



Building Multi-Channel Applications using Frameworks

Jörg Jungjohann, Systor AG



Building Multi-Channel Applications using Frameworks

Business Case

Technical Issues

Demonstration

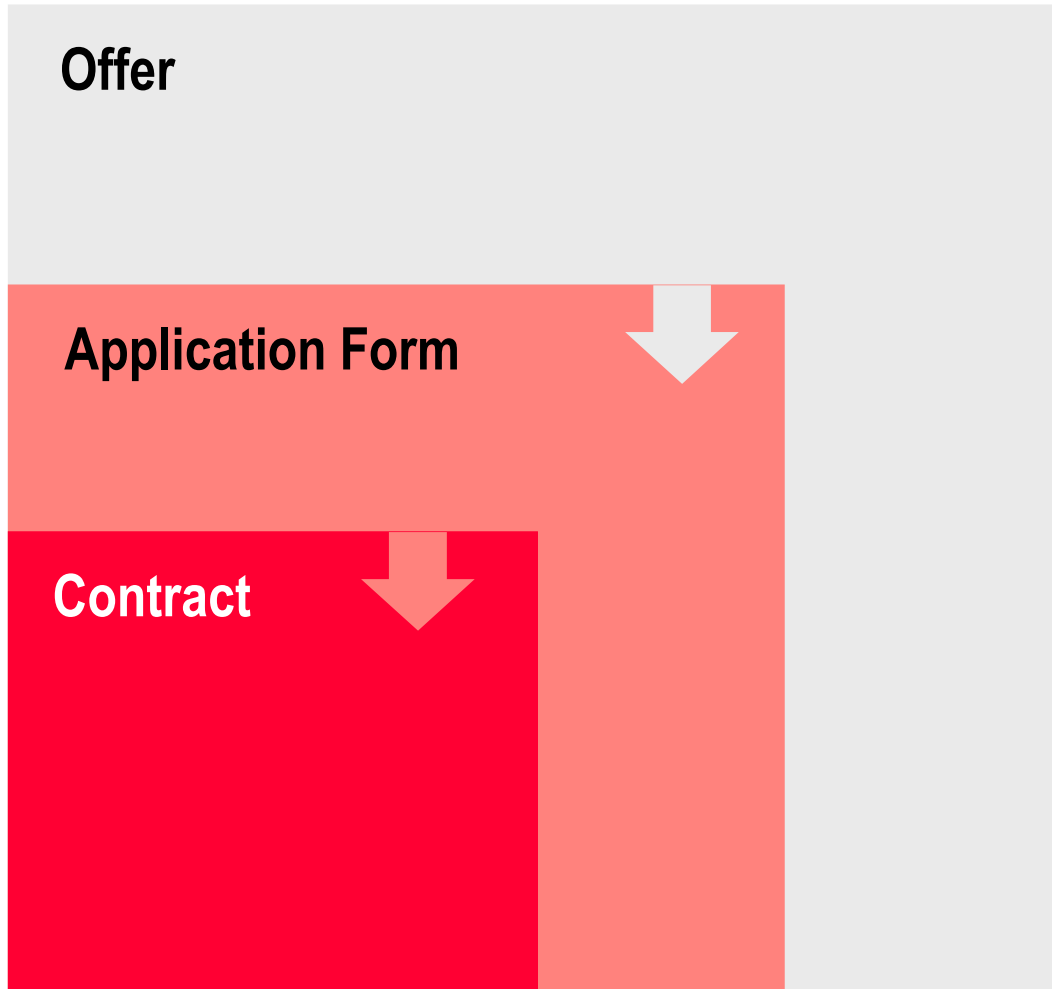


Business Case

Why multichannel?

- Customers want the same service on all channels
 - integration of existing applications
 - multi-channel enabling
 - new implementation of applications
- Example: insurance application
 - internet
 - WAP
 - fat client
 - data and contracts on host

Business Process Model



Insurance Prototype

SYSTOR AG - Insurance Prototype 0.5

Datei Bearbeiten Sprache Fenster Hilfe

Prozessauswahl

- Kunde
- Angebot
- Antrag
- Vertrag

Gesundheitsfragen

Partner Gesundheitsdaten

Aids Test positiv

Arbeitsfähig

Medikamente regelm.

