



DEV427

## MODEL-DRIVEN DEVELOPMENT USING PowerDesigner



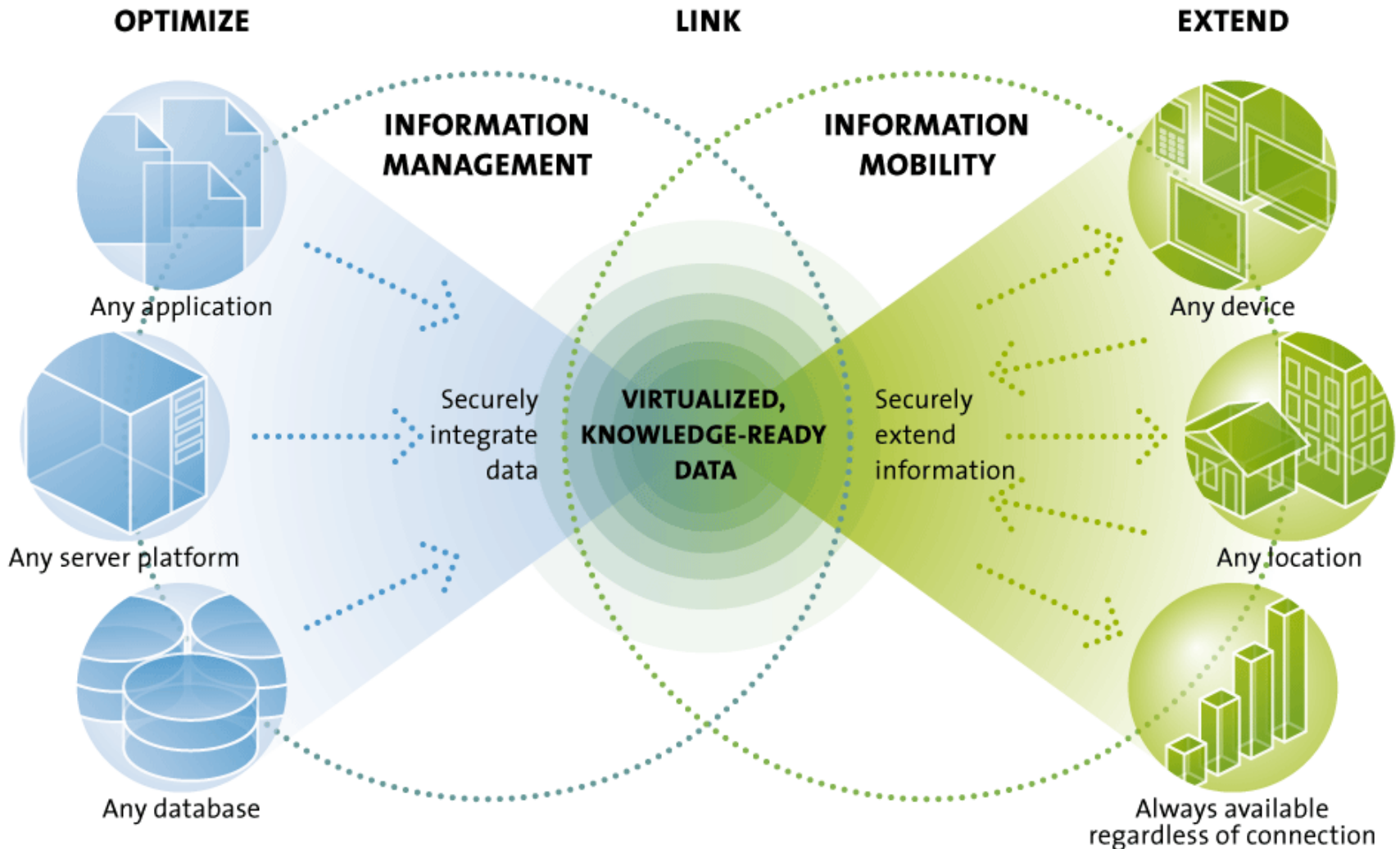
INFORMATION  
ANYWHERE

- Xiao-Yun WANG
- PowerDesigner Chief Architect
- [xwang@sybase.com](mailto:xwang@sybase.com)

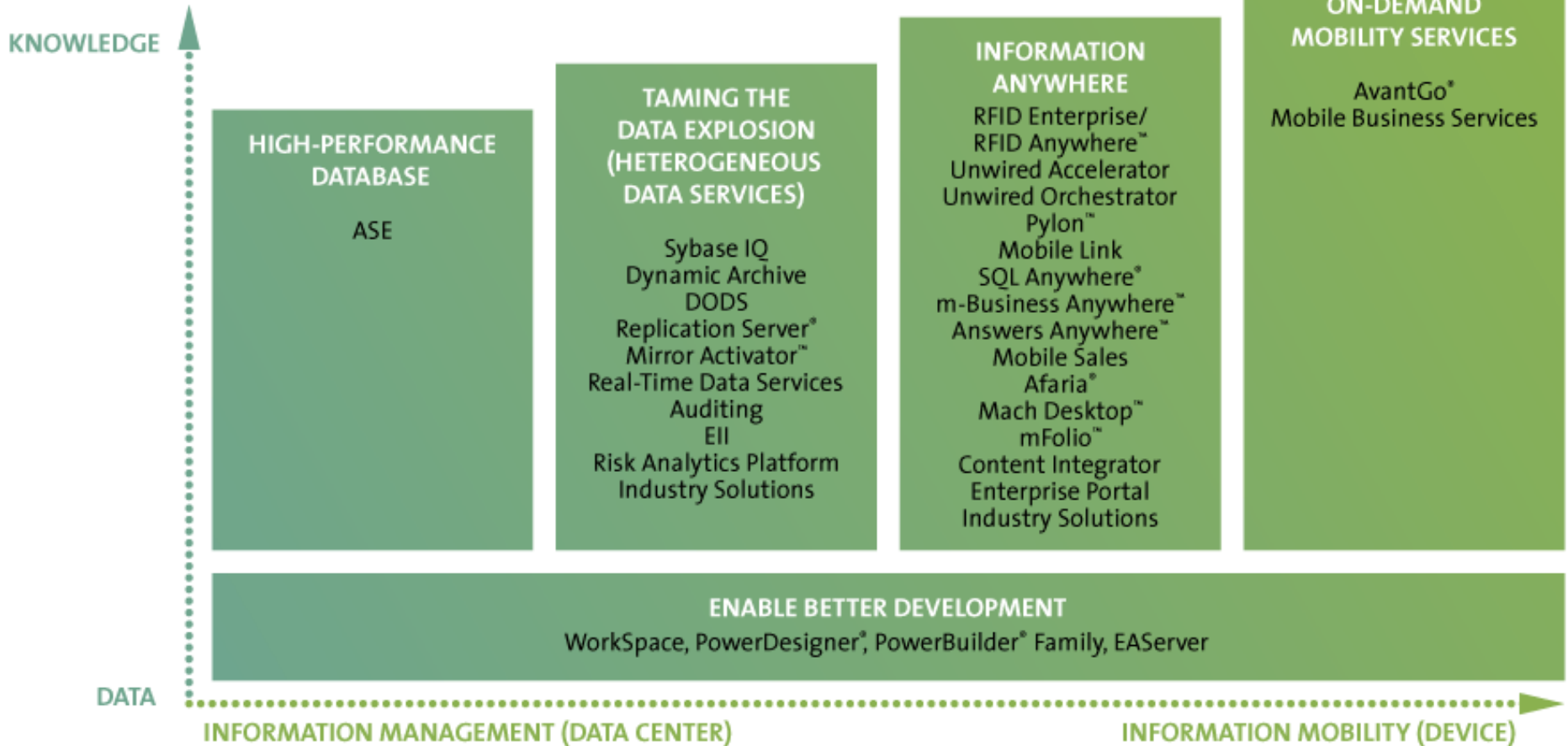
# OBJECTIVES

- 1. Understand what's Model-Driven Development**
- 2. Understand why Model-Driven Development can help you to development applications with better quality and quicker**
- 3. Learn how to apply Model-Driven Development in your projects using PowerDesigner, WorkSpace and PowerBuilder**

