

Anexo 1

Documento de Arquitectura de Software

Integración GIS - PGE

Luciana Canales
Alejandro Remiro
Maximiliano Felix

Contenido:

2	INTRODUCCIÓN.....	5
3	VISTA DE CASOS DE USO	7
3.1	DIAGRAMA DE CASOS DE USO CRÍTICOS	7
3.2	ACTORES	7
3.3	ESPECIFICACIÓN DE CASOS DE USO CRÍTICOS	8
4	VISTA LÓGICA.....	11
4.1	ESTILO ARQUITECTÓNICO	11
4.2	SUBSISTEMAS	12
4.3	DIAGRAMAS DE INTERACCIÓN.....	13
5	VISTA DE DISTRIBUCIÓN (DEPLOYMENT).....	15
5.1	ESCENARIO ORGANISMO CLIENTE	15
5.2	ESCENARIO PÚBLICO GENERAL	16
6	VISTA DE IMPLEMENTACIÓN.....	19
7	REFERENCIAS	21

2 Introducción

El sistema objetivo es una especie de intérprete bidireccional que hará la traducción de mensajes entre sistemas cliente de información geográfica, la Plataforma de Gobierno Electrónico Uruguayo [1] y los servidores de mapas que soportan los estándares de comunicación geográfica del OGC [2]: WMS y WFS.

Uno de los puntos principales a atacar es la traducción y enriquecimiento de mensajes. Ya que la PGE exige que los servicios web expuestos a través de sí deben utilizar el estándar SOAP; y los protocolos WMS y WFS son del estilo Rest[3] se debe realizar en los puntos de entrada y salida de los sistemas la traducción entre los mismos.

Así mismo existen requerimientos de seguridad para la comunicación con la PGE, tanto en el uso de estándares con WS-Security, SAML y Security Tokens que no son soportados por los protocolos mencionados y deben estar cubiertos por el sistema para el éxito de la integración.

El diseño debe contemplar las diferentes realidades en las que se encuentran los clientes, unos accediendo directamente desde redes inseguras como internet y otros ubicados físicamente dentro de organismos estatales o privados en redes con mayor control. En todos los casos la información debe estar accesible aunque varía ampliamente el tipo de información expuesta según el origen de la comunicación y el usuario que esté tratando de acceder a la misma.

3 Vista de Casos de Uso

3.1 Diagrama de Casos de Uso Críticos

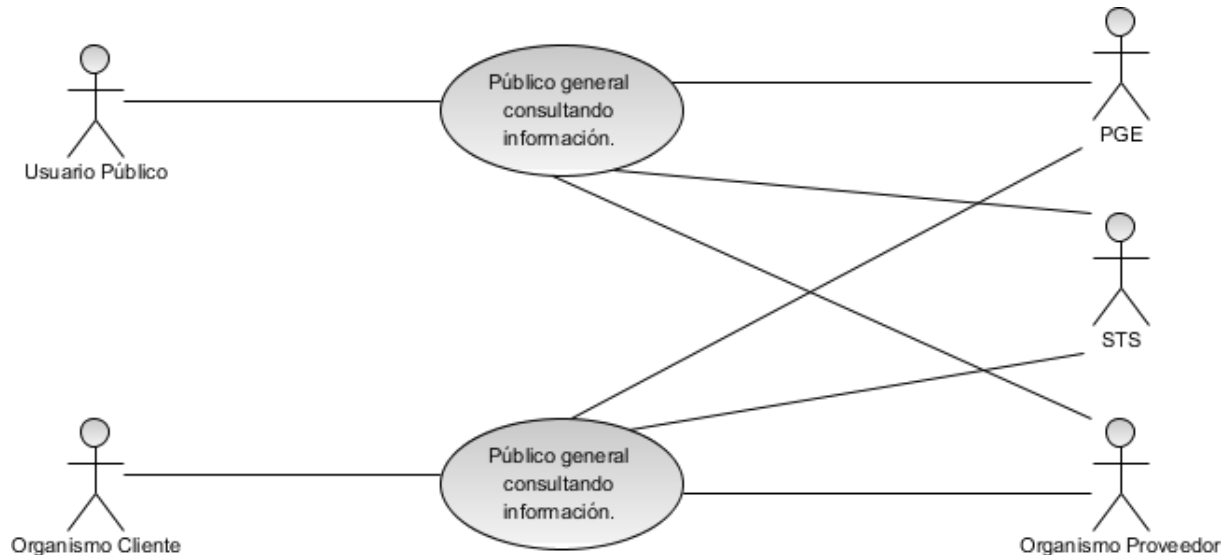


Diagrama de Casos de usos críticos

3.2 Actores

Organismo

Es una entidad que contiene datos geográficos y ofrece servicios tanto de consulta como de actualización. A su vez un organismo puede consultar o actualizar datos geográficos de otro organismo.

