Data Virtualization at Nationwide
Agenda

- Background
- What is Virtual Data – Isn’t all data real?
- Virtual Data and the Architectural Fit
- Example Use Cases
- Must Do’s Before Implementation
• Many things drive the need to duplicate and or extract specific pieces of data from the authoritative data store.
  
  ➢ Per Forrester, one of the key requirements that will drive increased adoption of virtual data approaches is the tactical demand for better data management and data integration alternatives to ETL or DBMS consolidation.

  ▪ Data virtualization facilitates the abstraction, transformation, and federation of data from a variety of disparate sources.

  ▪ Data is presented via a single access point to a consumer regardless of the physical location or nature of the various data sources. Some benefits include:
    ❑ Greater flexibility and agility due to faster creation of virtual data stores.
    ❑ Improved data quality because we reduce the number of physical copies, eliminating redundant data stores.
    ❑ Improved usage through creation of subject-oriented, business-friendly data objects.

• Once we begin duplicating data, without a strong governance structure, we tend to lose control of the data and its meaning.
Oh what a mess we make

A look at traditional data movement
Data Virtualization – What is it?

Data virtualization is the process of abstracting, transforming, federating and delivering data contained within a variety of information sources so that they may be accessed by a consuming application or users when requested without regard to their physical storage or heterogeneous structure.

This concept and software is commonly used within data integration, business intelligence, service-oriented architecture data services, cloud computing, enterprise search and master data management.

Key Characteristic

It keeps the data where it belongs (persisting data in it’s native data store as opposed to relocating it). It creates a façade and exposes it to consumers, but doesn’t move it. It can be used in read-only or read/write scenarios.

So, how is this different than a SOA implementation?
Integration Landscape

High Level Integration Implementation Patterns

I am a consumer application with a need to integrate with another application

- Point to Point Application Integration
  - Application integration that occurs between two applications, but is not reusable across other applications

- Enterprise Application Integration
  - Application integration that is reusable across business channels, has flexibility in the message, and whose implementation is governed

- Enterprise SOA Business Service
  - Formally modeled business message that is abstracted from the provider, is reusable across business channels and is governed

- ETL
  - Moves/Copies data from one source to another
    - Batch
    - Focused on historical data
    - Vary large volume
    - Pull, query-driven, scheduled
    - Metadata driven, complex flows
    - Strong profiling and cleansing capabilities
    - Limited Transaction and Messaging capabilities

- CDC
  - Moves/Copies data from one source to another
    - Near real-time
    - Transactional
    - Event driven
    - Limited transformation capability (more than EAI, less than ETL)
    - No impact to source; uses database log files to generate

- Virtualization
  - Leverages data at its source; does not move
    - Scales from transactional to large volumes
    - Abstracts consumer from source (like EAI)
    - Complex transformation, integration, profiling, cleansing capabilities
    - Flexible delivery; data source or service
    - Caching and query optimization
Data Integration and Application Integration

Data Virtualization tools are Data Integration tools that provide functionality similar to Enterprise Application Integration.

That overlap is the gray area where confusion arises around when to use which integration tool.

Other tools in this gray area are Change Data Capture and Informatica Web Service Gateway.

Because of this gray area, it’s difficult to provide clear cut prescriptive guidance.
Example Use Case – Single Underwriting Desktop

- Quick delivery timelines: POC was conducted in 4 days
- Multiple data source types: IMS, Teradata, DB2, etc.
- Some large volume data sources (ie, Teradata)
- Users hypersensitive to response time challenges
- 11 Data Sources
- 2-3 Attributes per source required
- Read-Only requirement
At least four of the five integration points listed were definitive candidates for data virtualization. The 5th source would be dependent on negotiation of non-functional requirements around availability.
Example Use Case – Business Intelligence

Ad-Hoc Reporting

Web Service

Data Virtualization Platform

Teradata Data Mart
8 Million Rows

UDB Agent Hierarchy
60 Thousand Rows

Excel NF Data
50 Thousand Rows

Website Contact Info
# Performance Was Met or Exceeded

<table>
<thead>
<tr>
<th>Category</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Windows 2008 R2 Standard</td>
</tr>
<tr>
<td>Processor</td>
<td>inter® Xeon® CPU L5530 ® 2.40GHz 2.40GHz (2 processors)</td>
</tr>
<tr>
<td>Installed Memory</td>
<td>3.00 GBS</td>
</tr>
<tr>
<td>System Type</td>
<td>64-bit Operating System</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rows processed</th>
<th>Execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td>join: 142,162,589; projection: 141,287,438</td>
<td>1 hour and 21 minutes and 4 seconds.</td>
</tr>
<tr>
<td>projection: 237,411,854</td>
<td>3 hours and 51 minutes</td>
</tr>
<tr>
<td>join: 8,383,234; projection: 5,102,815</td>
<td>10 minutes and 53 seconds.</td>
</tr>
<tr>
<td>4,297,869</td>
<td>3 minutes</td>
</tr>
<tr>
<td>128,130,455</td>
<td>1 hour and 25 minutes</td>
</tr>
<tr>
<td>Netezza: 260,634 ; Teradata: 4358057 ; join: 115319 ; projection: 115319</td>
<td>2 minutes and 23 seconds.</td>
</tr>
<tr>
<td>Netezza: 2309123; Teradata: 7885758 ; join: 82524 ; Projection: 82524</td>
<td>2 minutes and 58 seconds.</td>
</tr>
</tbody>
</table>
Must Do’s Prior to Implementation – Governance

• Very difficult to provide definitive prescriptive guidance

• Characteristics from earlier should drive general direction, but each case will have to be evaluated for “right fit”

• Must understand roles and responsibilities around source access – this can cause performance issues on the source if not careful! This needs to be balanced with Guiding Principles around “Data is Accessible”.

• Writes – when should this be allowed and when not. Very powerful, but need to understand possible implication to source business rules