Kuali student®

… the next generation student system is coming!

Brian Silzer
Audrey Lindsay
University of British Columbia

ARUCC
Jun 26, 2008

Kuali student ®

Kuali Student is...

a modular, open source, standards-based next generation student system

being developed through a community source process over a 5 year period,

that will be delivered through a service-oriented architecture and web services.

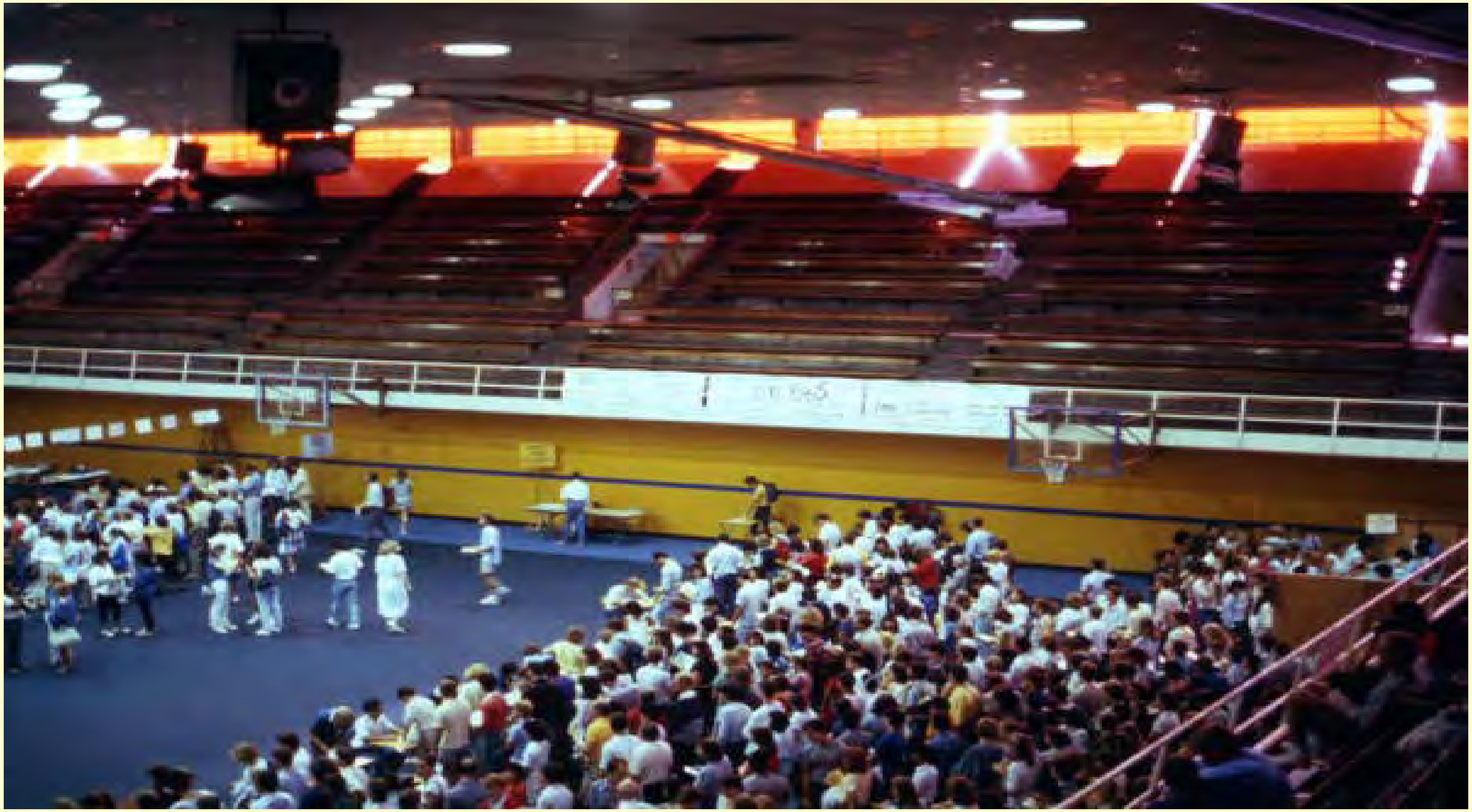
kuali student®

Agenda

- Why now?
- Community source and the Community
- Functional design and scope
- Technical architecture
- Development approach
- Where we are and where we’re going

Why Now?

