



OOPSLA 2000 Practitioner Report – Internet Applications
Tuesday October 17, 2000 3:30 – 5:00 PM

Porting a Core CGI Workflow System to a WAS Platform While Being Used and Extended to Support the Enterprise

Porting a Core CGI Workflow System Onto A Web Application Server Platform While Being Used and Extended To Support the Enterprise

David J. Arnone
PaineWebber
darnone@painewebber.com

10/17/2000

David J. Arnone
PaineWebber

1

- ❶ **The Application and Platform Migration**
- ❷ **The Framework**
- ❸ **Development Team and Environment**
- ❹ **Future Work**
- ❺ **Observations**
- ❻ **Questions**



The Application

- Supports the retail business of PaineWebber.
- Manages broker additions, changes and deletes.
- Manages application access and entitlements.
- Automates system account creation.
- Provides an approval process.

The Challenge

- Migrate the application onto a platform that scales.
- Extend domain to handle the enterprise.
- Add new functionality.
- “Modernize” the UI look and feel.
- Deliver every quarter.



The Original System



10/17/2000

David J. Arnone
PaineWebber

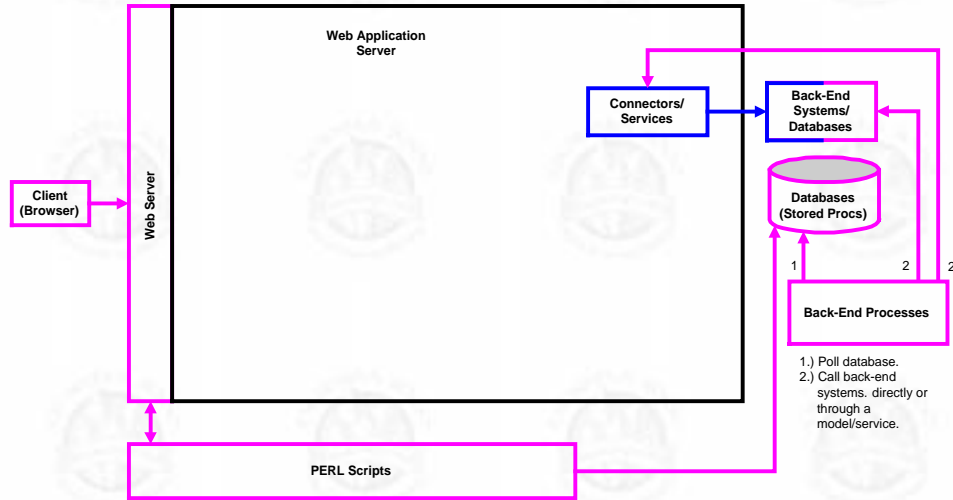
3

In describing the original system:

- The user enters input
- Upon completion the user submits the information and the system places a record into the database.
- The back-end daemon process polls the database for new records.
- When one is detected the action of creating accounts begins occasionally stopping for business approvals, if necessary.



The New Platform – Current System in Production



10/17/2000

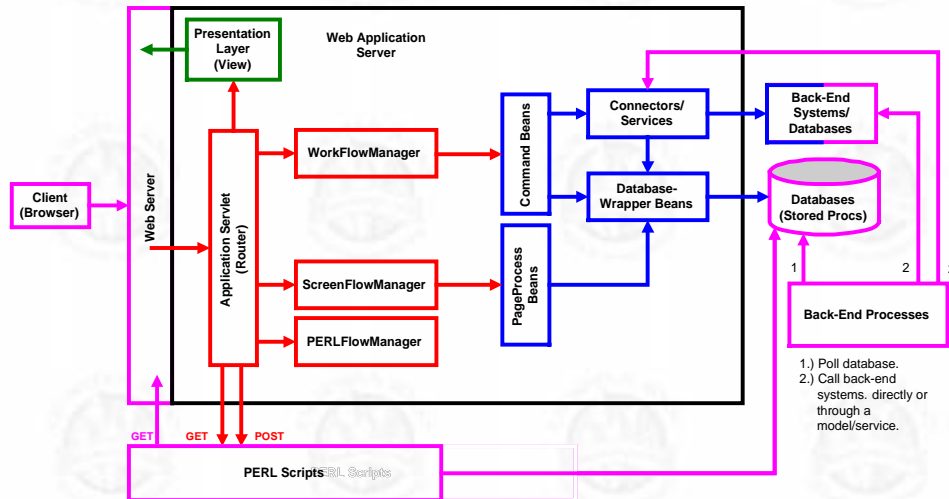
David J. Arnone
PaineWebber

4

- The current production system only ties Connectors and Services to the original system
- Connectors and Services are referred to as “horizontal slices” in the paper.
- This is how we began the transformation/migration of the platform.



