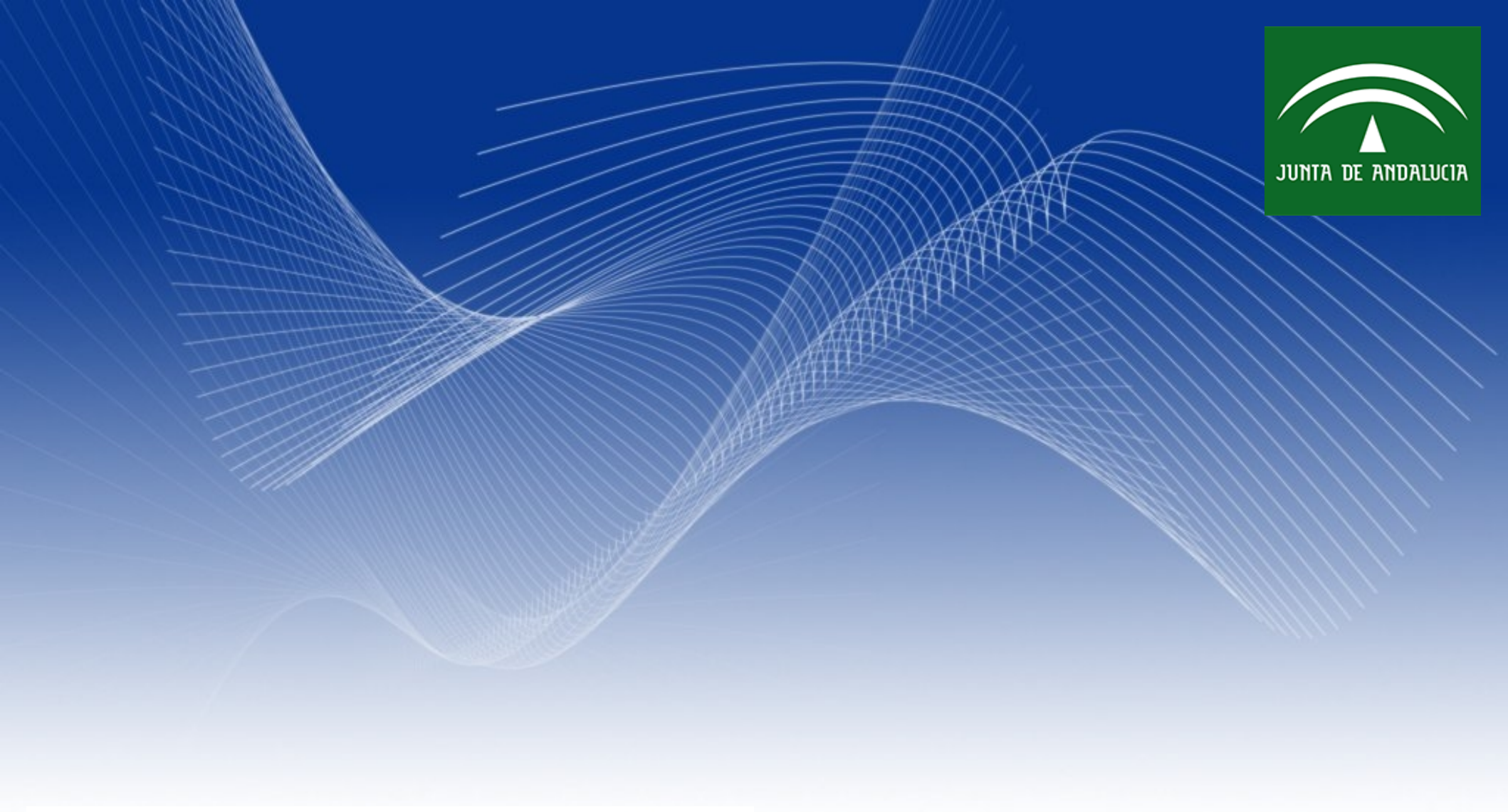


PLATINA

Plataforma de Interoperabilidad de Andalucía

Manuel Toscano
Servicio de Coordinación y Desarrollo de Sistemas Horizontales

Secretaría General de Telecomunicaciones y Sociedad de la Información

ÍNDICE

DIA 5: DESARROLLO SEGÚN PLATINA Y MADEJA

REPASO CONCEPTOS PLATINA. D. Manuel Toscano (9:00 – 9:30)

DESARROLLO SEGÚN PLATINA Y MADEJA.D. Manuel Toscano (9:30 – 11:30)

PRACTICA: DESARROLLO DE SERVICIOS WEB CONFORME A MADEJA Y PLATINA D. Felipe Chica (11:30-13:00)

EVOLUCIÓN FUTURA DE PLATINA D. Jorge Sánchez (13:00-14:30)

REPASO CONCEPTOS PLATINA: Índice

Introducción Platina

- Ley
- Objetivos Platina; SOA EAI,
- Platina como EAI
- Platina como SOA
- Arquitectura y Diseño de Platina.
- Proxy Genérico.

Leyes

- **EIF (European Interoperabilitiy Framework).** An Interoperability Framework can be defined as the overarching set of policies, standards and guidelines which describe the way in which organisations have agreed, or should agree, to do business with each other

- **Ley 11/2007** Las AAPP utilizarán las tecnologías en sus relaciones con las demás administraciones y con los ciudadanos de forma que se garantice la interoperabilidad: **Esquema Nacional de Interoperabilidad, Esquema Nacional de Seguridad**.

- **Marco de Interoperabilidad** Digital de la Junta de Andalucía. Garantizar a la ciudadanía y a las instituciones públicas y privadas la comunicación electrónica y el acceso a los servicios de la Administración Pública Andaluza sin dependencias tecnológicas. Garantizar la Interoperabilidad de los servicios y sistemas de información tanto para la relación con la ciudadanía e instituciones públicas y privadas como en el seno de la Junta de Andalucía

Objetivos

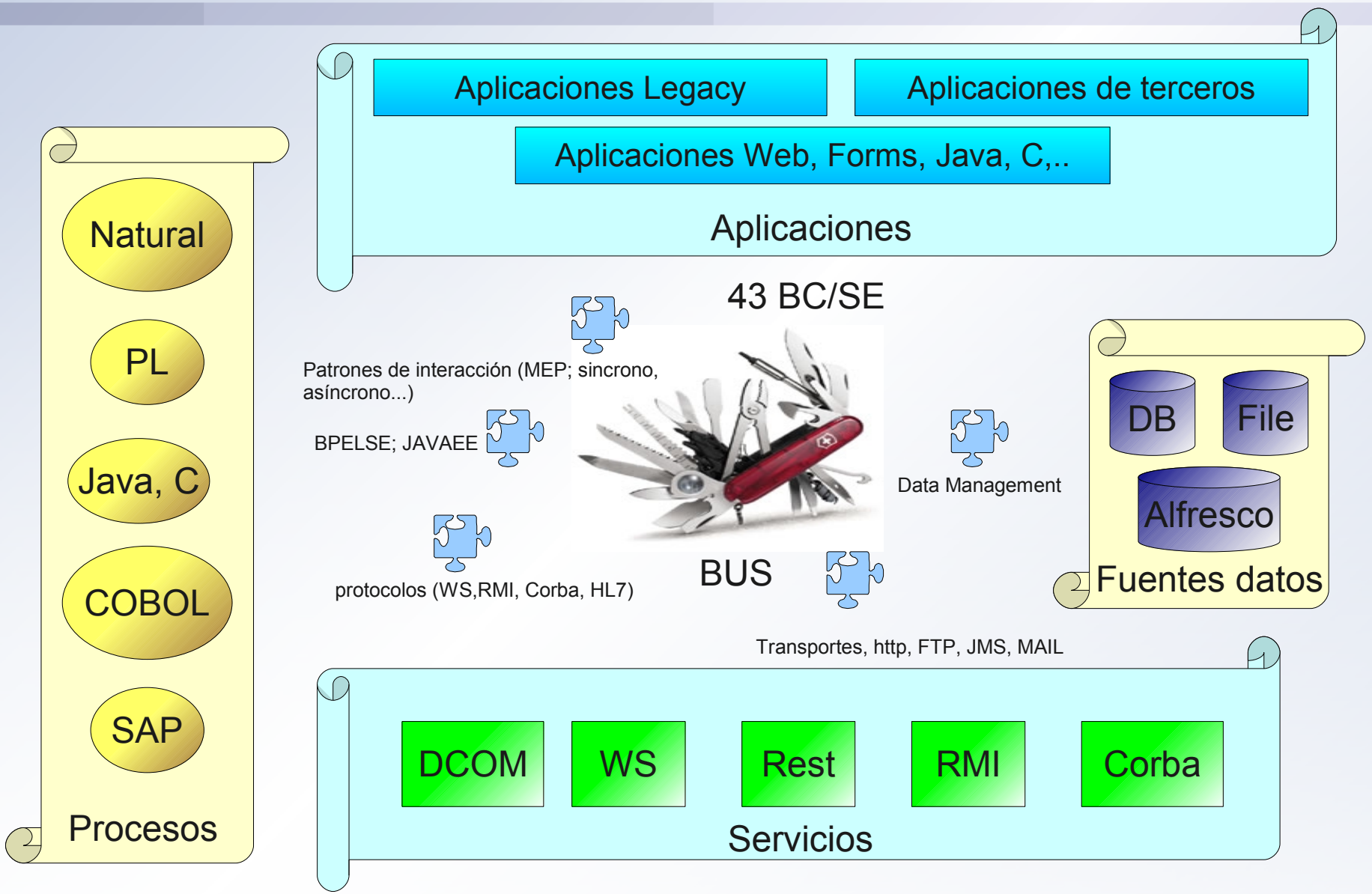
Platina tiene un doble objetivo:

