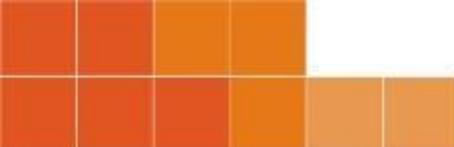


Esquema de Integración Externa Aranda

Versión 8.0



Historia de Revisiones

Fecha	Versión	Descripción	Autor
20/04/2010	1.0	Documento inicial	Juan Francisco Carrillo
22/10/2012	1.1	Actualización tecnologías integración	Francisco Luis Rodríguez
04/07/2013	1.2	Actualización tecnologías integración	Diego A. Ponce

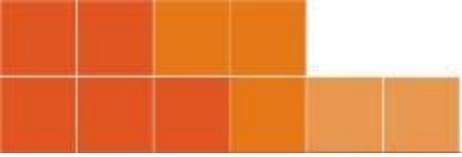


Tabla de Contenido

1.	Introducción	4
1.1	Propósito	4
1.2	Alcance	4
1.3	Definiciones, Acrónimos y Abreviaturas	4
1.4	Referencias	4
2.	Tecnologías de Integración	4
2.1	Web Services	4
3.	Esquemas de Integración	5
3.1	Por Demanda	5
3.2	Por Evento	6
3.3	Doble Vía	7
3.4	Ejemplo	7

1. Introducción

En la operación normal de los productos de Aranda Software es normal tener que intercambiar información con otros sistemas de información. La versión 8.0 de la suite incluye modificaciones en el esquema de integración externa con respecto a lo que se tenía en la versión 7.2. Estos cambios incluyen: 1) Exposición del API a través de WebServices para Service Desk, Self Service, CMDB, 2) Invocación de Web Services externos desde el modulo de reglas a partir de condiciones internas del sistema.

1.1 Propósito

Este documento tiene como propósito presentar el esquema general de integración de los productos de Aranda Software para la versión 8.0 con otras aplicaciones de terceros.

1.2 Alcance

El alcance de este documento no incluye la especificación y documentación de los Web Services referenciados, para tal fin remitirse a la documentación específica de los mismos.

1.3 Definiciones, Acrónimos y Abreviaturas

Sin definir.

1.4 Referencias

Sin definir.

2. Tecnologías de Integración

2.1 Web Services

Al igual que como las librerías del API, toda la lógica del negocio de cada aplicación se puede consumir a través de Web Services desarrollados para tal fin. Se recomienda este esquema de conexión para ambientes en el que el conector desarrollado deba estar corriendo en una maquina diferente donde se tenga la instalación de Aranda o para conectores desarrollados en otras plataformas diferentes a .NET. A continuación se presenta la lista de Web Services disponibles por aplicación:

- Service Desk: Aranda.ASDK.Service.Interfase
- Self Service: Aranda.ASS. Service.Interfase
- CMDB: Aranda CMDB.Service.Interfase

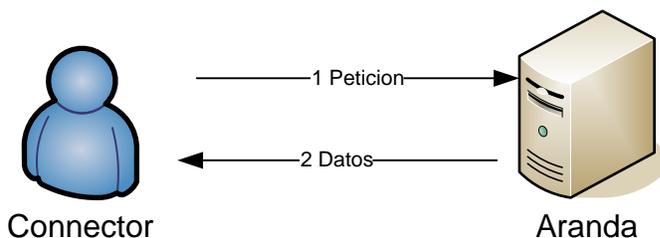
Los Web Services están desarrollados en .NET Framework 2 y corren sobre IIS (Internet Information Server). Una vez instalados es posible configurarlos para correr sobre https y consumirlos a través de proxies. Estos módulos no se incluyen en los instaladores oficiales de la versión.

Nota: referirse a la documentación de cada Web Service para los detalles de instalación, utilización y dependencias requeridas, en el documento [ARANDA WEB SERVICES DOCUMENTATION](#).

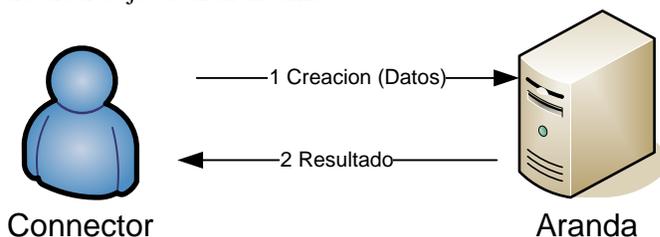
3. Esquemas de Integración

3.1 Por Demanda

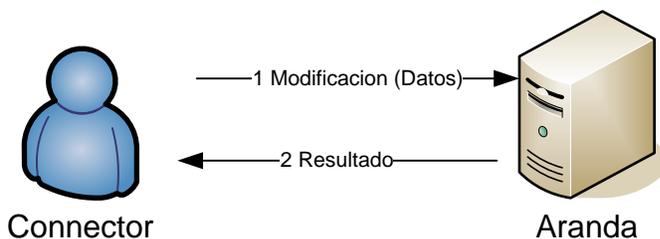
Para las dos tecnologías referidas anteriormente, es responsabilidad del conector invocar el API de Aranda ya sea para consultar información del estado de los objetos del sistema:



Para crear objetos en el sistema.



O para modificar el estado de los objetos del sistema.



Este esquema de conexión es adecuado si los objetivos de la conexión son:

- Interactuar con Aranda en respuesta a la interacción del usuario con el Sistema Externo.
- Interactuar con Aranda en respuesta a eventos en el Sistema Externo.

