteria for Software Specifications and Specification Techniques." International Journal of Software Engineering and Knowledge Engineering 12: 581-599.

Kitchenham, B. (2004). "Procedures for Perfoming Systematic Review." Joint Technical Report, Software Engineering Group, Departement of Computer Scinece Keele University, United Kingdom and Empirical Software Engineering, National ICT Australia Ltd.: Australia.

Kitchenham, B. (2007). "Guideline for performing Systematic Literature Reviews in Software Engineering. Version 2.3." University of Keele (Software Engineering Group, School of Computer Science and Mathematics) and Durham (Department of Conputer Science).

Kleppe, A. G., J. Warmer, et al. (2003). MDA explained, the model driven architecture: practice and promise, Addison-Wesley

Kock, N. y F. Lau (2001). "Information Systems Action Research: Serving Demanding Masters." Information Technology and People (special Action Research in Information Systems) 14(1): 6-11.

Kodituwakku, S. R., P. Bertok, et al. (2001). A Pattern Language for Designing and Implementing Role-Based Access Control in Proceedings of the 1st Australian Conference on Pattern Languages of Programming.

Kolovos, D. S., R. F. Paige, et al. (2009). On the evolution of OCL for capturing structural constraints in modelling languages. Rigorous Methods for Software Construction and Analysis, Springer Berlin Heidelberg: 204-218.

Kulkarni, V. y S. Reddy (2003). "Separation of Concerns in Model-Driven Development " IEEE Software 20 (5): 64-69.

Lankhorst, M. (2009). Enterprise Architecture at Work: Modelling, Communication and Analysis, Springer.

Lau, F. (1997). Review on the Use of Action Research in Information Systems Studies. Information Systems and Qualitative Research. A. S. Lee, J. Liebenau and J. I. Degross. Chapman and Hill. London: 31-68.

Lázaro, M. y E. Marcos (2005). "Research in Software Engineering: Paradigms and Methods." 1st Workshop on Philosophical Foundations of Information Systems Engineering, held at 17th Conference on Advanced Information System Engineering (CAiSE'05): 535-545.

Lea, D. (1999). "Christopher Alexander: an Introduction for Object-Oriented Designers." http://www.cmcrossroads.com/bradapp/docs/patterns-intro.html.

Lewin, K. (1946). "Action research and minority problems." Journal of Social Issues 2.

Li, B., S. Ge, et al. (2004). Research and Implementation of Single Sign-On Mechanism for ASP Pattern. Grid and Cooperative Computing – GCC 2004: 161-166.

Li, S., S. Liu, et al. (2010). A Research and Implementation of Model Execution Method Based on MOF. Third International Symposium on Computer Science and Computational Technology (ISCSCT'10) China.

Lobato, L. L., E. B. Fernandez, et al. (2009). Patterns to Support the Development of Privacy Policies. ARES '09. International Conference on Availability, Reliability and Security.

Lodderstedt, T., D. Basin, et al. (2002). SecureUML: A UML-Based Modeling Language for Model-Driven Security. 5th International Conference on the Unified Modeling Language (UML), 2002, Dresden, Germany, Springer.

Lucio, L., Q. Zhang, et al. (2014). Advances in Model-Driven Security. Advances in Computers. 93: 103-152.

Madroño, C. "Consorcio de Universidades de la Comunidad de Madrid y de la UNED para la Cooperación Bibliotecaria." from http://www.consorciomadrono.es/.

MagicDraw. from http://www.nomagic.com.

Maña, A., D. Serrano, et al. (2009). Development of Applications Based on Security Patterns. DEPEND '09. Second International Conference on Dependability.

MAP. "MAGERIT – versión 3.0. Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información."

McAfee Labs (2016). Predicciones sobre amenazas para 2016.

McTaggart, R. (1991). "Principles of Participatory Action Research." Adult Education Quarterly 41(3).

Mellor, S. J., S. Kendall, et al. (2004). MDA distilled, Addison Wesley.

Mernik, M., J. Heering, et al. (2005). "When and how to develop domain-specific languages." ACM Computer Surveys 37 (4): 316-344.

