Modelo de gestión de servicios PKI

Este trabajo de grado se publica en homenaje póstumo a Juan Carlos Huertas Amaya, quien fuera nuestro permanente colaborador.*

Diana Carolina Valbuena P.
Edgar Hernán López C.

Esta investigación tiene como objetivo general, diseñar e implementar un modelo orientado a servicios, haciendo uso de los protocolos y estándares que enmarcan la arquitectura SOA, para poner a disposición los servicios de una Infraestructura de Llaves Públicas (PKI).

La primera fase abordada para lograr tal objetivo fue la recopilación de información, pues para poder llegar a proponer el modelo orientado a servicios del que se habla, era necesario comprender y dominar varios temas donde los principales fueron PKI, SOA, Web Services y J2EE.

El primer tema estudiado fue la infraestructura de llaves públicas (PKI); aquí fue necesario conocer el concepto, cuáles son sus componentes, como se relacionan entre ellos, y conocer la importancia y aporte de esta tecnología en la seguridad informática de una organización.

Dado que este trabajo se desarrolló en el Banco de la República de Colombia, se entró a conocer en detalle la arquitectura PKI que allí tienen establecida, y la complejidad del producto comercial Entrust Authority adoptado por dicha organización para ofrecer funcionalidades PKI.

Una vez concluido el tema PKI el tema a abordar fue SOA (service oriented architecture), pues este es el marco general que envuelve cualquier implementación específica de un esquema orientado a servicios. Aquí la meta fue comprender los principios que soportan este enfoque, identificar los componentes que conforman cualquier arquitectura orientada a servicios y cómo éstos se relacionan, para concluir con la investigación de algunas formas posibles de lograr SOA.

Posteriormente, se profundizó en una tecnología particular existente para implementar lo que propone SOA: Web Services. La investigación en este tema fue extensa y detallada dado que fue la forma escogida para plantear el modelo orientado a servicios. Los estándares y protocolos de los Web Services fueron el foco de estudio; adicionalmente se plantearon y respondieron las siguientes interrogantes ¿cuáles son los componentes?, ¿cómo se describen, publican y localizan los servicios?, ¿qué ventajas ofrece esta tecnología?, ¿por qué los servicios Web son una forma válida para la implementación de Arquitecturas Orientadas a Servicios?

Finalmente, era necesario elegir una plataforma que permitiera llevar a la implementación el modelo propuesto, y en la que se pudieran cumplir con todas las especificaciones que éste tuviera. De las existentes, la plataforma escogida fue J2EE, dado que ofrece APIs para construir consumidores y proveedores de servicios web, y APIs para hacer uso de los estándares que hacen parte de la especificación de los Web services. Adicionalmente, J2EE permite el diseño y construcción de aplicaciones empresariales por capas y el uso de patrones de software lo cual facilita la mantenibilidad y reutilización de una aplicación.

Diseño e implementación

Después de realizar las labores de asimilación y análisis de los temas clave, el siguiente paso que se dio dejó como resultado el diseño e implementación del Modelo de Gestión de servicios PKI basado en una arquitectura orientada a servicios.

Lo primero que se hizo, fue una distinción entre los conceptos de arquitectura y modelo. Luego se realizó el diseño de una arquitectura, para finalmente pasar al diseño e implementación del modelo.

Conceptos de arquitectura y modelo

El objetivo de este trabajo hace referencia a un modelo, pero ¿Qué es un modelo? ¿Por qué no se habla de arquitectura? Dado que en la literatura se encuentran varias definiciones de estos dos conceptos, y no existe una última palabra acerca de qué son realmente, era necesario aclarar y tomar una posición ante lo que significaría arquitectura y modelo en este trabajo de grado.

Se define entonces arquitectura como un conjunto de componentes y la relación entre ellos, y modelo, como una implementación o aplicación de una arquitectura.

Dichos significados son los que se tienen en cuenta a lo largo del desarrollo de este trabajo.

Arquitectura preliminar

Antes de llegar a plantear un modelo que acogiera ciertas tecnologías e implementaciones específicas, se propuso tener como base una arquitectura sobre la cual se pudieran diseñar varios modelos.

