Architecture Summary

Prepared by Architecture Working Group

Goals and summary: April 4, 2012

Updated recommendation: Nov 7, 2013

Administrative
Computing &
Telecommunications

UC San Diego

Original Goals

- Document Current IT Infrastructure
- High-level Architecture Design
- Review Enterprise Middleware Options
- Devise Interim Plan
- Strategy for transitioning existing applications and web services
- Long-term Enterprise Middleware Plan
- High-level Project Plan / Recommendation

General Requirements

- SOA
- Technology Neutral
- Open Standards
- Sustainable infrastructure
- Integration of new technology over time
- Support current functionality
- Total Cost of Ownership
- Support multiple H/W and OS
- Catalog current architecture
- Plan for existing systems
- Support custom middleware components that adopt open standards
- Support open source and vendor-based components
- Review and amendment over time
- Support key Structural Components (*next slide)

Key Structural Components

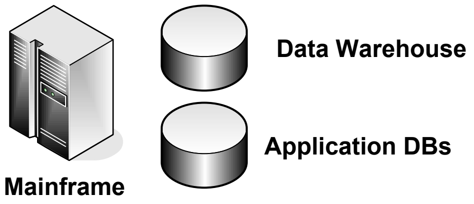
- User Interface Stack
- Enterprise Information Systems (Mainframe)
- Enterprise Workflow
- Identity Management
- Security
- Connectivity (DB, O/R mapping, and persistence)
- Data Services (use of data across diverse systems)
- Service Bus
- Web Service Registry
- System Monitoring
- System Logging
- Build & Test Framework
- Development Tools
- Business Intelligence

Architecture Inventory

- Overall System Diagram
  - Posted in SFEE
- Key Component Documentation
  - Posted in SFEE
- jLink Component Documentation
  - Posted in SFEE

Existing

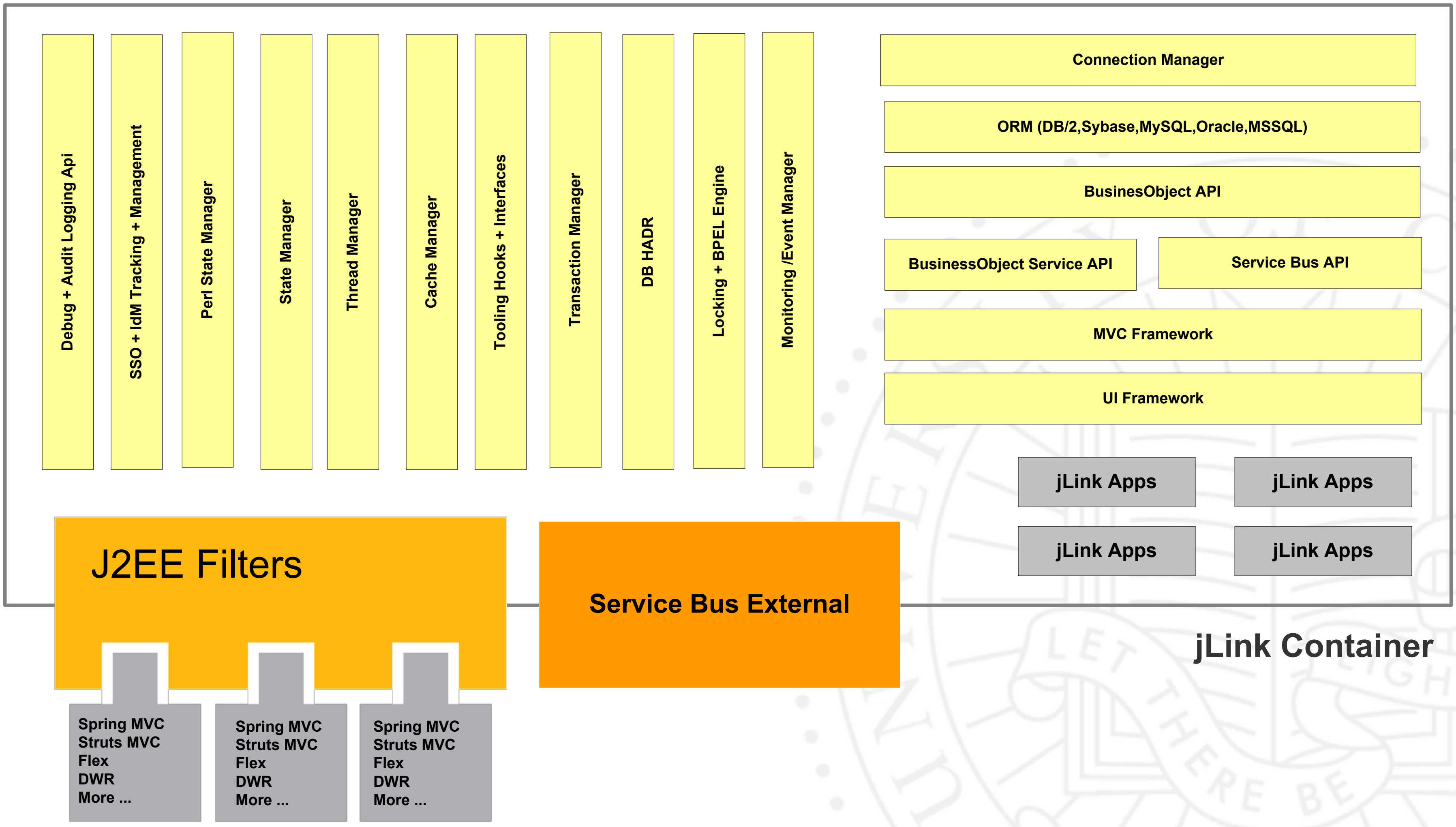
Middleware Architecture Diagram



Mainframe

Data Warehouse

Application DBs



Debug + Audit Logging Api

SSO + IdM Tracking + Management

Perl State Manager

State Manager

Thread Manager

Cache Manager

Tooling Hooks + Interfaces

Transaction Manager

DB HADR

Locking + BPEL Engine

Monitoring /Event Manager

Connection Manager

ORM (DB/2,Sybase,MySQL,Oracle,MSSQL)

BusinesObject API

BusinessObject Service API

Service Bus API

MVC Framework

UI Framework

jLink Apps

jLink Apps

jLink Apps

jLink Apps

J2EE Filters

Spring MVC
Struts MVC
Flex
DWR
More ...

