

SCENARIOS FOR METAMATRIX

- IT-structure today..... 3
- Enterprise Data store pattern..... 3
- Common data issues with the organization..... 5
- ..... 6
- Factors if there is a need to install a 'New Application'..... 6
  - ... The possible needs of a 'New Application' ..... 6
  - ... Result after deciding to have a 'New Application'..... 6
  - ... How is this single view to be achieved..... 6
  - ... What are the gaps..... 7
  - ... Outcome of above gaps ..... 8
- What the most organizations are looking for ..... 8
- ..... 8
- A typical data-definition gap..... 8
  - ... Data type, semantics, & vocabulary ..... 9
  - ... The gap between different data types of data structure..... 9
- ..... 9
- The need for Service Oriented Architecture (SOA)..... 9
- What is 'Service Oriented Architecture' ..... 10
  - ... Example of an 'Internet store-front' ..... 10
  - ... But why is this so important ..... 10
  - ... Let us look at the same example once again ..... 11
- ..... 11
- A possible MetaMatrix use case..... 11
  - ... Two companies merged that uses different databases ..... 11
  - ... Decisions made in the IT integration plan by the new CIO ..... 12
  - ... List of typical problems that the new company would face ..... 12
  - ... MetaMatrix steps that you would follow ..... 12
    - ..... Once all applications have been migrated ..... 13
    - ..... MetaMatrix has more features that stated above ..... 13
- ..... 13
- MetaMatrix is the ultimate solution..... 13

