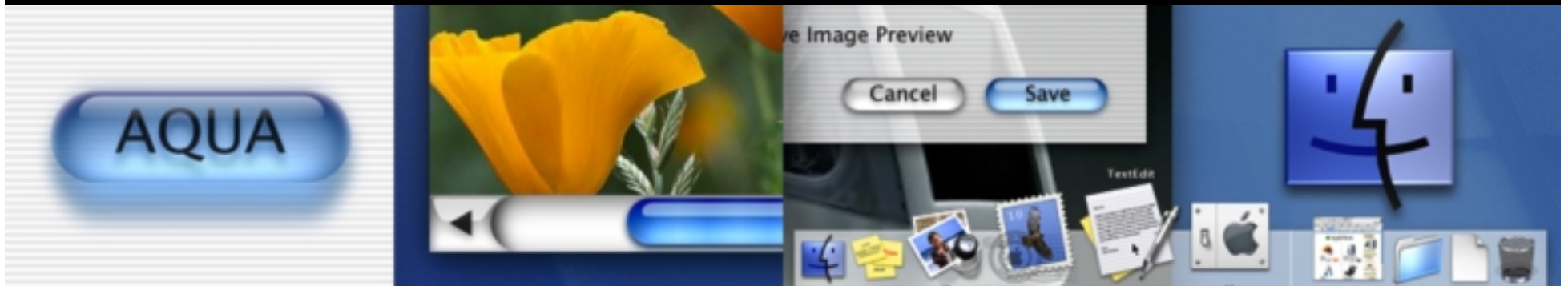




Session 406

WebObjects: Direct to Java Client



Andreas Wendker
Manager, WebObjects Rapid Development

Introduction

- Direct to Java Client
 - Is one of the newest technologies in WebObjects
 - Was introduced in WebObjects 4.5 for the first time
 - Has a similar purpose as Direct to Web, but is used for Intranet, not Internet applications



What You'll Learn

- Direct to Java Client is a very strong rapid development tool for desktop business applications
- Direct to Java Client is ideal for prototyping purposes
- Direct To Java Client is not a toy