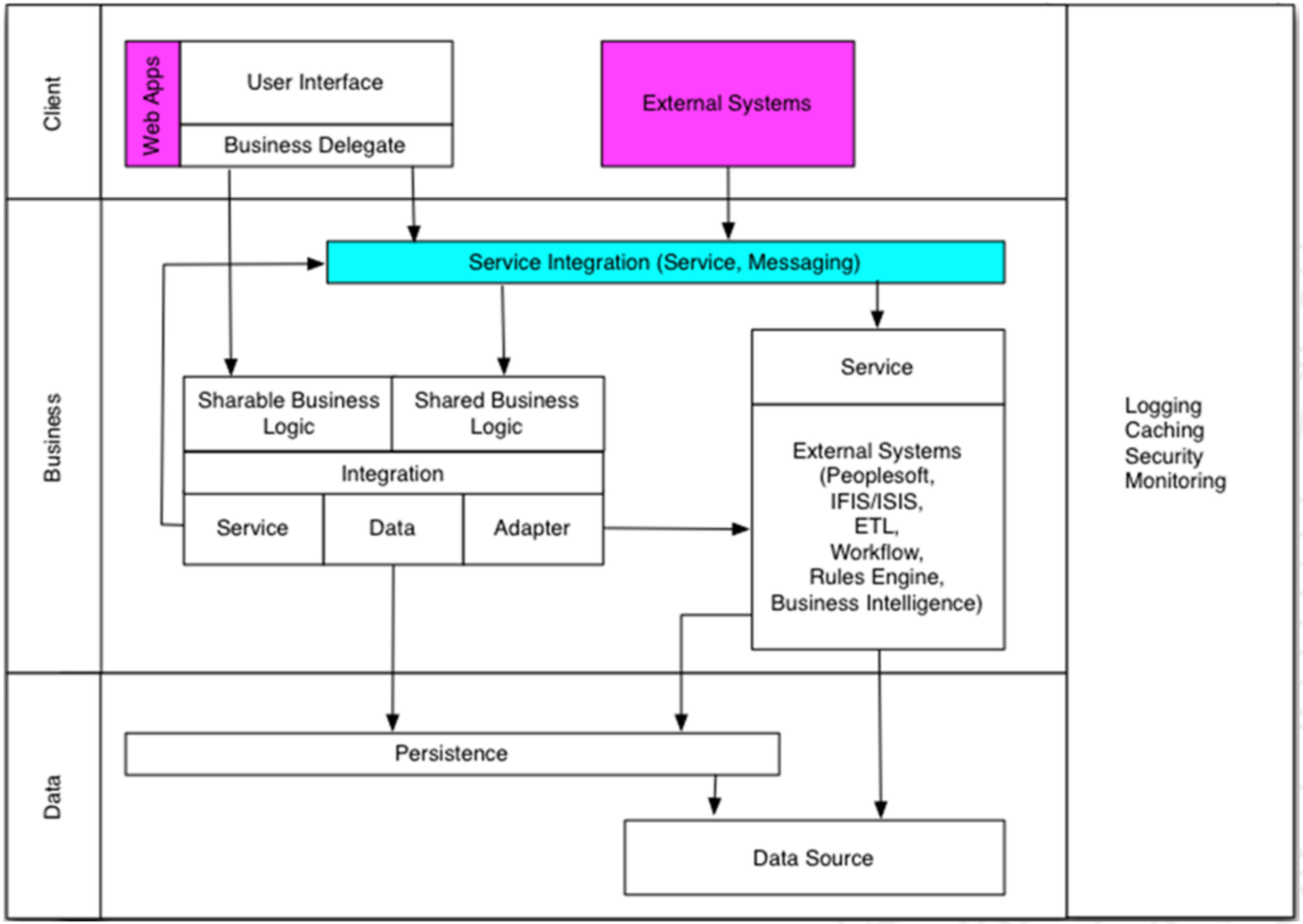
Spring MVC
Struts MVC
Flex
DWR
More ...

Spring MVC
Struts MVC
Flex
DWR
More ...

Service Bus External

jLink Container

Transition to Standards-Based Design



Enterprise Middleware Options

These met our requirements & design:

1. Red Hat JBoss Enterprise Middleware
2. Oracle Fusion Middleware
3. IBM WebSphere
4. Kuali Rice
5. jLink
6. Microsoft Application Platform
7. Hybrid (Tomcat + Java EE 6 Web Profile + Configuration and Monitoring tools + Workflow Engine, SOA registry, etc.)

Enterprise Middleware Options

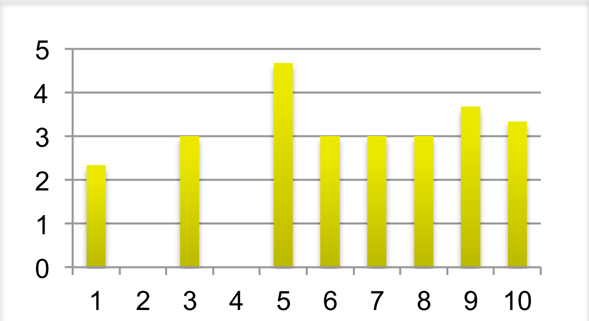
- Considered 3 additional factors:
  - Total Cost
  - Avoid Home-Grown
  - One ACT Solution
- Resulting list of products:
  1. Oracle Fusion Middleware (possible UC License)
  2. Red Hat JBoss Enterprise Middleware
  3. Hybrid (Tomcat + Java EE 6 Web Profile + Configuration and Monitoring tools + Workflow Engine + SOA registry, etc.)