The New Platform



10/17/2000

David J. Arnone
PaineWebber

5

- The new system introduces a web application server platform.
- An Application Servlet routes requests via URL redirection and understands what success/failure means in the application space.
- Enterprise Java Beans insulate the application from back-end connectors, services and databases and wrap database calls.
- Managers understand:
 - what command beans to interact with from configuration information (WorkFlowManager).
 - what screens come next (ScreenFlowManager).
- This architecture allows us to move and add front-end features as well as back-end system connections transparently.



Logical Architecture Migration - The Original System



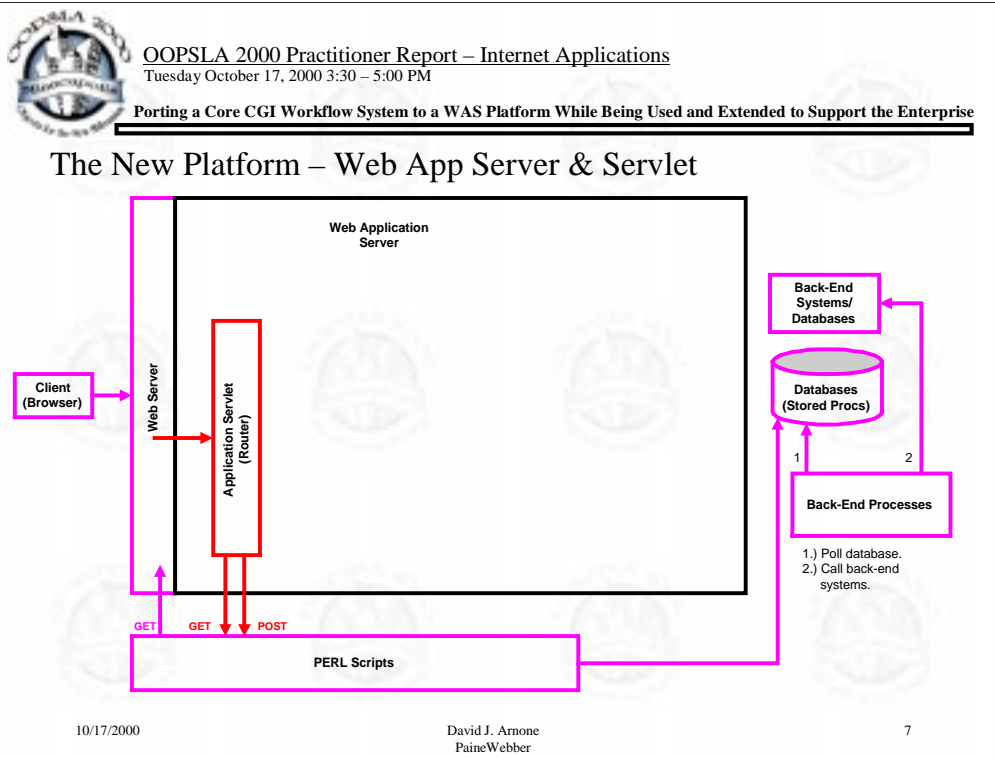
10/17/2000

David J. Arnone
PaineWebber

6

What I want to do is quickly step through the migration to describe how the framework works.