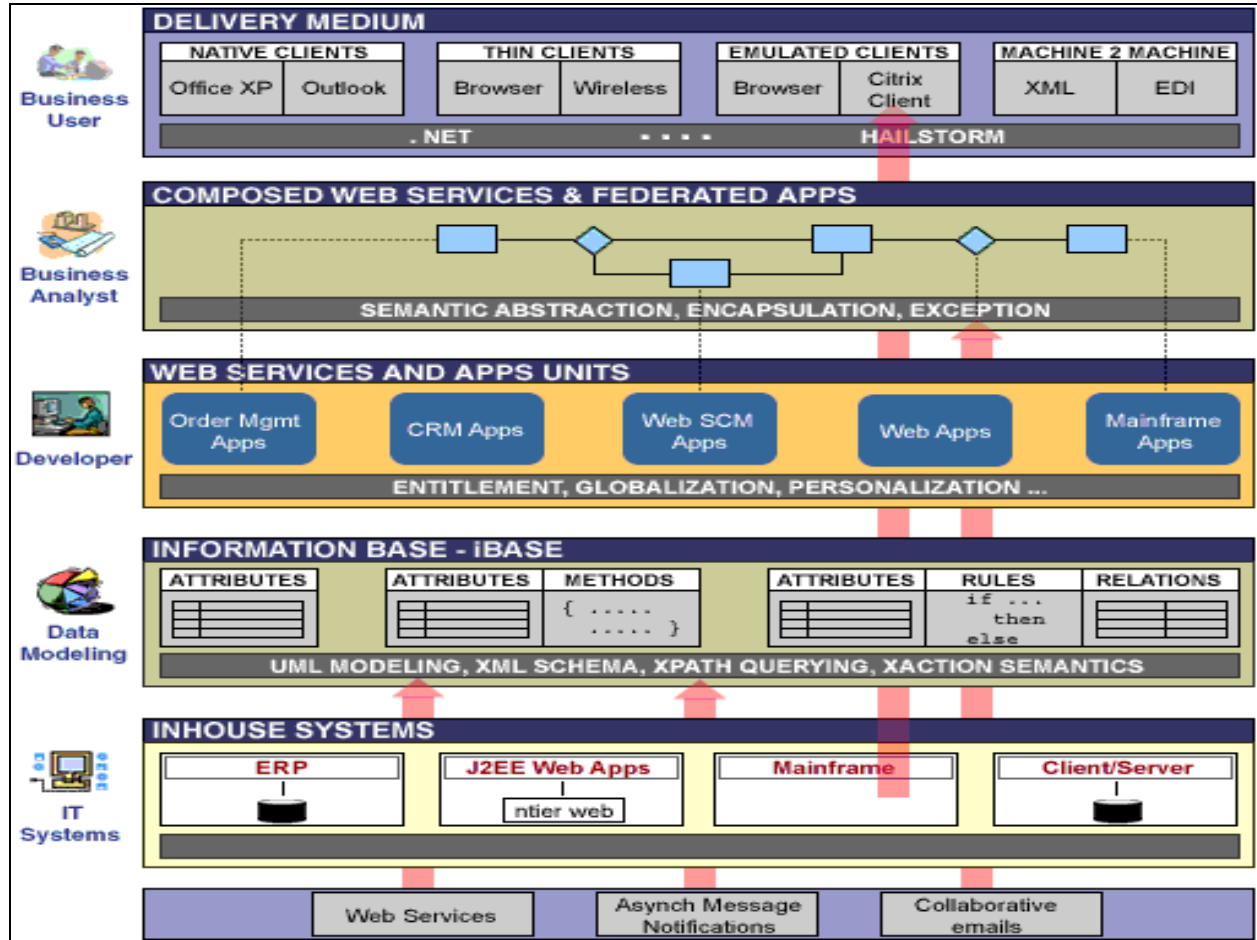
**SCENARIOS FOR METAMATRIX**

- ... MetaMatrix Enterprise..... 14
- ... MetaMatrix Dimension ..... 14
- ... MetaMatrix Query ..... 14
- ... Comparison of MetaMatrix offered tools..... 14
- ... How MetaMatrix Server works..... 15
- ... Benefits of using MetaMatrix server..... 16
- ... Need of MetaMatrix for an Enterprise..... 16
- ... How MetaMatrix works ..... 17
- The growth of SOA.....**Error! Bookmark not defined.**

## SCENARIOS FOR METAMATRIX

### IT-structure today

Many organizations are using multiple databases with different versions that too at remote locations on various operating systems.



### Enterprise Data store pattern

- Hundreds of Relational Databases
- Hundreds of Tables in each Relational Database
- Dozens of Relationships in each Relational Database
- Dozens of Columns in each Table

## SCENARIOS FOR METAMATRIX

The database data is often:

- Shared - multiple applications may be using the same data
- Distributed - data is spread across more than one locations
- Disparate - related data is managed at different places
- Dynamic - content & its structure undergo changes
- Valuable - often critically important for the enterprise

Also the business processes requires data from many isolated sources developed using varying technologies for decades.

Common issues are:

- How to integrate all this data in a usable manner
- How to describe the desired integration
- How to decouple applications from data sources & bring changes in those sources

To answer many of above issues, most corporate data is currently stored in:

- Departmental relational databases
- Data warehouse
- Separate instance for reporting

But all the above solutions have only been optimized for the known 'lines of businesses' or business functions. Also there is a huge team:

- Taking backups
- Maintaining data integrity
- Resolving issues with performance
- Ensuring its availability

Under such scenario & the given complex data organization – what if we decide to bring global changes to the database nomenclature across the organization?

**SCENARIOS FOR METAMATRIX**

**Common data issues with the organization**

In addition to the complexity of business, following are the challenges with 'Management of Data':

- Variable naming conventions

Possible solutions	Issues, if any
<ul style="list-style-type: none"> <li>• Change the entire software code to have proposed naming convention</li> <li>• Source code needs to be retested to ensure the consistency</li> </ul>	<ul style="list-style-type: none"> <li>• Process is time consuming</li> <li>• Testing do not ensure that the new code 100% error-free</li> </ul>

- File structure

Possible solutions	Issues, if any
<ul style="list-style-type: none"> <li>• Keep the data using common types, formats</li> <li>• Use one operating system to avoid data truncation</li> </ul>	<ul style="list-style-type: none"> <li>• Have to install a tool that can convert all possible data types</li> <li>• Maintain code &amp; train staff for forward compatibility</li> </ul>

- Data confidentiality

Possible solutions	Issues, if any
<ul style="list-style-type: none"> <li>• Code the data before sharing</li> <li>• Decode it back before using</li> </ul>	<ul style="list-style-type: none"> <li>• Design a full proof methodology for data backup &amp; coding-decoding</li> <li>• Again this process is time consuming &amp; needs a through testing</li> </ul>