- Proporcionar una **plataforma tecnológica** y un conjunto de **principios** y **normas** que **ayuden a resolver** los problemas de **integración** de **procesos, aplicaciones y datos** dentro de la Junta de Andalucía. En definitiva, este objetivo pretende la creación de un Enterprise Application Integration (EAI).
- El segundo objetivo es la construcción de una **Arquitectura SOA** dentro de la Junta de Andalucía.

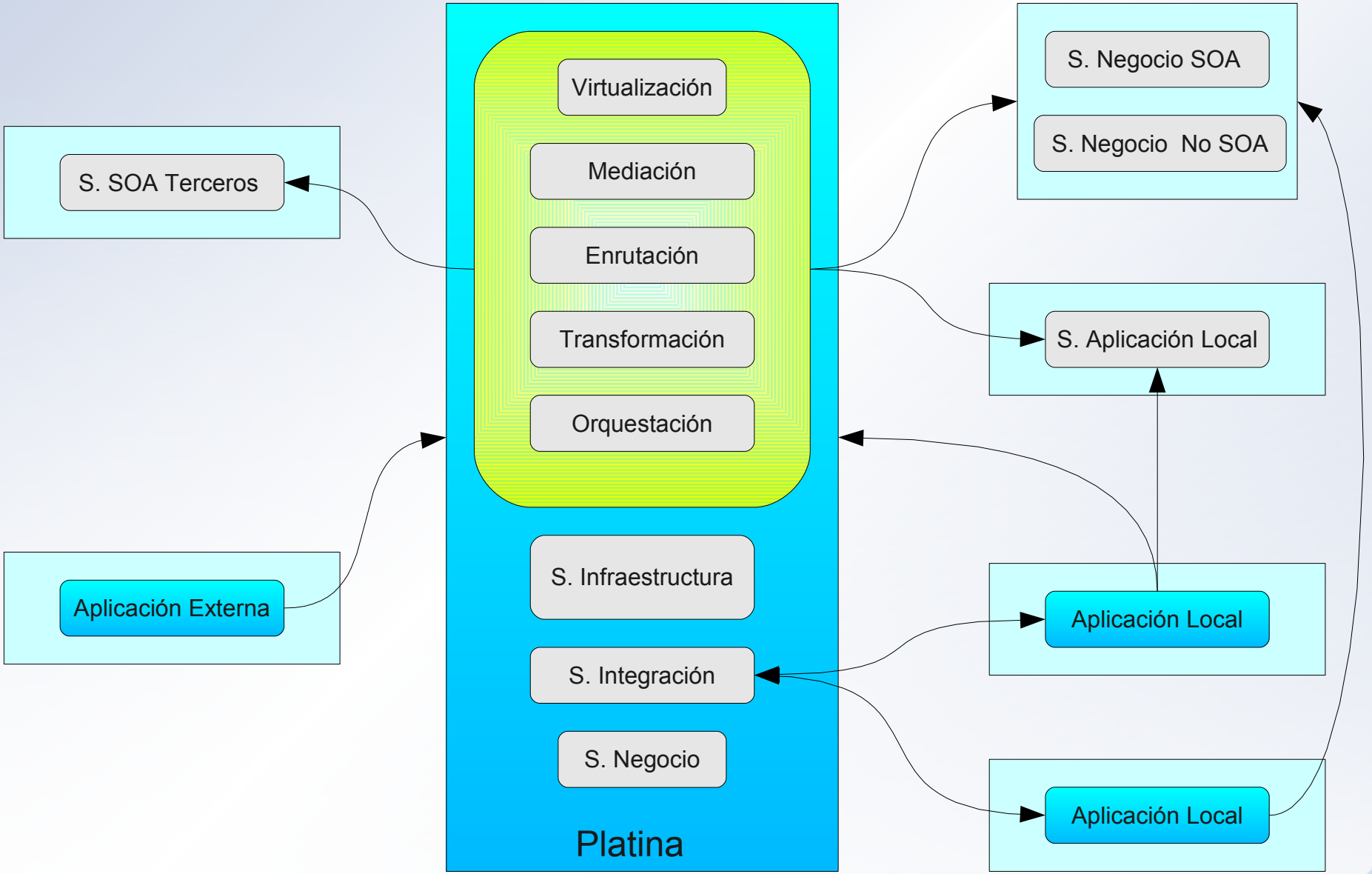
Ambos objetivos se resuelven empleando tecnología de buses de integración.

Las organizaciones podrán emplear platina como EAI, como Plataforma SOA o de ambas formas, en función de sus necesidades.