Meservy, T. O. y K. D. Fenstermacher (2005). "Transforming software development: an MDA road map." IEEE Computer 38 (9): 52-58.

Moebius, N., K. Stenzel, et al. (2012). Incremental Development of Large, Secure Smart Card Applications. 1st International Workshop on Model-Driven Security, ACM.

Moebius, N., K. Stenzel, et al. (2009). Model-Driven Code Generation for Secure Smart Card Applications. 20th Australian Software Engineering Conference, ASWEC 2009, IEEE Computer Society.

Moebius, N., K. Stenzel, et al. (2009). SecureMDD: A Model-Driven Development Method for Secure Smart Card Applications. International Conference on Availability, Reliability and Security, ARES 2009.

Moebius, N., K. Stenzel, et al. (2009). Generating Formal Specifications for Security-Critical Applications - A Model-Driven Approach. 2009 ICSE Workshop on Software Engineering for Secure Systems, IWSESS 2009.

Moebius, N., K. Stenzel, et al. (2010). "Formal Verification of Application-Specific Security Properties in a Model-Driven Approach." Engineering Secure Software and Systems 5965 of Lecture Notes in Computer Science: 166–181.

Monsalve, S. (2003). "John Nash y la teorıa de juegos." Lecturas matemáticas 24: 137-149.

Moral-García, S., S. Moral-Rubio, et al. (2014). "Enterprise Security Pattern: A New Type of Security Pattern." Security and Communication Networks Journal (Wiley) 7 (11): 1670-1690.

Moral-García, S., R. Ortiz, et al. (2011). "A New Pattern Template to Support the Design of Security Architectures: A Case Study." International Journal on Advances in Security (IARIA) 4(3&4): 173-184.

Moral-García, S., R. Ortiz, et al. (2010). A new Pattern Template to Support the Design of Security Architectures. The International Conferences on Pervasive Patterns and Applications. PATTERNS: 66-71.

Morrison, P. y E. B. Fernandez (2006). The credentials pattern. Proceedings of the 2006 conference on Pattern languages of programs. Portland, Oregon, ACM.

Mouratidis, H. (2011). "Secure Software Systems Engineering: The Secure Tropos Approach." Journal of Software 6(3): 331-339.

Muñoz-Arteaga, J., R. M. González, et al. (2009). "A methodology for designing information security feedback based on User Interface Patterns." Advances in Engineering Software 40(12): 1231-1241.

Nash, J. (1951). "Non-cooperative games." Annals of mathematics: 286-295.

Nash, J. F. (1950). "Equilibrium points in n-person games." Proc. Nat. Acad. Sci. USA 36(1): 48-49.

Neves, F. D. y A. Garrido (1998). Bodyguard. in Proceedings of Pattern Languages of Programming Conference (PLoP'96).

Nguyen, P. H., J. Klein, et al. (2013). A Systematic Review of Model Driven Security. the 20th Asia-Pacific Software Engineering Conference, APSEC 2013, IEEE Computer Society.

NIST. "National Institute of Standards and Technology (NIST): Computer Security Resource Center." from http://csrc.nist.gov/.

Oldevik, J. (2006). MOFScript user guide, Version 0.6

OMG (2003). MDA Guide, Version 1.0.1, Object Management Group.

OMG (2011). Meta Object Facility (MOF) 2.0 Query/View/Transformation (QVT) Specification, Version 1.1. O. M. Group.

OSF. "DATALOSS db - Open Security Foundation." from http://datalossdb.org/.

Padak, N. y G. Padak (1994). Guidelines for Planning Action Research Projects. . Ohio Literacy Resource Center.

Parnas, D. L. (1972). "On the Criteria To Be Used in Decomposing Systems into Modules " Communications of the ACM 15 (12): 1053-1058.

Pelaez, J., E. B. Fernandez, et al. (2009). "Misuse patterns in VoIP." Security and Communication Networks Journal. Wiley 2(6): 635-653.

Pilato, C., B. Collins-Sunsman, et al. (2008). Version Control with Subversion O'Reilly Media

PMI (2000). PMBOK: A Guide to the Project Management Body of Knowledge. Project Management Institute Communications, Unite States.