High-Level Review

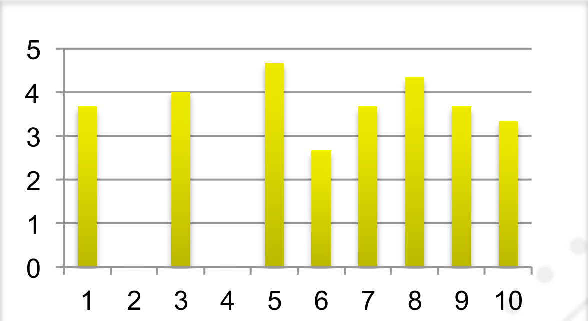
- Review Criteria (input via online form)
  1. Ease of Implementation / Adoption
  2. Upfront Cost
  3. Adaptable / Maintainable / Extensible
  4. Total Cost of Ownership
  5. Standards-based Compliance
  6. Security
  7. Performance / Scalable
  8. Reliability
  9. Integration
  10. Resource Needs

High-Level Review

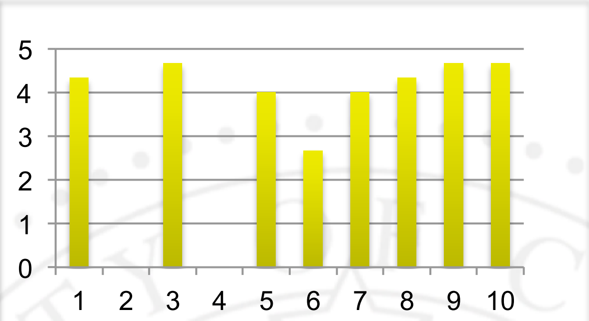
- No Clear Winner



Oracle Fusion



RedHat JBOSS

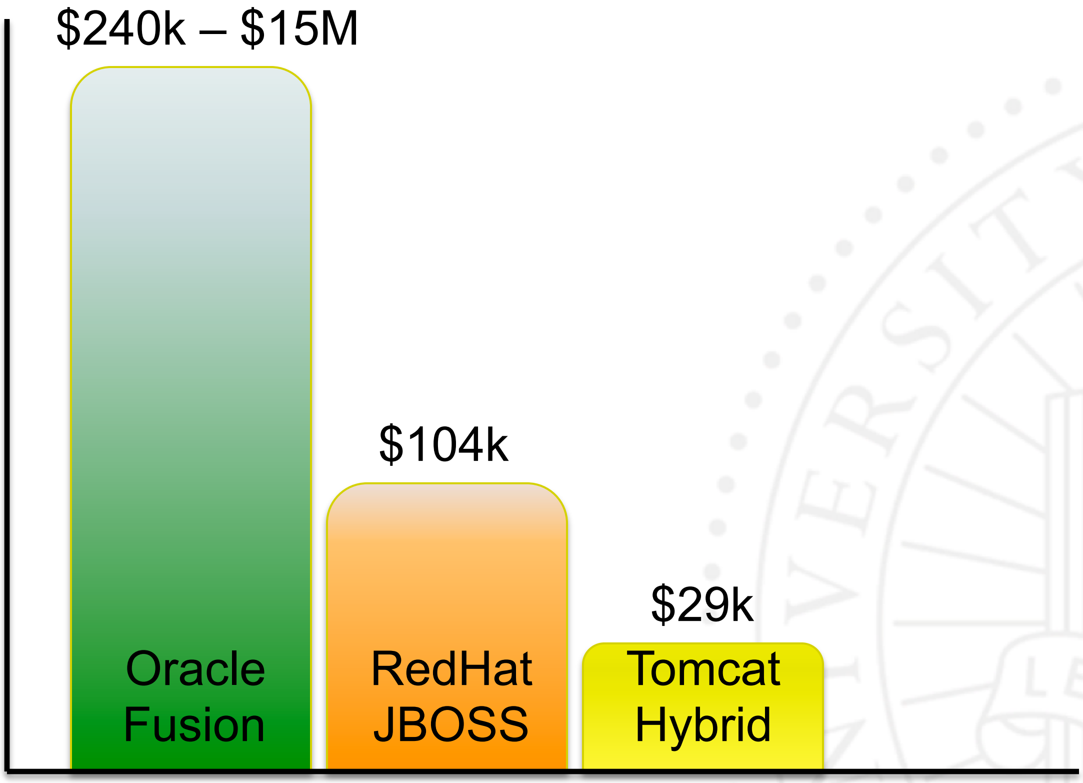


Tomcat Hybrid

- Total Costs Were left out (items 2 and 4)
- All have similar Standards-Based Compliance (item 5)
- Hybrid option has overall higher ratings due to its flexibility, however components such as ESB, Workflow, Configuration + Monitoring Tools would need to be managed individually

Cost Estimates

Annual Product License and Support Fee



Interim Plan – up to 1 year

1. Establish Applications Tech Lead Group
   - Serve as new technology vetting committee
   - Discuss best practices, documentation, how to use, how it fits into ACT's blueprint
   - Details and charter posted in SFEE
2. Adopt the J2EE stack
   - Alleviate limitations of current technologies
   - Identify approved vs. deprecated components
   - Details, chart of libraries, deprecation strategy posted in SFEE

Transition Plan for Existing Apps

1. Port all legacy apps to new product
   - **Pro**: Eliminates all legacy code
   - **Pro**: Brings all technology up-to-date
   - **Pro**: One middleware to support
   - **Con**: A multitude of effort/risk to port everything

2. Gradually deprecate pieces of the legacy middleware
   - **Pro**: Low risk
   - **Pro**: Low upfront cost/ effort
   - **Pro**: Less impact on clients
   - **Con**: Continued support for legacy middleware/systems

Deprecate Pieces of Middleware

- Identify alternative approach
- Clearly document alternative approach
- Train developers
- Set date for using alternative approach for **new** development
- Certain technologies will continue to use jLink until an alternative approach is identified, i.e.:
  - Web Services layer
  - Security layer
- Re-evaluate progress, and set date for transition of additional apps

References

System Diagram Link to diagram
Key Components Link to folder
jLink Components Link to folder
ACT Application Architecture Blueprint Link to bluprint diagram
High-Level Product Review Link to document
Interim Plan – Tech Leads Charter Link to charter
Interim Plan – Deprecation Plan Link to plan
Scope/Requirements Document Link to document

Recommendation – Step 1

- Key project goals:
  - Implement Interim Plan
    - Establish Tech Lead Group
    - Document Best Practices, Guidelines, Blueprint
    - Adopt Relevant J2EE Components
    - Implement Deprecation Process
  - Compile all System Documentation
    - Easy-to-access location
    - Easy to keep updated
    - Available to all of ACT

