kuali foundation

Kuali Student

Advanced CAMP
June 19-20, 2008

Jens Haeusser

Director, Strategy

Information Technology

Objectives

- At the end of this session you will have seen
  - The Kuali Student technical architecture vision
  - The results of our efforts to date
  - The working relationships in place to support our efforts

Agenda

- Kuali Student's Vision
  - Our vision for Phase I and Phase II (July 2007 – May 2008)
  - Our execution of the vision
- Kuali Student Architecture
  - Web Services Stack
  - Development Infrastructure
- Our Current Challenges
- Relationships with vendors, open source, and Kuali
- On the Leading Edge While Remaining Flexible
- Getting Validation of Our Work

Background

- Modular, standards-based next generation student system
- Community Source project with a 5 year timeline
  - UBC, Berkeley, Florida, Maryland, San Joaquin Delta, Southern California
  - MIT, Cambridge
- Person centric system
- Service Oriented Architecture
  - Enables integration at diverse institutions
  - Allows schools to implement *their* practices

Functional Vision

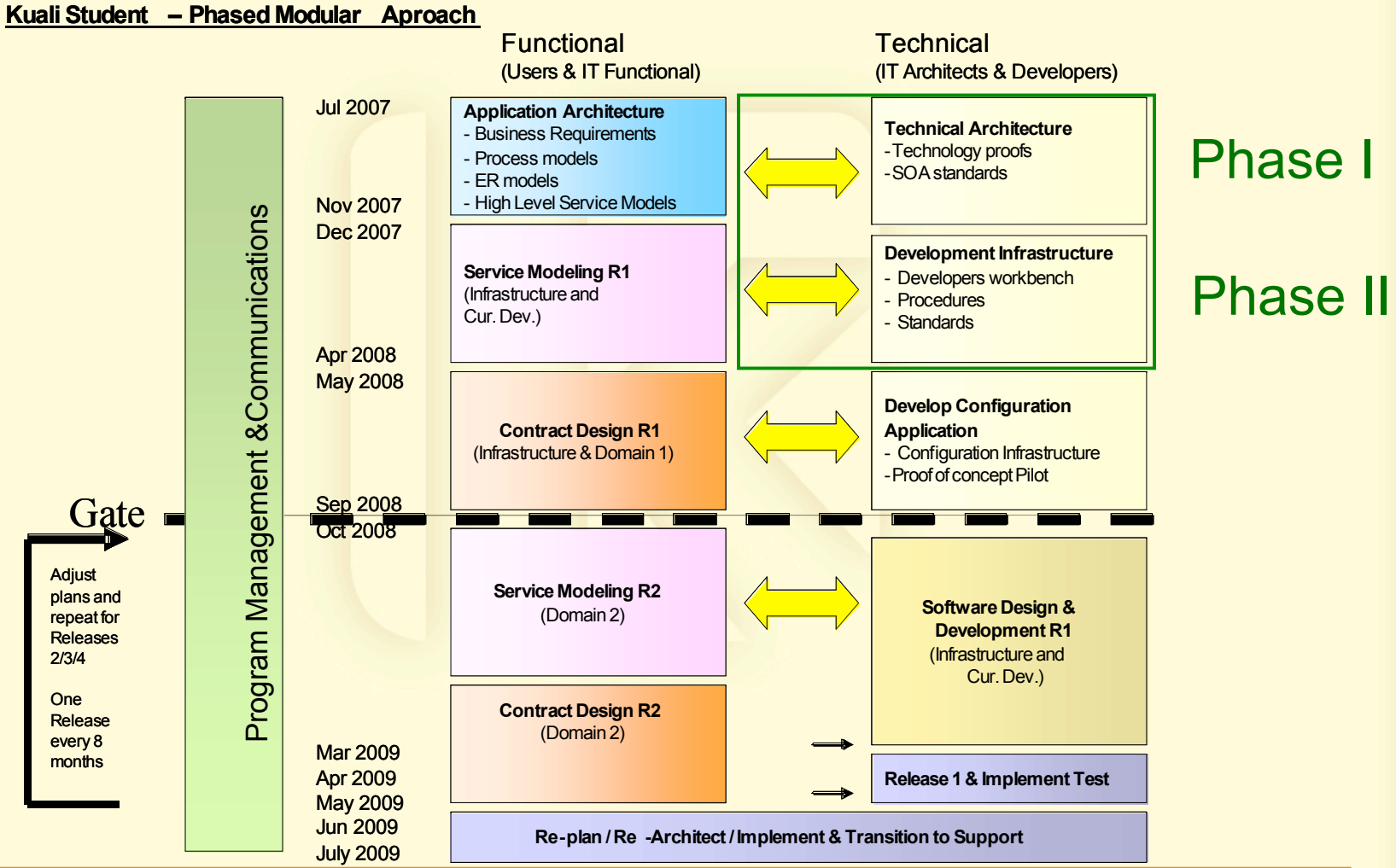
- Support end users by anticipating their needs
- Wide range of learners and learning activities.
- Wide range of business processes
- Easier to change business processes.
- Reduce time staff spend on routine task