Polo, M., F. Ruiz, et al. (2002). "Using a Qualitative Research Method for Building a Software Maintenance Methodology." Software Practice and Experience 32(13): 1239-1260.

Poole, J. D. (2001). Model-driven architecture: Vision, standards and emerging technologies. Workshop on Metamodeling and Adaptive Object Models, ECOOP, 2001.

Potter, B., J. Sinclair, et al. (1990). An Introduction to Formal Specification and Z, Prentice-Hall.

Rankel, D. S. (2002). Model Driven Architecture: Applying MDA to Enterprise Computing New York, USA, John Wiley & Sons

Rising, L. y D. E. Delano (1998). The Patterns handbook, Cambridge University Press.

Romanosky, S. (2003). "Enterprise Security Patterns." Information Systems Security Association Journal.

Romanosky, S., A. Acquisti, et al. (2006). Privacy patterns for online interactions. Proceedings of the 2006 conference on Pattern languages of programs. Portland, Oregon, ACM.

Rosado, D. G., C. Gutiérrez, et al. (2006). "Security patterns and requirements for internet-based applications." Internet Research: Electronic Networking Applications and Policy 16: 519-536.

Rubio, S. M. (2016). CASANDRA. Metodología de Análisis y Gestión de Riesgos de Seguridad de la Información basada en Teoría de Juegos. Madrid, Universidad Rey Juan Carlos.

Ryoo, J., P. Laplante, et al. (2010). A Methodology for Mining Security Tactics from Security Patterns. HICSS 2010 - the 43rd Hawaii International Conference on System Sciences. Honolulu, Hawaii

Sanchez, O., F. Molina, et al. (2009). "ModelSec: A Generative Architecture for Model-Driven Security." Journal of Universal Computer Science 15: 2957–2980.

Santiago, I., J. M. Vara, et al. (2013). Towards the effective use of traceability in model-driven engineering projects. International Conference on Conceptual Modeling, Springer Berlin Heidelberg.

Santiago, I., J. M. Vara, et al. (2013). Supporting Service Versioning-MDE to the Rescue. International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE): 212-217.

Sarmah, A., S. M. Hazarika, et al. (2008). Security Pattern Lattice: A Formal Model to Organize Security Patterns. DEXA '08. 19th International Conference on Database and Expert Systems Application.

Schmidt, C. D. (2006). "Model-driven engineering " IEEE Computer 39 (2): 25-31.

Schmidt, H., D. Hatebur, et al. (2011). A pattern-based method to develop secure software. Software Engineering for Secure Systems: Industrial and Research Perspectives, IGI Global.

Schnjakin, M., M. Menzel, et al. (2009). A pattern-driven security advisor for service-oriented architectures. Proceedings of the 2009 ACM workshop on Secure web services. Chicago, Illinois, USA, ACM.

Schumacher, M. (2003). B. Example Security Patterns and Annotations. Security Engineering with Patterns: 171-178.

Schumacher, M. (2003). Firewall Patterns. in Proceedings of Pattern Language of Programming (EuroPLoP'03).

Schumacher, M. (2003). Security Engineering with patterns - Origins, Theoretical Model, and New Applications, Springer-Verlag.

Schumacher, M. (2003). A. Sources for Mining Security Patterns. Security Engineering with Patterns: 167-169.

Schumacher, M., E. Fernandez-Buglioni, et al. (2006). Security Patterns: Integrating Security and Systems Engineering, Wiley.

SEI (1995). The Capability Maturity Model: Guidelines for Improving the Software Process. Software Engineering Institute.

SEI, C. M. U. (1988). "CERT Coordination Center."

Selic, B. (2003). "The pragmatics of model-driven development." IEEE Software 20 (5): 19-25.

Selic, B. (2008). "MDA manifestations." The European Journal for the Informatics Professional IX (2): 12-16.

Selic, B. (2012). "What will it take? A view on adoption of model-based methods in practice." Software & Systems Modeling 11(4): 513-526.