Si lo que se quiere es mantener sincronizada, información redundante en los dos sistemas, es responsabilidad del conector desarrollado:

1. Consultar periódicamente el estado de los objetos en Aranda
2. Determinar si hubo cambios en Aranda.
3. Actualizar el sistema externo según los cambios detectados.

En este caso es preferible utilizar el esquema de doble vía detallado más adelante.

3.2 Por Evento

En la versión 8.0 es posible invocar un Web Service externo en el momento que en Aranda los objetos del negocio cumplen una condición específica. Esta funcionalidad está disponible para las siguientes aplicaciones y los siguientes objetos:

- Service Desk:
 - Incidentes
 - Problemas
 - Cambios
 - Llamadas de Servicio
- Self Service:
 - Artículos
- CMDB:
 - CIs

Cada una de estas 3 aplicación cuenta con un modulo de reglas que opera de manera similar. Una regla es la combinación de una **Condición** y una lista de **Acciones**. Una **Condición** es una expresión booleana (True, False) que puede incluir operadores booleanos (NOT, AND y OR) y statements sobre el cada uno de los objetos soportados. Por ejemplo una condición para un Incidente en Service Desk puede ser: **Estado = “Registrado” AND Categoria = “Bug”**. Las reglas se definen independientemente para cada tipo de objeto (Incidentes, Problemas, Cambios, Llamadas de Servicio, Artículos, CIs).

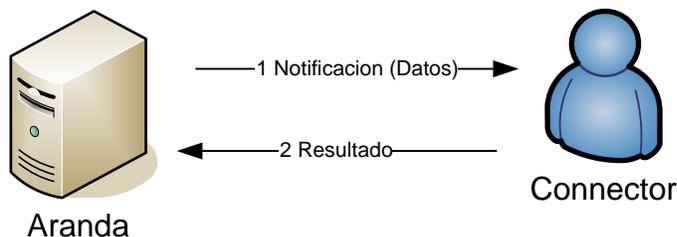
El otro elemento que compone una regla es la lista de **Acciones**. Cada tipo de objeto tiene un conjunto de acciones disponibles. Estas **Acciones** se ejecutan secuencialmente cuando el objeto cumple la **Condición**. En este documento nos centramos solo la acción **Invocar Web Service**. Esta acción requiere la configuración de un solo parámetro: la URL del Web Service a invocar.

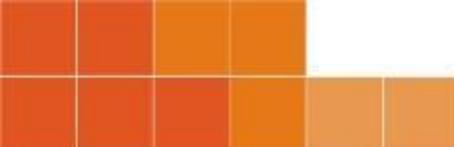
El Web Service a invocar deber ser desarrollase como parte del conector y debe cumplir la especificación del archivo ArandaExternalConnector.wsdl.

Se deben implementar solo 6 métodos, uno por cada tipo de objeto que será invocado cada vez que algún objeto del sistema cumpla la condición definida en las reglas.

```
bool ArticleOnchange (Aranda.ASS.Common.Entities.Article item);
bool ChangeOnchange (Aranda.ASDK.Common.Entities.ChangeDescription item);
bool CIOnchange (Aranda.CMDB.Common.Entities.InfoCI item);
bool IncidentOnchange (Aranda.ASDK.Common.Entities.IncidentDescription item);
bool ProblemOnchange (Aranda.ASDK.Common.Entities.ProblemDescription item);
bool RequestOnchange (Aranda.ASDK.Common.Entities.ServiceCallDescription
item);
```

Cada uno de estos métodos recibe como parámetro el objeto del negocio que incluye toda la información disponible y se debe retornar el resultado del proceso de notificación.





Este esquema de conexión es adecuado si se quiere:

- Notificar al Sistema Externo de modificaciones en Aranda.

3.3 Doble Vía

Los dos mecanismos anteriores se pueden utilizar simultáneamente para:

- Interactuar con Aranda en respuesta a la interacción del usuario con el Sistema Externo.
- Interactuar con Aranda en respuesta a eventos en el Sistema Externo.
- Notificar al Sistema Externo de modificaciones en Aranda.

Por último los siguientes objetos del API utilizados en el intercambio de información en ambos esquemas cuentan con un campo adicional “interfaceId”:

- Article
- ChangeDescription
- InfoCI
- IncidentDescription
- ProblemDescription
- ServiceCallDescription

Este campo es almacenado en la base de datos de Aranda y puede ser modificado solo por las interfaces de integración es decir solo puede ser modificado por el conector. Este campo se utiliza para relacionar un registro especificado en Aranda con su contraparte en el sistema externo.

3.4 Ejemplo

Suponga que se quiere mantener sincronizados bidireccionalmente el campo **Estado** de los Incidentes de Service Desk contra el campo **Estado** de un registro tipo **X** en un sistema externo. A continuación se muestran las actividades necesarias para lograr esta integración:

1. Implementar un cliente del Web Service Aranda.ASDK.Service.Interface\ Incident.asmx
 - a. Invocar el método Add cada vez que se cree un registro **X** en el sistema externo.
 - b. Invocar el método Update cada vez que se modifique el campo estado de un registro **X** en el sistema externo.
2. Implementar un Web Service que cumple la especificación ArandaExternalConnetor.wsdl (solo es necesario implementar el método IncidentOnchange)
 - a. Incluir en el método IncidentOnchange el código necesario para modificar el registro **X** en el sistema externo o crearlo si no existe.
3. Crear una regla de Incidentes:
 - a. Condición: **Status Change OR Estado = “Registrado”**
 - b. Acciones: **Invocar WebService**(parámetro: url del servicio del punto 2).