Technical Vision

- SOA and Web Services
  - SOA Design Methodology
  - SOA Governance
  - Web Services: SOAP, WSDL, XML Schema
- Web Services Stack
  - Standards-based
  - Adhere to Educational Community License (ECL)
  - Java as the Language and Platform of Choice (but that works with any technology that implements the service contracts
- Open Source Reference Implementation

Guiding Principles for KS Technical Architecture
http://www.kuali.org/assets/pdf/KS-GuidingPrinciplesforTechnicalArchitecture.pdf

Technical Vision

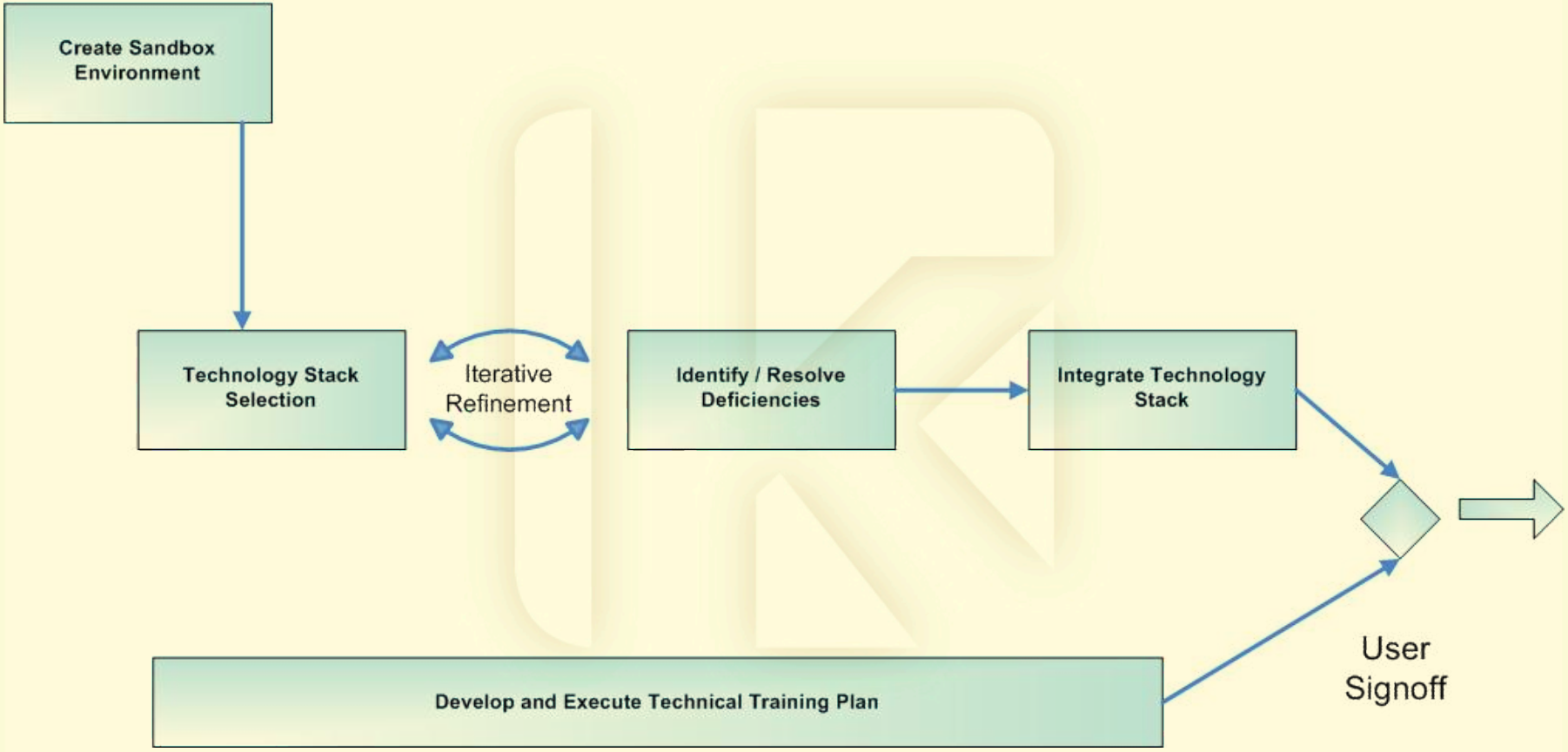


financial system research administration student rice infrastructure

Standards-Based Architecture

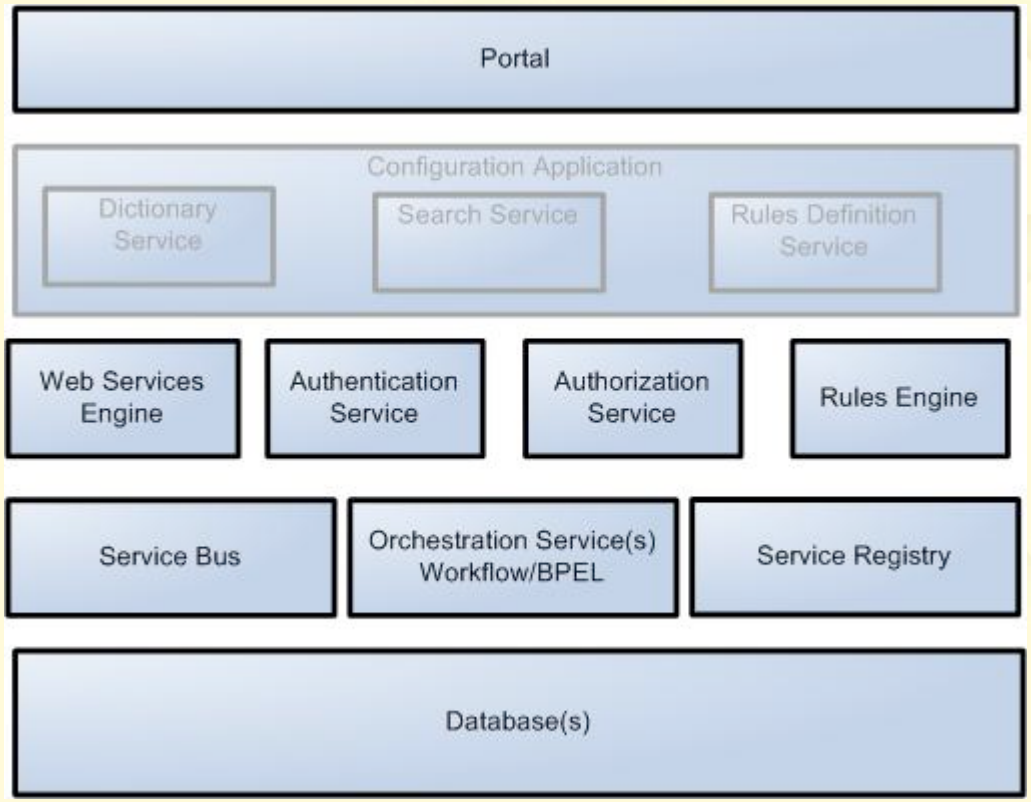
- Emphasis is on open, widely accepted and adopted standards, not on products
  - W3C
    - XML, XML Schema, SOAP, WSDL, etc.
  - OASIS
    - WS-Security, WS-Transaction, SAML, etc.
  - Java JSRs
    - JAX-WS (224), JAXB (222), JBI (208), Rules (94), Portlets (168/286), etc.
- Allows flexibility as open source products and product space evolves

Technical Vision
Technology Stack Selection – Phase I



See “Phase I Proof of Concept at
http://kuali.org/communities/ks/resources.shtml

Technical Architecture
Web Services Stack Recommendations



- uPortal

- KRice?
- KS BRMS

- Apache CXF/Sun Metro
- CAS, Acegi
- Kuali Identity Management (KIM)
- JBoss Drools

- Apache ServiceMix
- Sun OpenESB
- Kuali Enterprise Workflow (KEW)