# THE UNWIRED ENTERPRISE ACHIEVES AN INFORMATION EDGE



# SYBASE SOLUTIONS



# AGENDA

- **Model-Driven Development Overview**
- **Model-Driven Development using PowerDesigner**
- **Model-Driven Development using WorkSpace**
- **Model-Driven Development using PowerBuilder and PocketBuilder**



## MODEL-DRIVEN DEVELOPMENT OVERVIEW

INFORMATION  
ANYWHERE





# MODEL-DRIVEN DEVELOPMENT OVERVIEW

- **What's Model-Driven Development?**
- **Why using Model-Driven Development?**
- **Model-Driven Development Techniques**

# WHAT'S MODEL-DRIVEN DEVELOPMENT?

- **Model-Driven Development (MDD) is a style of Software Development**
  - Define application requirements, logic, structure and behavior in models or metamodels using a modeling tool (UML, Data model, Business Process model, ...)
  - Generate application prototype code to allow users to validate or generate an API for developers
  - Iterative design, generation, development and test



# WHY USING MODEL-DRIVEN DEVELOPMENT?

- **MDD provides a higher abstraction**
- **MDD allows users to focus on application structure and logic**
- **MDD accelerates development**
- **MDD is more productive**
- **MDD improves application quality**
- **MDD reduces errors and risk**
- **MDD allows applications to evolve easily**
- **MDD simplifies maintenance**

# MODEL-DRIVEN DEVELOPMENT TECHNIQUES 1/5

## ■ **Models definition**

- Users can define application requirements, logic, structure and behavior in models
- Use Requirements Model and Business Process Model for requirements
- Use Business Process Model for application logic and SOA
- Use UML for application structure and behavior
- Use Data Model for objects persistence

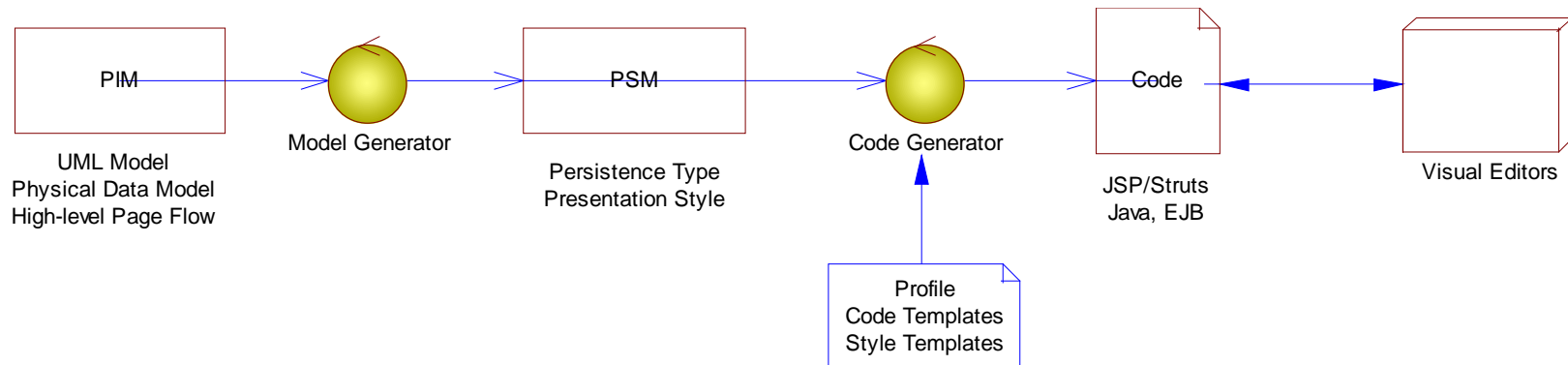
## ■ **Metamodels definition**

- A metamodel is a model of a model
- Users can define their own metamodels using UML
- Example of metamodel: WorkSpace BP Service is a metamodel, WorkSpace BP service file is a model
- A tool could be used to generate an API for developers to create instances (models) of the metamodel

# MODEL-DRIVEN DEVELOPMENT TECHNIQUES 2/5

## ■ MDA

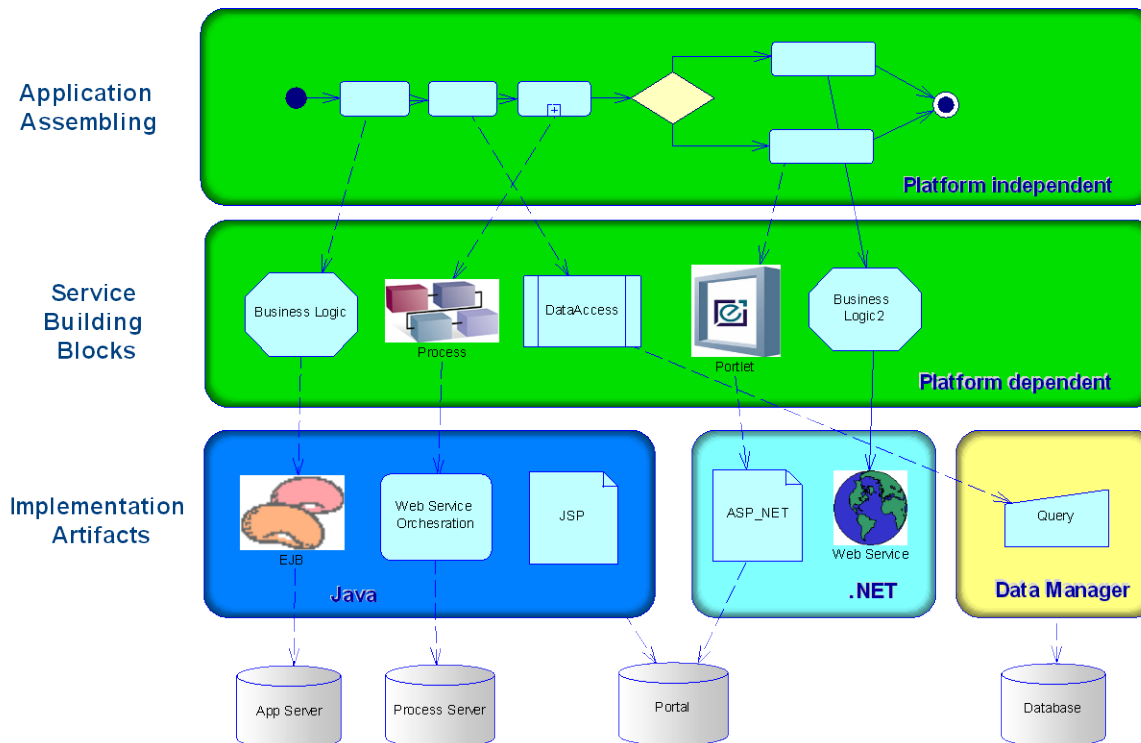
- MDA is defined by OMG
- Platform Independent Models (PIM) and Platform Specific Models (PSM)
- Model transformations
- Code generation



# MODEL-DRIVEN DEVELOPMENT TECHNIQUES 3/5

## ■ SOA

- Separate services development and services assembling
- Allow business users to assemble services using Business Process Model, Business Rules



# MODEL-DRIVEN DEVELOPMENT TECHNIQUES 4/5

- **UML Profile**
  - Used to extend a metamodel or define new metamodels
  - User-defined stereotypes, extended attributes, constraints
- **Code generation**
  - Define metadata in models
  - Generate code from models using templates, scripts, transformations or design patterns
- **Design Patterns**
  - Common solutions for similar problems
  - Example: how to handle notification? Observer pattern
  - Use design patterns to transform models and add additional code in users' models
  - Use design patterns to generate additional code