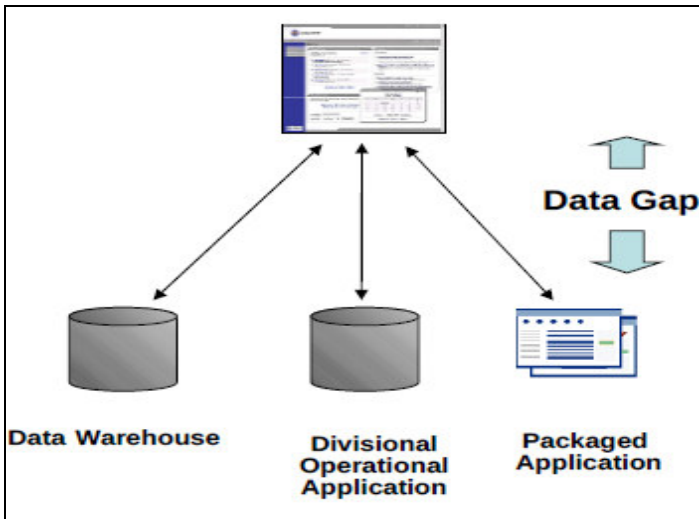
- Data imputation

Possible solutions	Issues, if any
<ul style="list-style-type: none"> <li>• Improve data quality by roll-forward where problems are known</li> <li>• Address issues on case-to-case basis as they arose</li> </ul>	<ul style="list-style-type: none"> <li>• How to certify that problems that exists with false-positives will not recur</li> <li>• Requires skilled people, time consuming &amp; needs a through testing</li> </ul>

\*\* But none of above solutions 'guarantee of successes'

## SCENARIOS FOR METAMATRIX

### Factors if there is a need to install a 'New Application'



When a new application is created, it often uses one or more existing legacy systems or existing data sources. Also, the new application may expect data in a format different than the way it is stored. Consider, e.g., the developer/integrator requires data to be gathered from several existing systems to create a "single view of the customer."

### ... The possible needs of a 'New Application'

- Data across business lines
- Data access to ensure efficient delivery over the network
- Data from diverse sources for the consolidated view

### ... Result after deciding to have a 'New Application'

To plan for extensive custom coding efforts to transform & integrate data and the high volume of work requires:

- Extended development cycles to have a standardize data
- Labor-intensive efforts by skilled people to create a solution
- A team to maintain the solution that is often complex & difficult
- & so on

In addition to above, the use of many different data sources places an undue burden on a stable running application, ensuring efficient delivery over the network for making business decisions.

### ... How is this single view to be achieved

## SCENARIOS FOR METAMATRIX

To answer this question, consider that over the decades, designers and developers have evolved efficient ways of working with data. Initially data was stored in “master files” separately from the application on punched cards, tape, disk storage & finally relational access structures. But each innovation in data manipulation had core design imperatives like maintain data integrity and maximize efficiency of storage & retrieval.

Suppose that today a developer creates a new application, Application A, that needs to consolidate data from three data sources B, C & D. Traditionally there are two ways of solving this problem:

1. Create a new data store that contains the data, pumped from the existing sources through a transformation process & stored in a form tailored for Application A
2. Connect Application A directly to data sources B, C, and D so that it can integrate the appropriate data dynamically

The **1<sup>st</sup> approach**, creating a new data store, results in some overhead in storage and management, and, more importantly, creates yet another redundant data store.

Reluctant to create & manage more data stores, many prefer the **2<sup>nd</sup> option** — connecting applications to multiple data sources so they can obtain data dynamically. The challenge is how and where to do it. Many application teams are still burying data access and integration logic in application code, where it is difficult to reuse and maintain.

In an effort to make this process efficient and scalable, forward-thinking architects are evolving a **3<sup>rd</sup> alternative** — connecting the application to the various data sources through a layer of data services.

### ... What are the gaps

Following are the major 3 factors that must be considered:

- **More & more data**

Organizations are managing more data than ever before plus there is a tremendous growth in the size & types of data sources, both inside & outside the organization. The sheer volume & wide distribution of data sources are increasing security concerns and at the same time access to the data must be controlled & audited.