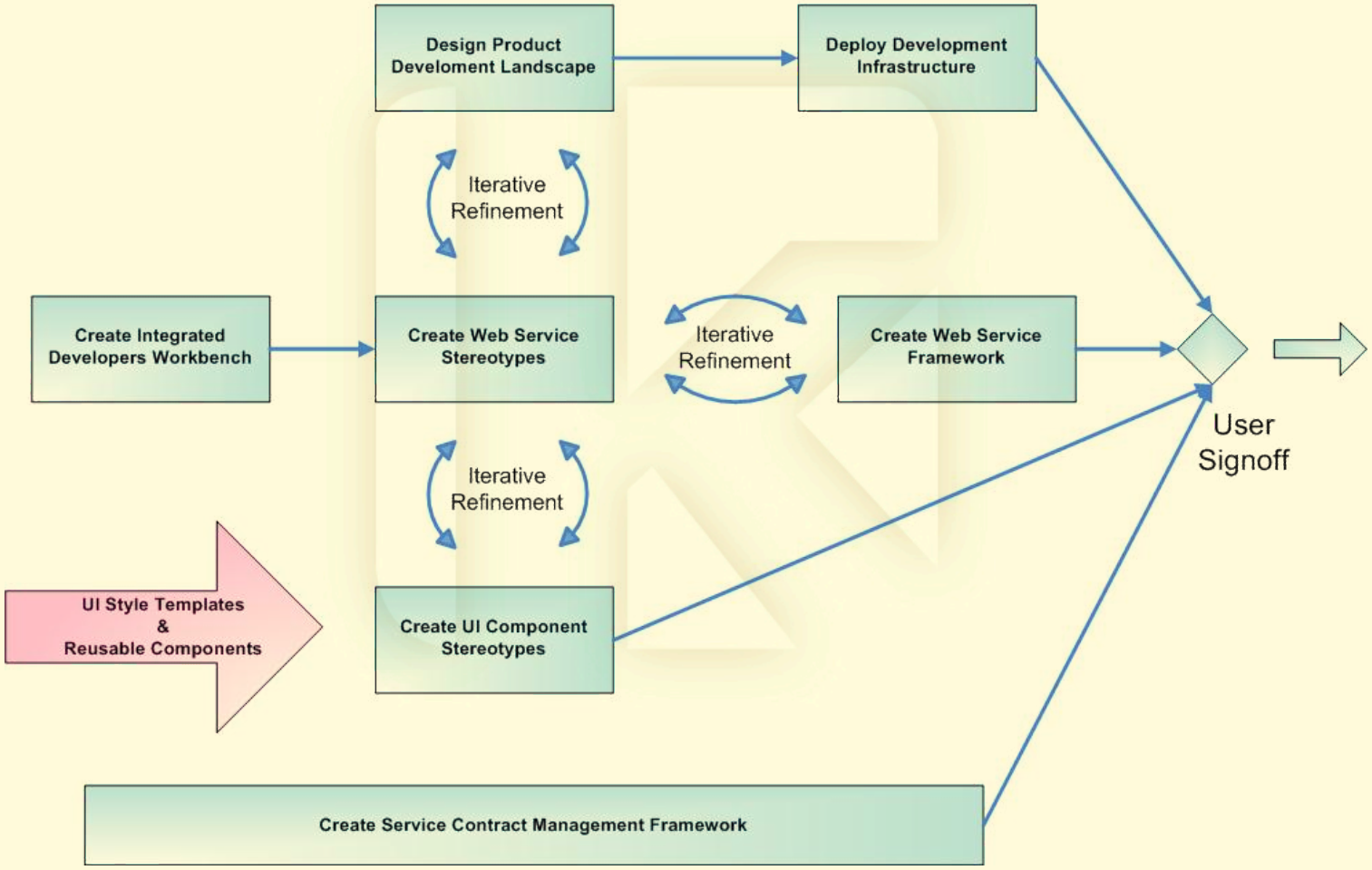
- Apache Derby
*Certified on MySQL & Oracle

Phase I Recommendations found at:
http://www.kuali.org/assets/pdf/KS+Phase+I+Recommendations+v2.0.pdf

Challenges

- WS-Transactions
  - No open source product implements WS-Transaction in a fully open source Web services environment
  - Working with Sun on a possible solution that works outside Glassfish (on Tomcat)
- BPEL
  - Selection made (Sun OpenESB), but there are issues with command line deployment options, lack of parallel forEach, and lack of support for compensating transactions that kept BPEL from being considered currently viable.
  - Working with Sun on solutions
- Workflow
  - No WS-* open source implementation of workflow
  - Kuali Student and Kuali Enterprise Workflow (KEW) will look to integrate KEW by
    - Ensuring KEW endpoints are exposed as Web services
    - Consumers and providers implement WS-* (specifically WS-Atomic-Transaction and WS-Security)
    - Interface definitions are optimized for remote access (distributed)

