# Powering Ell with MOA

Brad Wright

Randall M. Hauch

January 8, 2004


*Real Enterprise Information Integration*

# Topics

- What is an MOA and why is it important?
- EII as a case study for MOA
- MOA “in action”, a demonstration…
- Attributes of an MOA
- Miscellaneous Topics

# MOA and EII Defined

- Model-Oriented Architecture (MOA)
  - An architecture which utilizes models and metadata describing solutions to control behavior and deliver a solution or result.
  - "Data-driven" architectures where the "data" are formal metadata describing solutions in formal ways
  - Architectures which maximize result while minimizing code
- Enterprise Information Integration (EII)
  - Middleware-based integration of diverse, disparate data in real-time and on-demand
  - Data "federation"
  - A collection of products & technologies which mediate access to data on behalf of applications.

# Why are MOA and EII important?

- MOA represents a new trend in application architecture
  - Maximize adaptability, flexibility
  - Minimize development time
  - Reduce maintenance costs
  - Don't confuse with SOA!
    - SOA are an approach for how to componentize a solution and how the component find and interact with one another
    - MOA are an approach for how individual application components control and define *behavior*
- EII represents a new trend in data access
  - Decouple solutions from data
  - Minimize development time
  - Reduce maintenance costs

# Powering EII with MOA

MetaMatrix has developed an EII product that is an MOA

# About MetaMatrix

- Leading provider of Enterprise Information Integration (EII) products
  - MetaBase: modeling and metadata management
  - MetaMatrix Server: enterprise-grade integration engine
- Founded in 1998
- Headquartered in NYC
- Development in St. Louis
- Currently about 65 employees worldwide
- Privately held, backed by leading venture firms
  - Kleiner, Perkins, Caufield and Byers
  - Invus, Schroder's Finance Partners, Gateway Ventures
- www.metamatrix.com

# About MetaMatrix

- Customers
  - Financial: Merrill Lynch, CSFB, and others
  - Government: various departments in US, UK, and Canada
  - OEM: SAP
- News

*Intelligent Enterprise selects MetaMatrix as 2004 company to watch*
*- Dec 2003*

*MetaMatrix recognized as one of SDTimes' Top 100 Innovators and Leaders*
*- June 2003*

*MetaMatrix Named One of Top 100 Private Companies by AlwaysOn*
*- July 2003*

*New Enterprise Information Integration Sector to Fuel $7.5 Billion Market by 2003*
*- May 2002*

# About MetaMatrix Products

- Currently working on 8th release
  - 4.0 General Availability (GA) in about 3 weeks
- All products are written in Java
  - Except our ODBC driver!
- Process:
  - Mixture of XP, RUP and others to suit our needs
  - Iterative development and release
  - Currently on 6 month release cycles
  - Automated (constant) builds and unit/regression tests w/ reports
  - Organized around 4 product teams, testing team

# About MetaMatrix Products

- Our products run in or on:
  - App Servers: WebLogic, WebSphere, JBoss, SAP
  - RDBMs (our data): Oracle, DB2, SQL Server, Sybase
  - Platforms: Win2K, XP, Solaris, Linux
  - JREs: 1.3.1, 1.4.2
- Statistics (excluding test cases):
  - 1,109,790 lines of code
  - 606,880 non-comment LOC
  - 7,221 Classes
  - 51,413 Methods
  - 8,113 Unit tests (28.3% total coverage)
  - 31 external libraries (all of Eclipse is counted as 1)

metamatrix®

# Applications and Data

- As Java developers, we have many different frameworks for working with application data
  - JDBC, JCA, DOM, JAXP/JAXR/JAXM, EJB, etc.
  - Each framework helps isolate developers and applications from some of the low-level intricacies of various technologies
- These frameworks do *not* help with the bigger data problem: *integrating the data*
- What can help us when our application needs data
  - From multiple sources?
  - Is logically similar but not technically equivalent?
  - Is not in the form our application needs?
  - Is much more complex than our application needs?

