



Teiid

Scalable Information Integration

Noviembre 2010



Nicolás Escobar



**Facultad de
Ingeniería**

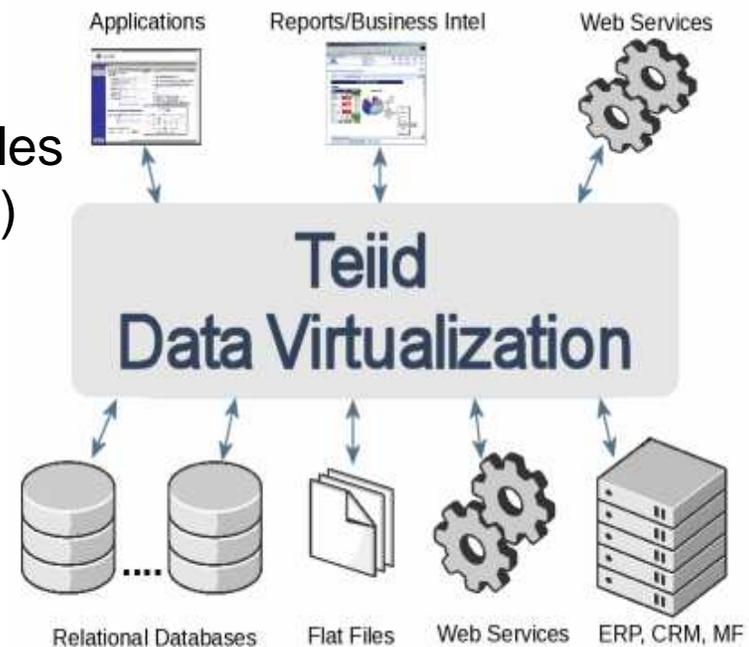


Agenda

- Qué es Teiid?
- Cómo se utiliza?
- Ejemplos:
 - Integración entre una base de datos relacional y un archivo
 - Utilización de Web Services como fuente de datos
- Ventajas
- Conclusiones

Qué es Teiid

- Sistema de virtualización de datos que permite a aplicaciones utilizar datos provenientes de fuentes heterogéneas
- Las fuentes pueden ser:
 - Bases de Datos relacionales
 - Archivos (txt, csv, xls, etc.)
 - Web Services
 - Creadas por uno mismo
- Teiid propone utilización de bases de datos virtuales (VDB's)