Today's organizations need real-time information at all levels for making business decisions in terms of performance that will help them compete and at the same time, there is the growing cost of data management is fostering renewed interest in reducing & controlling data redundancy.

- **The growing need for consolidated, consistent views of information**

## SCENARIOS FOR METAMATRIX

The consolidated views of information are driving companies to grapple with the master data management problem. Not only must data be brought together, but it must also be consistent across systems, product lines, and divisions.

E.g., a high level of customer service is not possible unless all relevant information available in real time. Projects that focus on reference data management or customer data integration (CDI) lead inevitably to discussions about how to create a single view of the truth that can be shared by all relevant systems.

- **The drive toward interoperability & standards**

Interoperability goals, regulatory requirements, and the sheer complexity of today's information are forcing adoption of standards. The gap between data source schemas and new formats contributes to an increase in the number of data transformation and integration projects.

- **The data access gap**

Each system may access data in a different way. Relational databases require SQL and specific or general relational database drivers. Spreadsheets are read from a file system, as are certain other file-based data sources. Data feeds may come through publish-and-subscribe systems; some systems may use Web services and XML-based data, while others still might require data access through a URL. Diverse security protocols make the problem even more complex.

### ... Outcome of above gaps

Application developers, who generally prefer to focus their attention on the logic, are instead spending time resolving the logistical issues of data access.

### What the most organizations are looking for

They need a system that not only functions smoothly under such circumstances but also caters to their unforeseen future business needs effectively to make accurate timely decisions. Business decision can only be made quicker if the data can be viewed online in the form of reports even in the most complex data management structure.

When a new application is created, it often uses one or more existing legacy systems or existing data sources. Also, the new application may expect data in a format different than the way it is stored. Consider, the developer/integrator requires data to be gathered from several existing systems to create a "single view of the customer."

### A typical data-definition gap

## SCENARIOS FOR METAMATRIX

### ... Data type, semantics, & vocabulary

Each database has typically been designed to serve the needs of one application. Hence, another application needing data from the created database is often true. E.g. the CUST\_ID defined in one system is altogether different with CUST\_NO defined in another system. Combining the two will require an analyst to resolve differences in length, data type, name, and other aspects of the definition.

A project of this nature generates an opportunity to create/ use a canonical "Customer Identifier" element. Here the developers must address semantic differences between systems like who is a 'Customer'. E.g. one who has directly purchased the company's products is a 'Customer' in one-system, while a reseller is called a Distributor. Another system might include both Customers & Distributors under 'Customer'.

Resolving such differences requires attention from someone who understands the meaning embedded in the systems. As a result, this work can be time-consuming and expensive. In contrast to this ideal approach, typically too much re-invention of the wheel occurs, with developers unable to know about or build upon work that has gone before.

### ... The gap between different data types of data structure

To further complicate matters, the basic data structures of sources and target are not always as similar as they would be in. Also with the widespread use of XML & Web services, XML-based standards are driving standards for a number of different industries and system types in order to promote interoperability.

But in reality is:

- the bulk of existing databases are relational
- with some using older network & hierarchical data structures
- projects may find diverse legacy
- data sources being mapped to XML-based data structures
- and many more

The challenge here is to facilitate the human intervention that must be done to create the right mappings.

### The data standards gap: Data integration meets the data model

For decades — the enterprise data model was the basic structure that would enable people & systems to work together more effectively. These days we see the data warehouse that helps organization's to share data models & database designs improves data consistency across diverse systems.

### The need for Service Oriented Architecture (SOA)

## SCENARIOS FOR METAMATRIX

The roll-out of new SOA based applications exceptionally challenging because of the cross-functional scope of newer business applications, security requirements & the demanding real-time nature of business interactions **but what about the data?**

### What is 'Service Oriented Architecture'

Take business of any organization; the things in a business that don't change very often:

- Gas stations everywhere sell petrol by the liters
- Restaurants still sell meals from a menu

In short, every business has these aspects that don't change very frequently & we call these functions as the core business functions of that organization. Whereas there are other things in a business that change very frequently such as:

- Prices, Rates
- Catalogs, Advertisements
- New → products/ marketing campaigns/ business areas/ customer areas

Hence, it is very important for any businesses to be able to quickly, in order to survive & yet, it is vital that those changes do not adversely affect the core business functions.