Público general

Es una persona con un perfil público que quiere actualizar o consultar información geográfica.

STS

Encargado de la autenticación. Provee tokens de seguridad que luego la PGE se encarga de verificar.

PGE

Es el componente que se encarga de procesar todos los pedidos y controlar que sean correctos así como también maneja los mecanismos de seguridad y de balanceo de carga.

3.3 Especificación de Casos de Uso Críticos

3.3.1 Público general consultando información

Descripción

Permite a un usuario público consultar información geográfica que ofrece un organismo.

Pre-condiciones

El organismo al que se quiere consultar está ofreciendo un servicio público para consultar información.

Flujo de Eventos

Flujo principal

- 1.1 Un usuario solicita consultar información geográfica
- 1.2 CtpRestController recibe la petición, solicita token al STS
- 1.3 El STS recibe la petición, genera el token y lo devuelve a CtpRestController
- 1.4 CtpRestController arma un pedido con el token, método y parámetros a invocar y lo envía a la PGE
- 1.5 La PGE recibe el pedido, chequea que los datos sean correctos y lo envía a CtpSoapConnector
- 1.6 CtpSoapConnector recibe el pedido y lo envía a al servidor de mapas correspondiente
- 1.7 El servidor de mapas recibe el pedido y devuelve la información geográfica correspondiente
- 1.8 CtpSoapConnector recibe la información y la devuelve a la PGE
- 1.9 La PGE recibe la información y la devuelve a CtpRestController
- 1.10 CtpRestController devuelve la información al usuario

Flujo alternativo 1.3.1 (STS falla)

- 1.3.1 El STS no valida al usuario y devuelve un error
- 1.3.2 CtpRestController envía un mensaje de error al usuario

Flujo alternativo 1.5.1 (PGE falla)

- 1.5.1 Los datos enviados a la PGE no son válidos y ésta devuelve un error
- 1.5.2 CtpRestController envía un mensaje de error al usuario

Flujo alternativo 1.7.1 (Servidor de mapas falla)

- 1.7.1 El servidor de mapas devuelve un error
- 1.7.2 CtpSoapConnector recibe el error y lo reenvía a la PGE
- 1.7.3 La PGE reenvía el error a CtpRestController
- 1.7.2 CtpRestController envía un mensaje de error al usuario

Post-condiciones

No hay cambios en el sistema.

La información solicitada es devuelta al usuario.

3.3.2 Instituciones colaborando en trámites

Descripción

Permite que un organismo solicite información de otro organismo para realizar trámites.

Pre-condiciones

El organismo origen tiene los permisos para acceder al servicio ofrecido.

Flujo de Eventos

Flujo Principal

- 4.1 Un organismo solicita consultar información geográfica de otro organismo
- 4.2 CtpRestConnector recibe la petición, solicita token al STS
- 4.3 El STS recibe la petición, genera el token y lo devuelve a CtpRestConnector
- 4.4 CtpRestConnector arma un pedido con el token, método y parámetros a invocar y lo envía a la PGE
- 4.5 La PGE recibe el pedido, chequea que los datos sean correctos y lo envía a CtpSoapConnector
- 4.6 CtpSoapConnector recibe el pedido y lo envía a al servidor de mapas correspondiente al organismo destino
- 4.7 El servidor de mapas recibe el pedido y devuelve la información geográfica correspondiente
- 4.8 CtpSoapConnector recibe la información y la devuelve a la PGE
- 4.9 La PGE recibe la información y la devuelve a CtpRestConnector
- 4.10 CtpRestConnector devuelve la información al organismo solicitante

Flujo alternativo 2.3.1 (STS falla)

- 4.3.1 El STS no valida el pedido de autenticación y devuelve un error
- 4.3.2 CtpRestConnector envía un mensaje de error al organismo

Flujo alternativo 2.5.1 (PGE falla)

- 4.5.1 Los datos enviados a la PGE no son válidos y ésta devuelve un error.
- 4.5.2 CtpRestConnector envía un mensaje de error al organismo.