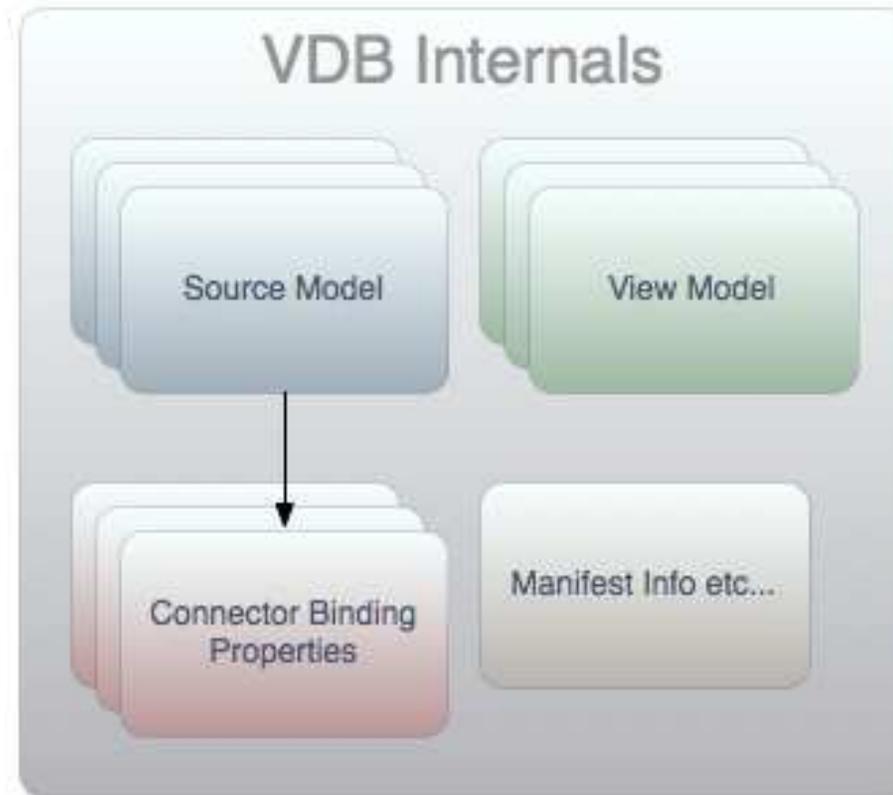




Base de datos virtual

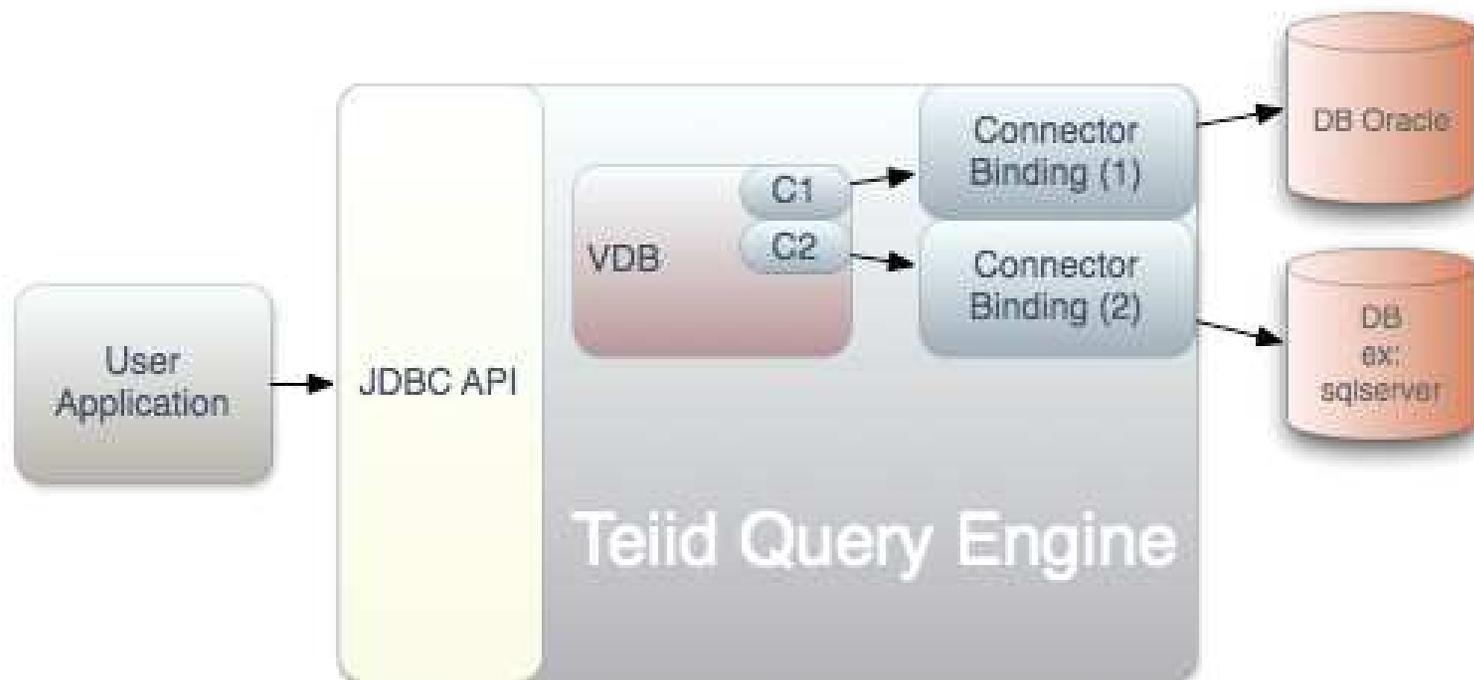
- Es un artefacto que combina una o más fuentes de datos físicas para brindar la integración de los datos contenidos en ellas [teiid]
- Se puede ver como contenedor de componentes utilizados para integrar datos de múltiples fuentes
- Contiene modelos que definen las características estructurales de las fuentes de datos, vistas y Web Services