# MODEL-DRIVEN DEVELOPMENT TECHNIQUES 5/5

- **EMF (Eclipse Modeling Framework)**
  - EMF is a framework that allows users to define a metamodel in XML format or in UML models
  - EMF generates the complete API that allows developers to create and modify objects, load/save objects in XML, manage notification, undo/redo, objects navigation, ...
  - PowerDesigner can design and generate EMF
- **Visual Studio DSL (Domain Specific Language)**
  - DSL allows users to define a metamodel in a Class Diagram
  - DSL generates an API that allows developers to work on objects instances
  - DSL can generate a graphical editor

# SYBASE®

## MODEL-DRIVEN DEVELOPMENT USING PowerDesigner

INFORMATION  
ANYWHERE



SYBASE  
**TechWave** 2005  
USER TRAINING & SOLUTIONS CONFERENCE



# MODEL-DRIVEN DEVELOPMENT USING PowerDesigner

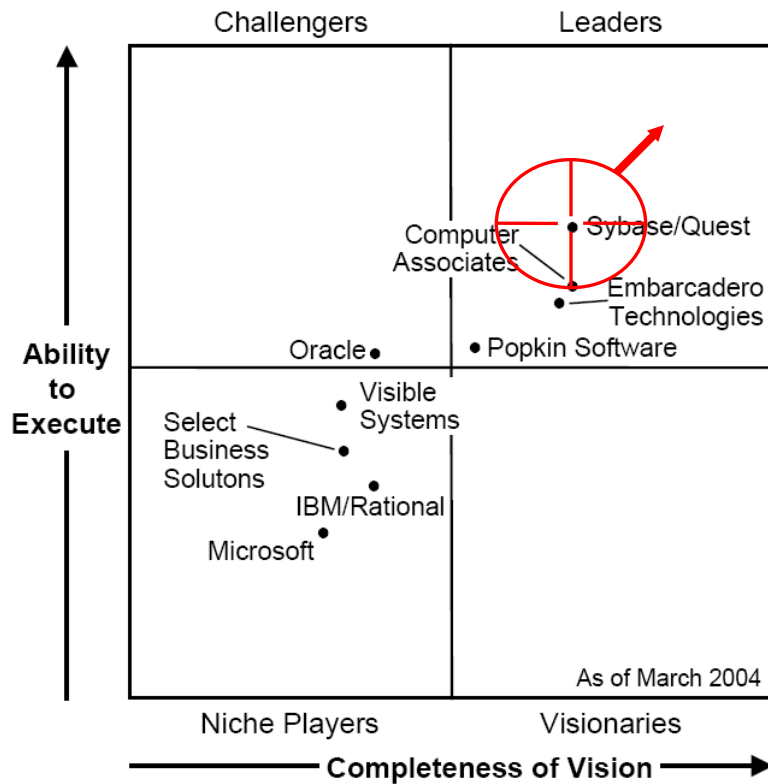
- **PowerDesigner overview**
- **PowerDesigner's Model-Driven Development features**
- **PowerDesigner 12 demo**
- **Advanced Model-Driven Development using PowerDesigner**

# PowerDesigner OVERVIEW

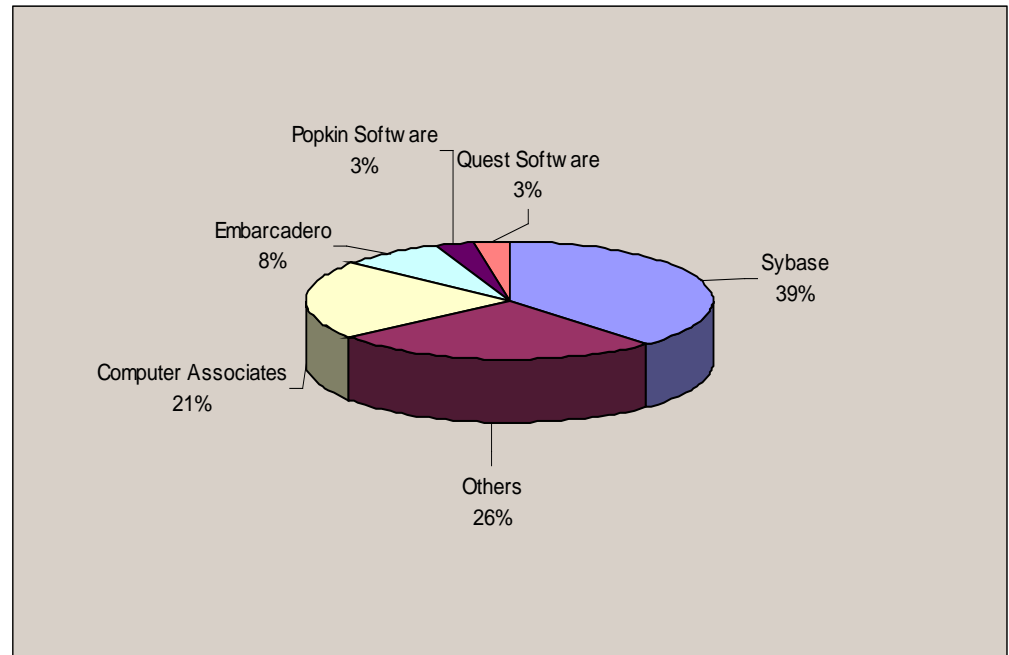
- **PowerDesigner is an integrated enterprise modeling tool**
  - Requirement analysis
  - Business Process modeling
  - Data modeling
  - UML modeling
  - XML modeling
  - Information Liquidity modeling

# PowerDesigner IS THE NO. 1 DATA MODELING TOOL

## Data Modeling Magic Quadrant



PowerDesigner  
Market share in 2002 was 39%  
(Document Gartner August 2003)

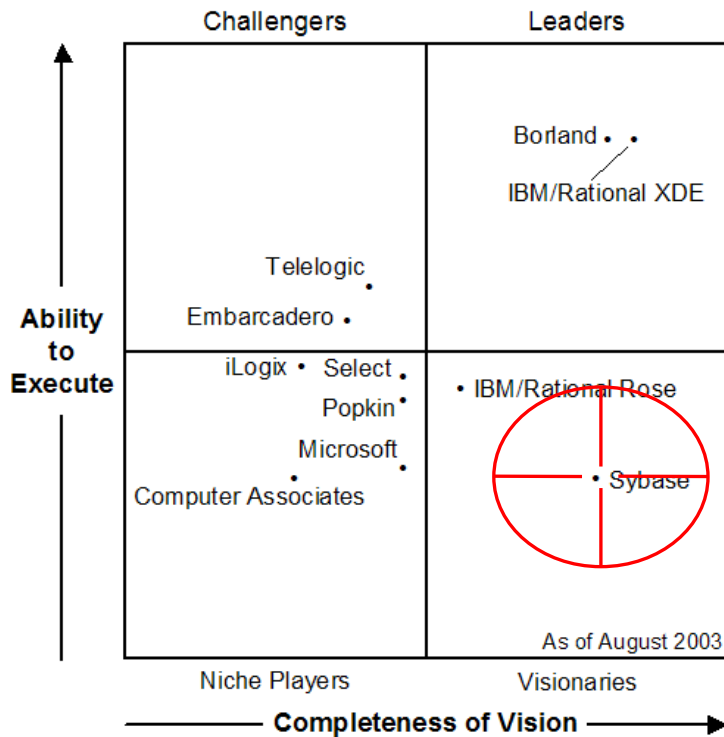


Note: Quest Software resells an OEM version of PowerDesigner under the name 'QDesigner'.