Recommendation – Step 1

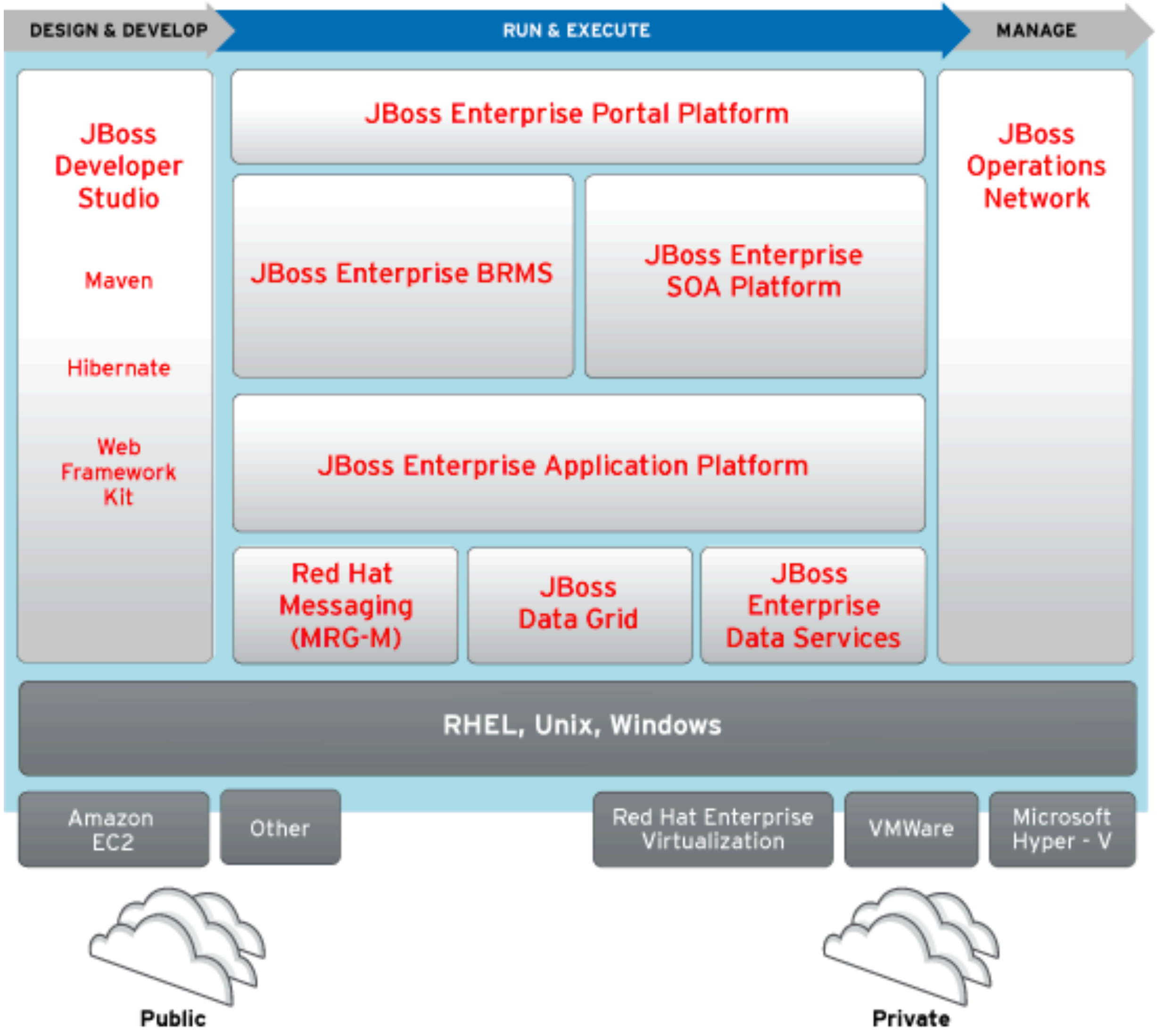
- Key project goals:
  - Implement Interim Plan
    - Establish Tech Lead Group
    - Document Best Practices, Guidelines, Blueprint
    - Adopt Relevant J2EE Components
    - Implement Deprecation Process
  - Compile all System Documentation
    - Easy-to-access location
    - Easy to keep updated
    - Available to all of ACT