Base de datos virtual



Base de datos virtual

- Una vez construida y puesta en funcionamiento la base de datos virtual, los datos pueden ser accedidos mediante una única JDBC API

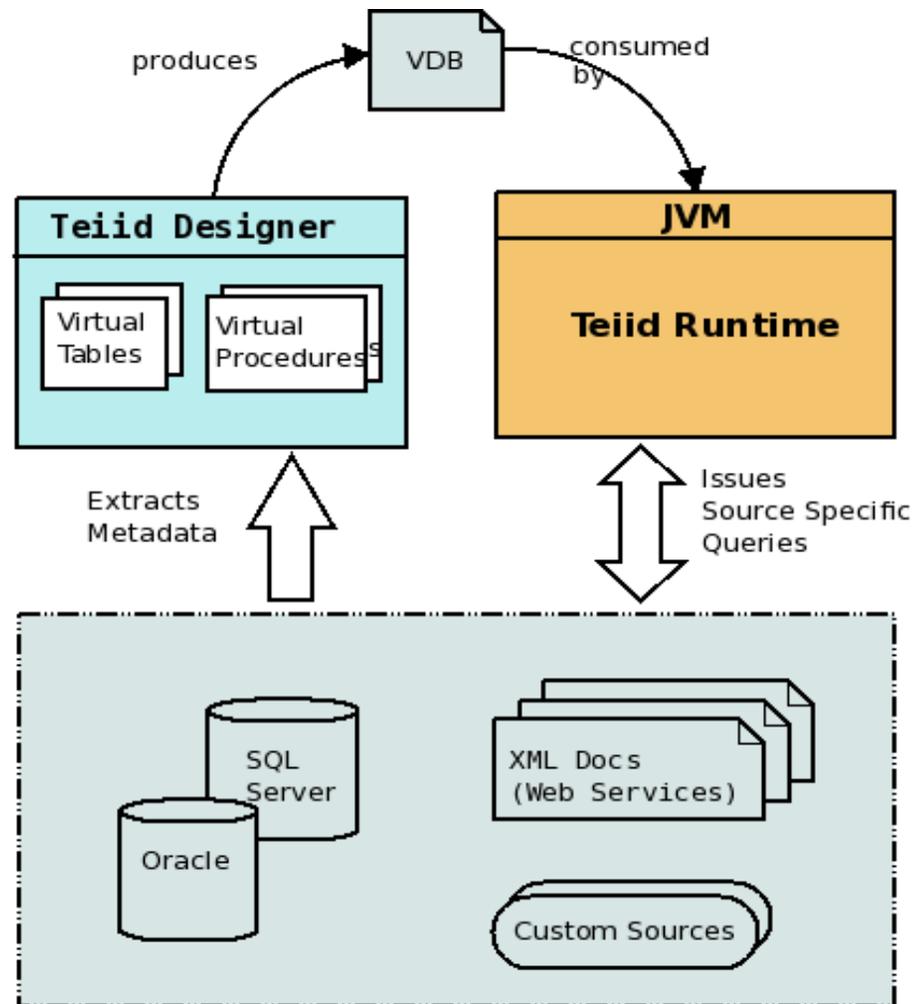




Cómo se utiliza?

- Construimos una VDB a partir de las fuentes de datos que disponemos
- Implementamos dicha VDB en el servidor de Teiid para que pueda ser accedida mediante una conexión JDBC
- Nos conectamos a la VDB desde una aplicación Java y ejecutamos las consultas que deseemos

Cómo se utiliza?