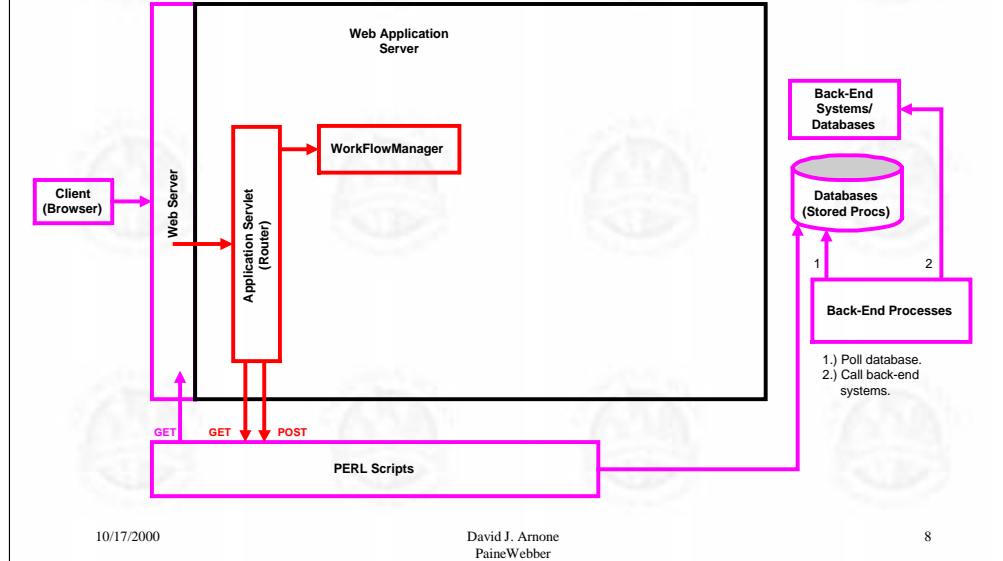
Given the original system.....



- We introduce the web application server platform and intercept all web requests with an Application Servlet.
- The information from the front-end is passed as a DataBundle – a name/value pair object that also contains request information as well as success/failure results.
- Upon server start-up the Servlet reads configuration information that provides application specific information like configuration files for the various managers, initial context, servers, etc.
- A part of its own initialization, the Servlet initializes the Managers and then passes the DataBundle to the.....

