RELACION ENTRE PATRONES DE SEGURIDAD CORE SECURITY PATTERNS (CSP) Y SECURITY PATTERNS PRACTICE (SPP)

MARITZA RAMOS

MÁSTER EN INVESTIGACIÓN EN INFORMÁTICA, FACULTAD DE INFORMÁTICA,
UNIVERSIDAD COMPLUTENSE DE MADRID

Trabajo Fin Máster en Sistemas Inteligentes

Calificación Obtenida: (8.0) NOTABLE
Convocatoria: 2013/2014

23/06/2014

Director:
Antonio Navarro Martin

Autorización de Difusión

MARITZA RAMOS

24/06/2014

El/la abajo firmante, matriculado/a en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: “RELACION ENTRE PATRONES DE SEGURIDAD CORE SECURITY PATTERNS Y SECURITY PATTERNS PRACTICE ”, realizado durante el curso académico 2013-2014 bajo la dirección de ANTONIO NAVARRO MARTIN, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Maritza Ramos

Autor

Antonio Navarro

Director

Resumen

La seguridad es un aspecto fundamental en las aplicaciones empresariales de hoy en día, es por ello que existen diversidad de herramientas y catálogos destinados a evitar ataques maliciosos a sistemas y aplicaciones, el enfoque de este trabajo de investigación se centra en dos catálogos de patrones de seguridad de diferentes autores que pueden ser aplicados en la vida real por desarrolladores.

El catálogo Core Security Patterns (CSP) contiene veintidós (22) patrones de seguridad que recopilan un conjunto de buenas prácticas desde el año 2005, y es actualmente uno de los textos de referencia que engloba los principales patrones de seguridad.

Otro importante catálogo es Security Patterns in Practice (SPP), que contiene sesenta y ocho (68) patrones que recopilan un conjunto de patrones de seguridad publicados en el año 2013.

Sin embargo, a la hora de diseñar o enseñar a construir una aplicación empresarial segura utilizando patrones de seguridad nos surgen importantes dudas: ¿qué catálogo es más adecuado? ¿Son quizás catálogos equivalentes? ¿Qué patrones diferenciadores incluyen cada catálogo?

El objetivo de este trabajo de investigación es responder a estas preguntas analizando ambos catálogos, y sus correspondientes patrones, estableciendo así una relación entre los patrones descritos en ellos.

De esta forma, y teniendo en cuenta los patrones de ambos catálogos, SPP y CSP, consideraremos para cada patrón:

- Descripción de cada patrón
- La relación existente entre ambos catálogos.
- Las ventajas de descripción en ambos catálogos, cuando haya una coincidencia entre dos patrones.
- Una serie de cuestiones interesantes a destacar en ambos catálogos.

Palabras clave

Patrones de seguridad, gestión de identidad, autenticación, autorización, control de acceso, servicios web seguros, capas, redes, capa de negocio.

Abstract

Security is a fundamental aspect of business applications, at the present exist a diversity of tools and catalogs designed to prevent malicious attacks on systems and applications, this research focuses in two catalogs of security patterns from different authors that could be applied in real life for developers.

The catalog Core Security Patterns (CSP) contains twenty two (22) patterns that collect a set of best practices since 2005 and is currently one of the most books referenced in security patterns.

Another important catalog is Security Patterns in Practice (SPP), which contains sixty eight (68) patterns. At present, they collect a set of security patterns published since 2013.

However, when designing or building a secure enterprise application, arise important questions, like: What catalog is more appropriate? Maybe, are they equivalent catalogs? What are the different patterns in each catalog?

The goal of this research work is to answer these questions by analyzing both catalogs and patterns and relating the patterns described in them.

Thus, for each pattern in both catalogs, SPP and CSP, we consider:

- Description of each pattern.
- Relationship between the two catalogs.
- The advantages of description in both catalogs, when there is agreement between two patterns.
- A number of interesting points to note in both catalogs

Keywords

Security patterns, identity management, authentication, authorization, access control, web service security, layers, networks, business layer.

Índice de contenidos

Índice de figuras

Índice de tablas

Agradecimientos

A *Venoni*,

Autor de mi felicidad...

Un especial agradecimiento a Antonio, mi tutor, por su ayuda y excelente gestión.

1. INTRODUCCIÓN

Durante la última década se han producido fenómenos tecnológicos tan relevantes como el crecimiento exponencial de usuarios en internet, la proliferación de dispositivos móviles y teléfonos inteligentes, la extensión del ancho de banda, la multiplicación del comercio electrónico, el cloud computing, y las redes sociales entre otros.

El gran impacto de los ordenadores y la tecnología informática ha generado la necesidad de diseñar y desarrollar nuevos sistemas de software, así como incorporar nuevas tecnologías de aplicaciones que se expanden rápidamente. Los ingenieros de software también tienen la tarea de evolucionar y mantenerse actualizados ante las nuevas áreas de especialización, cambios en la tecnología y nuevos retos al tratar de entender, rediseñar y mantener sistemas vulnerables y complejos.

Al realizar el análisis y diseño de un sistema, existen componentes de hardware y software que interactúan entre sí con sistemas complejos de información. En este proceso pueden ir surgiendo nuevos escenarios que no fueron previstos en un primer análisis, siendo imprescindible hacer varios análisis después para detectar debilidades y errores, evitando así que el sistema sea vulnerable a ataques de seguridad. Los patrones de seguridad representan hoy en día algunas de las mejores prácticas logradas por la industria a fin de detener o limitar los ataques en sistemas informáticos. Estos patrones nos ayudan a realizar un análisis global de un sistema complejo en su totalidad, permitiendo a los desarrolladores de aplicaciones tomar las medidas necesarias para garantizar la seguridad del software.

Cada patrón de diseño de software describe un problema y una solución, pudiéndose reutilizar esta solución una y otra vez millones de veces sin hacer lo mismo otra vez. Por lo tanto, un patrón es aplicable a diferentes problemas de diseño en distintas circunstancias (Gamma et. al., 1995).

La utilización de patrones de seguridad nos permite construir solidas arquitecturas de seguridad, facilitando su mantenimiento y previniendo vulnerabilidades que pueda tener un sistema.

Sin embargo, en la actualidad existen diversos catálogos de patrones de seguridad. Algunos patrones son considerados en distintos catálogos, mientras que otros, son específicos de determinados catálogos. Por otro lado, algunos patrones de seguridad son patrones

arquitectónicos que organizan la división de los sistemas software en módulos altamente cohesivos y débilmente acoplados, mientras que otros afectan a los protocolos de comunicación o a la forma en que los entornos de ejecución de software (p.e. máquinas virtuales o servidores de aplicaciones) acceden y ejecutan las declaraciones de un lenguaje.

Por tanto, a la hora de aplicar convenientemente estos patrones se hace necesario analizar las similitudes y diferencias entre los catálogos de patrones para decidir cuáles son más útiles a la hora de construir una aplicación software. Este aspecto está directamente ligado al aspecto docente. Al igual que uno de los principales catálogos de arquitectura multicapa que (Alur et al., 2007) incluye veintiún patrones de diseño, pero sólo unos pocos son fundamentales para implementar la arquitectura (controlador frontal y de aplicación, servicio de aplicación, transferencia, objeto del negocio/entidad compuesta, objeto de acceso a datos y almacén del dominio) (Navarro et al. 2012), sería deseable conocer cuáles son estos patrones básicos que garantizan la seguridad en un sistema software.

De los diversos catálogos de seguridad existentes, cabe destacar dos: Core Security Patterns, CSP, (Steel et al., 2005) y Security Patterns in Practice, SPP, (Fernández, 2013). El primero es un catálogo publicado por los ingenieros de Sun Microsystems/Oracle, y además de venir avalado por la empresa responsable de la arquitectura J2EE, en principio, parece que es susceptible de ser fácilmente integrado con el catálogo de arquitectura multicapa de la compañía (Alur et al., 2007). El segundo es un catálogo propuesto por un experto en seguridad, versión revisada y extendida de un catálogo previamente propuesto (Schumacher et al., 2006) en la prestigiosa serie de catálogos de patrones de Frank Buschmann (Buschmann et al., 2002; Buschmann et al., 2007; Buschmann et al., 2007; Schmidt et al., 2005). Sin embargo, a la hora de diseñar o enseñar a construir una aplicación empresarial segura surgen importantes dudas: ¿qué catálogo es más adecuado? ¿Son quizás catálogos equivalentes? ¿Qué patrones diferenciadores incluyen cada catálogo?

El objetivo de este trabajo de investigación es responder a estas preguntas analizando ambos catálogos de patrones y estableciendo una relación entre los patrones descritos en ellos.

Así, este trabajo de investigación analiza los catálogos de patrones de seguridad CSP y SPP con el objetivo de:

- Determinar qué catálogo de patrones es más adecuado para la inclusión de características de seguridad en una aplicación empresarial multicapa.

- Comparar los patrones expuestos en ambos catálogos de patrones de seguridad, estableciendo, en su caso, las relaciones, diferencias o similitudes entre ambos catálogos de patrones.
- Organizar los patrones de seguridad SPP según las capas de una arquitectura multicapa.
- Analizar y explicar los beneficios, ventajas y desventajas de la utilización de cada patrón.

Las siguientes secciones analizan el trabajo relacionado (Sección 2) y comparan ambos patrones. Al existir más patrones en el catálogo SPP que en CSP se ha decidido utilizar las categorías del catálogo SPP para estructurar este trabajo. Así, primero aparecen los patrones SPP que tienen algún tipo de relación con el catálogo CSP (Secciones 3 a la 9). Después, se incluye una sección de patrones sin relación en ambos catálogos (Sección 10). Finalmente se presentan las conclusiones, indicando las relaciones globales entre ambos catálogos, incluyendo la relación entre capas.

Para aquellos patrones relacionados se proporcionará:

- Una descripción del patrón.
- La relación existente entre ambos catálogos.
- Las ventajas de descripción en ambos catálogos, cuando haya una coincidencia entre dos patrones.
- Una serie de cuestiones interesantes a destacar en ambos catálogos.

2. ESTADO DEL ARTE

2.1 Introducción

Actualmente hay distintas razones por las que las aplicaciones son inseguras, tales como la falta de tiempo debido a plazos agresivos y presupuestos ajustados, la falta de conocimiento, pues los expertos en IT pocas veces lo son en seguridad, la falta de prioridades, la funcionabilidad y el rendimiento en general.

Los patrones de seguridad son una forma sistemática de capturar la experiencia de expertos acerca de buenos diseños y mejores prácticas de seguridad, en donde se recopilan todas las posibles soluciones para evitar cometer los mismos errores de proyectos anteriores en proyectos a futuro.

Esta sección analiza distintos catálogos de patrones para comprobar que los objetivos de este trabajo de investigación no están ya resueltos.

Antes de comenzar a hablar de patrones es necesario identificar el concepto de que es un patrón y cuáles son sus orígenes.

(Beck y Cunningham, 1987) comenzaron experimentando con la idea de aplicar patrones a la programación, presentado sus resultados en la conferencia OOPSLA de ese año. En los siguientes años, Beck, Cunningham y otros siguieron este trabajo.

Los patrones de diseño se hicieron populares en la informática después de que los patrones de diseño del libro: Design Patterns: Elements of Reusable Object-Oriented Software fuera publicado en 1994 por la llamada "Banda de los Cuatro", que se abrevio como "GOF" (Gamma et al., 1995) .En el mismo año, la primera conferencia de programación del lenguaje de patrones fue realizada y al año siguiente también, el Portland Pattern Repository se creó para la documentación del diseño de patrones.

El diseño de patrones de seguridad fue introducido para identificar las soluciones en problemas que ocurrían una y otra vez en la programación orientada a objetos. Otro de los autores (Yoder y Barcalow, 1998) adapto las mejoras en la seguridad de la información, se incluyeron varios patrones, entre los siete primeros: Single Access Point, Check Point, Roles, Session, Full View with Errors, Limited view, Secure Access Layer.

La clasificación de patrones puede variar según cada autor. A continuación se muestra la agrupación de los catálogos de patrones, según (Fernandez, 2013)

- Patrones de Arquitectura de Software: definen una arquitectura software para la construcción de una aplicación.
- Patrones Orientados a Objetos: son soluciones específicas a problemas concretos, pero no sirven para estructurar la arquitectura de una aplicación.
- Patrones de Seguridad: patrones centrados en la seguridad de las aplicaciones.

En esta sección sólo analizaremos los arquitectónicos y seguridad, al ser los que nos interesan para este trabajo.

2.2 Patrones de Arquitectura del Software

2.2.1 Core J2EE Patterns: Best Practices and Design Strategies, Second edition

El libro Core J2EE Patterns: Best Practices and Design Strategies, Second edition fue escrito por Deepak Alur, John Crupi y Dan Malks y fue publicado en junio de 2003 (Alur. et al. 2003)

El catálogo de patrones J2EE contiene veintiún patrones revisados y documentados que proporcionan soluciones para aplicaciones empresariales a través de estrategias diseñadas para las capas de presentación, negocio e integración. Cubre servlets, JSP (Kurniawan, 2012), EJBs (Panda et al., 2007), JMS (Jendrock et al., 2013), Web Services (Kalin, 2013). Está ilustrado con diagramas UML y muestra ejemplo de código simple para patrones, estrategias y *refactoring*.

Es un libro fácil de comprender y seguir, también se muestra claramente los estados de la relación entre los patrones y cómo interactúan entre ellos.

Presentation Tier Patterns

La capa de presentación se encarga de realizar la entrada y salida de datos con el usuario.

A continuación se describen los patrones de la capa de presentación.

- Intercepting Filter: intercepta las peticiones entrantes y las respuestas salientes y se aplica un filtro. Estos filtros pueden agregarse y eliminarse de una manera declarativa.
- Front Controller: es un contenedor que almacena la lógica común de procesamiento que se produce dentro de la capa de presentación y que de una u otra manera puede ser colocado erróneamente en una vista.
- Context Object: encapsula los estados en un protocolo de forma independiente para compartirlo a través de la aplicación, Usando un Context Object se hace fácil las pruebas, facilitando un ambiente más genérico de prueba con dependencia reducida sobre un contenedor específico.
- Application Controller: mantiene un control centralizado, recuperación e invocación de vistas y procesamiento de comandos. Mientas un Front Controller actúa como un punto de acceso centralizado y un controlador de requerimientos, el Application Controller es responsable de identificar e invocar comandos.
- View Helper: utiliza los componentes del Helper para encapsular la lógica relacionada con la recuperación de contenido, validación, adaptación y formato al modelo.
- Composite View: sugiere la composición de una vista de numerosas piezas, múltiples vistas pequeñas, ambas estáticas y dinámicas son piezas que se juntan para crear un plantilla individual.
- Service to Worker: Centraliza el control y requerimiento en el manejo de recuperación del modelo de presentación antes de volver a tener el control de la vista, la vista genera una respuesta dinámica basada en el modelo de presentación.
- Dispatcher View: Combina un controlador y un despachador con vistas y ayudas para manejar los requerimientos de los clientes y la preparación de presentaciones dinámicas como respuesta.

Business Tier Patterns

La capa de negocios está compuesta por los componentes que proveen la lógica de negocio para una aplicación. La lógica de negocio es el código que provee la funcionalidad para un dominio de negocio como la industria financiera o sitios de comercio electrónico. Esta capa realiza el procesamiento más pesado, incluyendo validación, reglas de negocio, flujos de trabajo e interfaces para sistemas externos.

A continuación se describen los patrones de la capa de negocio:

- Business Delegate: reduce el acoplamiento entre capas remotas y provee un punto de entrada para acceder a servicios remotos en la capa de negocio.
- Service Locator: encapsula los mecanismos de implementación para buscar componentes de servicio de negocio.
- Session Façade: provee servicios de grano grueso al cliente ocultando la complejidad de las interacciones de servicio de negocio.
- Application Service: centraliza y agrega el comportamiento para proveer una capa de servicio uniforme para la capa de servicios de negocio.
- Business Object: implementa el modelo de dominio conceptual utilizando un modelo de objeto.
- Composite Entity: implementa un objeto de negocio utilizando una entidad local. además agrega entidades de negocio sobre una entidad de grano grueso.
- Transfer Object: provee la mejor técnica y estrategia para intercambiar datos a través de la capas.
- Transfer Object Assembler: construye una composición Transfer Object de varias fuentes. Estas fuentes pueden ser componentes EJB, Data Access Objects u otros objetos arbitrarios de Java.
- Value List Handler: guarda los resultados de la ejecución de una *query* y devuelve subconjuntos del resultado a los clientes.

Integration Tier Patterns

Esta capa se comunica con la capa de recursos, escribiendo y leyendo los datos del almacén persistente.

A continuación se describen los patrones de la capa de integración:

- Data Access Object: elimina el acoplamiento entre las capas de negocio y de recursos, encapsulando toda la lógica de acceso a datos para crear, recuperar, eliminar y actualizar datos de un almacén persistente. Utiliza el Transfer Object para enviar y recibir datos.
- Service Activator: permite el procesamiento asíncrono en sus aplicaciones empresariales usando JMS, puede invocar un Application Service, Sesión Façade, Business Objects, también se puede utilizar varios activadores de servicio para proporcionar el procesamiento asíncrono paralelo para tareas de larga ejecución.
- Domain Store: proporciona un mecanismo poderoso para implementar la persistencia transparente para su modelo de objetos, combina y conecta otros patrones que incluyen el Data Access Objects.
- Web Service Broker: expone uno o más servicios en su aplicación para clientes externos como un servicio web utiliza XML y protocolos web estándar. Un Web Service Broker puede interactuar con Application Service y Session Façade, utiliza uno o más Service Activators para llevar a cabo el proceso asíncrono de una solicitud.

2.2.2 Patterns of Enterprise Application Architecture

El libro Patterns of Enterprise Application Architecture fue escrito por Martin Fowler, David Rice, Matthew Foemmel, Edward Hieatt, Robert Mee y Randy Stafford y fue publicado en Noviembre del 2002 (Fowler et al., 2002). Incluye un tutorial sobre el desarrollo de aplicaciones empresariales y hace una referencia detallada a los propios patrones. Cada patrón proporciona información de uso y aplicación, así como ejemplos de código detallados en Java o C #.

Los temas que incluye en este libro son:

- La división de una aplicación empresarial en capas.
- Los principales enfoques para la organización de la lógica de negocio.
- Un tratamiento a fondo de mapeo entre objetos y bases de datos relacionales.
- El uso del patrón *Model- View- Controller* para organizar una presentación Web.
- Manejo de la concurrencia de datos que abarca varias transacciones.
- Diseño de interfaces de objetos distribuidos.

Los patrones que se incluyen en el libro son:

Domain Logic Patterns

Son los patrones que gestionan y organizan los roles e interacciones de componentes específicos del dominio:

- Transaction Script: organiza toda la lógica primaria como un procedimiento individual, haciendo llamadas directamente a la base de datos o a través de una base de datos oculta.
- Domain Model: crea una web de objetos interconectados, donde cada objeto representa algunos significados individuales un modelo objeto del dominio que incorpora ambos comportamiento y datos.
- Table Module: organiza una lógica de dominio con una clase por tablas de base de datos., Service Layer: este es un conjunto de operaciones disponibles desde la perspectiva de las capas de la interface del cliente.

Data Source Architectural Patterns

Esta familia de patrones ofrece diseños flexibles y desacoplados para conectar una aplicación a una fuente de datos, encapsula la lógica de acceso a los datos:

- Table Data Gateway: tiene una interface simple, usualmente consiste en varios métodos de búsqueda para obtener datos de una base de datos y actualizar, insertar y borrar métodos.
- Row data Gateway: proporciona un objeto que imita una fila de una base de datos.
- Active Record: es un objeto que oculta una fila de una tabla o vista de una base de datos. Encapsula el acceso de la base de datos y agrega una lógica de dominio en los datos.
- Data Mapper: es una capa de software que separa los objetos en memoria de la base de datos, este es responsable de transferir los datos entre los dos.

Object-Relational Behavioral Patterns

Son patrones que se esfuerzan por resolver los problemas de diseño con respecto a cómo y cuándo los objetos asociados se conservan y se cargan desde un origen de datos relacional:

- Unit of Work: mantiene una lista de objetos afectados por una transacción de negocio y coordina los cambios y la resolución de problemas.
- Identity Map: asegura que cada objeto se encuentra cargado una sola vez, manteniendo cada objeto cargado en un mapa. Busca objetos en un mapa cuando los referencia a ellos.
- Lazy Load: es un objeto que no contiene todos los datos que se necesitan pero conoce como obtenerlos.

Object-Relational Structural Patterns

Son patrones que resuelven problemas de diseño relacionado con la composición de los objetos dentro del origen de datos relacional:

- Identity Field: guarda un campo ID de la base de datos en un objeto para mantener la identidad entre un objeto en memoria y una fila de la base de datos.
- Foreign Key Mapping: mapea una asociación entre objetos y una referencia de clave foránea entre tablas.
- Association Table Mapping: guarda una asociación como una tabla con una clave foránea a las tablas que están unidas por una asociación.
- Dependent Mapping: tiene una clase persistente (el dependiente) que depende de otra clase persistente (el dueño).
- Embedded Value: mapea un objeto sobre varios campos de otras tablas de objetos.
- Serialized LOB: guarda un gráfico de los objetos serializándolos sobre un objeto grande (LOB), el cual guarda en un campo de la base de datos.
- Single Table Inheritance: representa la jerarquía de herencia de clases como una tabla individual que tiene columnas para todas las tablas de varias clases.
- Class Table Inheritance: representa una jerarquía de herencia de clases con una tabla para cada clase.
- Concrete Table Inheritance: representa una jerarquía de herencia con una tabla por una clase en concreto en la jerarquía.
- Inheritance Mappers: es una estructura para organizar los mapeos a la base de datos que manejan una jerarquía de herencia.

Object-Relational Metadata Mapping Patterns

Son patrones que resuelven los problemas causados por la disparidad entre el modelo orientado a objetos y el modelo relacional de un sistema:

- Metadata Mapping: mantiene detalles del mapeo de un objeto relacional en meta datos.
- Query Object: es un objeto que representa las búsquedas de la base de datos.
- Repository: intercede entre el dominio y las capas de mapeo de datos utilizando una colección como interfaz para acceder a objetos del dominio.

Web Presentation Patterns

Son patrones que resuelven los problemas de comportamiento y estructurales que se producen en la capa de presentación de un sistema:

- Model View Controller: separa la interacción de la interface del usuario sobre tres diferentes roles.
- Page Controller: es un objeto que maneja un requerimiento de una página específica o acción de un sitio web.
- Front Controller: un controlador que maneja todos los requerimientos de un sitio web.
- Template View: provee información sobre una página HTML a través de marcadores incrustados en dicha página.
- Transform View: es una vista que procesa los datos del dominio elemento tras elemento y los transforma en HTML.
- Two Step View: cambia los datos del dominio sobre HTML en dos pasos: primero genera una página lógica y después traduce la página lógica en HTML.
- Application Controller: es un punto centralizado de navegación de manejo de pantalla y del flujo de una aplicación.

Distribution Patterns

Son patrones que resuelven problemas de diseño que afectan a los sistemas o componentes y sus interacciones distribuidas:

- Remote Facade: provee una fachada de grano grueso sobre objetos de grano fino para mejorar la eficiencia sobre una red.
- Data Transfer Object: es un objeto que lleva datos entre procesos con la finalidad de reducir el número de llamadas a métodos.

Offline Concurrency Patterns

Son patrones que sirven para gestionar transacciones de negocio en aplicaciones distribuidas:

- Optimistic Offline Lock: previene conflictos entre transacciones de negocios concurrentes al detectar un conflicto y deshacer la transacción.
- Pesimistic Offline Lock: Previene conflictos entre transacciones de negocios por aprobación de una sola transacción de negocio al tiempo que se accede a los datos.
- Coarse-Grained Lock: bloquea un conjunto de objetos relacionados con un bloque individual.
- Implicit Lock: permite a la infraestructura o capa de código a adquirir bloqueos fuera de línea.

Session State Patterns

Son patrones que resuelven problemas relacionados con la administración del estado dentro de un sistema de varios niveles:

- Client Session State: guarda el estado de la sesión en el cliente.
- Server Session State: mantiene el estado de la sesión en un sistema de servidores en una forma serializada.
- Database Session State: guarda los datos de la sesión en la base de datos.

Base Patterns

Son patrones primitivos, base de los patrones anteriores:

- Gateway: es un objeto que encapsula el acceso al sistema externo o a un recurso.
- Mapper: es un objeto que configura una comunicación entre dos objetos independientes.
- Layer Supertype: es un tipo que actúa como un super tipo en esta capa.
- Separated Interface: define una interfase en un paquete separado de su implementación.
- Registry: es un objeto conocido que otros objetos pueden utilizar para encontrar objetos comunes y servicios.
- Value Object: un objeto simple y pequeño, fácil de crear.
- Money: representa el valor monetario.
- Special Case: es una subclase que provee un comportamiento especial para casos particulares.
- Plugin: son clases de enlaces durante la compilación.
- Service Stub: remueve la dependencia sobre un servicio problemático durante las pruebas.

2.2.3 SOA Design Patterns

El libro SOA Design Patterns fue escrito por Thomas Erl y publicado en diciembre del 2008 (Erl, 2008). Ofrece una colección de patrones que aborda uno a uno un conjunto típico de problemas de diseño SOA con soluciones bien articuladas. Contiene contribuciones de Microsoft, Oracle y Red Hat.

Los patrones que se incluyen en el libro están agrupados en diversas categorías:

- *Logical Inventory Layer Patterns*. Establecen un mecanismo general de organización de los servicios dentro de un inventario de grupos lógicos. Cada capa se basa en un tipo de servicio y por lo tanto representa un conjunto de servicios que se ajustan a este tipo. Estos tipos corresponden a las clasificaciones de la industria que se conoce como modelos de servicio.
- *Inventory Centralization Patterns*. Hacen frente a los aspectos físicos de la arquitectura de inventario de servicios.
- *Inventory Implementation Patterns*. Para hacer frente a los problemas de diseño más comunes relacionados con la arquitectura del inventario de servicios, estos patrones

ofrecen un conjunto de patrones especializados que ayudan a resolver los problemas a nivel de aplicación.

- *Inventory Governance Patterns*. Al diseñar primero un inventario de servicios, hay pasos que se pueden tomar para asegurarse de que se reduce el esfuerzo y el eventual impacto de tener que gobernar el inventario. Esta categoría describe un conjunto de patrones que proporcionan algunas soluciones fundamentales en diseño y tiempo específicamente con la implementación del inventario.
- *Foundational Service Patterns*. Estos patrones de diseño representan los pasos más esenciales requeridos para dividir y organizar la lógica y sus soluciones de servicios soportados. En muchos sentidos, estos patrones pueden ser considerados la teoría fundamental en la orientación a servicios.
- *Service Implementation Patterns*. Cada uno de estos patrones de diseño afecta a la arquitectura de servicios de una manera específica, afectando su implementación física. La mayoría la considera especializada, lo que significa que se van a utilizar para los requisitos específicos y pueden no ser necesarios en absoluto (y rara vez se utilizan todos juntos).
- *Service Security Patterns*. Dado que las soluciones orientadas a servicios se componen típicamente de servicios agregados, cada parte dentro de una arquitectura de composición puede convertirse en un objetivo potencial para un fallo de seguridad. A menudo tienen que estar equipadas con controles adicionales que les permitan soportar ataques maliciosos.
- *Service Contract Design Patterns*. Esta categoría hace un gran énfasis en el diseño de los contratos de servicios. El principio del diseño del contrato de servicio estandarizado, requiere que todos los contratos estén conformes en un inventario de servicio dado.
- *Legacy Encapsulation Patterns*. Ofrece un conjunto de patrones dedicados a abordar los desafíos comunes con la encapsulación de servicios de los sistemas heredados y sus entornos.
- *Service Governance Patterns*. Los patrones de esta categoría se centran en el gobierno de servicios.

- *Capability Composition Patterns*. Son patrones que proporcionan los medios para componer la lógica del servicio que se descompone, divide y se simplifica a través de la identificación del servicio.
- *Service Messaging Patterns*. Estos patrones proveen varias técnicas para procesar y coordinar el intercambio de datos entre servicios.
- *Composition Implementation Patterns*. Proporcionan un conjunto variado de soluciones de diseño que se ocupan de cuestiones a nivel de implementación relacionadas primariamente con la gestión del servicio en tiempo de ejecución y la estructura de composición.
- *Service Interaction Security Patterns*. Cuando se diseñan soluciones empresariales, las cuales ofrecen composiciones de servicios complejos, los servicios pueden estar sujetos a una variedad de escenarios, donde cada uno pueden introducir riesgos de seguridad únicos y requerimientos.
- *Transformation Patterns*. Estos patrones han sido parte de la integración de arquitecturas por muchos años y continuaran siendo evaluados actualmente en implementaciones en SOA.

2.2.4 J2EE Design Patterns

El libro J2EE Design Patterns fue escrito por William Crawford y Jonathan Kaplan, y publicado en septiembre de 2003 (Crawford y Kaplan, 2003).

El libro de Crawford es muy práctico y muestra cómo aplicar los patrones en el mundo real. Los autores extienden los patrones de diseño a áreas no cubiertas por otros libros, como el modelado de datos, modelado de transacciones / procesos y la interoperabilidad.

Los Patrones de Diseño J2EE ofrecen una amplia cobertura de las cinco áreas problemáticas que enfrentan los desarrolladores empresariales:

- Mantenimiento (extensibilidad).
- Rendimiento (escalabilidad del sistema).
- Modelado de datos (Modelado de Objetos de Negocios).
- Transacciones (Modelado de procesos).
- Mensajería (interoperabilidad).

El catálogo identifica antipatrones, así como distintas capas de patrones:

- *Presentation Tier Patterns*. Incluye los patrones para la capa de presentación de aplicaciones empresariales. Esta capa se asemeja mucho a los patrones de los catálogos de Alur et al. (2003) y Fowler et al. (2002).
- *Advanced Architectural Patterns*. Se centra en los patrones que dividen la capa de presentación en componentes pequeños y reutilizables
- *Scalability Patterns*. Estos patrones aumentan la escalabilidad de la capa de presentación utilizando variaciones de este concepto.
- *Business Tier Patterns*. Esta capa implementa las reglas de negocio y se asemeja mucho a la capa de los catálogos de Alur et al., (2003) y Fowler et al., (2002).
- *Business Tier Interface Patterns*. Esta capa conecta el modelo de dominio con el cliente, (aplicaciones del cliente) o con la capa de presentación del lado del servidor (en aplicaciones Web). Estos patrones también implementan alguna lógica de negocio, controlando y limitando las acciones de la capa de presentación se puede llevar a cabo el modelo de dominio.
- *Concurrency Patterns*. Esta categoría cuenta con dos elementos: la gestión de las transacciones y gestión de la concurrencia. La gestión de transacciones, se abstrae a un alto nivel, controla la implementación del uso de casos individuales dentro del sistema, lo que garantiza que los cambios en los datos subyacentes se realizan constantemente y que el sistema está siempre en un estado válido. La gestión de concurrencia controla cómo diferentes usuarios acceden a un sistema de manera simultánea.
- *Messaging Patterns*. La mensajería es un componente importante en el entorno de cada empresa orientada al servicio.

2.3 Patrones de Seguridad

2.3.1 Growing a Pattern Language for Security

Este artículo del autor (Hafiz et al., 2013) se basa en identificar y describir todos los patrones de seguridad que han sido publicados en varios lugares como libros, artículos y

catálogos, se explica cómo han sido catalogados y clasificados estos patrones. También explica si son útiles para ayudar a encontrar los patrones apropiados para cada aplicación. En este trabajo se analiza los Core Secure Patterns (CSP) en conjunto con otros 151 patrones, pero a diferencia del trabajo actual, no considera los patrones del catálogo SPP, publicado en 2013.

2.3.2 Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management

El libro de Core Security Patterns: Best Practices and Strategies for J2EE Web Services, and Identity Management fue escrito por Christopher Steel, Ramesh Nagappan, Ray Lai y publicado en octubre de 2005 (Steel et al., 2005)

Los autores explican los fundamentos de la seguridad de las aplicaciones Java desde la base. Se introduce una metodología de seguridad de gran alcance, un marco de seguridad independiente del proveedor, una lista de verificación de evaluación detallada, y veintidós patrones de arquitectura de seguridad.

El libro se centra en diversos aspectos de la seguridad de las aplicaciones empresariales:

- Lo que funciona y lo que no: las mejores prácticas J2EE de aplicaciones de seguridad, y los errores comunes que deben evitarse.
- La implementación de las funciones de seguridad de plataforma de Java en las aplicaciones del mundo real.
- El establecimiento de los servicios de seguridad Web mediante firma XML, XML Encryption, WS -Security, XKMS, y me WS perfil de seguridad básico.
- El diseño de gestión de la identidad y de servicio los sistemas de aprovisionamiento utilizando SAML, Liberty, XACML, y SPML.
- Diseño de soluciones de identificación personal segura utilizando tarjetas inteligentes y biométricas.
- Verificación de metodologías de diseño de seguridad, patrones, las mejores prácticas, las estrategias defensivas y listas de verificación de evaluación.

- Estudio de caso de extremo a extremo en la arquitectura de seguridad: la arquitectura de diseñar e implementar una solución de seguridad de extremo a extremo para aplicaciones a gran escala.

El catálogo identifica cuatro capas fundamentales en las que agrupa los patrones:

- *Web Tier Security Patterns*. Para aplicaciones J2EE, la capa Web representa el punto de entrada, la puerta frontal para todos los usuarios. También es el punto inicial para gestionar un ataque en búsqueda de debilidades en la seguridad de una aplicación.
- *Business Tier Security Patterns*. Incluye los componentes responsables de implementar la lógica de negocios en la aplicación.
- *Web Services Tier Security Patterns*. Incluye los servicios web, basados en estándares XML, para el desarrollo y despliegue de componentes de aplicación.
- *Identity Management & Service Provisioning*: Esta capa proporciona un marco común de diseño, unificando el Single-Sign On (SSO) y los mecanismos globales de cierre de sesión para el uso de aplicaciones heterogéneas.