Registration at UBC 1986



kuali student®

Why Now?

- Many student systems don’t meet current needs
- Vendor solutions may not be the answer
- Development of in-house systems is challenging
- Collaboration and open source systems development works
- We can build systems that do more for users

Why Community Source?

Benefits

- Shared resources means more efficient development
- Institutions share ideas and create innovative solutions, leveraging their user experiences
- Contributing institutions have direct input into functions and features
- Sustainability – a community that contributes to enhancements can ensure sustained development
- Support – commercial partners for implementation and support are encouraged

Kuali Student will

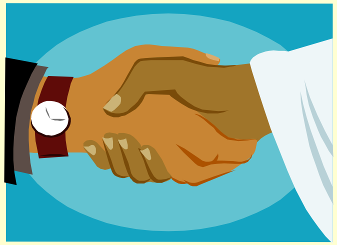
- Build a community of interest
- Encourage commercial affiliates
- Establish procedures and standards for development
- Share implementation experiences

Why did UBC join the Community?

- Provide *specific* input on product direction
- Access project documentation and artifacts as they are developed
- Have early access to software for testing and implementation
- Contribute enhancements to ensure the quality and suitability of the end product
- Help develop support processes and product release strategies.
- Contribute knowledge and experience to the community
- **Implement some or all of Kuali Student sooner**

What are the Contribution Opportunities?

- Founders
  - – substantial commitment in money and people
- Partners
  - – significant commitment to core product
- Contributors
  - – Help sustain, or enhance by developing new modules
- Adopters
  - – commitment to adopt some modules



- Supporters
  - – stay connected through distribution lists

Founder & Partners

Founders

- University of British Columbia
- University of California, Berkeley
- University of Maryland, College Park
- Florida State University
- San Joaquin Delta College
- University of Southern California

Partners

- Massachusetts Institute of Technology
- University of Cambridge

Other Partners

kuali foundation®

The Andrew W. Mellon Foundation

Supported by:

- AACRAO
- NITLE

  Advancing liberal education in the digital age

Vision: Functional Objectives

- Support 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 tasks

Functional Design: Elements

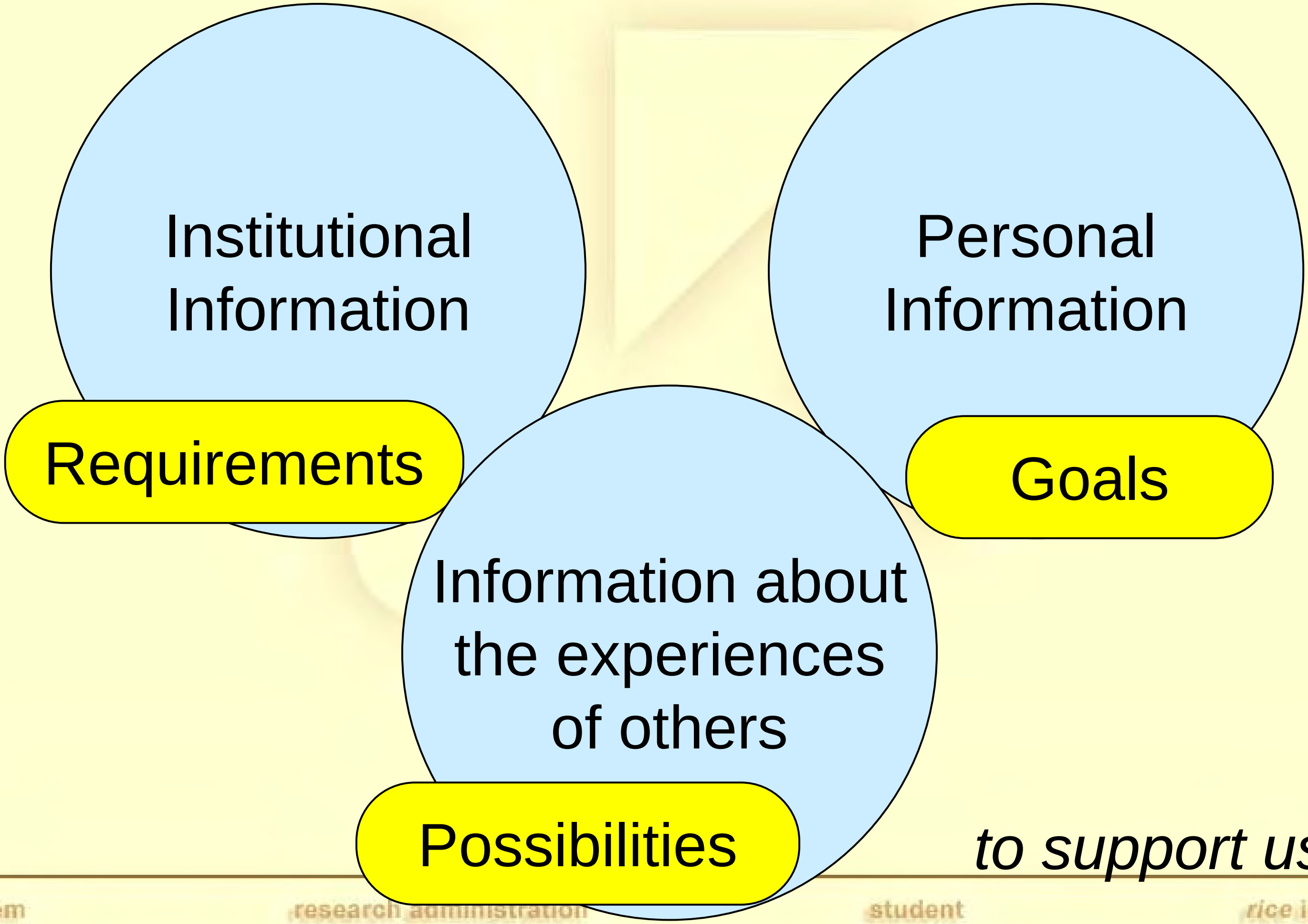
- High level entities
  - learning units, person identity, time
- Concierge
  - use what we know to help people achieve their goals
- Rules engines, work flow
  - rules and logic are not in the code
- Modular, configurable system
  - your processes, not someone else’s “best practices”
- Managed access to information
  - people can see what they should be able to see
- Internationalization
  - language, characters, currencies, date formats

Entity: Learning units

- A learning unit can represent:
  - a course; a single lecture in a course; a 15 minute student presentation in a course
  - participation in community service
  - a year of study
  - a degree program
  - any non-credit, continuing studies, or other activity
- A "learning unit number" is like a SKU...
- We can also have:
  - learning results
  - learning plans
  - learning resources

Concierge

We should use:



Institutional Information

Requirements

Personal Information

Goals

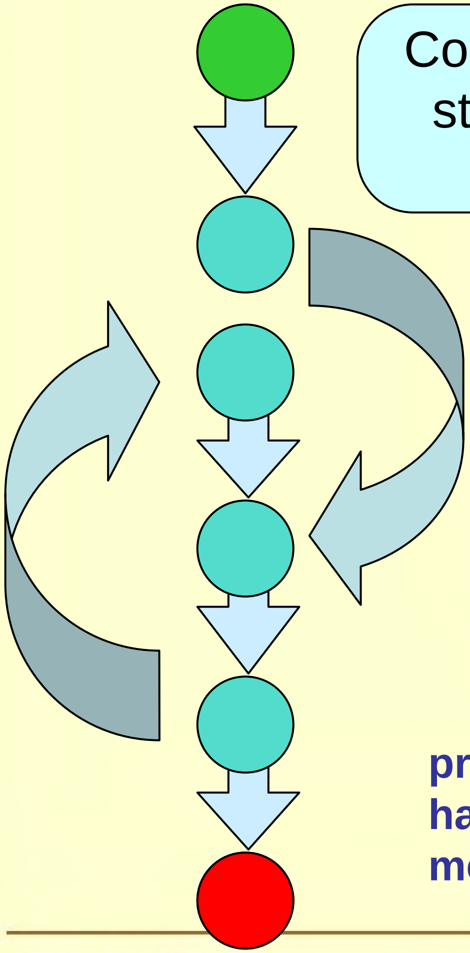
Information about the experiences of others

Possibilities

to support users

Concierge concepts

ability to register triggered by accepting offer of admission



Concierge "sees" student accept offer

Concierge sits looking and listening for changes in a person's state, institution rules, peoples experiences, etc.