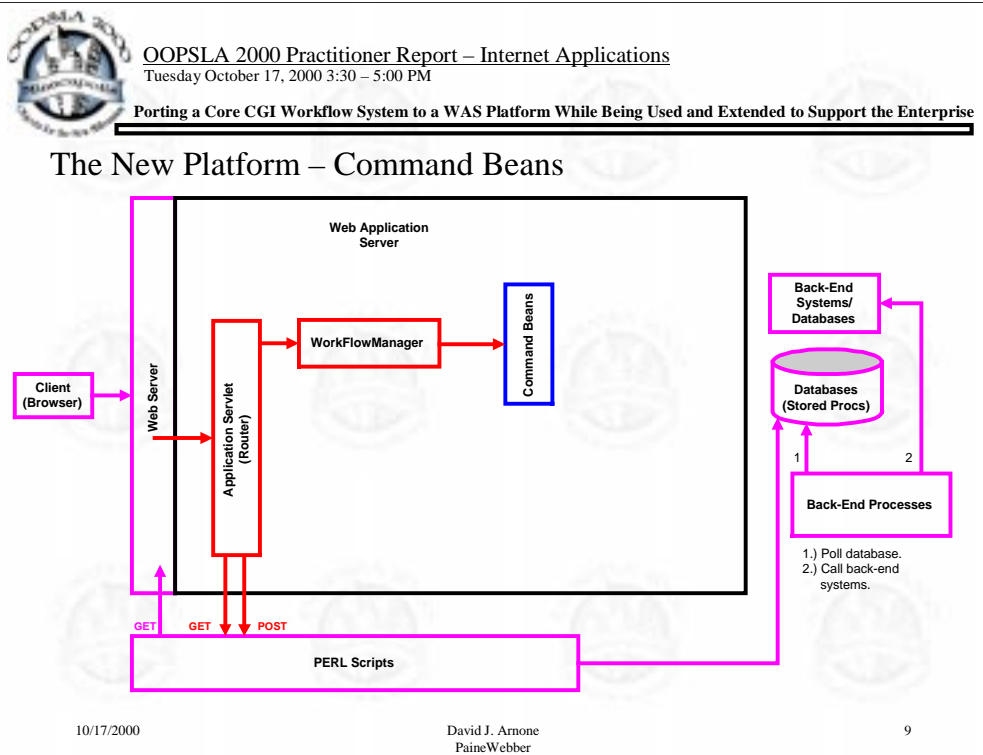


The New Platform - WorkFlowManager



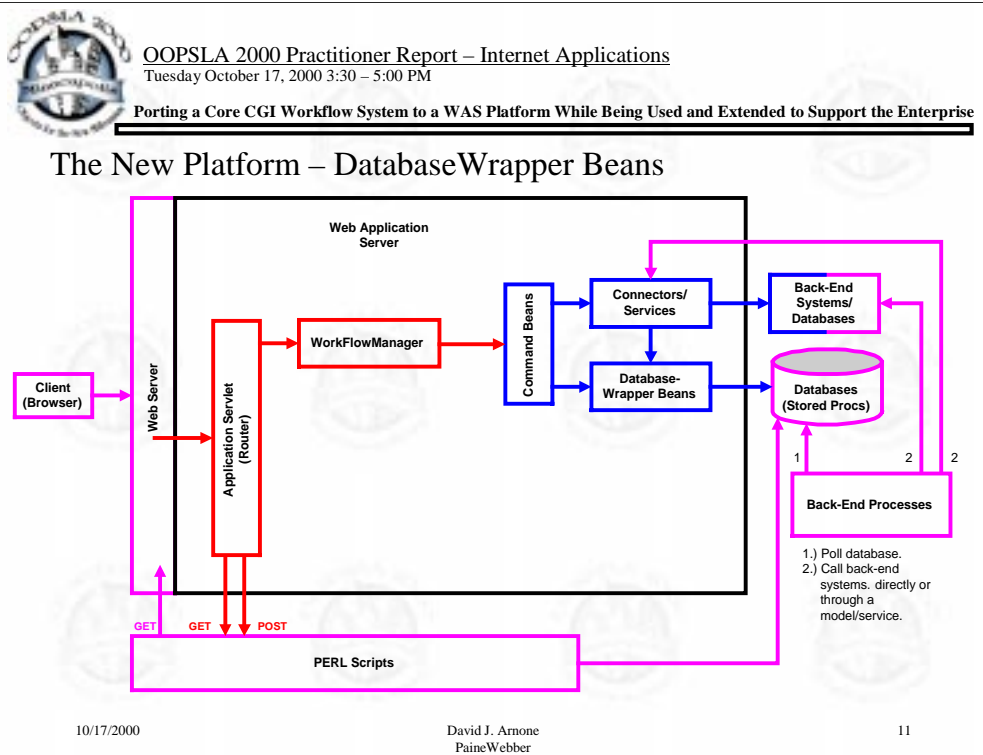
WorkFlow Manager.

- The WorkFlowManager interrogates the DataBundle to determine what action it should perform.
- Configuration information for the WorkFlowManager specifies various commands and identifies what Command Beans it must call and in what order.
- Given the command, the WorkFlowManager passes the DataBundle to the appropriate.....



Command Bean.

- **Command Beans are stateless session beans that contain business logic.**
- **Command Beans extend a framework specific SessionBean (FMW_SessionBean) and implement the Command interface.**
- **The Command interface has one method called processRequest(DataBundle databndl)**
- **At this point the Command Bean either calls a**

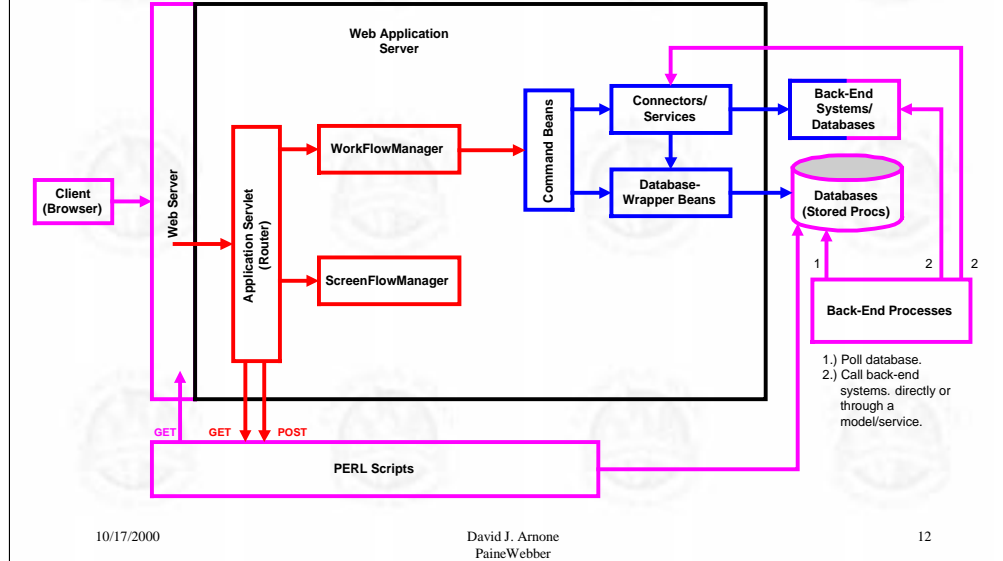


Pass the DataBundle in an Accessor to a DatabaseWrapper Bean.