SOA is the practice of sequestering the *core business functions* into independent services that don't change frequently. In short, SOA is a group of services, which communicate with each other. At the highest level, SOA is nothing more than separating changeable elements from unchangeable elements.

### ... Example of an 'Internet store-front'

To make this clear, imagine an internet store-front:

- Customers use a browser to talk to the presentation software that displays the store's website
- The software interprets the customer's request; invokes services that do things like registering an order
- The services have no idea they are talking to a website
- For the customer, they are just talking to a screen that accept & return data in a standard format

### ... But why is this so important

Consider that internet store-front again:

- It presents the user with a catalog

## SCENARIOS FOR METAMATRIX

- allows the user to move items into out of a shopping cart
- finally, accepts the eventual order

Marketing people are likely to want to bring-in change to the steps frequently. E.g.:

- Change from a shopping cart metaphor to scrollable receipt on the sidebar
- Present more or less descriptive data in the product list
- Experiment with different colors, font-faces, and layouts or any other presentation options

But none of this has anything to do with the core business functions. That's why the separation is important. It protects the core information processing assets of the business from the one that is changing OR can be changed depending upon the circumstances. Normally, it is the presentation layer that jitters & spins OR the business processes.

### ... Let us look at the same example once again

Perhaps our business has decided to offer fine wines as one of the products it sells. Now the software needs different type of change as:

- Selling alcohol requires that the age of the customer be verified
- We use a 3<sup>rd</sup> party tool that provides this verification
- This service must be called for any order that contains alcohol products

The decision to call this service is neither a presentation decision, nor a service decision. Rather it is part of the business process for a particular kind of order. As businesses evolve we might add more & more steps to their business processes. Note, those processes don't change much; but the pathways through the processes do. Therefore, we want to separate the business process from the services and from the presentation.

SOA is not about any particular technology. Rather it is a design philosophy that decouples well heeled business functions from processes & presentations. It is the Model-View-Controller (MVC) of enterprise software.

Let us understand - HOW services can be constructed, and how the decoupling of presentation, process, and functions can be achieved.

### • A possible MetaMatrix use case

#### ... Two companies merged that uses different databases

- Your company uses an Oracle DB
- There are around 10 different applications are running such as Oracle, Java, C, etc

## SCENARIOS FOR METAMATRIX

- The company merges with a competitor & the competitor stores its information in a DB2 DB
- & there are similar number of applications running at both sides

### ... Decisions made in the IT integration plan by the new CIO

- Over the next 24 months, all existing applications storage will be migrated to a new unique schema. So on an average, about one application will be migrated every month
- A set of new applications will be developed ASAP so that the new company can start rolling-out plans based on the combined information of the merged entities

### ... List of typical problems that the new company would face

- It is not possible to migrate all the existing applications in one shot
  - Which means many applications will be using the old and the new schema at the same time
- It is not possible to develop a brand new applications including a pair of legacy schema
  - As you only want to use the new schema for all new applications
- It is not possible to import the data directly OR replicate the data due to data formats
  - Due to data inconsistencies or data formats but you want to keep a single repository of the data

### ... MetaMatrix steps that you would follow

- Your architects would define a clean & new DB schema after thoroughly studying the legacy systems
- Using the MetaMatrix Eclipse-based tools →
  - Architects will graphically implement the new schema by:
    - ❖ Capturing Oracle DB, DB2 DB and merge the legacy systems
    - ❖ Defining any required transformation to perform the mapping
  - Architects will set a load of settings such as:
    - ❖ Security scheme
    - ❖ Caching, etc
- Once defined, the new schema & transformation logic is stored in MetaMatrix's meta-data repository
- The team developing the new applications (based on JBoss AS & Hibernate, obviously) use the MetaMatrix JDBC driver and the newly defined schema. At runtime, MetaMatrix loads the referenced schema from the repository (it can load and run several schema at the same time obviously) and acts as a SQL database.
- In parallel, the migration team will focus on one application at a time & upgrade each to the new data model. Note, each application can be migrated to the new schema independently of the others