2.3.3 Security Patterns in Practice: Designing Secure Architectures Using Software Patterns

El libro Security Patterns in Practice: Designing Secure Architectures Using Software Patterns fue escrito por Eduardo Fernandez-Buglioni y fue publicado en mayo de 2013 (Fernandez, 2013). El libro contiene sesenta y ocho patrones que han sido considerados en este trabajo (todos salvo los centrados en sistemas operativos). Todos los patrones presentados en el libro han sido discutidos en conferencias y se han analizado ampliamente.

Este libro describe la motivación y la experiencia en el uso de patrones, el catálogo de patrones, ejemplos de la vida real para situarse en el escenario en el cual se puede aplicar los patrones, muestra una colección de patrones de seguridad, el problema, solución, implementación, diagrama UML, consecuencias, usos conocidos y referencias.

La metodología se basa en construir sistemas de seguridad utilizando los patrones y siguiendo los ejemplos que se muestran en el libro, el enfoque es estrictamente de ingeniería esto no indica que no se utilice la teoría pero se solo se indica cuando es necesario.

Los patrones muestran descripciones detalladas de nivel intermedio, en donde se detalla el significado del patrón y evaluar sus posibilidades, no existe material de fondo acerca de la seguridad, en conclusión es un libro interdisciplinario

El catálogo clasifica los patrones en distintas capas:

- *Patterns for Identity Management*. Contiene patrones que describen reglas de control de acceso a recursos en función del nivel de confianza.
- *Patterns for Authentication*. Incluyen patrones para identificar la identidad del usuario y transmitirla a distintos componentes de seguridad que la necesiten.
- *Patterns for Access Control*. Una vez que el sujeto se le ha dado el acceso a un sistema, se necesita controlar el acceso a los recursos específicos. Los derechos de los sujetos en el sistema se definen utilizando algunos modelos de control de acceso y se expresa en forma de reglas de autorización. Los modelos de seguridad son una expresión más detallada y precisa de las políticas que se utilizan como guía para construir y evaluar sistemas, por lo general se describen de manera formal o informal.
- *Security Patterns for Networks*. Identifica distintos patrones de seguridad aplicables a redes TCP/IP.
- *Patterns for Web Services Security*. Esta capa contiene patrones de seguridad y normas para los servicios web. Muchos han sido los patrones identificados en la comunidad de los servicios web. Algunos patrones descritos aquí son versiones especializadas de patrones fundamentales como XACML.
- *Patterns for Web Services Cryptography*. Contiene patrones para el cifrado y descifrado de mensajes aplicados a servicios web.
- *Patterns for Secure Middleware*. Incluye patrones para la construcción de una capa de web de middleware segura.
- *Misuse Patterns*. Esta categoría identifica amenazas de alto nivel. Son más bien antipatrones que patrones.

- *Patterns for Cloud Computing Architecture*. Más que patrones de seguridad, identifica distintas arquitecturas relacionadas con cloud computing y menciona la necesidad de incluir elementos de seguridad en estas arquitecturas.

3. PATRONES DE GESTIÓN DE IDENTIDAD

3.1 Circle of Trust

Descripción

El patrón de Circle of Trust (Figura 1) describe una relación de confianza entre proveedores de servicio según la arquitectura y los requerimientos de servicio. Dentro de este círculo de confianza los usuarios pueden realizar transacciones de negocio en un entorno seguro e integrado, evitando recordar múltiples contraseñas y utilizar diferentes protocolos.

Cada proveedor de servicios establece una relación de negocio con un conjunto de proveedores de servicios, redes e infraestructura, estas relaciones se materializan por la existencia de acuerdos de negocios entre servicios que podrían incluir el intercambio de información y confianza concretamente, los proveedores tendrían que intercambiar credenciales a través de algunos canales externos para reconocerse uno al otro.

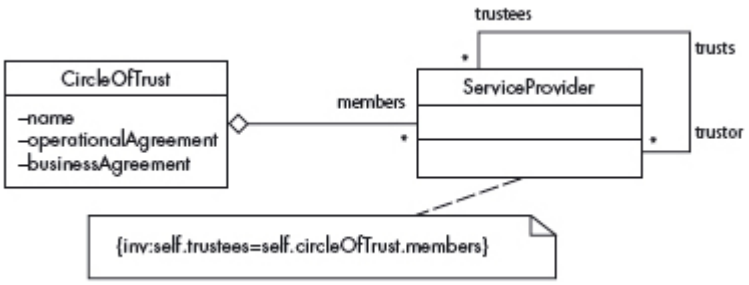


Figura 1. Diagrama de clase del patrón SPP Circle of Trust

Relación entre ambos catálogos

El patrón Circle of Trust no está identificado explícitamente como patrón en el catálogo CSP, sin embargo sí es mencionado como un elemento que constituye el Liberty Alliance Architecture. Es un patrón muy sencillo que sólo tiene sentido en contextos de federaciones de identidades.

Ventajas por catálogos

El patrón SPP Circle of Trust está relacionado con los patrones SPP Identity Provider, SPP Identity Federation y SPP Liberty Alliance Identity Federation. El catálogo CSP ofrece ejemplos de situaciones reales y permite entender fácilmente la definición de los patrones y las fases de la arquitectura Liberty Alliance.

A destacar en ambos catálogos

SPP

- Existe un entorno de confianza entre los proveedores de identidad debido a que son miembros de una federación y no existe la preocupación por el mal uso de la información.
- Los usuarios pueden interactuar de forma más segura con potenciales desconocidos utilizando los servicios de Circle of Trust.
- Los servicios pueden intercambiar información acerca de los usuarios utilizando los acuerdos que describen tecnologías de uso común.

CSP

- Los proveedores de servicio necesitan tener algún tipo de acuerdo antes que sus cuentas puedan estar federadas, se deben intercambiar credenciales a través de algunos canales externos para tener disponible la confianza y se puedan reconocer unos a otros.
- Los sujetos y los proveedores de servicio de la federación de identidades necesitan confiar en el proveedor.
- Los sujetos pueden desarrollar transacciones dentro de la entidad federada de forma segura.
- Las diferentes representaciones de identidad pueden ser consolidadas dentro de la misma entidad federada.
- No existe la necesidad de re-autenticarse en un nuevo dominio.
- Los sujetos pueden controlar tanto si una identidad local puede ser federada o no, parte de la información del usuario puede mantenerse de forma confidencial.

3.2 Identity Provider

Descripción

El patrón SPP Identity Provider (Figura 2) representa a un servicio que contiene toda la información requerida para la autenticación y la autorización de una identidad que accede a un servicio, es responsable de almacenar y enviar partes de la información del usuario (partes que forman su identidad) a aplicaciones y servicios que lo necesitan y permite la centralización de la información de la identidad del sujeto para un dominio seguro, se define un dominio seguro como un set de identidades que son gestionadas por el Identity Provider.

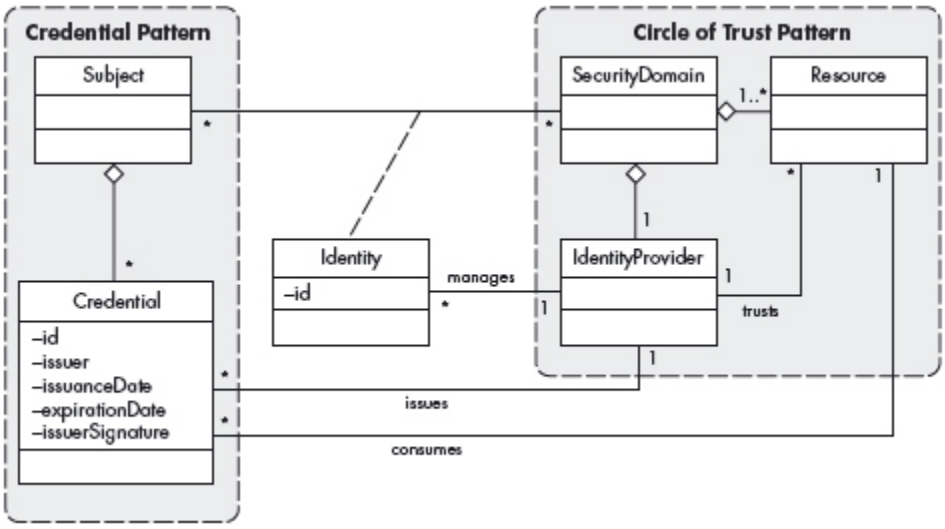


Figura 2. Diagrama de clase del patrón SPP Identity Provider

Relación entre ambos catálogos

A pesar de que el patrón SPP Identity Provider no se considera explícitamente como un patrón en el catálogo CSP, diversos patrones CSP utilizan un Identity Provider como componente propio: CSP Message Inspector, CSP Message Interceptor Gateway, CSP Secure Message Router, y CSP Single Sign-On delegator.

Por tanto, al formar parte de tantos patrones CSP, sí que parece tener sentido el considerarlo como un patrón en sí mismo.

Ventajas por catálogos

El patrón SPP Identity Provider ofrece un ejemplo de cómo podemos ver el patrón en situaciones reales, permitiendo entender más fácilmente la definición del patrón, el diagrama de clase explica fácilmente cómo se gestiona el patrón, mientras el patrón CSP Message Inspector y sus patrones relaciones, se describen sus tareas a través de los diagramas de secuencia representando el Identity provider en cada uno de los patrones.

A destacar en ambos catálogos

SPP

- El proveedor de identidad puede ser un cuello de botella en la red.
- El mantenimiento de costes es reducido.
- El sistema es consistente en términos de usuarios.

CSP

- El patrón CSP Message Inspector encapsula todos los mecanismos de seguridad a nivel de mensaje, facilitando la reutilización, esto facilita una solución común de reusabilidad para proteger múltiples puntos finales de servicio.
- El patrón CSP Message Inspector incorpora mecanismos de verificación seleccionando los elementos del mensaje con el propósito de identificar ataques a nivel del mensaje con requerimientos falsos.
- El patrón CSP Message Inspector ofrece extensibilidad permitiendo incorporar más mecanismos y funcionabilidades para proveer adherencia a nuevos estándares y mejorar el nivel de seguridad en el mensaje.

3.3 Identity Federation

Descripción

El patrón de Identity Federation (Figura 3) permite la formación de una identidad creada dinámicamente dentro de una federación de identidad que consiste en varios proveedores de servicios. Por lo tanto, la identidad y la información de seguridad en relación a un sujeto pueden

ser transmitidas de forma transparente al usuario entre los proveedores de servicios de diferentes dominios de seguridad.

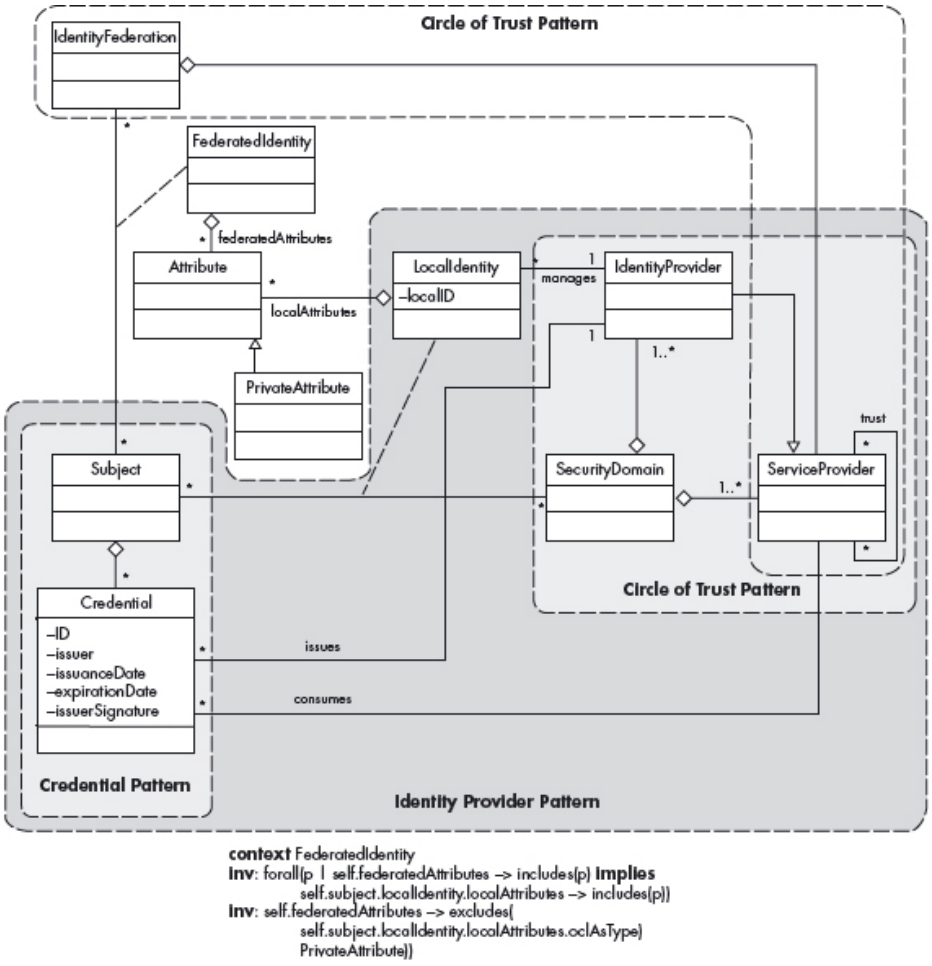


Figura 3. Diagrama de clase del patrón SPP Identity Federation

Relación entre catálogos

El patrón SPP Identity Federation es básicamente una abstracción de la federación de identidades que forma parte de Liberty Alliance, y como tal, no es considerado como patrón en CSP. No obstante, al considerar el catálogo CSP la federación de identidades Liberty Alliance, en cierta medida, sí que es considerado en dicho catálogo al igual que en el SAML.

Ventajas por catálogos

El patrón SPP Identity Federation muestra un diagrama de secuencia para el caso de uso de dos identidades federadas mientras en el catálogo CSP se utiliza SAML (Security Assertion

Markup Language) se puede ver una infraestructura basada en estándares permitiendo el SSO Single Sign – On sin requerir el uso de la arquitectura de seguridad de terceros, sin embargo no provee un mecanismo de autenticación para el usuario. SAML 1.1 agrega un soporte de identidad a la red definida por las especificaciones del Liberty Alliance.

A destacar en ambos catálogos

SPP

- Los proveedores de servicios necesitan tener algún tipo de acuerdo antes de que sus identidades puedan estar federadas, tienen que intercambiar las credenciales a través de algún canal externo de confianza y reconocerse entre sí.
- Incluso cuando la información sensible de un sujeto se clasifica como privada, un dominio de seguridad todavía puede revelar la información privada del sujeto a otras partes, violando la privacidad del sujeto.
- Un token de seguridad puede ser robado y presentado por un atacante, dando como resultado un robo de identidad. Esto puede mitigarse con el uso de las fechas de vencimiento e identificadores únicos para las credenciales.
- A pesar de una fecha de expiración y la característica única de un ID de la credencial, esto garantiza frescura. La revocación incondicional de una credencial no está direccionada en esta solución.
- Los sujetos pueden acceder a recursos dentro de la federación de identidad de una forma continua y segura sin re-autenticarse en cada nuevo dominio.
- Varias representaciones diferentes de la identidad del usuario pueden ser consolidadas bajo la misma identidad federada.
- Los sujetos pueden clasificar algunos de los atributos como ‘privado’. Por tanto un proveedor de identidad puede identificar que atributos debieran transmitirse a otros lugares y cuáles no.
- Parte de las credenciales de seguridad pueden ser encriptadas. De esta forma la privacidad del sujeto puede ser protegida.

- La credencial de seguridad puede ser firmada, así la integridad y la autenticidad es protegida. Los atacantes no pueden falsificar los tokens de seguridad o cambiar alguno de los atributos de los sujetos.
- Un sujeto puede acceder a un recurso de dominios de seguridad de forma anónima, ya que no todos los atributos de una identidad local (como un nombre) son requeridos para ser federados.

CSP

- La identidad federada extiende el uso de la identidad de la red dentro de una compañía o empresa para multiplicar entidades de negocio o infraestructuras de seguridad.
- La Identidad federada se refiere al uso de información de la identidad entre compañías y aplicaciones a través de infraestructuras de seguridad diferentes sobre la red.

3.4 Liberty Alliance Identity Federation

Descripción

El patrón SPP Liberty Alliance Identity Federation (Figura 4) permite la unión de las identidades a través de múltiples proveedores, siguiendo diferentes estándares.

Los proveedores de servicios como portales de internet, instituciones financieras entre otros, ofrecen servicios a los usuarios, estos proveedores gestionan un conjunto de atributos del usuario como nombre, fecha de nacimiento, número de identidad, que contribuyen a la identidad de los usuarios. Los servicios web tienen identidades que pueden ser usadas para acceder a otros servicios.

Entre los proveedores de servicios que comprenden una federación de identidad, al menos uno actúa como proveedor de identidad, el cual es el responsable de la gestión de la identidad federada. Una identidad federada es la composición de varias identidades locales. Por lo tanto, la información de la identidad sobre un usuario o servicio puede ser transmitida entre los proveedores de servicios de una manera que es transparente para los sujetos. En particular, su estado de autenticación se puede propagar para realizar un inicio de sesión único dentro de la federación de identidades. El sujeto tiene que indicar su consentimiento a fin de que cada una de las identidades locales pueda ser federada.

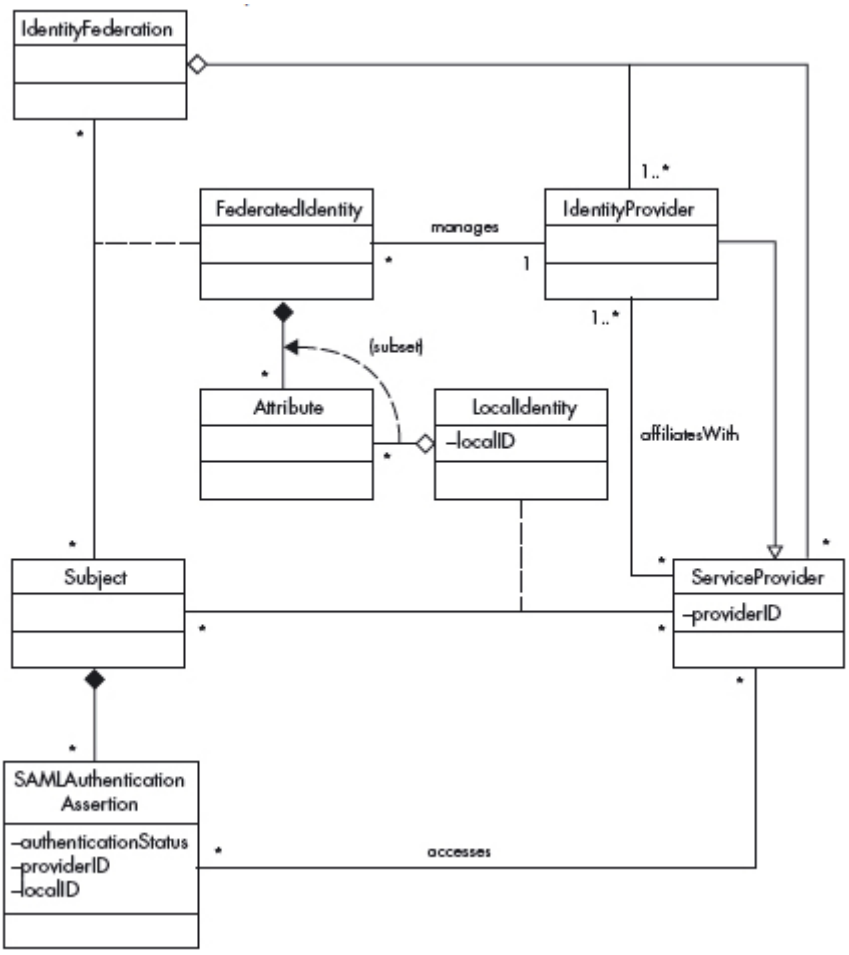


Figura 4. Diagrama de clase del patrón SPP Liberty Alliance Identity Federation

Relación entre catálogos

A pesar de que el catálogo CSP no considera la federación de identidades Liberty Alliance como un patrón en sí mismo, sí que lo describe con bastante profusión.

Es evidente que, a pesar de no considerarlo como un patrón, el catálogo CSP describe en mayor profundidad la federación de identidades Liberty Alliance y su relación con estándares para definir credenciales como SAML.

Ventajas por catálogos

El patrón SPP Liberty Alliance Identity Federation explica detalladamente su estructura, se describe un diagrama de clase y un diagrama de secuencia para el caso de uso federar dos identidades locales.

En el catálogo CSP se utiliza el SAML (Security Assertion Markup Language) permitiendo el SSO Single Sign – On sin requerir el uso de la arquitectura de seguridad de terceros. SAML 1.1 agrega soporte a la de identidad en una red definida por las especificaciones del Liberty Alliance Specifications. La descripción del patrón en el catálogo CSP es mucho más extensa. Quizás por esa razón no se considera un patrón en este catálogo.

A destacar en ambos catálogos

SPP

- Los proveedores de servicios necesitan tener algún tipo de acuerdo antes de que sus cuentas pueden estar federadas, tienen que intercambiar las credenciales a través de algún canal externo de confianza y reconocerse mutuamente.
- Tanto los sujetos como los prestadores de servicios de la federación de identidades necesitan confiar en el proveedor de identidad.
- Los sujetos pueden llevar a cabo las transacciones dentro de la federación de identidad de forma continua y segura.
- No es necesario re-autenticarse en cada nuevo dominio.

CSP

- El Liberty Alliance habilita a los clientes a proteger la privacidad y la seguridad de la información.
- Protege la privacidad y la seguridad de la identidad en la red sin la participación de terceros.

4. PATRONES DE AUTENTICACIÓN

4.1 Authenticator

Descripción

La función del patrón de autenticación SPP (Figura 5) es verificar si la identidad del usuario que intenta acceder al sistema es correcta, durante el proceso de autenticación, las aplicaciones transfieren las credenciales del usuario para comprobar su identidad, cuando el usuario solicita acceso a un recurso determinado. Las credenciales de usuario y los datos asociados deben mantenerse en privado y no deben ponerse a disposición de otros usuarios o aplicaciones coexistentes.

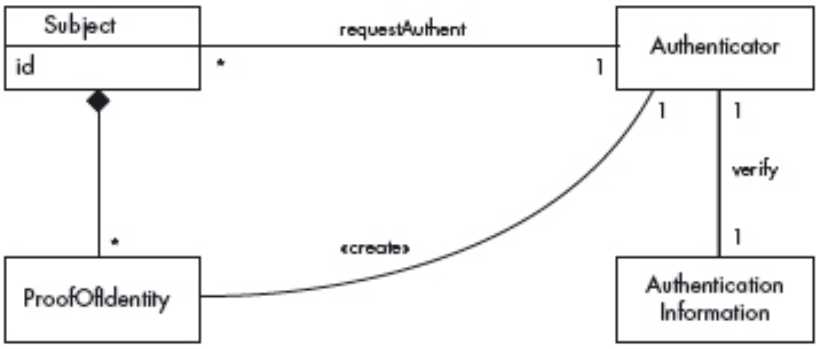


Figura 5. Diagrama de clase del patrón SPP Authenticator

Relación entre catálogos

Los patrones CSP Authentication Enforcer (Figura 6) y SPP Authenticator (Figura 5) muestran características semejantes bastantes cercanos en filosofía, debido a que su objetivo es centralizar el acceso como un punto único para recibir las interacciones del sujeto con el sistema, se aplica un protocolo para verificar la identidad del sujeto y este puede ser simple o complejo dependiendo de las necesidades de la aplicación.

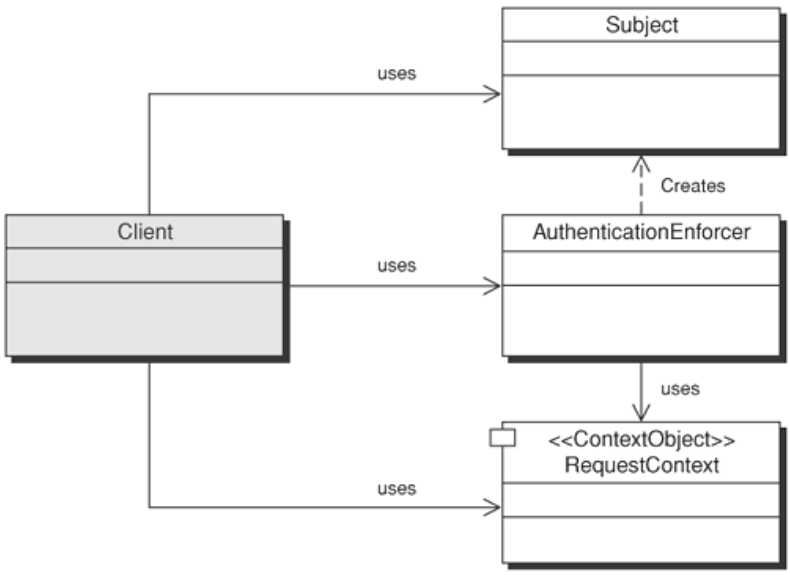


Figura 6. Diagrama de clase del patrón CSP Authentication Enforcer

Ventajas por catálogos

Una de las ventajas que ofrece el patrón de autenticación SPP con respecto al patrón CSP es que está relacionado con los patrones Remote Authenticator/Authorizer Pattern y Credential Pattern que pueden ser utilizados para realizar la autorización si se incluyen los permisos del usuario. Para sistemas distribuidos donde los usuarios pueden tener acceso a varios sistemas, es conveniente utilizar un Single Sign On (SSO).

Sin embargo, el patrón CSP define tres estrategias para implementar una autenticación: la estrategia del contenedor autenticado, dar autenticación y el módulo de login JAAS.

A destacar en ambos catálogos

SPP

- Dependiendo del protocolo de autenticación utilizado este puede ser simple o complejo. Si es complejo, los usuarios gastan tiempo y se pueden irritar.
- La complejidad en general y el costo del sistema incrementan el nivel de seguridad.

CSP

- Mantener todos los códigos del User Login en clases separadas de la clase de aplicación ayuda a gestionar el cambio de mecanismos de autenticación a futuro.

- Elegir la mejor seguridad. La autenticación del HTTP básico es usualmente la más vulnerable a ataques.

4.2 Remote Authenticator/Authorizer

Descripción

Este patrón proporciona la autenticación y la autorización cuando se accede a recursos compartidos en sistemas distribuidos, un sistema con autenticación centralizada es más fácil de manejar y potencialmente más seguro, pero no es lo suficientemente flexible como para los sistemas distribuidos (Figura 7).

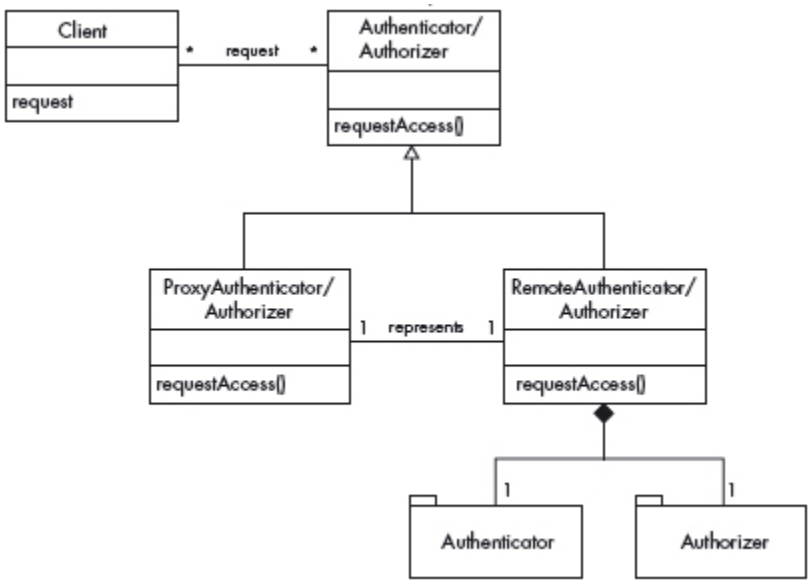


Figura 7. Diagrama de clase del patrón SPP Remote Authenticator/Authorizer Pattern

Relación entre catálogos

El catálogo CSP ofrece un patrón de Authorization Enforcer que provee un punto centralizado para recursos autorizados y crea una clase para manejar las autorizaciones de requerimiento HTTP, mientras que el patrón SPP Remote Authentication/Authorizer guarda la información de autenticación y autorización en múltiples ubicaciones. Por tanto un SPP Remote Authentication/Authorizer es la versión remota de un CSP Authentication Enforcer.

Ventajas por catálogos

El patrón SPP Authenticator /Authorizer muestra un ejemplo de autenticación remota en un sistema de servicio de usuario RADIUS utilizando una mejora en la respuesta, mientras el patrón CSP Authorization Enforcer describe un ejemplo de código para una estrategia de autorización utilizando la librería permisos JSP.

A destacar en ambos catálogos.

SPP

- El roaming autoriza a dos o más entidades administrativas a permitir a cada usuario sintonizar a cualquiera en una red de entidades para un servicio.
- Guarda el login del usuario y los permisos de acceso en una sola ubicación hace que sea más fácil y segura de mantener.
- Configura un punto de entrada único que pueda redireccionarse hacía el servidor correcto de forma transparente para el usuario, donde el login y la información de acceso del usuario puedan ser validados. Se puede realizar el redireccionamiento utilizando un servidor especializado en Autenticación/Autorización.
- El almacenamiento de autenticación de usuario y la información de autorización en varios lugares hace que sea redundante, difícil de administrar y propenso a inconsistencias.
- Aunque la información de autenticación se puede almacenar en cualquier lugar, este lugar debe ser transparente para los usuarios.
- Los mensajes adicionales aumentan la carga, lo que reduce el rendimiento de peticiones sencillas. Esto no es un problema si los accesos remotos no son muy frecuentes.
- El sistema es mucho más complejo que un sistema que valida directamente a los clientes.

CSP

- Provee puntos de enfoque en revisiones de control de acceso, eliminando el riesgo de código repetitivo.
- Permite la reutilización a través de mecanismos de encapsulación de control de acceso a través de interfaces comunes.
- Provee un control de acceso de grano fino.

4.3 Credential

Descripción

El patrón SPP Credential (Figura 8) proporciona un medio seguro para la autenticación de registros y la autorización de información en sistemas distribuidos, puede considerarse una prueba de identidad y de autorización.

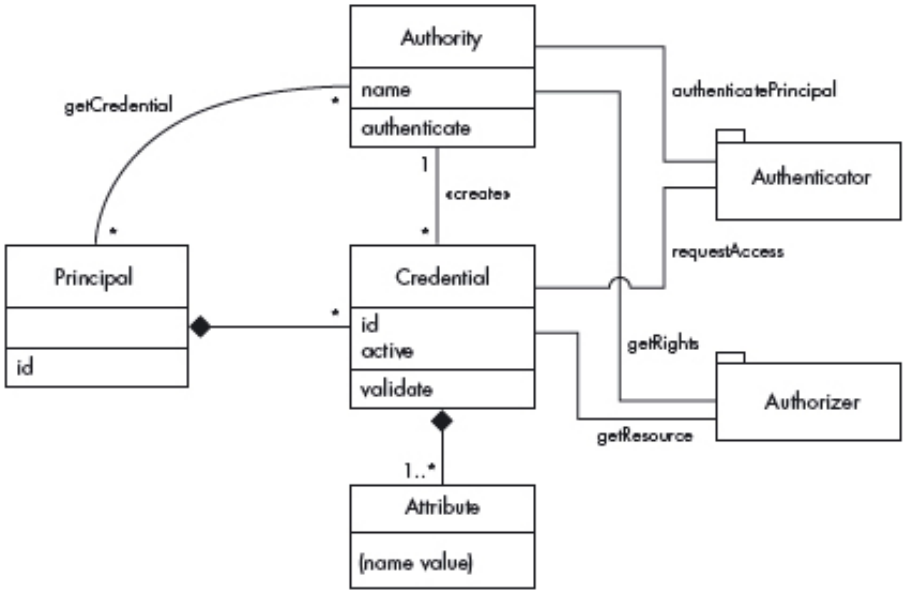


Figura 8. Diagrama de clase del patrón SPP Credential

Relación entre catálogos

A pesar de que el catálogo CSP no considera explícitamente este patrón, sí que lo menciona de manera implícita tanto el patrón CSP Authentication Enforcer como en el CSP Authorization Enforcer. Por este motivo se ha considerado un patrón común a ambos catálogos.

Ventajas por catálogos

El patrón SPP Credential describe los cuatro casos de uso primarios y el patrón CSP Authentication Enforcer define tres estrategias para implementar una autenticación: la estrategia del contenedor autenticado, dar autenticación y el módulo de login JAAS.

En el caso del patrón CSP Authorization Enforcer, autorización se enfoca creando un control de acceso mejorado utilizando el Standard Java Security clases API. Para ello utiliza un módulo de login en JAAS, que fuerza el control de acceso siguiendo las políticas de JAAS como medio de autorización. El catálogo CSP también incluye ejemplos de código de programación utilizando como estrategia la autorización programada.

A destacar en ambos catálogos

SPP

- Puede ser difícil encontrar una autoridad en la que se pueda confiar. Esto puede resolverse con cadenas (árboles) de credenciales, por lo cual una autoridad certifica a otra autoridad.
- Hacer credenciales a prueba de manipulaciones incurre en un tiempo extra y es complejo.
- El almacenamiento de credenciales que se utiliza fuera de los sistemas deja a los mecanismos de autorización abiertos a ataques sin conexión.
- La autenticación de grano fino e información de autorización puede ser grabada de forma persistente.
- Una credencial de una autoridad confiable puede ser considerada prueba de identidad y de autorización.
- Es posible proteger credenciales usando encriptación u otros medios.