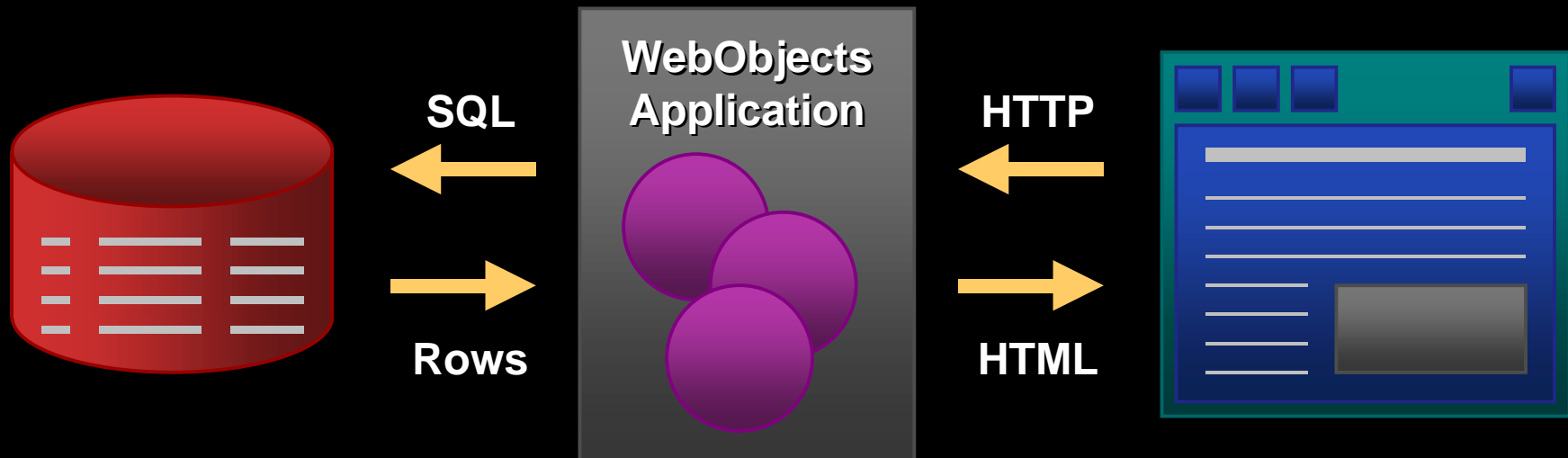


Agenda

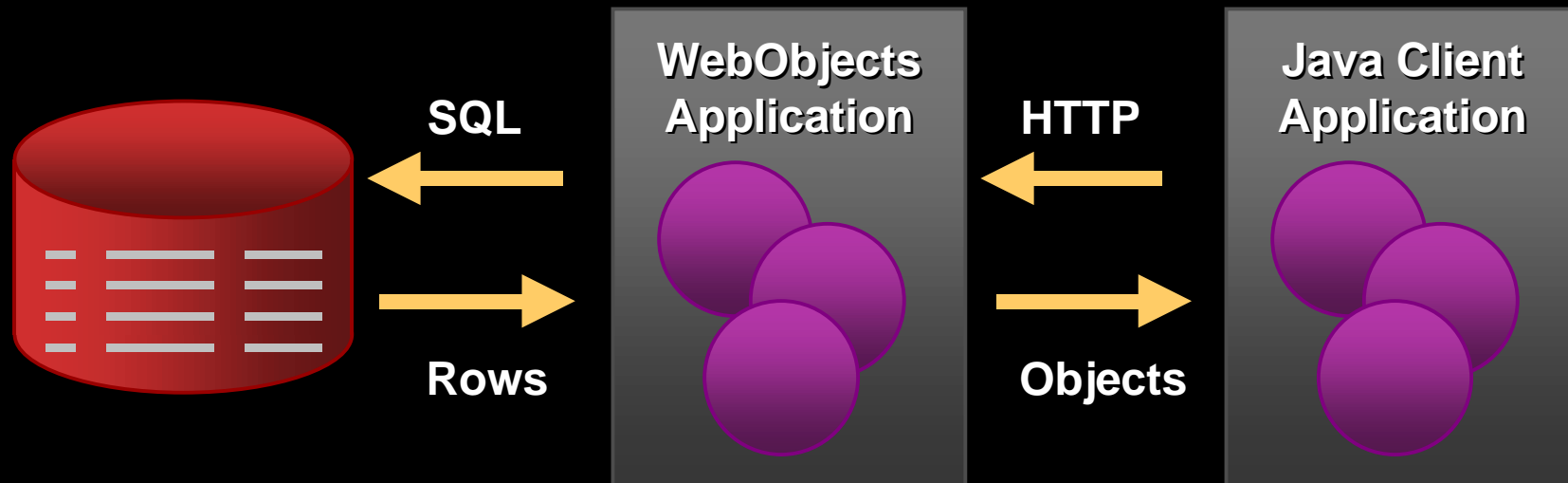
- Technical Description
- Demo: Development Process
- Customization
- Demo: Customized Application
- Benefits
- Summary
- Q&A



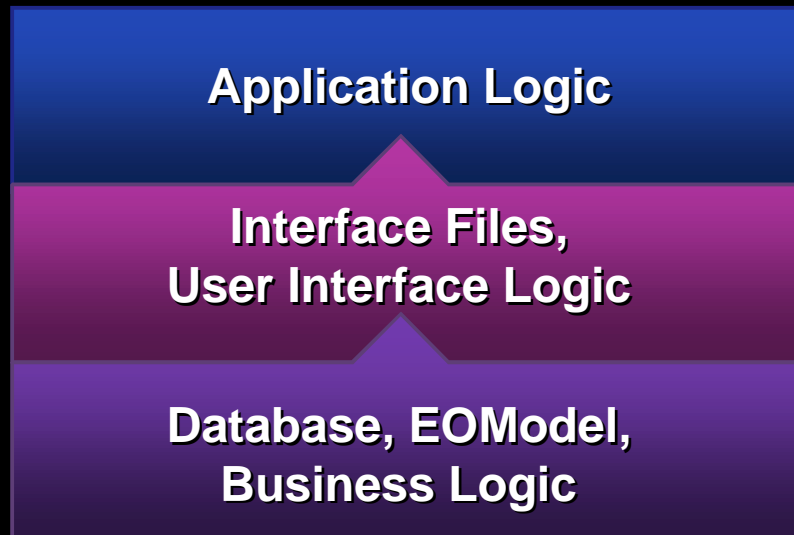
Web Application Architecture



Java Client (Desktop) Architecture



Traditional Development



- Pieces depend on each other
- High maintenance cost
- Testing often requires all pieces
- Long bootstrapping time, prototyping difficult
- Similar development steps repeated again and again