# The World of Enterprise Information

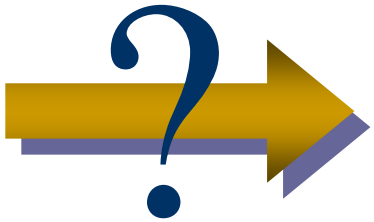
- It's ***Shared***
  - Developers may not be in control of structure or content
  - Multiple applications may be using the same data
- It's ***Distributed***
  - Data is spread across multiple locations
- It's ***Disparate***
  - Similar or related data is managed in different places
  - Each store may have its own protocols, semantics, behaviors
- It's ***Dynamic***
  - Content and structure undergo changes
- It's ***Valuable***
  - Often critically important to the enterprise

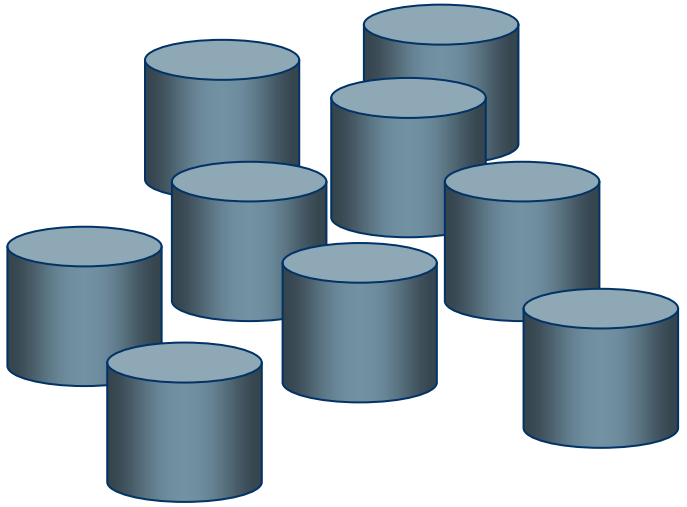
# The World of Enterprise Information

- Typical Global 1000 company (or gov't agency)
  - Hundreds (or more!) of relational databases
  - Hundreds of tables in each database
  - Dozens of relationships in each database
  - Dozens of columns in each table
- That's ***many tens of thousands of columns***!
- Data is isolated into silos
- Business processes require data from many isolated sources developed using varying technologies over decades