Technical Vision
Development Infrastructure – Phase II



financial system research administration student *rice* infrastructure

Development Infrastructure

- Development Environment & Technologies
  - Maven & Subversion
  - Continuous Integration
  - Deployment Process
  - JPA (Persistence)
  - JUnit Testing
  - Logging/Auditing
  - Change Management
  - Error Propagation (UI/Services/Database)
- User Interface
  - Google Web Toolkit (GWT)
  - Validation framework
  - Portal strategy
  - Internationalization strategy
- Rules
  - Business Rule Management System (BRMS)
    - Searchable database of rules
    - User interface for defining rules
  - Run-time
    - Will produce readable translations for errors and successfully executed business rules
- Identity Management, AuthN, AuthZ
  - Work with Kuali Identity Management (KIM) team

Architecture, Infrastructure, Methodology Proofs

- Integrating the Technology Stack, Development Infrastructure, and SOA Methodology through Proof-of-Concepts
  - PoC 1 / Jan 2, 2008
    - Prove that the selected technologies integrate (uPortal, Metro & CXF, ServiceMix, ODE, Drools, Derby)
    - See “Phase I Proof of Concept” at http://kuali.org/communities/ks/resources.shtml
  - PoC 2a / June 1, 2008
    - Initial end-to-end methodology proof (functional and technical)
    - An implementation of Person and of Leaning Unit and Learning Unit Relation
    - Flow: Sign In → Display List of Courses → Register for a Course
  - PoC 2b / Nov 1, 2008
    - This will be a prototype
    - Fully-realized PoC 2a
    - Apply full end-to-end methodology
    - Apply full set of interfaces and interface implementations
    - Will be used as a model for completing Release 1
    - Also used by implementation teams at each institution for
      - Load testing
      - Hardware sizing
      - Further implementation teams’ understanding of the technical architecture

Relationships

- Sun
  - Performed review of Proof-of-Concept and currently reviewing Phase I Technical Recommendations
  - Implemented Glassfish-based Web Services stack against our Proof-of-Concept services
  - Working with Kuali Student on transactions, BPEL
- IBM
  - Performed review of technical and application architecture (planning phase)
  - Will be reviewing our Phase I Technical Architecture Recommendations
- uPortal
  - Kuali Student is working with uPortal community on uPortal 3 enhancements
- Kuali Foundation
  - KRice – building roadmap for interoperability
  - Kuali Enterprise Workflow (KEW) – will be using KEW as workflow and making needed interoperability enhancements

Validation

- Validation by external parties at each phase
  - Both Functional and Technical validation
  - Planning Phase
    - Sun
    - IBM
    - Independent Consultant
  - Phase I
    - Currently with Sun
    - IBM in review
  - Swapping (Plug-and-Play)
    - Sun stack swap
- Enterprise Service Bus
  - Our separate, hands-on evaluation of ESBs selected Apache ServiceMix, the same conclusion as that of the Mellon Foundation funded research on ESBs.

Mellon ESB Report: http://tid.ithaka.org/enterprise-service-bus-project/esb-narrative-rc-4.pdf

Leading Edge

- We are Open to Change
  - Stack selections are not static
  - Selections are based in great part on the standards they implement
  - If an obvious better choice comes along, or one technology leapfrogs one we're using, we'll replace

- "Swappability"
  - We aim for stack components that are plug-and-play
  - Kuali Student documentation will always enumerate the level of swappability of each component
  - See Section 13 "Swappable Infrastructure" of the Phase I Recommendations document found at:
    http://www.kuali.org/assets/pdf/KS+Phase+I+Recommendations+v2.0.pdf

KS & KRice
Kuali Foundation Technology Choices

	Google Web Toolkit	uPortal 3.0			UI
Identity Management: KIM	Workflow: KEW	Business Rules: BRMS	KNS Dictionary & Search		Middleware
	Code Management: Subversion	Build: Maven	Unit Test: JUnit		Eclipse Workbench
	JPA: Hibernate	JAXB	JAX-WS		Mapping Frameworks
Database: Derby	Service Engine: CXF/Metro	Servlet Container: Tomcat	ESB: ServiceMix	Rules Engine: Drools	Technology Stack