Ejemplos

- Veremos dos ejemplos utilizando construcción de VDB dinámicas, es decir, a partir de un archivo XML
- Se debe crear un archivo `<nombre>-vdb.xml` que contendrá información sobre la VDB, las fuentes que integra y las preferencias para importar la metadata
- El archivo detalla para cada esquema creado el nombre del mismo, tipo de traductor y otras propiedades
- Un traductor provee una capa de abstracción entre el motor de Teiid y las fuentes de datos físicas, que convierte las consultas de Teiid en comandos específicos de las fuentes y las ejecuta



Ejemplos

- Teiid brinda traductores predefinidos que definen fuentes de tipo: oracle, ws, archivo, ldap, sqlserver, etc.
- Debemos también crear los datasources que proveerán la información a la VDB, deben tener el siguiente formato de nombre: **<nombre>-ds.xml**
- Veamos los ejemplos

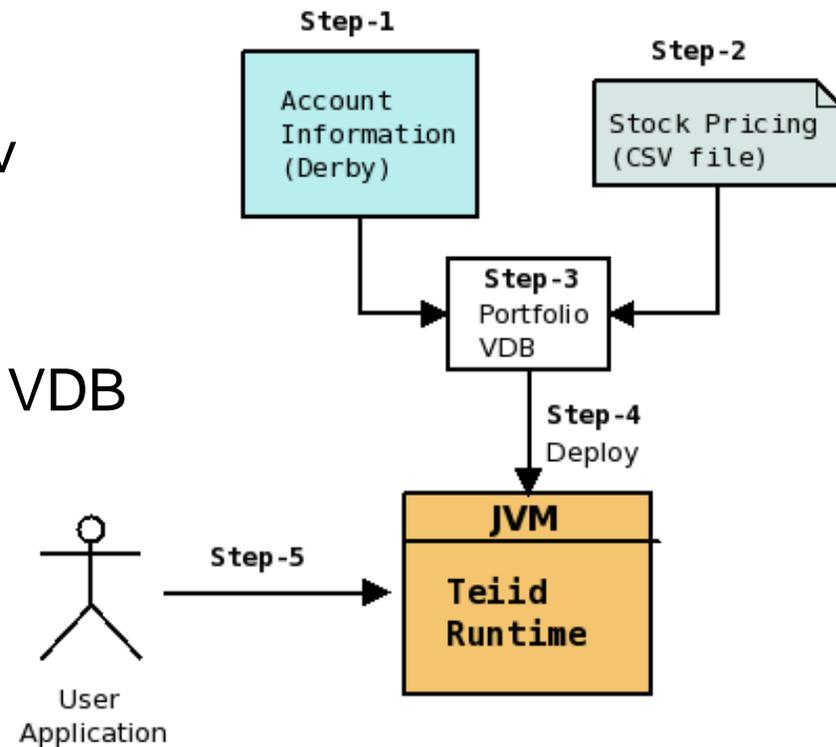
Ejemplo 1 - Inversores

- Integraremos una base de datos relacional Derby con un archivo csv
- BD Derby - contiene la información de los inversores
Las tablas son: clientes, cuentas, acciones y productos
- Archivo csv - contiene los precios actualizados del stock de productos
- Luego de construir la VDB, con una sola consulta podremos conocer los precios de los productos que posee acciones el inversor

Ejemplo 1 - Inversores

Pasos:

1. Crear la base de datos
2. Disponer del archivo csv
3. Construir la VDB
4. Implementar en Teiid la VDB
5. Acceder a la VDB



Ejemplo 1 - Inversores

- Ejemplo de consulta:

```
select stock.* from
  product,
  (call MarketData.getTextFiles ('*.txt')) f,
  TEXTTABLE (f.file COLUMNS symbol string, price bigdecimal HEADER) stock
where product.symbol=stock.symbol
```