Complete

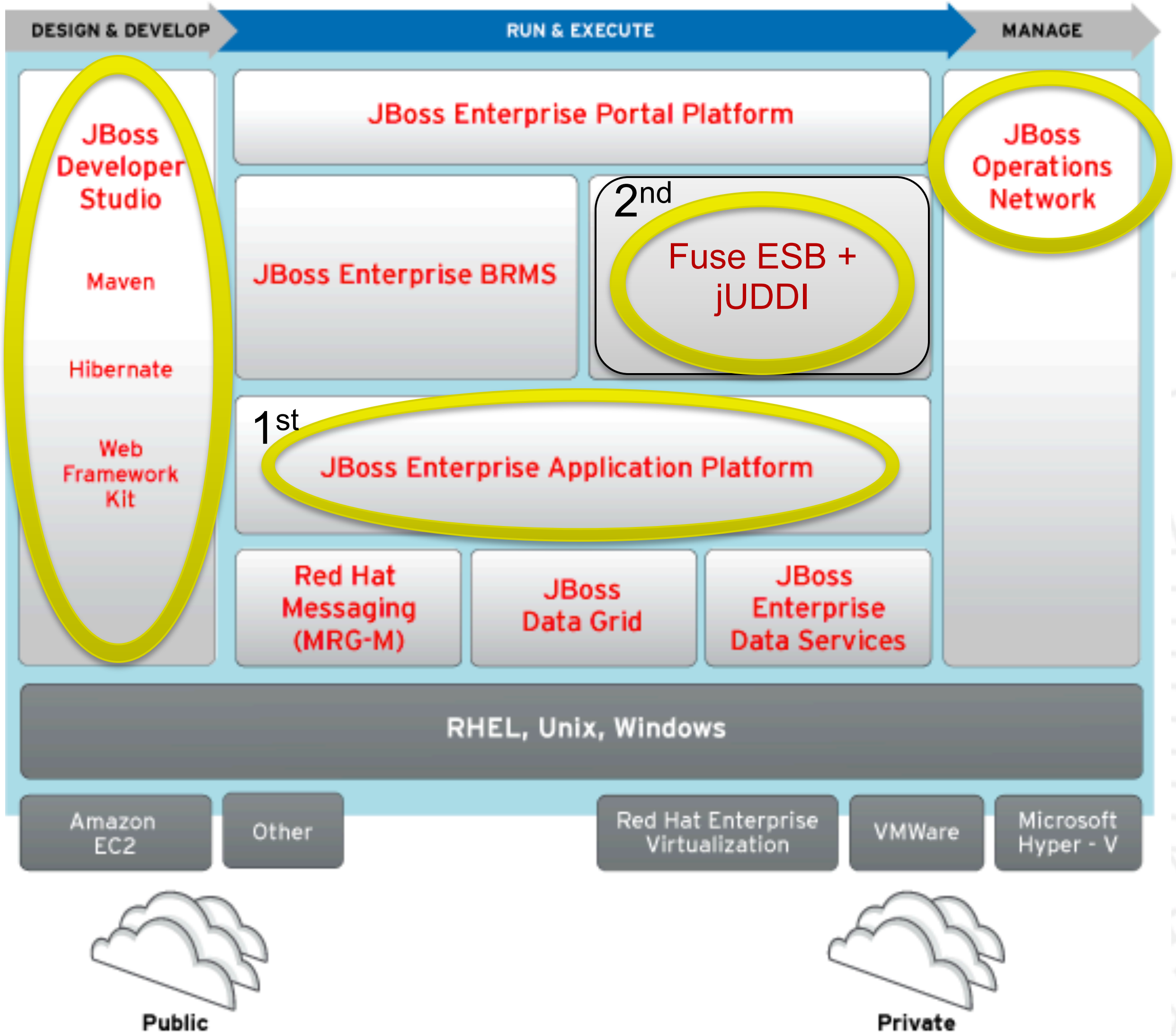
Recommendation – Step 2

- Enterprise Architect (work group) to:
  - Finalize Comparison of Products
    - Pricing, Installation, Configuration, Integration
  - Calculate TCO
  - Select Product
  - Install New Product
  - Transition to New Product
  - Hire Consultants to help with above aspects

Jboss Middleware Stack



Start With Key Components



JBoss EAP POC Findings

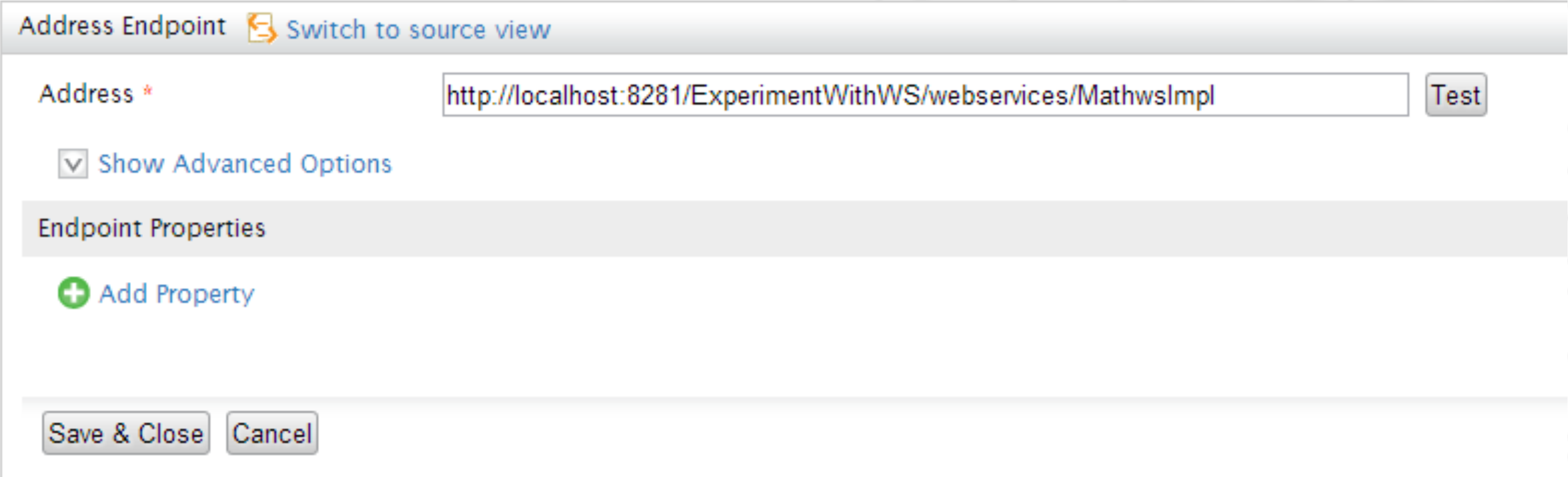
- Stable and robust
- Clustering and software load balancing
- Enterprise-grade product based on open standards
- Reasonable license cost
  - Note: Already budgeted based on last years assessments
- Easy to adopt
- Large community

Fuse ESB POC Findings

- Red Hat still consolidating 3 ESB solutions
  - Old: JBoss SOA Platform
  - Current: Fuse ESB
  - Future: Fuse Service Works
- License cost is changing
  - Fuse: Fixed price per instance
  - RedHat Fuse: Price by core count
  - RedHat Fuse Service Works: Increased price

Fuse POC Findings (cont.)

