



IBM Italy - Innovation Lab

# Data access issues in a distributed federated system

BIOINFOGRID Initial training course  
Bari, March 8 - 10 2006

G. Scioscia, G. Pappadà, V. Quinto

Bari, March 10 2006

# Agenda

- Data access issues in a distributed federated system (G. Scioscia)
- Real case study: federated access to three different bioinformatics databases (G. Pappadà)
- Solution for the real case study using WebSphere Information Integrator (V. Quinto)



IBM Italy - Innovation Lab

## Data access issues in a distributed federated system

G. Scioscia

## Issues Concerning Data Integration

- An increasing number of grid applications manage data at very large scales of both size and distribution.
- The complexity of data management on a grid arises from the **scale**, **dynamism**, **autonomy**, **heterogeneity** and **distribution** of data sources.

### Mission to accomplish:

- These complexities should be made transparent to grid applications through a layer that enables ease of data access and processing, dynamic migration of data for workload balancing, parallel data processing, and collaboration.

### Viable approaches:

- **Data Federation**: data are logically integrated
- **Data Warehouse**: data are physically integrated

## Data Federation vs Data Warehouse

- Data warehousing ‘cleaning up’ data and placing it into a centralized repository works well in situations where **data are relatively static** and data types are not too diverse
- Moving data into a warehouse can **limit the specialized search** capabilities available with (through) the original data source.
- Building and maintaining enterprise wide warehouse on the scale required by most large research organization with hundreds of data sources can be both **costly and risky to implement**.
- Data warehouse centralization clash with the basic grid-concepts of data **replication** and **distribution** according to monitored statistics
- Data federation allow to access **current data** from **multiple, heterogeneous, dislocated** data sources **simultaneously**, with a single query



**For bioinformatics problem Data Federation seems the most promising solution**

## Overview of IBM DB2 Information Integrator

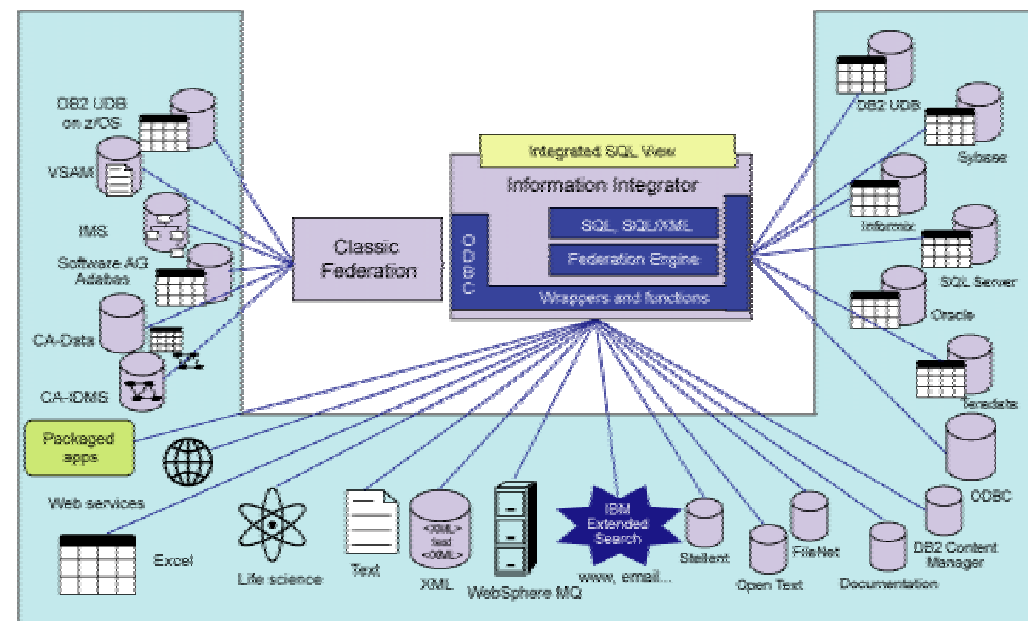
The data-federation capabilities of Information Integrator allow multiple, heterogeneous data sources to be accessed as if they were a single data source, regardless of where they reside.