- Accessors encapsulate the DataBundle and provide context and get/set operations for DataBundle manipulation.
- DatabaseWrapper Beans are stateless session beans that execute stored procedures and return the results via a PersistentResult object (result set cache).
- The Command Bean performs internal business logic based upon returned information in the PersistentResult, modifies the DataBundle via the Accessor and returns the DataBundle back to the Servlet via the WorkFlowManager.
- The Servlet will interrogate the DataBundle for success or failure and call the.....



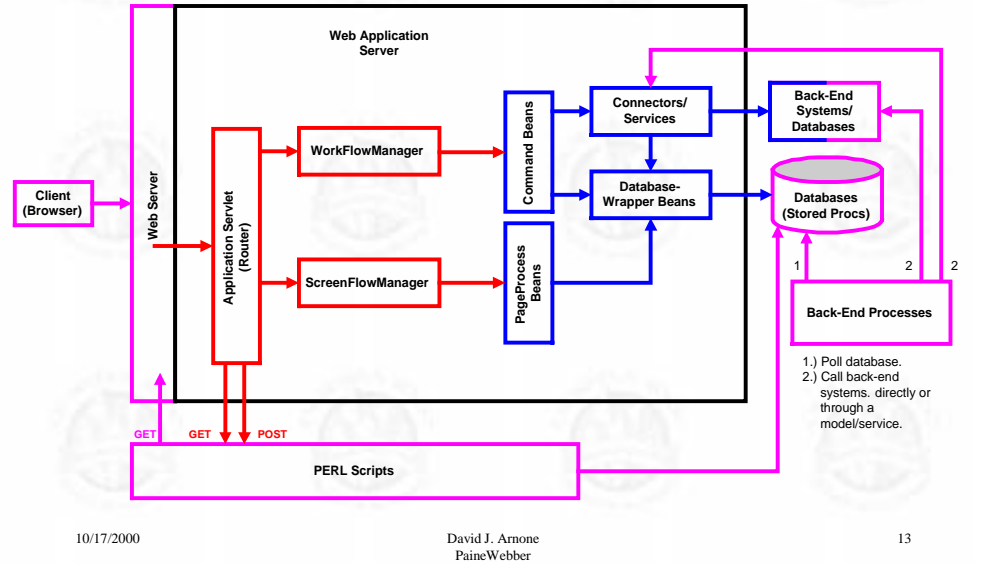
The New Platform – The ScreenFlowManager



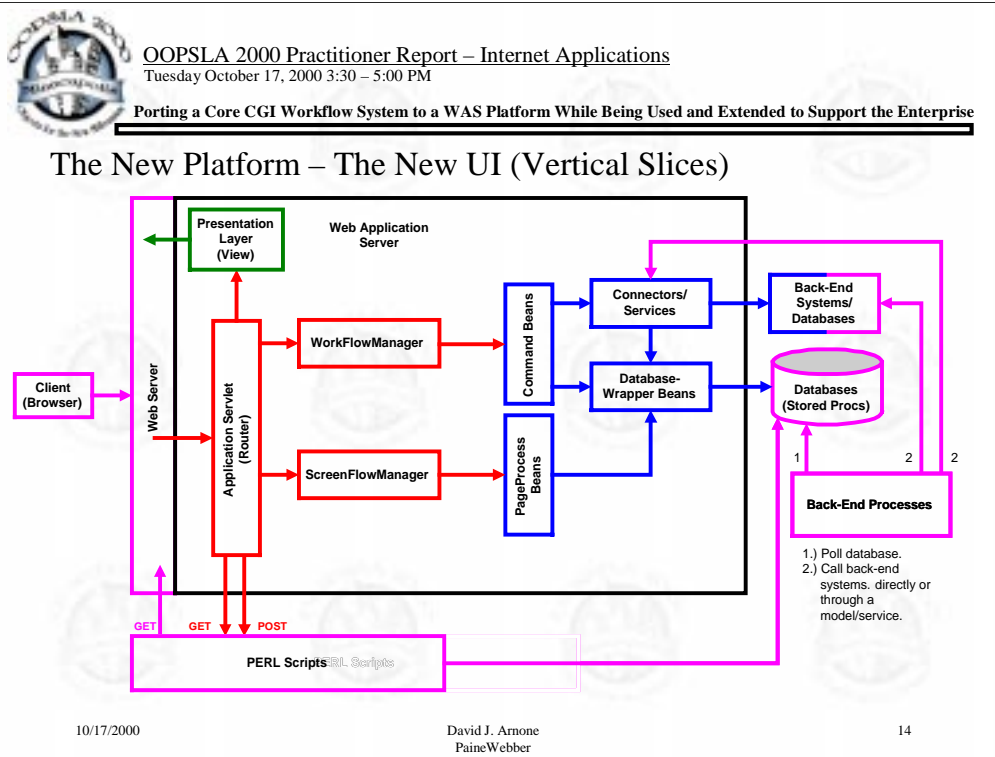
- ScreenFlowManager to provide the next JSP or indicate what PERL code should be called.
- Again, the ScreenFlowManager understands what to provide from its configuration data.