Source: Gartner Research (March 2004)

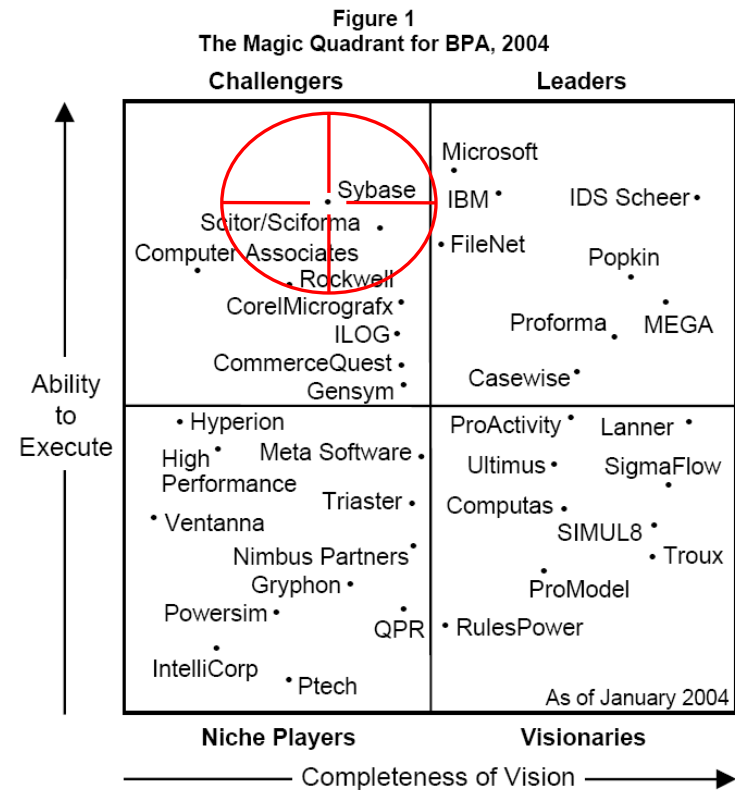
# OOAD & BPA MARKETS

## Object Modeling MQ



**Visionary**

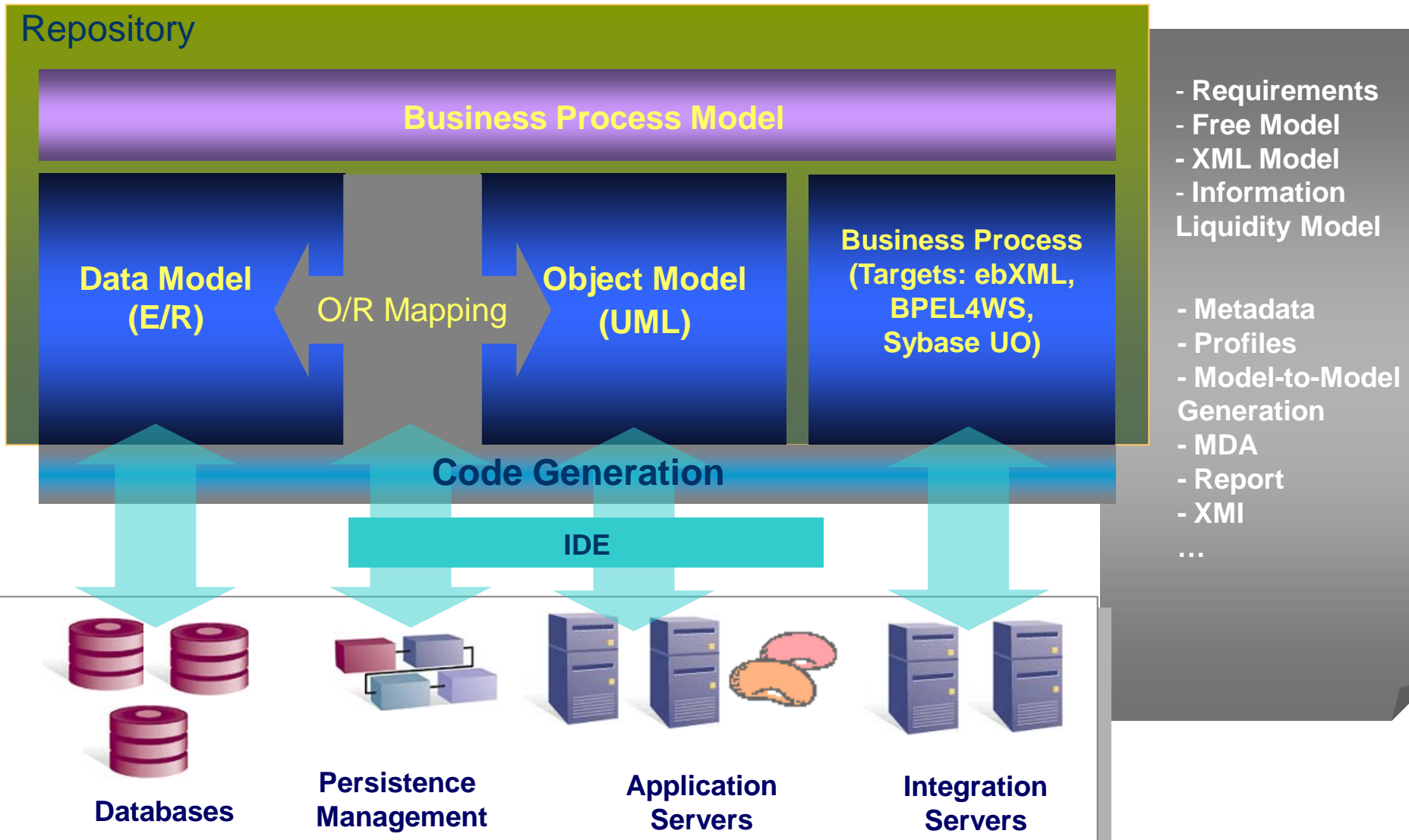
## Business Process Analysis MQ



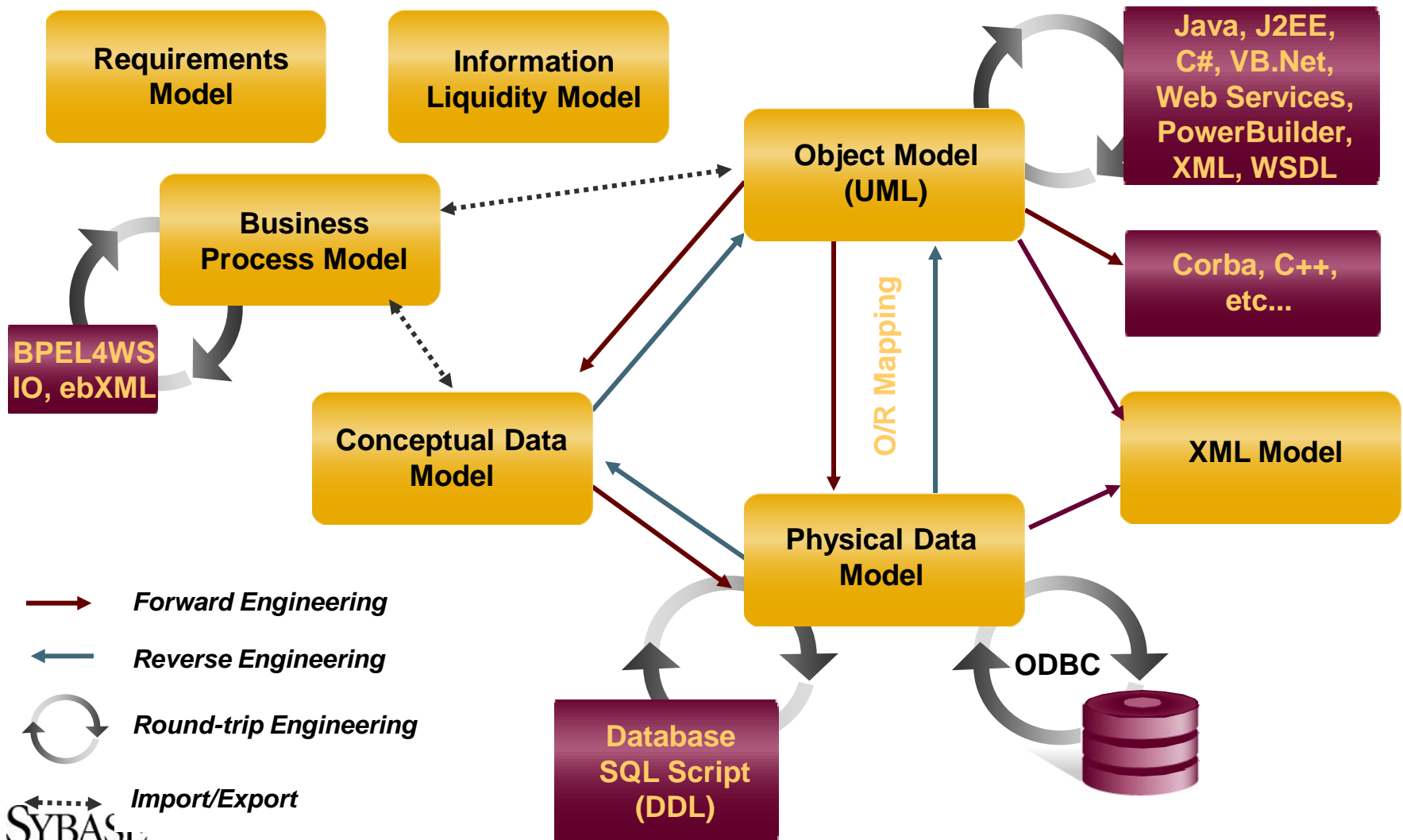
**Challenger**

Source: Gartner Research (January 2004)

# PowerDesigner STRUCTURE



# PowerDesigner MODULES



# PowerDesigner's MODEL-DRIVEN DEVELOPMENT FEATURES

- **Enterprise modeling**
- **Support MDA**
  - Support profiles for all models
  - Support PIMs, PSMs and models transformation
- **Code generation**
  - Generate Java, .NET, PowerBuilder, ...
  - Generate O/R mapping code
  - Generate WorkSpace services, Web services, BPEL, ...
  - Generate XML Schema
- **Database generation**
  - Create and modify database
  - Support XML in database
- **Customization**
  - Support user-defined code generation



# PowerDesigner 12 DEMO

- **Define requirements**
- **Business Process analysis**
- **UML analysis**
  - Uses Cases, Sequence Diagram, Class Diagram
- **Impact analysis**
- **Report generation**
- **Design application**
- **Design database**
- **Define O/R mapping using mapping editor (new)**
- **Generate database**
- **Generate application code and unit tests for Hibernate (new)**
- **Design and generate replication for IQ (new)**
- **Information Liquidity Model for enterprise architecture (new)**

# ADVANCED MODEL-DRIVEN DEVELOPMENT Using PowerDesigner

- **PowerDesigner metamodel**
- **Extended Model Definition (profile)**
- **Code generation templates and Generation Template Language (GTL)**
- **Programming PowerDesigner with VBScript, Java, .NET or PowerBuilder**
- **PowerDesigner Addins**
- **Models transformations**
- **Import/export models**

# PowerDesigner METAMODEL

- **PowerDesigner metamodel defines all the internal objects used by PowerDesigner**
- **The Examples\metamodel.oom model shows you the definition of all PowerDesigner objects**
- **The PowerDesigner Scripting Objects help is an online help for all PowerDesigner objects**

# EXTENDED MODEL DEFINITION

- **An Extended Model Definition defines**
  - All features of UML profiles and more
  - Stereotypes, criteria
  - Extended attributes, collections
  - Custom commands, menus, forms
  - Code generation templates
  - Custom checks
  - Custom symbols
  - Transformations
- **Examples of usage**
  - Generate codes for other languages
  - Extend existing code generation
  - Custom symbols for a specific language

# CODE GENERATION TEMPLATE AND GENERATION TEMPLATE LANGUAGE

- **PowerDesigner uses code templates to generate various code (Java, .NET, PB, XML, ...)**
- **The code templates use the Generation Template Language (GTL)**
- **GTL supports**
  - Macros (.if, .foreach\_item, ...)
  - Variables
  - Access PowerDesigner objects, attributes and collections
- **Examples of usage**
  - Generate codes for other languages
  - Generate codes using your own framework