Grösse

Gewicht

Letzter Arztbesuch

Letzte Operation

Arzt

Titel **Herr**

Name

Vorname

Arzt Adresse

Strasse

Land/PLZ/Ort **Schweiz**

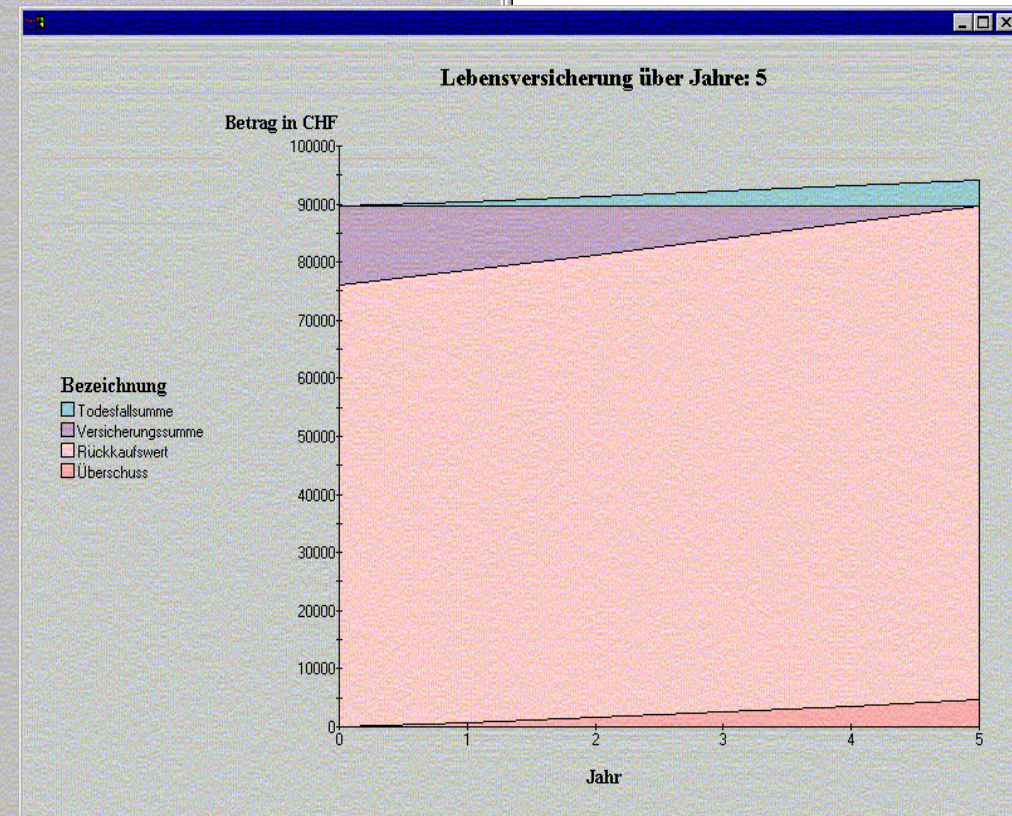
Telefon

Fax

eMail

Bereit

2.11.1999



Systor Solution

Systor Insurance Prototype

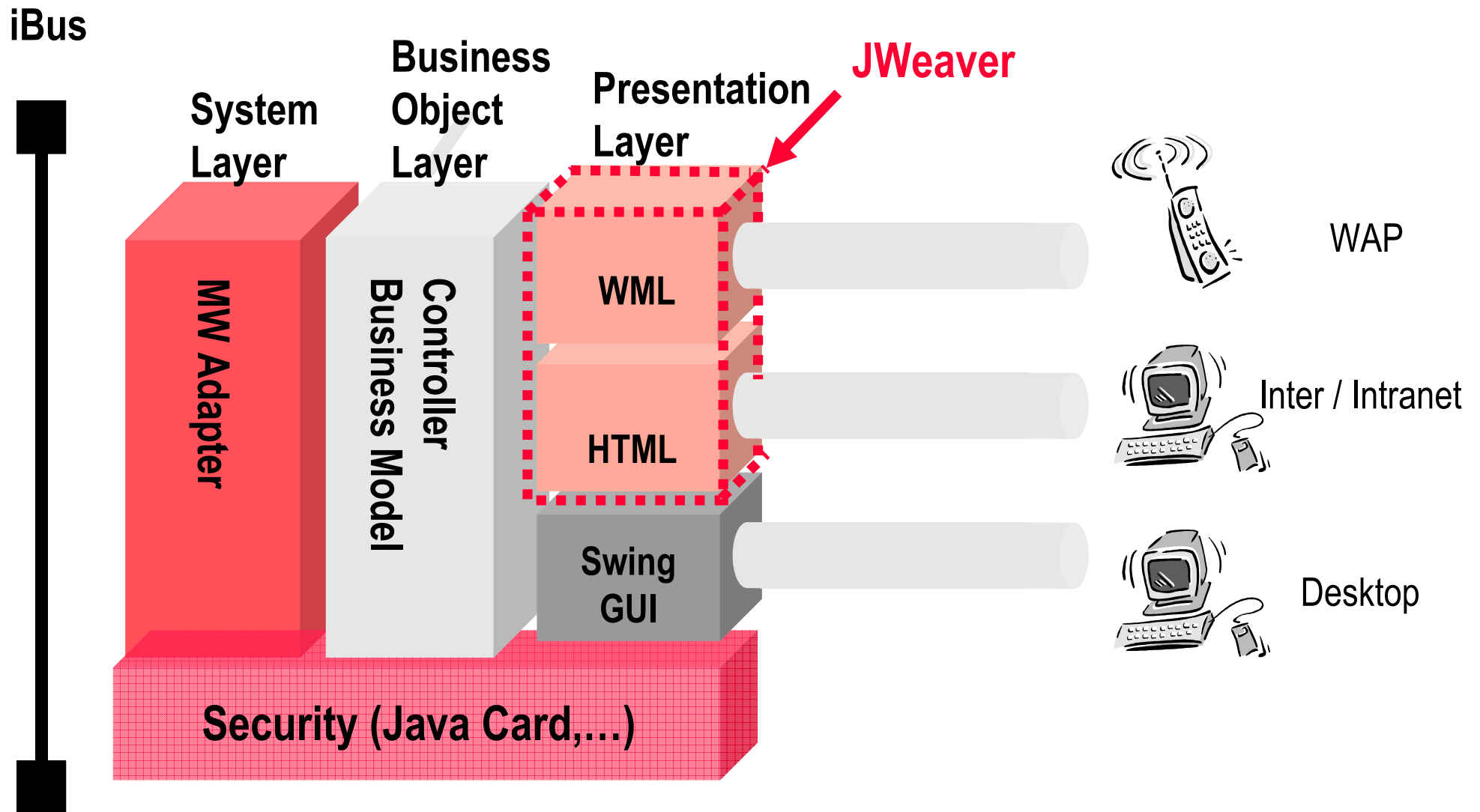
JWeaver

Servlet Framework for multichannel-enabling



Multi-Channel Insurance Case

Client





Technical Issues

Contents

- What is common, what is different between the channels
- Problems with HTML / WAP User Interfaces
- HTML / WML generation
- How to use JSP / Servlets / Templates
- Generating images on the server with Java
- Navigation, session handling