Concierge checks student info, program, required courses, elective opportunities, and guides student to solution that works for her

process ends when student has complete program that meets her needs

Uses:
- **Information**
- **Rules Engine**
- **Workflow**

Tier 1 Business Domains Release 1

- Person Identity
  - – manage Person info
  - – support Authorization, Authentication
  - – manage Groups, Organizations
  - – manage Contact info
- Learning Unit Management
  - – manage catalog of Learning Experiences
  - – manage creation, approval new LUs
  - – manage evaluation, review of existing LUs

Tier 1 Business Domains

- Enrollment
  - manage Learner to LU relationships
  - manage Provider to LU relationships
  - manage Learning Results
- Program Audit and Academic Evaluation
  - support evaluation, status towards Learning Objectives
  - supports ongoing evaluation of academic progress
- Student Financials
  - product pricing
  - assessment of additional fees
  - determine invoice and payment plans
  - payment processing

Tier 2 Business Domains

- Admissions
  - capture Application info
  - manage Evidence
  - automate process workflow
  - evaluate Learner’s qualifications
- Scheduling
  - manage LU “offerings”
  - schedule Resources
  - manage Calendars
- Financial Aid
  - manage Awards, Financial Aid Resources
  - maintain student Characteristics and Needs
  - assign Awards to students

Out of Scope Functionality

Tier 3 – Out of scope for Founders

- Recruitment
- Event Management
- Housing
- Athletics
- Alumni
- Family Financial Planning
- Elections
- Student Life