  - Generate additional codes (unit tests, user-interfaces, ...)

# PROGRAMMING PowerDesigner USING VBScript, Java, .NET and PowerBuilder

- **All PowerDesigner objects are COM objects**
- **VBScript can be used to access PowerDesigner objects inside PowerDesigner**
- **Any programming language that supports COM & OLE automation can access PowerDesigner objects**
  - VBScript, VB, C#, VB .NET, Java, PowerBuilder, Java script, ...
- **Examples of usage**
  - Import/export models or files from/to other tools
  - Create user-interfaces (wizards, generation windows, ...)
  - Add additional commands in PowerDesigner

# PowerDesigner ADDINS

- **PowerDesigner supports Addins**
- **An addin is an ActiveX that implements a special interface**
- **An addin can be developed using many languages (VB, C++, .NET) that uses OLE automation**
- **Examples of usage**
  - Import/export models or files from/to other tools
  - Create user-interfaces (wizards, generation windows, ...)
  - Add additional commands in PowerDesigner



# MODELS TRANSFORMATIONS

- **PowerDesigner supports models transformations**
  - Inter-models generation (CDM->PDM, OOM->PDM, ...)
  - Intra-models generation (OOM->OOM, BPM->BPM, ...)
- **PowerDesigner supports user-defined transformations**
  - You can define objects transformations in Extended Model Definitions
- **You can use PowerDesigner to implement any levels of PIMs and PSMs defined by MDA**

# IMPORT/EXPORT MODELS

- You can import models created by other tools in PowerDesigner to generate your code
- You export PowerDesigner OOM models in XMI format and using other tools to generate code
- PowerDesigner imports Rose, ERwin models
- PowerDesigner supports a partner product MetaIntegration MIMB that can import and export almost any kind of models
- You can develop your own import/export functions using OLE automation or Addin

# ADVANCED MODEL-DRIVEN DEVELOPMENT DEMO

- **Define an Extended Model Definition**
  - Extended attributes and custom forms
  - Custom commands and menus
  - Code generation templates
- **Generate custom code**
- **Program PowerDesigner using**
  - VBScript
  - Visual Basic
  - Java
  - .NET

# SYBASE®

## MODEL-DRIVEN DEVELOPMENT USING WorkSpace

INFORMATION  
ANYWHERE



SYBASE  
**TechWave** 2005  
USER TRAINING & SOLUTIONS CONFERENCE

# MODEL-DRIVEN DEVELOPMENT USING WorkSpace

- **WorkSpace overview**
- **WorkSpace Enterprise Modeling demo**

# WorkSpace OVERVIEW

- **WorkSpace is an integrated analysis, design, development and deployment product**
- **It simplifies the development for Sybase server products**
- **It uses a common framework based on the Eclipse platform**
- **It supports**
  - Requirements analysis and Enterprise Modeling
  - Portal development and mobilization of existing applications
  - SOA for Services development and assembling
  - Data management (SP debugging, real-time messaging, ...)