Desktop vs Web

- “Fat Client” vs web application
 - Web applications are entirely server based
 - All objects for all users are on the server
 - Web User Interfaces are different
 - Many problems due to browser
 - different standards
 - designed for static content not for applications
 - designed for display rather than editing data

What is common / different

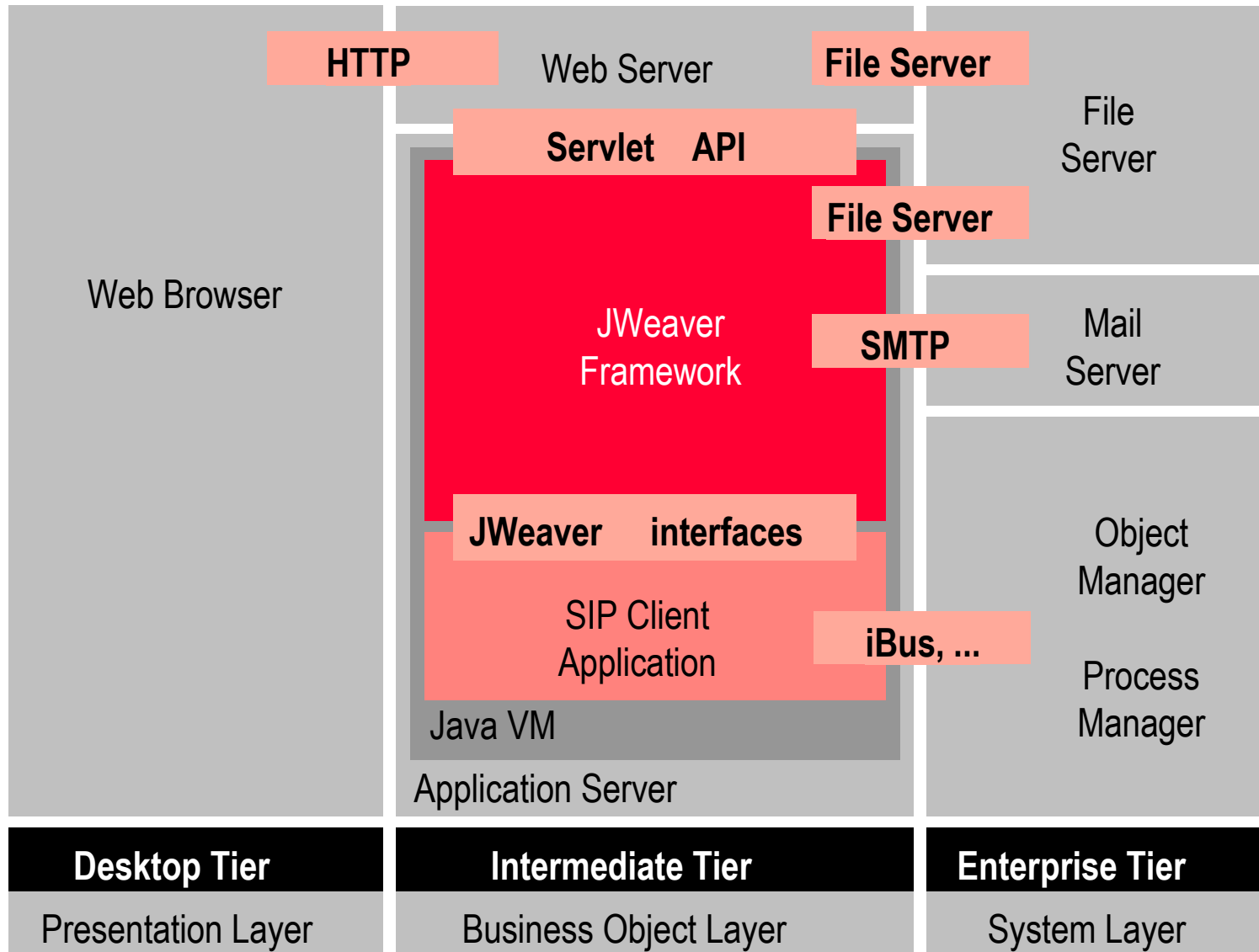
- UI and functionality will vary on the channel
- Mobile vs Browser
 - HTML and WML have different GUI components
 - “Card” and “Deck” cannot be used in HTML
- Business logic must be unique due to customer expectations
 - Use Business Object Layer for all channels
- Use controller for identical functionality on both channels
- Re-implementation of view-component for each channel
- Servlet API just Java API to web server
 - “request string” in, “response string” out