El propósito de esta arquitectura, es generalizar el diseño de manera tal, que este trabajo no quede ligado a la implementación que se hizo para el Banco de la República, sino que sirva como base a otras organizaciones que deseen adoptar esta iniciativa de ofrecer servicios de seguridad en un esquema orientado a servicios.

La arquitectura que se plantea se muestra en la Figura 1, y representa una organización distribuida de los componentes que debería tener cualquier modelo que se base en ella. En dicha arquitectura se adopta el concepto de componentes y contenedores de los que habla J2EE.

Cada nodo representa una máquina diferente. Un conjunto de componentes conforman el consumidor de servicios o Service Requestor y otro conjunto lo que sería el proveedor de servicios o Service Provider.

Diseño del modelo

Teniendo como base la arquitectura planteada en la Figura 1, descrita anteriormente, se diseñó el modelo en el que se puede evidenciar una adopción particular de tecnologías para la implementación de algunos componentes. Por ejemplo, para la implementación de toda la parte de servicios se optó por hacer uso de los APIs del JWSDP (Java Web Services Developer Pack), específicamente de los APIs JAX-RPC (Java API for XML-based) y JAXR (Java API for XML Registries), que hacen parte de la especificación 1.4 de J2EE; los componentes que en el modelo representan al PKI, son propios del producto comercial "Entrust Authority".

Este modelo no está ligado a algún PKI en particular porque el Entrust Authority PKI soporta estándares, y por lo tanto, cualquier PKI que les de soporte, puede tenerse en cuenta.

La Figura 2 presenta el diseño del modelo el cual está conformado por dos capas: la capa cliente y la capa lógica de negocio.

Aunque JAX-RPC es un API Java, esto no implica que los clientes y los Servicios Web deban ser desplegados en plataformas Java. Con JAX-RPC un cliente Java puede hacer llamadas y acceder métodos de Servicios Web en plataformas que no son Java, y de la misma forma un cliente que no esté en una plataforma Java puede acceder métodos de Servicios Web que estén en una plataforma Java (ver Figura 3).

Mecanismo de auditabilidad

Unos de los objetivos específicos de esta investigación es desarrollar un mecanismo de auditabilidad que permiten registrar en una base de datos los eventos realizados en el modelo de la Figura 2. Lo que se buscaba, era llevar control de las acciones efectuadas por las aplicaciones consumidoras de los servicios Web PKI.

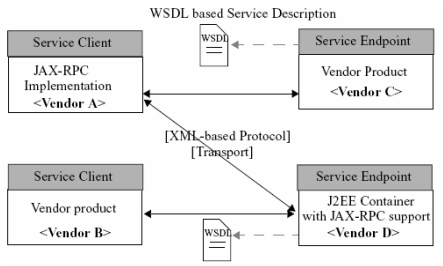

Figura 3: Interoperabilidad ofrecida por JAX-RPC.

Como tal, Entrust Authority no ofrece ninguna utilidad que cumpla con esta misión, por ejemplo, si algún mensaje es firmado digitalmente, no queda ningún detalle ni registro de esta firma; de allí la iniciativa de crear el mecanismo de auditabilidad.

La idea general se plasma en la Figura 4.

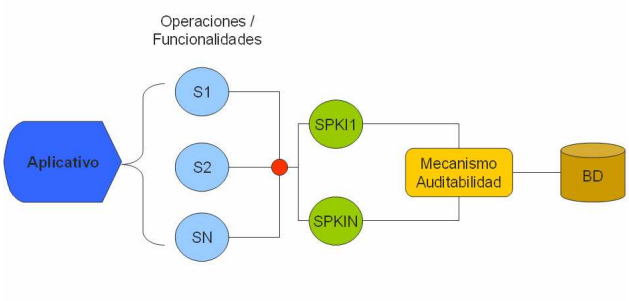

Figura 4: Diseño del mecanismo de auditabilidad

El diseño del mecanismo contempla un aplicativo, que puede ser visto como un conjunto de servicios, los cuales invocan a los Servicios web PKI que son los que hacen uso del mecanismo de auditabilidad, que también fue implementado como un Servicio web llamado RegistroVisor, quien se encarga de ir a una base de datos y almacenar el evento.

Un ejemplo a nivel de implementación, sería un aplicativo de transferencia de fondos en el que uno de sus Servicios web es Consignar, este servicio invocaría al servicio web PKI de Firma Digital para firmar digitalmente la operación señalada. El servicio Web de firma invocaría al servicio Web RegistroVisor (mecanismo de auditabilidad) quien finalmente sería el encargado de almacenar la operación en la base de datos.