# Challenges of Enterprise Information Integration

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

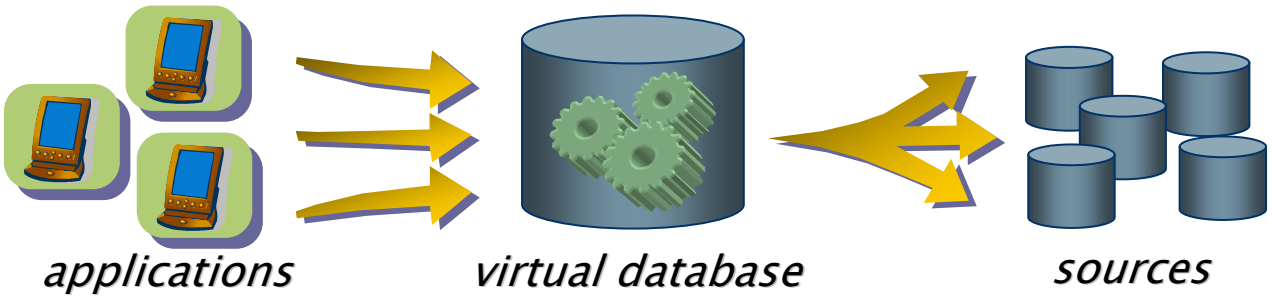




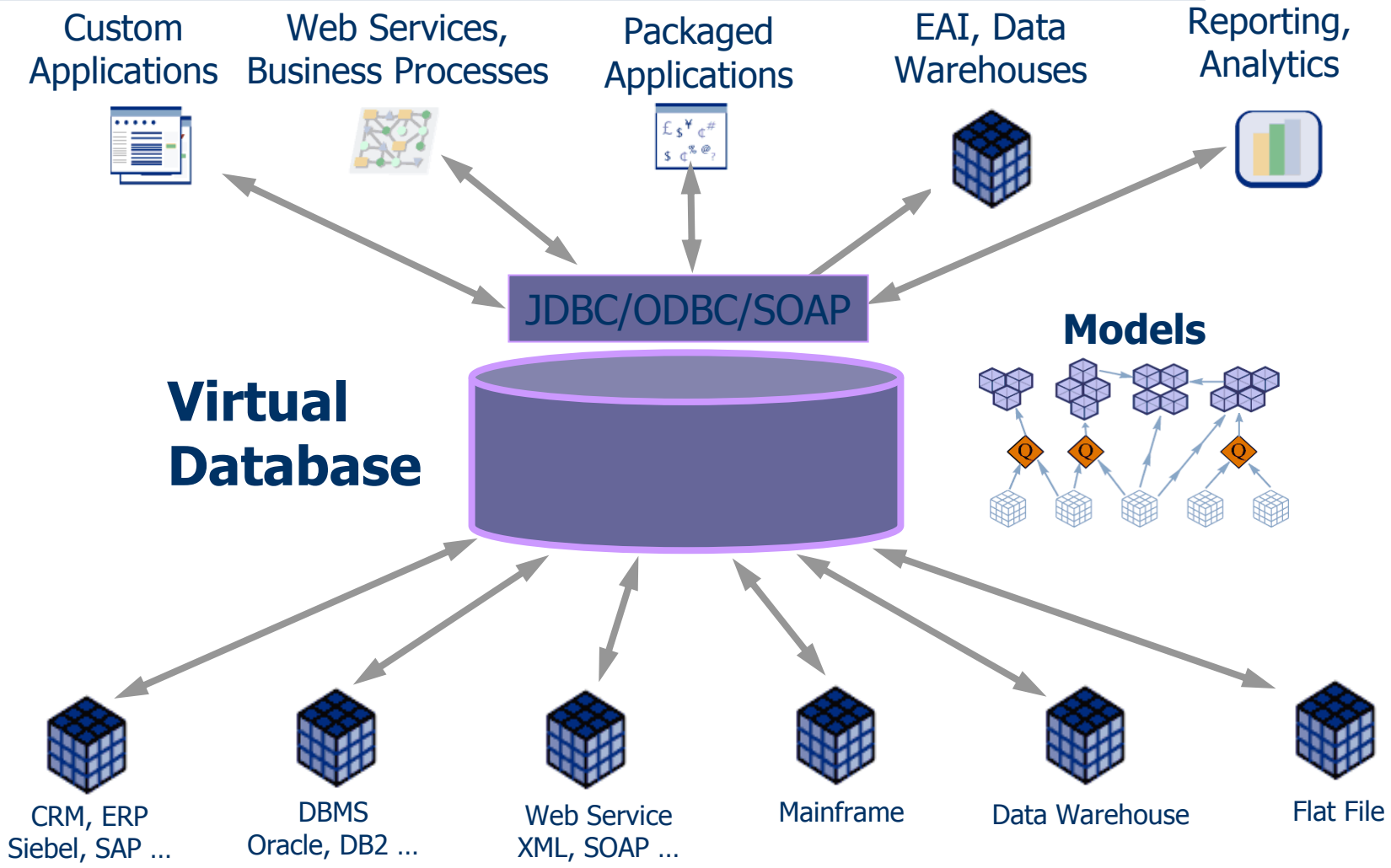
# EII Needs MOA

- EII needs a familiar metaphor: *virtual database*
- EII needs to be *model-driven*