The Framework: JWeaver

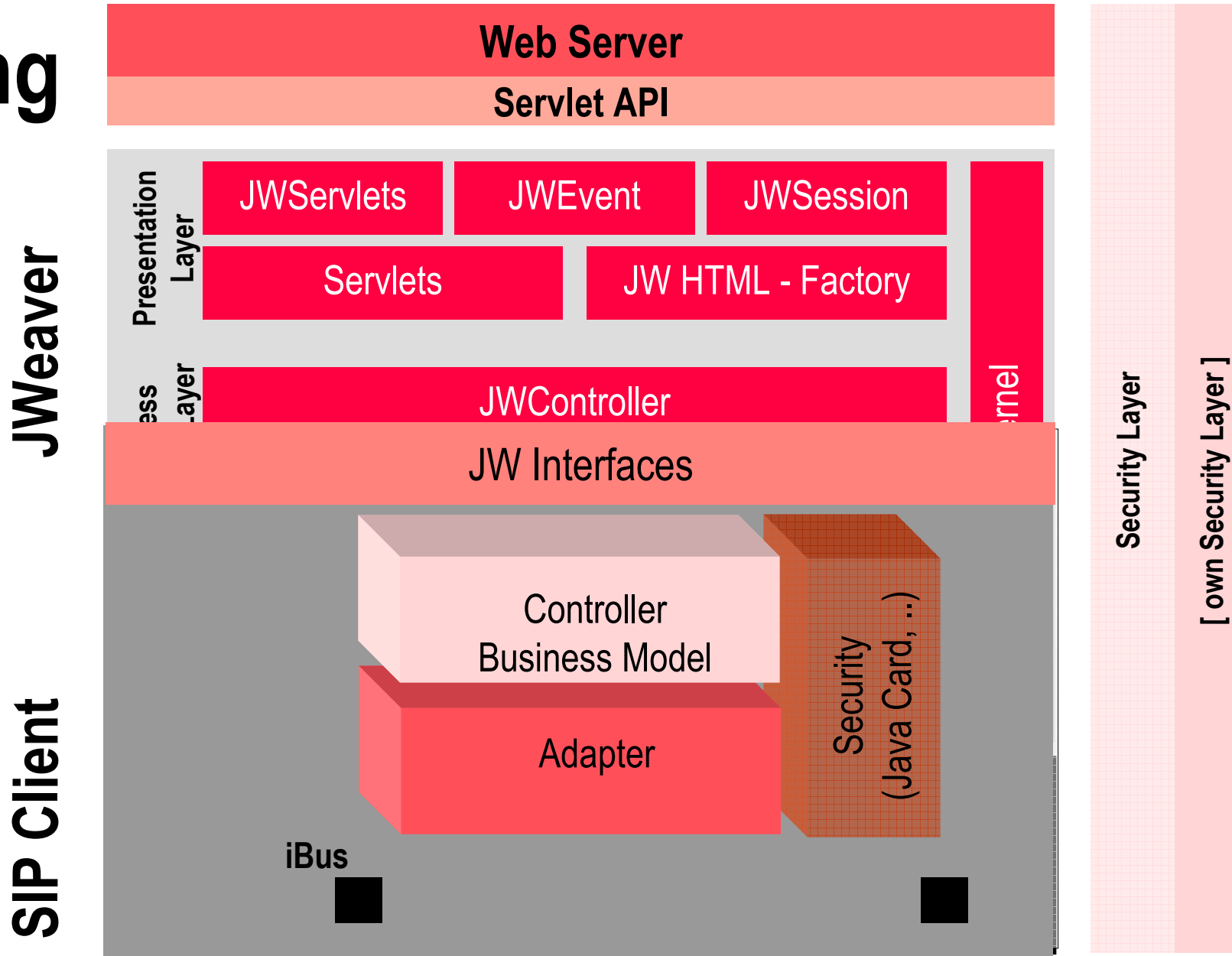
- A Knowledge Framework
 - Architecture
 - Servlet course

- A Java Software Framework
 - Java Software Framework built on the Java Servlet API
 - Based on experiences from several projects in Intranet and Internet and WAP
 - Focused on web and GUI problems