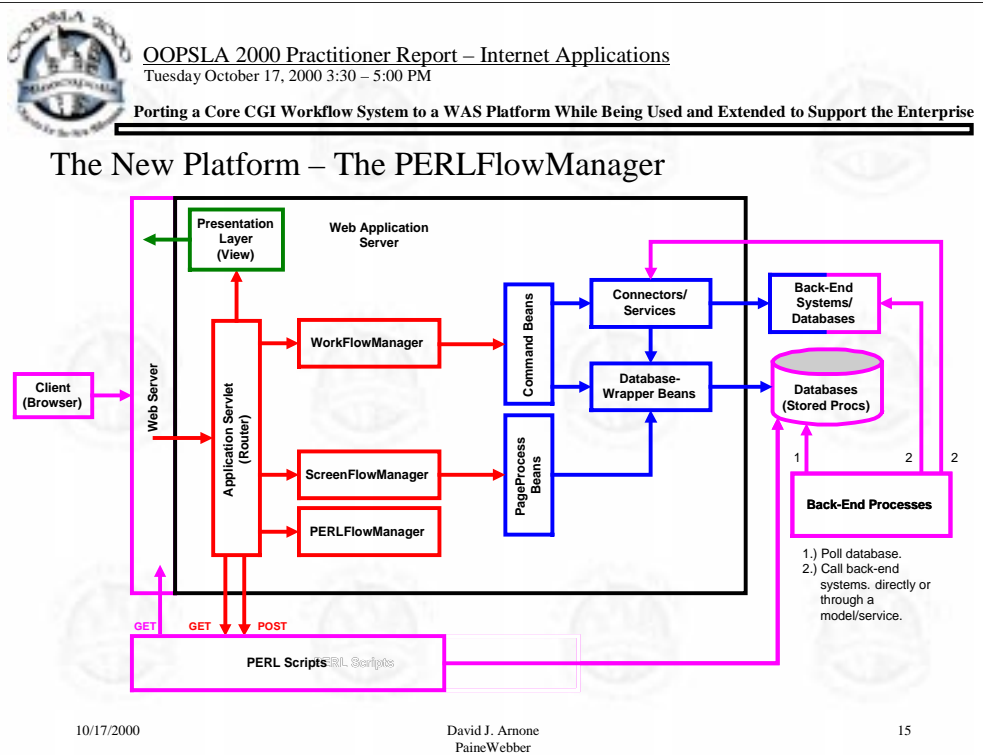
The New Platform – PageProcess Beans



- If *valid values* are required for the JSP the information is obtained through a DatabaseWrapper Bean via an insulating PageProcess Bean.
- Again, everything is passed as a DataBundle or Accessor.



- Processing then returns to the Servlet which returns the JSP to the client.
- If the feature is in PERL, the Servlet calls the.....