Flujo alternativo 2.7.1 (Servidor de mapas falla)

- 4.7.1 El servidor de mapas devuelve un error
- 4.7.2 CtpSoapConnector recibe el error y lo reenvía a la PGE
- 4.7.3 La PGE reenvía el error a CtpRestConnector
- 4.7.2 CtpRestConnector envía un mensaje de error al organismo

Post-condiciones

La información solicitada es devuelta al organismo solicitante para que continúe con el trámite.
No hay cambios en el sistema.

4 Vista Lógica

Tomando en cuenta la solución propuesta por Raquel Sosa en su tesis de maestría [4] se tiene un conjunto de 5 subsistemas independientes que interactúan para resolver los casos de uso. Con respecto al Diagrama de Arquitectura, de izquierda a derecha se tiene: El cliente GIS, el CTP de entrada llamado RestConnector, la PGE propiamente dicha, el CTP de salida llamado SoapConnector y el servidor de mapas que se quiere enmascarar.

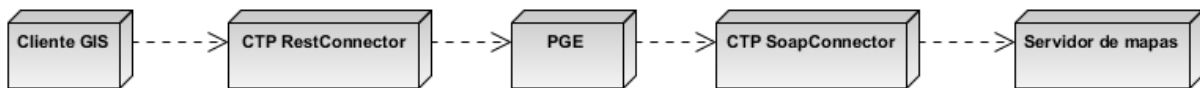


Diagrama de Arquitectura

Cliente GIS: Cualquier programa utilizado para consumir datos geográficos utilizando protocolos WMS y WFS. Puede ser desde una aplicación web usando OpenLayers hasta Gvsig.

CTP RestConnector: es un sub sistema encargado de recibir pedidos REST especificados según los protocolos WMS y WFS. Transforma estos pedidos a formato SOAP para que atraviesen la PGE. Y los encamina hacia ella. También según la configuración agregará información a los pedidos que requieran seguridad y otros datos exigidos por la PGE.

PGE: Es la plataforma de gobierno electrónico del estado uruguayo [1].

CTP SoapConnector: Este sistema estará registrado en la PGE como un proveedor de servicios, al cual se encaminarán los pedidos WMS y WFS previamente transformados por el CTP RestConnector. Su función es volver a componer el pedido REST para enviarlo al servidor de mapas configurado, y transformar la respuesta del mismo en un mensaje SOAP para que recorra el camino inverso. La idea es que exista un SoapConnector por cada servicio GIS expuesto a través de la PGE.

Servidor de mapas: Es un servidor que soporte los protocolos WMS y WFS. En principio habrá soporte para GeoServer [5] y MapServer [6].

4.1 Estilo Arquitectónico

En esta sección se comentará el estilo arquitectónico que respecta a los componentes CTP, para los demás sistemas referirse a la documentación específica de cada uno.

Ambos CTP están implementados como una orquestación de servicios dentro de un ESB. Por tanto respetan la arquitectura provista por los ESB que es en definitiva la arquitectura de los sistemas de mensajería.

Los sistemas de mensajería utilizan una arquitectura de pipeline de filtros, donde cada filtro afecta el mensaje que se transmite al siguiente filtro o bien hace trabajo de ruteo cambiando el orden o los filtros que se ejecutan sobre un mensaje particular.

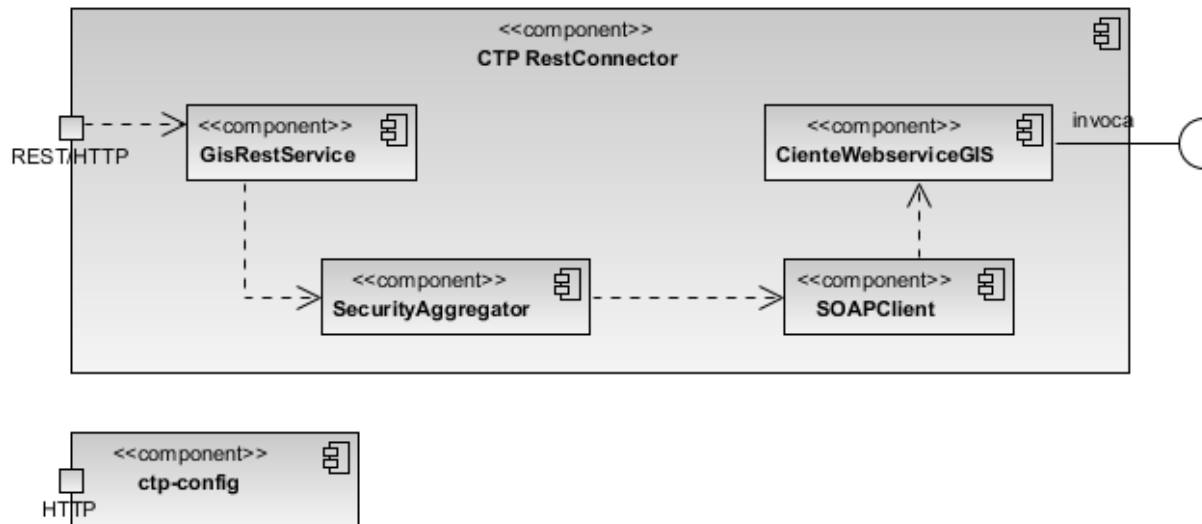
Los CTP están implementados utilizando la tecnología JBOSS ESB, la cual provee una interfaz de filtro que se implementa, y luego se define un servicio donde se declaran los filtros que utiliza y en qué orden se ejecutan.