# WorkSpace OVERVIEW



Web App, Mobile, Portal, Data Management,  
Integration/Web Services, Enterprise Modeling

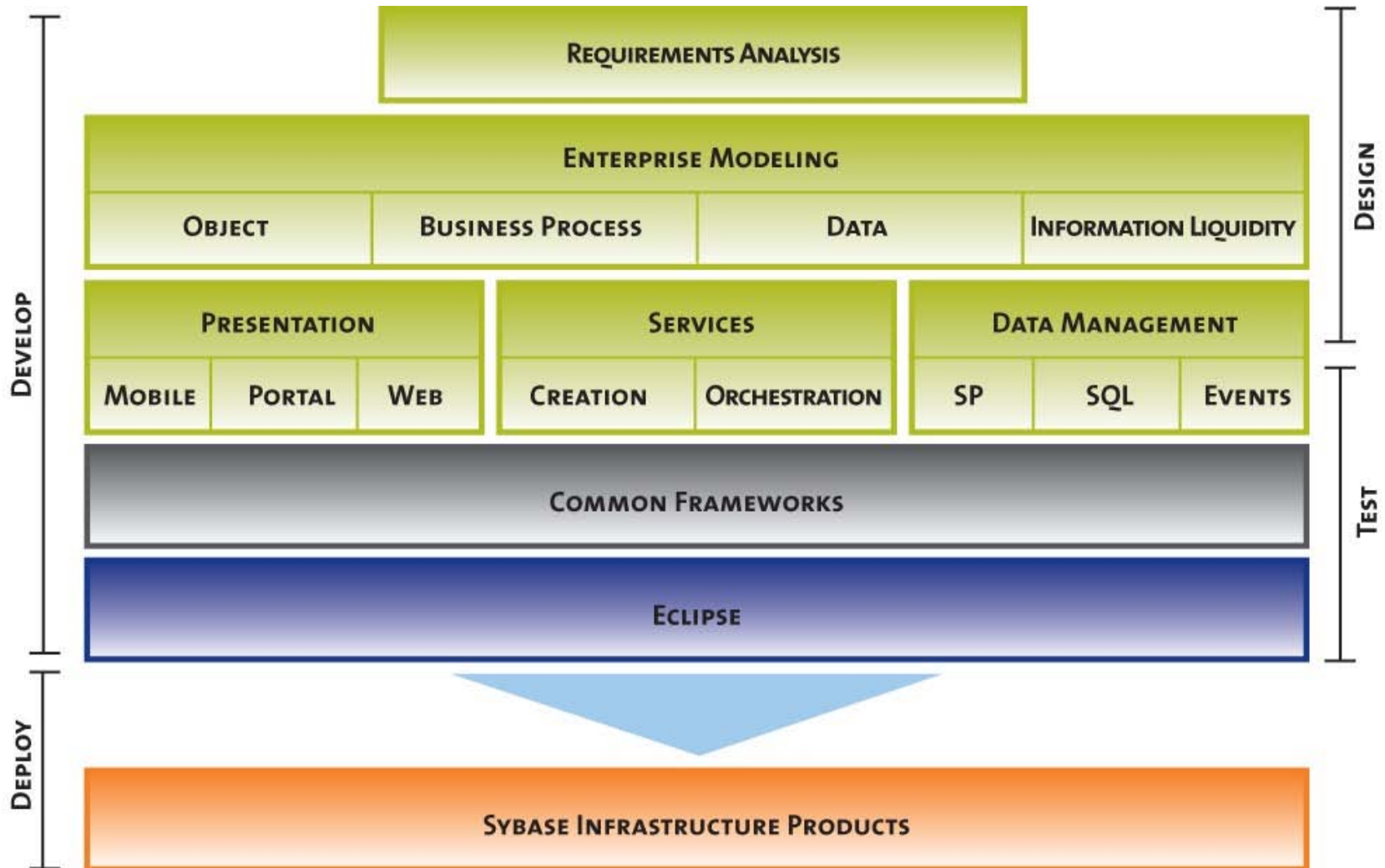
Common Frameworks

Eclipse



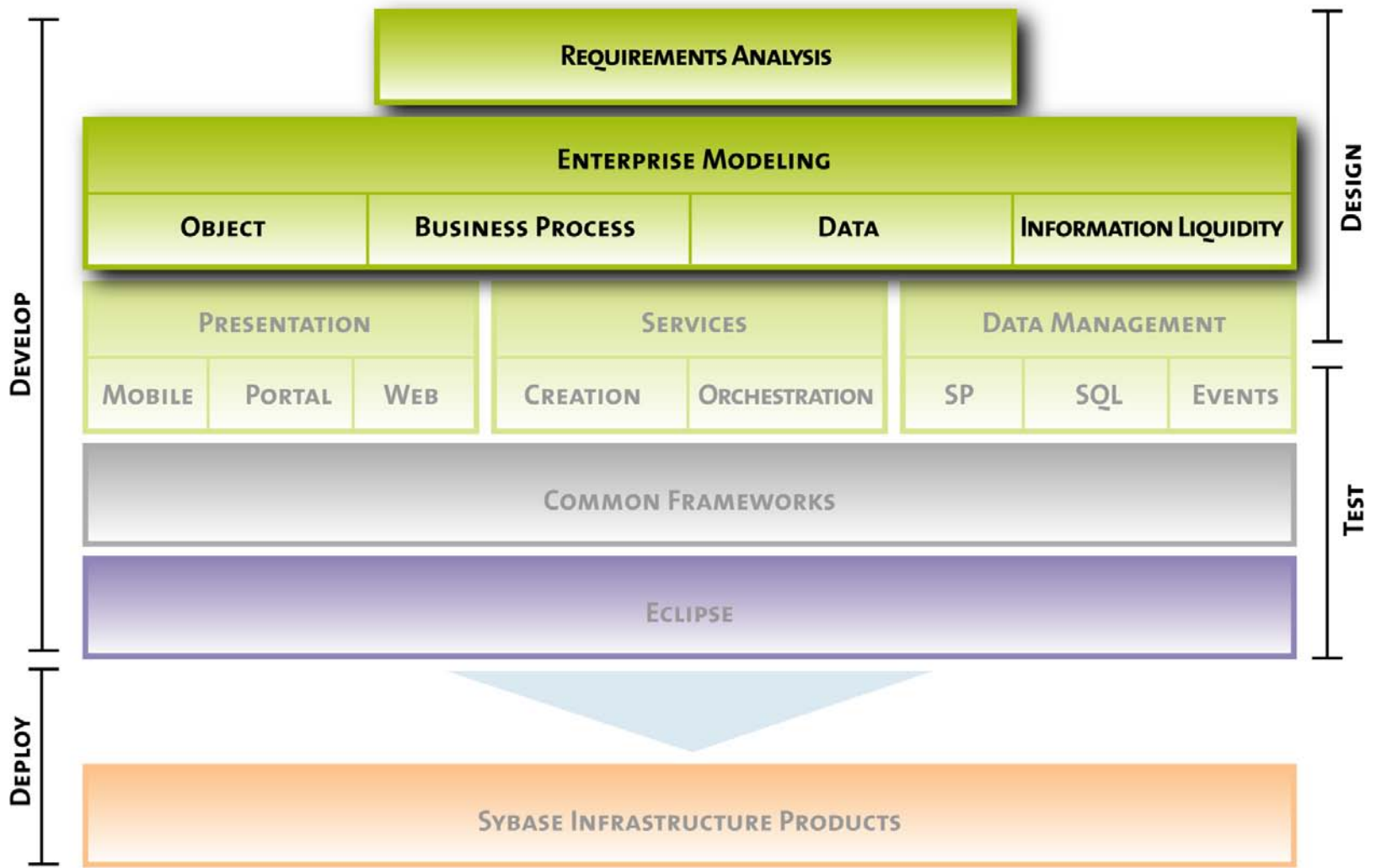
Sybase Infrastructure Products

# WorkSpace OVERVIEW

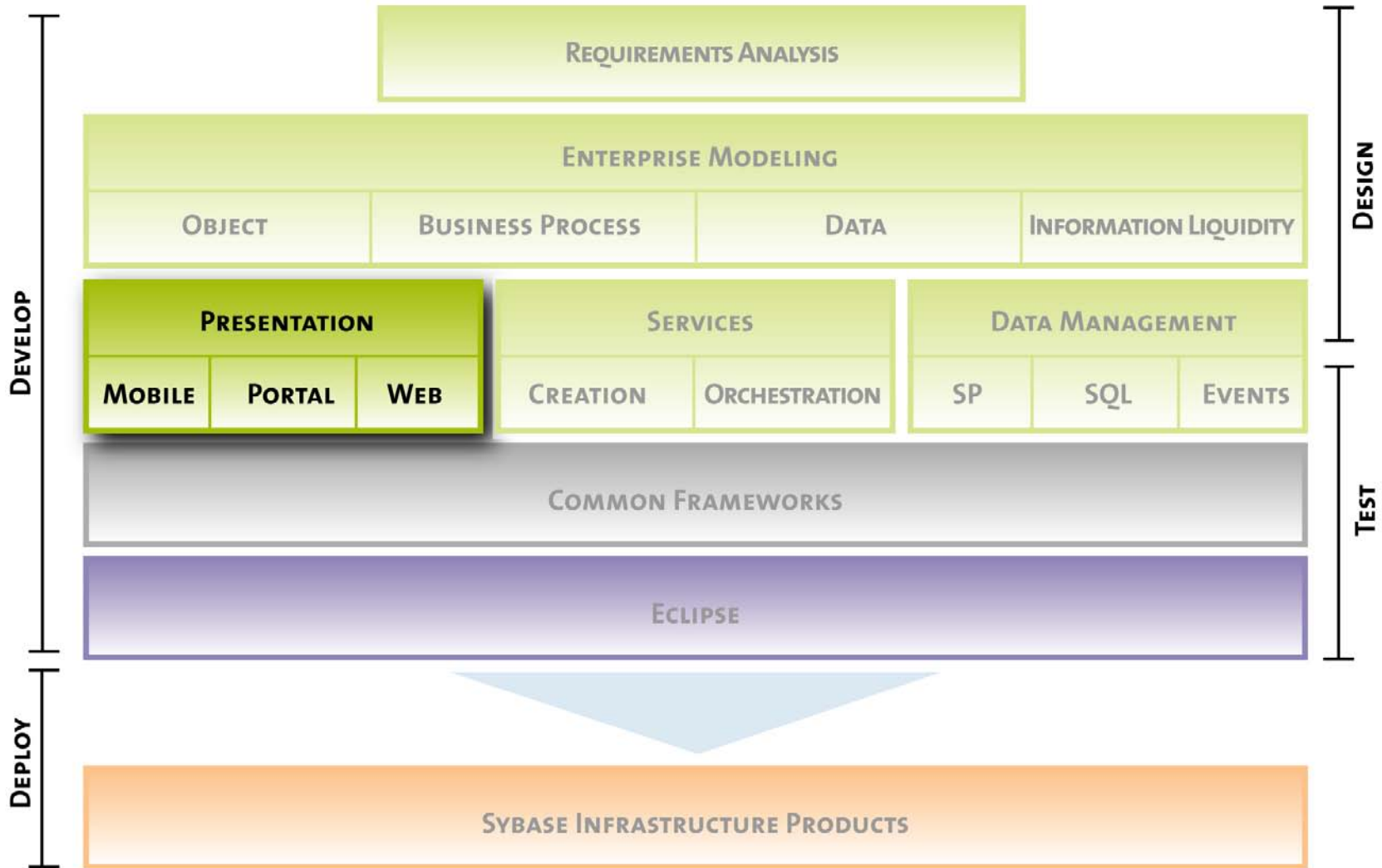




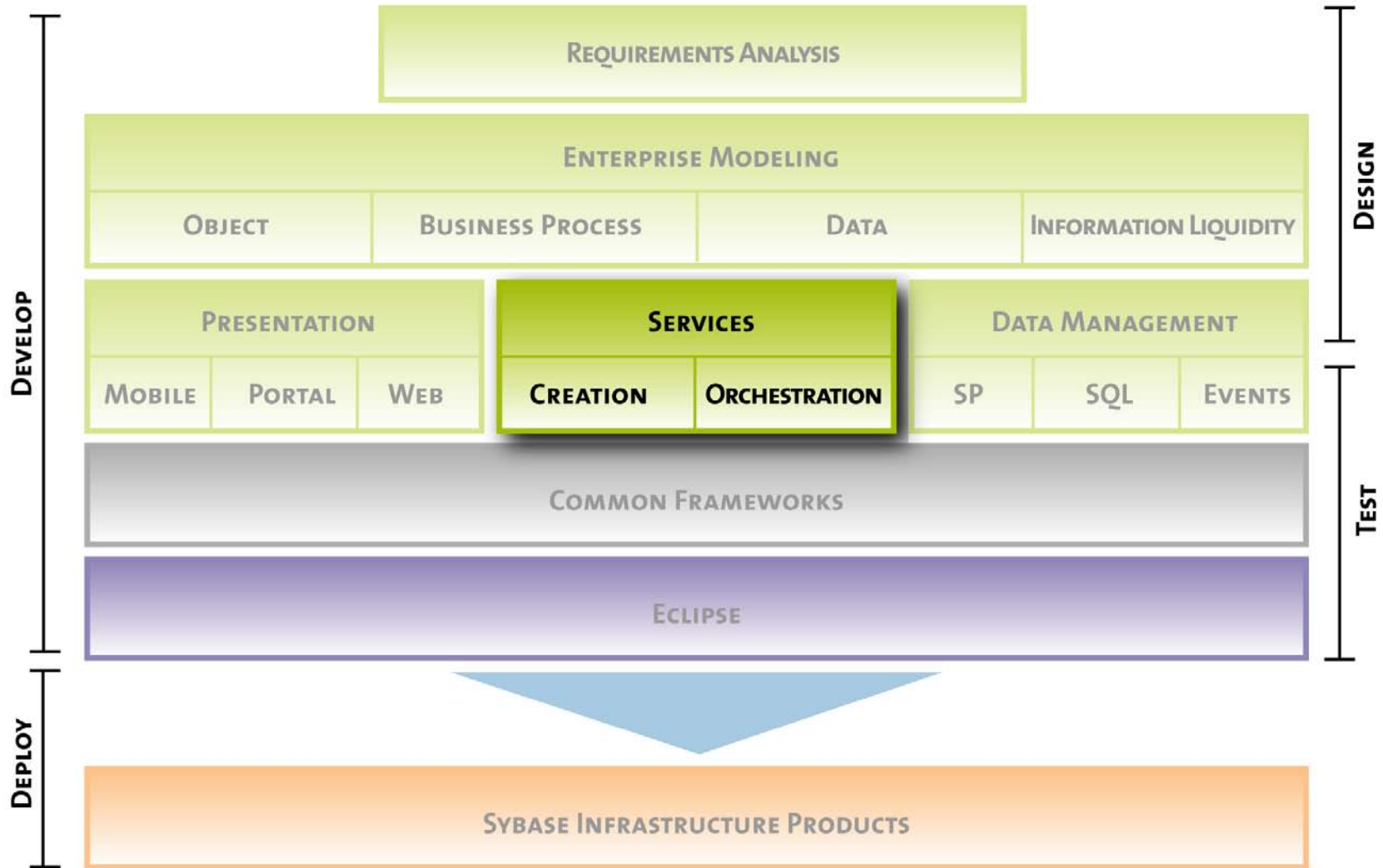
# WorkSpace ENTERPRISE MODELING



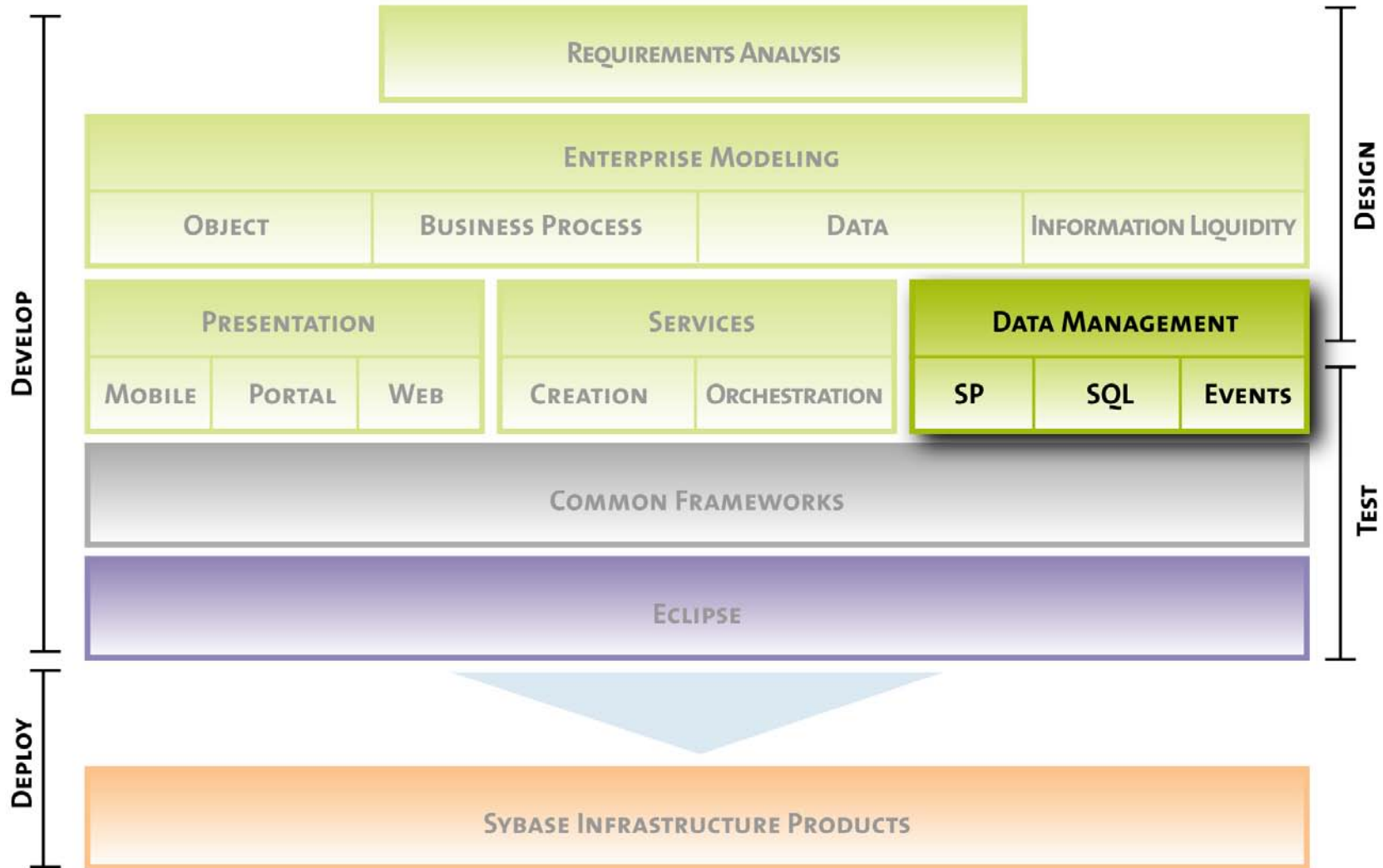
# WorkSpace MOBILE & PORTAL DEVELOPMENT



# WorkSpace SERVICES DEVELOPMENT



# WorkSpace DATA MANAGEMENT



# WorkSpace ENTERPRISE MODELING

- **The WorkSpace Enterprise Modeling feature is based on PowerDesigner**
- **Integration of PowerDesigner in WorkSpace IDE using Eclipse plugin**

# WorkSpace ENTERPRISE MODELING FEATURES

## **Requirements Analysis and Enterprise Modeling**

- Requirements analysis
- Business Process Modeling, UML, Data Modeling

## **Data Management**

- Design and generate ASE, IQ and ASA
- Design and generate RepServer. Support RTDS.
- Design and generate MobiLink

## **Services Development**

- Design and generate XML Schema and DTD
- Design and generate WorkSpace Java and EJB services
- Design and generate WorkSpace Business Process services

# WorkSpace DEMO

- **Integration with WorkSpace IDE**
- **Design and generate database schema**
- **Create the database using WorkSpace**
- **Design and generate a Java service**
- **Design and generate an EJB service**
- **Design and generate a Business Process service**
- **Cheat Sheets**
- **Online help**

# SYBASE®

## MODEL-DRIVEN DEVELOPMENT USING PowerDesigner, PowerBuilder and PocketBuilder

INFORMATION  
ANYWHERE



SYBASE  