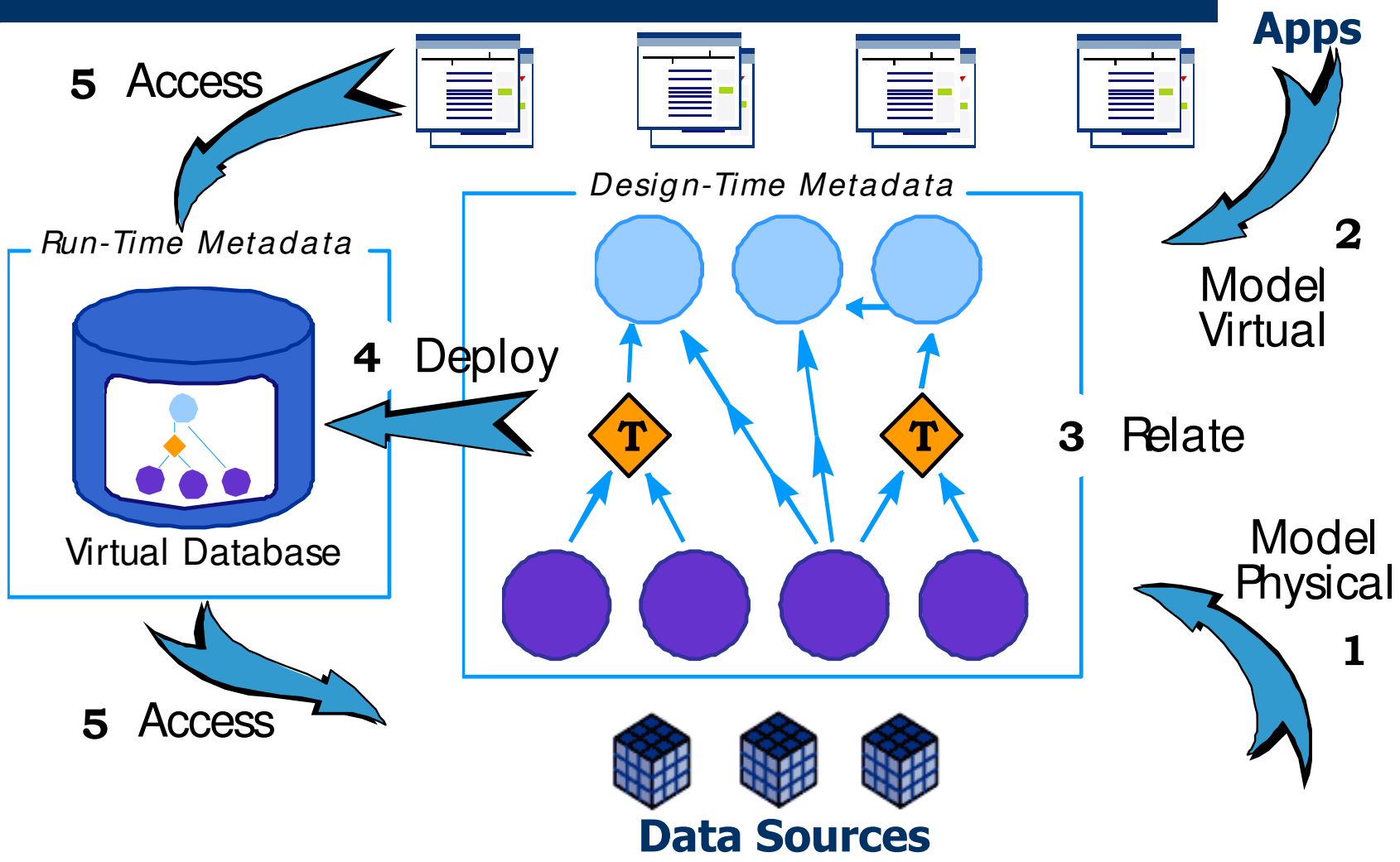
**Given the large diversity in systems to be integrated, EII needs to be an MOA – an architecture whose behavior is defined by models not code.**