Direct to Java Client's Idea

- Generate application logic and user interface dynamically just based on the EOModel
 - Develop, test, polish business logic first (before investing time into user interface)
 - Produce prototypes in shortest time



Direct to Java Client Development

**Database, EOModel,
Business Logic**



Default Application



Customized Application



Polished Application

- Creating database and EOModel sufficient to run default application
- Customization is easy and quick (efficient prototyping)
- Delay polish (user interface layout, application logic) until business logic finalized

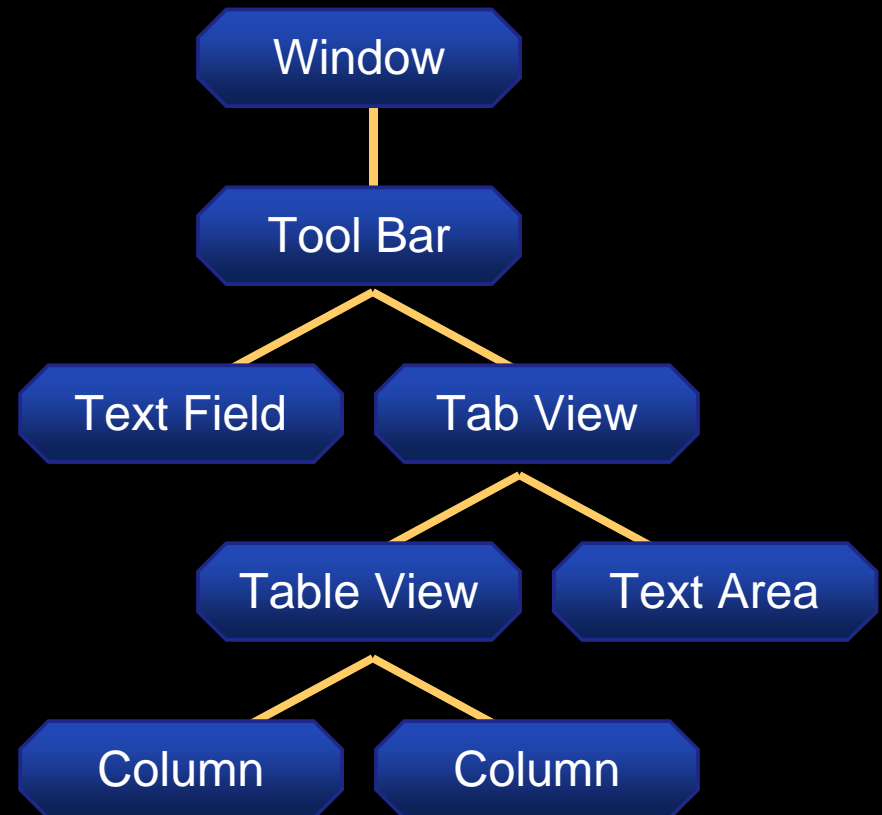
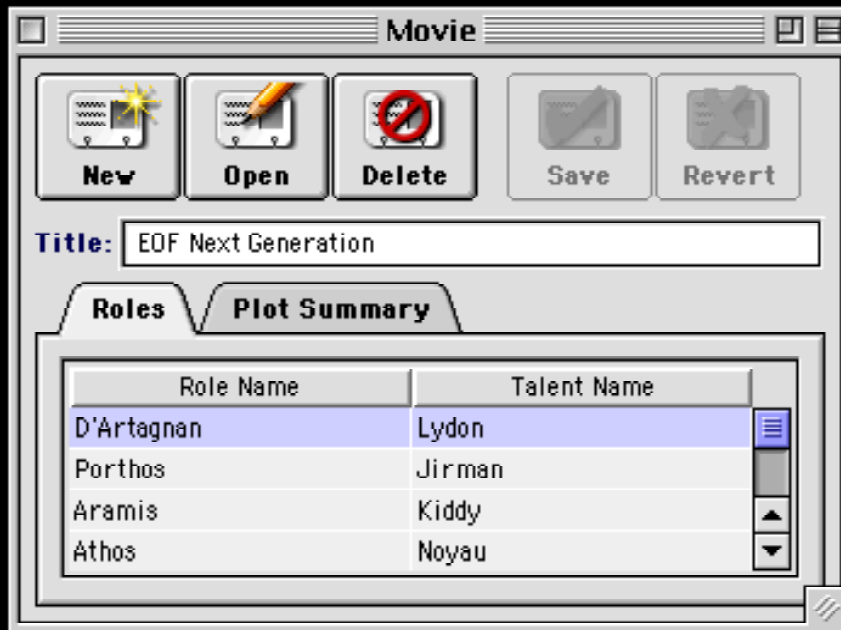