CSP

- Mantener todos los códigos del User Login en clases separadas de la clase de aplicación, ayuda a gestionar el cambio de mecanismos de autenticación a futuro.
- Elegir la mejor seguridad. La autenticación del HTTP básico es usualmente la más vulnerable a ataques.
- Provee puntos de enfoque en revisiones de control de acceso, eliminando el riesgo de código repetitivo.
- Permite la reutilización a través de mecanismos de encapsulación de control de acceso a través de interfaces comunes.
- Provee un control de acceso de grano fino.

5. PATRONES DE CONTROL DE ACCESO

5.1 Authorization

Descripción

Este patrón provee una encapsulación genérica de los mecanismos de autorización para definir una forma estándar de controlar el acceso a aplicaciones basadas en web (Figura 9). En varias aplicaciones existen diferentes tipos de usuarios y roles, y cada uno de ellos requiere acceso basándose en un criterio definido por las reglas del negocio y las políticas específicas de un recurso. La aplicación debe forzar a que el usuario tenga disponible el acceso solo a los recursos que esté autorizado.

El patrón SPP Authorization conocido también como una Matriz de Acceso, describe quien está autorizado para acceder a determinados recursos del sistema, por cada sujeto activo se verifica a que recursos puede acceder y que puede hacer con estos.

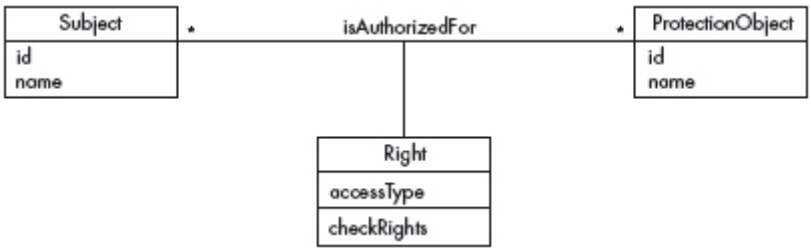


Figura 9. Diagrama de clase del patrón SPP Authorization Pattern

Relación entre catálogos

El patrón SPP Authorization es parte de la implementación de un CSP Authentication Enforcer (Figura 10). Se utiliza para comprobar los permisos de acceso de cada sujeto y, por tanto, es parte de la implementación de un CSP Authorization Enforcer. El patrón CSP Authorization Enforcer propone la creación de un controlador de acceso que lleva a cabo las comprobaciones de autorización utilizando las clases API de Java Standard Security y el patrón

SPP Authorization propone que para cada sujeto activo que pueda acceder a recursos, objetos o protección de objetos, se indique a cuales recursos puede tener acceso.

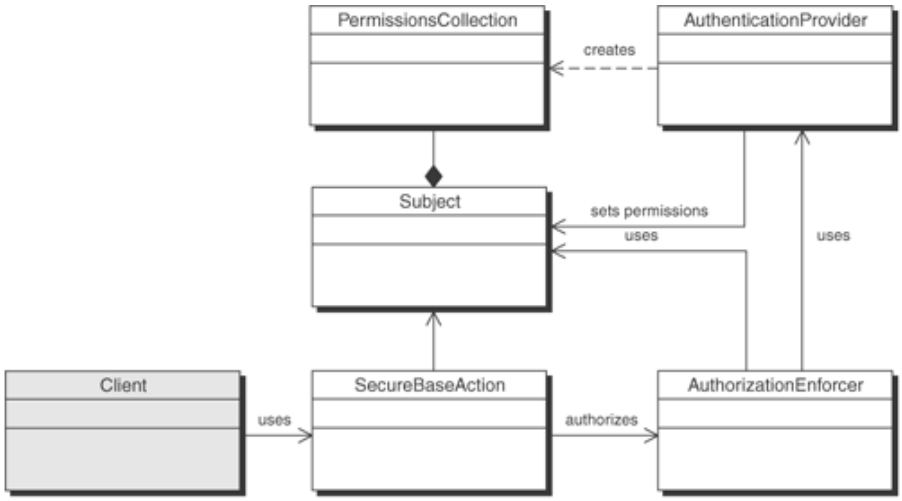


Figura 10. Diagrama de clase del patrón CSP Authorization Enforcer

Ventajas por catálogos

Una de las ventajas que ofrece el patrón SPP Authorization con respecto al patrón CSP Authorization Enforcer es que está relacionado con los patrones de control de acceso Role-Based Access control y Reference Monitor quienes son participantes activos en el proceso de la autorización.

El patrón CSP Authorization Enforcer enfoca la autorización en la creación de un control de acceso mejorado utilizando el Standard Java Security API clases y un módulo de login en JASS para forzar el control de acceso siguiendo las políticas de JAAS como medio de autorización. El catálogo CSP incluye ejemplos de código de programación utilizando como estrategia la autorización programada.

A destacar en ambos catálogos

SPP

- El patrón SPP Authorization se puede aplicar a cualquier tipo de recurso. Los sujetos pueden ejecutar procesos, usuarios, roles, grupos de usuario.

- Los objetos de protección pueden ser transacciones, datos, áreas de memoria, unidades de I/O, archivos u otros recursos.
- Los tipos de acceso son definidos individualmente. Es fácil agregar o remover autorizaciones.
- Las reglas de autorización pueden ser protegidas de la misma forma que otras estructuras de datos como las relaciones.
- Las reglas se pueden aplicar a cualquier tipo de sujeto u objeto.
- Si existen muchos usuarios u objetos, debe ser escrito un número largo de reglas. Esto hace más difícil la administración y se tiende a cometer errores.
- Definir las reglas de autorización no es suficiente, también es necesario forzar el mecanismo.

CSP

- Provee puntos de enfoque en revisiones de control de acceso, eliminando el riesgo de código repetitivo.
- Permite la reutilización a través de mecanismos de encapsulación del control de acceso a través de interfaces comunes.
- Provee un control de acceso de grano fino.

5.2 Role-Based Access Control

Descripción

El patrón SPP Role –Based Access Control (Figura 11) describe como asignar los permisos basados en las funciones o tareas que realiza el usuario en un ambiente donde se requiere el control de acceso a los recursos del sistema.

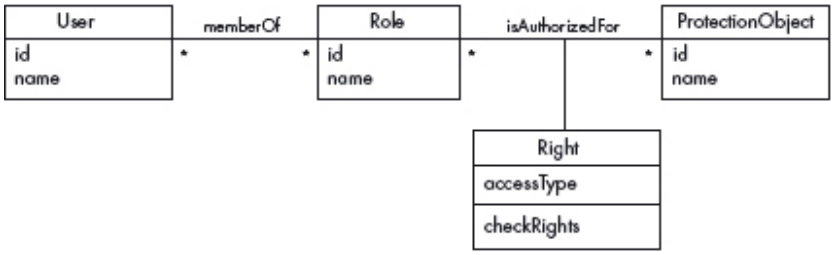


Figura 11. Diagrama de clase del patrón SPP Role-Based Access Control

Relación entre catálogos

El patrón SPP Role – Based Access Control, al ser un SPP Authorization, forma parte de un patrón CSP Authorization Enforcer.

Ventajas por catálogos

Una de las ventajas que ofrece el patrón SPP Role –Based Access Control con respecto al catálogo CSP es que es un elemento del control de acceso en conjunto con el patrón Authorization y describe un diagrama de clase del modelo RBAC extendido, mientras el patrón CSP Authorization Enforcer enfoca la autorización en la creación de un control de acceso mejorado utilizando el Standard Java Security API clases y un módulo de login en JASS para forzar el control de acceso siguiendo las políticas de JAAS como medio de autorización.

A destacar en ambos catálogos

SPP

- El número de permisos individuales puede ser muy largo. Conceder estos permisos a usuarios individuales puede requerir varias reglas de autorización, haciendo difícil para los administradores mantener el seguimiento de estas reglas y asociar el significado de la semántica.
- Algunas instituciones pueden no tener definidos los roles en sus organizaciones lo cual genera más trabajo, ya que estos roles deben estar definidos.
- Los permisos dados para un rol deben corresponder a una tarea.
- Permite a los administradores reducir la complejidad de la seguridad, ya que si hay muchos más usuarios que roles, el número de roles se vuelve mucho más pequeño.
- Las políticas de la organización en relación a las funciones de trabajo pueden ser reflejadas en la definición de roles y la asignación de usuarios a roles.
- En casos que se requiera solo la manipulación de las asociaciones entre los usuarios y roles, basta con acomodar a los usuarios juntos.
- Los roles pueden ser estructurados dentro de una jerarquía flexible y de roles reducidos.
- Los usuarios pueden activar más de una sesión en un tiempo flexible y funcional.

CSP

- Provee puntos de enfoque en las revisiones de control de acceso, eliminando el riesgo de código repetitivo.
- Permite la reutilización a través de mecanismos de encapsulación del control de acceso a través de interfaces comunes.
- Provee un control de acceso de grano fino.

5.3 Multilevel Security

Descripción

El patrón SPP Multilevel Security (Figura 12) describe como categorizar la información sensible y prevenir su divulgación, este describe como asignar una clasificación (autorizaciones) a usuarios y clasificaciones (niveles de sensibilidad) a los datos además de como separar unidades organizacionales diferentes en categorías. El acceso de usuarios a datos está basado en políticas, mientras los cambios en las clasificaciones están desarrollados por procesos de confianza que permiten transgredir las políticas.

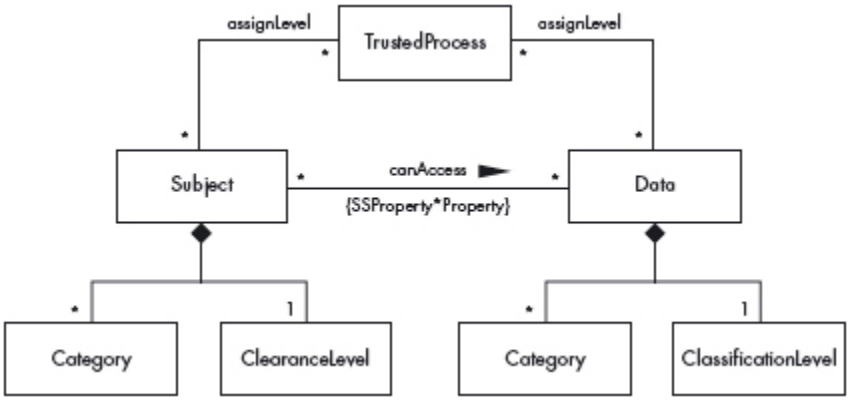


Figura 12. Diagrama de clase del patrón SPP Multilevel Security

Relación entre catálogos

El patrón SPP Multilevel Security, al ser un SPP Authorization, forma parte de un CSP Authorization Enforcer.

Ventajas por catálogos

Según el análisis realizado en este trabajo de investigación, una de las ventajas que ofrece el patrón SPP Multilevel Security con respecto al CSP es que es un elemento del control de acceso en conjunto con el patrón SPP Authorization, indica ejemplos de usos conocidos en este patrón, mientras el patrón CSP Authorization Enforcer enfoca la autorización en la creación de un control de acceso mejorado utilizando el Standard Java Security API clases y un módulo de login en JASS para forzar el control de acceso siguiendo las políticas de JAAS como medio de autorización.

A destacar en ambos catálogos

SPP

- Se necesitan programas fiables para asignar a los usuarios y datos clasificados.
- Este modelo se puede manejar discretamente y prevenir la fuga de información.
- Es necesario un modelo dual que maneje también la integridad.
- La clasificación de usuarios y datos es relativamente simple y puede seguir las políticas de organización.
- Este modelo de patrón al ser seguro puede ser probado.
- El patrón es útil en procesos aislados y dominios ejecutados.

CSP

- Provee puntos de enfoque en revisiones de control de acceso, eliminando el riesgo de código repetitivo.
- Permite la reutilización a través de mecanismos de encapsulación del control de acceso a través de interfaces comunes.
- Provee un control de acceso de grano fino.

5.4 Policy-Based Access Control

Descripción

El patrón SPP Policy –Based Access Control (Figura 13) está basado en políticas que describen como decidir si un sujeto está autorizado o no a acceder a un objeto según las políticas definidas en el Repositorio Central de Políticas.

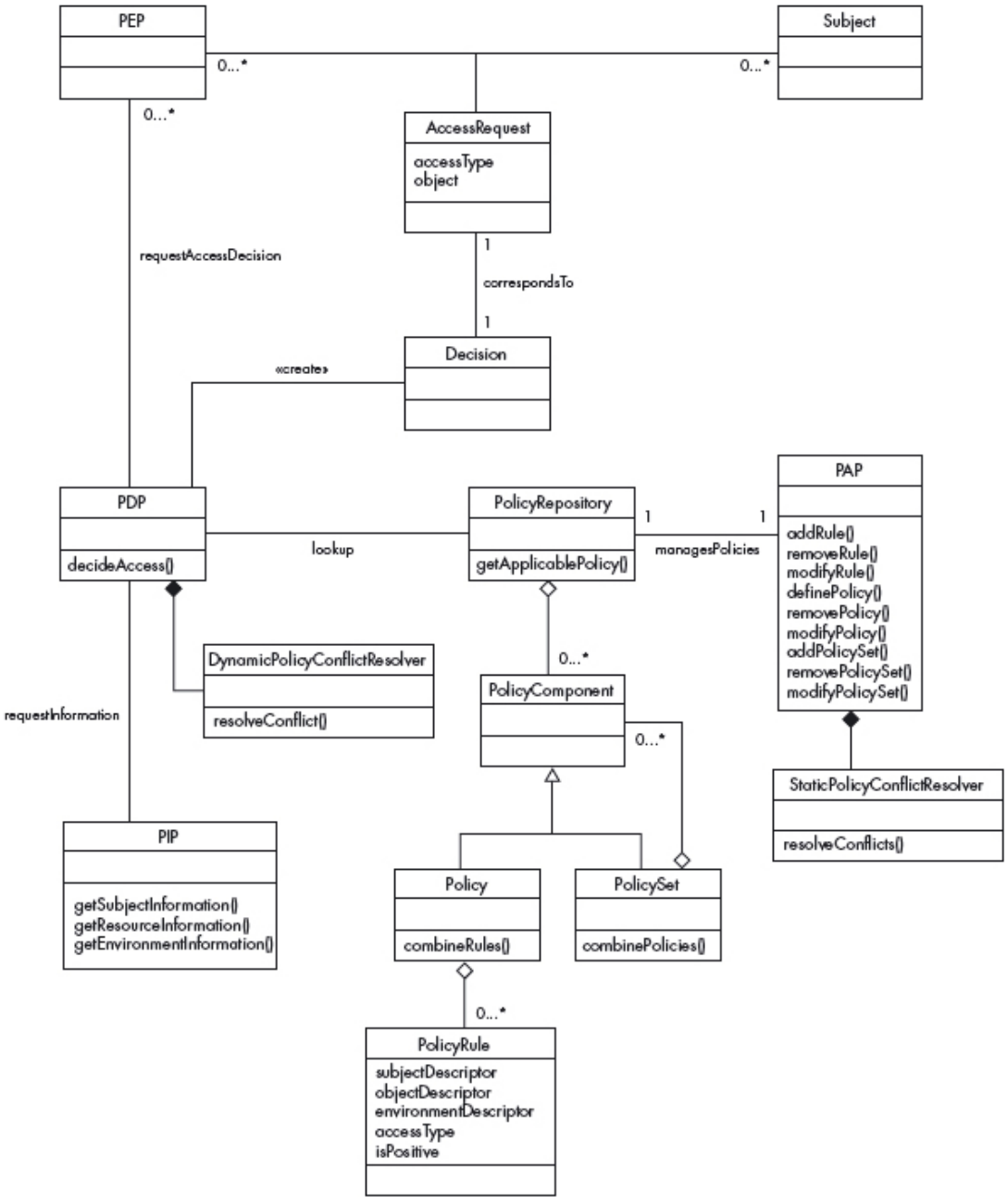


Figura 13. Diagrama de clase del patrón SPP Policy-Based Access Control

Relación entre catálogos

El patrón SPP Policy – Based Access Control, al ser un SPP Authorization, forma parte de un CSP Authorization Enforcer.

Ventajas por catálogos

Una de las ventajas que ofrece el patrón SPP Policy –Based Accesss Control con respecto al CSP es que es un elemento del control de acceso en conjunto con el patrón Authorization y se utiliza XACML y XML para expresar reglas de autorización y decidir los accesos, mientras el patrón CSP Authorization Enforcer enfoca la autorización en la creación de un control de acceso mejorado utilizando el Standard Java Security API clases y un módulo de login en JASS para forzar el control de acceso siguiendo las políticas de JAAS como medio de autorización.

A destacar en ambos catálogos

SPP

- El patrón puede soportar la matriz de acceso, RBAC o modelos multinivel para el control de acceso.
- Existen menos probabilidades de que puedan ser desarrollados accesos ilegales.

CSP

- Provee puntos de enfoque en revisiones de control de acceso, eliminando el riesgo de código repetitivo.
- Permite la reutilización a través de mecanismos de encapsulación del control de acceso a través de interfaces comunes.
- Provee un control de acceso de grano fino.

5.5 Access Control List (ACL)

Descripción

El patrón SPP Access Control List, ACL, (Figura 14) permite el acceso controlado a objetos indicando qué sujeto puede acceder a un objeto y de qué forma, en el sistema no todos los sujetos pueden acceder a cualquier objeto. Los permisos de acceso están definidos y pueden

ser modelados como una matriz de acceso, en la cual las celdas representan a los sujetos y cada columna representa un objeto.

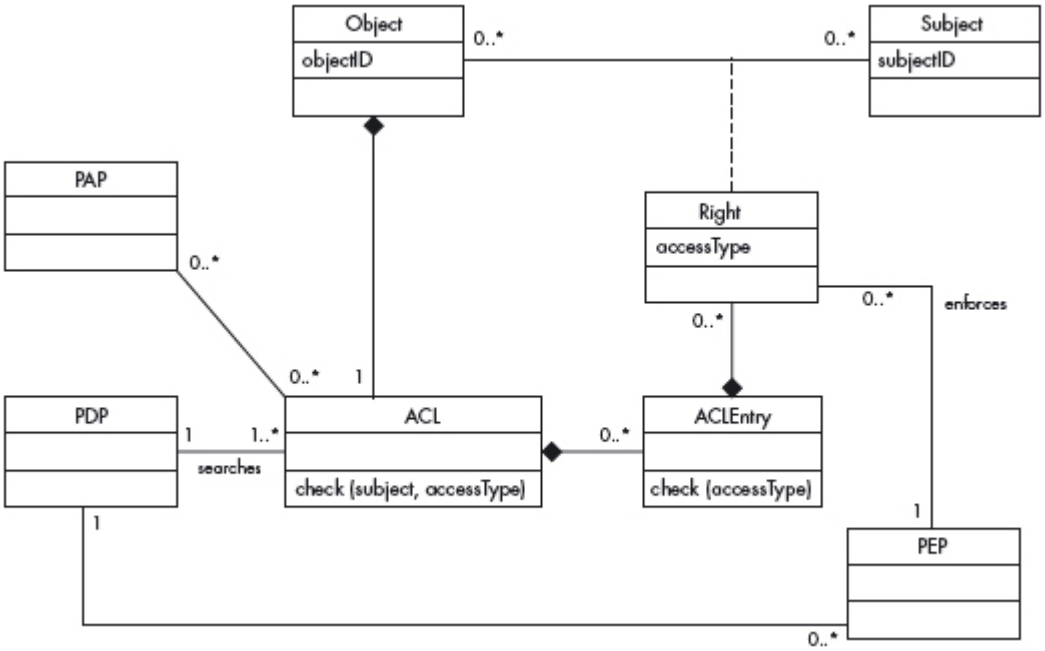


Figura 14. Diagrama de clase del patrón SPP Access Control List (ACL)

Relación entre catálogos

El patrón SPP Access Control List (ACL), al ser un SPP Authorization, forma parte de un CSP Authorization Enforcer.

Ventajas por catálogos

Una de las ventajas que ofrece el patrón SPP Access Control List (ACL) con respecto al CSP es que es un elemento del control de acceso relacionado con el patrón Authorization. Los modelos de la matriz de acceso y el RBAC SPP pueden ser implementados utilizando ACLs mientras el patrón CSP enfoca la autorización en la creación de un control de acceso mejorado utilizando el Standard Java Security API clases y un módulo de login en JASS para forzar el control de acceso siguiendo las políticas de JAAS como medio de autorización.

A destacar en ambos catálogos

SPP

- La administración de sujetos se hace más difícil. La eliminación de un objeto puede implicar un análisis de todas las ACL.
- El tiempo utilizado accediendo a un ACL es menor que el tiempo que se utiliza accediendo a una matriz centralizada.
- El acceso no autorizado a objetos por requerimiento de olvido de nombre de sujetos legítimos no es posible, porque los requerimientos son solo de usuarios autenticados.

CSP

- Provee puntos de enfoque en revisiones de control de acceso, eliminando el riesgo de código repetitivo.
- Permite la reutilización a través de mecanismos de encapsulación del control de acceso a través de interfaces comunes.
- Provee un control de acceso de grano fino.

5.6 Capability

Descripción

El patrón SPP Capability (Figura 15) permite el acceso controlado a objetos que proveen una credencial o ticket a un sujeto, permitiéndole el acceso a un objeto de forma específica. Este patrón emite un conjunto de capacidades para cada sujeto. Las capacidades deben ser implementadas de forma que permita a las políticas de decisión verificar su autenticidad y que un usuario malicioso no pueda falsificarla.

En sistemas distribuidos donde el acceso a recursos debe ser controlado, cualquier sujeto no puede acceder a cualquier objeto, los permisos de acceso son definidos y pueden ser modelados como una matriz de acceso, en la cual cada celda representa el sujeto y cada columna representa un objeto.

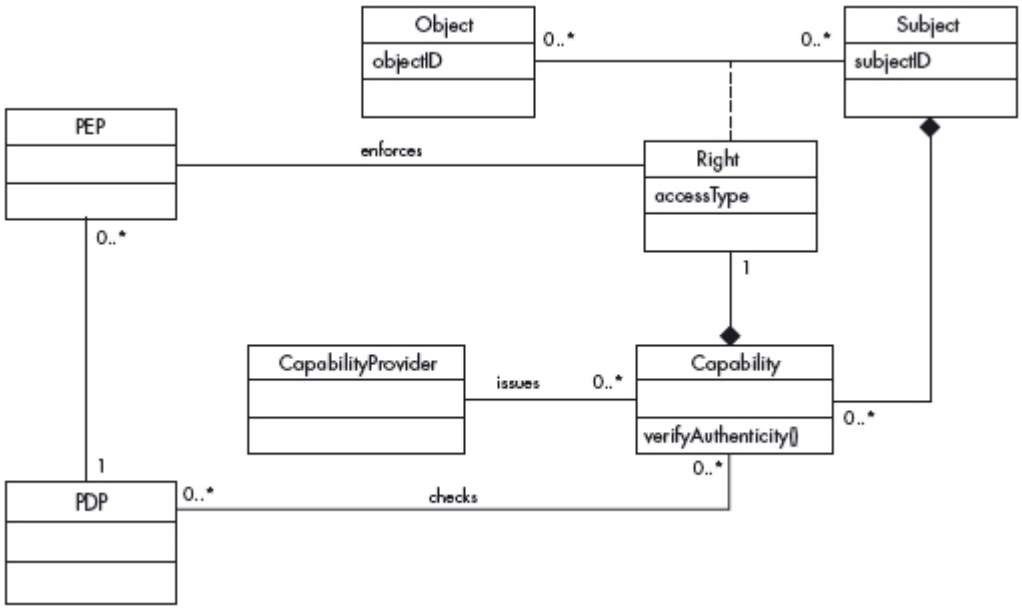


Figura 15. Diagrama de clase del patrón SPP Capability

Relación entre catálogos

El patrón SPP Capability, al ser un SPP Authorization, forma parte de un CSP Authorization Enforcer.

Ventajas por catálogos

Una de las ventajas que ofrece el patrón SPP Capability con respecto al CSP es que es un elemento del control de acceso en conjunto con los patrones Authorization y Access Control List (ACL). Además, el catálogo SPP describe la implementación en hardware y software, mientras el patrón CSP Authorization Enforcer enfoca la autorización en la creación de un control de acceso mejorado utilizando el Standard Java Security API clases y un módulo de login en JASS para forzar el control de acceso siguiendo las políticas de JAAS como medio de autorización.

A destacar en ambos catálogos

SPP

- El tiempo utilizado para acceder a una autorización es menor al tiempo que pudiera haber sido utilizado, porque la capacidad es enviada junto al requerimiento, buscando en toda la matriz o en la lista de control de acceso (ACL).
- Es difícil para los usuarios maliciosos falsificar o modificar capacidades. La administración de los objetos es más difícil, agregar un objeto implica la emisión de capacidades a cada usuario autorizado. En la práctica, la matriz no es muy compleja. Los sujetos tienen derechos sobre algunos objetos y por lo tanto la mayoría de las entradas están vacías.

CSP

- Provee puntos de enfoque en revisiones de control de acceso, eliminando el riesgo de código repetitivo.
- Permite la reutilización a través de mecanismos de encapsulación del control de acceso a través de interfaces comunes.
- Provee un control de acceso de grano fino.

5.7 Reified Reference Monitor

Descripción

El patrón SPP Reified Reference Monitor (Figura 16) describe como forzar las autorizaciones cuando un sujeto requiere un objeto protegido y provee al sujeto con una decisión. Este patrón describe cómo definir un proceso abstracto que intercepta todos los requerimientos de recursos y verifica que se cumpla con las autorizaciones.

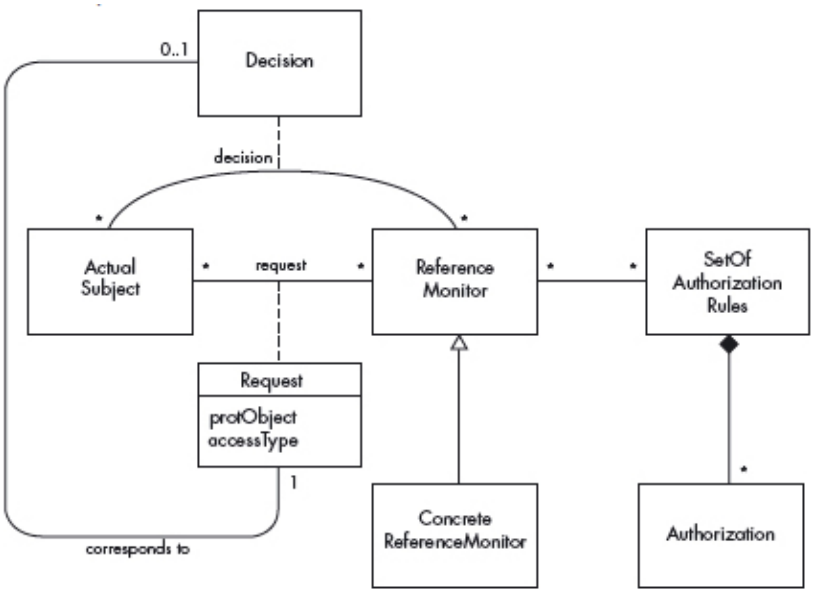


Figura 16. Diagrama de clase del patrón SPP Reified Reference Monitor

Relación entre catálogos

A pesar de no estar identificado como patrón en el catálogo CSP un patrón SPP Reified Reference Monitor es el núcleo de un CSP Authorization Enforcer, ya que garantiza que cuando un sujeto accede a un recurso tiene el preceptivo derecho. Cabe destacar que a pesar de que también es conocido como un Intercepting Filter, Application Controller no tiene relación con un CJ2EE Intercepting Filter ni un CSP Intercepting Validator.

Ventajas por catálogos

Una de las ventajas que ofrece el patrón SPP Reified Reference Monitor con respecto al CSP es que es un elemento del control de acceso en conjunto con el patrón SPP Authorization y define el concepto de Reference Monitor, mientras el patrón CSP Authorization Enforcer enfoca la autorización en la creación de un control de acceso mejorado utilizando el Standard Java Security API clases y un módulo de login en JASS para forzar el control de acceso siguiendo las políticas de JAAS como medio de autorización.

A destacar en ambos catálogos

SPP

- Se necesitan las implementaciones específicas (monitores de referencia en concreto) para cada tipo de recurso. La comprobación de cada solicitud y la toma de decisiones puede provocar una disminución del rendimiento.
- Es necesario mantener las decisiones en la memoria, de modo que cuando el sujeto pide el mismo objeto, se conoce la decisión y no se compromete el rendimiento.
- Si todos los requerimientos son interceptados, se puede asegurar que se cumplirá con las reglas.
- El sujeto tiene un mejor conocimiento de las decisiones hechas por el monitor de referencia para aprobar o negar este requerimiento.
- La implementación no es limitada utilizando el proceso de abstracción.

CSP

- Provee puntos de enfoque en revisiones de control de acceso, eliminando el riesgo de código repetitivo.
- Permite la reutilización a través de mecanismos de encapsulación del control de acceso a través de interfaces comunes.
- Provee un control de acceso de grano fino.

5.8 Controlled Access Session

Descripción

El Patrón SPP Controlled Access Session (Figura 17) describe como proveer un contexto en el cual un sujeto (usuario, sistema) puede acceder a recursos con diferentes permisos sin necesidad de re-autenticarse cada vez que se accede a un nuevo recurso.

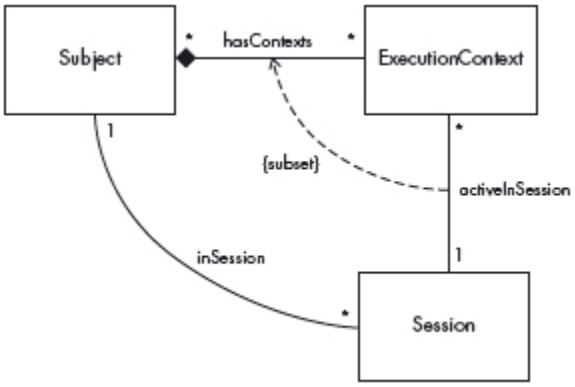


Figura 17. Diagrama de clase del patrón SPP Controlled Access Session

Relación entre catálogos

A pesar de que el patrón SPP Controlled Access Session no está explícitamente identificado en el catálogo CSP, es el mecanismo utilizado por un CSP Authorization Enforcer para garantizar acceso a distintas páginas, una vez autenticado el usuario.

Ventajas por catálogos

Una de las ventajas que ofrece el patrón SPP Controlled Access Session con respecto al CSP es que es un elemento del control de acceso en conjunto con el patrón SPP Authorization y SPP Session– Based Role Based Access Control y describe los casos conocidos actualmente. Sin embargo, en el catálogo CSP no es considerado como patrón y se supone implícito a un CSP Authorization Enforcer.

A destacar en ambos catálogos

SPP

- Si fuese necesario aplicar un acceso de grano fino, este puede ser ineficiente si hay que incluir muchos contextos y desarrollar actividades complejas.
- Usar sesiones puede ser confuso para otros usuarios.
- Algunas funciones pueden ser realizadas implícitamente en una sesión.
- Una vez que el sujeto empieza una sesión no tiene que re-autenticarse: este estatus se mantiene en una sesión.

- Se puede excluir combinaciones de contexto que pueden resultar en posibles violaciones de acceso o conflicto de intereses.

CSP

- Los mismos que el CSP Authorization Enforcer.

5.9 Session-Based Role-Based Access Control

Descripción

El patrón SPP Session- Based Role –Based Access Control (Figura 18) describe el acceso a recursos basados en el rol de un sujeto y limita los permisos que pueden ser aplicados en un tiempo dado, basándose en los roles definidos por la sesión de acceso.

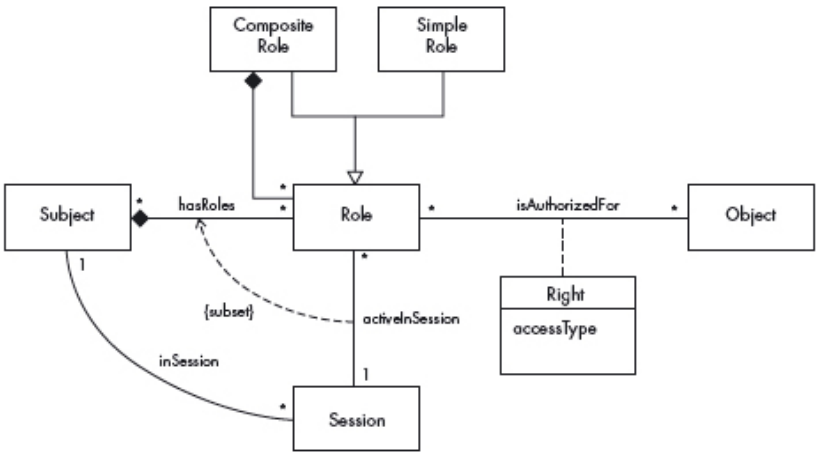


Figura 18. Diagrama de clase del patrón SPP Session-Based Role-Based Access Control

Relación entre catálogos

El SPP Session-Based Role-Based Access Control es una mezcla del SPP Role-Based Access Control y el SPP Controlled Access Session, por tanto, al igual que éstos, puede ser utilizado por un CSP Authorization Enforcer.

Ventajas por catálogos

El patrón SPP Session –Based Role – Based Access Control describe los pasos para una implementación real en un ejemplo, mientras, el patrón CSP Authorization Enforcer enfoca la

autorización en la creación de un control de acceso mejorado utilizando el Standard Java Security API clases y un módulo de login en JASS para forzar el control de acceso siguiendo las políticas de JAAS como medio de autorización.

A destacar en ambos catálogos

SPP

- Los derechos de grano fino pueden ser asignados a los roles para forzar las políticas conocidas.
- Requiere complejidad adicional para definir qué roles pueden ser utilizados juntos y cuáles deben ser mutuamente exclusivos.
- Si se tienen que utilizar varios roles para desarrollar su trabajo, puede crear confusión en los usuarios.
- Si hay muchos usuarios u objetos, se debe crear un gran número de normas. Esto hace que la administración sea difícil y sea propensa a errores.
- Puede ser difícil para el administrador de seguridad percibir que un tema determinado necesita un permiso o las implicaciones de una nueva regla. Además, no existe una relación semántica entre sujetos y objetos.
- Definir las reglas de autorización no es suficiente. También es necesario un mecanismo de aplicación.