# What is EII?

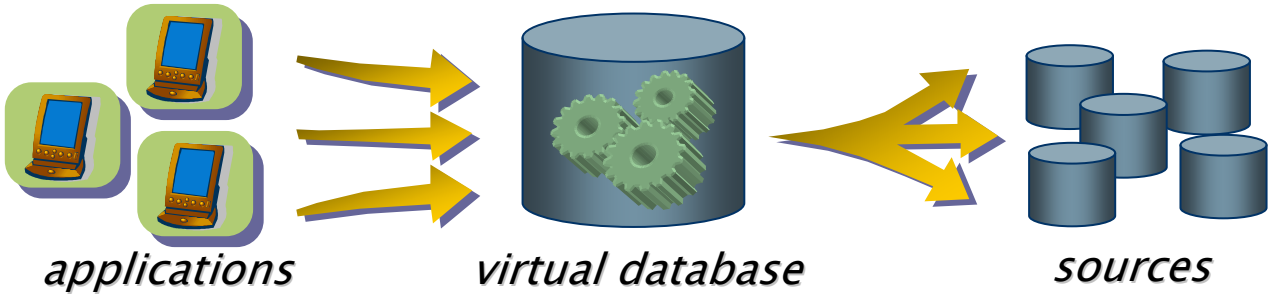


# The "Process" of an EII MOA



# Enterprise Information Integration

- Applications make single/unified requests to the virtual database, which accesses and integrates the data




- Virtual database appears as a normal database, but
  - Data from multiple disparate source is accessed and integrated without client having to know those details
  - Only necessary data is accessed and integrated
  - Allows access to real data in real-time without copies
  - Updates are allowed

metamatrix®

# Enterprise Information Integration

- Makes it easier for applications to get and use the information they need
  - In the form and structure needed by the application
  - Efficiently and in real-time on demand
- The EII engine is designed to
  - Return the correct data (!)
  - Scale to large numbers of clients and sources
  - Be fault tolerant of failures
  - Abstract the applications from the details of the sources
  - Process requests efficiently and very quickly
  - Make use of many different types of sources (not just DBMSs)
  - Support updates and use distributed transactions

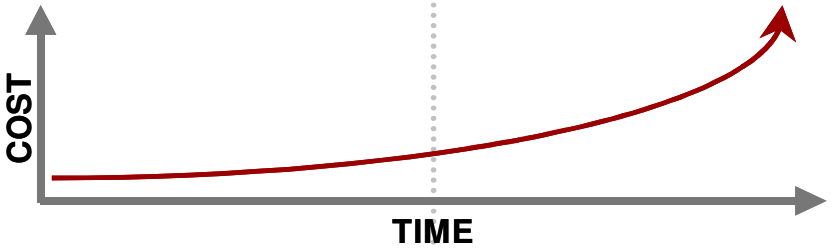
# Build it yourself?

- Some of our best customers tried to do it themselves!
- GIGA Reseach's Perspective:

**Data Integration Costs — Pay Now or Pay Later**

- **Costs for tool-based vs. mostly manual methods are inversely related.**

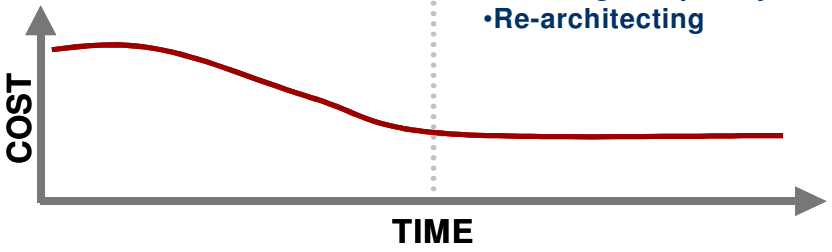
**Mostly Manual:**
- Homegrown ODS
- Hand-coded ETL
- Low-end replication