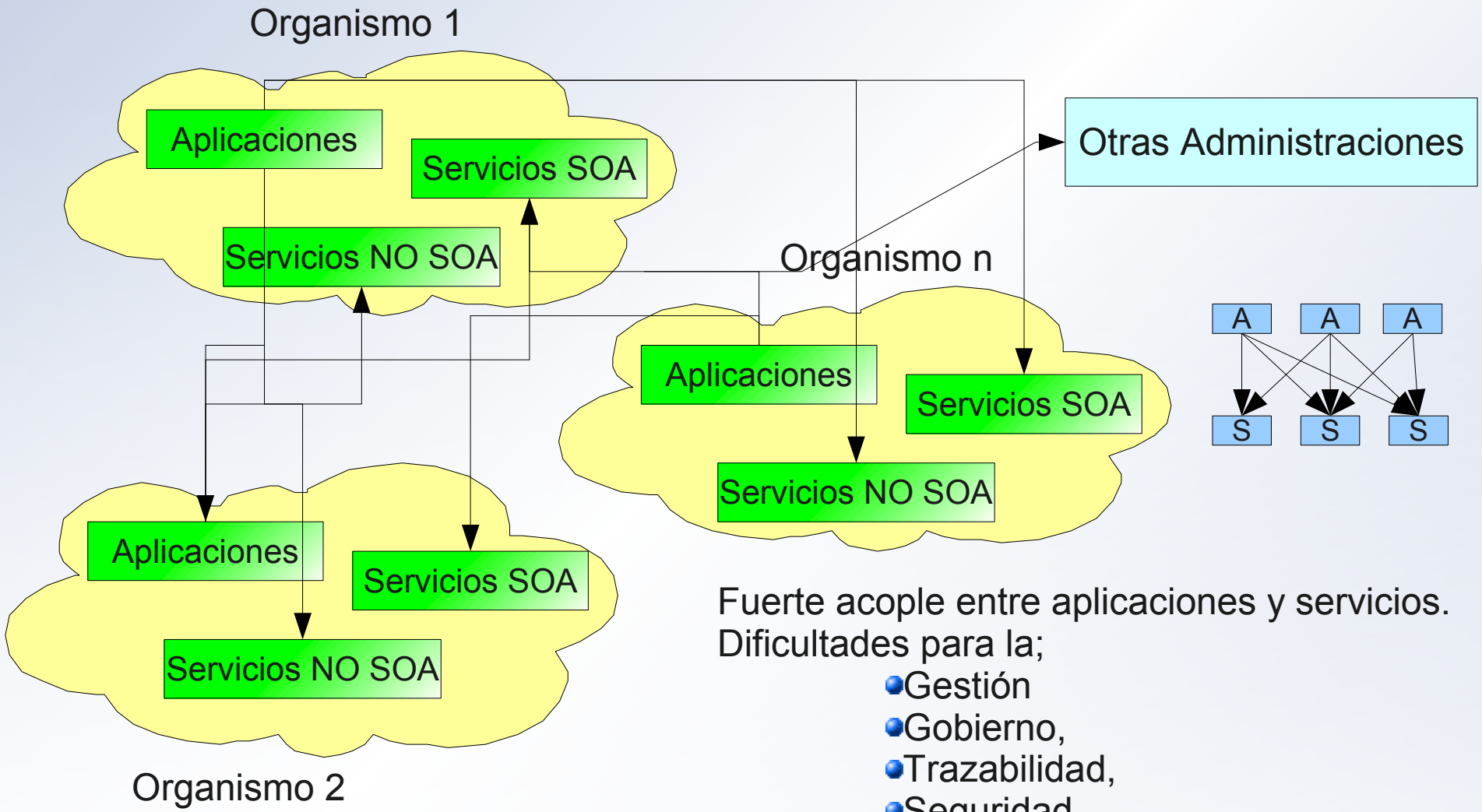
Platina como EAI: Problema y Solución



Platina como EAI: Técnicas de Integración



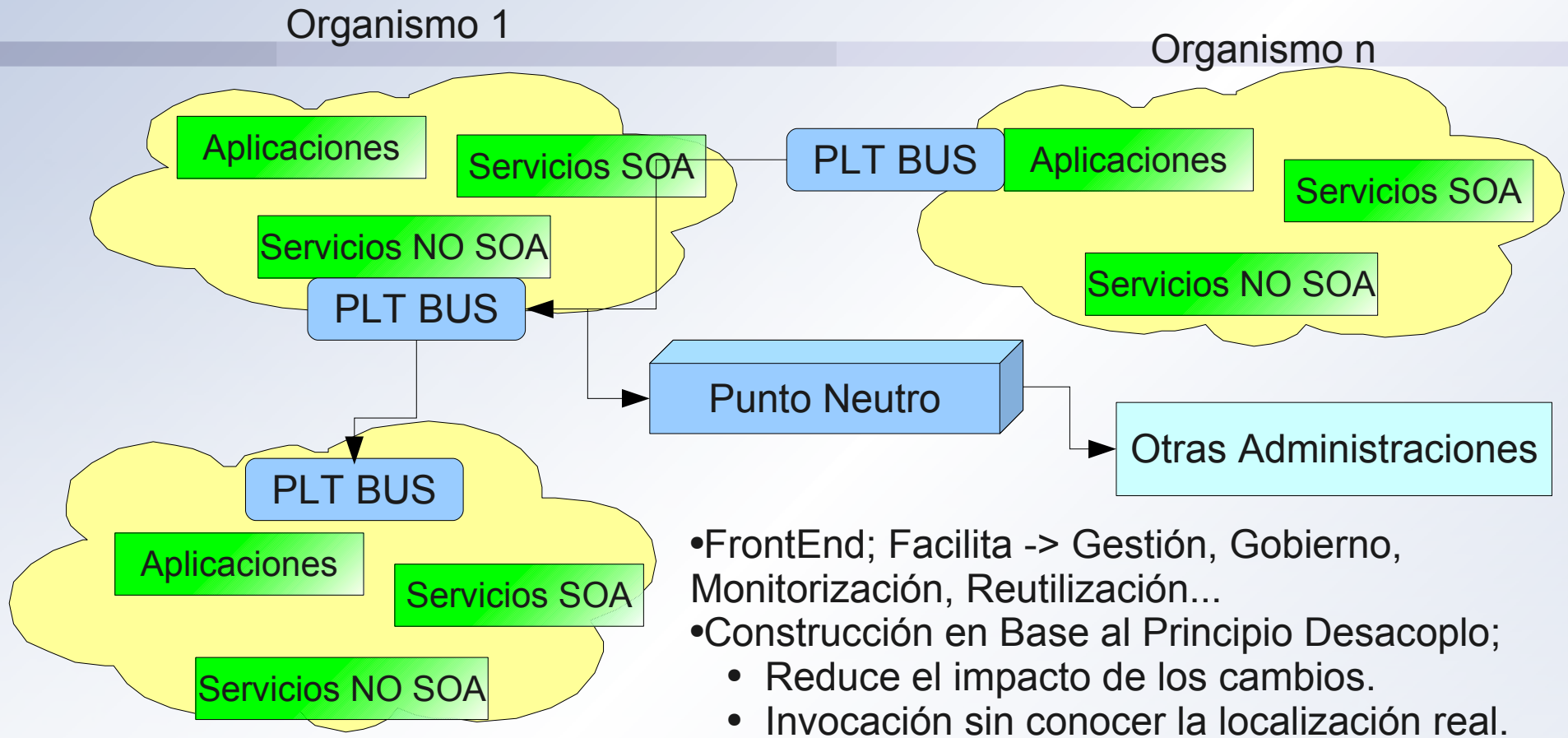
Platina como SOA: Problema SOA en la JA

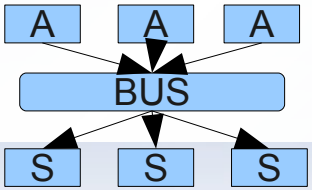


Fuerte acople entre aplicaciones y servicios.
Dificultades para la;

- Gestión
- Gobierno,
- Trazabilidad,
- Seguridad
- Reutilización.