**TechWave** 2005  
USER TRAINING & SOLUTIONS CONFERENCE



# MODEL-DRIVEN DEVELOPMENT USING PowerDesigner & PowerBuilder

- **PowerBuilder 10.0 integrates PowerDesigner Class Diagram into its IDE**
- **The Class Diagram allows PowerBuilder developers to:**
  - Reverse engineer existing PowerBuilder applications
  - Document existing applications
  - Refactor the applications
  - Generate a report
  - Regenerate the application
- **Developers can develop additional features based on Class Diagram using PowerScript**

# PowerBuilder DEMO

- **Reverse engineer PowerBuilder application using Class Diagram**
- **Refactor PowerBuilder application**
- **Document the PowerBuilder application**
- **Regenerate PowerBuilder application**
- **Access PowerDesigner metadata using PowerScript**

# MODEL-DRIVEN DEVELOPMENT USING PowerDesigner & PocketBuilder

- **PowerDesigner Class Diagram allows PocketBuilder developers to:**
  - Reverse engineer existing PowerBuilder applications
  - Document existing applications
  - Generate a report
  - Refactor the applications
  - Regenerate the application
- **PowerDesigner Information Liquidity Model allows users to design and generate MobiLink synchronization**

# PocketBuilder DEMO

- **Define MobiLink synchronization using Information Liquidity Model (ILM)**
- **Reverse engineer and generate PocketBuilder application using Class Diagram**
- **Test PocketBuilder application**

# PowerDesigner ROADMAP

- **PowerDesigner 12      Q1 2006**
  - UML 2
  - Mapping editor
  - Improved database generation, reverse engineering
  - Report wizard and list report
  - Support ASE 15.0
  - Support PowerBuilder 10.5
- **PowerDesigner 12.1    Q2 2006**