CSP

- La autorización de grano fino le permite proteger adecuadamente la aplicación sin imponer una mejora que pudiera exponer innecesariamente a vulnerabilidades de seguridad.
- Provee puntos de enfoque en revisiones de control de acceso, eliminando el riesgo de código repetitivo.
- Permite el reúso a través de mecanismos de encapsulación del control de acceso a través de interfaces comunes.
- Las sesiones pueden incluir todo los roles necesarios para los sujetos autorizados de algunas tareas.

- Los usuarios pueden activar más de una sesión a tiempo para la flexibilidad funcional (algunas tareas pueden requerir múltiples roles).

5.10 Security Logger and Auditor

Descripción

Describe cómo realizar un seguimiento de las acciones sensibles a la seguridad de los usuarios con el fin de determinar quién hizo qué y cuándo se hizo (Figura 19). Proporciona un control centralizado de la funcionalidad del registro que se puede utilizar en varios lugares a lo largo de la solicitud de la aplicación con fines de auditoría.

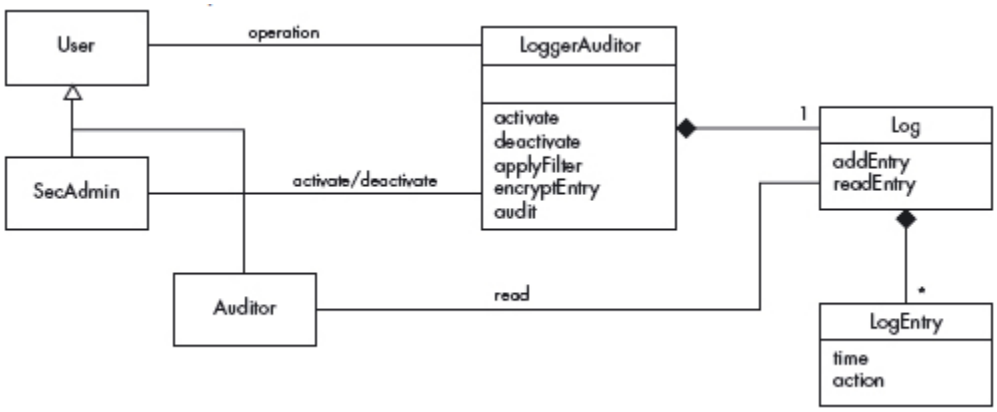


Figura 19. Diagrama de clase del patrón SPP Security Logger and Auditor

Relación entre catálogos

Los patrones SPP Security Logger and Auditor y CSP Secure Logger (Figura 20) muestran características semejantes debido a que su objetivo principal es mantener un registro de seguridad de forma controlada y centralizada, donde los usuarios autorizados puedan acceder y se puedan realizar auditorías a estos registros. Por tanto, CSP Secure Logger es un SPP Security Logger and Auditor y un CSP Audit Interceptor lo utilizaría.

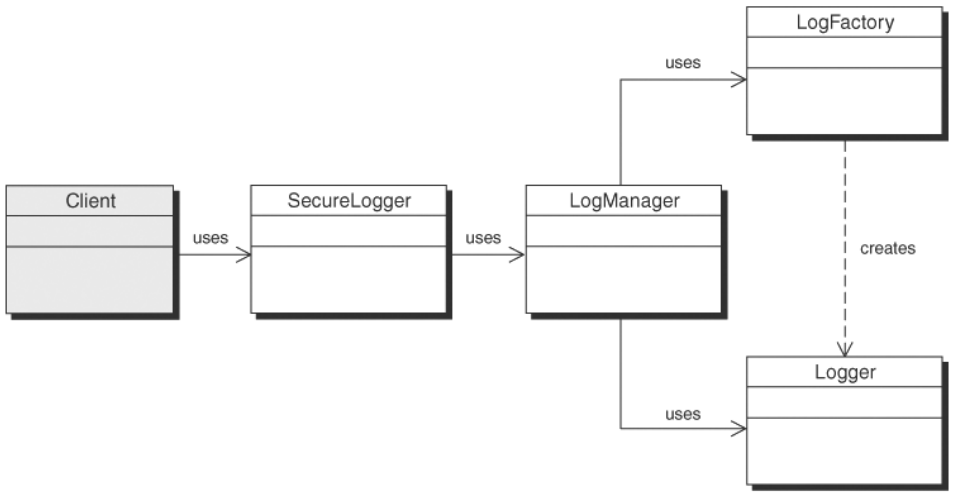


Figura 20. Diagrama de clase del patrón CSP Secure Logger Pattern

Ventajas por catálogos

Una de las ventajas que ofrece el patrón CSP Secure Logger es una mayor explicación de cómo y de qué forma se puede implementar este patrón para que sea más útil, además indica la relación que existe con el patrón CSP Audit Interceptor de la capa de negocio. Sin embargo, el patrón SPP Security Logger and Auditor indica muy superficialmente la implementación del patrón.

A destacar en ambos catálogos

SPP

- Los diseñadores de software deben tomar una decisión en cuanto a la granularidad con la que se registran los objetos. Así, hay un equilibrio entre la seguridad y el rendimiento.
- Llevar a cabo el análisis forense, no es fácil y se requieren especialistas.
- Proteger el registro agrega sobrecarga y coste.

CSP

- Cifrar los datos o establecer un canal seguro a un almacén de registros.
- Proporciona confidencialidad en la comunicación con un almacenamiento seguro, si el canal de comunicación no es seguro, abre la posibilidad de que un atacante pueda comprometer el canal de comunicación y modificar los datos que están en tránsito.

6. PATRONES DE SEGURIDAD PARA REDES

6.1 Abstract Virtual Private Network

Descripción

El patrón SPP Abstract Virtual Private Network (Figura 21) describe cómo configurar un canal seguro entre dos puntos finales utilizando un túnel cifrado con autenticación en cada punto final.

Se protegen las comunicaciones mediante el establecimiento de un túnel criptográfico entre dos puntos finales en una de las capas de protocolo de comunicación, agregando funciones de autenticación en cada punto final.

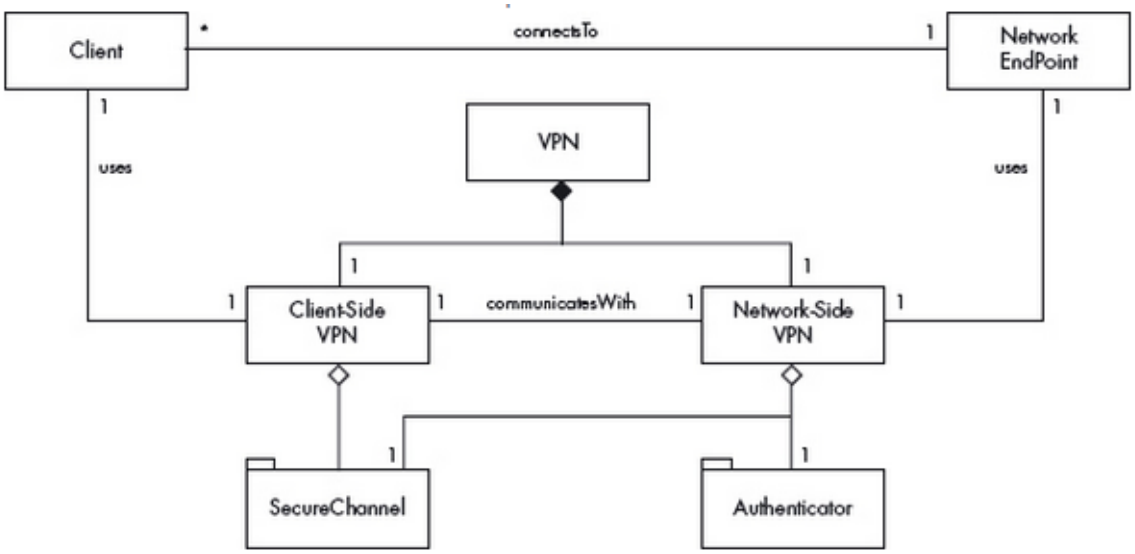


Figura 21. Diagrama de clase del patrón SPP Abstract Virtual Private Network

Relación entre catálogos

Es una red virtual privada y no tiene relación con los CSP, salvo que utiliza un SPP Authenticator, y ese es un CSP Authentication Enforcer.

Ventajas por catálogos

Una de las ventajas que ofrece el patrón SPP Abstract Virtual Private Network SPP con respecto al CSP es que explica el patrón de forma detallada a través de ejemplos, gráficos y

secuencias de clase. El patrón CSP Authentication Enforcer muestra ejemplos de código ilustrando diferentes configuraciones de autenticación.

A destacar en ambos catálogos

SPP

- Si la conexión de la VPN está comprometida, el atacante puede obtener acceso completo a la red interna, sin embargo las restricciones pueden restringir este acceso.
- En el caso de utilizar la VPN con un usuario final, el ordenador remoto utilizado por el ordenador privado es vulnerable a los ataques desde afuera.
- La criptografía puede proteger los mensajes de ser leídos o modificados por los atacantes.
- La autenticación mutua entre usuarios finales es posible.
- Se pueden agregar autorizaciones a acceder a recursos específicos de cada término.

CSP

- Sugiere mantener todos los códigos del User Login en clases separadas de la clase de aplicación. Esto ayuda a gestionar el cambio de mecanismos de autenticación a futuro.
- Elegir la mejor seguridad. La autenticación del HTTP básico es usualmente la más vulnerable a ataques.

6.2 IPSec VPN

Descripción

El patrón SPP IPSec VPN describe cómo establecer un canal seguro entre dos puntos finales utilizando un túnel cifrado a través de la capa IP, con la autenticación en cada punto final utilizando las instalaciones de IPSec.

Relación entre catálogos

La seguridad en la capa IP de un VPN, no tiene relación con el catálogo CSP, salvo la identificada en el patrón SPP Abstract VPN, al utilizar un SPP Authenticator, que es un CSP Authentication Enforcer.

Ventajas por catálogos

El catálogo SPP considera explícitamente este patrón, cosa que no hace el catálogo CSP.

A destacar en ambos catálogos

SPP

- Solo puede proteger comunicaciones basadas en IP.
- Puede proteger comunicaciones entre clientes y las puertas IPSec en arquitecturas puerta a puerta.
- La VPNs IPSec requiere un paquete de software entre 6-8MB, que puede ser difícil de configurar.
- IPSec es soportado por la mayoría de los sistemas operativos.
- La VPN es transparente a los clientes en la arquitectura de una puerta a otra.
- Se pueden utilizar una variedad de protocolos autenticados.

6.3 TLS Virtual Private Network

Descripción

El patrón SPP TLS Virtual Private Network (Figura 22) escribe cómo configurar un canal seguro entre dos puntos finales utilizando un túnel cifrado a través de la capa de transporte, con la autenticación y autorización en cada punto final.

Utiliza servidores proxy TLS inversos (comúnmente conocidos como servidores proxy SSL) para conectar a usuarios remotos. Un usuario remoto que necesita acceder a las aplicaciones de la organización utiliza la dirección URL principal para el servidor proxy en su navegador web, y se conecta a él a través de HTTP protegido con TLS. El usuario se autentica en el servidor proxy. Una vez autenticado, el usuario puede acceder a las aplicaciones designadas, según se especifica en los controles de acceso del servidor proxy.

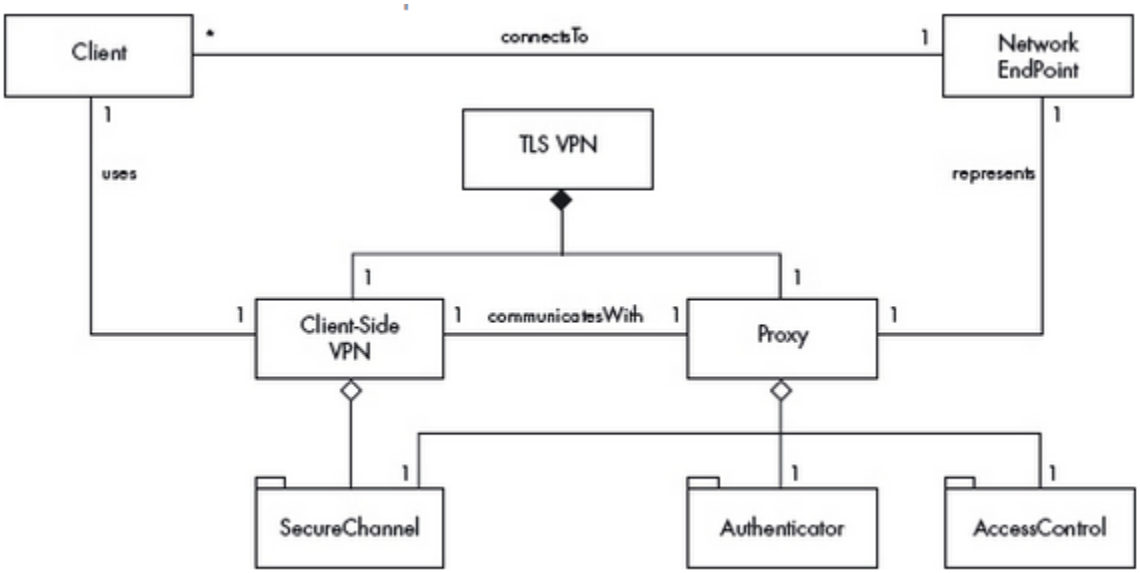


Figura 22. Diagrama de clase del patrón SPP TLS Virtual Private Network

Relación entre catálogos

La seguridad en la capa de transporte de una VPN, no tiene relación, salvo la identificada en la SPP Abstract VPN al utilizar un SPP Authenticator, que es un CSP Authentication Enforcer.

Ventajas por catálogos

El catálogo SPP considera explícitamente este patrón, cosa que no hace el catálogo CSP.

A destacar en ambos catálogos

SPP

- Comprometer al servidor proxy puede permitir al atacante interceptar los datos y autenticar las credenciales de diferentes aplicaciones.
- TLS (SSL) es un protocolo complejo que ha estado teniendo problemas de seguridad en algunas implementaciones.
- Desde que el cliente se conecta a la capa de red, no se encuentran en la misma red como un cliente IPSec, esto reduce severamente los ataques a la red.
- El servidor proxy puede autenticarse para el usuario por medio de un certificado.

6.4 Transport Layer Security

Descripción

El patrón SPP Transport Layer Security (Figura 23) describe cómo proporcionar un canal seguro entre un cliente y un servidor por el cual los mensajes de la aplicación se comunican a través de la capa de transporte de Internet. El cliente y el servidor están mutuamente autenticados y la integridad de sus datos se conserva.

Establece un canal seguro cifrado entre el cliente y el servidor mediante algoritmos que pueden ser negociados entre ellos. Proporciona los medios para que el cliente y el servidor se autentiquen mutuamente de manera de preservar la integridad de los mensajes.

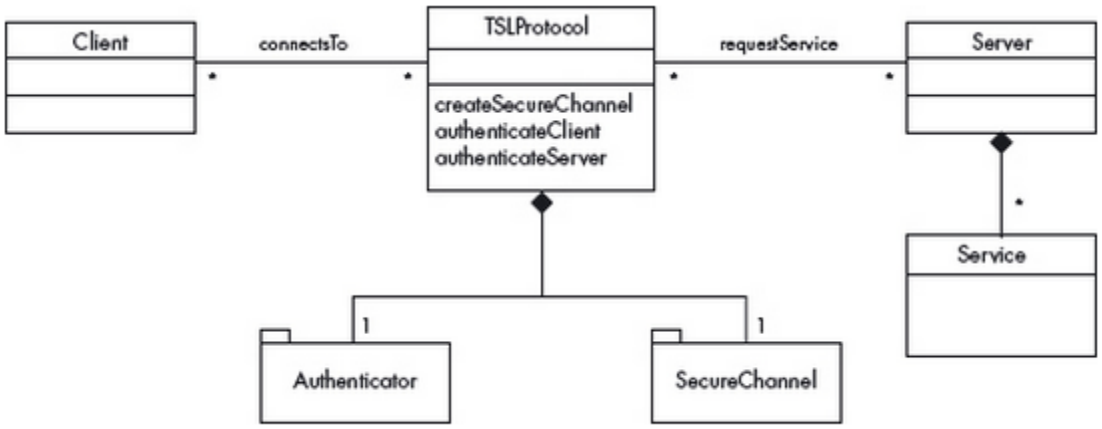


Figura 23. Diagrama de clase del patrón SPP Transport Layer Security

Relación entre catálogos

El patrón SPP Transport Layer Security es un patrón CSP Secure Pipe (Figura 24), pues establece un canal seguro entre dos puntos utilizando un túnel criptográfico en la capa de transporte con autenticación y autorización en ambos lados.

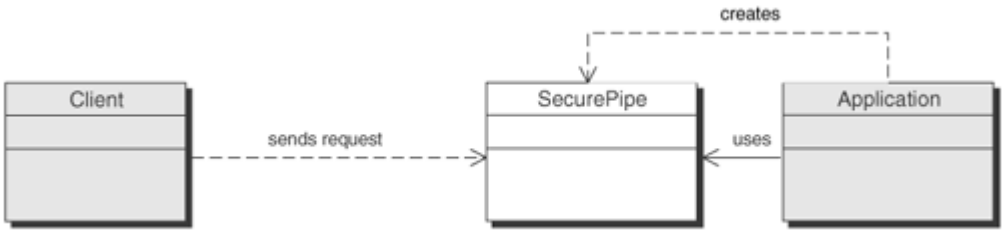


Figura 24. Diagrama de clase del patrón CSP Secure Pipe Pattern

Ventajas por catálogos

El patrón SPP Transport Layer Security menciona las fases, estructuray un caso de uso del protocolo Handshake mientras CSP indica como puede ser implementado el patrón Secure Pipe en la capa de aplicación utilizando Java Secure Socket Extensions (JSSE). El catálogo CSP incluye además ejemplos de código para crear un RMI seguro que utiliza SSL.

A destacar en ambos catálogos

SPP

- Los protocolos SSL/TLS, no están diseñados para manejar múltiples partes, sin embargo la variante MTLS puede manejar varias partes.
- Este patrón provee una forma simple y estándar para proteger los datos enviados a través de una red. No requiere la capa lógica de aplicación y reduce la complejidad de la implementación.

CSP

- Hace uso de SSL / TLS entre el cliente y el servidor Web. Sin ello, los mecanismos para garantizar la privacidad y la integridad de los datos se deben realizar en la propia aplicación, lo que aumenta la complejidad y la reducción de capacidad de administración.
- Establece la confidencialidad e integridad a través de un canal seguro entre el servidor y el cliente.
- El cliente y el servidor pueden ser mutuamente autenticados.
- Se puede cambiar fácilmente los algoritmos de encriptación y autenticidad. Los usuarios no necesitan desarrollar alguna operación para establecer un canal seguro.

6.5 Abstract IDS (Intrusion Detection System)

Descripción

El patrón SPP Abstract IDS (Figura 25) permite la monitorización de todo el tráfico que pasa en una red, para detectar posibles ataques y desencadenar una respuesta adecuada. Cada solicitud de acceso a la red se analiza para comprobar si se ajusta a la definición de un ataque. Si se detecta un ataque, una alerta se eleva y se toman las medidas necesarias.

El modelo abstracto IDS define las características básicas de cualquier sistema de detección de intrusiones (IDS).

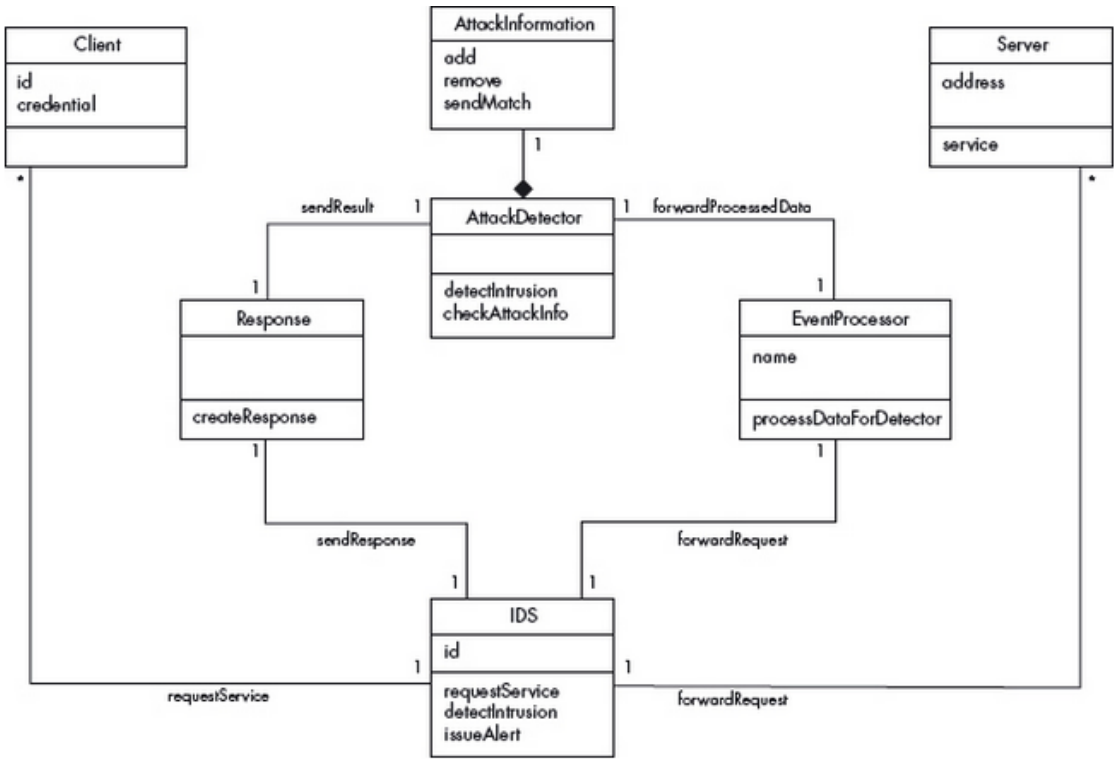


Figura 25. Diagrama de clase del patrón SPP Abstract IDS

Relación entre catálogos

Es un mecanismo adicional al firewall para monitorizar y analizar el tráfico de red para detectar intrusos, y hasta cierto punto, un CSP Message Interceptor Gateway es un SPP Abstract IDS para mantener la seguridad de los servicios web.

Ventajas por catálogos

El patrón SPP Abstract IDS está relacionado con los patrones Signature Based IDS y Behavior – Based IDS que pueden agregar funcionabilidades según el tipo de ataque mientras el CSP Message Interceptor Gateway está relacionado con el patrón CSP Message Inspector que es utilizado para verificar y validar la calidad en los mecanismos de seguridad a nivel del mensaje aplicado a los Servicios web XML.

A destacar en ambos catálogos

SPP

- Algunos ataques pueden ser muy rápidos y pueden ser difícil de reconocer en tiempo real.
- IDS provee una capa de seguridad para ampliar la autenticación y encriptación.
- La información detectada puede ser modificada para incluir nuevos ataques o nuevos comportamientos.

CSP

- Actúa como un control centralizado y un subsistema de procesamiento para forzar las tareas relacionadas con la seguridad a través de todo el servicio hasta los puntos finales.
- Restringe los accesos directos y centraliza todos los mecanismos de seguridad.
- Encapsula y protege todo los accesos directos en los puntos finales del servicio.
- Ofrece extensibilidad para incorporar más mecanismos y funcionabilidades relacionadas con el nivel de seguridad de transporte y mensaje.
- Separa el modelo de arquitectura segura con el servicio final.

6.6 Signature Based IDS

Descripción

El patrón SPP Signature Based IDS (Figura 26) describe cómo comprobar todas las solicitudes de acceso a la red con un conjunto de firmas de ataques existentes, con el fin de detectar posibles ataques y desencadenar una respuesta adecuada. Para detectar la presencia de ataques, se compara la firma actual del atacante con la firma que se tiene previamente de otros ataques conocidos.

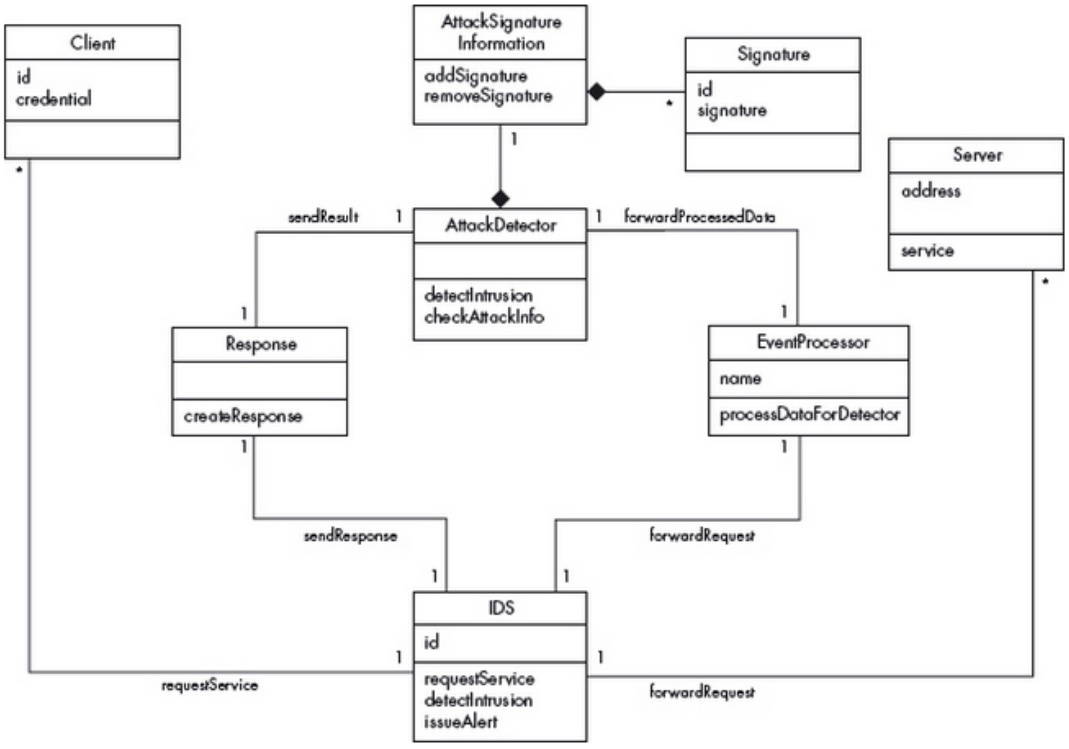


Figura 26. Diagrama de clase del patrón SPP Signature Based IDS

Relación entre catálogos

El patrón SPP Signature Based IDS analiza las firmas de las peticiones, por tanto tiene relación con un Abstract IDS, es decir, un CSP Message Interceptor Gateway podría ser un SPP Signature-Based IDS.

Ventajas por catálogos

El patrón SPP Signature Based IDS está relacionado con los patrones Abstract IDS y Behavior – Based IDS que pueden agregar funcionabilidades según el tipo de ataque mientras el CSP Message Interceptor Gateway está relacionado con el patrón CSP Message Inspector que es utilizado para verificar y validar la calidad en los mecanismos de seguridad a nivel del mensaje aplicado a los Servicios web XML.

A destacar en ambos catálogos

SPP

- Trabaja para detectar ataques conocidos.
- En algunos ataques no está bien definida la firma y el atacante puede distinguir la firma.
- Si todas las firmas conocidas de ataques están disponibles en una base de datos, los ataques pueden ser detectados en tiempo real.
- Es relativamente fácil agregar nuevos ataques para su detección.

CSP

- Actúa como un control centralizado y un subsistema de procesamiento para forzar las tareas relacionadas con la seguridad a través de todo el servicio hasta los puntos finales.
- Restringe los accesos directos y centraliza todos los mecanismos de seguridad.
- Encapsula y protege todo los accesos directos en los puntos finales del servicio.
- Ofrece extensibilidad para incorporar más mecanismos y funcionabilidades relacionadas con el nivel de seguridad de transporte y mensaje.
- Separa el modelo de arquitectura segura con el servicio de punto final.

6.7 Behavior – Based IDS

Descripción

El patrón SPP Behavior – Based IDS (Figura 27) describe cómo comprobar todas las solicitudes de acceso en contra de los patrones de tráfico de la red con el fin de detectar posibles desviaciones del comportamiento normal (anomalía) que pueden indicar un ataque y desencadenar respuestas inapropiadas.

Se observa el tráfico a través de una red y se trata de encontrar las desviaciones del comportamiento normal o esperado. Cualquier desviación de la conducta normal es tratada como un signo de intrusión.

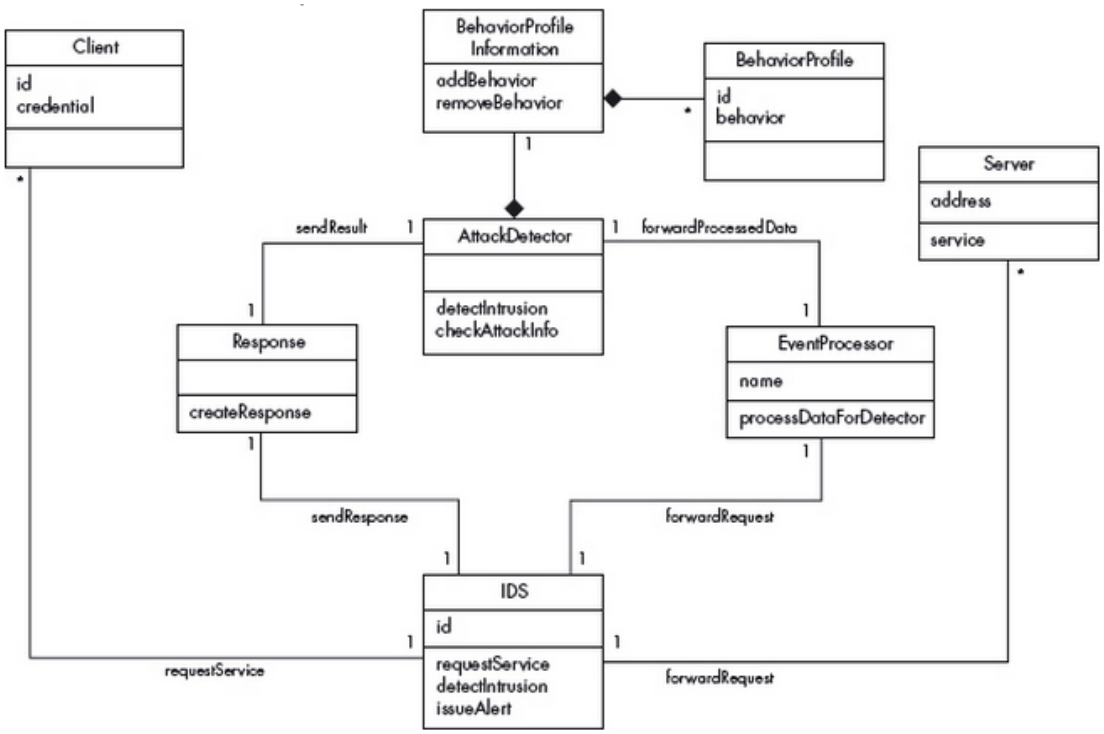


Figura 27. Diagrama de clase del patrón SPP Behavior –Based IDS

Relación entre catálogos

El patrón SPP Behavior – Based IDS es un SPP Abstract IDS basado en analizar el comportamiento de las peticiones. Por tanto tiene relación de un SPP Abstract IDS, es decir, un CSP Message Interceptor Gateway podría ser un SPP Behavior-Based IDS.

Ventajas por catálogos

SPP Behavior- Based IDS está relacionado con los patrones Abstract IDS y Signature Based IDS que pueden agregar funcionabilidades según el tipo de ataque mientras el CSP Message Interceptor Gateway está relacionado con el patrón CSP Message Inspector que es utilizado para verificar y validar la calidad en los mecanismos de seguridad a nivel del mensaje aplicado a los Servicios web XML.

A destacar en ambos catálogos

SPP

- La detección puede ser efectiva en contra de nuevos ataques que pudieran causar un comportamiento anormal en el tráfico de la red.
- El IDS es usualmente bueno en una red wireless.
- Genera muchos falsos positivos.
- No puede ser implementado en redes que no tienen un patrón de red predecible.

CSP

- Actúa como un control centralizado y un subsistema de procesamiento para forzar las tareas relacionadas con la seguridad a través de todo el servicio hasta los puntos finales.
- Restringe los accesos directos y centraliza todos los mecanismos de seguridad.
- Encapsula y protege todo los accesos directos en los puntos finales del servicio.
- Ofrece extensibilidad para incorporar más mecanismos y funcionabilidades relacionadas con el nivel de seguridad de transporte y mensaje.
- Separa el modelo de arquitectura segura con el servicio de punto final.

7. PATRONES PARA SERVICIOS WEB SEGUROS

7.1 Application Firewall

Descripción

El patrón SPP Application Firewall (Figura 28) permite filtrar las llamadas y respuestas de/para aplicaciones empresariales, basadas en políticas de control de acceso. Las aplicaciones empresariales se ejecutan en sistemas de acceso distribuido de una red local, internet u otra red externa. Estos sistemas distribuidos incluyen filtros de paquetes o proxy basados en firewalls. Las políticas de cada aplicación son centralizadas dentro de la aplicación de Firewall y la aplicación puede ser accedida por el firewall a través de una política de autorización.