Out of Scope

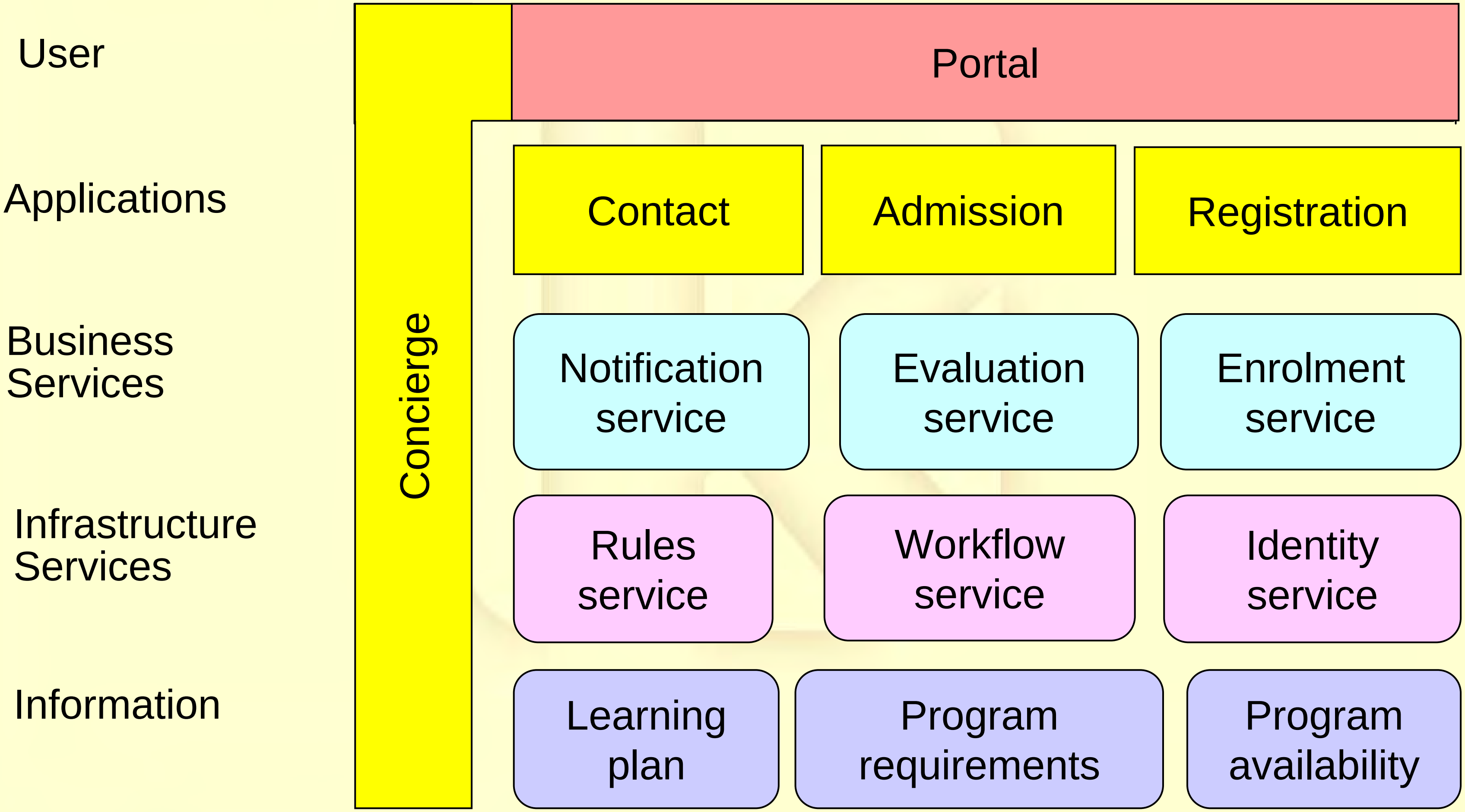
- Learning Management System
- Student Portfolio
- Financial (FMIS) system
- Campus Calendar
- Facilities Management
- Library
- Parking

Vision: Technical Objectives

- Develop an architecture based on Service-orientation, implemented using Web Services
- Use an Open Source technology stack
- Define & publish service contract specifications & development standards
- Produce a software product based on a set of services