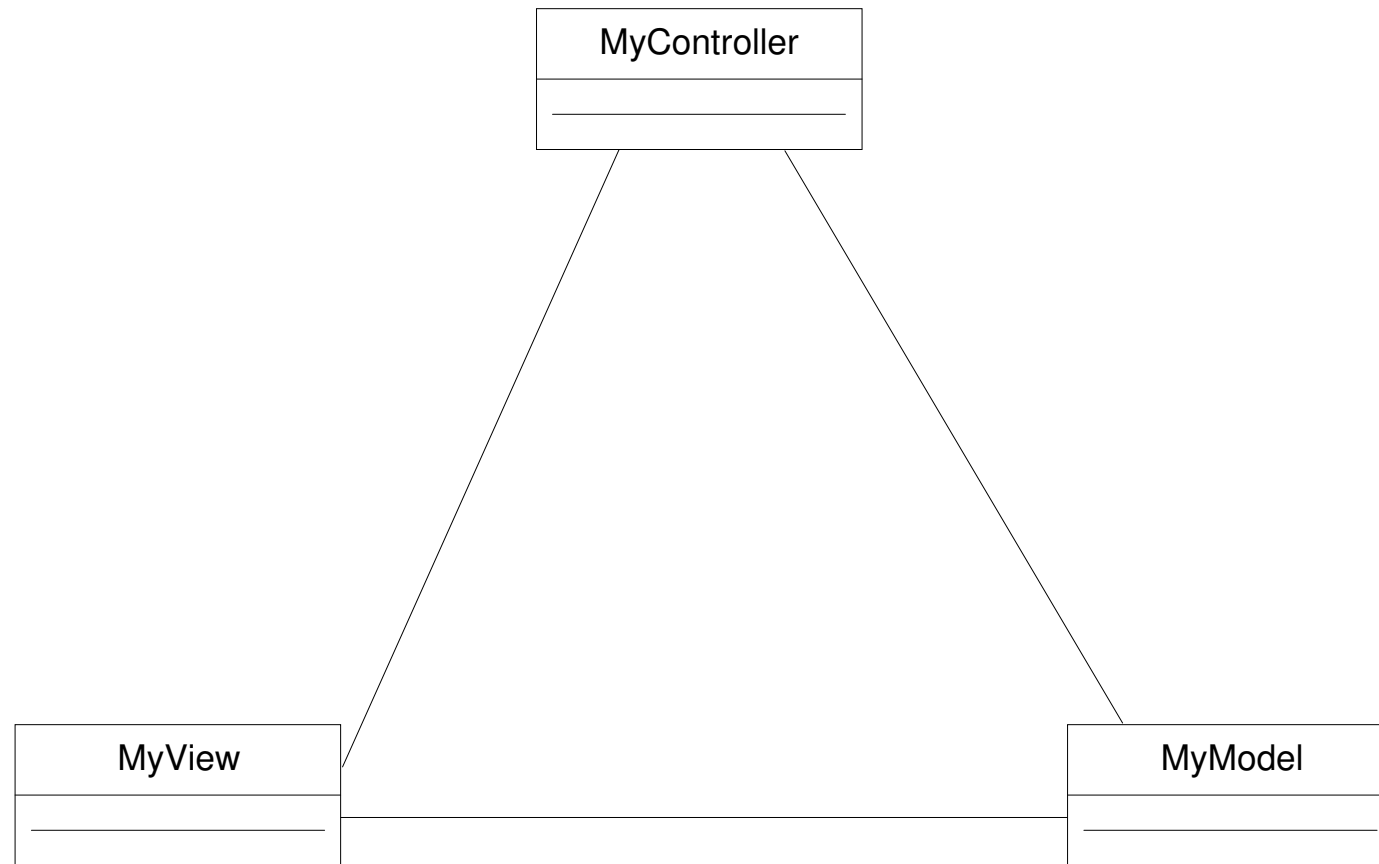
3 Tier Architecture



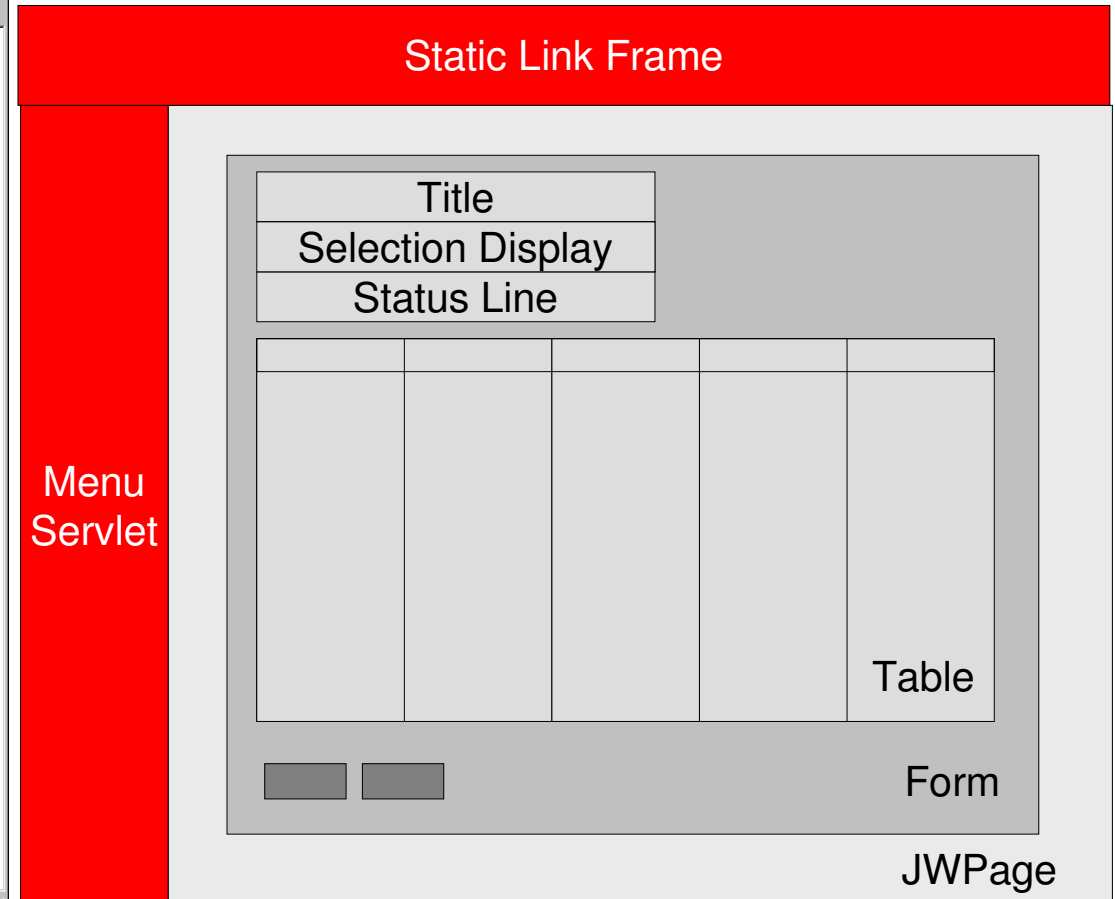
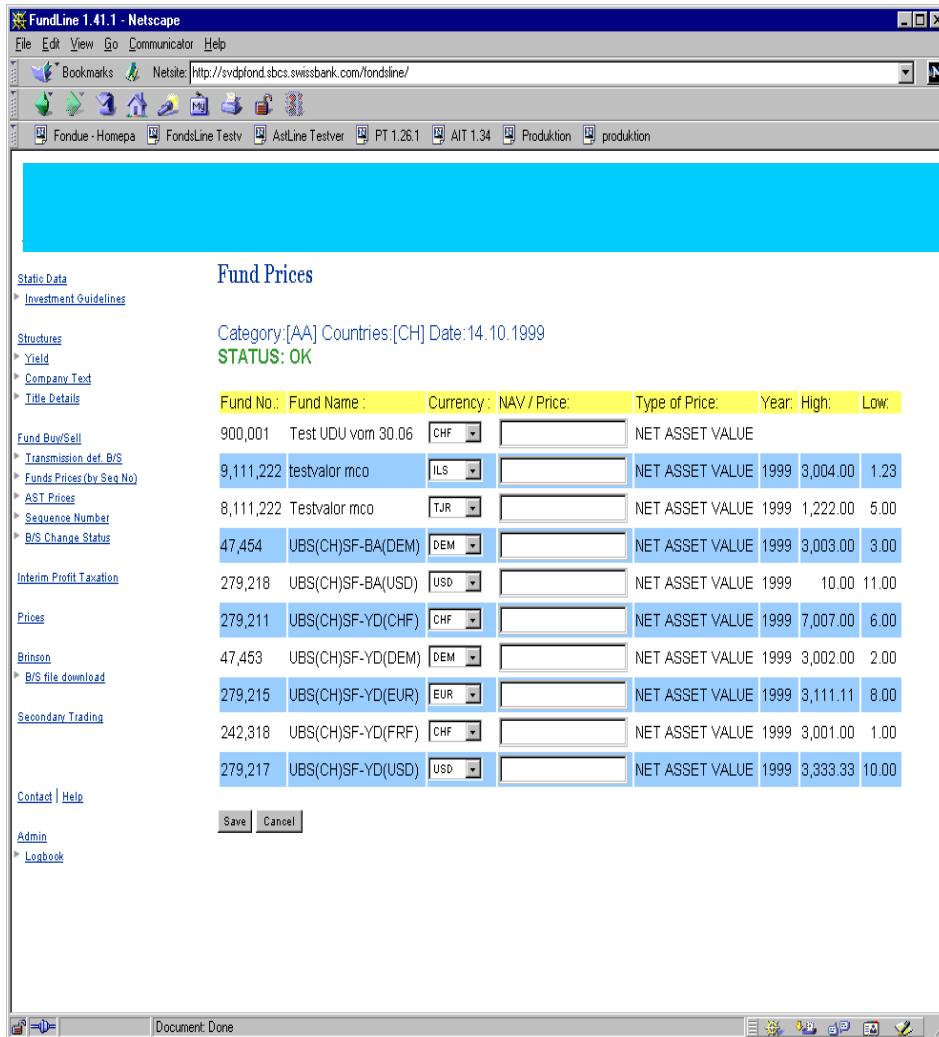
Layering