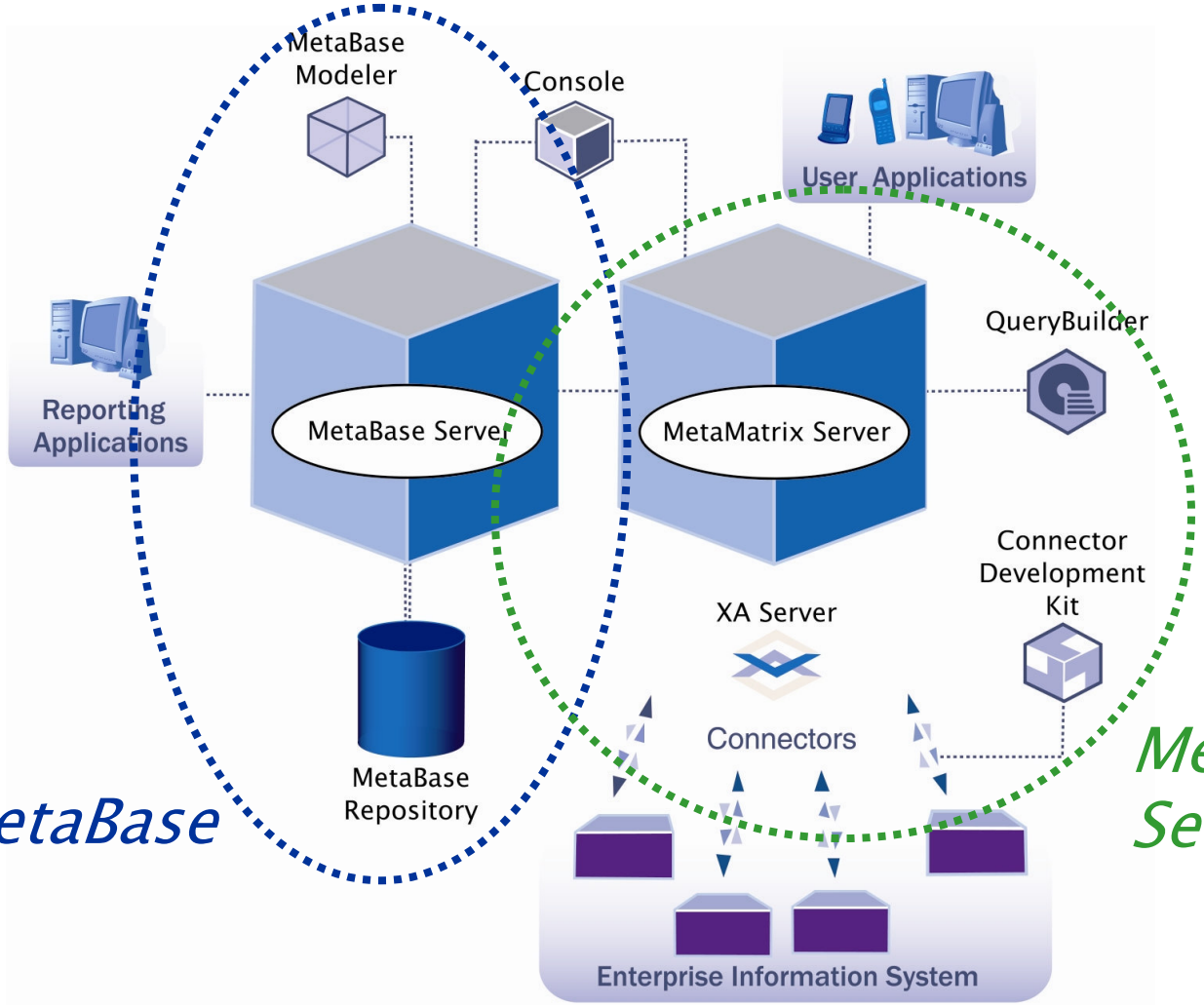

**Startup Costs:**
- Software licenses
- Training
- Hardware

**Maintenance Costs:**
- Changing business requirements
- Growing complexity
- Re-architecting

**Tool-Based:**
- ETL
- EAI
- EII
- Replication



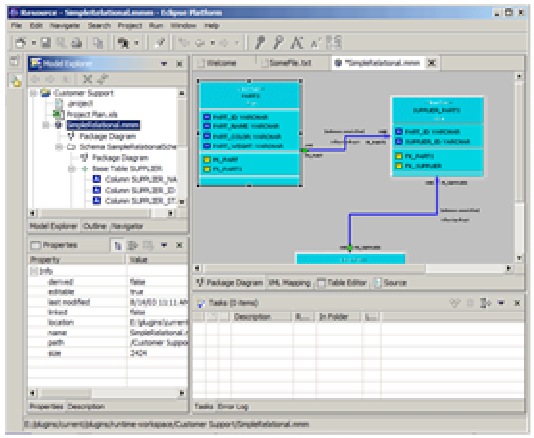
# The MetaMatrix System

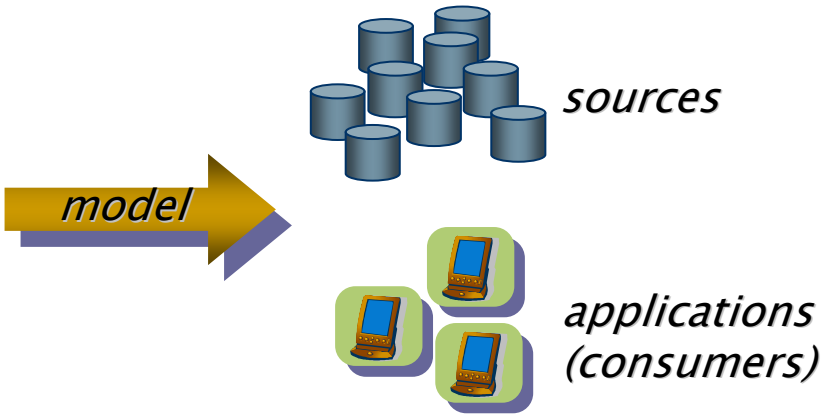


# MetaMatrix and EII

## Step 1: Model Integration Behavior

- Use Modeler to create models of
  - The physical information sources