## SCENARIOS FOR METAMATRIX

### ..... Once all applications have been migrated

The new company can either decide to keep running things this way; Migrates the newly created database itself to the new schema. However, keeping this mediator can have several advantages:

- During the 24 months of the migration program, maybe other mergers take place or the new schema itself has to be modified/ improved
- Even if the physical database itself is being migrated to the new schema, it is better to keep MetaMatrix in the middle with a null "identity" mapping. That way you can make schema changes without affecting the applications
- Even after the DB is migrated to the new schema, it is likely that requests from other departments will come up, asking to combine both departments' databases for reporting purposes

### ..... MetaMatrix has more features that stated above

- On the back-end: ability to aggregate relational & non-relational data sources. Typical examples include mainframes API (through adapters), XML documents and even Excel documents
- On the front-end: ability to represent the aggregated information not only as a relational source but also as an XML source & perform queries through the XQuery interface or as a Web Service

Hence MetaMatrix is not just a one-to-many relational aggregation layer, but really a many-to-many aggregation layer, relational or not.

### • MetaMatrix is the ultimate solution

Forward-thinking organizations are overcoming these challenges by creating a reusable layer of data services to mediate between data sources & applications.

Now by hiding the complexity of data, it is possible to simplify development & integration of applications that use the data. On the longer duration, Data-services lead to shorter project timelines, less costly projects, and reduced maintenance costs. Data-services can easily expressed as either entity-relationship models or XML schemas, enabling

## SCENARIOS FOR METAMATRIX

developers to work with a standard, canonical view of data regardless of the number and type of data sources used to create that view.

### ... MetaMatrix Enterprise

MetaMatrix Enterprise is a powerful data service management system that enables rapid, model-driven creation, deployment, and management of data services. It provides a repository for storing data service definitions with relevant metadata, and a robust execution environment that provides enterprise performance, data integrity, and security. It addresses the diverse data service requirements of today's larger corporations & government agencies.

### ... MetaMatrix Dimension

MetaMatrix Dimension helps optimization of data services that are implemented as standards-compliant Web services. It supports quick Web service-enablement of data sources, and also provides declarative tools for creating Web services & delivering the XML-based data structures. It is ideal for projects focusing on Web services & XML.

MetaMatrix technology decouples applications from their data sources & makes valuable data assets available as services in an SOA, freeing data from single application silos.

### ... MetaMatrix Query

MetaMatrix Query helps the Java developers to access data from diverse relational sources. Also, it simplifies development and maintenance by configuring data access and integration in a single, isolated component.

With MetaMatrix Query the application can:

- Connect to several individual databases
- Extract the data from the metadata
- Integrate the data within the application using SQL OR through Web service interfaces
- Support data sources like Oracle, Microsoft SQL Server, Sybase, IBM DB2 & MySQL
- Optimize the performance of data retrievals

### ... Comparison of MetaMatrix offered tools

	<b>MetaMatrix Dimension</b>	<b>MetaMatrix Enterprise</b>
Target Users	Developers, Architects	Developers, Architects, Modelers, DBAs
Focus Use	Projects	Projects or Architectures
Target Project Size	Small-Medium	Medium-Large (Enterprise)
Number of Sources	1-20	Any number
Types of	Relational, delimited text files, XML, Web services	Any type of source, including packaged