MVC Design



Composite pattern



View implementation 1

- Generation of HTML / WML with HTML / WML libraries
 - Reuse and inheritance of GUI components
 - Ideal for complex repeating user interfaces (HTML)
 - Simple to use for developers
 - Checked at compile time
 - IDE can be used

View Implementation 2

- Template engines: e.g. WebMacro
 - Change of design without compiler
 - Limited syntax
 - No tools
 - Special scripting syntax
- Java Server Pages
 - Easy to change also by designers
 - Tools to come for syntax-check or integrated development
 - Standard
 - “quick & dirty” possible

Reference Projects

- Used in and based on experiences from the following projects:
 - www.fundgate.ubs.com
 - 50'000 Intranet users
 - 15'000 Internet users (hits per day)
 - Systor Web Shop
 - 50'000 users
 - Person Information System (Kanton Basel Stadt)
 - 3000 users
 - Several other inter / intranet applications

Issues 1

- Session tracking
 - Cookies
 - can be disabled on browser
 - URL-rewriting
 - use secure session ID
 - only on intranet
- Navigation in applications
 - What does „reload“ mean when you just bought a book
 - What means „back“?
 - How to handle bookmarks
 - Session timeout

Issues 2

- Number of servlets
 - One servlet per page
 - One servlet per application
- Cache and proxies
 - Header tag “no cache”
 - Header tag “expires”
- Multithreading and testing
 - Servlets are multithreaded
 - Take care in design
 - Testing, testing, testing (with test tool)

Issues 3

- GET or POST
 - GET displays parameters in URL
 - POST asks for “repost data” on “back” etc.
- Frames
 - For internet usually not wanted
 - Useful in the intranet
 - Use includes / templates for frameset

Dynamic Image Generation

- Use Java for image generation (e.g. charts) maintains portability
 - Performance is reasonable
 - Class libraries are available
 - Advanced Imaging API has many features
- JDK 1.2
 - `System.getOffScreenImage()`
- JDK 1.1.x
 - Graphics Object needed
 - `frame.addNotify()` creates peer object
 - On UNIX XServer necessary
 - Use Xvfb