Platina como SOA: Enfoque Solución Platina

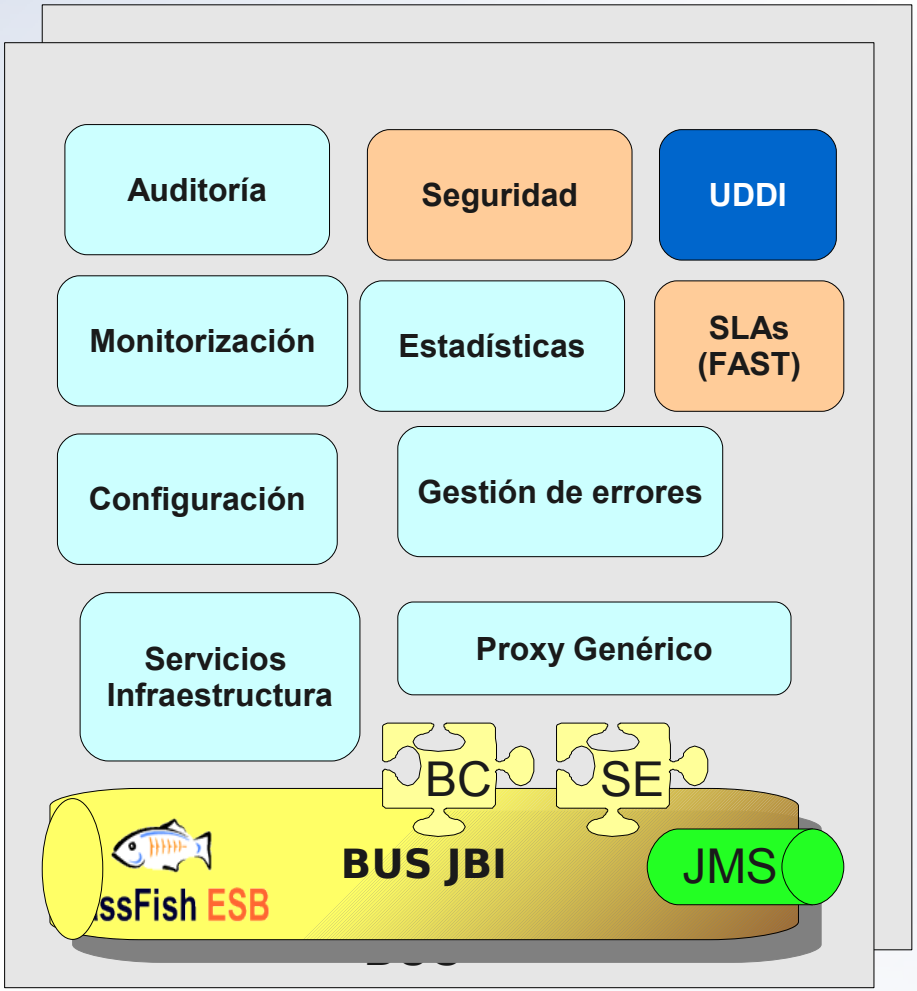


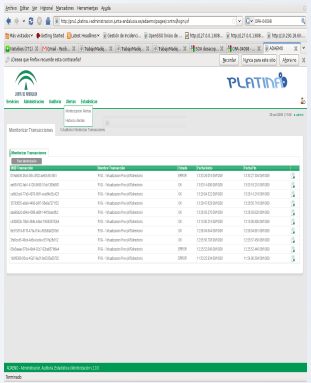


- FrontEnd; Facilita -> Gestión, Gobierno, Monitorización, Reutilización...
- Construcción en Base al Principio Desacoplo;
  - Reduce el impacto de los cambios.
  - Invocación sin conocer la localización real.
  - Virtualización frente al alojamiento de Servicios.
- Uso de EAI para la creación de servicios SOA en base a no SOA.
- Catálogo Central; Gobierno, Descubrimiento
- Fomento de Pautas y Politícas.
- Posibilidad de Crear y Promocionar de servicios Infraestructura

Arquitectura y Diseño: Nodo Platina

Catálogo Central





ADAEMO

Componente Proxy Genérico - Virtualización

El componente Proxy Genérico es un elemento clave en Platina:

- **Facilita** la **publicación de Servicios Web** existentes a través de Platina, convirtiendo una tarea que antes requería cierto desarrollo en un proceso administrativo.
- Basa su funcionamiento en la idea de XML Gateway (**Proxy SOAP**) de forma que permite Virtualizar los servicios Web existentes en un organismo.
- La **Virtualización** de servicios permite desacoplar los servicios de sus consumidores, permitiendo la **publicación** del servicio por distintos **endpoint, interfaces, protocolos**, etc, generando un servicio Virtual Nuevo.
- El servicio virtual, en sí, no contiene lógica de negocio, sólo expone una nueva interfaz sobre un servicio existente.
- Además de las funciones básicas de Virtualización, el **Proxy Genérico** añade a los servicios virtualizados funciones avanzadas como **auditoría, reintentos, seguridad, QoS**. Esto da un aspecto de valor añadido a PLATINA, ya que **permite** que los servicios **deleguen** en platina aspectos como **la gestión de errores, auditoría, monitorización, definición de la seguridad**, **SLA,** etc...

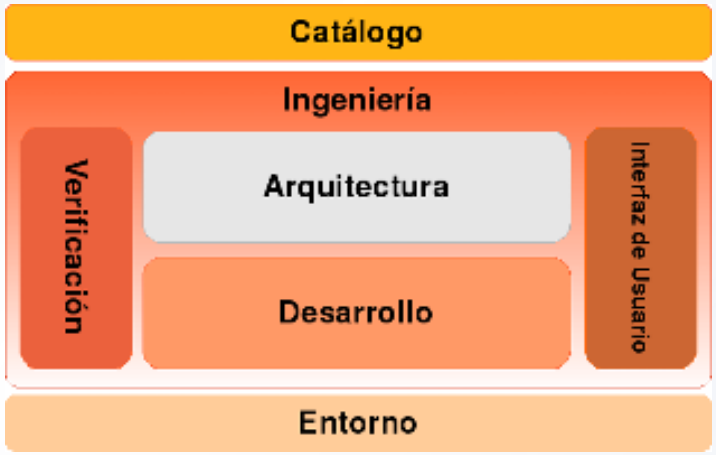
Platina y Madeja - Índice

Platina y Madeja

- Pautas Platina
- Pautas de creación de WS

MADEJA

- MADEJA es el **Marco de Desarrollo de la Junta de Andalucía**. Su misión es proporcionar un entorno que permita a todos los implicados en el desarrollo y en la explotación del software tener una referencia clara de cuáles son las **directrices** que han de guiar esta actividad, así como **dar a conocer los recursos y herramientas** que están a su disposición.
- Subsistemas



Pautas de Platina: Índice

Pautas de Platina

- ¿Cuándo Usar Platina?
- ¿Cómo usar Platina?
  - Escenarios USO.
- Identificación de Aplicaciones y Servicios
- Política Seguridad

MADEJA:¿Cuando Usar Platina?

A grandes rasgos un organismo debería plantearse la instalación de un Platina cuando:

- Se quiera dotar de una **Arquitectura SOA** al Organismo.
- Exista la necesidad de implantar un sistema **EAI** o un Bus para la integración de aplicaciones, datos y procesos de negocio.

Las pautas de cuando usar platina están divididas en dos grupos, en función de cada uno de los objetivos de Platina.

¿Cuando Usar Platina? - SOA

JUNTA DE ANDALUCIA

PLATINA
Plataforma de Interoperabilidad de Andalucía

Sistema	Pauta		Carácter	Observaciones
SOA	Cuando se desee crear una infraestructura SOA dentro de la organización		Recomendado	
	Se desee un punto centralizado para el aprovisionamiento y/o consumo de servicios		Recomendado	
	Se requiera desacoplar consumidores y Servicios; ○ Disminuyendo las ríguidas dependencias entre servicios y sus consumidores, lo que facilita, entre otras cosas, la evolución de los servicios reduciendo el impacto sobre los consumidores. ○ Conectando recursos sin tener en cuenta su ubicación, (UDDI, proxys).		Recomendado	
	Provisión de servicio	Proveer servicios para su consumo fuera del organismo de origen	Recomendado	
		Provean servicios para su consumo interno a la organización y el número de consumidores/servicios sea suficientemente alto como para que sea aconsable su publiación mediante Platina	Recomendado	
		Proveer servicios de infraestructura	Recomendado	
	Facilitar el gobierno, gestión y monitorización de los servicios provistos y/o consumidos por un organismo		Recomendado	

¿Cuando Usar Platina? - EAI

Sistema	Pauta	Carácter	Observaciones
EAI	Se necesite una plataforma para la integración de servicios, aplicaciones, procesos y datos	Recomendado	
	Se desee incrementar y/o facilitar la integración de procesos o servicios existentes mediante la publicación por distintos transportes (http, mail), protocolos (WS,RMI, Corba) o usando distintos patrones de interacción (MEP; sincrono, asíncrono...).	Recomendado	
	Se requiera realizar procesos de mediación sobre servicios existentes que requieran trasformaciones, particiones, ampliaciones o enriquecimiento de mensajes sin modificación de los servicios existentes.	Recomendado	
	Se requiera realizar procesos de enrutamiento de mensajes ya sean simples o basados en contenido, asunto, itinerario (origen/destino), SLAs, etc.	Recomendado	
	Se necesite realizar procesos de integración de datos EII (Enterprise Information Integration); Data Mashup, Data Migration, Data Integraction, Data Quality (componente JBI MURAL)	Recomendado	
	Se desee crear un front-end común para un conjunto de aplicaciones, de forma que se proporciona un punto de acceso único para todas estas aplicaciones	Recomendado	
	Independencia de proveedor: extrayendo las políticas o reglas del negocio de las aplicaciones e implementándolas en Platina, de forma que cualquiera de las aplicaciones puedan ser cambiadas sin que dichas reglas de negocio deban ser reimplementadas	Recomendado	
	Se desee centralizar en una sola herramienta la gestión de requisitos no funcionales asociados a los servicios, como puedan ser; Auditoría, Estadísticas, Seguridad, SLAs, etc	Recomendado	
	Se requiera una herramienta que facilite la migración de servicios existentes no SOA a servicios SOA	Recomendado	Justifiación por la que se usan ESB para crear infraestructuras SOA