Etapa de diseño

En primera instancia se plantea una arquitectura de aplicación basada en J2EE, para luego sobre ella construir e implementar el modelo de los casos de aplicación.

Arquitectura de aplicación

La arquitectura planteada se muestra en la Figura 5 y hace uso de patrones J2EE en su diseño. La idea de esta arquitectura es visualizar el diseño de la aplicación por capas y ofrecer flexibilidad en la distribución y ubicación física de los componentes que la conforman.

Capa de datos

La capa de datos provee los servicios de almacenamiento de datos en un repositorio permanente.

Modelo del caso de aplicación

Una vez planteada y conocida la arquitectura, se procedió a diseñar el modelo del caso de aplicación presentado en la Figura 6.

Este modelo representa la forma como realmente se implementaron los casos de aplicación Transferencia de Fondos Banrep y Visor Banrep, y está basado en la arquitectura de la Figura 5. Muestra la distribución real de los componentes que conforman las aplicaciones, especificando el tipo de tecnologías que se emplean para la comunicación entre ellos.

Conclusiones

Las tecnologías y herramientas para lograr una estrategia PKI vía SOA estaban ahí, pero no existía como tal un diseño e implementación real que permitiera llevarlo a la práctica.

Este trabajo de grado le sirve al banco, sus proveedores y contratistas, así como a cualquier persona u organización que quiera construir aplicaciones seguras y además tomar las ventajas de una arquitectura orientada a servicios.

La adopción e implementación de SOA por medio de Servicios Web otorga innumerables ventajas competitivas a las organizaciones, entre ellas la rápida adaptación al cambio y el alto nivel de respuesta a un mercado variable.

Los Servicios Web son una herramienta tecnológica que permite acelerar el crecimiento de los negocios electrónicos para posibilitar la incorporación de Web services con diferentes funcionalidades, en diversos lenguajes de programación y sistemas operativos.

Los mensajes SOAP escritos en XML lograron unificar las tecnologías existentes mediante Internet, el canal de comunicación masivo y de cobertura mundial.

Los Servicios Web le proporcionan los mecanismos de comunicación e interacción necesarios entre diferentes aplicaciones que interactúan entre sí para presentar información dinámica al usuario, así como proporcionar interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas.

Llevar un control de acciones entre las aplicaciones y los servicios del sistema PKI a través de un mecanismo de auditabilidad, permite a las organizaciones mitigar los incidentes informáticos y aplicar controles más eficientes.

El diseño orientado a servicios implementado por medio de Servicios Web que da oportunidad de poner a disposición servicios del PKI, permite usar las funcionalidades de una PKI de forma sencilla.

El aplicativo de verificación de firmas nace de la necesidad actual de seguridad, permitiendo verificar la legitimidad de una operación sin importar el tiempo transcurrido.

El diseño e implementación de las arquitecturas y modelos propuestos basados en estándares dan la ventaja de interoperabilidad e integración entre aplicaciones.

Este trabajo denominado "Modelo de gestión de servicios PKI basado en una arquitectura orientada a servicios" fue dirigido por el ingeniero Juan Carlos Huertas Amaya, fallecido hace unos meses en Bogotá. Lo presentaron los estudiantes Diana Carolina Valbuena P. y Edgar Hernán López C., de la Pontificia Universidad Javeriana.

Referencias bibliográficas

[1]. Sun Microsystems Inc.4150 Network Circle, Catálogo de Patrones de Diseño J2EE. Y II: Capas de Negocio y de Integración.
http://www.programacion.com/java/tutorial/patrones2/1/

[2]. 2001-2002 Sun Microsystems, Inc., Core J2EE Pattern Catalog, Core J2EE Patterns - Data Access Object.
http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html

[3]. Sun Microsystems Inc.4150 Network Circle, Catálogo de Patrones de Diseño J2EE. I.- Capa de Presentación, 1996-2006.
http://www.programacion.com/java/tutorial/patrones/4/

[4]. Oracle Application Server 10g J2EE and Web Services An Oracle White Paper August 2005 Pag. 4.