  - The virtual information needed by information consumers
  - The transformation between the virtual and physical
- Manage models in repository
  - Allows sharing, searching and configuration control

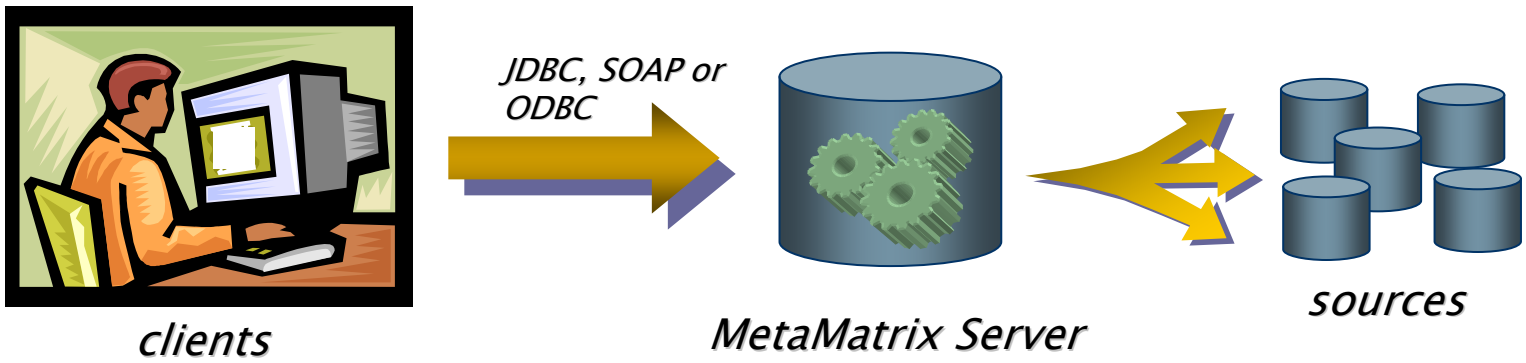




# MetaMatrix and EII

## Step 2: Deploy Models and Execute

- Deploy the models to the MetaMatrix Server
  - Multiple VDBs and multiple versions of same VDB
- Clients connect to VDB via JDBC, SOAP or ODBC
  - Submit queries or execute procedures
  - Obtain result sets or XML documents