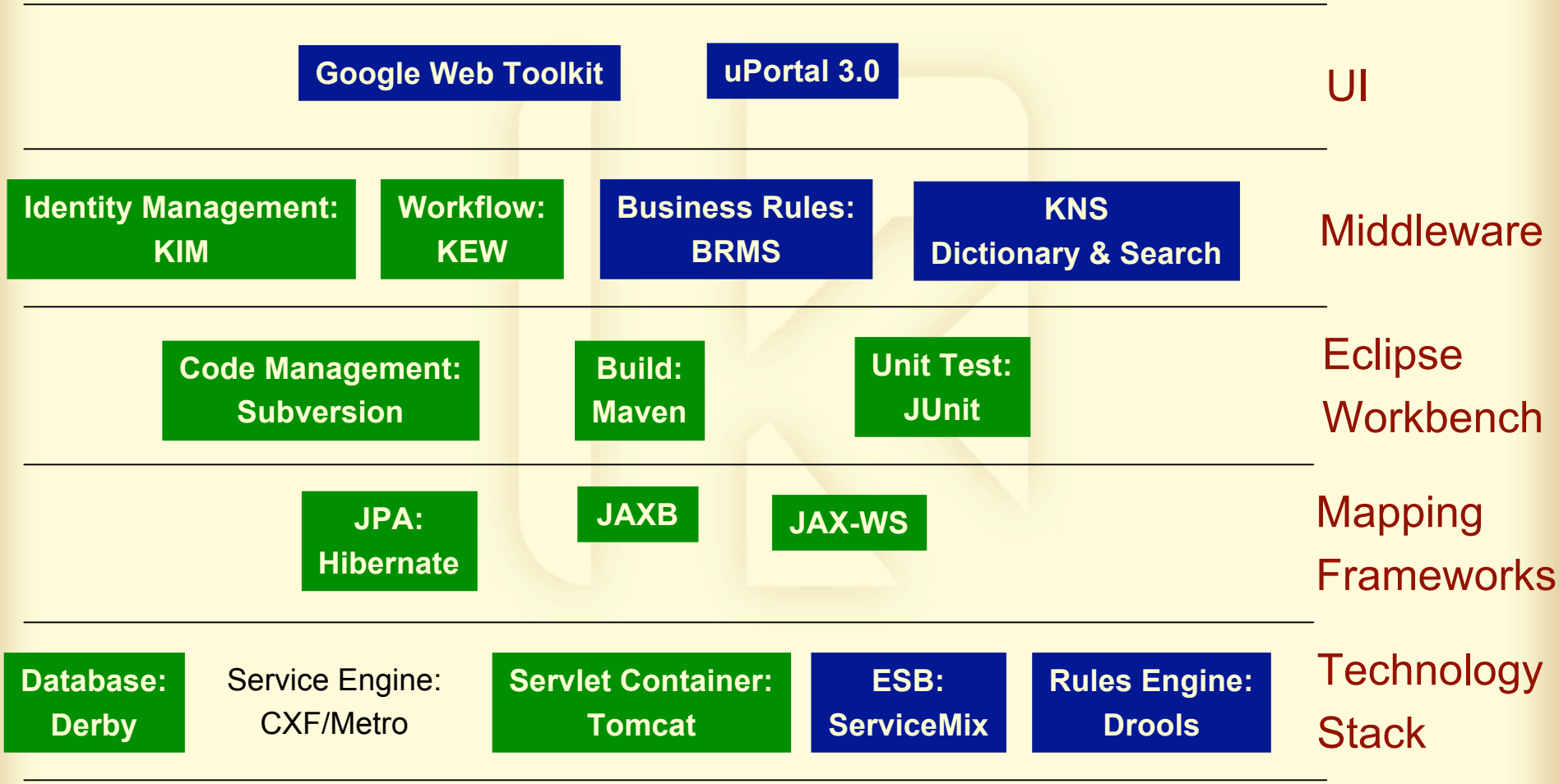
financial system research administration student rice infrastructure

KS & KRice
Current Alignment

Google Web Toolkit, uPortal 3.0	UI
Identity Management: KIM, **Workflow: KEW**, Business Rules: BRMS, KNS Dictionary & Search	Middleware
Code Management: Subversion, **Build: Maven**, **Unit Test: JUnit**	Eclipse Workbench
JPA: Hibernate, **JAXB**, **JAX-WS**	Mapping Frameworks
Database: Derby, Service Engine: CXF/Metro, **Servlet Container: Tomcat**, ESB: ServiceMix, Rules Engine: Drools	Technology Stack

financial system | research administration | student | rice infrastructure

KS & KRice
Planned Alignment



Questions?

Jens Haeusser
jens.haeusser@ubc.ca

Kuali Student

http://www.kuali.org/communities/ks/

http://www.kuali.org/communities/ks/application_architecture_documents.shtml