  - Visual Studio .NET 2005 plugin
- **Beyond PowerDesigner 12**
  - Support Enterprise Architecture
  - Improve Enterprise Modeling

# SUMMARY

- **Model-Driven Development approach allows users to:**
  - Define business requirements and metadata in models before the implementation
  - Generate the default applications code from models
  - Accelerate the development
  - Improve applications quality
  - Simplify maintenance
- **PowerDesigner provides all the necessary tools for business users, analysts, designers and developers to implement Model-Driven Development**

# PowerDesigner RESOURCES

## **PowerDesigner Web site**

<http://www.sybase.com/powerdesigner>

## **PowerDesigner Newsgroup**

<news://forums.sybase.com/sybase.public.powerdesigner.general>

## **CodeXChange**

<http://powerdesigner.codexchange.sybase.com>

## **PowerDesigner Advanced Documentation**

Accessible from PowerDesigner help menu

## **PowerDesigner VBScript online help**

Accessible from PowerDesigner help menu

## **Microsoft Windows Script online help**

Can be downloaded from <http://msdn.microsoft.com>

## **OLE Automation samples**

OLE Automation\\*.\*

## **PowerDesigner Metamodel**

Examples\MetaModel.com

# OTHER PowerDesigner SESSIONS

- **AM15 Data Modeling with PowerDesigner 11**
- **TDE221 Data, where it is, where it came from, where it is going - a Modeling approach**
- **DEV422 PowerDesigner Future Directions**
- **DEV423 PowerDesigner Advanced Tips and Tricks**



**THANK YOU!**

**QUESTIONS?**



DEV427

## MODEL-DRIVEN DEVELOPMENT USING PowerDesigner



INFORMATION  
ANYWHERE

- Xiao-Yun WANG
- PowerDesigner Chief Architect
- [xwang@sybase.com](mailto:xwang@sybase.com)