### A federated system consists of:

- A DB2® instance that operates as a federated server
- A database that acts as the federated database
- One or more data sources
- Clients (users and applications) that access the database and data sources

**Key concept:**  
 DB2 II interacts with different data sources by means of specific wrappers. Each wrapper encapsulates a comprehensive knowledge about the source it is constructed for.

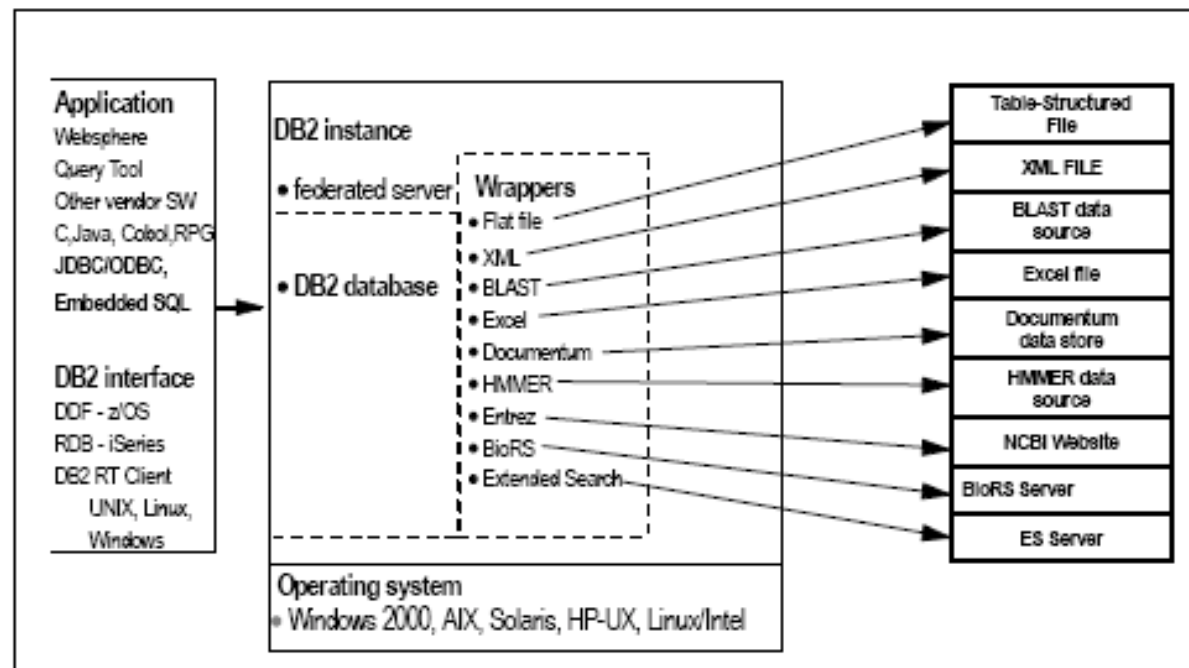
With a federated system, you can send distributed requests to multiple **data sources** and **processes** within a single SQL statement.



## DB2 Information Integrator and Life Sciences

Life Sciences is an all new world of disparate data and applications, different from relational, with a tremendous need for integration.

- DB2 Information Integration enables a DB2 federated system to integrate genetic, chemical, biological, and other research data from distributed sources.
- DB2 Information Integration enables to federate Life Sciences services and processes as well (Blast, Entrez, BioRS query system, HMMER, the Kyoto Encyclopedia of Genes and Genomes, etc.)
- A Java wrapper SDK to develop wrappers for specific, custom data sources is also available



## Data Handling in Data-Grid

The key concept for data grid is **virtualization**: data location, data access methods and data sources should be all transparent to the application.



Unfortunately, this isn't the case in all current implementations...

### What is OGSA-DAI?

- OGSA-DAI is grid middleware designed to make it easy to access data in a data grid.
- It exposes data resources through Web services-based interfaces
- It allows metadata about data and the data sources to be queried via a service interface.

### Grid's User Requirements

Present requirements emerge calling for location and heterogeneity transparency that enable full dynamic choices of data sources. These transparencies are not part of either today's current federation technologies or OGSA-DAI middleware.