4.2 Subsistemas

En esta sección se describe cada uno de los subsistemas. Debido a que son el objeto de este proyecto se describen los CTPs y como se utiliza una versión reducida y minimalista de la PGE implementada específicamente para simular la PGE real en el contexto de este proyecto también se incluirán sus detalles en el siguiente apartado.

4.2.1.1 CTP RestConnector

Diagrama de Componentes.



Descripción

El RestConnector está compuesto de tres filtros con responsabilidades bien separadas que van agregando información al mensaje ESB para finalmente transformar el pedido de REST en un mensaje SOAP y un cliente de web service SOAP generado en algún framework de web services[7] que lo envía hacia la PGE.

El punto de entrada es un Getway HTTP cuya implementación es provista por la plataforma ESB. Este recibe las conexiones HTTP en una URI particular del servidor y lo transforma en un mensaje ESB que se transmite al primer filtro y se va pasando entre estos procesándolo como sea debido.

El primer filtro es el GisRestService, este se encarga de parsear el pedido WMS o WFS para averiguar a qué servicio geográfico expuesto a través de la PGE se quiere invocar. Para esto toma datos del pedido y consulta una base de conocimiento. Carga toda la información necesaria y delega al siguiente filtro.

El SecurityAggregator es el encargado de resolver las necesidades de autenticación y autorización que el servicio de la PGE requiera. Con la información proporcionada por el filtro anterior conformará el token de seguridad SAML que se requiere para atravesar la PGE.

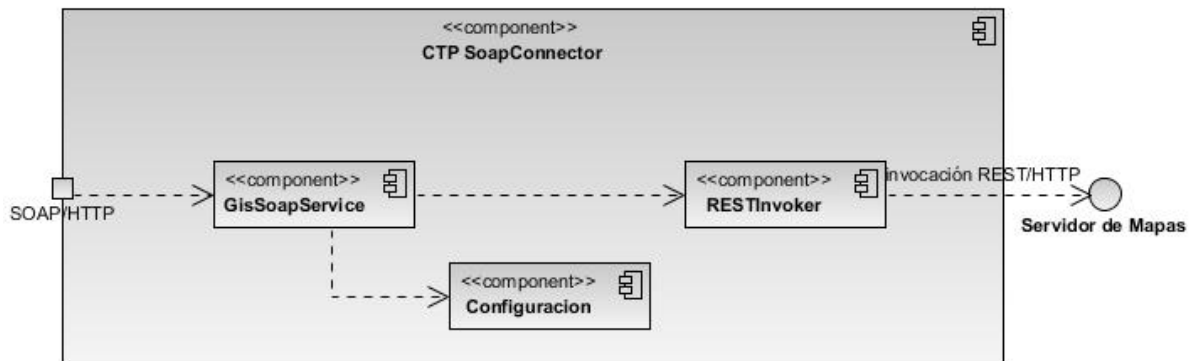
Finalmente el SOAPClient utiliza un cliente de webservice generado a partir del webservice SOAP que provee el CTP SOAPConnector, para invocar el servicio geográfico con los datos necesarios incluyendo los parámetros del pedido original y los agregados por la PGE. Con la diferencia de que el endpoint invocado no es el propio SoapConnector sino el proxy del servicio expuesto por la PGE.

Al recibir la respuesta se coloca en el mensaje ESB lo enviado por el proveedor y el servidor ESB se encarga automáticamente de convertir eso en una respuesta HTTP adecuada, ya que se ha terminado de invocar los filtros configurados para ese servicio.