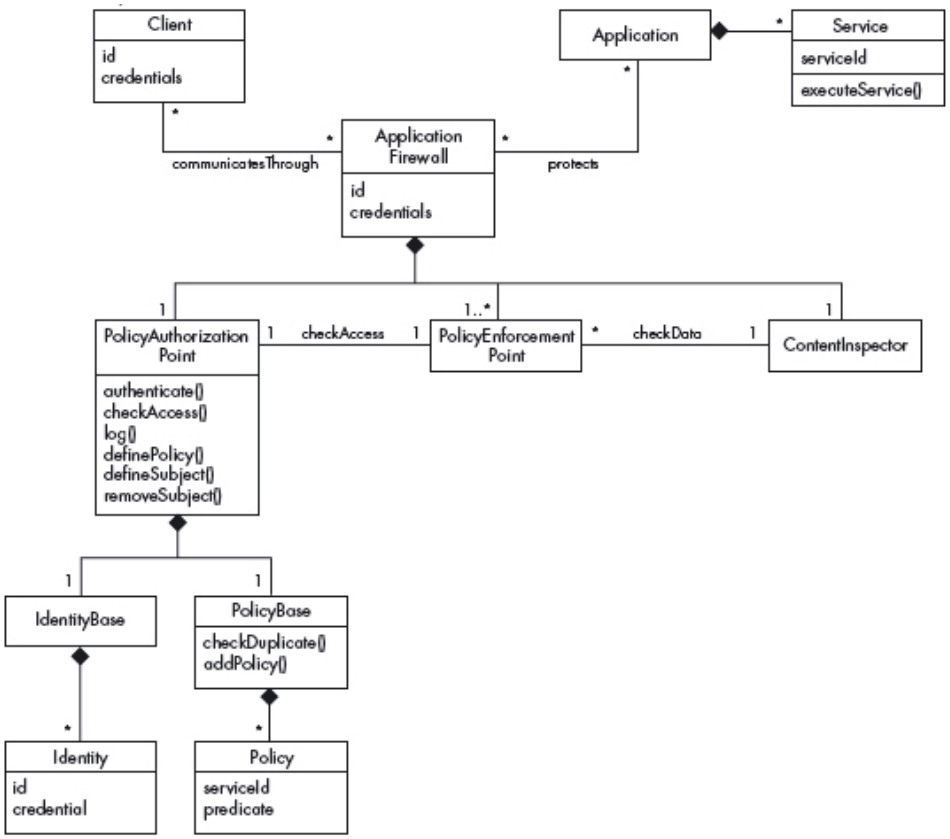


Figura 28. Diagrama de clase del patrón SPP Application Firewall

Relación entre catálogos

El patrón SPP Application Firewall permite el filtrado de llamadas para aplicaciones empresariales basándose en las políticas de control de acceso. Es por tanto una generalización del CSP Message Interceptor Gateway (Figura 29).

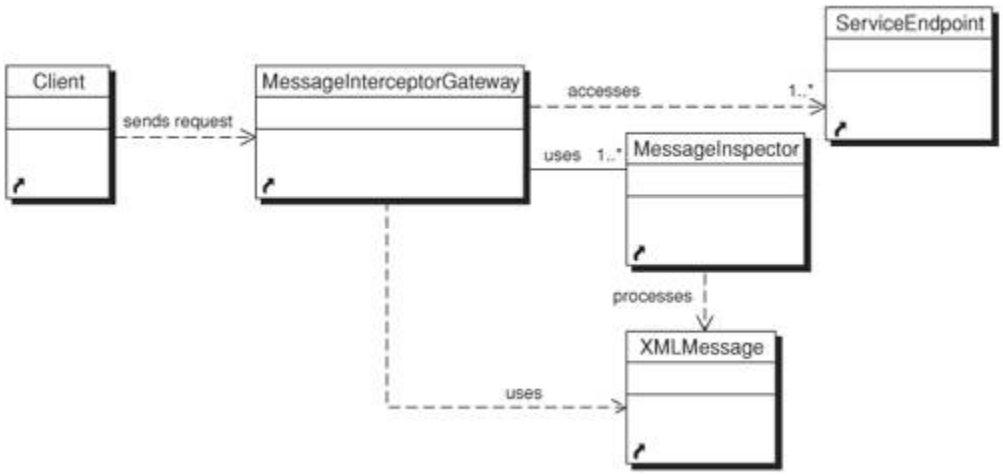


Figura 29. Diagrama de clase del patrón CSP Message Interceptor Gateway

Ventajas por catálogos

En el SPP Application Firewall se describe como agregar una nueva política y la configuración de un proxy opuesto con agentes múltiples, mientras en el patrón CSP se centra en aspectos XML.

A destacar en ambos catálogos

SPP

- Los patrones pueden ser combinados con un sistema de detección de intrusos que facilite la prevención de algunos ataques.
- Diferentes tipos de usuario o tipos de acceso solo requieren reglas específicas. El firewall de aplicaciones podría afectar el rendimiento del sistema protegido, ya que es un cuello de botella en la red, esto se puede mejorar teniendo en cuenta el servidor de seguridad como un concepto virtual y el uso de varios dispositivos físicos en la aplicación.
- La solución es redundante para aplicaciones existentes que ya implementan su propio control de acceso.

CSP

- Las aplicaciones podrían afectar el rendimiento del sistema protegido, ya que es un cuello de botella en la red.
- La solución es redundante para aplicaciones existentes que ya implementan su propio control de acceso.

7.2 XML Firewall

Descripción

El patrón SPP XML Firewall (Figura 30) permite el filtro de mensajes XML para/de aplicaciones empresariales basada en políticas de control de acceso y el contenido del mensaje. Es un tipo particular de SPP Application Firewall basado en tecnologías XML. Las políticas de cada aplicación son centralizadas dentro de la aplicación del Firewall y se puede acceder por el firewall a través de una política de autorización.

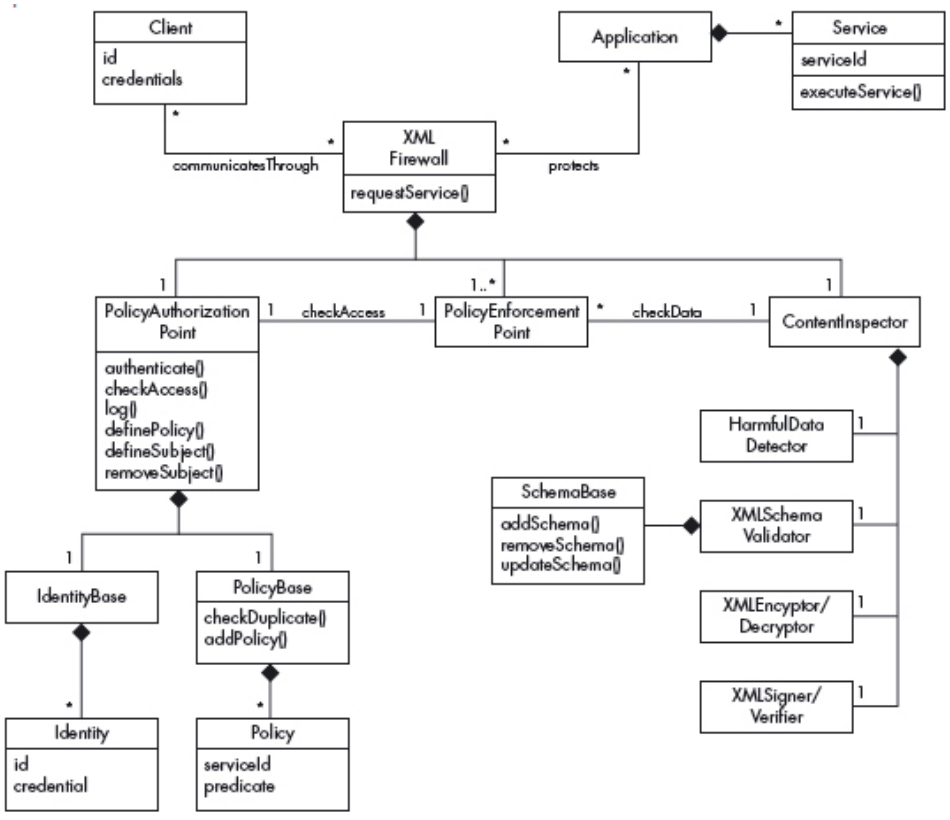


Figura 30. Diagrama de clase del patrón SPP XML Firewall

Relación entre catálogo

Analizando ambos catálogos parece que un CSP Message Interceptor Gateway (Figura 31) sería el patrón SPP XML firewall con el CSP Message Inspector integrado, y que el CSP Message Inspector estaría más cercano a un XACML Access Control Evaluation.

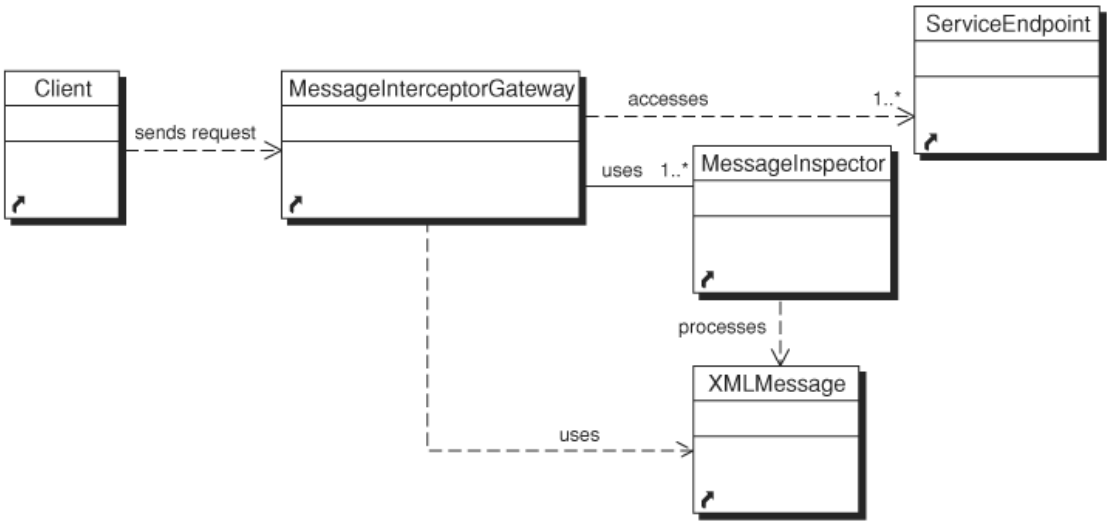


Figura 31. Diagrama de clase del patrón CSP Message Interceptor Gateway

Ventajas por catálogos

El patrón SPP XML Firewall describe los aspectos dinámicos de un XML Firewall utilizando un diagrama de secuencia y describe un filtrado en el requerimiento de un cliente encriptado y firmado con la autenticación del usuario.

El patrón CSP Message Interceptor Gateway tiene la ventaja de estar relacionado con los patrones CSP Message Inspector, Además se describe la estrategia del agente web interceptado y el XML Firewall.

A destacar en ambos catálogos

SPP

- Provee un alto nivel de seguridad para en documentos y requerimientos XML.

- Al ser detectado un fallo, debe ser reemplazado de forma transparente con una infraestructura que no ponga en peligro las entradas existentes o cualquier estado de procesamiento.
- Debe existir un mecanismo de recuperación que pueda realizar todos los procesos u operaciones.

CSP

- Las aplicaciones podrían afectar el rendimiento del sistema protegido, ya que es un cuello de botella en la red.
- La solución es redundante para aplicaciones existentes que ya implementan su propio control de acceso.

7.3 XACML Authorization

Descripción

El patrón SPP XACML Authorization (Figura 32) puede ser utilizado por una organización para representar las reglas de autorización de una manera estándar en formato XACML.

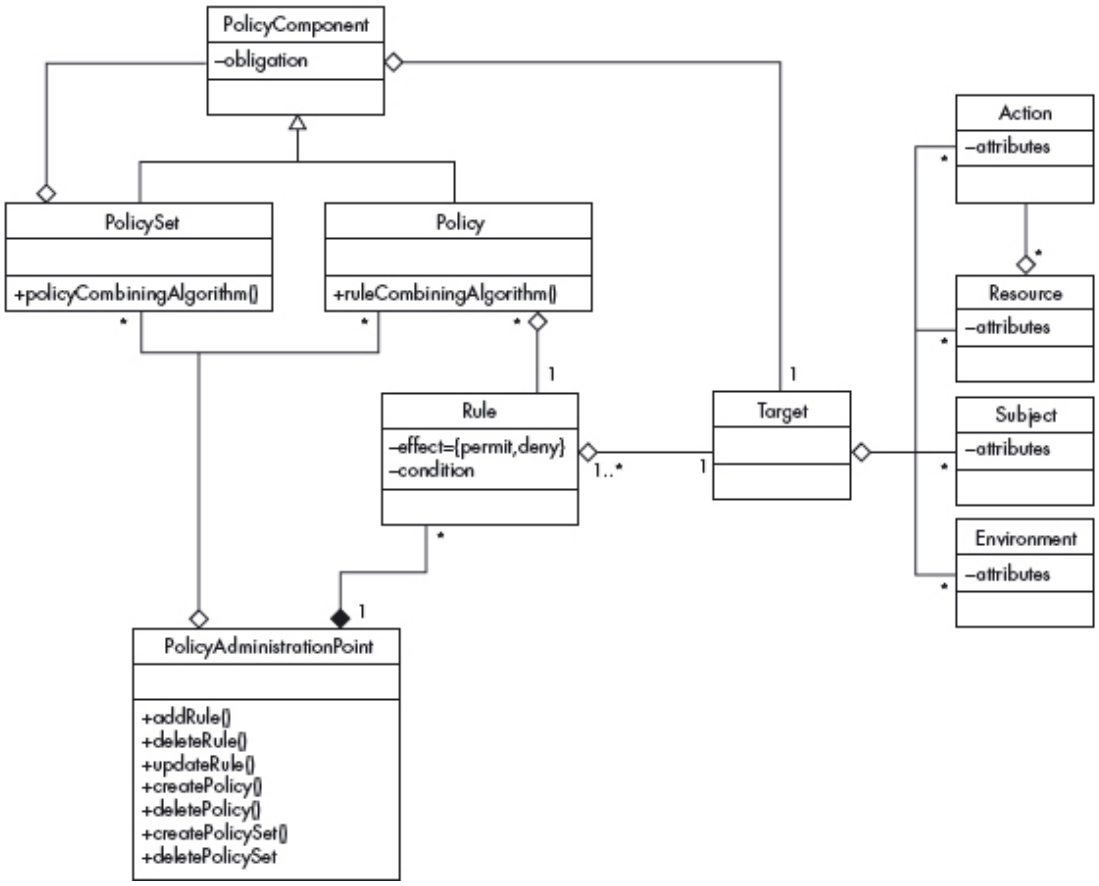


Figura 32. Diagrama de clase del patrón SPP XACML Authorization

Relación entre catálogos

El patrón SPP XACML describe una autorización XACML. Por tanto no es específica de servicios Web pero si sería utilizada por un CSP Message Inspector (Figura 33).

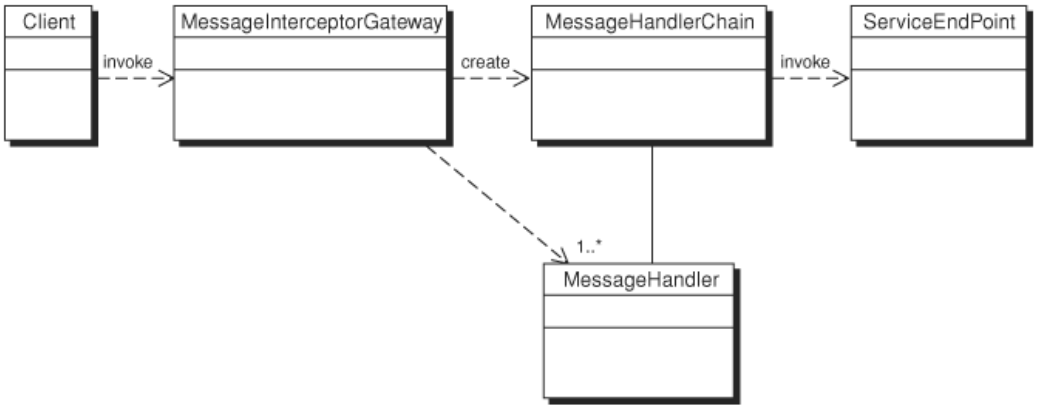


Figura 33. Diagrama de clase del patrón CSP Message Inspector

Ventajas por catálogos

El patrón SPP XACML Authorization describe cómo crear una nueva política mientras el patrón CSP Message Inspector muestra la relación con los patrones CSP Security Logger (Web Tier), CSP Audit Interceptor (Business Tier), CSP Message Interceptor Gateway (Web services tier). Además, el catálogo CSP contiene un ejemplo de la fuente de código que muestra la implementación de un Logging handler utilizando Apache Axis.

A destacar en ambos catálogos

SPP

- Una variedad de políticas puede ser descrita, como un lenguaje de políticas que incluye: los recursos, el sujeto y los atributos del ambiente.
- Una variedad de tipos de usuarios puede ser definida.
- Las políticas y las reglas pueden ser combinadas fácilmente.
- Una política puede especificar condiciones complejas. La estructura de una política es compleja, esta puede requerir el incremento de tiempo en el procesamiento para evaluar un requerimiento.

CSP

- Encapsula todos los mensajes a nivel de los mecanismos de seguridad,
- Ofrece extensibilidad permitiendo incorporar más mecanismos y funcionabilidades.

7.4 XACML Access Control Evaluation

Descripción

El patrón SPP XACML Access Control Evaluation (Figura 34) describe cómo decidir si la solicitud está autorizada a acceder a un recurso según las políticas definidas por el patrón SPP XACML Authorization.

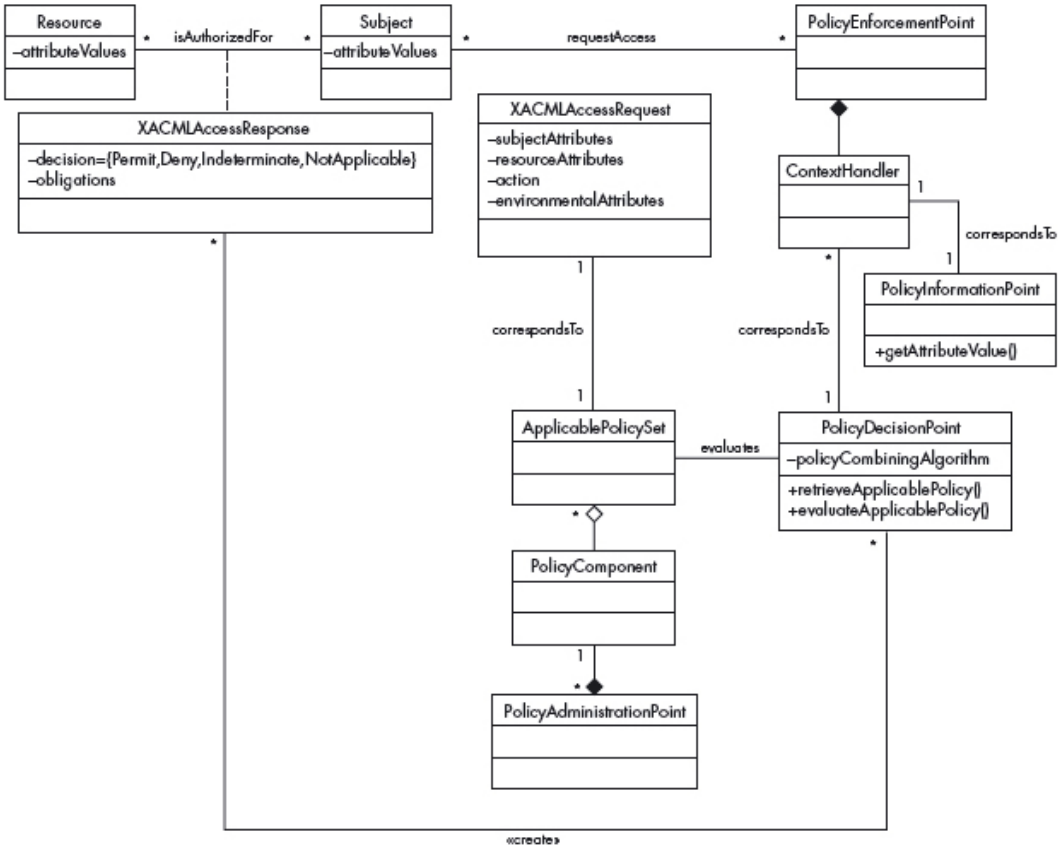


Figura 34. Diagrama de Clase del patrón SPP XACML Access Control Evaluation.

Relación entre catálogos

Este patrón es un SPP Policy-Based Access Control para servicios web basado en XML y, por tanto, sería utilizado por un CSP Message Inspector con sus patrones relacionados.

Ventajas por catálogos

El patrón SPP XACML Access Control Evaluation describe como tener el control de un requerimiento de acceso a un recurso, mientras el patrón CSP Message inspector muestra la relación con los patrones: CSP Security Logger (Web Tier), CSP Audit Interceptor (Business Tier), CSP Message Interceptor Gateway (Web services tier). Además, el catálogo CSP contiene ejemplo de fuente de código que muestra la implementación de un Logging handler utilizando Apache Axis.

A destacar en ambos catálogos

SPP

- El patrón puede soportar la matriz de acceso, RBAC o modelos multiniveles de control de acceso.
- Este puede afectar el rendimiento del sistema protegido, motivado a que XML es un lenguaje documentado de manera extendida.
- Es intrusivo para los servicios web existentes que implementan la seguridad, ya que requieren la implementación de un manejador de contexto.

CSP

- Encapsula todos los mensajes a nivel de los mecanismos de seguridad.
- Ofrece extensibilidad permitiendo incorporar más mecanismos y funcionabilidades.

7.5 Web Services Policy Language

Descripción

El patrón SPP Web Services Policy Language (WSPL) (Figura 35) describe cómo representar las políticas de control de acceso para los servicios Web de una organización de manera estándar, y permite un servicio web al cliente para expresar sus requerimientos.

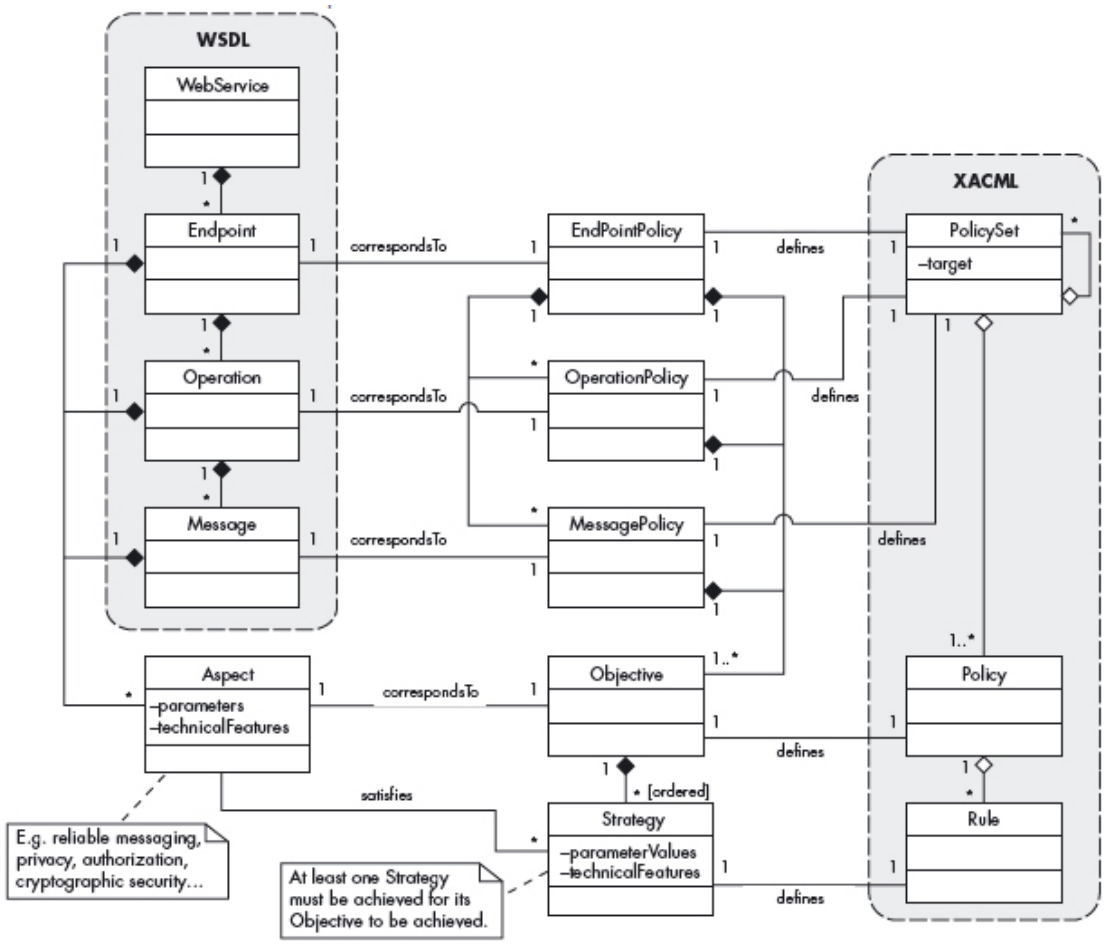


Figura 35. Diagrama de clase del patrón SPP Web Services Policy Language

Relación entre catálogos

El patrón SPP Web Services Policy Language podría ser utilizado por un CSP Secure Message Router para garantizar que las invocaciones a servicios se pueden hacer, siempre que estas políticas no fuesen XACML, en cuyo caso se utilizaría el CSP Message Inspector.

Ventajas por catálogos

El patrón CSP Secure Message Router detalla las estrategias de un Proveedor de mensajes XML y SSO Liberty Alliance. El patrón SPP Web Services Policy Language puede implementar la arquitectura definida por el patrón XML Firewall.

A destacar en ambos catálogos

SPP

- Las políticas de clientes y proveedores pueden ser combinadas para decidir cómo podría ocurrir una invocación de servicio.
- Este patrón puede afectar el rendimiento del sistema protegido.
- Es intrusivo para los servicios web existentes que ya implementan la seguridad, ya que requieren la implementación de un manejador de contexto.

CSP

- Centraliza todos los mecanismos de seguridad y configura las políticas de control de acceso.
- Encapsula todos los accesos directos para participar en los puntos finales del servicio.

7.6 WS-Policy

Descripción

El patrón SPP WS - Policy (Figura 36) describe cómo definir un conjunto de afirmaciones que pueden ser usadas y extendidas por otras especificaciones de un servicio web, con la finalidad de describir una amplia gama de requisitos y capacidades de servicio, incluida la seguridad y la fiabilidad entre otros. Este patrón también proporciona una manera de comprobar las solicitudes formuladas por los solicitantes a fin de verificar que cumplen sus afirmaciones y sus condiciones antes de interactuar con un servicio Web.

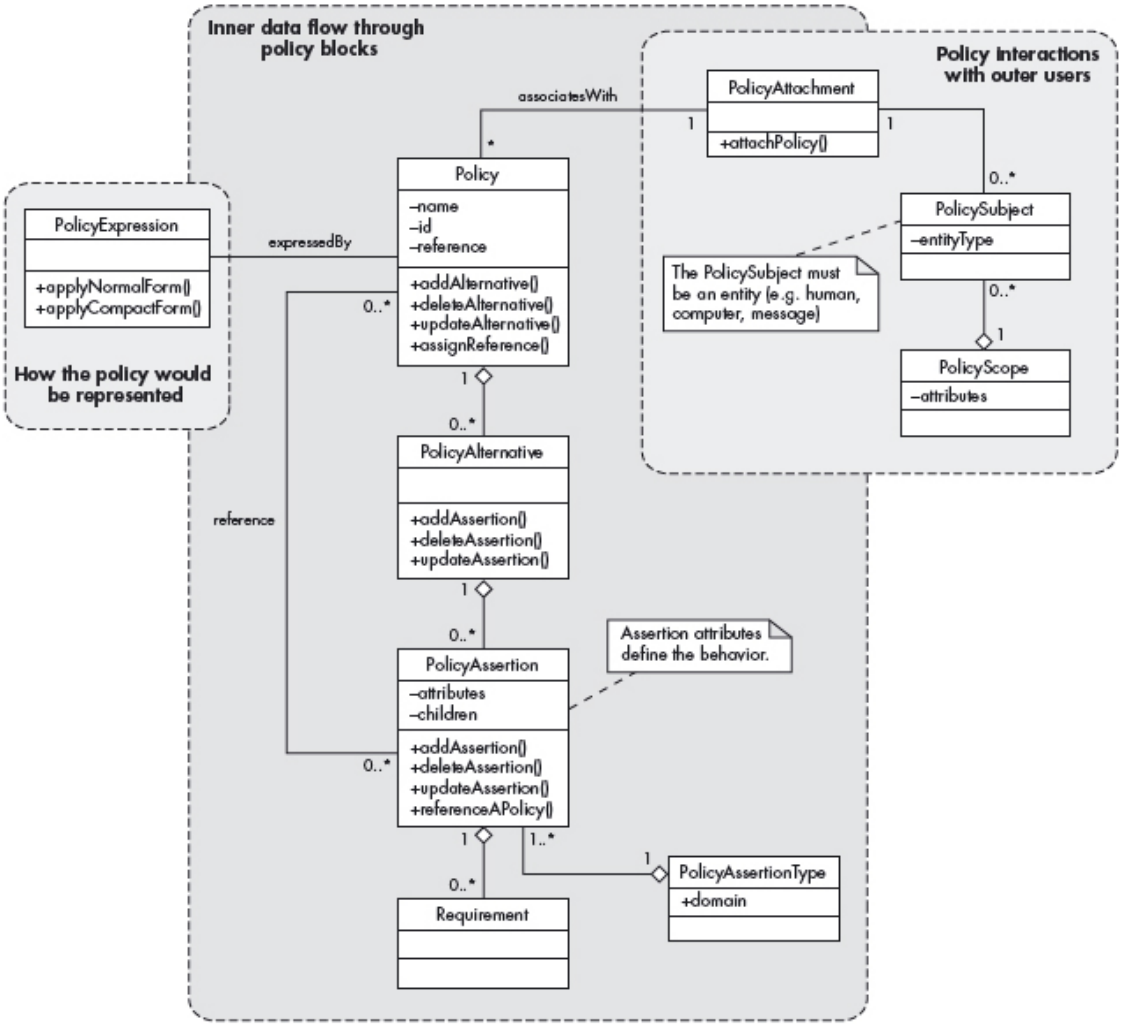


Figura 36. Diagrama de clase del patrón SPP WS-Policy

Relación entre catálogos

Es un lenguaje de políticas específico para servicios web, podría ser utilizado por un CSP Secure Message Router para garantizar que se cumplen las políticas de acceso o por un CSP Message Inspector.

Ventajas por catálogos

El patrón SPP WS- Policy describe un ejemplo basado en un sistema web Ajiad's que utiliza el patrón WS-Policy, agregando parte del código. El patrón CSP Secure Message Router detalla las estrategias de un proveedor de mensajes XML y SSO Liberty.

A destacar en ambos catálogos

SPP

- El WS-Policy es una especificación inmadura que todavía está cambiando.
- El estándar WS-Policy es un documento con muchos detalles lo cual puede hacer al patrón mucho más complejo.
- Seguridad de los datos en el servicio web.
- Es posible definir políticas para protegerse.
- Intercambio de mensajes garantizados. Este patrón ofrece una forma de asegurar el intercambio de mensajes entre los socios.

CSP

- Para mantener la seguridad de los datos del servicio web se pueden usar las políticas de otros estándares de servicios Web como WS-Security.
- Es posible definir políticas para proteger las políticas en sí mismas.
- El patrón ofrece una forma segura en el intercambio de mensajes entre socios.
- Utilizando un mecanismo de firma apropiado se puede proteger las políticas de falsificaciones.
- Centraliza todos los mecanismos de seguridad y configura las políticas de control de acceso.
- Encapsula todos los accesos directos para participar en los puntos finales del servicio.

7.7 WS-Trust

Descripción

El patrón de WS-Trust (Figura 37) describe cómo definir un servicio de token de seguridad y un motor de confianza que son utilizados con los servicios Web para autenticar otros servicios web, usando las funciones definidas en este patrón, las aplicaciones pueden entablar una comunicación segura después de establecer confianza, el objetivo de WS-Trust es permitir a las aplicaciones construir un intercambio de mensajes seguros (SOAP). La confianza está representada a través del intercambio de tokens de seguridad. Estas aplicaciones proveen un protocolo para renovar y validar estos tokens de seguridad

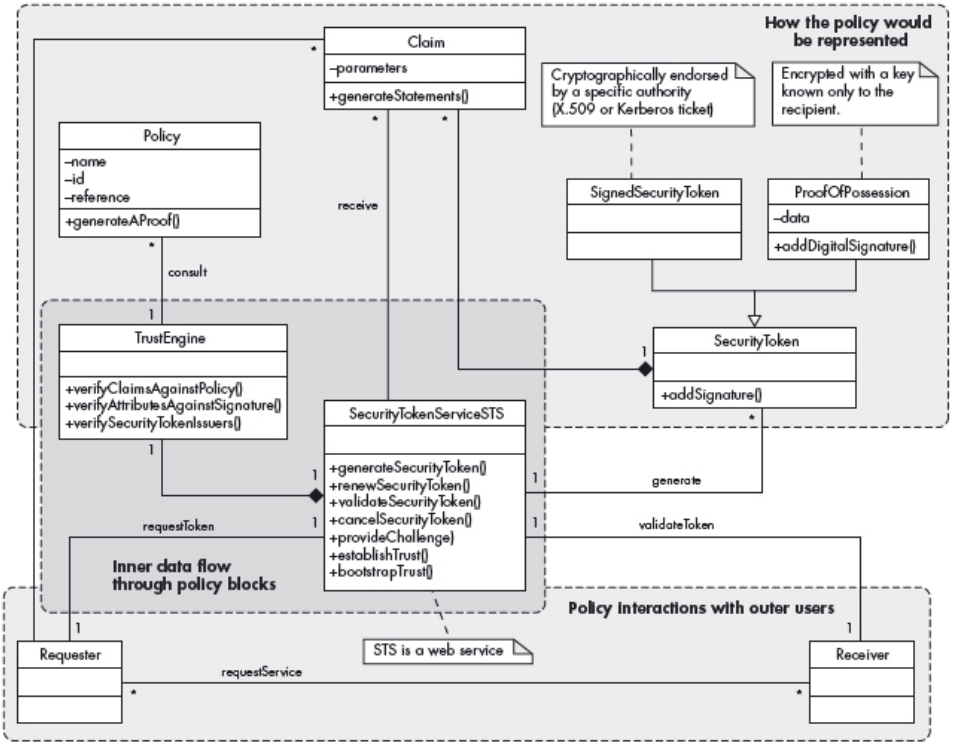


Figura 37. Diagrama de clase del patrón SPP WS-Trust

Relación entre catálogos

El objetivo del patrón SPP WS Trust es permitirle a las aplicaciones construir el intercambio de mensajes SOAP confiables. Esta confianza es representada a través de intercambios de tokens de seguridad. Esta especificación provee un protocolo publicado para renovar y validar estos tokens de seguridad. (OASIS WS-Trust 1.3). Por tanto, sería utilizado por un CSP Message Interceptor Gateway.