- - **Information federation across Institutions**
- - **Failover access** (The grid middleware should autonomically fail over to either another data resource with a similar data set or to multiple data resources each with a subset of the data.)
- - **Large dataset access** (more efficient ways to fetch data in parallel from large data sources)
- - **High level of access through SQL**

## The Need for a Grid-Wrapper for DB2 Information Integrator

So far, we have established:

- DB2 Information Integrator provides federation capabilities.
- OGSA-DAI provides transparent data access to grid data sources (at least for relational databases, XML data sources and files).
- A data-information grid provides data virtualization by cleanly abstracting application access to, provision of, and administration of data.
- Real users and applications need a grid wrapper to bring data federation (WebSphere Information Integrator) and uniform data access (OGSA-DAI) into a data-information grid.

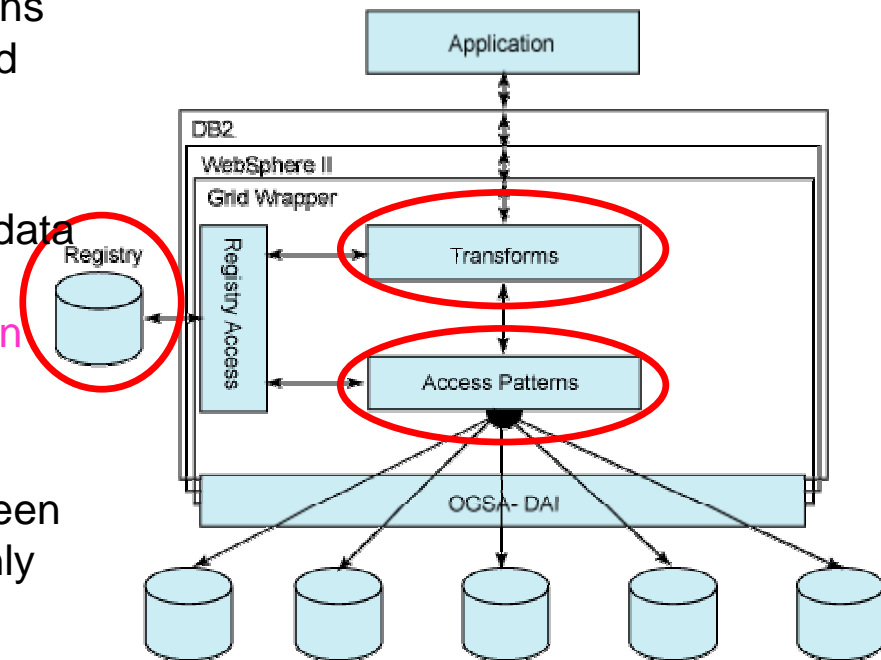
### A viable solution:

- Putting DB2 in front of relational and non-relational data sources is one way to "unify" the access language, as long as the query transformation and optimization: **DB2 Information Integrator is well suited for this task!**
- Coupling the dynamic aspect of the grid with DB2 Information Integrator means that DB2 Information Integrator can get greater flexibility, and provide the schema integration and query transformation needed to help grid application developers.
- A nice side effect of this proposal is that current DB2 applications could actually be run against a data grid without any modifications.