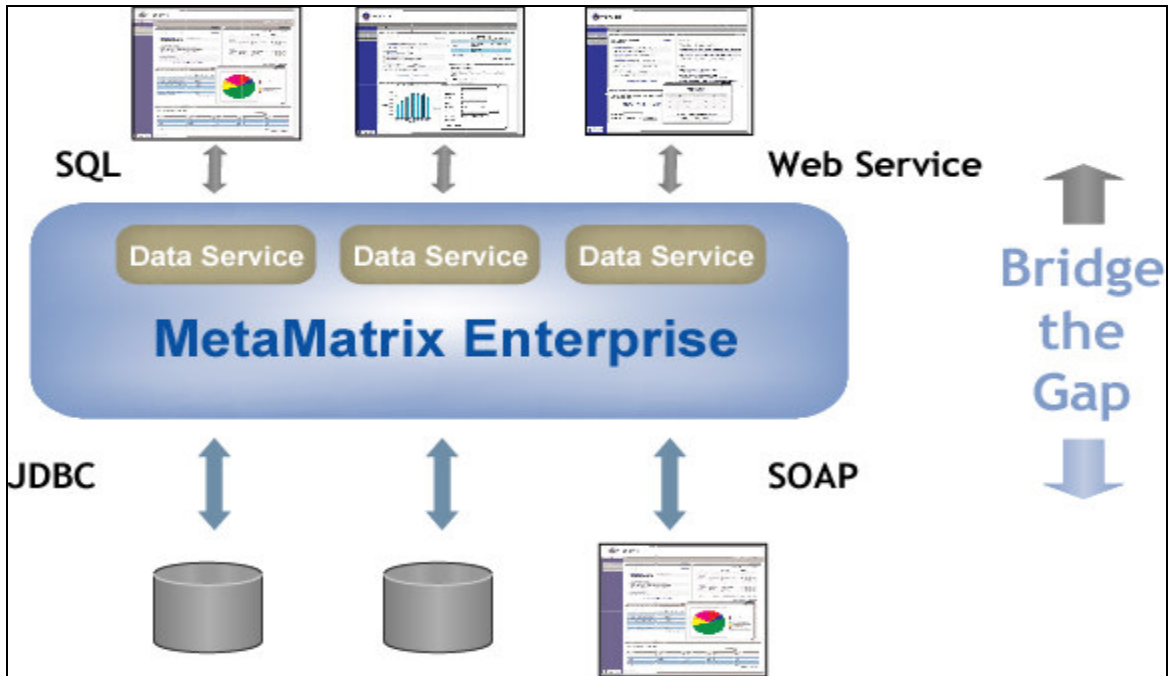
## SCENARIOS FOR METAMATRIX

Sources		applications. Custom connectors can also be developed & integrated
Core Use Case	<b>Web services projects:</b> Integrated data/ information from multiple sources & expose as semantic business views through <u>Web services</u> like (SOAP), JDBC, or ODBC APIs	<b>Data services for SOAs:</b> Enterprise-scale deployment, with repository for model-sharing with a console
Data Operations	Single/multi-source read; single-source update	Single/ multi-source read & Single/ multi-source update with XA transactions
Access Modes	WS-I compliant Web services	JDBC – SQL/ XQuery; ODBC – SQL/ XQuery; WS-I compliant Web services
Modeling	Import source information, define XML Views (backed by <u>XML Schemas</u> ); Expose those views as Web service operations. Use multiple layers of abstraction (views) if desired. Define/ use custom datatypes & scalar functions	Import & model enterprise data models from Popkin, ERwin, and Rational-Rose
Model Abstraction (Views)	Relational & XML views (not exposed); Web services	Relational views, XML views & Web services
Model Sharing	File system	Repository or file system
Deployment	Web service deployed as Web application (WAR) to Web server	Data services are deployed to MetaMatrix Server
Security	Authentication, command and session payload.	Data & model authorization.
Management / Monitoring	N/A	Admin API; Console for configuration; Runtime monitoring, management, troubleshooting & tuning

### ... How MetaMatrix Server works

The MetaMatrix Server enables companies to integrate their enterprise (disparity of information sources) information systems into a virtual database in the form of Metadata. The application users connect to the virtual database to gain real-time access with the help of single sign-on access plus a security.

## SCENARIOS FOR METAMATRIX



### ... Benefits of using MetaMatrix server

- Cuts development time
- Increases ROI by providing: Data integration (integration cost is ~60% of total cost)
- Combine/ integrate data from different islands of information by creating a single virtual database
- Ensure interoperability
- Uses standard protocols/ languages such as XML & SQL to take advantage of skill sets already exist
- Allows any number of applications to use the virtual database for data access
- Also allows the physical data sources to change without effecting downstream applications

### ... Need of MetaMatrix for an Enterprise

Today's complex business environment is subject to increasing regulation, forcing the enterprises to become more accountable & transparent with their data. Under pressure from regulations & potential legal liability, enterprises struggle to achieve a single source of data that can be trusted as being fully accurate & accountable.

### SCENARIOS FOR METAMATRIX

... How MetaMatrix works