[5]. The J2EE™ 1.4 Tutorial For Sun Java System Application Server Platform Edition 8.2 Eric Armstrong.

[6] J2EE Technology in Practice, by Rick Cattell and Jim Inscore (Addison-Wesley, 2001): http://java.sun.com/j2ee/inpractice/aboutthebook.html

[7]. Security Toolkit for the Java® Platform, Programmer's Guide.

[8]. Programmer's Reference, Detailed API reference information, Javadoc reference material.

[9] Maria Gabriela Farias Terrens, Andrea Infante Salgado. Modelo de seguridad para aplicaciones en canales públicos orientado a Web Services. Julio de 2005.

[10] Heather Kreger, IBM Software Group. Web Services Conceptual Architecture (WSCA 1.0). Mayo 2001. IBM Software Group

[11] James McGovern, Sameer Tyagi, Michael Stevens and Sunil Matthew. Java Web Services Architecture. 2003. Morgan Kaufmann Publishers.

[12] Ramesh Nagappan, Robert Skoczylas, Rima Patel Sriganesh. Developing Java Web Services Architecting and Developing Secure Web Services Using Java. 2003 Wiley Publishing Inc., Indianapolis, Indiana. Pag 23-617.

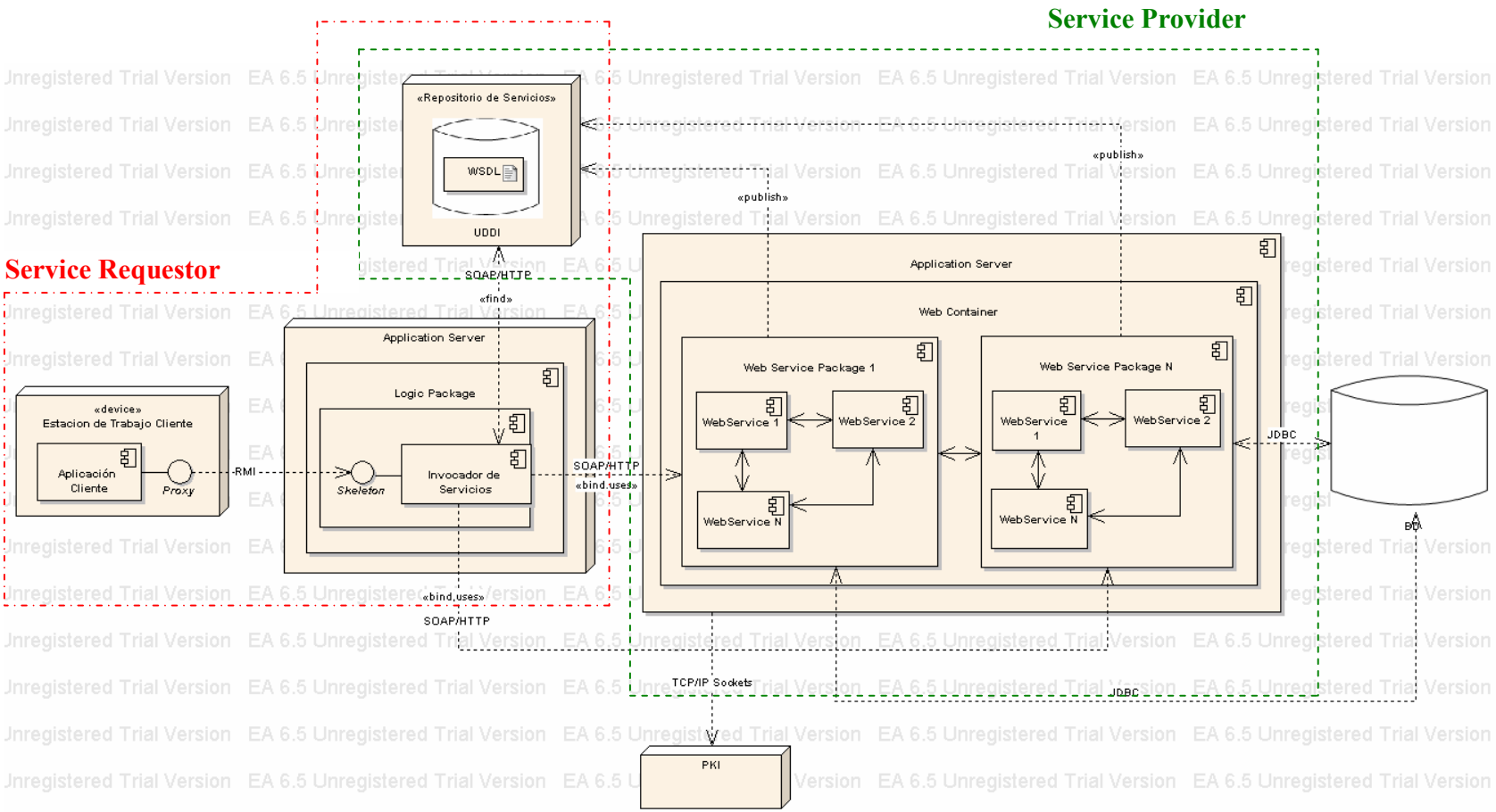

Figura 1. Arquitectura preliminar.

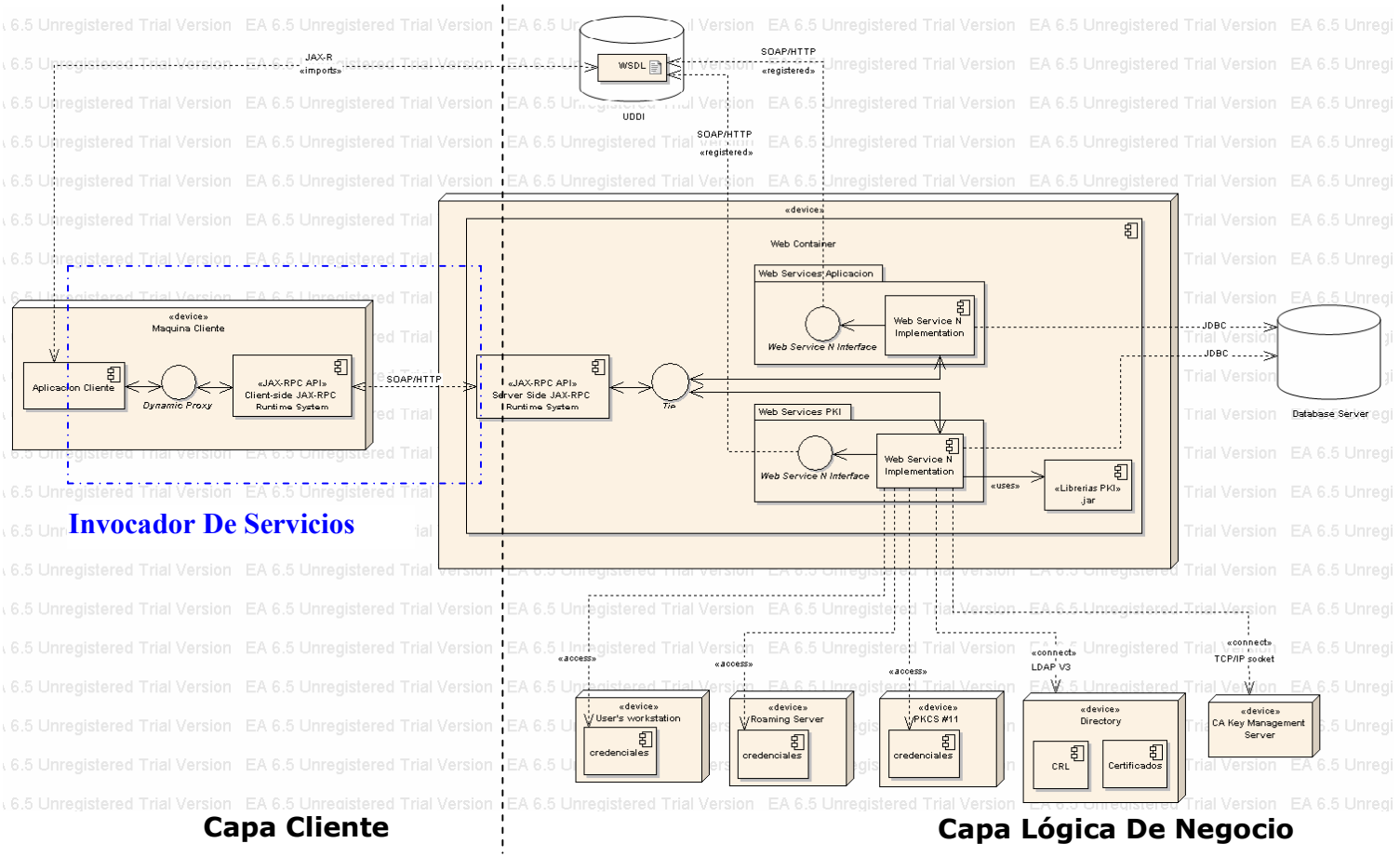

Figura 2. Modelo Propuesto.