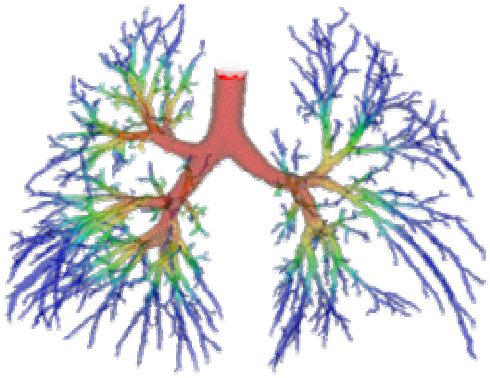


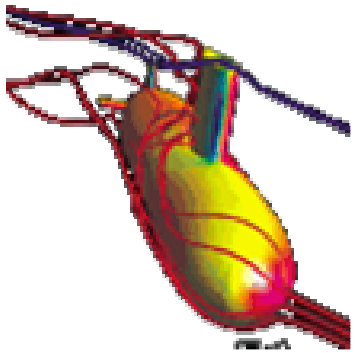
metamatrix®

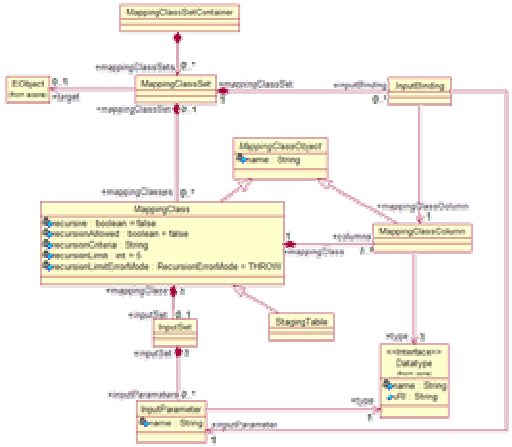
# Model-Oriented Architectures

# Terminology

- A ***model*** is a precise and accurate description of a system that is used for a purpose
  - Often graphical ways of viewing and manipulating the models
  - Usually make it easier to understand the real system







- A software system is ***model-oriented*** if it uses models to dictate its behavior or functionality at runtime

metamatrix®

# Benefits of being Model-Oriented

- Do *not* write code to define integration logic
- Models are easier to create and maintain than would be code
  - They are rich yet more abstract (hide much of the detail)
  - Can view graphically
- Models can be easily reused and exchanged
- Models describe what sources are available
  - Repository of models provides an enterprise catalog

# Demonstration

## MOA In Action: MetaBase Modeler

# Modeling-Related OMG Standards

- Meta-Object Facility (MOF)
  - Defines an architecture for modeling
  - Uses “metamodels” to define behavior and structure of models
- Model-Driven Architecture (MDA™)
  - Defines architecture using models to drive processes and systems
    - Unfortunately interpreted by many to be for application development
- XML Metadata Interchange (XMI)
  - Defines rules that dictate how models defined with MOF are serialized to XML files
- Common Warehouse Metamodel (CWM)
  - Defines metamodels for various types of information systems
  - Relational, record, hierarchical, OLAP, etc.
- Unified Modeling Language (UML)
  - The well-known metamodel for object-oriented systems

# Conclusion

- An MOA System
  - Uses (OMG) standards to define, manage and exchange models
  - Uses models to describe systems and their behavior
  - Consumes models at runtime to produce the behavior
- MOA Benefits to MetaMatrix
  - Solve a very complex problem
  - Makes EII easy to our customers
  - Minimized our development effort

# Miscellaneous Topics About Java Development At MetaMatrix

metamatrix®

# Our Use of Open Source

- Our approach is to use best-of-breed tools
- Many open source products are considered best-of-breed
  - Apache: Xerces, Axis, Ant, Commons, Lucene, RegExp
  - Eclipse: JDT, SDT, PDE, Team, EMF, XSD,
  - Others: JUnit, JAXEN, JBoss, JDOM, SAXON, ConcurrentUtil (Doug Lea)
- The Eclipse plug-in architecture fits into this philosophy
  - Widespread adoption by companies and open-source community
  - Many plug-ins are open-source
  - Plug-ins contribute new functionality to other plug-ins
  - Very active and responsive newsgroups

metamatrix®

# Development Tools

- Eclipse (SDT/PDE)
  - Plus various plugins
- Builds
  - ANT, Cruise Control,
- Coverage and Testing
  - Clover, JUnit
- Apps and Tools:
  - DefectTracker, Wiki, Quickbase
  - Squirrel, DBVisualizer
- Java APIs:
  - J2SE/J2EE 1.3 and 1.4
  - JDBC, JMS, JAXP, JSP, Servlet, EJB, JNDI, JTA/JTS, RMI, …