User Interface Generation

- Direct to Java Client defines a great number of controllers for all kinds of tasks
 - Application object
 - User interface controllers (windows, tab views)
 - Entity level controllers (query, list, form)
 - Property level controllers (text fields, action buttons)
- Controllers know how to create, lay out the necessary user interface



Controller Hierarchies



Controller Hierarchy Generation

- Server side is responsible to generate the hierarchy of controllers in XML format

```
<WINDOWCONTROLLER position="Center" label="Movie">  
  <TOOLBARCONTROLLER>  
    <TEXTFIELDCONTROLLER label="Title"/>  
    <TABVIEWCONTROLLER>  
      <TABLECONTROLLER label="Roles">  
        <COLUMNCONTROLLER label="Role Name"/>  
        <COLUMNCONTROLLER label="Talent Name"/>  
      </TABLECONTROLLER>  
      <TEXTAREACONTROLLER label="Plot Summary"/>  
    </TABVIEWCONTROLLER>  
  </TOOLBARCONTROLLER>  
</WINDOWCONTROLLER>
```

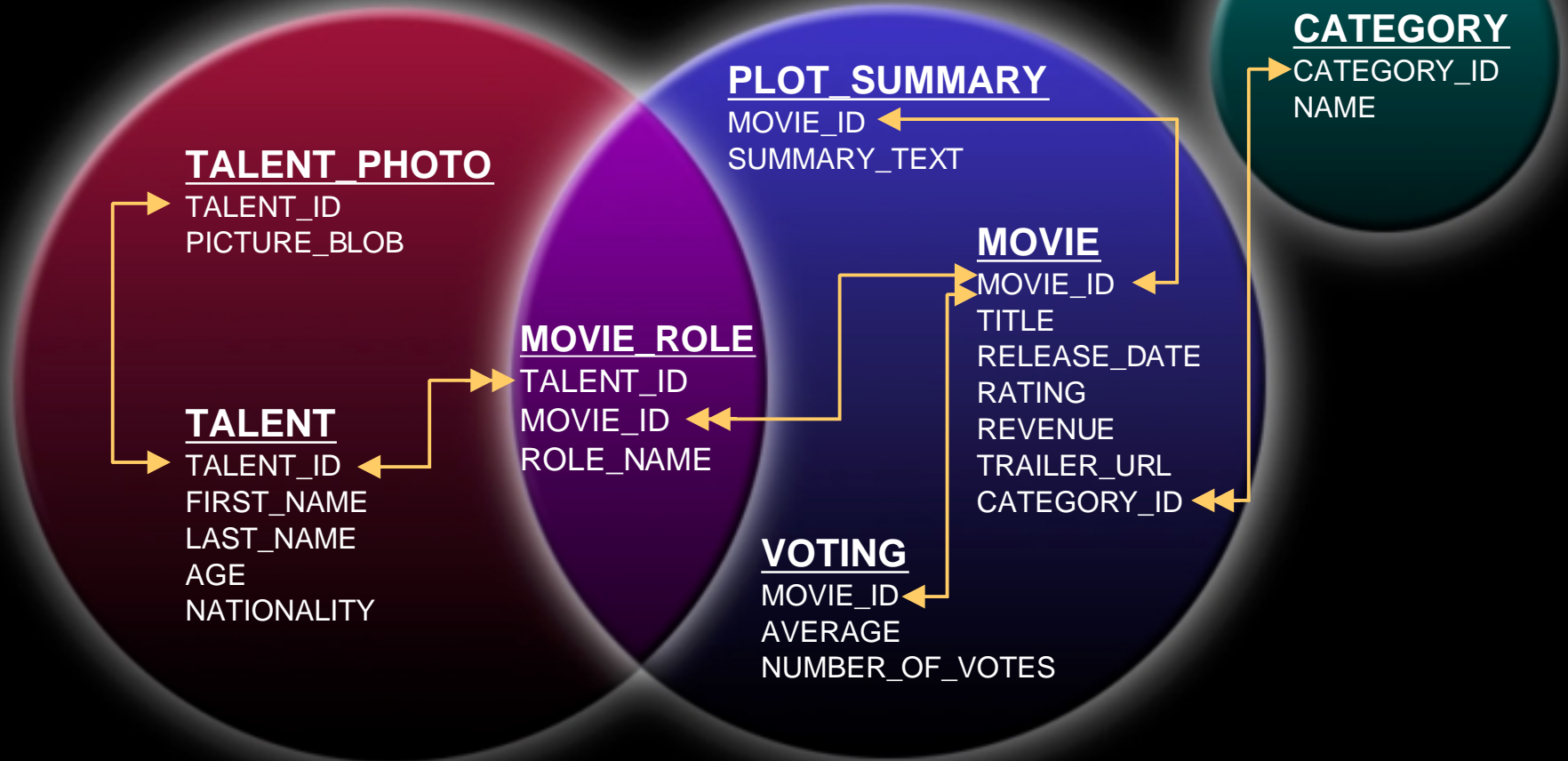


Server Side Rule System

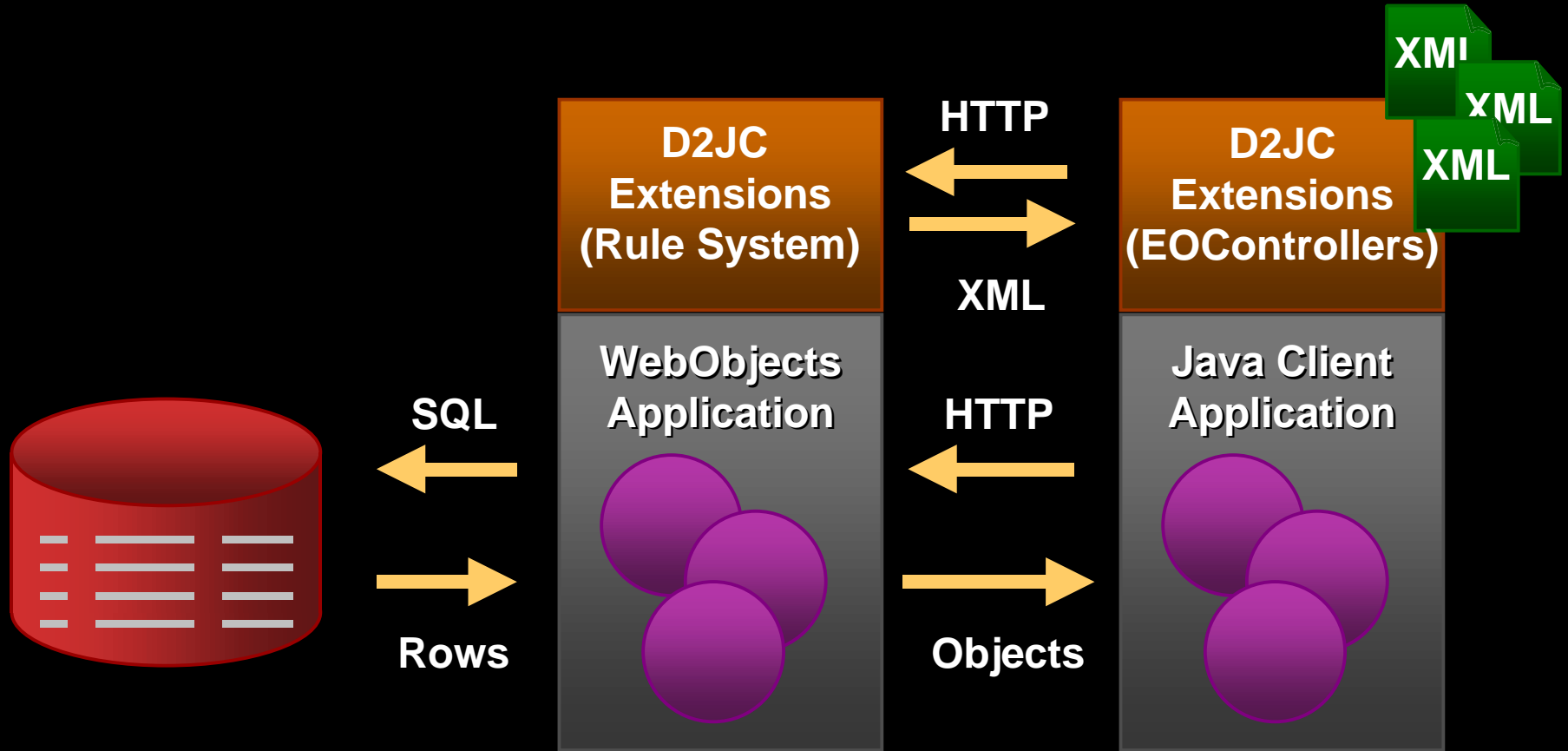
- Customizable rule system analyzes EOModel
- Default rules identify, categorize root level entities
 - Main entities
 - Query window
 - Form window with master/detail user interfaces
 - Modal dialogs for selection
 - Enumeration entities
 - Simple editor window
 - Combo boxes or other widgets for selection