Por otro lado se proyecta una aplicación web llamada ctp-config que será implementada en Grails. La funcionalidad de esta aplicación es brindar una interfaz de usuario agradable para configurar los diferentes servicios que pueden ser accedidos desde esa instancia del CTP y la seguridad que cada uno requiera. La información se guarda en una base de datos que será accedida tanto por el CTP como por la aplicación de configuración.

4.2.1.2 CTP SoapConnector

Diagrama de Componentes



Descripción

El SoapConnector también está implementado en la tecnología ESB pero es bastante diferente al anterior. En este caso se configura un servicio cuyo punto de acceso en un procesador SOAP provisto por el ESB, el mismo recibe pedidos HTTP que contengan un mensaje SOAP, lo procesa e invoca a la clase que lo implementa, en este caso el componente GisSoapService, el cual tiene declarados como métodos todos los definidos por los estándares WMS y WFS. Este componente recibe toda la información y utiliza la configuración para armar la URL del servidor de mapas configurado. Dado que se utilizará una instancia de este sub sistema por cada servicio geográfico, es decir por cada servidor de mapas que se quiera exponer, solo se configura un servidor y su URL, IP, puerto y la URI. Esta URI puede variar según el protocolo y según la implementación del servidor, por ejemplo existen variantes entre GeoServer y MapServer.

El componente RestInvoker es el encargado de armar el pedido HTTP correspondiente, basado en los parámetros del pedido original y la URL provista por GisSoapService.

Para la respuesta, simplemente se retorna lo enviado por el servidor de mapas, los componentes provistos por ESB que implementan el web service se encargan de convertirlo en un mensaje SOAP Response. Solo se debe tener en cuenta que para el método GetMap la respuesta es una imagen en formato binario y no un XML como en todos los demás métodos.