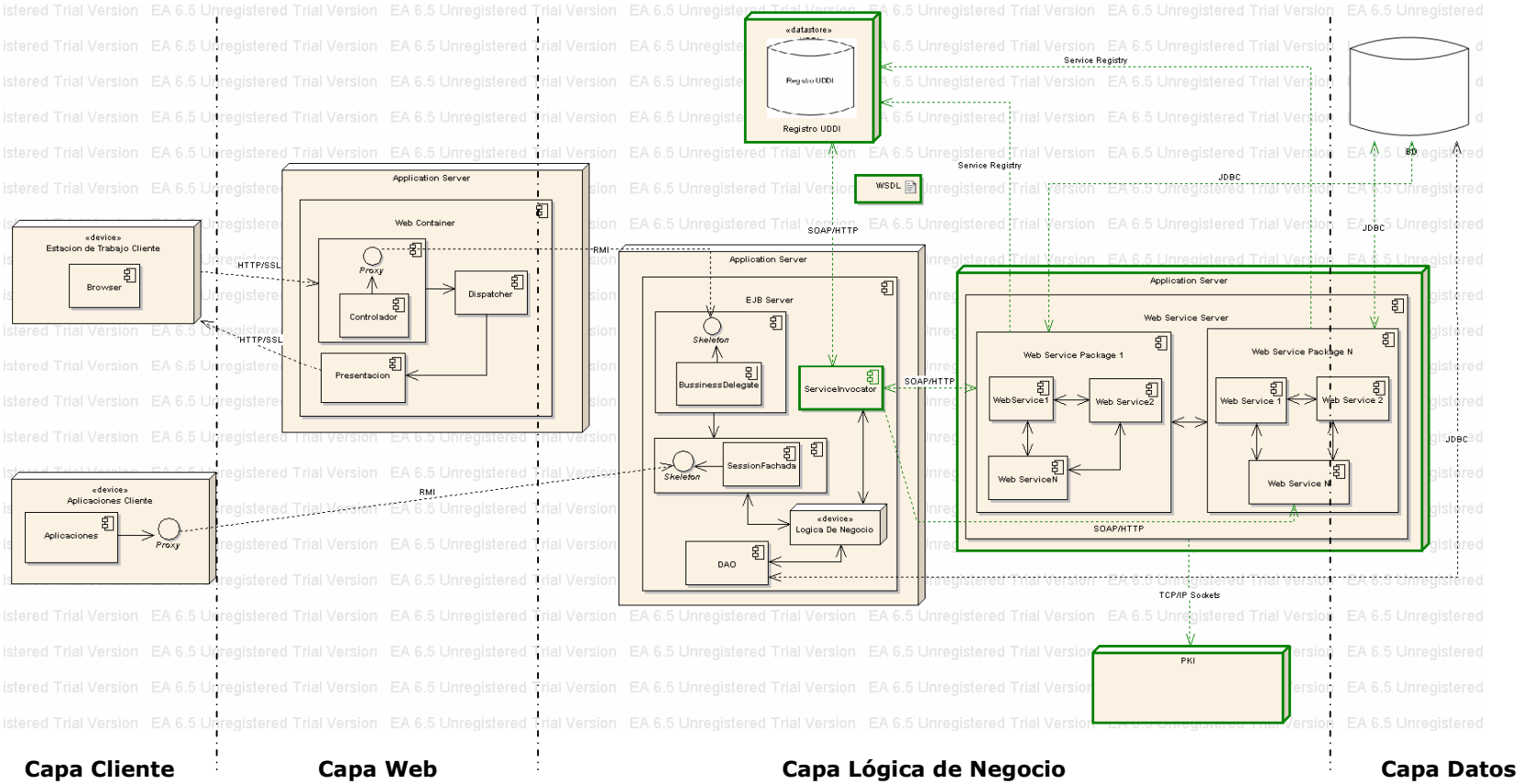

Figura 5. Arquitectura del caso de aplicación.[1]

[1] Los componentes que constituyen el modelo orientado a servicios se representan en color verde.