EOModel Analysis

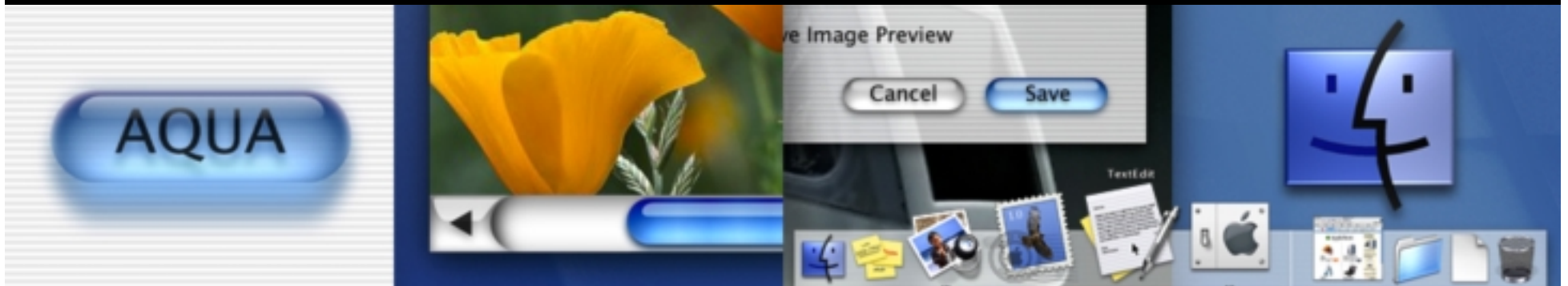


Direct to Java Client Architecture





DEMO



Development Process, Default Application

Customization

- Assistant
- Writing rules (Rule Editor)
- Freezing XML
- Implementing controller classes
- Freezing interface files (Interface Builder)
 - Smooth transition into traditional development



Customization

Rules

- Application logic changes (different kind of windows, controller interaction)
- Integration of custom controller classes
- Layout customization
- Multiple user environments (access permissions)



Customization

Freezing XML

- Bypass rule system
 - Layout customization
 - Integration of custom controller classes
 - Specification of parameters not supported by Assistant
 - Apply changes that are complicated to achieve in Assistant or Rule Editor



Customization

Implementing Controller Classes

- Integration of new functionality
- Support for new widgets/associations
- Application logic changes
(different interaction of controllers)
- Addition of problem/application specific functionality