- Complicated to use
  - Fuse: Hand code
    - cxf:http://localhost:8281/ExperimentWithWS/webservices/MathwslmI?wsdlURL="http://localhost:8281/ExperimentWithWS/webservices/MathwslmI?wsdl"&serviceName=&portName=&dataFormat=MESSAGE
  - WSO2: Visual creation and transformation tool



Address Endpoint

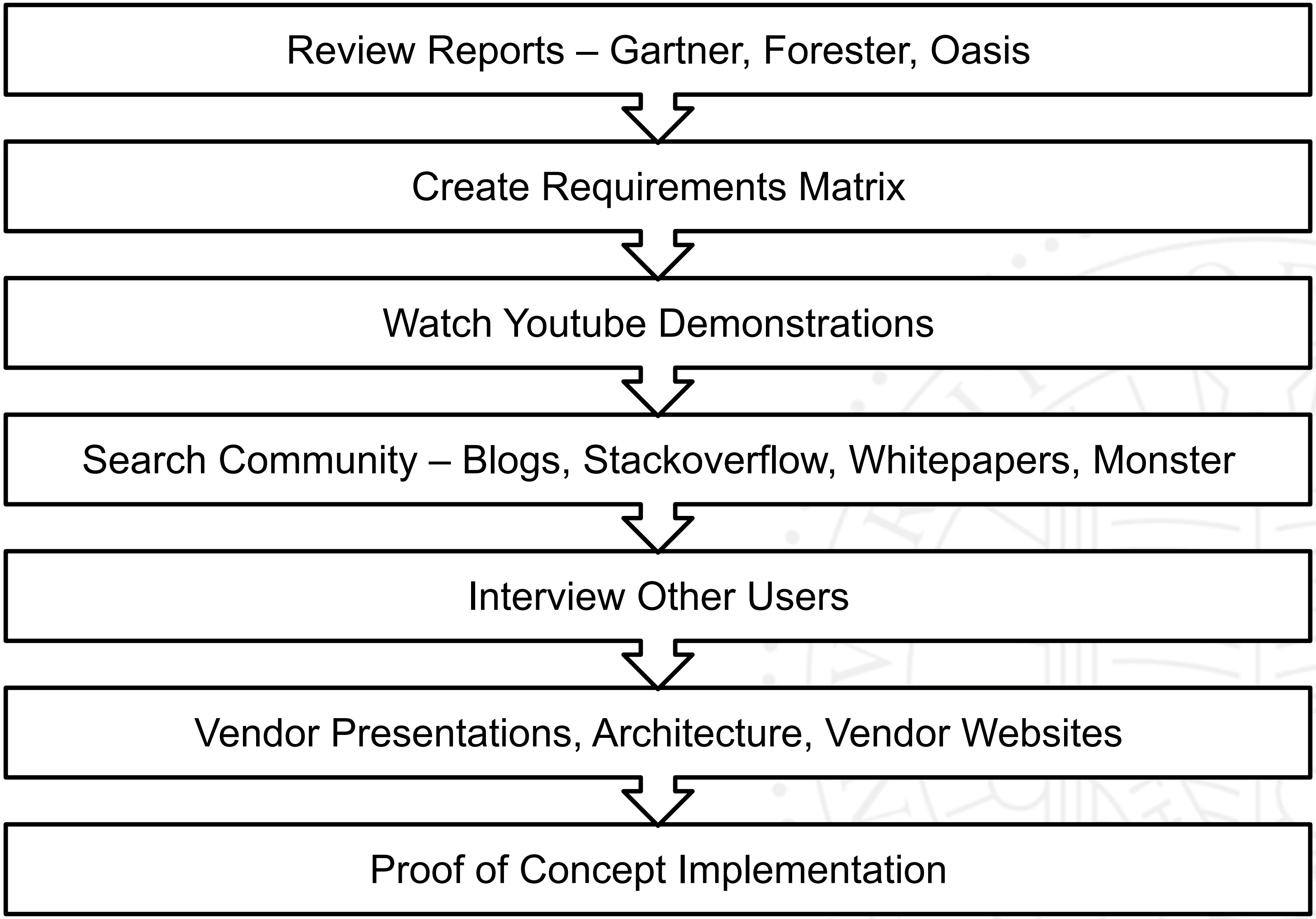
Address * http://localhost:8281/ExperimentWithWS/webservices/MathwsImpl

Endpoint Properties

Restart of ESB Assessment

- Based on Fuse POC findings, the ESB assessment process was restarted.
- Products reviewed
  - Fuse + JBoss
  - Mule
  - Layer7
  - WSO2
  - Talend