¿Cómo Usar Platina? - SOA I

Pauta		Carácter	Observaciones
Contrucción de la Arquitectura SOA siguiendo el principio de desacoplo entre consumidores y proveedores, en lugar de usar platina como un Servidor de Aplicaciones donde desplegar Servicios de Negocio		Recomendado	
Desarrollo de los servicios de forma tradicional, desplegados en servidores independientes y publicados a través de Platina.		Recomendado	
División conceptual de Servicios en; Servicios de Negocio, Aplicación e infraestructura, siguiendose las pautas identificadas para cada tipo. ○ Servicios de Negocio enfocados como tareas o procesos y servicios de aplicación como Entidades compuestas por operaciones ○ Teconología de Web Services para la creación de Servicios de NegocioA. Libertad tecnológica para la creación de Servicios de Aplicación y de infraestructura.		Recomendado	Vocabulario común
Puerta única de acceso a los servicios.Front-End	Publicación a través de Platina de los Servicios de Negocio y Aplicación consumidos desde otros organismos.	Recomendado	
	Acceso de los Servicios de Negocio y Aplicación de terceros organismos a través de Platina	Recomendado	
	Acceso de los Servicios de Negocio y Aplicación propios del organismo a través de Platina	Recomendado	
Empleo de técnicas de virtualización de servicios para la publicación de Servicios a través de Platina.		Recomendado	
Delegación en Platina de Aspectos no funcionales de los servicios, como son: ○ La gestión de acceso a los servicios (gestión de aplicaciones consumidoras). ○ Securización de los servicios. ○ Auditoría y traceabilidad de las comunicaciones.		Recomendado	

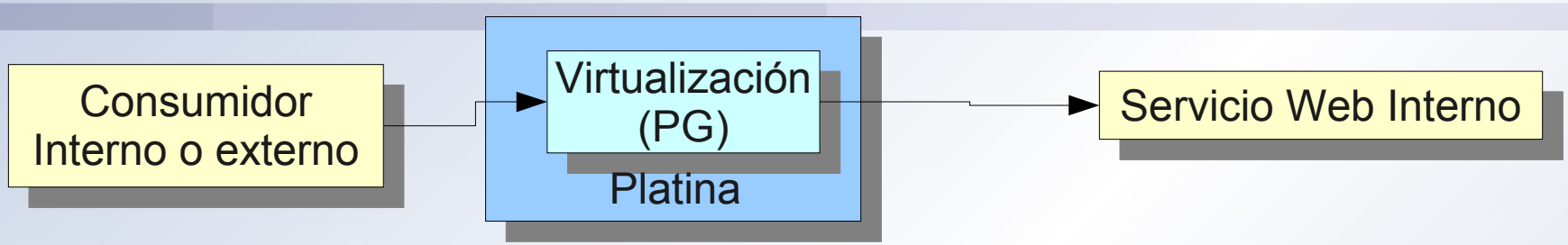
¿Cómo Usar Platina? - SOA II

Uso de ADAEMO para la gestión, monitorización y descubrimiento local de servicios dentro de un organismo: o Registro en Platina de los servicios disponibles dentro la organización o Registro en Platina de los servicios externos de interes para la organización o Registro de los aplicaciones (internas o externas) consumidoreas de servicios locales. o Registro de las aplicaciones locales consumidoras de servicios externos. o Uso de las herramientas de monitorización, alarmas y estadísticas.	Recomendado	
Uso del catálogo central de Servicios de Platina para el registro de Servicios y puntos de publicación de mismos, con el objeto de facilitar: o Gobierno Central de Servicios para toda la Junta. o Descubrimiento de servicios publicados. o Reutilización del Software a nivel de servicios.	Recomendado	
Uso de los servicios de infraestructura de platina, como Auditoría, STS, Trasnferencia Ficheros	Recomendado	

¿Cómo Usar Platina? - EAI

Pauta	Carácter	Observaciones
Resolver dentro de platina los problemas de integración de aplicaciones, servicios y procesos, cuando no sea viable la modificación directa de la aplicacón, servicio o proceso.	Recomendado	
En globlar los problemas de integración dentro de la tipología de problemas conocidos por platina; mediación, enrutación, transformación y virualización, resolviendolos, siempre que sea posible, empleando los componentes de platina diseñados para cada caso concreto y siguiendo las pautas definidas en cada uno de ellos.	Recomendado	
Uso de los servicios de infraestructura. Independientemente de los mecanismo de integración, platina proporciona servicios de infraestructura especialmente diseñados para resolver problemas concretos de integración (SAML).	Recomendado	

Escenarios - Publicación de un Servicio Web Existente.



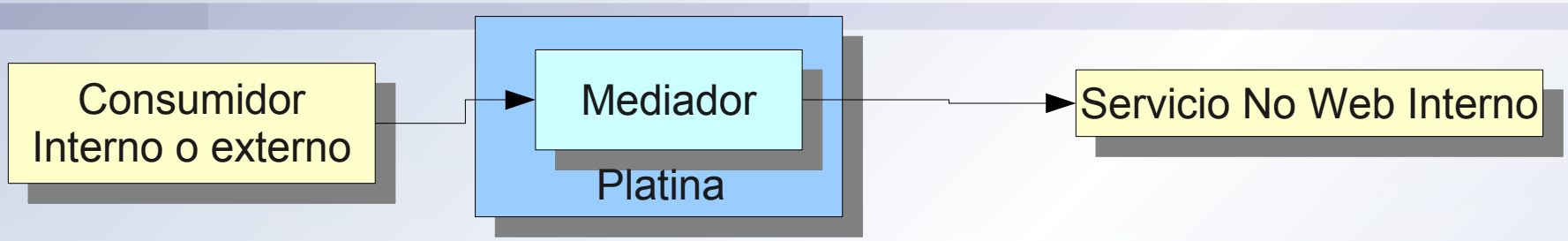
La recomendación de Platina es que este escenario:

- Se resuelva aplicando **Virtualización** de Servicio.
- Uso del **Proxy Genérico**.

Con ello se consigue:

- Desacoplar Consumidores y Proveedores.
- Posibilidad de proporcionar distintas interfaces del servicio, seguridad, QoS, etc.
- Posibilidad de delegar en Platina, aspectos como auditoría, gestión de consumidores, Gestión Errores, Monitorización, Seguridad.

Escenarios - Publicación de un Servicio No Web



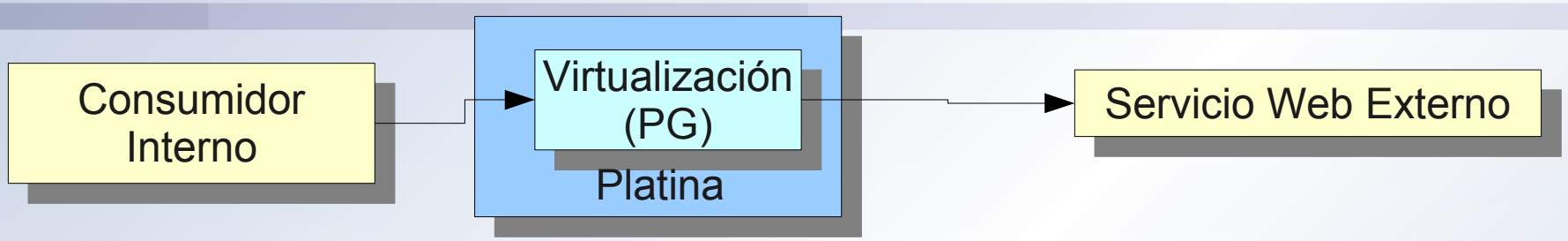
La recomendación de Platina es que este escenario:

- El servicio no Web sea convertido en Servicio Web
- La conversión a servicio web se realiza mediante el desarrollo de un componente de transformación en Platina, desarrollado en base al componente JBI más apropiado al tipo de servicio a integrar.