Ventajas por catálogos

El patrón SPP WS- Trust describe como crear un token de seguridad y como acceder a un recurso utilizando un token y el CSP Message Interceptor Gateway tiene la ventaja de estar relacionado con el patrón CSP Message Inspector. El catálogo CSP describe además la estrategia del Agente web interceptado y el XML Firewall.

A destacar en ambos catálogos

SPP

- Se extienden los mecanismos de seguridad WS – Security. Por tanto, se pueden manejar problemas como tokens de seguridad.
- Se tiene la opción de implementar la infraestructura del WS- Policy para soportar patrones de confianza.
- Todos los tokens de seguridad son intercambiados, firmados y estampados entre las partes relacionadas con claves únicas que son conocidas sólo por los receptores.
- La eficiencia del WS-Trust puede desmejorar por la ida y vuelta de múltiples tokens requeridos. Por tanto, se debe buscar una solución para disminuir el número de mensajes intercambiados.
- El estándar WS-Trust es un documento con muchos detalles lo cual puede hacer el patrón mucho más complejo.

CSP

- La eficiencia del WS-Trust puede sufrir por las repetitivas idas y vueltas de múltiples requerimientos de token.
- El estándar WS-Trust es un documento extenso y con varios detalles lo cual hace que el patrón sea complejo.

7.8 SAML Assertion

Descripción

Define una unidad de gestión de identidades en el dominio de los sujetos, proporcionando una manera de comunicación segura sobre un sujeto en particular y entre los diferentes dominios de seguridad, basándose en una colección de declaraciones relacionadas con la seguridad del sujeto definido con un formato XML común, para que pueda extenderse fácilmente (Figura 38).

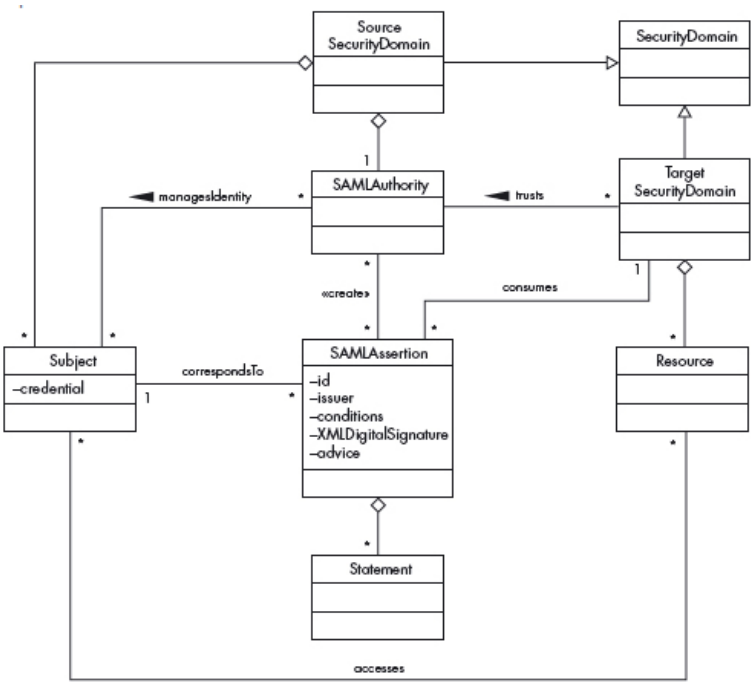


Figura 38. Diagrama de clase del patrón SPP SAML Assertion

Relación entre catálogos

La relación entre ambos catálogos consiste en que el patrón SPP SAML Assertion sería utilizado por los patrones CSP Message Inspector, CSP Assertion Builder, CSP Single Sign-On Delegator y CSP Credential Tokenizer, que incluyen la tecnología SAML y el Liberty Alliance. Además, el catálogo CSP, a pesar de no considerarlo patrón, sí que describe el lenguaje SAML con bastante nivel de detalle.

Ventajas por catálogos

El patrón SPP SAML Assertion describe las tres variantes dependiendo del tipo de afirmación que pueda ser: afirmación SAML basada en atributos, en autenticación y en autorización.

Por otro lado, el catálogo CSP describe con bastante nivel de detalle y lenguaje y relaciona a sus posibles usuarios: CSP Message inspector, CSP assertion builder, CSP SSO Delegator, CSP Credential Tokenizer.

A destacar en ambos catálogos

SPP

- Las aplicaciones con un propósito o dominio pueden ser implementadas a varios niveles de controles de seguridad.
- Los límites de la seguridad o métodos de autenticación son expresados utilizando un lenguaje común.
- La identidad puede hacer mal uso de la autoridad SAML.
- Debe existir una relación de confianza entre los dominios de autoridad SAML y el objetivo de seguridad. Esta relación permitirá verificar el origen y la integridad de la declaración.

CSP

- Debe existir una relación de confianza entre la SAML authority y el dominio objetivo de seguridad.
- Puede existir un posible abuso de identidad por el SAML authority.
- Los arquitectos y desarrolladores necesitan estar seguros que la información que está en cache está protegida y solo es accesible por el Single Sign On Delegator (SSO).
- El username password token es muy vulnerable a ataques utilizando un diccionario de claves.
- El token certificado X.509v3 es una medida de alta seguridad en relación al username password token. Sin embargo puede tener problemas en la distribución de certificados debido a un error humano pueden ser revocados.

8. PATRONES PARA CRIPTOGRAFÍA EN SERVICIOS WEB

8.1 Symmetric Encryption

Descripción

El patrón SPP Symmetric Encryption (Figura 39) protege la confidencialidad del mensaje haciendo que el mensaje sea ilegible para aquellos que no tienen la clave de acceso. El cifrado simétrico utiliza la misma clave para el cifrado y el descifrado no es específico para servicios Web

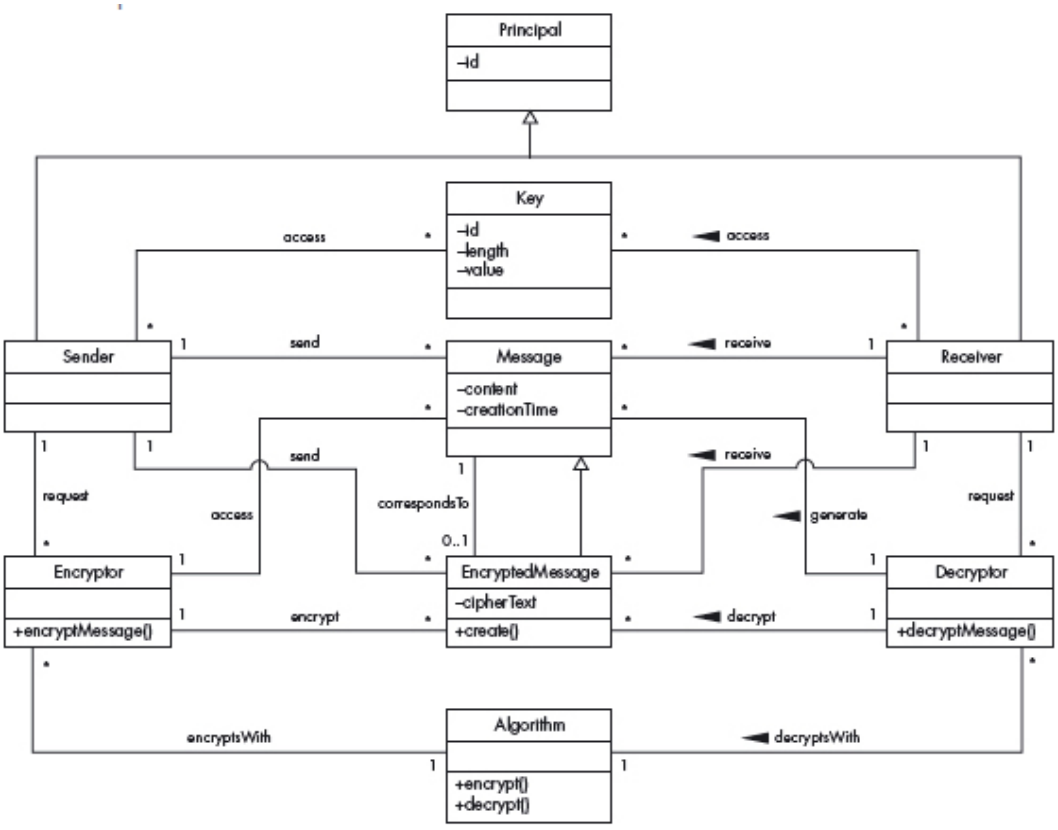


Figura 39. Diagrama de clase del patrón SPP Symmetric Encryption

Relación entre catálogos

A pesar de no ser específico para servicios webs, el patrón SPP Symmetric Encryptation puede ser utilizado en el CSP Secure Message Router.

Ventajas por catálogos

El patrón SPP Symmetric Encryption describe como encriptar un mensaje y agrega varios ejemplos gráficos. El patrón CSP Secure Message Router detalla las estrategias de un proveedor de mensajes XML y SSO liberty.

Además, a pesar de no considerarlo un patrón en sí mismo, el catálogo CSP también describe los procesos de encriptación por clave simétrica.

A destacar en ambos catálogos

SPP

- Los algoritmos de encriptación toman un tiempo aceptable para encriptar mensajes existentes.
- Es posible seleccionar de varios algoritmos de encriptación, el apropiado según la necesidad de la aplicación.
- Las operaciones de criptografía son complejas y pueden afectar el rendimiento de la aplicación, especialmente en dispositivos móviles.
- La encriptación no provee la integridad de los datos. Los datos encriptados pueden ser modificados por un atacante, es necesario por tanto verificar que el mensaje no haya sido cambiado.
- La encriptación no previene un ataque de respuesta, porque un mensaje encriptado puede ser capturado y reenviado sin ser des encriptado. Por tanto es mejor utilizar otra medida de seguridad como Time Stamps o Nonces para prevenir este tipo de ataque.

CSP

- Para mantener la seguridad de los datos del servicio web se pueden usar las políticas de otros estándares de servicios Web como WS-Security.
- Es posible definir políticas para proteger las políticas en sí mismas.
- El patrón ofrece una forma segura en el intercambio de mensajes entre socios.
- Utilizando un mecanismo de firma apropiado se puede proteger las políticas de falsificaciones.
- Centraliza todos los mecanismos de seguridad y configura las políticas de control de acceso.

- Encapsula todos los accesos directos para participar en los puntos finales del servicio.

8.2 Asymmetric Encryption

Descripción

El patrón SPP Asymmetric Encryptation (Figura 40) proporciona la confidencialidad del mensaje, debido a que mantiene en secreto la información, de manera tal que sólo pueda ser entendida por los destinatarios que tienen una clave de acceso válida. En el cifrado asimétrico se utiliza un par de claves pública / privada para el cifrado y el descifrado, respectivamente.

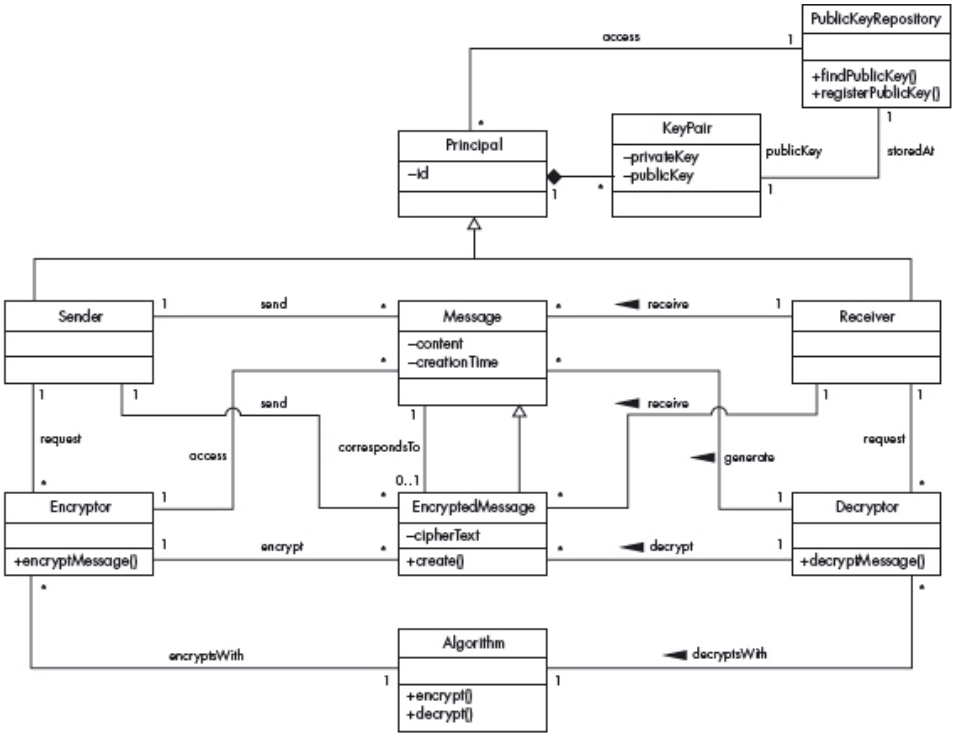


Figura 40. Diagrama de clase del patrón SPP Asymmetric Encryption

Relación entre catálogos

A pesar de no ser específico para servicios web, el patrón SPP Asymmetric Encryptation puede ser utilizado en el CSP Secure Message Router.

Ventajas por catálogos

El patrón SPP Asymmetric Encryption describe como encriptar un mensaje, el patrón CSP Secure Message Router detalla las estrategias de un proveedor de mensajes XML y SSO Liberty Alliance.

Además, a pesar de no considerarlo un patrón en sí mismo, el catálogo CSP también describe los procesos de encriptación por clave asimétrica.

A destacar en ambos catálogos

SPP

- Sólo los receptores que poseen una clave privada pueden hacer que el mensaje encriptado pueda ser leído otra vez.
- La encriptación asimétrica no requiere una clave secreta para ser compartida entre todos los participantes. Cualquiera puede buscar una clave pública en el repositorio y enviar el mensaje al dueño de la clave pública.
- Las operaciones de criptografía son complejas y pueden afectar el rendimiento de la aplicación. La encriptación asimétrica es más lenta que la encriptación simétrica. Lo mejor es utilizar la combinación de ambos algoritmos: encriptación asimétrica para la distribución de claves y encriptación simétrica para intercambio de mensajes.
- La encriptación no provee la integridad de los datos. Los datos encriptados pueden ser modificados por un atacante. Por tanto, se necesita verificar el mensaje que no haya sido modificado.
- La encriptación no previene un ataque de respuesta, porque un mensaje encriptado puede ser capturado y reenviado sin ser des encriptado. Por tanto, es mejor utilizar otra medida de seguridad como Time Stamps o Nonces para prevenir este tipo de ataque.
- Este patrón asume que la clave pública pertenece a la persona autorizada. Sin embargo como se puede verificar si la persona que tiene la clave es quien dice ser, se puede usar certificados. Si el certificado no es seguro podemos tener perdida de seguridad.

CSP

- Para mantener la seguridad de los datos del servicio web se pueden usar las políticas de otros estándares de servicios Web como WS-Security.

- Es posible definir políticas para proteger las políticas en sí mismas.
- El patrón ofrece una forma segura en el intercambio de mensajes entre socios.
- Utilizando un mecanismo de firma apropiado se puede proteger las políticas de falsificaciones.
- Centraliza todos los mecanismos de seguridad y configura las políticas de control de acceso.
- Encapsula todos los accesos directos para participar en los puntos finales del servicio.

8.3 Digital Signature with Hashing

Descripción

El patrón SPP Digital Signature with Hashing (Figura 41) permite a un emisor demostrar que un mensaje se originó a partir de él. Además, proporciona la integridad del mensaje, indicando si un mensaje ha sido alterado durante la transmisión.

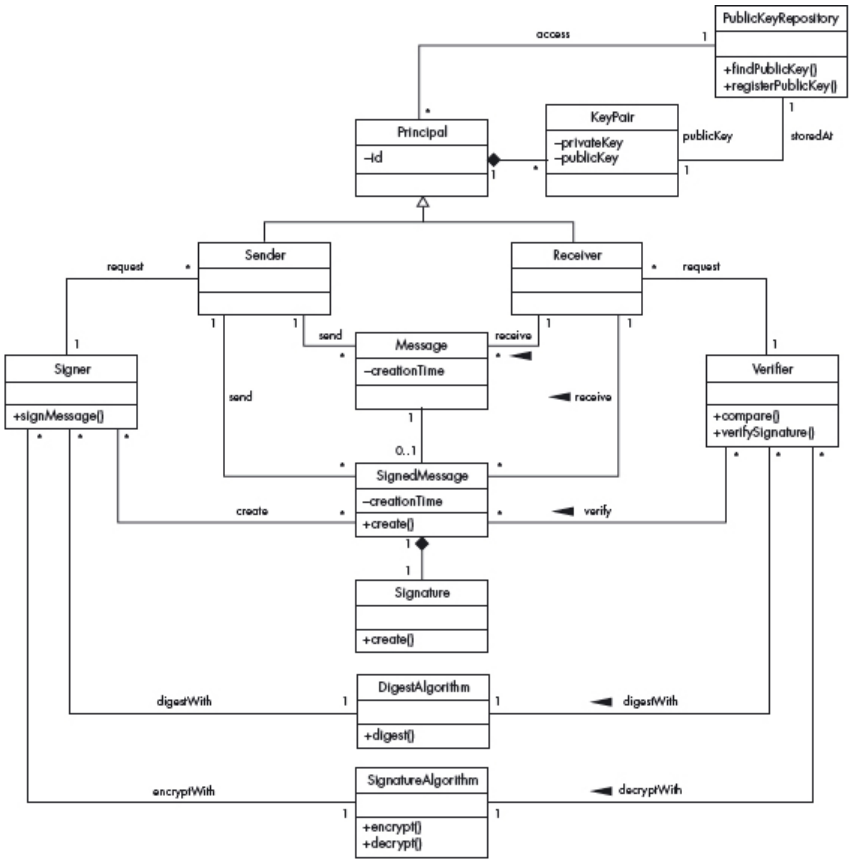


Figura 41. Diagrama de clase del patrón SPP digital Signature with Hashing

Relación entre catálogos

El patrón SPP Digital Signature with Hashing puede ser utilizado por el patrón CSP Secure Message Router siempre y cuando no sea especifico SOAP.

Ventajas por catálogos

El patrón SPP Digital Signature with hashing describe como firmar un mensaje y verificar una firma. El patrón CSP Secure Message Router detalla las estrategias de un proveedor de mensajes XML y SSO liberty.

A destacar en ambos catálogos

SPP

- Los algoritmos disponibles que pueden ser usados por firmas digitales no requieren un máximo rendimiento ni toman mucho tiempo.

- El mensaje es comprimido dentro de una cadena usando un algoritmo de hash para firmar. Como resultado, el proceso de firmar es rápido y el mensaje es mucho más corto.
- Se necesita establecer una infraestructura de claves públicas confiables. Una forma de obtenerla es a través de los certificados.
- Ambos entes que envían y reciben deben tener un acuerdo previo de firma y algoritmo de hashing que lo soporte (no necesariamente es un documento XML). Los usuarios deben implementar el protocolo de firma apropiadamente. El almacenaje requerido y la energía pueden no estar disponibles, por ejemplo en los dispositivos móviles.

CSP

- Para mantener la seguridad de los datos del servicio web se pueden usar las políticas de otros estándares de servicios web como WS-Security.
- Es posible definir políticas para proteger las políticas en sí mismas.
- El patrón ofrece una forma segura en el intercambio de mensajes entre socios.
- Utilizando un mecanismo de firma apropiado se puede proteger las políticas de falsificaciones.
- Centraliza todos los mecanismos de seguridad y configura las políticas de control de acceso.
- Encapsula todos los accesos directos para participar en los puntos finales del servicio.

8.4 XML Encryption

Descripción

El patrón XML Encryption (Figura 42) proporciona confidencialidad al ocultar la información sensible seleccionada en un mensaje utilizando criptografía.

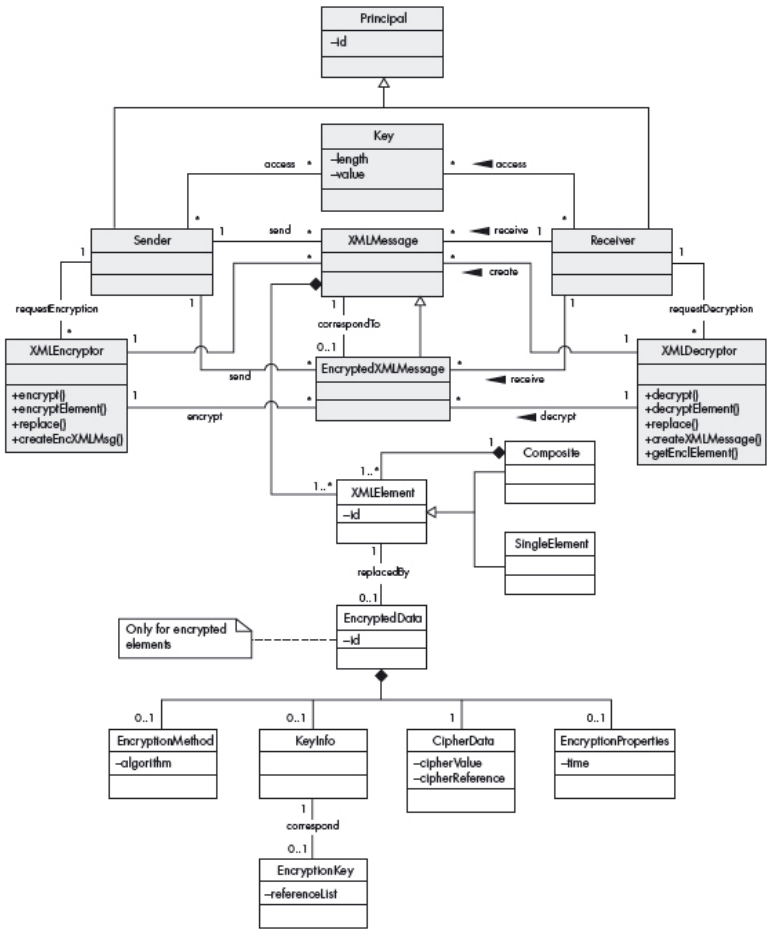


Figura 42. Diagrama de clase del patrón SPP XML Encryption

Relación entre catálogos

Como ya hemos comentado, este patrón lo utilizaría un SPP XML Firewall para encriptar/desencriptar parte de los mensajes SOAP y, por tanto, lo utilizaría un CSP Message Interceptor Gateway que utiliza además un CSP Message Inspector.

Ventajas por catálogos

El SSP XML Encryption describe como encriptar elementos XML y tiene un ejemplo ilustrado de como una parte encriptada es incrustada dentro de un mensaje XML. El patrón CSP Message Interceptor Gateway tiene la ventaja de estar relacionado con el patrón CSP Message Inspector. Además, el catálogo CSP describe la estrategia del agente web interceptado y el XML firewall.

A destacar en ambos catálogos

SPP

- Los datos encriptados son un elemento XML que reemplaza los datos a ser encriptados.
- Todo el mensaje XML o solo una parte de este puede ser encriptado.
- Si el que envía y recibe no han intercambiado previamente la clave, la clave puede ser enviada dentro del mensaje encriptado usando un sistema de clave pública.
- Las características generales de la encriptación simétrica y asimétrica aplican en este apartado.
- La estructura es compleja y los usuarios pueden confundirse.

CSP

- Las aplicaciones podrían afectar el rendimiento del sistema protegido, ya que es un cuello de botella en la red.
- La solución es redundante para aplicaciones existentes que ya implementan su propio control de acceso.

8.5 XML Signature

Descripción

El patrón XML Signature (Figura 43) permite a un ente demostrar que un mensaje se originó a partir de este. Además proporciona la integridad del mensaje detectando si un mensaje ha sido alterado durante la transmisión. La firma XML también proporciona la integridad del mensaje y requiere de canonización antes del hash y la firma.

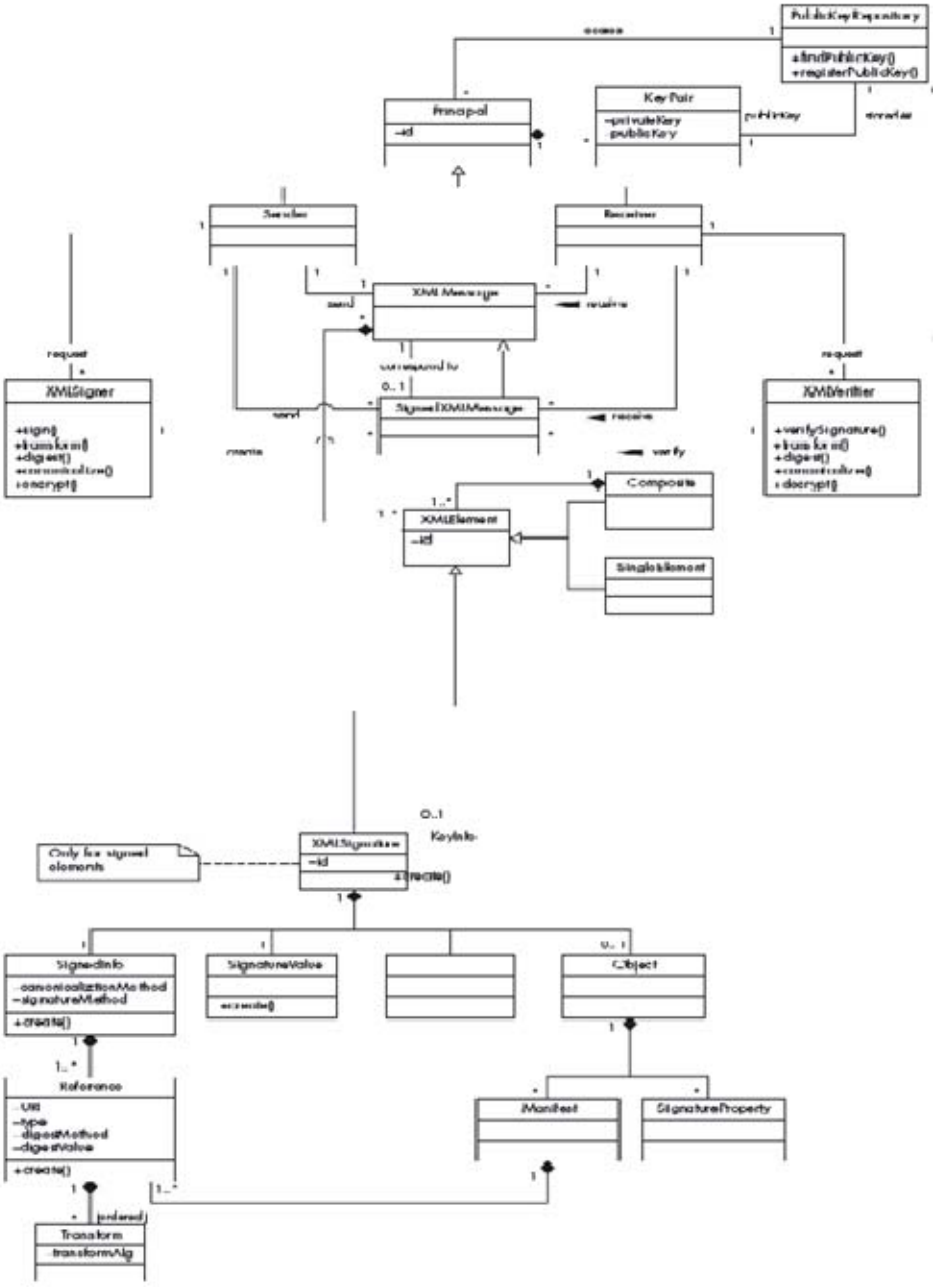


Figura 43. Diagrama de clase del patrón SPP XML Signature

Relación entre catálogos

No se corresponde directamente con ningún patrón CSP, aunque sí viene descrito como tecnología de seguridad en el catálogo CSP. Por otro lado este patrón sería utilizado por un CSP Message Interceptor Gateway que utiliza además un CSP Message Inspector.

Ventajas por catálogos

El SSP XML signature describe como firmar con diferentes elementos de un mensaje XML y como verificar una firma XML con referencias múltiples. A pesar de no considerarlo un patrón en sí, el catálogo CSP sí que describe con mayor nivel de detalle el proceso de firma XML.

A destacar en ambos catálogos

SPP

- Una clave privada es usada para firmar un mensaje, la firma valida usando una clave pública, el cual prueba que se ha creado y enviado el mensaje.
- Algún cambio en los datos que fueron indirectamente firmados producirá una invalidación del mensaje.
- Los documentos XML son descritos a sí mismo, el que envía y recibe no necesita estar de acuerdo con el algoritmo que será usado.
- Se necesita establecer una infraestructura de claves públicas confiables, una forma de obtenerla es a través de los certificados.
- Los usuarios deben implementar el protocolo de firma apropiadamente.
- El patrón no describe el estándar completo.

8.6 WS-Security

Descripción

El estándar WS-Security (Figura 44) describe cómo incorporar mecanismos de seguridad existentes, tales como: encriptación XML, firma digital y tokens de seguridad en mensajes SOAP para proporcionar confidencialidad, integridad y autenticación.

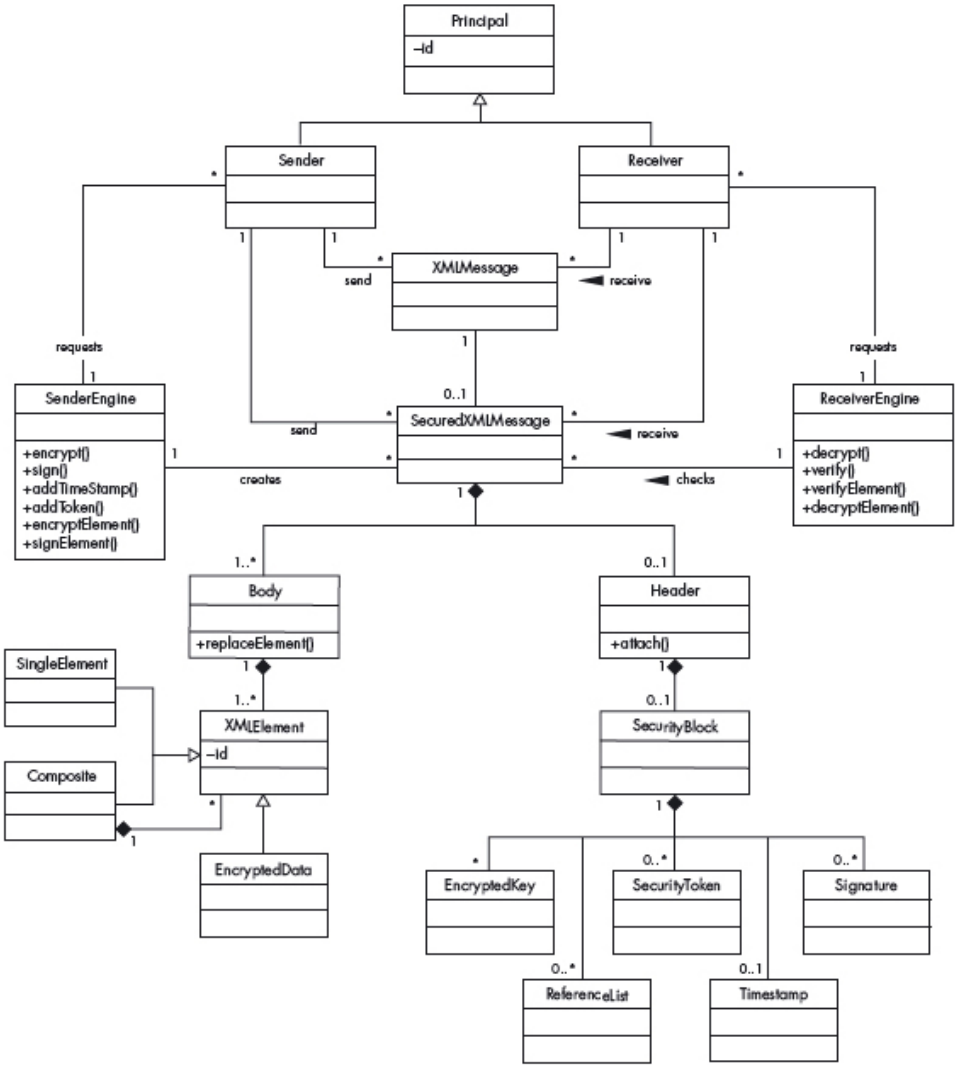


Figura 44. Diagrama de clase del patrón SPP WS-Security

Relación entre catálogos

Aunque no es considerado como un patrón en el catálogo CSP sí que viene descrito como una tecnología de seguridad. En lo referente a patrones, sería utilizado por CSP Message Interceptor Gateway y un CSP Message Inspector.

Ventajas por catálogos

El SSP WS- Security describe como encriptar un elemento utilizando un clave encriptada y como firmar un elemento utilizando tokens de seguridad. A pesar de no considerarlo un patrón en sí, el catálogo CSP sí que describe con mayor nivel de detalle el proceso de firma XML.