4.3 Diagramas de Interacción

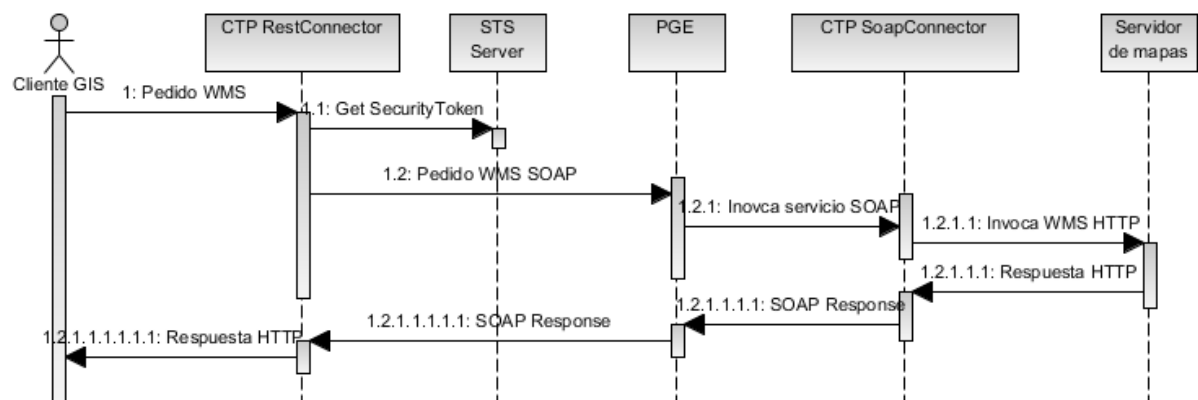


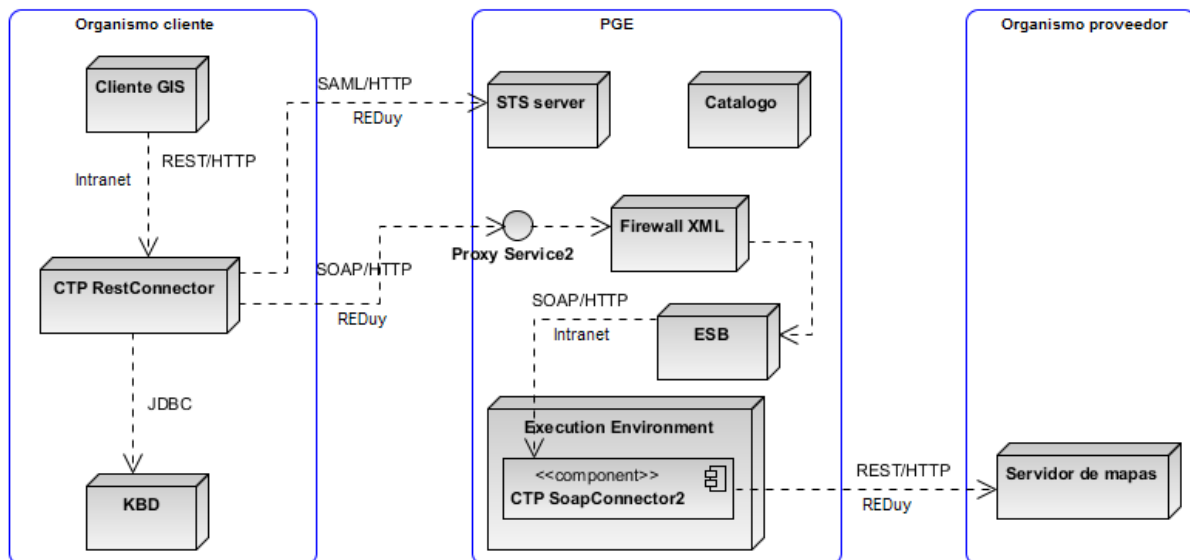
Diagrama de Secuencia para caso de uso Público General

Para el resto de los casos de uso, la interacción entre los sistemas y componentes es igual, lo que diferencia a cada caso de uso es el despliegue de los subsistemas que cambia según el entorno en que se encuentra el cliente.

5 Vista de Distribución (Deployment)

Uno de los puntos interesantes en el diseño fue encontrar la manera de desplegar los diferentes componentes teniendo en cuenta los requerimientos planteados en los escenarios de uso de la tesis de Raquel Sosa, así como también la interacción con los sistemas preexistentes, es decir tanto la PGE, como la REDuy[8] y los servicios e infraestructura con los que hoy cuentan los diferentes organismos estatales que utilizan a diario información geográfica.

5.1 Escenario Organismo cliente



Despliegue del escenario Organismo cliente.

Descripción

Uno de los escenarios más comunes y oficiales será cuando por algún proceso administrativo de alguna índole, puede ser apertura de un expediente, control de recursos naturales, etc. Un organismo estatal con acceso a la REDuy [8], quiere utilizar servicios geográficos de uno o más organismos estatales, cuyos servicios están expuestos a través de la PGE.

En este caso el organismo cliente tendrá levantado una instancia del CTP RestConnector, al cual se le configurará los servicios geográficos que puede acceder. De esta manera se obtienen ciertos efectos interesantes:

1. El CTP es interno al organismo y solo se accederá desde la intranet del mismo, lo que minimiza los requerimientos de seguridad para accederlo.
2. Solo se configuran los servicios que el organismo, o las diferentes áreas del mismo utilizan, reduciendo la tereas de mantenimiento de configuración a lo estrictamente necesario, y sacando esta responsabilidad de los encargados de la PGE.
3. La información sensible de autenticación del organismo no sale de sus fronteras.

Otro aspecto importante de esta arquitectura es que el CTP SoapConnector, como se mencionó anteriormente, invoca solo un servicio geográfico por instancia, y utilizando los Execution Environment [9] provistos por la PGE, se puede aprovechar al máximo todo el poder de procesamiento y tolerancia a fallas de la infraestructura de la misma y debido a que el mantenimiento es mínimo y simple, las responsabilidades del personal encargado de la PGE no se ven demasiado afectados, incluso según la política elegida, el mantenimiento de esos EE puede ser delegado a los organismos proveedores que los soliciten.