Con ello se consigue:

- Desacoplar Consumidores y Proveedores.
- Proporcionar distintas interfaces al servicio.
- Posibilidad de delegar en platina, aspectos como auditoría, gestión de consumidores, seguridad.

Escenario Consumición de servicios Externos



La recomendación de Platina es que este escenario:

- Se resuelva aplicando **Virtualización** de Servicio.
- Uso del **Proxy Génerico**.
- El uso de ese servicio por parte de las aplicaciones del organismo se haga a través del Servicio Virtualizado.

Con ello se consigue:

- Desacoplar Consumidores y Proveedores.
- Posibilidad de proporcionar distintas interfaces del servicio, seguridad.
- Posibilidad de delegar en platina, aspectos como auditoría, gestión de consumidores, seguridad, de forma centralizada a todas las aplicaciones.

Identificación de Aplicaciones y Servicio I

- La descentralización, distribución y federación de platinas y aplicaciones, además de las necesidades de gobierno, requieren la **identificación única de servicios y aplicaciones**.
- Cada aplicación y servicio dentro de la plataforma vienen identificados mediante **dos urn**. Una urn que **identifica a la aplicación o servicio y otra urn que identifica a la instancia** concreta del servicio.
- Las **urns** empleadas para la **identificación** de aplicaciones y servicios son **compartidas** por todas las instancias de esa aplicación o servicio **y** son **determinadas por el organismo impulsor** de la aplicación o servicio.
- Las **urns** empleadas para la identificación de las **instancias** de aplicaciones o servicios han de ser **únicas en cada instancia** de aplicación. Estas serán fijadas por el **organismo responsable de la instancia** del servicio o aplicación.

Identificación de Aplicaciones y Servicio II

Para facilitar el uso y la creación de urns únicas dentro de toda la Junta de Andalucía, se propone el siguiente **formato de urn**, el cual **posibilita la creación de distintos espacios de urns** sobre los cuales cada organización concreta puede trabajar con total libertad, evitándose así problemas de colisión de urns.

Formato urn para aplicación

urn:es:juntadeandalucia:(organismo impulsor):(aplicacion)

Formato urn para instancia de aplicación

urn:es:juntadeandalucia:(organismo instancia):(aplicacion)

Formato urn para Servicio

urn:es:juntadeandalucia:(organismo impulsor):(aplicacion):(servicio):(version)

Formato urn para instancia de servicio

urn:es:juntadeandalucia:(organismo instancia):(aplicacion):(servicio):(version)

Pautas Creación WS

El éxito de SOA está fuertemente ligado a la calidad de los servicios SOA

Índice Pautas WS

Publicadas en Madeja:

http://madeja.i-administracion.junta-andalucia.es/madeja/

Subsistema Arquitectura/Integración

Tabla de pautas	Tipo	Carácter
Creación de Servicios Web	Directriz	Obligatoria
Reglas de Codificación	Directriz	Obligatoria
Política de Versionado	Directriz	Obligatoria
Identificación de Mensajes y EndPoints (WS-Addressing)	Directriz	Obligatoria
Seguridad	Directriz	Obligatoria
Manejo Errores	Directriz	Obligatoria

Tabla de recursos	Tipo	Carácter	Versión
Conjunto de APIs y Frameworks de Servicios Web	-	Recomendado	-
Referencias de Especificaciones y Estándares	-	Recomendado	-
Plataforma de Interoperabilidad de la Junta de Andalucía	-	Recomendado	-

Pautas Creación: Estado del “Arte” en la creción de WS

Existen **muchas formas distintas de entender, crear e implementar** un WS:

- Existen **3 Modelos** para la creación de Servicios: Modelo RPC, SOA y REST
- Existen **dos aproximaciones** a la hora de desarrollar un WS: Contract-first y Code-First.
- Existen **diversos mecanismos** para la **codificación** de un WS: RPC/encoded, RPC/literal y document/literal

Cada una de estas variaciones tienen sus ventajas e inconvenientes, a la hora de desarrollar el servicio, en el grado de interoperabilidad del mismo, en la capacidad para asumir cambios, etc.

Pautas Creación de WS: Modelo RPC, SOA y REST

Dentro del mundo de los Web Services, tres modelos de programación conviven en la actualidad; el Modelo RPC, el Modelo de Mensajes (Modelo SOA) y el modelo REST

- El **Modelo RPC** era el modo tradicional de desarrollo en la Junta, Su filosofía puede resumirse en la invocación de objetos remotos. **Presenta limitaciones y problemas de interoperabilidad**.

- El **Modelo de Mensajes** basa su funcionamiento en el intercambio de documentos XML. Engloba al Modelo RPC, **elimina los problemas y limitaciones** de este.

- El Modelo Rest se basa en la invocación libre de recursos Web.

Pautas Creación de WS: Code-first y Contract-First

Existen dos aproximaciones para la creación de un WS; "Code-First" y "Contract-First".

- En la aproximación "Code-First", primero se codifica el código del servicio dentro del lenguaje de programación y después se genera automáticamente el WSDL. Esta opción presenta una serie de inconvenientes debido principalmente a lo ligado del WSDL con respecto al servicio y framework y a las dificultades de personalizarlo.

- En la opción "Contract-First",la idea es la contraria, primero se genera el WSDL y después se codifica el servicio. Esta opción carece de los inconvenientes de la opción anterior y aporta numerosas ventajas a la hora de desarrollar y mantener los Servicios Webs.

Pautas Creación de WS: Estilos de codificación RPC/encoded, RPC/literal y document/literal

En la actualidad, existen tres mecanismo para la codificación de los mensajes SOAP; RPC/encoded, RPC/literal y Document/literal.

- La codificación **RPC/encoded no** está **recomendada** por la WS-I y queda prohibido su uso.
- La codificación **RPC/literal** no está aconsejada por MADEJA, aunque **se permitirá** su uso en aquellos Servicios Webs que se desarrollen siguiendo un modelo RPC y que no se integren dentro de PLATINA.
- La codificación **document/literal es la recomendada** dentro de MADEJA, siendo obligatoria para Servicios Web que se integren dentro de PLATINA.

Pautas Creación de WS

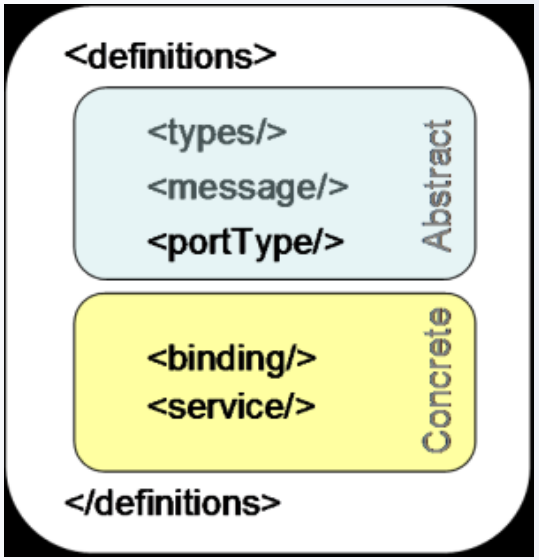
La recomendación de MADEJA puede resumirse en tres puntos

- **Modelo De Mensajes o Modelo SOA**
- **Aproximación Contract-First**
- **Estilo document/literal**

Pautas	Carácter	Observaciones
Modelo de Mensajes (SOA)	Recomendado	Obligatorio para servicios que hacen uso de PLATINA
Modelo RPC	Permitida	Para servicios internos que NO hacen uso de PLATINA
Modelo REST	Permitida	Para servicios internos que NO hacen uso de PLATINA
Aproximación Contract-first	Recomendado	-
Aproximación Code-first	Permitida	Para servicios internos que NO hacen uso de PLATINA
Estilo document/literal	Recomendado	Obligatorio para servicios que hacen uso de PLATINA
Estilo RPC/Encoded	No recomendado	No soportado por la WS-I
Estilo RPC/literal	Permitido	Para servicios internos que NO hacen uso de PLATINA

Reglas de Codificación

- **Codificación UTF-8**: WSDLs, esquemas xsd y mensajes SOAPs deberán estar codificados en UTF-8
- **Codificación de los Namespaces**
  - **urn:es:juntadeandalucia:(consejería):(aplicación):(servicio):(componente):versión**
- **Codificación de los WSDL**



Política de Versionado

Al igual que cualquier otro componente software, los **Servicios Webs** no son elementos inmutables en el tiempo, sino que **están sujetos a cambios** a lo largo de toda su vida

El **versionado** dentro de los Servicios Webs **es un tema abierto** en la actualidad, en **MADEJA** se da establece una **Política**, que recoge un conjunto de normas y recomendaciones para establecer el versionado de WS.