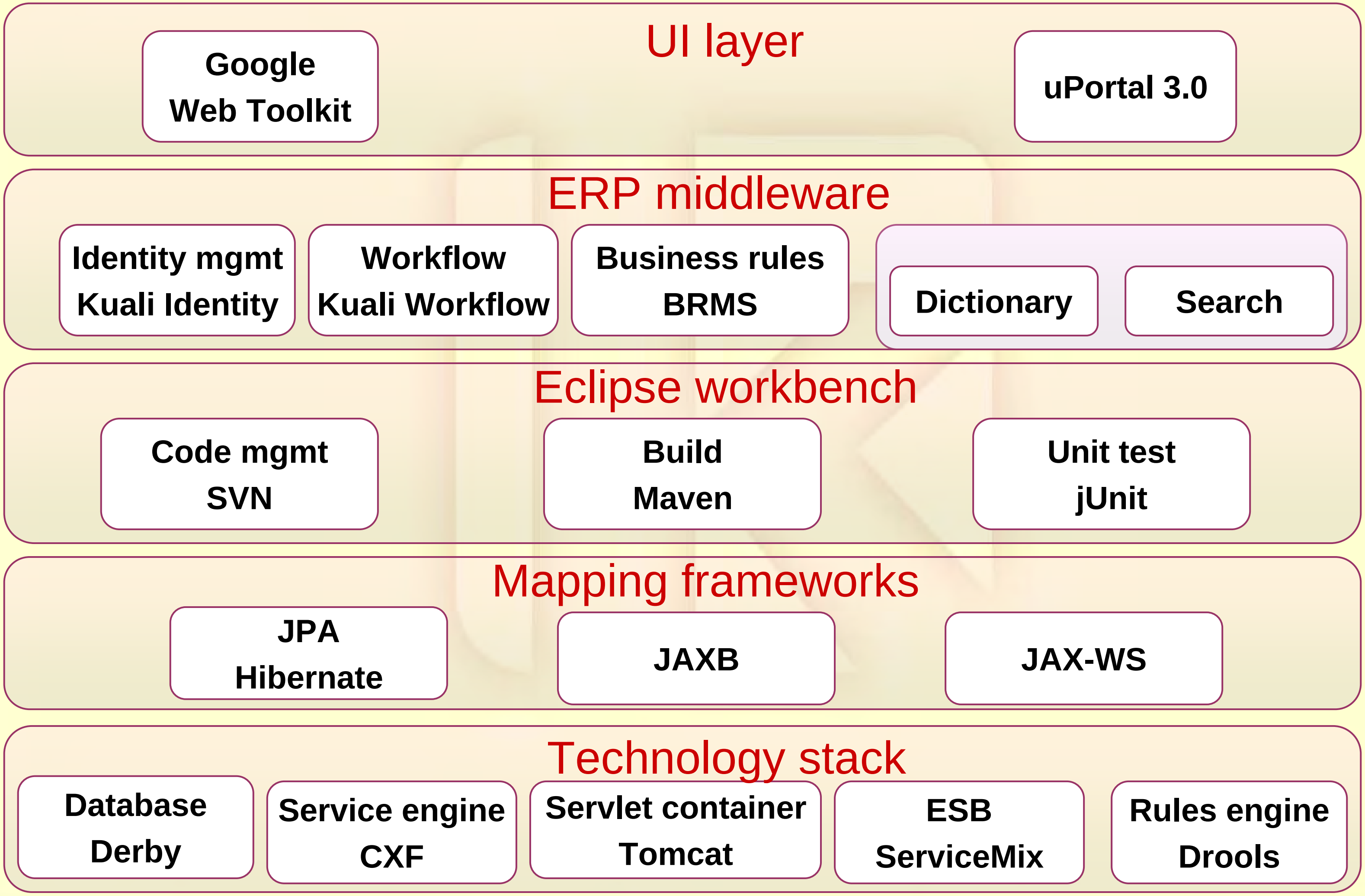
kuali student®

Architecture



User	Concierge	Portal		
Applications		Contact	Admission	Registration
Business Services		Notification service	Evaluation service	Enrolment service
Infrastructure Services		Rules service	Workflow service	Identity service
Information		Learning plan	Program requirements	Program availability

Tools and technologies



UI layer

- Google Web Toolkit
- uPortal 3.0

ERP middleware

- Identity mgmt: Kuali Identity
- Workflow: Kuali Workflow
- Business rules: BRMS
- Dictionary
- Search