## Architecting the Grid Wrapper

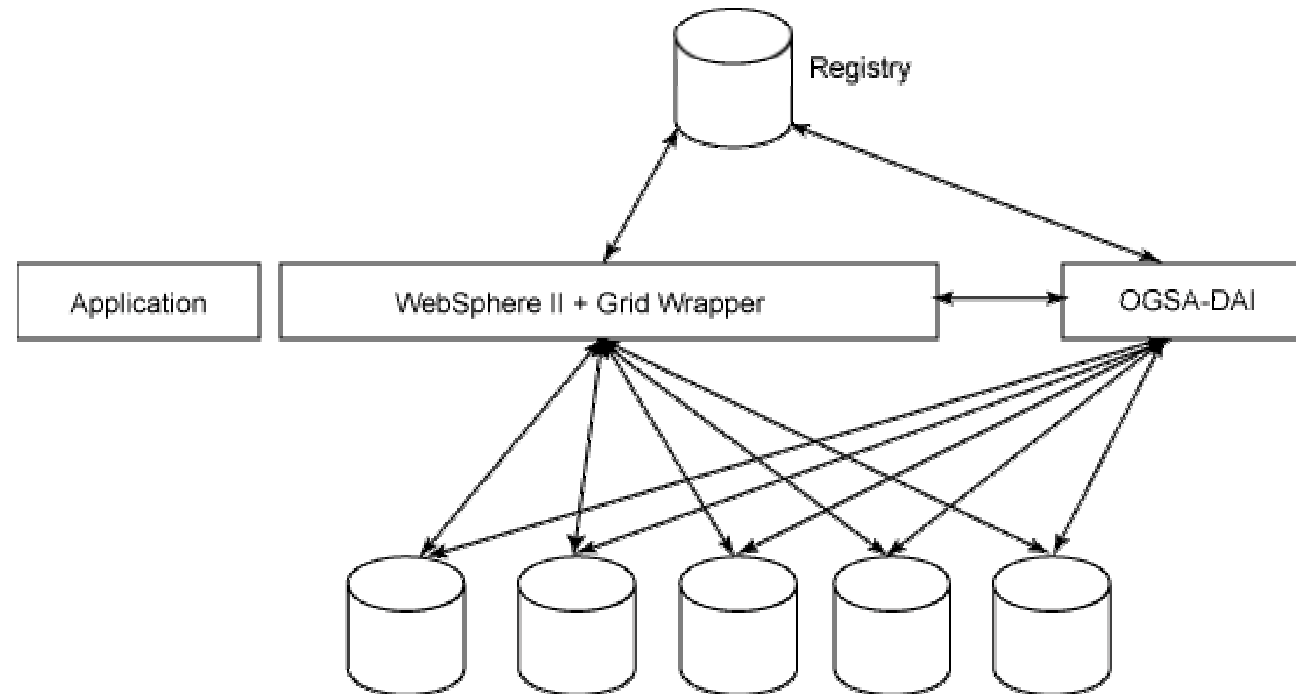
The grid wrapper high-level architecture highlights three key components to bridge the gap from DB2 Information Integrator to OGSA-DAI:

- The **registry**, a facility that supports loose coupling and late binding of data sources to data source representations in a transparent manner. It contains metadata about how and which data are federated (**location transparency**).
- **Access patterns** encapsulate the semantics of data sources and the way in which they have to be accessed to satisfy a given query (**data distribution transparency**).
- **Transforms** abstract away the differences between data sources, allowing applications to consider only one query model (**data management system transparency**).



## Separating Grid from Application Functionality

- The Grid Wrapper removes the tight coupling of DB2 Information Integrator to the data sources and enables late binding.
- DB2 Information Integrator becomes the application's gateway to the data grid.
- The Grid Wrapper can use an OGSA-DAI data source registry as its metadata repository.



## How does DB2 II Grid Wrapper works?

A registry enables the dynamic aspects of the grid to be fully used.

### Example:

By getting metadata information from a registry (XML, relational, or OGSA-DAI), the wrapper will make decisions to fetch the data from the proper database or databases. This is totally transparent to the DB2 user or application.

For instance, before the wrapper, the query looks like this: "select \* from mytable." After the wrapper is installed, nicknames created on *mytable*, and the registry loaded with information such as "mytable is in database 1 but also replicated in database 2 and 3", the wrapper can make the decision at run time, just before shipping the query to the remote database in order to actually ship it to Database 1 by default. If Database 1 is down (an error code comes back) the wrapper can automatically try Databases 2 and 3. The wrapper could also decide to fetch data from both databases if the table is partitioned on these two databases.