La base para la **Política** de **versionado** de WS de MADEJA consiste en:

- Identificación y difusión del Problema, identificando los tipos de cambios que se producen en los WS y su impacto.
- Uso del namespace como elemento clave que nos permite versionar.
- Establecimiento de las políticas en función del tipo de WS.

Versionado: Tipos cambios

Tipos de Cambios:

- Cambios **con compatibilidad hacia atrás**, son cambios en el Servicio Web que no "rompen" el servicio y que por tanto, no obliga a reimplementar los consumidores de dicho servicio. Inclusión de nuevas operaciones. Inclusión de nuevos tipos dentro del esquema.

- Los cambios **sin compatiblidad hacia atrás**, son cambios que si "rompen" el servicio y que por tanto, obligan a reimplementar los consumidores de dicho servicio; Eliminación de una operación, Renombrado de una operación, Cambio en los parámetros de una operación, Cambios en un tipo de datos existente dentro del esquema.

Política Versionado: Pautas

MADEJA permite la creación de servicios Webs según tres modelos: RPC, SOA y REST. Las políticas de versionado variarán en cada caso para adaptarse a las posibilidades que proporciona cada modelo.

Tipo	Trazabilidad	Pauta
RPC	-	El elemento empleado para dirigir la política de versionado es el namespace del servicio, al cual se le dotará de un número de versión
SOA	SIN	Esta aproximación sería similar a la del modelo RPC. El namespace del servicio incluirá un número de versión, el cual no variará ante los cambios con compatibilidad hacia atrás, y que se incrementaría en caso contrario.
SOA	CON	En esta aproximación se asignará un namespace con versión al Servicio Web. A diferencia de los casos anteriores, este namespace sólo será empleado para definir al servicio, no será reutilizado por otros elementos como los types, bindigs, etc. Al igual que en los otros casos, cuando se produzca un cambio que implique perdida en la compatibilidad hacia atrás, el número de versión del servicio se incrementará para obligar así a regenerar las interfaces en los consumidores del servicio.
REST	-	Sin recomendación para el mantenimiento de la versión

Identificación de Mensajes y EndPoints

- La identificación de los mensajes SOAP y EndPoints es un **elemento esencial** a la hora de realizar procesos de **auditoría, monitorización, detección de duplicados,** etc.
- La especificación **WS-Addressing recoge mecanismos** para la identificación de estos elementos de forma estandarizada e independiente al protocolo de transporte empleado

- **Pauta:**
  - La especificación WS-Addressing debe ser recogida, por todos los servicios Webs (RPC y SOA) desarrollados para la Junta de Andalucía.
  - El elemento MessageID definido por esta especificación como opcional pasará a ser obligatorio dentro de MADEJA para todas las aplicaciones.

Manejo Errores: Tipos de Errores

Existen 3 tipos de errores:

- Errores en la invocación del servicios,
- Errores específicos del servicio,
- Warnings.

- La notificación de errores dentro de los Servicios Webs deberá realizarse mediante el empleo de **SOAP Faults**.
- Siempre que sea posible, los mensajes SOAP Faults seguirán la estructura definida en la especificación **SOAP 1.2** en lugar de la recogida en SOAP 1.1
- Se deberá **evitar** incluir dentro de los mensajes de error **información sensible del servicio, detalles de la implementación del mismo e información no útil**, de cara a los consumidores, dentro de los mensajes de error

Manejo Errores: Recomendaciones básicas para Errores del Servicio

- Los errores han de ser **comunicados de forma independiente** al S.O., lenguaje de programación y aplicación.
- Los errores han de ser presentados de tal forma que se **facilite su interpretación y procesado**. Utilización de códigos y subcódigos de error.
- **Nunca exponer detalles internos** de implementación del servicio a los consumidores, ya que esto supone un riesgo de seguridad para el propio servicio y aporta información NO Útil de cara al consumidor del mismo.
- **Nunca devolver la traza de las excepciones**.
- **Sólo devolver los errores tratables por el consumido**r. Errores no procesables por el consumidor deberán ser enmascarados y transmitidos mediante un error genérico

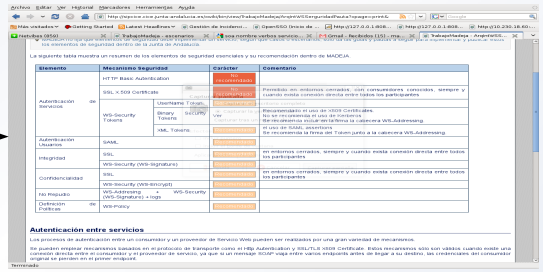
Política de Seguridad en Platina

Objetivos

Definir normas y recomendaciones para la Securización de los Servicios publicados a través de la Plataforma Platina.

Flujo de Seguridad

Determinar los requisitos de Seguridad asociados al Servicio; Autenticación, Integridad, Confidencialidad y No Repudio



Identificar los escenarios de publicación del servicio dentro de Platina.

Asociados a los escenarios hay pautas
Asociados a los Requisitos de Seguridad hay Pautas sobre los mecanismo de seguridad que los cubre
El tercer paso es mapear ambos conjuntos de pautas para determinar los mecanismo de seguridad más adecuados para cada escenario

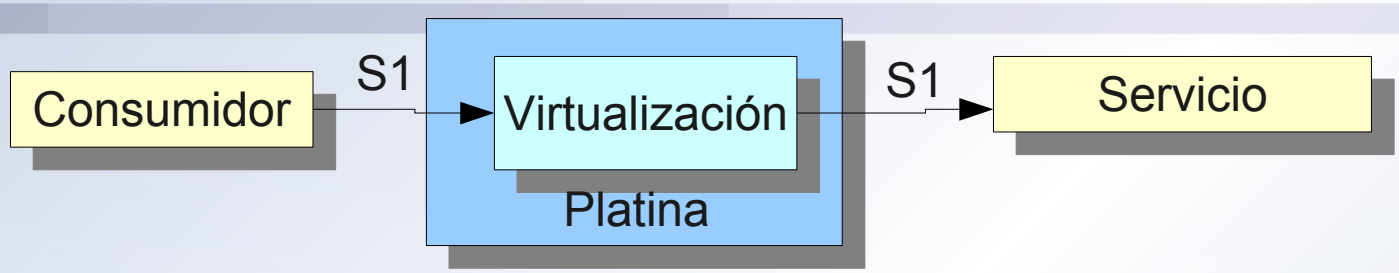
De la lista de mecanismos de seguridad resultantes del paso anterior, elegir el mecanismo más adecuado para cada servicio, en función del nivel de seguridad, potencial de ataque, complejidad, interoperabilidad, etc