Demonstration Insurance Prototype

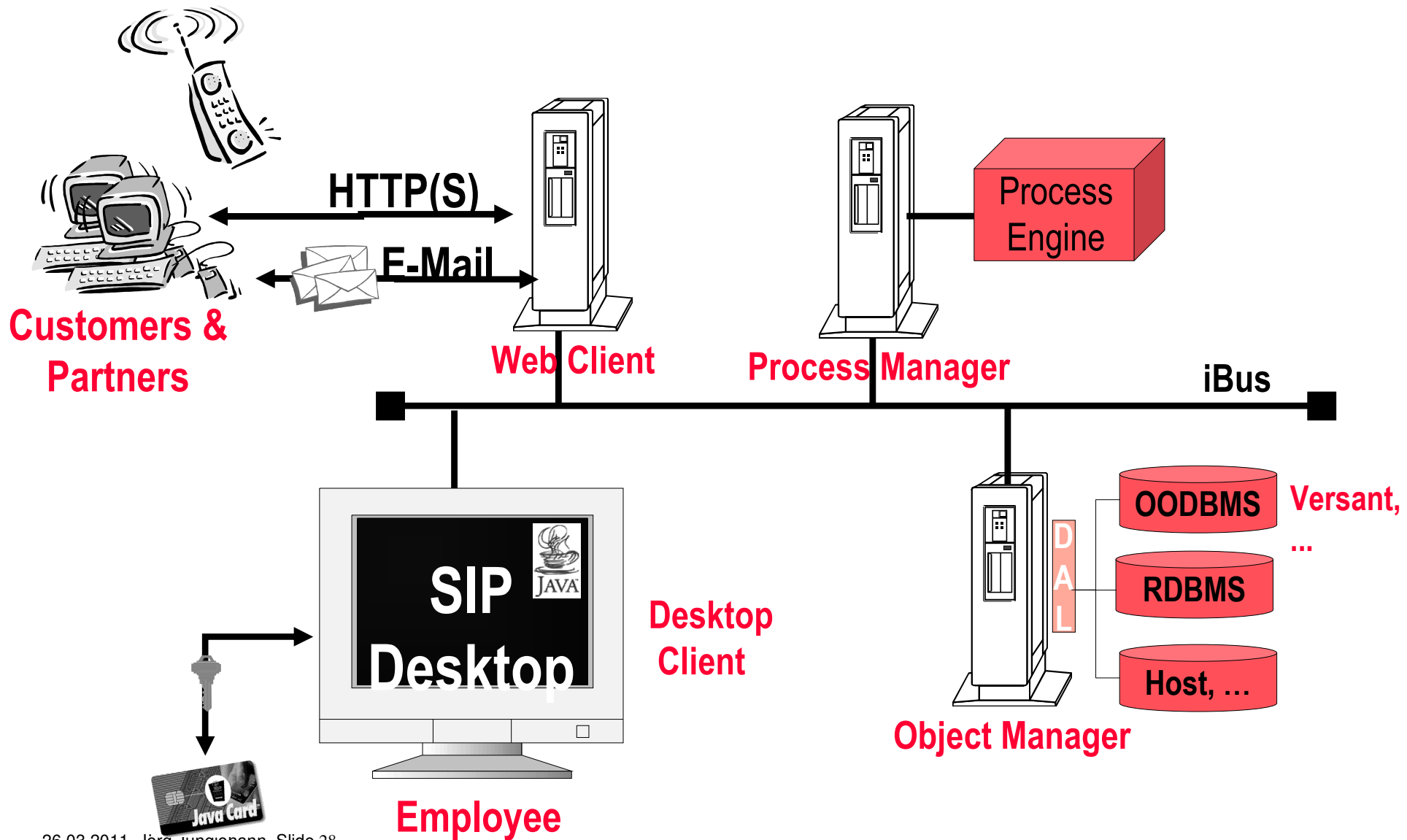
Showcase

- Life insurance
- Core process
 - Client data
 - Pricing
 - Contract

Features

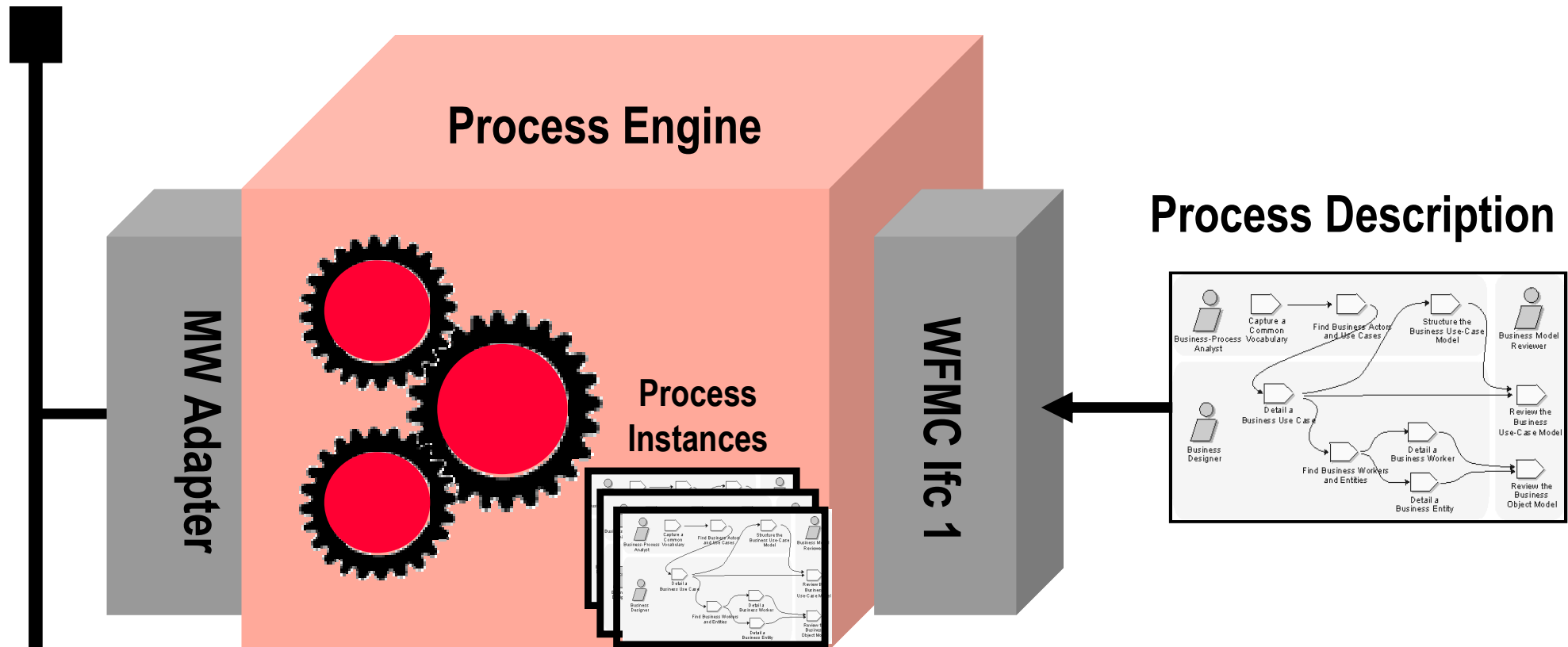
- Pure Java (Portable)
- Scaleable
- Open and Flexible
- Interoperable
- State of the Art
- Multichannel Enabled
- Component oriented
- VAA 3.0 based