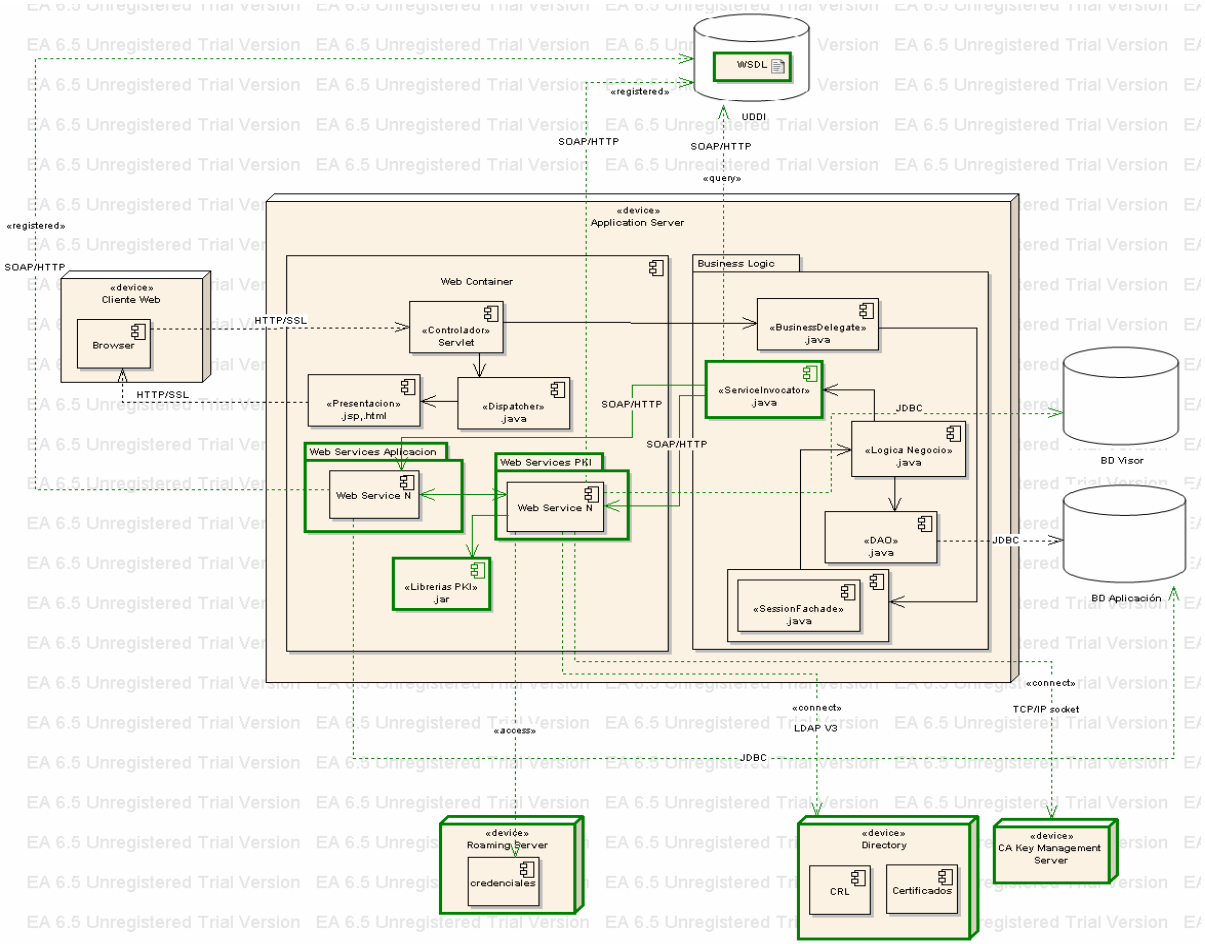

Figura 6. Modelo del caso de aplicación2.

[2] Se representan en color verde los componentes que constituyen el modelo orientado a servicios.