Eclipse workbench

- Code mgmt: SVN
- Build: Maven
- Unit test: jUnit

Mapping frameworks

- JPA: Hibernate
- JAXB
- JAX-WS

Technology stack

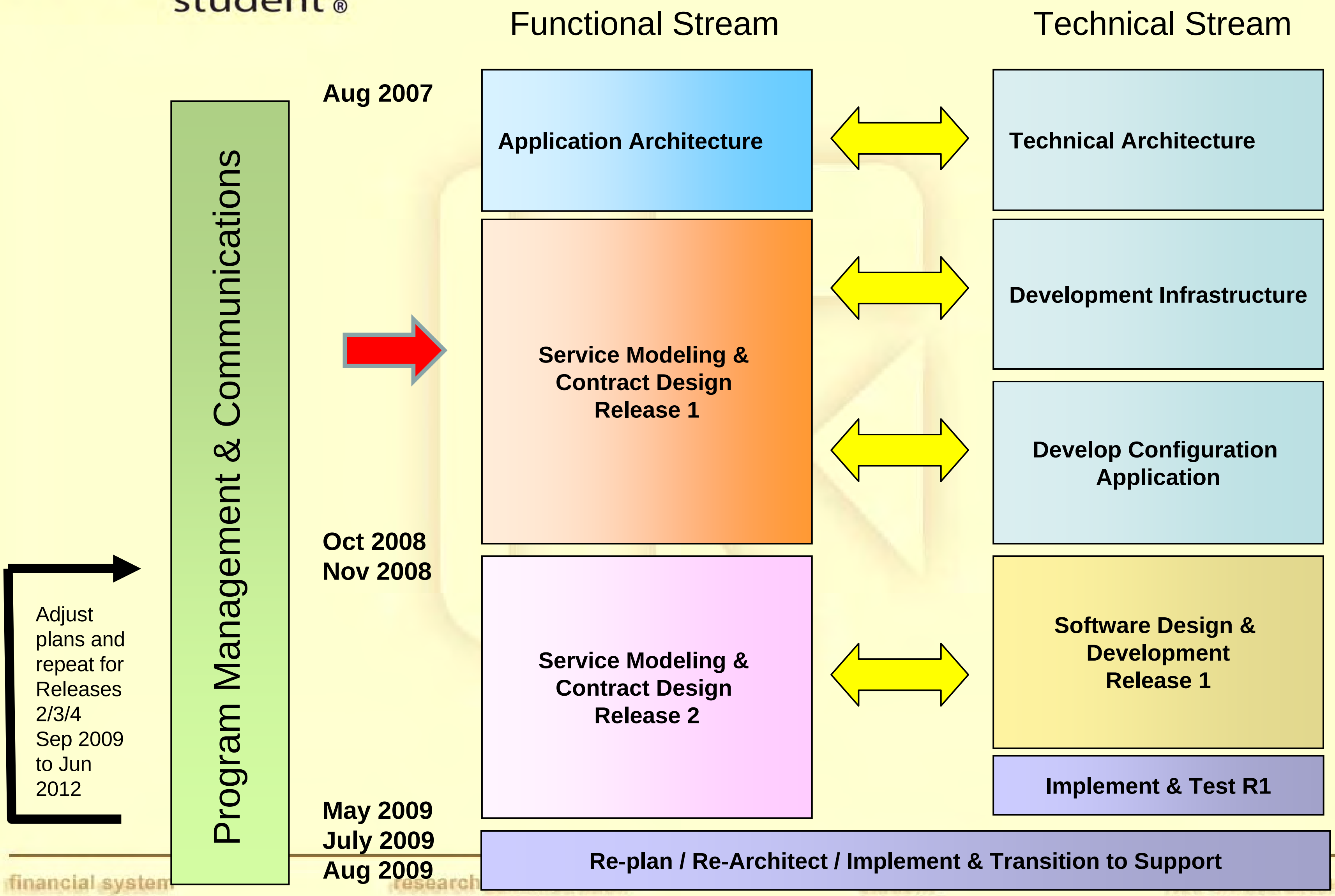
- Database: Derby
- Service engine: CXF
- Servlet container: Tomcat
- ESB: ServiceMix
- Rules engine: Drools

Development Approach

- KS is a 5 year project, with participating institutions distributed across a wide geographical area
- A project of this complexity requires a structured approach to development and project management
- Agility, phases, time boxing, reusability and iterations

- Separate implementation projects at each institution
  - Kuali Student does NOT include implementation
  - Product is “configured” by a separate team
    - dictionary; search; rules; BPEL; authorization