Nodos

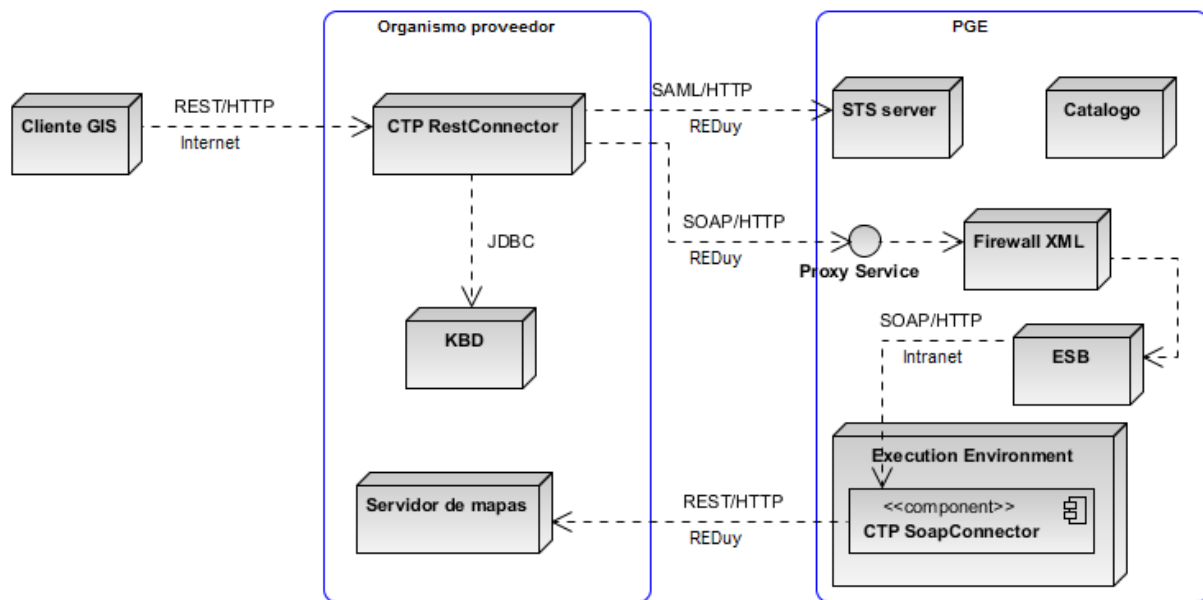
CTP RestConnector y su base de conocimiento: estarán dentro de las fronteras del organismo cliente. Al cual se le configuran solo los servicios geográficos que se usarán desde ese organismo.

Execution Environment: Esta infraestructura provista por PGE para hostear servicios, sirve para albergar las instancias del CTP SoapConnector que invocarán los servicios geográficos. Esto permite a la PGE manejar todos los aspectos, desde seguridad hasta balanceo de carga.

Conexiones

Todas las conexiones se realizan en redes controladas, por un lado los clientes geográficos harán sus invocaciones en una intranet, mientras que las conexiones entre los organismos y la PGE serán a través de REDuy.

5.2 Escenario Público general



Despliegue del escenario Público general

Descripción

El segundo escenario surge por la necesidad de brindar servicios geográficos, protegidos por la PGE a través de internet. En este caso el cliente estará en cualquier parte del mundo conectado a internet y debe ser capaz de acceder a información geográfica, no sensible y sin capacidades de modificarla, por tanto no requiere configuración de seguridad por IP.

Exponer servicios estatales, cualquiera sean ellos, siempre los deja expuestos a ataques, también a sobrecarga, en este contexto los CTP RestConnector siempre son un punto vulnerable y no pueden estar dentro de la PGE pues el acceso a la misma es lo que tratan de resolver. Por otro lado mantener un cluster de estos CTP en otra infraestructura ajena a la PGE y los organismos, también es posible pero muy costosa, en cuanto a recursos informáticos y administrativos. Sin embargo, si se toma la decisión, de que el organismo proveedor sea el encargado de mantener un CTP RestConnector disponible en internet, para los servicios geográficos propios, es importante que estos CTP solo puedan acceder a los servicios GIS del organismo.

Este segundo enfoque tiene grandes ventajas:

1. Los puntos vulnerables están distribuidos y en caso de sobrecarga o ataque solo afectan a un servicio determinado y no a todos los servicios GIS disponibles.

2. La configuración en los CTP es mínima.
3. Si un CTP es atacado y queda fuera de servicio, no se afecta al servidor de mapas ni al CTP SoapConnector que lo accede, con lo cual el servicio sigue estando disponible para los organismos clientes que dependan de él.
4. La responsabilidad de configuración, mantenimiento y disponibilidad del CTP RestConnector queda totalmente del lado del organismo proveedor.

Otro aspecto importante es que los organismos clientes no deben usar los CTP expuestos a internet de los proveedores, sino su CTP propio. Si bien no hay restricciones tecnológicas al respecto, ha y que tener en cuenta que estos CTP solo podrán acceder a información no sensible y no tienen permisos de escritura sobre esos datos tampoco. Mientras que esas restricciones pueden no existir si se acuerda entre los organismos las pautas de seguridad adecuadas y se configuran en el CTP interno al organismo cliente.

Cabe destacar que este escenario es suplementario al anterior y ambos puede y tal vez deban coexistir para brindar el máximo de disponibilidad de la información y poder cumplir con los escenarios de uso planteados en la tesis de maestría de Raquel Sosa.