Customization

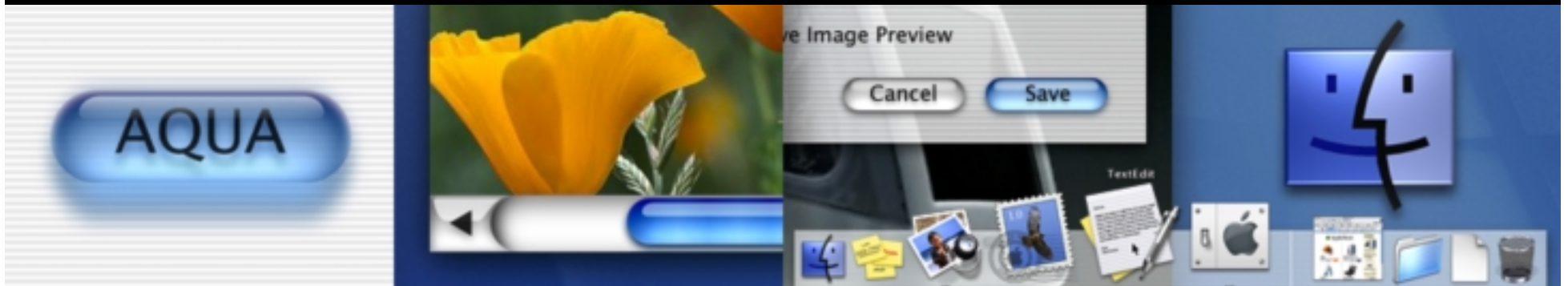
Freezing Interface Files

- Traditional development
- Maximum control over layout
 - Placement of views on pixel basis
- Integration with help of rule system
- High maintenance cost
 - EOModel changes not picked up automatically
 - Layout may be platform specific (multiple files)





DEMO



Customized Application

Benefits

- Fast (cheap) development
- Flattened learning curve
(only learn parts needed, one after the other)
- Quick reaction to requirement/database changes
- Reduced testing time
- Maintenance easier
- Focus on business logic (your core business)



Direct to Java Client Is Not a Toy!

- Speeds up the development process
- Makes development and maintenance easier
- Very extensible
- Supports different ways of customization for different needs
- Easy integration of traditionally developed parts
- Smooth transition into traditional development



What Kind of Applications?

- Database administration
- Movie database
- Order processing
- Address management
- And many more...



Summary

- Direct to Java Client enables rapid development and prototyping of desktop business applications
 - Dynamic generation of applications simply based on EOModel
 - Controller classes provide predefined, tested functionality
 - Powerful ways of customization
 - Smooth transition into traditional development



Roadmap

412 EOF Synchronization & Caching:

Client-Server done right

Room J2
Thurs., 2:00 p.m.

415 Advanced EOF:

Everything you ever wanted to know...

Room J2
Fri., 9:00 a.m.



For More Information

<http://www.apple.com/webobjects>

Visit the WebObjects lab downstairs!
Everyday from 11:00 a.m.–2:00 p.m.

Try out your WebObjects 4.5 Evaluation CD!

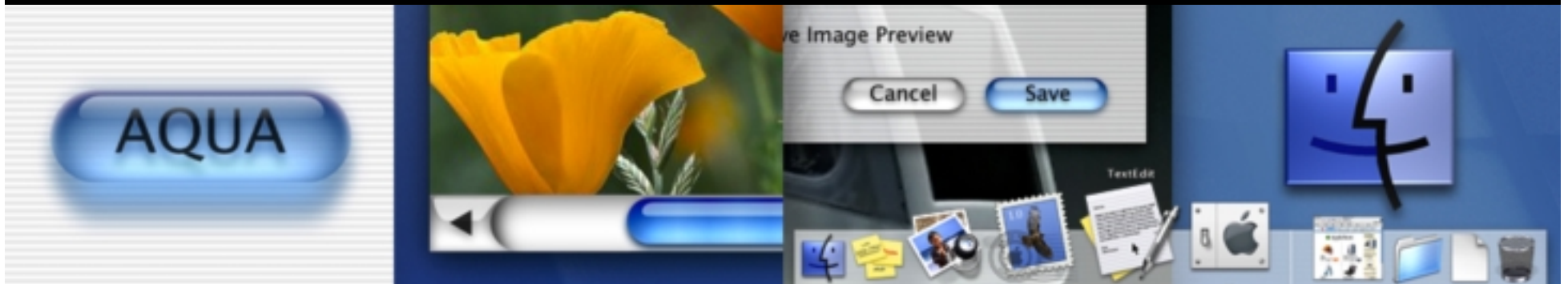
WebObjects Community BOF
Wed., 6:30 p.m.–8:00 p.m.





Session 406

Q&A



WebObjects Engineers

Who to Contact

Toni Trujillo Vian

Director, WebObjects Engineering
wofeedback@group.apple.com

Ernest Prabhakar

Product Line Manager, WebObjects
webobjects@group.apple.com





WWDC

Worldwide Developers Conference 2000



Think different.