IBM Italy - Innovation Lab

## Real case study: federated access to three different bioinformatics databases




G. Pappadà

## Real case study

### GOAL

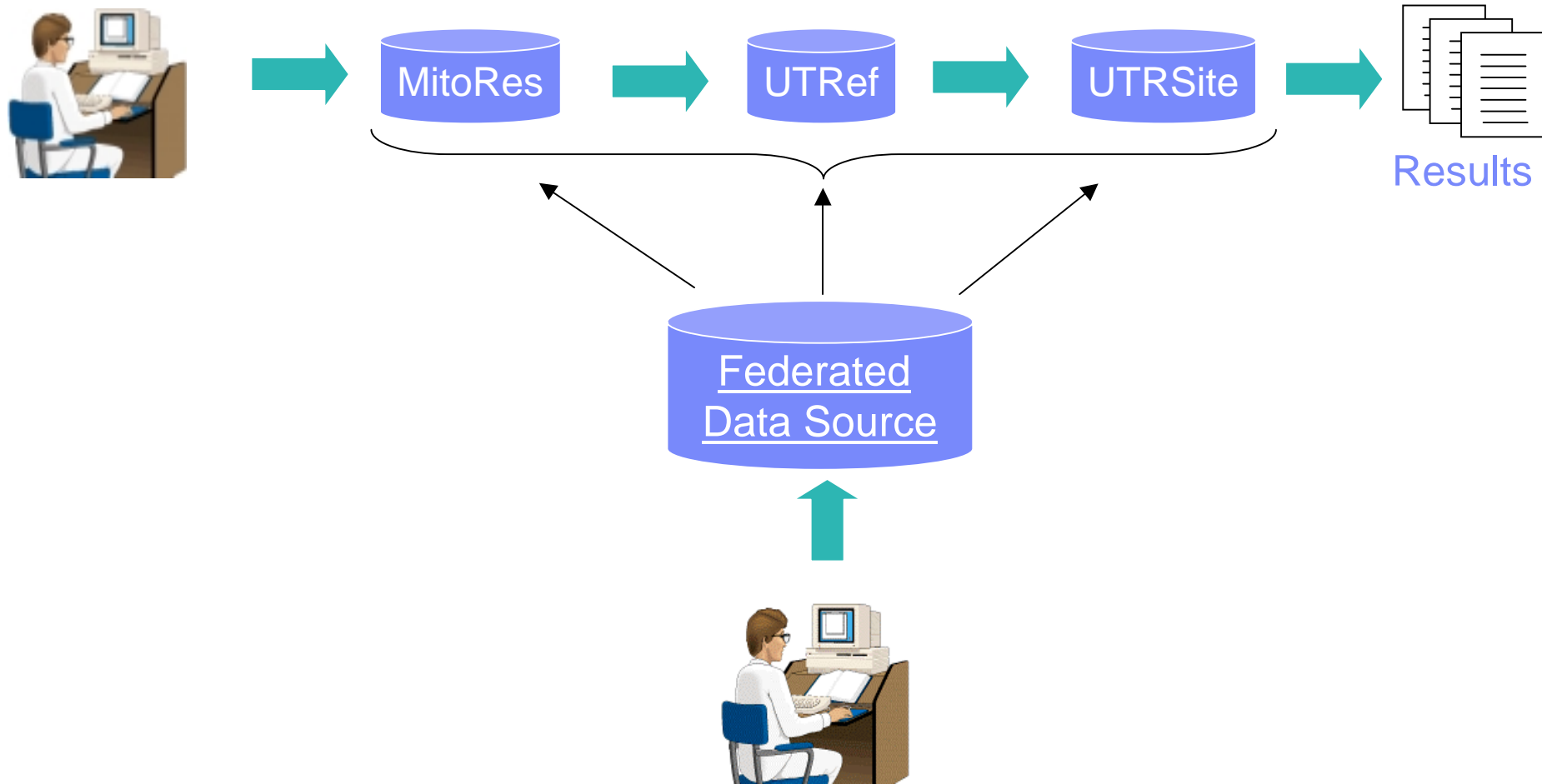
A biologist wants to search for specific information associated to a certain protein; pieces of information are located in different DBs

### STEPS

- He searches a specific database ( MitoRes) for the name of the protein ([topoisomerase](#)) and for the name of the species he is interested in ([Homo sapiens](#))
- Then the biologist analyzes the results of his query to find useful information (e.g. links to other databases)
- He has to jump from one database to another ( UTREFdb,  UTRSite) in order to gain information he needs.