ESB Evaluation Approach



Review Reports – Gartner, Forester, Oasis

Create Requirements Matrix

Watch Youtube Demonstrations

Search Community – Blogs, Stackoverflow, Whitepapers, Monster

Interview Other Users

Vendor Presentations, Architecture, Vendor Websites

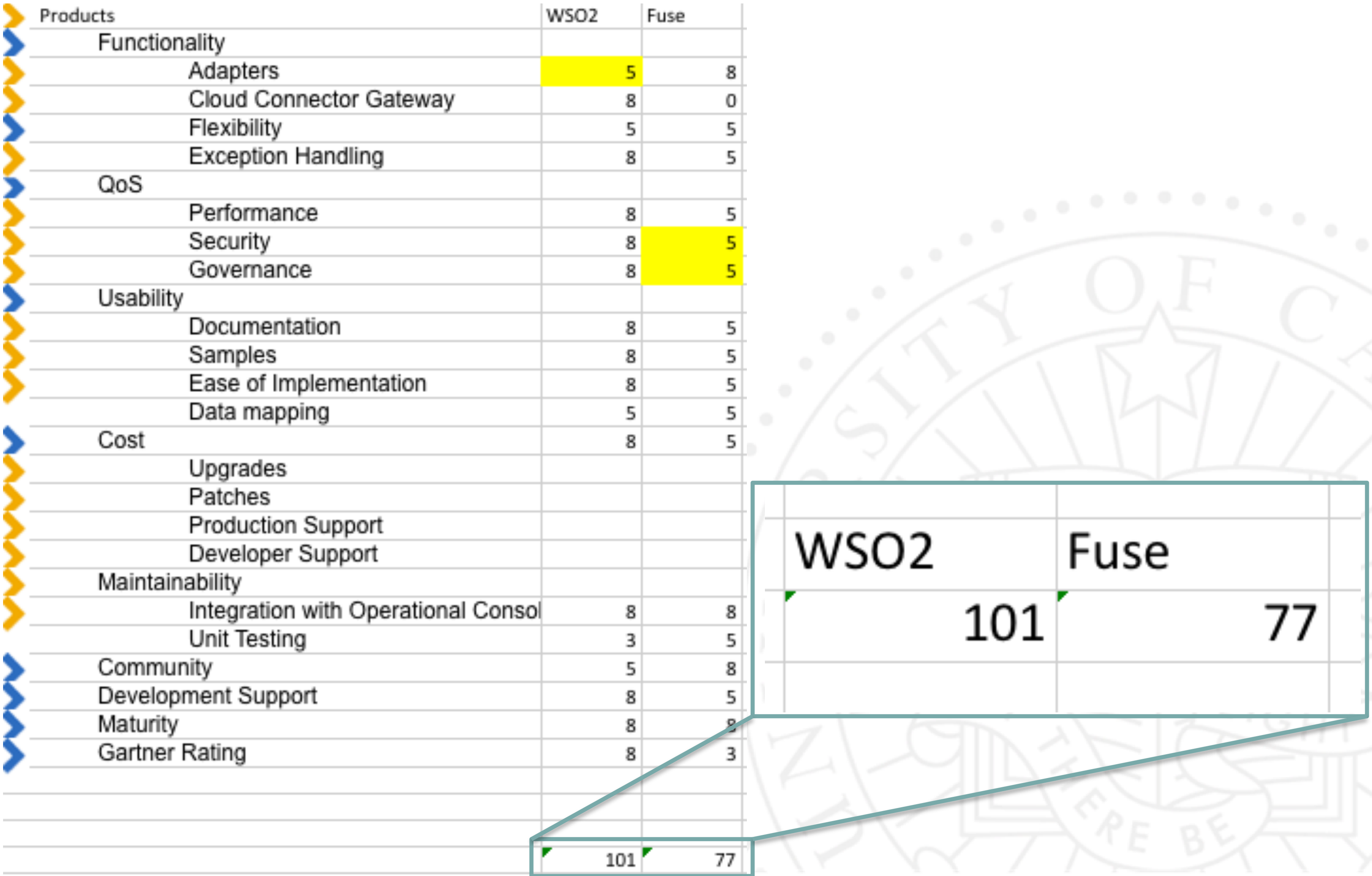
Proof of Concept Implementation

ESB Selection Criteria

- Functionality
  - Technical Specs
  - Flexibility
  - Modular Architecture
- Quality of Service
  - Performance
  - Security
  - Governance
- Usability
  - Documentation
  - Samples
  - Ease of Use

- Maturity
- Company Size and Vision
- Maintainability
- Support
- Cost
- Community
- Industry Ratings

Comparison Results



Products	WSO2	Fuse
Functionality		
Adapters	5	8
Cloud Connector Gateway	8	0
Flexibility	5	5
Exception Handling	8	5
QoS		
Performance	8	5
Security	8	5
Governance	8	5
Usability		
Documentation	8	5
Samples	8	5
Ease of Implementation	8	5
Data mapping	5	5
Cost	8	5
Upgrades		
Patches		
Production Support		
Developer Support		
Maintainability		
Integration with Operational Consol	8	8
Unit Testing	3	5
Community	5	8
Development Support	8	5
Maturity	8	8
Gartner Rating	8	3
	101	77

WSO2	Fuse
101	77

WSO2 POC Findings

- Easy to set up
- Easy to use
  - One-click quality of service attributes, i.e. security
- Flexible license model
  - Fee per instance – can cancel subscription any time.
- Fastest rated ESB
- Smallest overall footprint

ESB Risks

- Fuse ESB
  - RedHat ESB product is evolving and will require migration effort
  - Licensing model is evolving and will require larger investment
  - Steep learning curve will hinder adoption and use
- WSO2
  - Smaller community
  - Potential for buy-out by larger organization

WSO2 Community

- Using and considering WSO2
  - University of Wisconsin Madison
  - UCLA Med Center
  - Boeing, Trimble, Expedia, Stubhub, eBay
  - University of Michigan and NYU (investigating)
- WSO2 Con annual convention held in S.F.
  - November 2013
  - 150 participants (50% using, 50% investigating)
- Online community support
  - www.stackoverflow.com