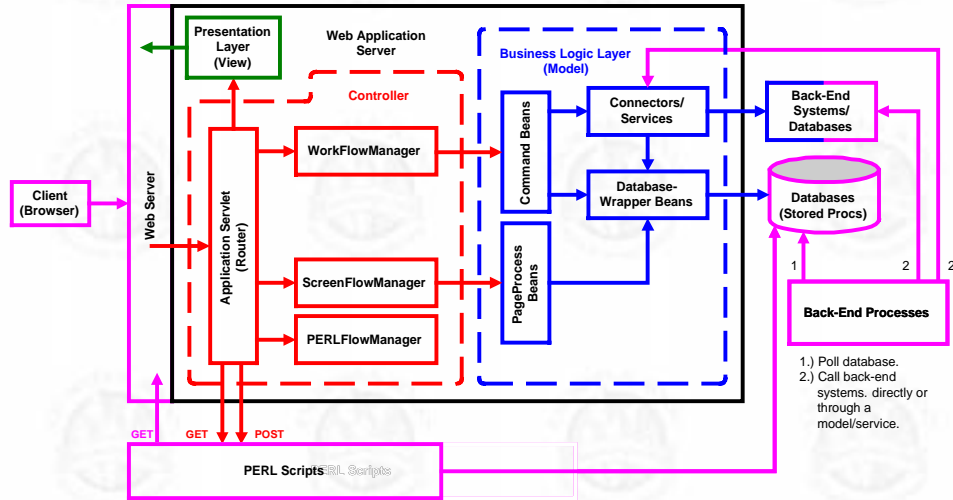


PERLFlowManager which squashes the PERL query.

- The Servlet then calls the PERL code with a post for large data sets and a get for small data sets.
- Posts are handed back to the servlet while gets write directly back to the client.



The New Platform – MVC Design Pattern



10/17/2000

David J. Arnone
PaineWebber

16

- If you partition the pieces you will see a Model/View/Controller architecture on top of a 4th tier.
- The Command Bean and PageProcess Beans insulate the application from changes in the back-end.
- The Database Wrapper Beans insulate the details of the database from the Command and PageProcess Beans.



The Framework Pieces

☉ Application Specific Elements

- ☉ Application Servlet.
- ☉ Database and Stored Procedures.

☉ Configurable and Reusable Elements

- ☉ WorkFlowManager.
- ☉ ScreenFlowManager.
- ☉ *LanguageBaseFlowManager*.
- ☉ DataBundle - information passed as attribute/value pairs with state.

☉ Extendable Elements

- ☉ Accessor Base Class.
- ☉ FMW_SessionBean extends SessionBean implements Command.
- ☉ FMW_DBSessionBean extends SessionBean.

- **FMW_SessionBean extends SessionBean and implements the Command interface.**
- **FMW_SessionBean adds logging and home lookup methods.**
- **FMW_DBSessionBean adds methods for database connection and access.**
- **The Command interface has one method – processRequest(DataBundle databundle)**



Framework Development

- Paired Programming and Iterative Approach
 - Daily design, investigation and implementation cycles.
 - Proof of concept and prototype developed in pairs – 2, 4, then 5
 - Weekly design, investigation and implementation cycles.

- The framework is more than just the code.....
 - Programming standards.
 - Logging and exception handling rules.
 - *Developed a Cookbook.*

•Logging is asynchronous.



Progress to Date

☉ Back-end Services (Horizontal Slices)

- ☉ 1 new service implemented.
- ☉ 1 service re-written.
- ☉ 3 new connectors implemented (new features).
- ☉ 4th connector completion 4Q2000 and enhancements to others.

☉ Front-end Migration (Vertical Slices)

- ☉ 2 original screen sets migrated.
- ☉ 3 new screen sets and components implemented (new features).
- ☉ A work in progress.



The Process

- Process Elements – XP??
 - Simple Design
 - Use Cases
 - Pair Programming *Dialect*
 - Testing (JUnit)
 - Integration
- Going Forward
 - Refactoring
 - Review and iterate.....



Development Team Composition

- 2 Architects
- 4 Lead Developers
 - 3 Framework & Development
 - 1 UI Design/Web Enablement
- 2 Senior Developers
- 5 *Junior* Developers (16, 15, 13, and two 4 months)
- Attrition
 - 2 Lead Developers
 - 1 Senior Developer
 - 1 *Junior* Developer