Nodos

Igual que el anterior, solo cambia la infraestructura donde están hosteados el CTP RestConnector y el servidor de mapas.

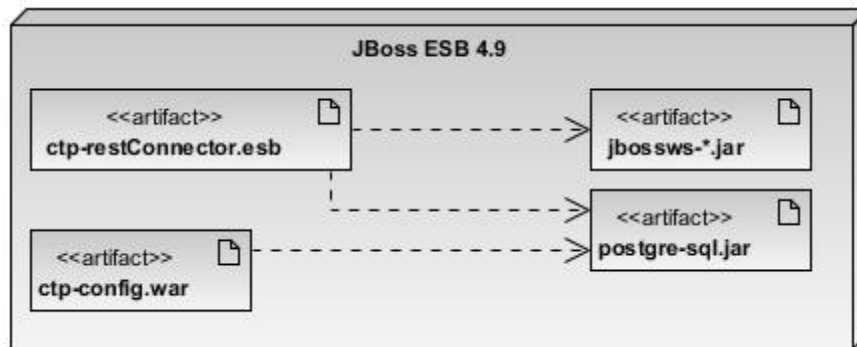
Conexiones

En este caso el gran cambio es que el cliente GIS está en internet, pero el resto de la comunicación entre PGE y organismo se mantiene en la REDuy.

6 Vista de Implementación

CTP RestConnector

El siguiente diagrama muestra los artefactos que implementan este subsistema y las dependencias más importantes.



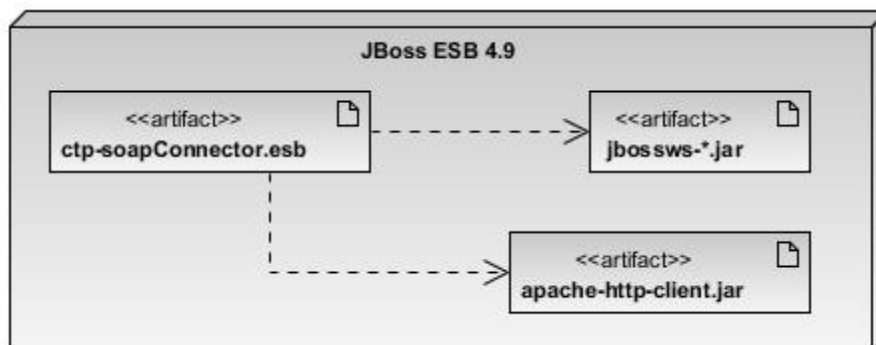
Artefactos de CTP RestConnector

Se utiliza un empaquetado “.esb” para implementar los servicios del CTP RestConnector. Estos archivos son la forma de definir servicios en el JBoss ESB y contienen tanto los archivos de configuración XML como las clases que implementan los filtros de los servicios.

También existe el ctp-config.war que es una aplicación Grails que se utiliza como interfaz para configurar los datos de la base de conocimiento del CTP.

CTP SoapConnector

El diagrama a continuación tiene los artefactos del subsistema.



Artefactos del CTP SoapConnector

Al igual que el CTP anterior este también está empaquetado dentro de un “.esb” y como no lleva base de conocimiento ya que su configuración es por propiedades del sistema, no depende de una base de datos.

7 Referencias

- [1] Plataforma de Gobierno Electrónico de Uruguay
http://agesic.gub.uy/innovaportal/v/1454/1/agesic/guia_de_uso_de_la_plataforma_de_ge_del_estado_uruguayo.html
- [2] Open Geospatial Consortium
<http://www.opengeospatial.org/>
- [3] Representational state transfer, por Roy Fielding en 2000.
http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
- [4] Tesis de maestría de Raquel Sosa, 2011
http://www.fing.edu.uy/~raquels/TesisRaquelSosa_vf_1.2.pdf
- [5] GeoServer
<http://geoserver.org/>
- [6] MapServer
<http://mapserver.org/>
- [7] Framework de cliente SOAP implementación de JBoss
<http://jbossws.jboss.org/>
- [8] REDuy
<http://www.agesic.gub.uy/innovaportal/v/3928/1/agesic/reduy.html>
- [9] Entornos de ejecución de la PGE, página 10, Capítulo 3.
http://agesic.gub.uy/innovaportal/file/1454/1/capitulo_3.pdf
- [10] Architectural Blueprints — The “4+1” View Model of Software Architecture. Kruchten, Philippe. 1995.
<http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf>
- [11] Software Architecture Links. Bredemeyer Consulting.
<http://www.bredemeyer.com/links.htm>