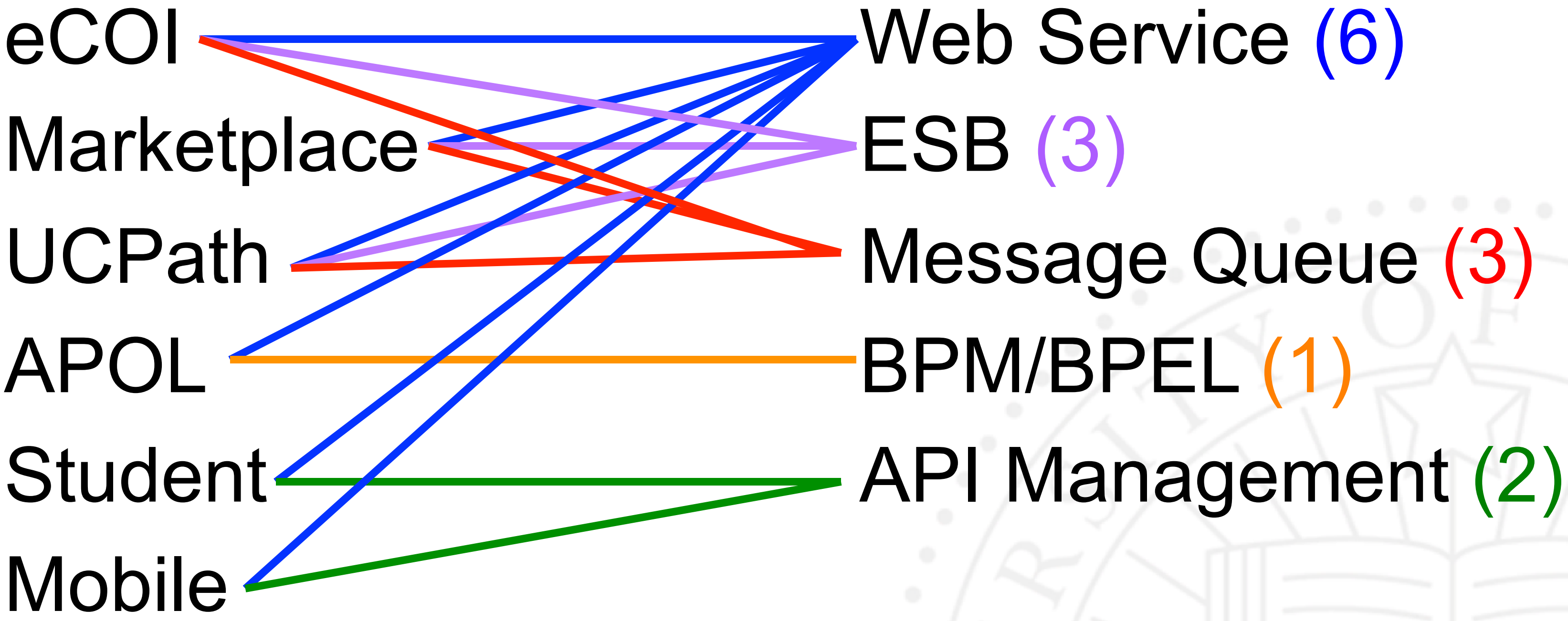
Recommendation: App Server

- JBoss Enterprise Application Platform
  - Platform for Web Applications
  - Replaces JLink middleware functionality with standard J2EE and industry-standards.
  - Annual support subscription
  - Initial consultant setup & training

Recommendation: ESB

- WSO2 ESB
  - Platform for creating and managing reusable web services
  - Platform for system to system communication
  - Replaces JLink middleware functionality
  - Visual development tool
  - Independent from Web App Server environment
  - Annual support subscription
  - Quickstart implementation & training

Immediate Use Cases



eCOI

Marketplace

UCPath

APOL

Student

Mobile

Web Service (6)

ESB (3)

Message Queue (3)

BPM/BPEL (1)

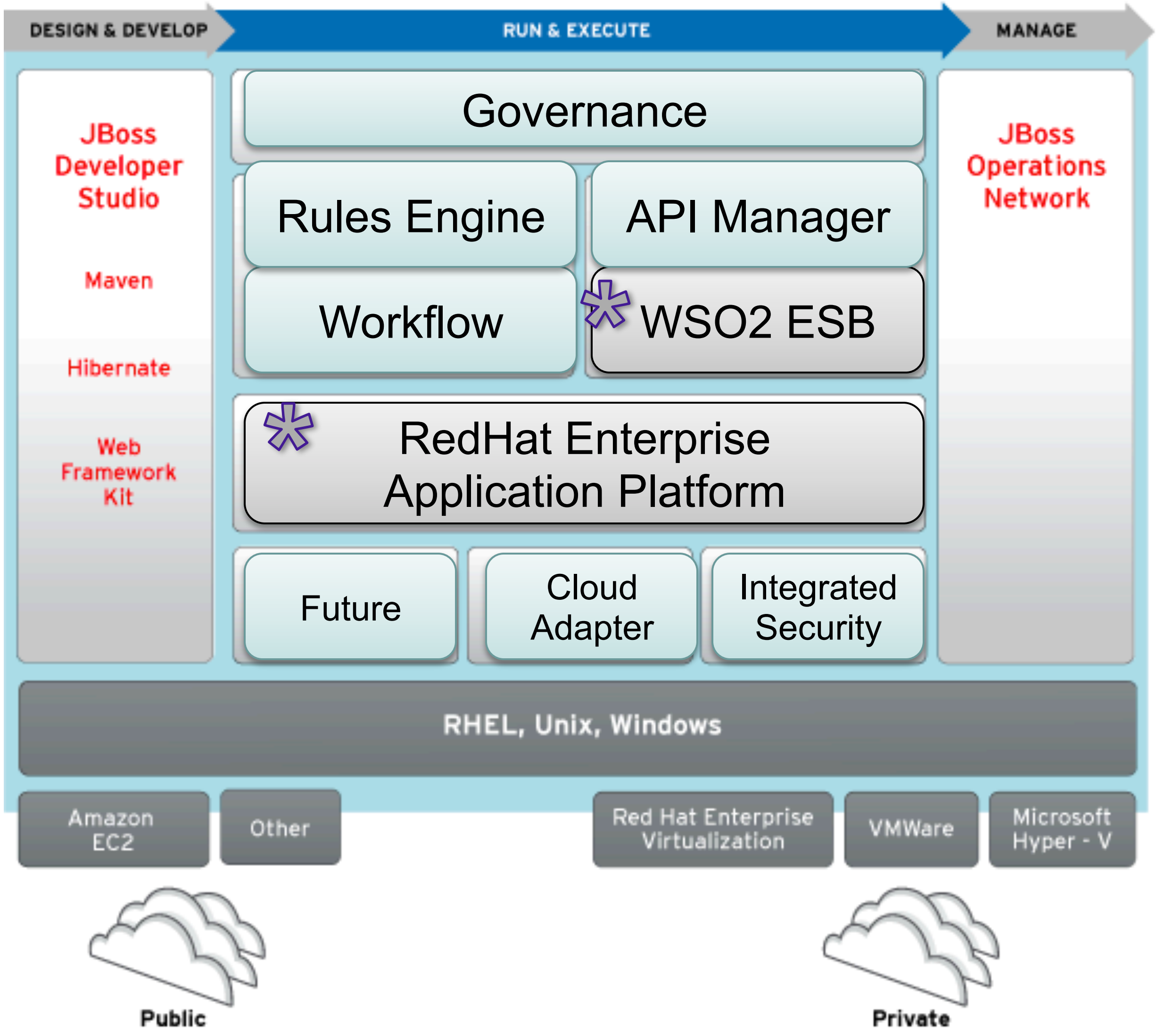
API Management (2)

Prioritized ESB/Middleware Needs

1. App Server (J2EE stack)
2. Enterprise Service Bus (ESB)
3. Message Queue
4. Web Services (SOAP and RESTful)
5. API Management
6. BPEL/BPM
7. Cloud Connectivity

**Overarching needs: Security & Governance

Proposed Middleware Services



ESB References

ESB Heatmap Comparison Likn to heatmap
ESB Evaluation Summary Link to evaluation summary
SOA Overview and WSO2 solution Link to SOA overview, needs and WSO2 solution

JBoss EAP Architecture Design

WSO2 ESB Architecture Design