Pautas Seguridad: RNF vs Mecanismos Seguridad

PLATINA
Plataforma de Interoperabilidad de Andalucía

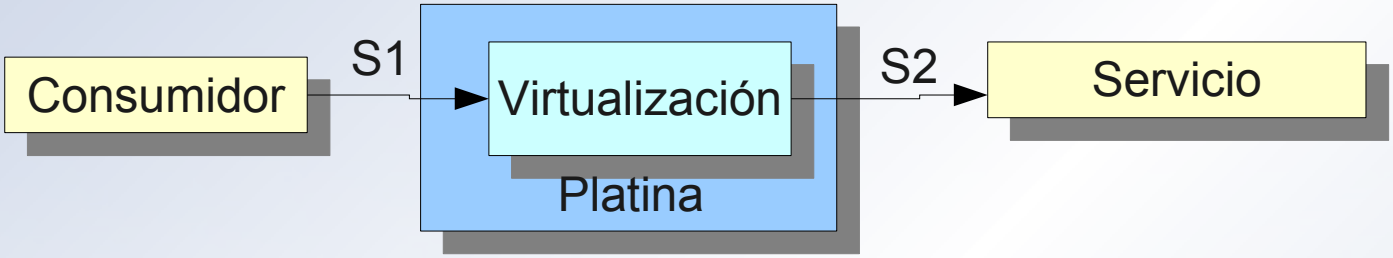
Elemento	Mecanismo Seguridad		Carácter	Comentario
Autenticación de Servicios	HTTP Basic Autentication		No recomendado	
	SSL X.509 Certificate		No recomendado	Permitido en entornos cerrados, con consumidores conocidos, siempre y cuando exista conexión directa entre todos los participantes
	WS-Security Tokens	UserName Token	Recomendado	
		Binary Security Tokens	Ver	Recomendado el uso de X509 Certificates. No se recomienda el uso de Kerberos Se recomienda incluir en la firma la cabecera WS-Addressing.
		XML Tokens	Recomendado	el uso de SAML assertions Se recomienda la firma del Token junto a la cabecera WS-Addressing.
Autenticación Usuarios	SAML		Recomendado	
Integridad	SSL		Recomendado	en entornos cerrados, siempre y cuando exista conexión directa entre todos los participantes
	WS-Security (WS-Signature)		Recomendado	
Confidencialidad	SSL		Recomendado	en entornos cerrados, siempre y cuando exista conexión directa entre todos los participantes
	WS-Security (WS-Encrypt)		Recomendado	
No Repudio	WS-Addresing + WS-Security (WS-Signature) + logs		Recomendado	
Definición de Políticas	WS-Policy		Recomendado	

Escenario: Platina Elemento Pasivo de Seguridad



- **Seguridad** definida por el s**ervicio final**.
- **Platina no verifica** la seguridad ni autentica al consumidor.
- Platina invoca al servicio final con un **mecanismo de seguridad compatible** con el servicio final y pasa las credenciales del consumidor sin modificarlas ni invalidarlas.
  - Para que esto sea posible los **mecanismo** empleados en la **autenticación** de las aplicaciones han de estar **basados en mensaje**, nunca en protocolo.
  - Para lograr la **integridad y confidencialidad** de las comunicaciones se recomienda el uso de **mecanismos basados en mensajes**, aunque se puede emplear SSL.

Escenario: Platina Elemento Activo de Seguridad I



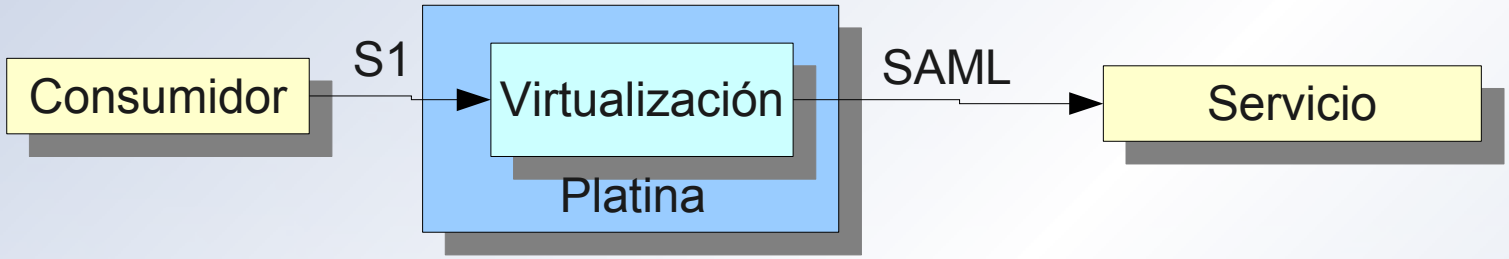
- **Platina verifica** el cumplimiento de las restricciones de seguridad y autentica aplicación. Existe un registro de consumidores dentro de Platina.
- La invocación al **servicio final** puede seguir unas **reglas** de securización **distintas** a las de invocación del servicio virtualizado. Mediación en Seguridad
- En la llamada del **servicio virtualizado** hacia el final, el servicio virtualizado **se puede identificar** como servicio consumidor o pasar las credenciales de la aplicación consumidora.

Escenario: Platina Elemento Activo de Seguridad II

Este escenario tiene pocas restricciones de seguridad.

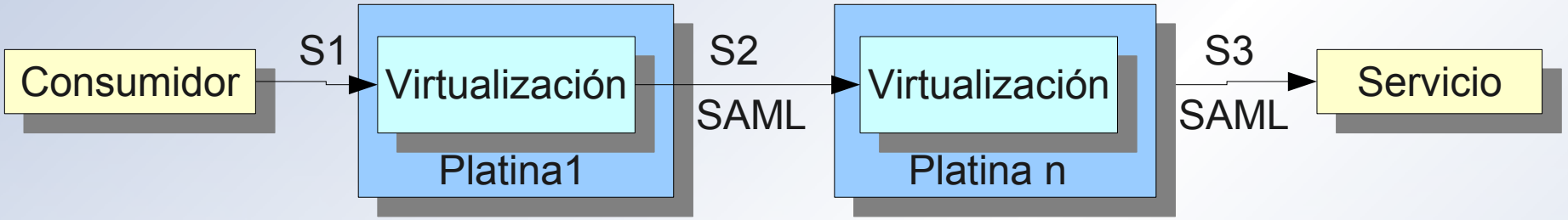
- Se puede emplear **cualquier mecanismo** de seguridad para logra la confidencialidad e **integridad** de las comunicaciones, hacia platina y desde platina al servicio final (si aplica), pudiendo incluso emplearse mecanismos distintos en ambos casos (mediación de seguridad).
- El mecanismo de **autenticación** empleado, en principio, puede ser cualquiera, excepto en el caso en el que el Servicio Virtualizado tenga que pasa las credenciales del consumidor al servicio Final. En este caso, el mecanismo de autenticación ha de ser basado en mensaje no en protocolo y este debe seguir siendo válido en la llamada al servicio final.

Escenario: Platina como Identity Provider (IdP)



- **Platina verifica** el cumplimiento de las restricciones de seguridad y autentica aplicación. Existe un registro de consumidores dentro de Platina.
- **Platina actúa como idP**, autenticando a la aplicación consumidora y generando un Token SAML.
- La invocación a**l servicio final** puede realizarse con **cualquier mecanismo** de seguridad, el servicio Virtualizado incluirá en la llamada al servicio final el token SAML con la identidad de la Aplicación consumidora.
- No obstante, se recomienda que la invocación al servicio final se haga siempre con: **SSL + Usernametoken del Servicio Virtualizado + SAML** consumidor.

Escenario: Federación de Platinas I



- El **primer platina** actúa como **idP**, autenticando a la aplicación consumidora y generando un Token SAML. Existe un registro de consumidores dentro del primer Platina.
- La llamada entre Platinas se realiza mediate **Mutual Certificate** y se pasa el **token SAML** de la aplicación consumidora.
- **Ningún Platina intermedio verifica** la identidad de la aplicación consumidora. El último Platina, puede hacer una última validación de autenticación, o no, contra su propio registro de consumidores.

Escenario: Federación de Platinas II

- La invocación al servicio final puede realizarse con cualquier mecanismo de seguridad, el servicio Virtualizado **incluirá** en la llamada al servicio final el **token SAML** con la identidad de la Aplicación consumidora.
- No obstante, **se recomienda** que la invocación al servicio final se haga siempre con: **SSL + Usernametoken del Servicio Virtualizado + SAML consumidor**.

¿?