A destacar en ambos catálogos

SPP

- Usando la cabecera de un mensaje SOAP, se puede especificar las características de seguridad de un mensaje como una encriptación XML, firmas XML y tokens de seguridad.
- Se puede especificar diferentes partes del mensaje con diferentes tipos de encriptación, diferentes claves y firmas.
- El patrón no describe detalles de encriptación, firmas digitales o tokens de seguridad. Estos requieren estándares separados.
- El estándar no indica si se debe firmar o encriptar todo o una parte del mensaje o solo la cabecera. El diseñador definirá estos aspectos dejando muy abierto este tema.
- WS-Security es una especificación inmadura que todavía está cambiando.

9. PATRONES PARA SEGURIDAD MIDDLEWARE

9.1 Secure Broker

Descripción

El patrón SPP Secure Broker (Figura 45) desacopla las comunicaciones de las aplicaciones. Un Secure Broker debe introducir la autenticación mutua entre servidores y clientes, proporcionando una autorización y un monitor de referencia para controlar el acceso a los recursos y controlando la criptografía para prevenir los ataques de mensajes.

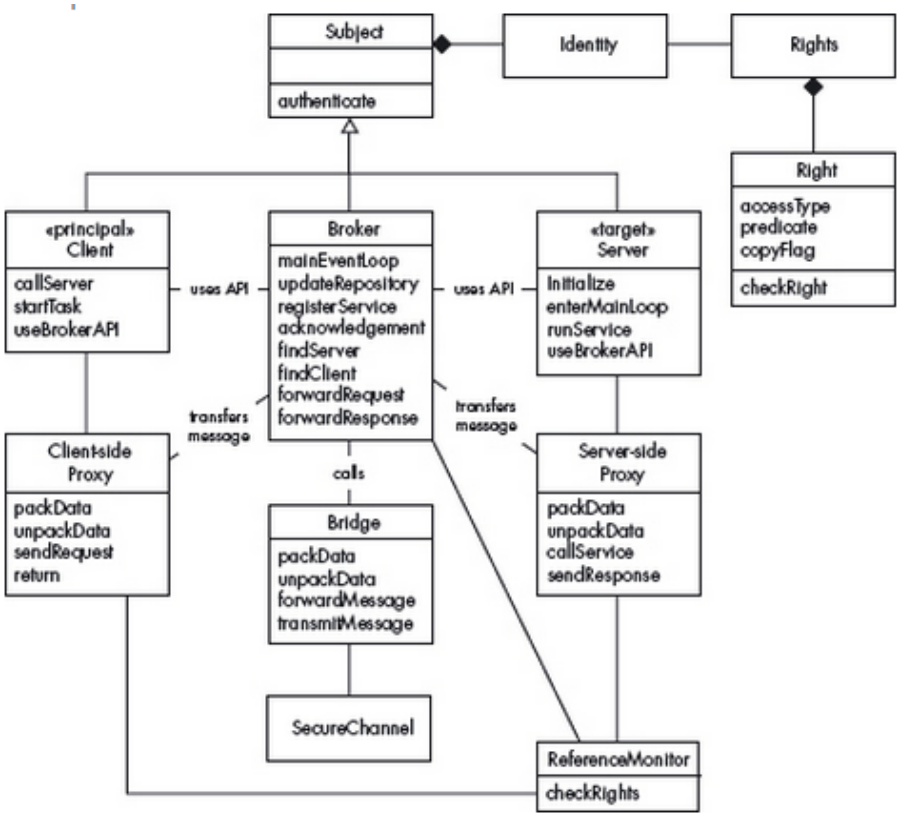


Figura 45. Diagrama de clase del patrón SPP Secure Broker

Relación entre catálogos

El patrón CSP Secure Message Router es un tipo especializado de SPP Secure Broker para la invocación de servicios distribuidos.

Ventajas por catálogos

El patrón SPP Secure Broker descríbe la implementación de CORBA y .NET Remoting. El patrón CSP Secure Message Router detalla las estrategias de un proveedor de mensajes XML y SSO Liberty Alliance.

A destacar en ambos catálogos

SPP

- El control de acceso puede ser implementado permitiendo las restricciones en el uso de la información privilegiada y funcional. La encriptación puede manejar todos estos problemas.
- La autenticación y el control de acceso previenen quitar las entradas válidas.
- Existe complejidad agregada.

CSP

- Para mantener la seguridad de los datos del servicio web se pueden utilizar las políticas de otros estándares de servicios Web como WS-Security.
- Es posible definir políticas para proteger las políticas en sí mismas.
- El patrón ofrece una forma segura en el intercambio de mensajes entre socios.
- Utilizando un mecanismo de firma apropiado se puede proteger las políticas de falsificaciones.
- Centraliza todos los mecanismos de seguridad y configura las políticas de control de acceso.
- Encapsula todos los accesos directos para participar en los puntos finales del servicio.

9.2 Secure Pipes and Filters

Descripción

El patrón SPP Secure Pipes and Filters (Figura 46) proporciona una manera segura para procesar datos en diferentes etapas o pasos, añadiendo mecanismos de seguridad básicos (como los casos de patrones de seguridad). Para cada paso proporciona autenticación, autorización y registro y oculta datos.

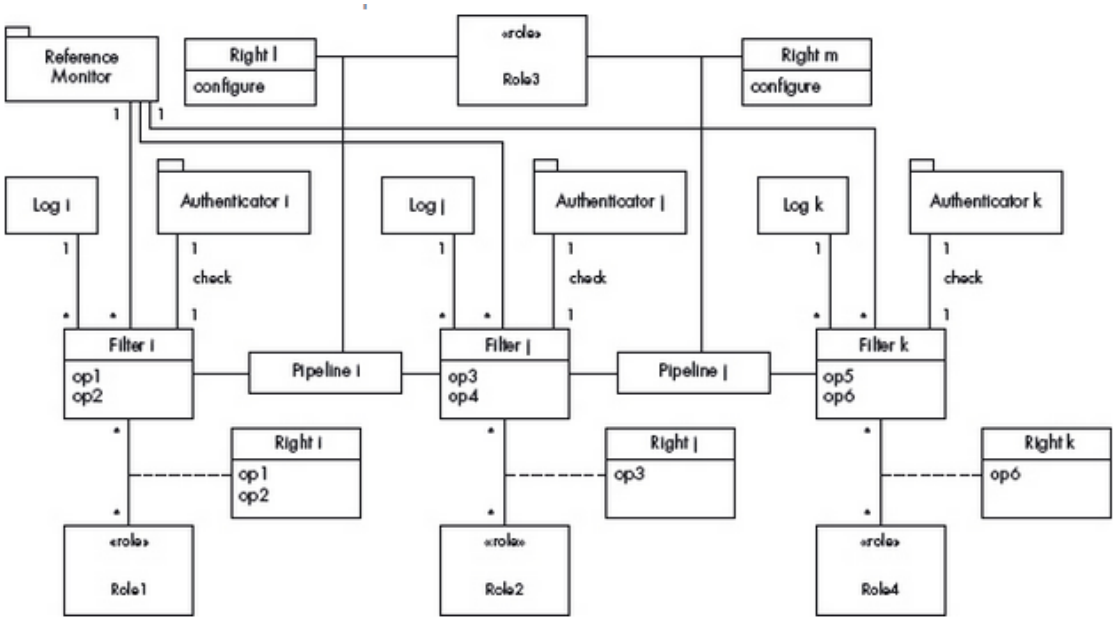


Figura 46. Diagrama de clase del patrón SPP Secure Pipes and Filters

Relación entre catálogos

Proporciona un manejo seguro en flujos de datos. El flujo de datos sería entre actividades desarrolladas por distintas personas en una organización. Por tanto, no está relacionado con un CSP Intercepting Validator, es decir, no es un CJ2EE intercepting filter. Sin embargo, al utilizar este patrón un SPP authenticator, utilizaría también un CSP authentication enforcer.

Ventajas por catálogos

El Patrón SPP Secure Pipes and Filters está relacionado con los patrones SPP Authorization, SPP Role –Based Access Control y SPP Authenticator, mientras el patrón CSP

Authentication Enforcer define tres estrategias para implementar una autenticación: la estrategia del contenedor autenticado, dar autenticación y el módulo de login JAAS.

A destacar en ambos catálogos

SPP

- Se pueden asignar privilegios según las funciones necesarias en cada plataforma de procesamiento.
- Cada plataforma de filtros puede autenticar a los usuarios antes de que ellos sean autorizados a desarrollar funciones específicas.
- El uso de encriptación entre plataformas es posible, agregando la posibilidad de mensajes seguros.
- El rol de administrador puede controlar la reconfiguración de plataformas para acomodar los cambios en el proceso. El sistema es más complejo y necesitara servicios extras que deberán ser agregados.

9.3 Secure Blackboard

Descripción

El patrón SPP Secure Blackboard (Figura 47) describe cómo proporcionar una gestión segura de los datos cuando fuentes conocidas acceden a un recurso compartido. Cada fuente conocida lee los datos de la pizarra, se aplica algún tipo de procesamiento o transformación de datos y se actualiza la pizarra. Con el fin de evitar violaciones de integridad y confidencialidad, los derechos a la lectura y actualización de datos se controlan de acuerdo con los derechos predefinidos, y sus acciones se registran. Las fuentes se autentican antes de poder acceder a la pizarra.

Este patrón permite añadir mecanismos de seguridad para controlar las amenazas y además proporciona una manera de acceder a los datos de pizarra de una variedad de fuentes de conocimiento de una manera segura, mediante la adición de algunos mecanismos de seguridad básicos para el componente de control, que proporciona la autenticación (Authenticator), la

autorización (Role-based Access Control), y la monitorización (Security Logger and Auditor) en cada operación de acceso.

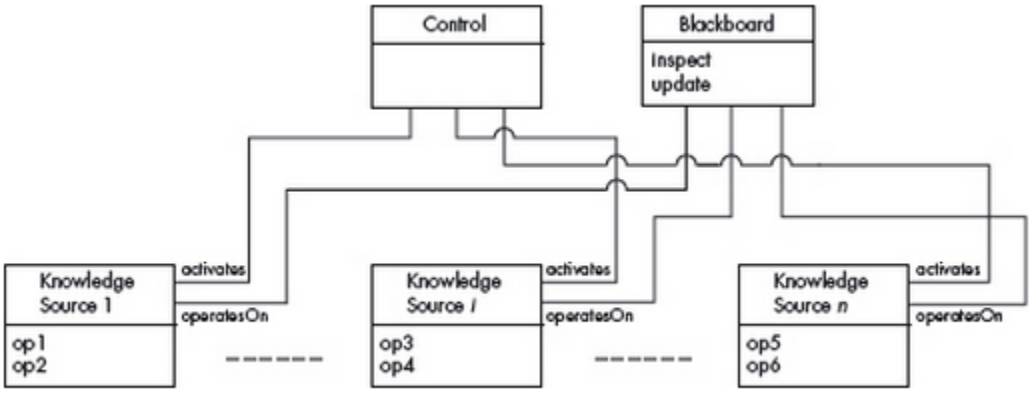


Figura 47. Diagrama de clase del patrón SPP Secure Blackboard

Relación entre catálogos

El patrón SPP Secure Blackboard se relaciona con los patrones SPP Security Logger y SPP Authenticator a su vez tienen su equivalente en el catálogo CSP con los patrones CSP Secure Logger, CSP Intercepting Validator y CSP Authentication Enforcer.

Ventajas por catálogos

En el patrón SPP Secure Blackboard describe un diagrama de objeto del patrón mientras que en el catálogo de patrones relacionados CSP se muestran ejemplos de código.

A destacar en ambos catálogos

SPP

- La fuente de conocimiento puede ser agregada o removida dinámicamente.
- Se pueden definir roles de permisos.
- Los servicios de autenticación pueden validar si la fuente de los datos es legítima y los canales pueden ser encriptados.
- Se puede controlar quien puede reconfigurar la fuente de conocimiento.
- Se puede registrar los accesos al blackboard para una auditoria futura.
- Los tres mecanismos de seguridad incorporados no son suficientes para controlar todos los posibles trucos de seguridad y deben ser completados con mecanismos adicionales.

9.4 Secure Adapter

Descripción

El patrón SPP Secure Adapter (Figura 48) también conocido como Secure Wrapper describe cómo convertir una interfaz de una clase existente en una interfaz más conveniente, mientras preserva la seguridad de la entidad adaptada. Se deben identificar las posibles amenazas para la interfaz adaptada, definiendo políticas y sus correspondientes mecanismos de detención.

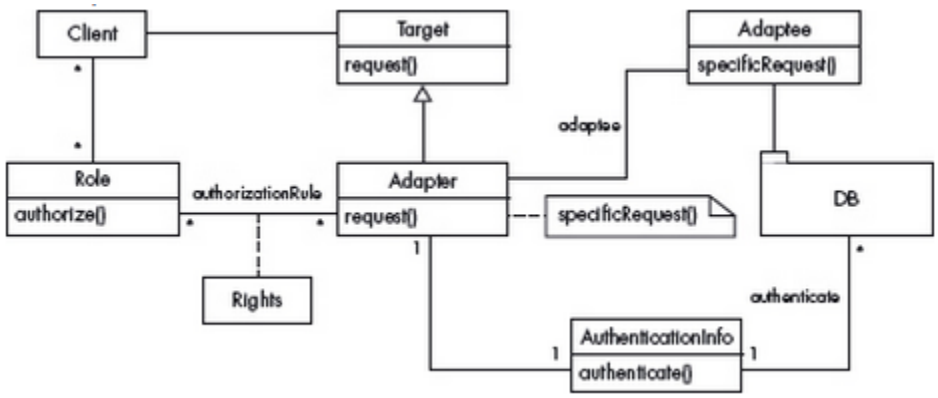


Figura 48. Diagrama de clase del patrón SPP Secure Adapter

Relación entre catálogos

Para servicios de aplicación, el patrón SPP Secure Adapter sería el equivalente de un patrón CSP Secure Façade (Figura 49).

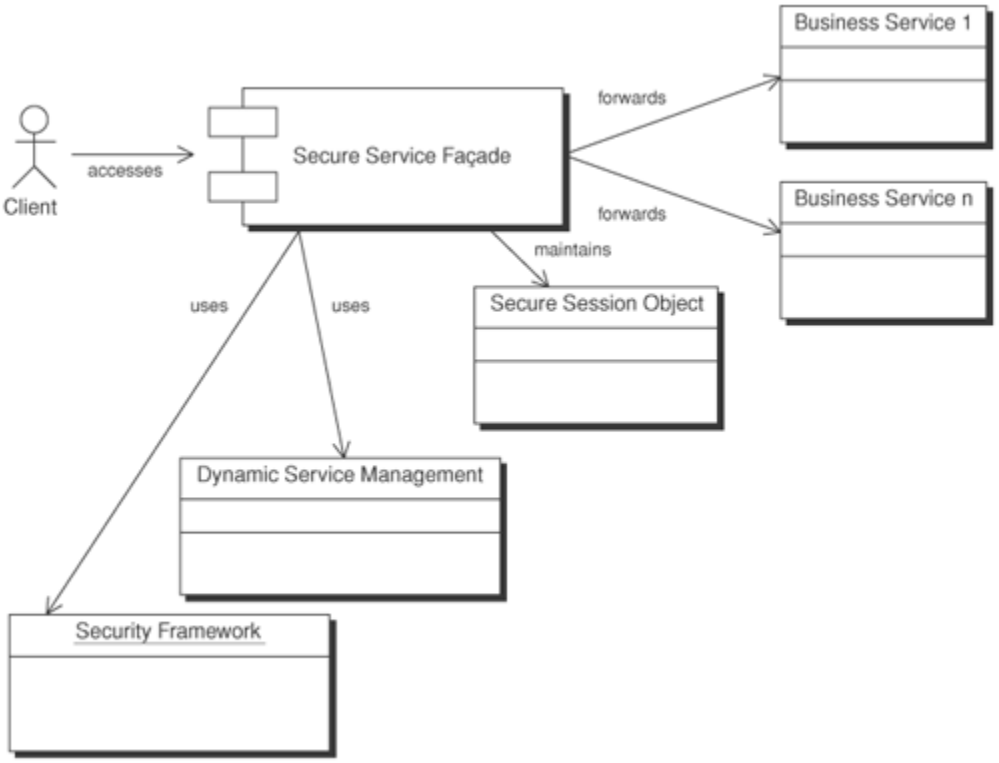


Figura 49. Diagrama de clase del patrón CSP Secure Service Facade

Ventajas por catálogos

El patrón SPP Secure Adapter identifica las posibles amenazas y los mecanismos para detener los problemas de seguridad que se puedan presentar, mientras el patrón CSP Secure Façade presenta ejemplos de código fuente para interfaces remotas y locales.

A destacar en ambos catálogos

SPP

- Los clientes pueden ser autenticados.
- Se puede aplicar la autorización a los requerimientos de los usuarios.
- Se puede utilizar criptografía para evitar ataques en los mensajes utilizados.
- Los micro kernel utilizan adaptadores para adaptar requerimiento de procesos que puedan tener formatos diferentes.
- Los sistemas basados en CORBA utilizan adaptadores para adaptar requerimientos remotos.

CSP

- Autentica los requerimientos de la capa de negocio.
- El patrón Secure Session Façade permite a los desarrolladores insertar auditorias en puntos de entrada y salida de la capa de negocio.

9.5 Secure Distributed Publish/Subscribe

Descripción

El patrón SPP Secure Distributed Publish / Subscribe (Figura 50) describe cómo desacoplar los editores de los eventos de los interesados en los acontecimientos (abonados) en un sistema distribuido. La suscripción y publicación se realizan de forma segura. Usa un canal de eventos seguros mediante el cual los editores envían a sus eventos y los suscriptores interesados pueden recibir los eventos. Los suscriptores deberán registrarse para los eventos en los que estén interesados.

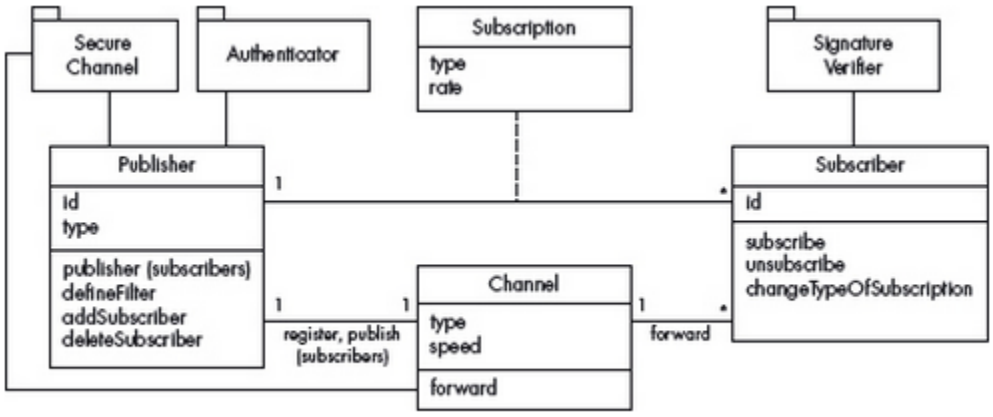


Figura 50. Diagrama de clase del patrón SPP Secure Distributed Publish/Subscribe

Relación entre catálogos

El patrón SPP Secure Distributed Publish/Subscribe incluye tareas de seguridad en un servicio de publicación/suscripción y sólo se relaciona con el catálogo CSP a través de los patrones SPP que contiene, un SPP Authenticator, es decir, un CSP Authentication Enforcer.

Ventajas por catálogos

El patrón SPP Secure Distributed Publish/Subscribe describe los usos conocidos actualmente, mientras que no es mencionado en el catálogo CSP, salvo por los patrones SPP que contiene y son asimilables a patrones CSP.

A destacar en ambos catálogos

SSP

- Transparencia localizada.
- Si los eventos son sensibles se pueden encriptar en el canal del evento, también se puede digitalizar las firmas para autenticarlas.
- En un sistema distribuido puede sufrir un ataque de denegación de servicio el cual no puede ser controlado a nivel de red.

9.6 Secure Model-View –Controller

Descripción

El patrón SPP Secure Model-View-Controller (Figura 51) permite a los usuarios acceder y/o modificar de forma segura la información sensible que se encuentra en el componente del modelo. Los patrones básicos de seguridad se aplican para proporcionar autenticación, autorización, comunicaciones seguras y logging. Además, podría ser necesario para limpiar los datos entrantes y/o salientes, para prevenir los ataques maliciosos tales como la inyección SQL o ataques de *cross-site scripting*.

El control de acceso puede agregarse en el nivel del modelo y/o en los niveles de controlador y vista. Agregando de control de acceso a nivel de controlador y la vista es menos intrusivo para el modelo, sin embargo, es de grano grueso. El control de acceso a nivel de modelo puede ser más fino y podría basarse en los atributos específicos del modelo.

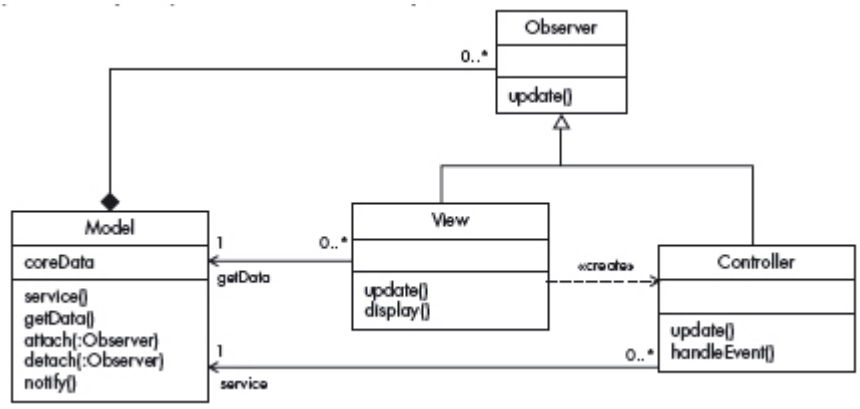


Figura 51. Diagrama de clase del patrón SPP Secure Model-View –Controller

Relación entre catálogos

Es un Model View Controller que lleva a cabo tareas de seguridad tanto en presentación como en negocio. En lo referente a presentación, una CSP Secure Base Action podría llevar a cabo algunas tareas antes de llegar a negocio. También utilizaría un CSP Authentication Enforcer, un CSP Authorization Enforcer, un CSP Intercepting Validator y un CSP Secure Logger.

Ventajas por catálogos

El patrón SPP Secure Model View- Controller describe la estructura del patrón y muestra un diagrama de secuencia para el caso de uso propagación de un cambio para el modelo. El catálogo CSP no lo identifica explícitamente, pero sí que lo supone en otros patrones.

A destacar en ambos catálogos

SPP

- Los controles de seguridad agregan complejidad a la arquitectura.
- Un sistema de autenticación puede confirmar a los usuarios que están en el modelo correcto.
- Un sistema de autorización puede hacer cumplir la confidencialidad y la integridad.
- Un registro de seguridad puede registrar todas las acciones sensitivas de seguridad.

10. PATRONES SIN RELACIÓN

10.1 Patrones SPP sin relación con CSP

10.1.1 Secure Three – Tier Architecture

El patrón SPP Secure Three – Tier Architecture proporciona una arquitectura de aplicaciones en niveles o capas en la que cada nivel ofrece, un nivel diferente de la responsabilidad. Uno tiene que ver con el nivel de la parte de presentación del sistema (interfaces de sistema y de usuario), otro se ocupa de la lógica de negocio - el núcleo del sistema - y el último nivel se ocupa del almacenamiento de datos.

Este patrón representa la arquitectura multicapa con cuestiones de seguridad y sería como una especie de compendio de todo el catálogo CSP. Es muy discutible por tanto que se pueda considerar como un patrón, al ser más bien una arquitectura software.

10.1.2 Secure Enterprise Service Bus

El Patrón SPP Secure Enterprise Service Bus proporciona una infraestructura conveniente para integrar una variedad de servicios distribuidos y componentes relacionados de una manera sencilla y segura. Introduce una estructura de bus común que proporciona funciones básicas de negociación, así como un conjunto de otros servicios apropiados.

Este patrón un *Enterprise Service Bus* que tiene en cuenta tareas de seguridad. A pesar de guardar cierta relación con el patrón CSP Secure Message Router, es demasiado genérico como para asimilarlo a ningún patrón CSP.

10.1.3 Worm

El patrón del gusano describe como un gusano puede propagarse por muchos lugares causando daños. Esto sucede cuando un usuario abre un archivo adjunto de un mensaje o ejecuta un archivo y el núcleo del gusano comienza a ejecutarse. Alternativamente, realiza la invasión a través de un puerto sin protección o defectuoso y comienza a descargar porciones restantes de los de otros sitios o redes. Utiliza algún procedimiento para ocultar la estructura del gusano.

En el catálogo CSP no se menciona un patrón del uso incorrecto para la identificación de amenazas. El patrón SPP Worm es una mejora sistemática que ofrece el autor en el catálogo SPP haciendo un análisis de las actividades de los casos de uso y postulando las posibles amenazas.

10.1.4 Denial of Service in VoIP

Los ataques VoIP DoS oprimen los recursos limitados con la finalidad de interrumpir las operaciones con VoIP, usualmente inundando al usuario de mensajes, provocando una respuesta degradada. Uno de los métodos para lanzar un ataque DOS es inundar un servidor de VoIP (por ejemplo Gatekeeper) con solicitudes repetidas de un servicio legítimo en un intento de sobrecargarlo. Esto puede causar una degradación grave o indisponibilidad total del servicio de voz.

Un ataque de inundación también puede ser lanzado contra los teléfonos IP, gateways o cualquiera de los componentes de la red de VoIP que aceptan señalización. Con esta forma de ataque DOS, el sistema de destino procesa tantos paquetes ilegítimos que será incapaz de procesar los paquetes legítimos. Los atacantes también pueden usar el ataque de inundación TCP SYN (también conocido como ataque de agotamiento de recursos) para obtener resultados similares. Este ataque inunda el puerto con los paquetes de sincronización, que normalmente se utilizan para iniciar una conexión.

En el catálogo CSP no se menciona un patrón del uso incorrecto para la identificación de amenazas. El patrón SPP Denial of Service in VoIP es una mejora sistemática que ofrece el autor en el catálogo SPP haciendo un análisis de las actividades de los casos de uso y postulando posibles amenazas.

10.1.5 Spoofing Web Service

Un servicio web suplantado trata de imitar la identidad del usuario para robar sus credenciales. Las normas de seguridad tales como WS-Security utilizan las credenciales del usuario para proteger la comunicación entre los servicios web mediante la firma y el cifrado de mensajes. Si un atacante logra obtener credenciales válidas, pueden comunicarse con otros servicios web. En este ataque, el usuario no sabe que sus credenciales han sido robadas hasta que el daño está hecho (como alteración de la información, el acceso a sus recursos, la alteración de privilegio, incluso adjuntando el código malicioso que podría ser perjudicial para el servidor).

Cuando el atacante tiene las credenciales, se puede utilizar el archivo WSDL para descubrir las políticas de seguridad de los servicios web y crear un mensaje válido. Esta situación

está condicionada a que el atacante obtiene las credenciales de un usuario, el usuario valido no conoce del robo y no ha informado del mismo.

En el catálogo CSP no se menciona un patrón del uso incorrecto para la identificación de amenazas. El patrón SPP Spoofing Web Service es una mejora sistemática que ofrece el autor haciendo un análisis de las actividades de los casos de uso y postulando posibles amenazas.

10.1.6 Infrastructure as a Service (IaaS)

Es una estructura que se compone de muchos servidores, almacenamiento y red, que puede ser compartida por varios usuarios y se puede acceder a través de Internet. Estos recursos se proporcionan a los usuarios en forma de infraestructura-como-un-servicio (IaaS). IaaS se basa en la tecnología de virtualización, que crea recursos unificados que pueden ser compartidos por diferentes aplicaciones. Esta capa base - IaaS - puede ser utilizada como una referencia para los requisitos no funcionales.

En el catálogo CSP no se menciona un patrón que se utilice en la arquitectura de la nube. El patrón SPP Infrastructure as a Service es parte del catálogo de patrones en la nube que ofrece el autor implicando el desarrollo de nuevos patrones de seguridad. No obstante, difícilmente puede considerarse un patrón de seguridad en sí mismo.

10.1.7 Platform as a Service (PaaS)

PaaS ofrece entornos de ejecución virtual con herramientas compartidas y bibliotecas para el desarrollo y despliegue de aplicaciones en la nube. PaaS utiliza IaaS como capa de base (servidores, almacenamiento y red), y oculta la complejidad de la gestión por debajo de la infraestructura

En el catálogo CSP no se menciona un patrón que se utilice en la arquitectura de la nube. El patrón SPP Platform as a Service es parte del catálogo de patrones en la nube que ofrece el autor implicando el desarrollo de nuevos patrones de seguridad. No obstante, difícilmente puede considerarse un patrón de seguridad en sí mismo.

10.1.8 Software as a Service (SaaS)

Las aplicaciones SaaS se entregan como un servicio a los usuarios normalmente a través de Internet vía navegadores web o APIs. Los SaaS basados en la nube permiten acceder a los usuarios a las aplicaciones demandadas, en el cual la computación y el almacenamiento están alojados en la nube sin necesidad de instalar ningún software en sus equipos locales. SaaS se puede desarrollar y distribuir mediante la plataforma-como-un-servicio (PaaS) o de infraestructura como servicio (IaaS).

En el catálogo CSP no se menciona un patrón que se utilice en la arquitectura de la nube. El patrón SPP Software as a Service es parte del catálogo de patrones en la nube que ofrece el autor implicando el desarrollo de nuevos patrones de seguridad. No obstante, difícilmente puede considerarse un patrón de seguridad en sí mismo.

10.1.9 Patrones para la seguridad de sistemas operativos

El catálogo SPP considera tres categorías de patrones para asegurar sistemas operativos:

- Gestión de procesos seguros.
- Ejecución segura y gestión de archivos.
- Arquitectura y administración de sistemas operativos seguros.

Son patrones específicos del dominio de sistemas operativos y no tienen su contrapartida en el catálogo CSP, centrado en patrones de seguridad para aplicaciones empresariales, las cuales se construyen sobre sistemas operativos.

10.2 Patrones CSP sin relación con SPP

10.2.1 CSP Container Managed Security

Define los roles a nivel de aplicación en tiempo de desarrollo y desempeña los mapeos de roles de usuarios en el tiempo de despliegue o a partir de ahí.

Los servidores de aplicaciones no son considerados en el catálogo SPP, por lo que no tiene contrapartida en este catálogo.

10.2.2 CSP Dynamic Service Management

Es una aplicación de Java Management Extensions (JMX) con fines de monitorización. Es un patrón peculiar y muy específico de entornos J2EE. Por lo tanto, no tiene su contrapartida en el catálogo SPP.

10.2.3 CSP Obfuscated Transfer Object

Utiliza un objeto de transferencia ofuscado para proteger el acceso de datos dentro y entre las capas. Permite a los desarrolladores definir los elementos dentro de los datos que serán protegidos. Esto significa que la protección puede variar entre aplicaciones o implementaciones, dependiendo de los requerimientos del negocio.

En el catálogo SPP no hacen mayor referencia a la capa de negocio por tal motivo no tiene su equivalente en este catálogo con el patrón CSP Obfuscated Transfer Object.

10.2.4 CSP Policy Delegate

Este patrón localiza de forma segura la lógica de negocio remota de manera oculta para sus clientes. En el catálogo SPP no hacen mayor referencia a la capa de negocio por tal motivo no tiene su equivalente en este catálogo con el patrón CSP Policy Delegate.

10.2.5 CSP Secure Session Object

Es un mecanismo abstracto para encapsular la autenticación y la autorización como credenciales, roles y privilegios. Estos se utilizan como un transporte seguro, que permite a los componentes cruzar a través de las capas o el sistema de mensajería asíncrono para verificar que el que creo el requerimiento es autenticado y autorizado por un servicio en particular

En el catálogo SPP no hacen mayor referencia a la capa de negocio por tal motivo no tiene su equivalente en este catálogo con el patrón CSP Secure Session Object.

10.2.6 CSP Intercepting Validator

Es un filtro web que se encarga de hacer validaciones de seguridad tales como la detección de un ataque por inyección SQL.

En el catálogo SPP no se menciona un patrón relacionado a la validación de datos de negocio y datos de tipo string, entero, formato, largo, rango, valores nulos y legales antes de invocar una transacción.

10.2.7 CSP Secure Service Proxy

Este patrón intenta asegurar y controlar el acceso a los componentes J2EE expuestos como puntos finales de los servicios web. Estos actúan como un proxy proveyendo una interfase común para los componentes del proveedor de servicio.

En el catálogo SPP no se menciona un patrón relacionado con J2EE que sea su equivalente.

10.2.8 CSP Intercepting Web Agent

Este patrón ayuda a proteger aplicaciones Web a través de un agente web que intercepta requerimientos en el servidor web y provee autenticación, autorización, encriptación y capacidades de auditoria.

En el catálogo SPP no existe un patrón que coincida con CSP Intercepting Web Agent aunque si bien es cierto que las relaciones que mantiene proveyendo autenticación, autorización, encriptación y capacidades de auditoria si se encuentran explicitas en el catálogo. La diferencia fundamental es que el patrón CSP está explícitamente basado en una tecnología J2EE que no es considerada, ni generalizada por el catálogo SPP.

10.2.9 CSP Password Synchronizer

Es un mecanismo que describe como realizar una sincronización de contraseñas de forma segura a través de múltiples aplicaciones.

En el catálogo SPP no hacen referencia al diseño, estrategias y buenas prácticas para cambios de contraseñas simultáneos, por tal motivo no tiene su equivalente en este catálogo.

11. CONCLUSIONES Y TRABAJO FUTURO

La comparativa realizada en este trabajo entre los catálogos SPP y CSP ha producido diversas aportaciones.