10/17/2000

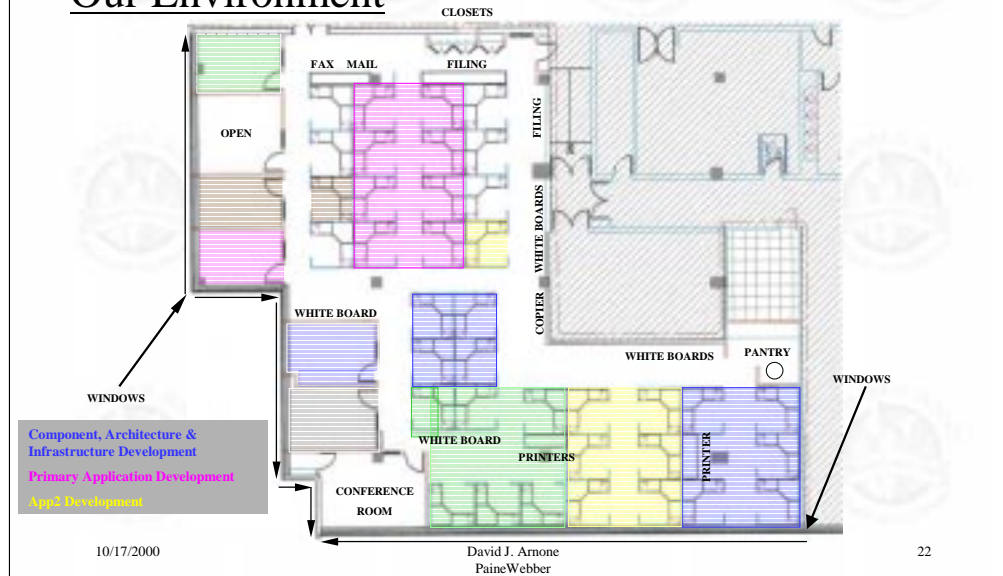
David J. Arnone
PaineWebber

21

- **40% of development staff are recent undergraduates.**
- **Attrition involved commuting issues – time and difficulty.**
- **The people who have departed represent 80% of the framework pioneers.**



Our Environment

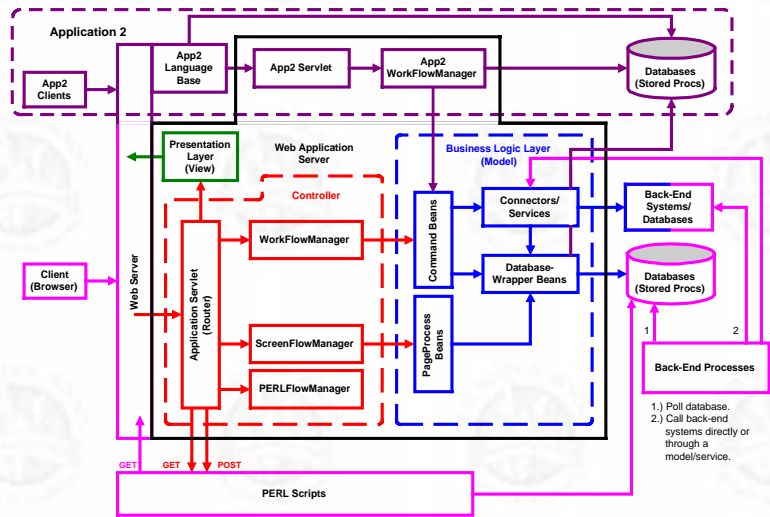


- The blue represents my component and infrastructure team. We own the architecture and the Connectors/Services.
- The pink is the development team for the first application.
- The yellow is where the second application development team is located.
- Green and yellow represent other teams in the organization outside of the scope of this talk and gray is where our senior manager resides.

A high resolution (300%+) will show the floor layout underneath the colors.



Future Work



10/17/2000

David J. Arnone
PaineWebber

23

- We can leverage the flexibility of the Framework by.....
- Implementing a newly configured Servlet and WorkFlowManager for a second application to access and reuse beans in the first application.
- The second application is built on a different language base than PERL.
- This work also is place-holder for platform migration of the second application.



Observations

- Having a system in production helps more than hinders. Always have a working system.
- In everything you do start small – then iterate, iterate and keep iterating. Stay small.
- Following a paired programming approach has preserved domain and solution space knowledge among developers.
- Environment definitely has an impact.



Observations (cont.)

- Applying XP is considered a best practice for WAS development.
- Telecommuting has a place in the process.
- If not for Java, we wouldn't
 - work as fast,
 - code as well.



OOPSLA 2000 Practitioner Report – Internet Applications
Tuesday October 17, 2000 3:30 – 5:00 PM

Porting a Core CGI Workflow System to a WAS Platform While Being Used and Extended to Support the Enterprise

Porting a Core CGI Workflow System Onto A Web Application Server Platform While Being Used and Extended To Support the Enterprise

David J. Arnone
PaineWebber
darnone@painewebber.com

10/17/2000

David J. Arnone
PaineWebber

26