HEXAGON | X Xalt

XALT DATA CONNECTORS

Xalt's capabilities for mobile and cloud work as a cloud-based enterprise software platform that connects all your administrative database systems to your users on any PC browser or any mobile device. This part of our platform provides a flexible and secure environment for the creation and delivery of an unlimited number of business apps, utilising a single download from the Apple app store or Google play.

ARCHITECTURE OVERVIEW

Connectors are pieces of Java code that facilitate the data movement between your data sources and the Xalt Cloud. Connectors reside on the Connector Gateway (Figure 1 below), where they translate the requests issued by the Xalt Cloud into a format consumable by the data source, and then translate the response back into the format needed by the Xalt Cloud. You may think of them as language packs that you snap in when you want to converse with a new system. In Figure 1, the connectors invoke the communication lines between the Connector Gateway and the various data sources – ERP, CRM, Big Data, etc. The connectors are also responsible for gathering the metadata used in Xalt. In most cases, a single connector is responsible for a single language (such as SQL), but in some cases where the language does not

provide a programmatically efficient method of metadata discovery (such as REST web services), a connector must be custom built to the specifications of the data source.

The metadata gathered by the connector includes the data structures available from the data source, their properties, and the data types that make up those properties. Using a relational database as an example, the connector retrieves a list of tables or views, the columns that make up the tables, and the data types of each column. Aside from gathering metadata, a connector provides four basic data functions: create, read, update, and delete. These functions are the basic interfaces needed to allow Xalt to interact effectively with your data.
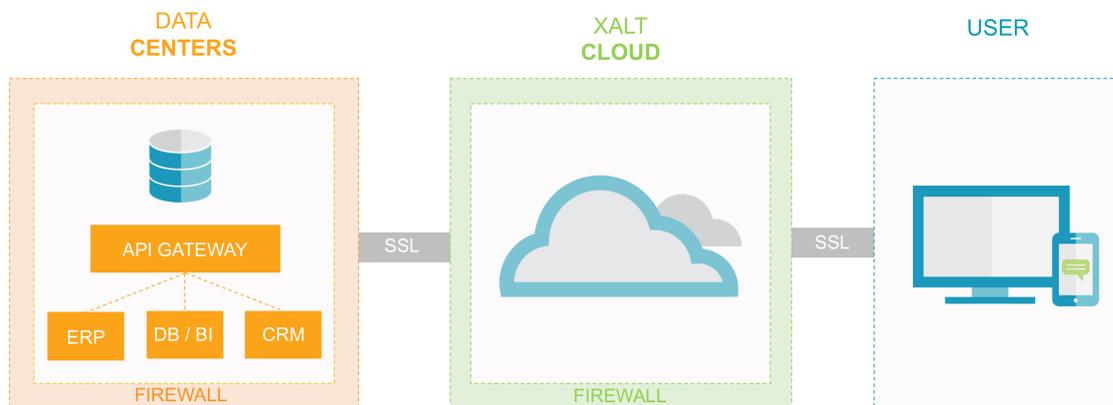
Figure 1

CONNECTING DIRECTLY TO A DATABASE

Xalt allows for direct connection to your database or data source in three ways:

1. SQL Direct via JDBC – this connection type supports Microsoft SQL Server, IBM DB2, Oracle, Postgresql, and MySQL.

2. IBM i Direct – this connection type supports IBM DB2 databases that reside on an IBM i Server.

3. Infor IDF System-Link – this connection type supports the Infor IDF System-Link API to access data. No direct connection is made; instead requests are sent to the IDF system via XML.

Further, the tight coupling to the database associated with these connectors means that access to stored procedures is available.

CONNECTING TO APIS AND WEB SERVICES

Xalt can also connect to databases via APIs and web services. We support the following basic API and/or web services standards: Hypertext Transfer Protocol (HTTP/HTTPS); eXtensible Markup Language (XML); JavaScript Object Notation (JSON); Simple Object Access Protocol (SOAP); web service Definition Language (WSDL), Representational State Transfer (REST), and Open Data Protocol (OData), among other common protocols. APIs or web services are declared as data sources in Xalt. There is no limit to the number of data sources that can be created in Xalt; therefore, there is no limit to the number of APIs or web services you can connect to.

For example, using web services, the WSDL is parsed to provide the metadata needed by Xalt to construct data objects, queries, and details from the data source. To provide the functions needed for create, read, update, and delete, a small configuration file is created and copied to the satellite server that maps the relationships between the function and the service call needed. This file is necessary, as there is no reliable way to infer the proper function to call based solely on the WSDL. An example configuration file for a single object is shown in Figure 2.

When the satellite is called upon to read data from a data source, it checks to see if it has the SOAP message format needed to make the call based on the object and function being requested. If it does not have it, the connector calls the WSDL file and retrieves the SOAP format needed. It then stores that format in memory for later use. The call is made and the results are returned to the cloud for distribution to the user. Similar scenarios also occur for creating or editing data. Actions give the user more flexibility, as they can call a web service not defined in the data source's WSDL by defining a custom call in the action.

The following are example systems that Xalt accesses via web services:

- SAP NetWeaver Gateway
- Oracle eBusiness Suite, JD Edwards, and RightNow
- Salesforce.com and all Force.com applications
- SocialCast

```
<config Name="getPurchaseOrder">
    <create>processPurchaseOrder</create>
    <read>getPurchaseOrder</read>
    <update>processPurchaseOrder</update>
    <delete>processPurchaseOrder</delete>
</config>
```

Figure 2

CONNECTING TO BUSINESS INTELLIGENCE

Xalt can also connect to multi-dimensional data cubes from many of the enterprise business intelligence systems via XML for Analytics (XML/A). Using XML/A, Xalt can interrogate and gather the meta data for a data cube that enables users to create dashboards and reports without the need to write any code. Reports can display aggregated data, cross-join any number of dimensions, and filter on any combination of dimensions, including date and time. Reports automatically generate Multi-Dimensional eXpressions (MDX) queries to fetch data, so no changes are needed in the data cube. Multi-dimensional data can be incorporated into applications along with any other data source.

The following are example business intelligence systems that support XML/A:

- Microsoft SQL Server Analytics Services
- SAP BI (BW release 3.0A or greater)
- Infor PM OLAP Server
- Pentaho

SUMMARY

By utilising the standard connector interfaces available, Xalt can connect to any database or web services of your choice. This model also stores all your business logic inside your datacenter so there is no need to alter business processes or duplicate authorisation lists. This allows Xalt to focus on delivering a rich, native interface to your users.