System Architecture



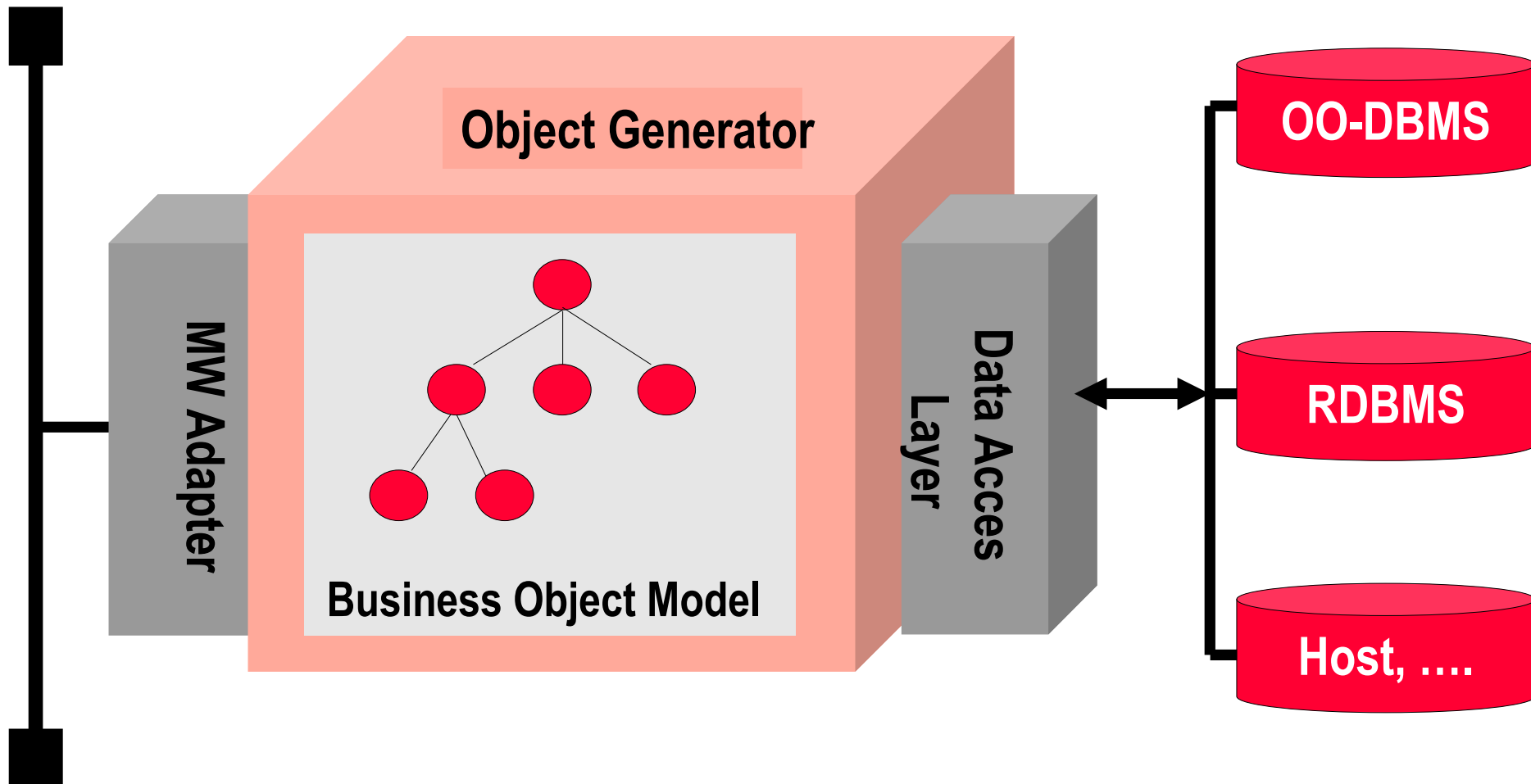
Process Manager

iBus



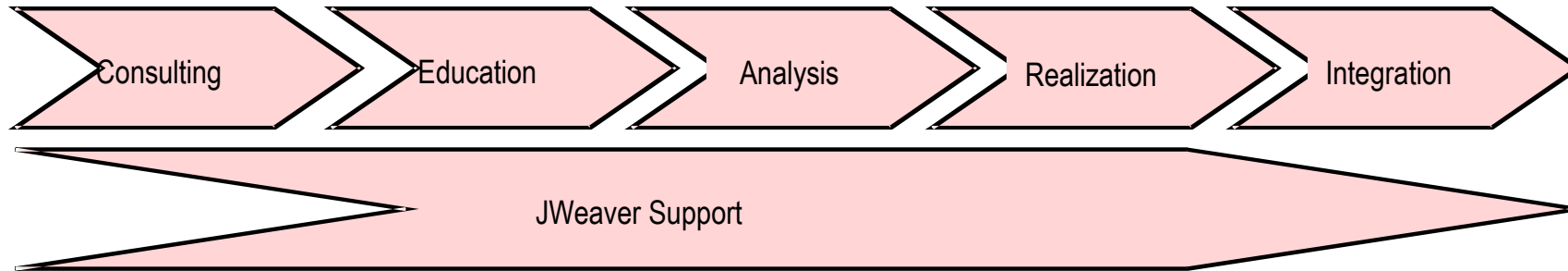
Object Manager

iBus





**Thanks for your
Interest**



■ Consulting

- Experienced JWeaver Architects
- OTC Architectural Blueprint

■ Education

- JWeaver / Servlet course
 - 2 days
 - Examples
 - Practical Training
- JWeaver CookBook

■ Analysis

- PULSE
- RAW
- Prototyping

■ Realization

- > 20 Java Engineers

■ Integration

- Experiences in heterogenous system-environments
- Systors CC's