Sendall, S. y W. Kozaczynski (2003). "Model transformation: The heart and soul of model-driven software development." IEEE Software 20(5): 42-45.

Shaw, M. (2002). "What makes good research in software engineering?" International Journal on Software Tools for Technology Transfer (STTT) 4(1): 1-7.

Sherwood, J., A. Clark, et al. (2009). Enterprise Security Architecture, SABSA White Paper.

Shin, J., H. Son, et al. (2015). "Development of a cyber security risk model using Bayesian networks." Reliability Engineering & System Safety 134: 208-217.

Sorniotti, A., P. El Khoury, et al. (2009). A Security Pattern for Untraceable Secret Handshakes. SECURWARE '09. Third International Conference on Emerging Security Information, Systems and Technologies.

Spanoudakis, G., C. Kloukinas, et al. (2007). Towards security monitoring patterns. Proceedings of the 2007 ACM symposium on Applied computing. Seoul, Korea, ACM.

Srba, J. (2008) "How to Read and Present a Scientific Paper." 33.

Stahl, T., M. Völter, et al. (2006). Model-Driven Software Development: Technology, Engineering, Management John Wiley & Sons

Steinberg, D., F. Budinsky, et al. (2008). EMF: Eclipse Modeling Framework, Addison-Wesley.

Tratt, L. (2005). "Model transformations and tool integration." Software and Systems Modeling 4 (2): 112-122.

Truyen, F. (2006). The Fast Guide to Model Driven Architecture, The Basics of Model Driven Architecture (MDA), Object Management Group (OMG).

Vara, J. M. (2009). M2DAT: a technical solution for Model-Driven development of Web Information Systems PhD Thesis. Madrid, Rey Juan Carlos University.

Vara, J. M. y E. Marcos (2012). "A framework for model-driven development of information systems: Technical decisions and lessons learned." Journal of Systems and Software 85 (10): 2368-2384.

Vogel, O. (2001). "EuroPLoP 2001 design fest designing a three-tier architecture pattern language." Design Fest EuroPLoP 2001, POSA3, Irsee, Germany, July 2001.

Völter, M. (2009). "Best Practices for DSLs and Model-Driven Development " Journal of Object Technology 8 (6): 79-102.

Völter, M. (2011). "From Programming to Modeling-and Back Again " Software IEEE 28 (6): 20-25.

Völter, M. "MD* Best Practices." from http://www.voelter.de.

Wadsworth, Y. (1998). What is Participatory Action Research? Action Research International, Paper 2.

Washizaki, H., E. B. Fernandez, et al. (2009). Improving the Classification of Security Patterns. DEXA '09. 20th International Workshop on Database and Expert Systems Application.

Watson, A. (2008). "A brief history of MDA " Upgrade, the European Journal for the Informatics Professional 9 (2): 7-11.

Welch, I. (1999). Reflective Enforcement of the Clark-Wilson Integrity Model. in Proceedings of the 2nd Workshop in Distributed Object Security (OOPSLA).

Whittle, J., J. Hutchinson, et al. (2014). "The State of Practice in Model-Driven Engineering." IEEE Software 31(3): 79-85.

Wood-Harper, T. (1985). Research Methods in Information Systems: Using Action Research. Research Methods in Information Systems, Amsterdam. North-Holland.

Wood, C. C. (2000). Information Security Policies Made Easy, Version 7.

Wu, G., W. Liu, et al. (2009). Model Interpretation Development: Analysis Design of Automatic Control System. Interpretation of Information Processing Regulations.

Yazar, Z. (2002). A qualitative risk analysis and management tool – CRAMM, SANS InfoSec Reading Room White Paper.

Yoder, J. y J. Barcalow (1997). "Architectural Patterns for Enabling Application Security." Fourth Conference on Patterns Languages of Programs (PLoP'97).

Yskout, K., T. Heyman, et al. (2006). "An inventory of security patterns." Technical Report CW-469, Katholieke Universiteit Leuven, Department of Computer Science.

Zhang, Y., Y. Xiao, et al. (2012). "A survey of cyber crimes." Security and Communication Networks Journal. Wiley 5(4): 422-437.