Here is how a biologist should work to gain the information about his protein using the [MitoRes Search Interface](#)

# Real case study (summary)



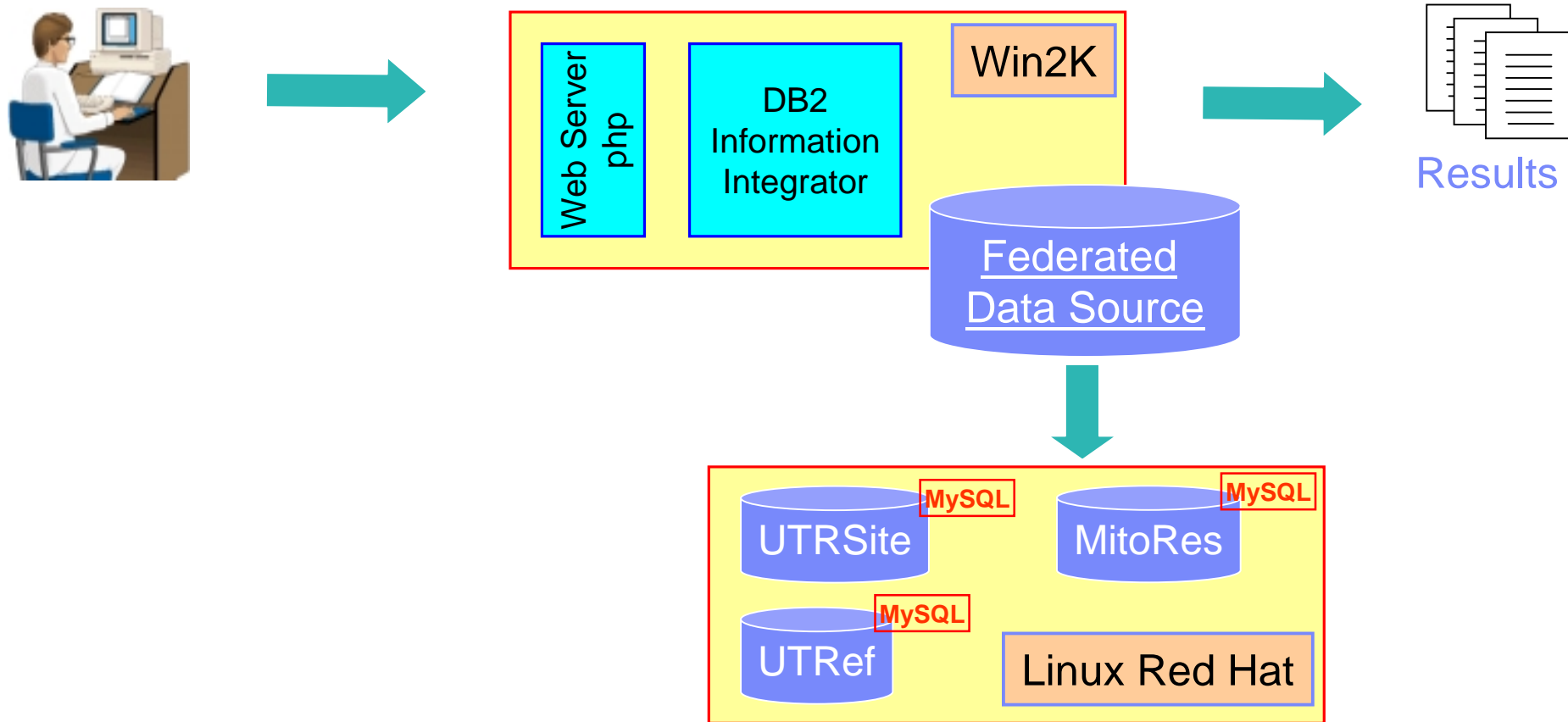


IBM Italy - Innovation Lab

## Solution for the real case study using WebSphere Information Integrator

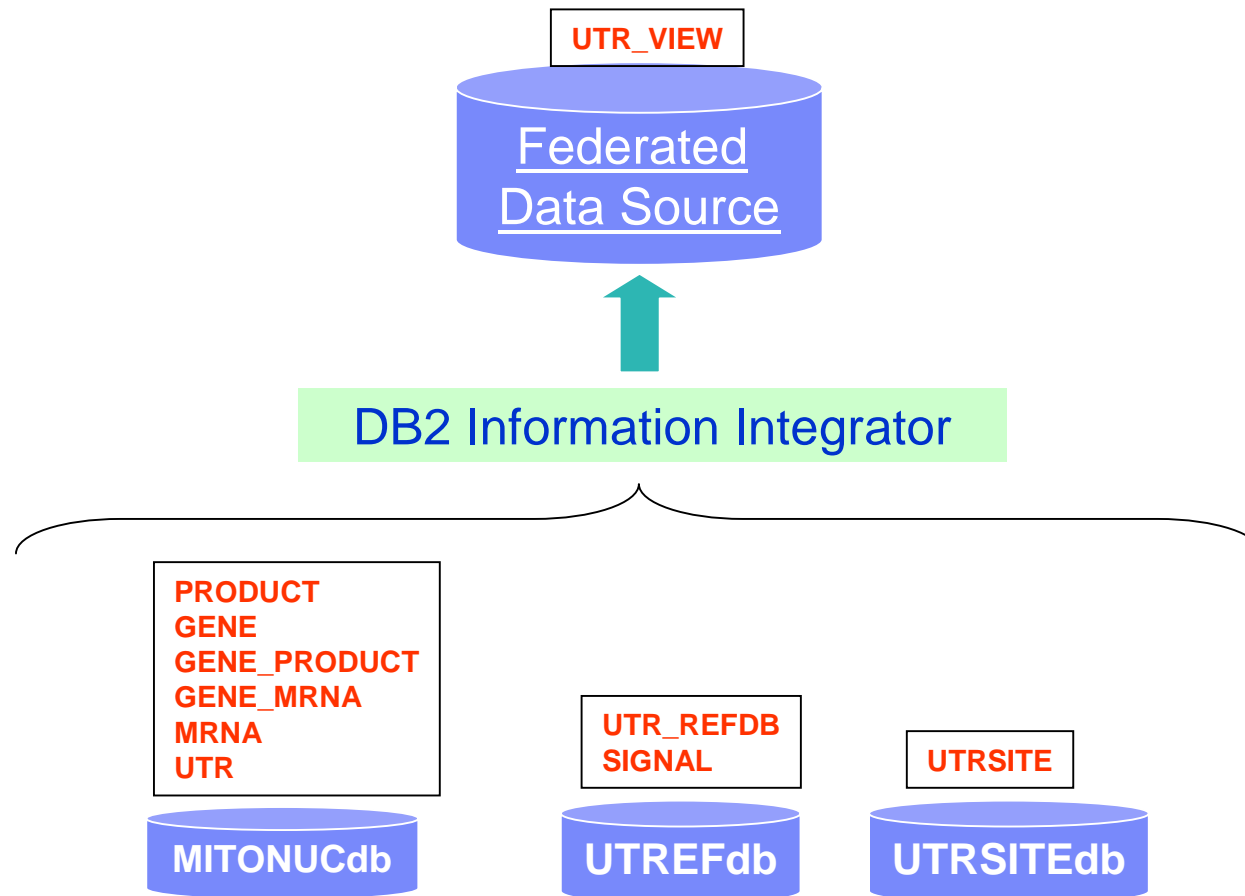
V. Quinto

# Real case study (System Topology)



**DEMO:** <http://9.87.104.199/>

# Solution for the real case study



## References & Contacts

- DB2 Information Integrator (re-branded as WebSphere Information Integrator)  
(<http://www-306.ibm.com/software/data/integration/>)
- Information Integrator Grid Wrapper  
(<http://www.alphaworks.ibm.com/tech/gridwrapper>)  
(<http://www-128.ibm.com/developerworks/grid/library/gr-feddata/>)
- IBM Academic Initiative  
(<http://www-304.ibm.com/jct09002c/university/scholars/>)

### Emerging Technologies Team's contacts:

- Pietro Leo ([pietro\\_leo@it.ibm.com](mailto:pietro_leo@it.ibm.com))
- Gaetano Scioscia ([g\\_scioscia@it.ibm.com](mailto:g_scioscia@it.ibm.com))