- **MarketData** es el esquema asociado con la fuente de dato de tipo archivo de texto
- El procedimiento **getTextFiles** utiliza por detrás el traductor especificado en el esquema **MarketData** (de tipo texto) para acceder al archivo y devolverlo
- La función **TEXTTABLE** crea una tabla a partir del archivo, se debe especificar cuantas columnas se desean crear, sus nombres y tipo de datos





Ejemplo 2 - WS Clima



- Utilizaremos Web Services como fuentes de datos para consultar un servicio meteorológico
- Sitio web de la organización: <http://www.weather.gov>
- Brinda dos servicios, uno REST y otro SOAP
- Construiremos la VDB y luego ejecutaremos consultas para ambos servicios



Ejemplo 2 - WS Clima



Creación de la VDB:

1. Crearemos el archivo XML que representa la VDB, especificando un único esquema, de nombre **Weather** que tendrá un traductor de tipo ws
2. Definimos un único `datasource` en el cual se detalla como endpoint la URL correspondiente

Ejemplo 2 - WS Clima

- Ejemplo de consulta - REST

```
select t.* from
  (call weather.invoke(action=>'GET', endpoint=>querystring
    ('', '38.99,-77.02 39.70,-104.80 47.6,-122.30' as listLatLon,
    'time-series' as product, '2010-11-19T00:00:00' as "begin",
    '2010-11-21T00:00:00' as "end", 'maxt' as maxt, 'mint' as mint ))) w,
  XMLTABLE('/dwml/data/location' passing w.result
    columns "location-key" string, lattitude string path 'point/@latitude',
    longitude string path 'point/@longitude') t
```

- El procedimiento **invoke** utiliza por detrás el traductor especificado en el esquema **Weather** para realizar el pedido y devuelve el XML retornado por el ws
- La función **XMLTABLE** crea una tabla a partir del documento XML, se debe especificar cuantas columnas se desean crear, sus nombres y tipo de datos



Ejemplo 2 - WS Clima

- Ejemplo de consulta - SOAP

```
select xmlserialize(document w.result as string) from
(call weather.invoke(
  action=>' http://www.weather.gov/forecasts/xml/DWMLgen/wsd1/ndfdXML.wsd1#LatLonListZipCode' ,
  endpoint=>' http://www.weather.gov/forecasts/xml/SOAP_server/ndfdXMLserver.php' ,
  binding=>' SOAP11' ,
  request=>' <ns1:LatLonListZipCode xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://www.weather.gov/forecasts/xml/DWMLgen/wsd1/ndfdXML.wsd1">
<zipCodeList xsi:type="ns2:zipCodeListType"
xmlns:ns2="http://www.weather.gov/forecasts/xml/DWMLgen/schema/DWML.xsd">
63303</zipCodeList></ns1:LatLonListZipCode>' )) as w
```

- La función **XMLSERIALIZABLE** devuelve un string con el contenido del XML retornado por el ws



Ventajas

- Interfaz familiar para acceder a la VDB: JDBC
- Lenguaje de consulta familiar: SQL
- Múltiples fuentes de datos lucen como una única fuente
- El motor de consultas de Teiid es un componente Java, se integra fácilmente como cualquier otra librería
- Teiid Designer brinda una interfaz interesante para crear bases de datos virtuales y modelos de fuentes de datos
- Muchas otras que todavía no he experimentado, p.e. optimización, transacciones distribuidas, utilización de Teiid Designer, Query Planner, etc.



Conclusiones

- Teiid es una fuerte herramienta para integrar en tiempo real diferentes fuentes de datos, de esta manera podemos evitar el “uso de procesos batch” para la integración
- Permite tratar a fuentes de datos no relacionales como relacionales
- Con una herramienta de este tipo nos ahorramos el trabajo de integrar los datos por nuestra propia vía
- Existen pocos ejemplos, en mi opinión deberían haber más

Preguntas