La primera y más directa es trazar una equivalencia entre los patrones definidos en ambos catálogos. Esta equivalencia se ha mostrado como dos tablas de conversión, una que organizada por categorías SPP considera sus equivalentes CSP (Tabla 1), y otra que considerando las categorías CSP considera sus equivalentes SPP (Tabla 2). En importante considerar que en este trabajo no se han tenido en cuenta los patrones de sistemas operativos presentados en el catálogo SPP, ya que claramente, no tenían relación con los patrones del catálogo CSP.

Tabla 1. Conversión del catálogo SPP al catálogo CSP

Patrones SPP	Relación con el catálogo CSP
Patterns for Identity Management	
Circle of Trust	Mencionado como elemento de Liberty Alliance
Identity Provider	Utilizado por: Message Inspector,Message interceptor gateway, Secure message router, Single sign-on delegator
Identity Federation	Mencionado como elemento de Liberty Alliance y en SAML
Liberty Alliance Identity Federation	Identificado como una tecnología a tener en cuenta
Patterns for Authentication	
Authenticator	Authentication Enforcer
Remote Authenticator/Authorizer	Versión remota del Authenticator Enforcer y Authorization Enforcer
Credential	Implícito en Authentication Enforcer y Authorization Enforcer, aunque no explícitamente documentado
Patterns for Access Control	**Patterns for Access Control**
Authorization	Utilizado por: Authorization Enforcer
Role-Based Access Control	Utilizado por: Authorization Enforcer
Multilevel Security	Utilizado por: Authorization Enforcer
Policy-Based Access Control	Utilizado por: Authorization Enforcer. Presente también en la descripción de SAML y XACML
Access Control List	Utilizado por: Authorization Enforcer
Capability	Utilizado por: Authorization Enforcer
Reified Reference Monitor	Utilizado por: Authorization Enforcer

Controlled Access Session	Utilizado por: Authorization Enforcer
Session-Based Role-Based Access Control	Utilizado por: Authorization Enforcer
Security Logger and Auditor	Secure Logger. Utilizador por: Audit Interceptor
Patterns for Secure Process Management	
Secure Process/Thread	X
Controlled-Process Creator	X
Controlled-Object Factory	X
Controlled-Object Monitor	X
Protected Entry Points	X
Protection Rings	X
Patterns for Secure Execution and File Management	
Virtual Address Space Access Control	X
Execution Domain	X
Controlled Execution Domain	X
Virtual Address Space Structure Selection	X
Patterns for Secure OS Architecture and Administration	**Patterns for Secure OS Architecture and Administration**
Modular Operating System Architecture	X
Layered Operating System Architecture	X
Microkernel Operating System Architecture	X
Virtual Machine Operating System Architecture	X
Administrator Hierarchy	X
File Access Control	X
Security Patterns for Networks	
Abstract Virtual Private Network	Utiliza: Authentication Enforcer
IPSec VPN	Utiliza: Authentication Enforcer
TLS Virtual Private Network	Utiliza: Authentication Enforcer
Transport Layer Security	Secure Pipe
Abstract IDS	Generaliza: Message Interceptor Gateway
Signature-Based IDS	Generaliza: Message Interceptor Gateway
Behavior - Based IDS	Generaliza: Message Interceptor Gateway
Patterns for Web Services Security	

Application Firewall	Generaliza: Message Interceptor Gateway
XML Firewall	Message Interceptor Gateway
XACML Authorization	Utilizado por: Message Inspector
XACML Access Control Evaluation	Utilizado por: Message Inspector
Web Services Policy Language	Utilizado por: Secure Message Router, Message Inspector
WS-Policy	Utilizado por: Secure Message Router, Message Inspector
WS-Trust	Utilizado por: Message Interceptor Gateway
SAML Assertion	Utilizado por: Message Inspector, Assertion Builder, Single Sign-On Delegator,Credential Tokenizer
Patterns for Web Services Cryptography	
Symmetric Encryption	Identificado como tecnología de seguridad. Utilizado por: Secure Message Router
Asymmetric Encryption	Identificado como tecnología de seguridad. Utilizado por: Secure Message Router
Digital Signature with Hashing	Identificado como tecnología de seguridad. Utilizado por: Secure Message Router
XML Encryption	Identificado como tecnología de seguridad. Utilizado por: Interceptor Gateway, Message Inspector
XML Signature	Identificado como tecnología de seguridad. Utilizado por: Interceptor Gateway y Message Inspector
WS-Security	Identificado como tecnología de seguridad. Utilizado por Interceptor Gateway, Message Inspector
Patterns for Secure Middleware	
Secure Broker	Generaliza: Secure Message Router
Secure Pipes and Filters	Utiliza: Authentication Enforcer
Secure Blackboard	Utiliza: Secure Logger, Intercepting Validator, Authentication Enforcer,
Secure Adapter	Secure Service Façade
Secure Three-Tier Architecture	X (sin relación con ningún patrón concreto, representa todo el catálogo CSP)
Secure Enterprise Service Bus	X (sin relación con ningún patrón concreto, representa buena parte del catálogo CSP)
Secure Distributed Publish/Subscribe	Utiliza: Authentication Enforcer
Secure Model-View-Controller	Utiliza: Secure Base Action, Authentication Enforcer, Authorization Enforcer, Secure Logger, Intercepting Validator
Misuse Patterns	
Worm	X
Denial-of-Service in VoIP	X
Spoofing Web Services	X

Patterns for Cloud Computing Architecture	
Infrastructure-as-a-Service	X
Platform-as-a-Service	X
Software-as-a-Service	X

Tabla 2. Conversión del catálogo CSP al catálogo SPP

Patrones CSP	Relación con el catálogo SPP
Web Tier Security Patterns	
Authentication Enforcer	Authenticator. Es la versión no remota de un Remote Authenticator/Authorizer Enforcer Utilizado por: Credential, Abstract Virtual Private Network, IPSec VPN, TLS Virtual Private Network, Secure Pipes and Filters, Secure Blackboard, Secure Distributed Publish/Subscribe, Secure Model-View-Controller
Authorization Enforcer	Authorization Versión no remota de un Remote Authenticator/Authorizer Utiliza: Credential, Role-Based Access Control, Multilevel Security, Policy-Based Access Control,Access Control List, Capability, Reified Reference Monitor, Controlled Access Session, Session-Based Role-Based Access Control Usado por Secure Model-View-Controller
Intercepting Validator	Utilizado por: Secure Blackboard, Secure Model-View-Controller
Secure Base Action	Utilizado por: Secure Model-View-Controller
Secure Logger	Security Logger and Auditor Utilizado por: Secure Blackboard, Secure Model-View-Controller
Secure Pipe	Transport Layer Security
Secure Service Proxy	X
Intercepting Web Agent	X
Business Tier Security Patterns	
Audit Interceptor	Utiliza: Security Logger and Auditor
Container Managed Security	X
Dynamic Service Management	X

Obfuscated Transfer Object	X
Policy Delegate	X
Secure Service Façade	Secure Adapter
Secure Session Object	X
Web Services Tier Security Patterns	
Message Interceptor Gateway	XML Firewall Utiliza: Identity Provider, WS-Trust Especialización de: Abstract IDS, Signature-Based IDS, Behavior - Based IDS, Application Firewall
Message Inspector	Utilizado por: Identity Provider Utiliza: XACML Authorization, XACML Access Control Evaluation, Web Services Policy Language, WS-Policy, SAML Assertion, XML Encryption, XML Signature, WS-Security
Secure Message Router	Utiliza: Identity Provider, Web Services Policy Language, WS-Policy, Symmetric Encryption, Asymmetric Encryption, Digital Signature with Hashing Especializa: Secure Broker
Identity Management & Service Provisioning	
Assertion Builder Pattern	Utiliza: SAML Assertion
Single Sign-on (SSO) Delegator Pattern	Utiliza: Identity Provider, SAML Assertion
Credential Tokenizer Pattern	Utiliza: SAML Assertion
Password Synchronizer Pattern	X

Las tablas anteriores reflejan que 44 patrones SPP, un 65%, mantienen relación con alguna tecnología o patrón CSP. Además, 14 patrones CSP, un 63,6%, mantienen relación con algún patrón SPP. Por tanto, parece razonable afirmar que, a pesar de ser catálogos distintos, ambos catálogos están bastante relacionados. Los apéndices A y B reflejan la información recogida en estas tablas en términos de diagramas de clase UML.

Las Tablas 3 y 4 contabilizan el número de patrones con alguna relación por catálogo y capa.

Tabla 3. Número y porcentaje de patrones SPP relacionados con CSP por capas SPP

Categoria SPP	Nro. de Patrones CSP	Porcentaje
Patterns for Identity Management	4	100%
Patterns for Authentication	3	100%
Patterns for Access Control	10	100%
Patterns for Secure Process Management	0	0%
Patterns for Execution File Management	0	0%
Patterns for OS Architecture and Administration	0	0%
Security Patterns for Networks	7	100%
Patterns for Web Services Security	8	100%
Patterns for Web Services Cryptography	6	100%
Patterns for Secure Middleware	6	75%
Misuse Patterns	0	0%
Patterns for Cloud Computing Architecture	0	0%
Total	**44**	**65%**

Tabla 4. Número y porcentaje de patrones CSP relacionados con SPP por capas CSP

Categoria CSP	Nro. de Patrones SPP	Porcentaje
Web Tier Security Patterns	6	75%
Business Tier Security Patterns	2	28,5%
Web Services Tier Security Patterns	3	100%
Identity Management & Service Provisioning	3	75%
Total	**14**	**63,6%**

Por capas SPP, vemos que, exceptuando las capas de sistemas operativos, misuse y cloud computing, casi todos los patrones SPP tienen alguna relación con los patrones CSP. Quedan también sin relación aquellos patrones de la capa de middleware que están más cercanos a arquitecturas que a patrones. Por capas CSP, la capa de negocio es la que menos relación tiene con patrones SPP.

En general el catálogo SPP considera como patrones cuestiones consideradas como

tecnologías o arquitecturas en el catálogo CSP. Además, el catálogo SPP describe los patrones de una forma resumida, sin incluir ejemplos de código fuente, diagramas de secuencias ni explicaciones más tangibles para desarrolladores y programadores, a diferencia del catálogo CSP.

Los patrones SPP son más genéricos que los CSP y pueden utilizarse en aplicaciones genéricas que requieran seguridad, mientas que los patrones CSP están centrados aplicaciones empresariales.

Por tanto, en el contexto de asignaturas de ingeniería del software qué se centran en arquitecturas multicapa que deben incluir patrones de seguridad, parece más adecuado el catálogo CSP, ya que define patrones específicos para este tipo de aplicaciones, en especial para la capa de negocio. No obstante este catálogo incluye muchas tecnologías que también deberían ser explicadas para comprender los patrones CSP, que en el catálogo SPP están consideradas como patrones.

Con respecto al trabajo futuro, a pesar de haber establecido una relación entre dos de los principales catálogos de seguridad quedan todavía por relacionar otros catálogos de patrones, tanto de seguridad como arquitectónicos. En este sentido sería interesante analizar la relación entre el catálogo CJP y CJ2EE, estableciendo relaciones entre cada capa de ambos catálogos. También sería interesante analizar la posibilidad de utilizar la aproximación de desarrollo dirigido por modelos para incluir patrones de seguridad en diseños de arquitectura multicapa de manera automática.

12. REFERENCIAS

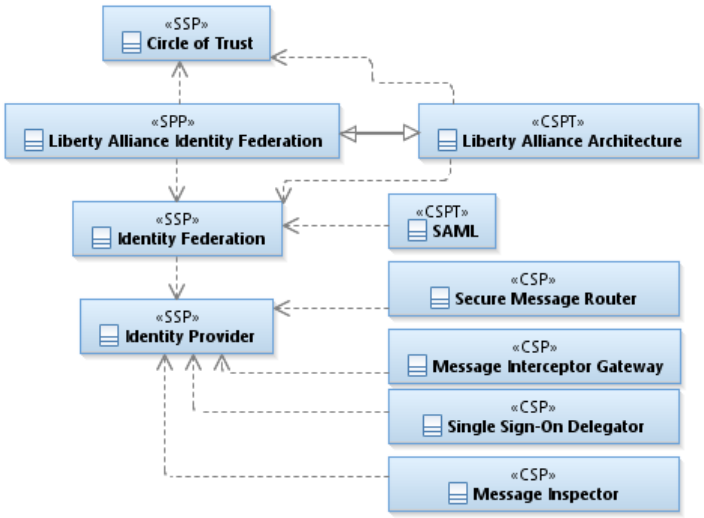
1. Alur, D.; Crupi, J. y Malks, D. (2003). *Core J2EE patterns: Best practices and design strategies* (2nd. ed, 9th printing ed.). Upper Saddle River (New Jersey): Prentice Hall PTR,.
2. Beck, Kent; Cunningham, Ward. (1987). Submitted to the OOPSLA-87 workshop on the Specification and Design for Object-Oriented Programming. Technical Report No. CR-87-43
3. Buschmann, F.; Meunier, Regine; Rohnert, Hans; Sommerland, Peter; Stal, Michael.(2002). *Pattern-oriented software architecture. A system of patterns.* (Repr. ed.). Chichester (West Sussex): John Wiley & Sons.
4. Buschmann, F.; Henney, Kevlin; C. Schmidt, Douglas. (2007). *Pattern-oriented software architecture, volume 4: A pattern language for distributed computing.* Chichester: John Wiley & Sons.
5. Buschmann, F.; Henney, Kevlin; C. Schmidt, Douglas. (2007). *Pattern-oriented software architecture. Volume 5: On patterns and pattern languages.* Chichester, England; Hoboken, NJ: J. Wiley.
6. Crawford, W. (2003). *J2EE design patterns* (1st ed.). Sebastopol (California): O'Reilly.
7. Erl, T. (2008). *SOA design patterns* (1st ed.). Upper Saddle River, N.J.: Prentice Hall.
8. Fernandez, E. B. (2013). *Security patterns in practice: Designing secure architectures using software patterns*. Chichester, West Sussex: John Wiley & Sons Ltd.
9. Fowler, M; Rice, David; Foemmel, Matthew; Hieatt; Edward, Mee; Robert y Stafford; Randy. (2002). *Patterns of enterprise application architecture* (13th print. ed.). Boston: Addison-Wesley.

10. Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. (1995). *Design patterns. Elements of reusable object-oriented software*. Reading, Mas: Addison-Wesley Publishing Company.
11. Jendrock, E. (2013). *The java EE 6 tutorial advanced topics* (4th ed.). Upper Saddle River, NJ: Addison-Wesley.
12. Kalin, M. (2013). *Java web services up and running* (2nd ed.). Sebastopol, Calif., O'Reilly.
13. Kurniawan, B. (2012). *Servlet and JSP a tutorial* (1st ed.). Brossard, Quebec, Canada. Brainy Software.
14. M. Hafiz, P. Adamczyk; R. E. Johnson (2013). *Growing a pattern language (for security)*. Proceedings of the 18th Conference on Pattern Languages of Programs (PLoP),
15. Navarro, A., Cristobal, J., Fernández-Chamizo, C., & Fernández-Valmayor, A. (2012). *Architecture of a multiplatform virtual campus. Software: Practice and Experience 42(10): 1229-1246*
16. Panda, D. (2007). *EJB 3 in action*. Greenwich, Conn.: Manning Publications Co.
17. Schmidt, D.; Rohnert, H.; Stal, M.; Buschmann, F. (2005). *Pattern-oriented software architecture, volume 2: Patterns for concurrent and networked objetcts* (1st ed.). Chichester: John Wiley and Sons.
18. Schumacher, M.; Fernandez-Buglioni, E.; Hybertson, D.; Buschmann, F.; Sommerlad, P. (2006). *Security Patterns: Integrating Security and Systems Engineering.*Wiley Software Patterns Series. England
19. Steel, C.; Nagappan, R.; Lai, R. (2005). *Core security patterns: Best practices and strategies for J2EE, web services, and identity management*. Upper Saddle River, NJ: Prentice Hall PTR.

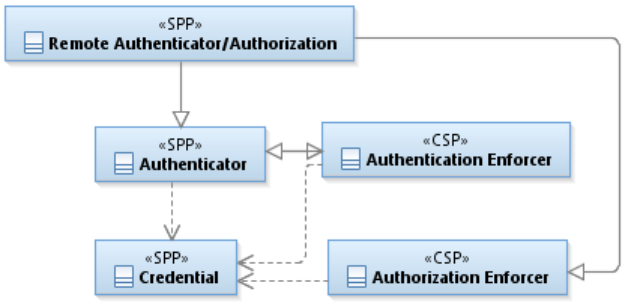
20. Yoder Joseph; Barcalow, Jeffrey. (1998). Workshopped at PloP'97. Monticello, Illinois USA

APÉNDICE A – DIAGRAMAS DE CLASE UML RELACIÓN SPP A CSP

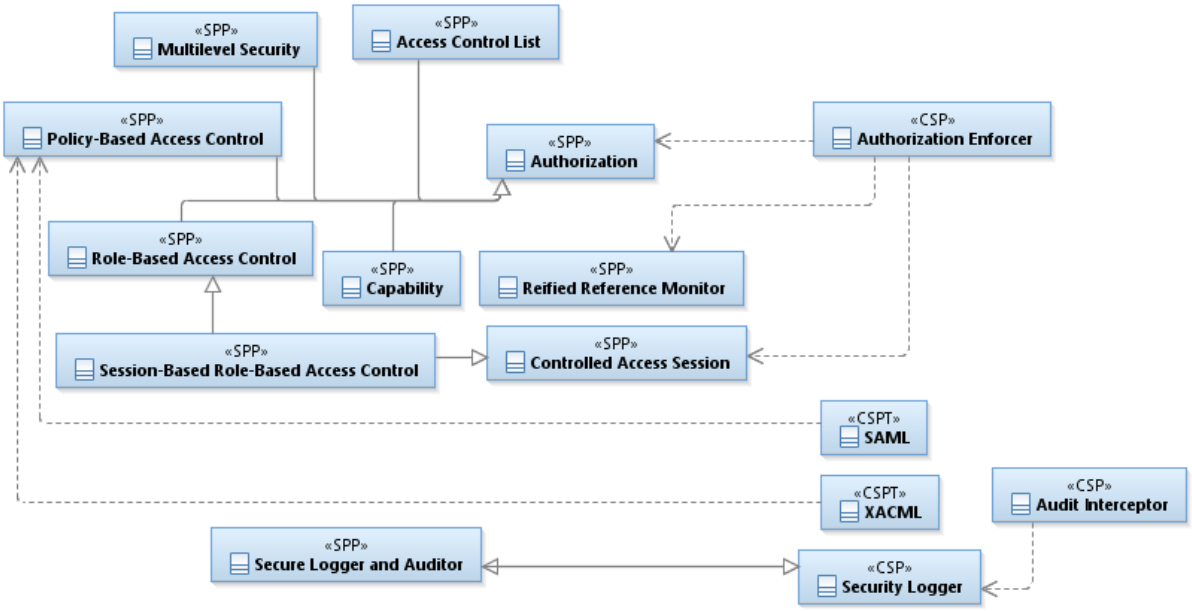
A.1 Identity Management



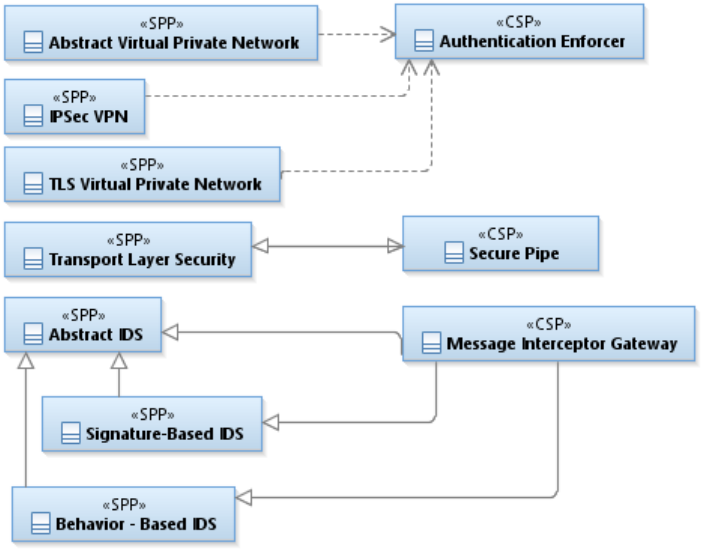
A.2 Authentication



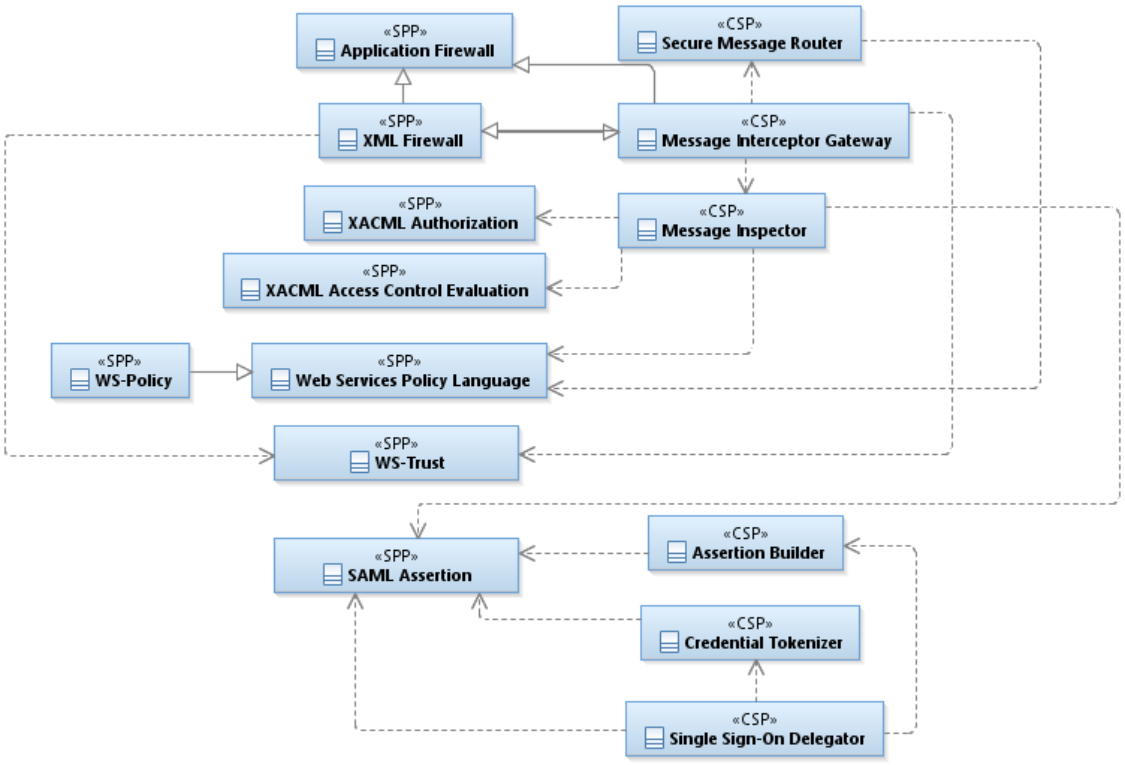
A.3 Access Control



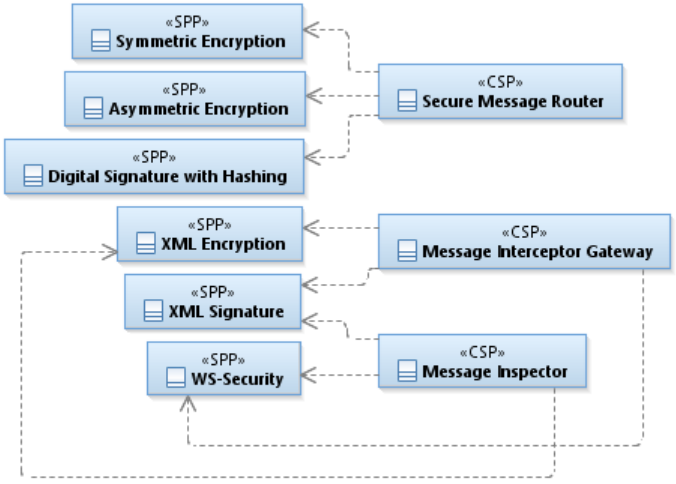
A.4 Networks



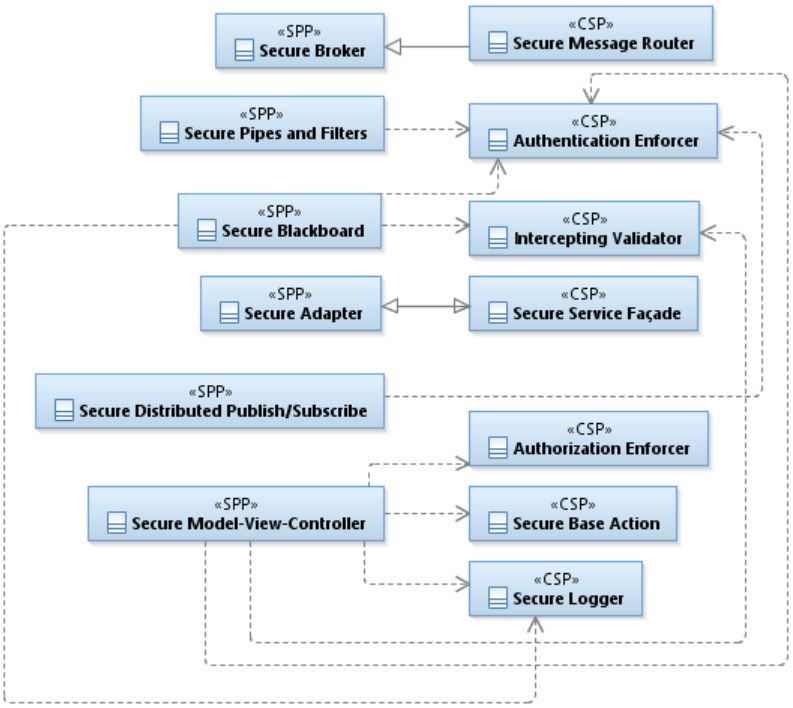
A.5 Web Services Security



A.6 Web Services Cryptography

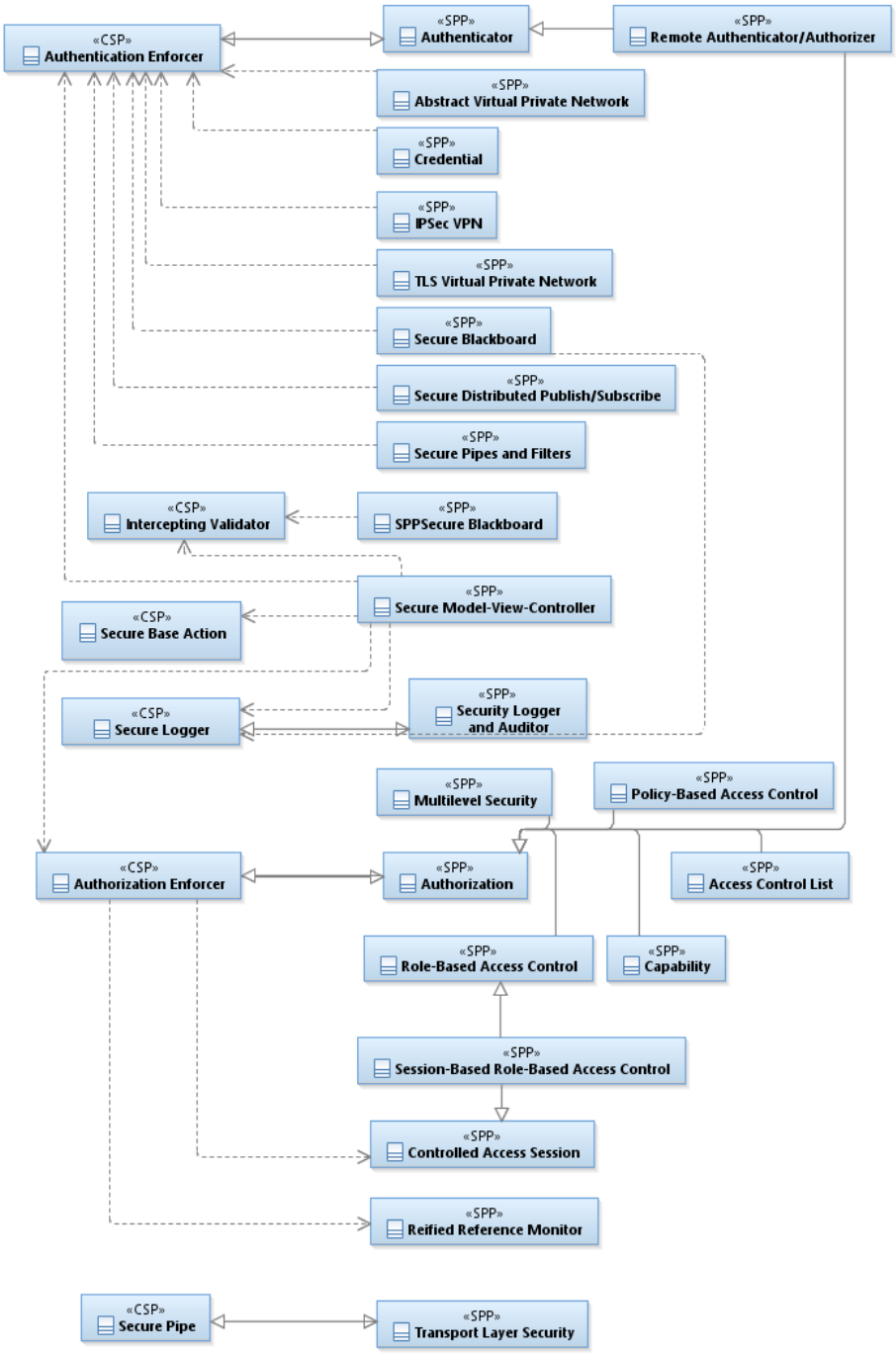


A.7 Secure Middleware

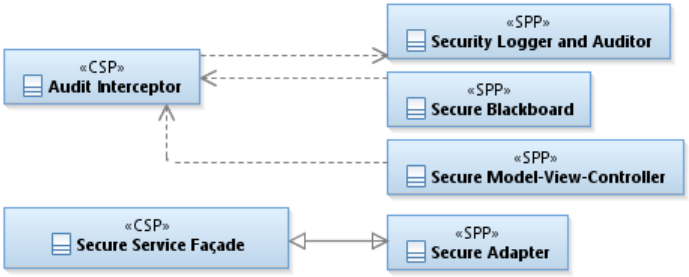


APÉNDICE B - DIAGRAMAS DE CLASE UML RELACIÓN CSP A SPP

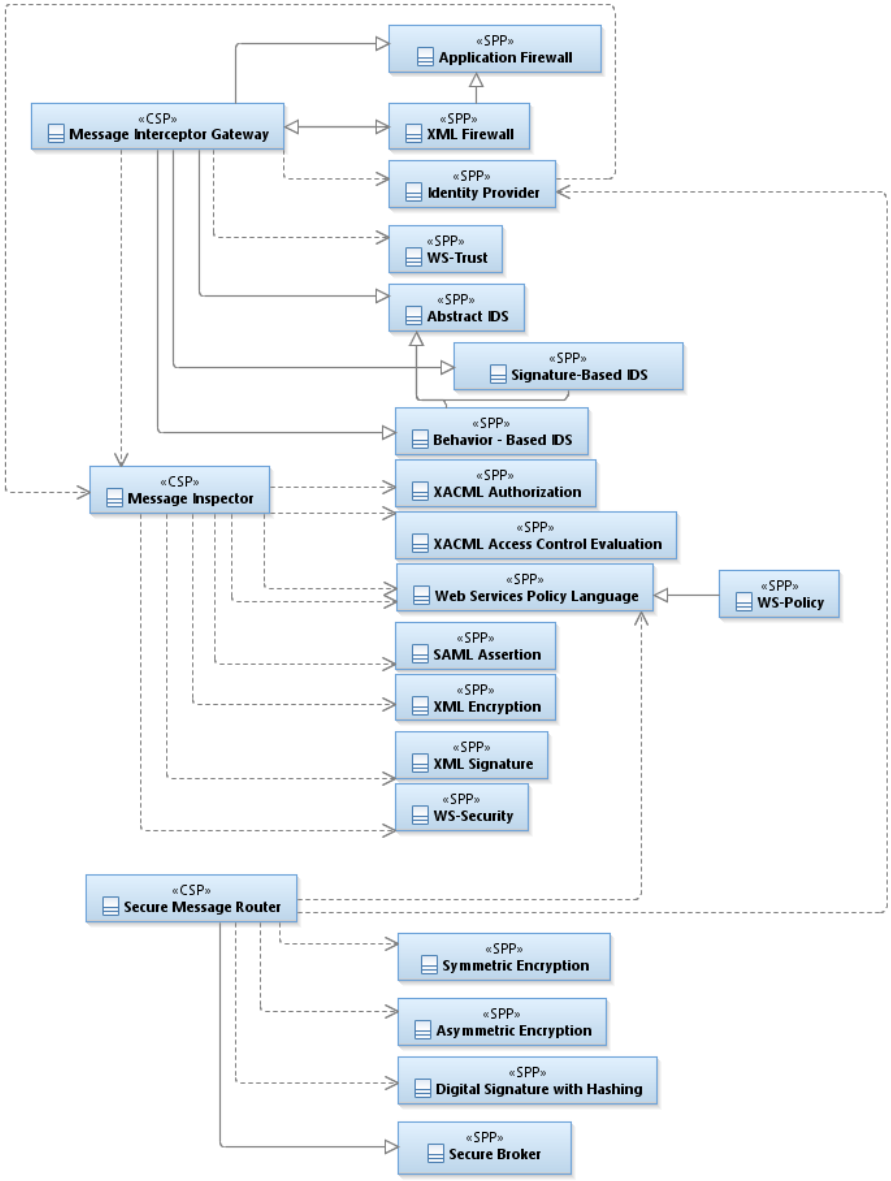
B.1 Web Tier



B.2 Business Tier



B.3 Web Services Tier



B.4 Identity Management and Service Provisioning