Phased Modular Approach



Functional Stream

Technical Stream

Program Management & Communications

Aug 2007

Application Architecture

Technical Architecture

Service Modeling & Contract Design Release 1

Development Infrastructure

Develop Configuration Application

Oct 2008

Nov 2008

Service Modeling & Contract Design Release 2

Software Design & Development Release 1

Implement & Test R1

Adjust plans and repeat for Releases 2/3/4 Sep 2009 to Jun 2012

May 2009

July 2009

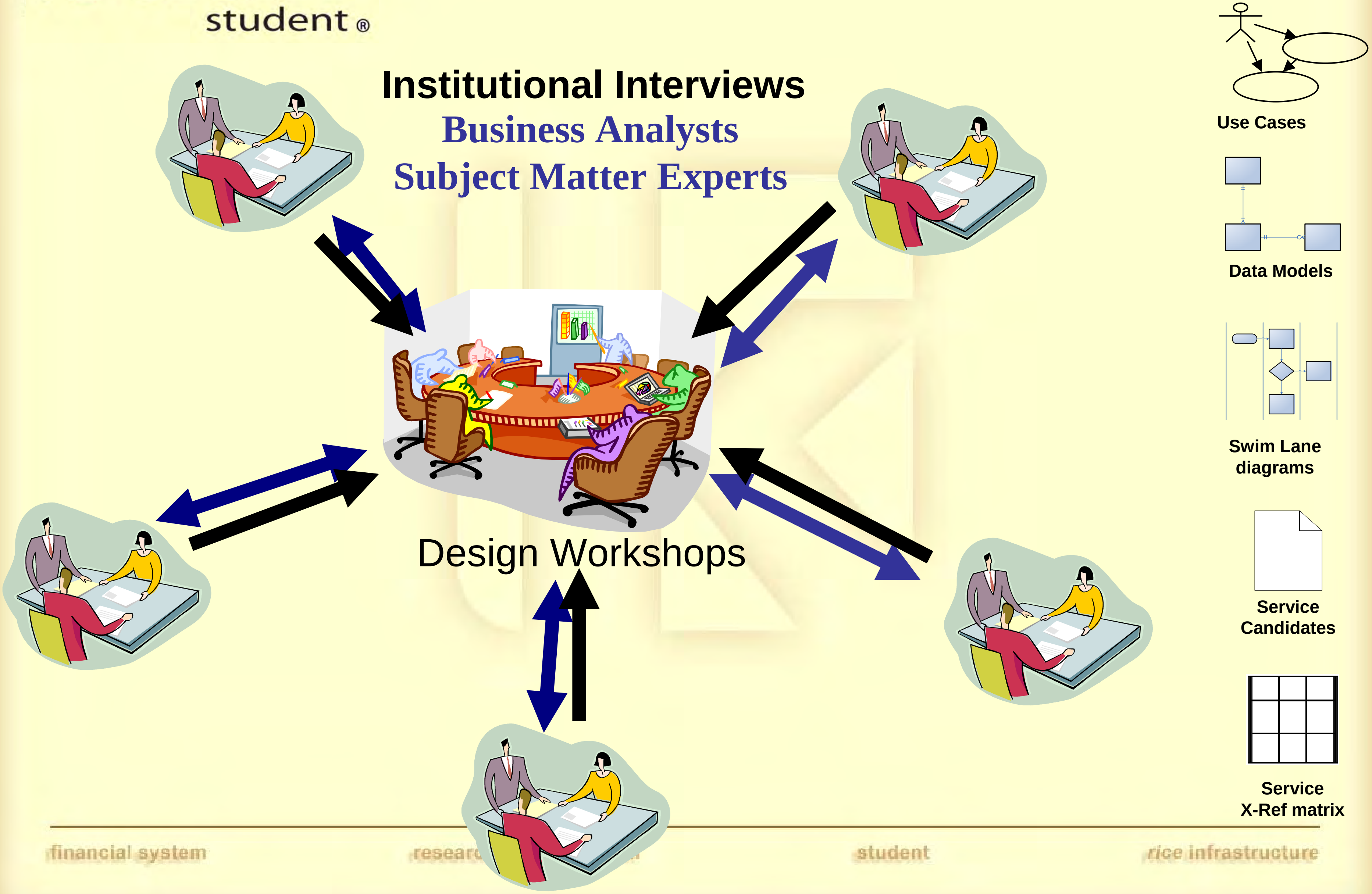
Aug 2009

Re-plan / Re-Architect / Implement & Transition to Support

Collaboration Tools

- Distance Challenges
  - East and west coasts
  - US and Canada plus UK and Australia
- Collaboration Tools
  - Face-to-Face Workshops
  - Video Conferencing
    - HD video conferencing bridge
    - Skype + Breeze
  - Recording
    - Wiki
  - Informal
    - IM, Googletalk, e-mail, phone

Document High-Level Requirements



Institutional Interviews

Business Analysts

Subject Matter Experts

Design Workshops

Use Cases

Data Models

Swim Lane diagrams

Service Candidates

Service X-Ref matrix

Vision: Sustainability

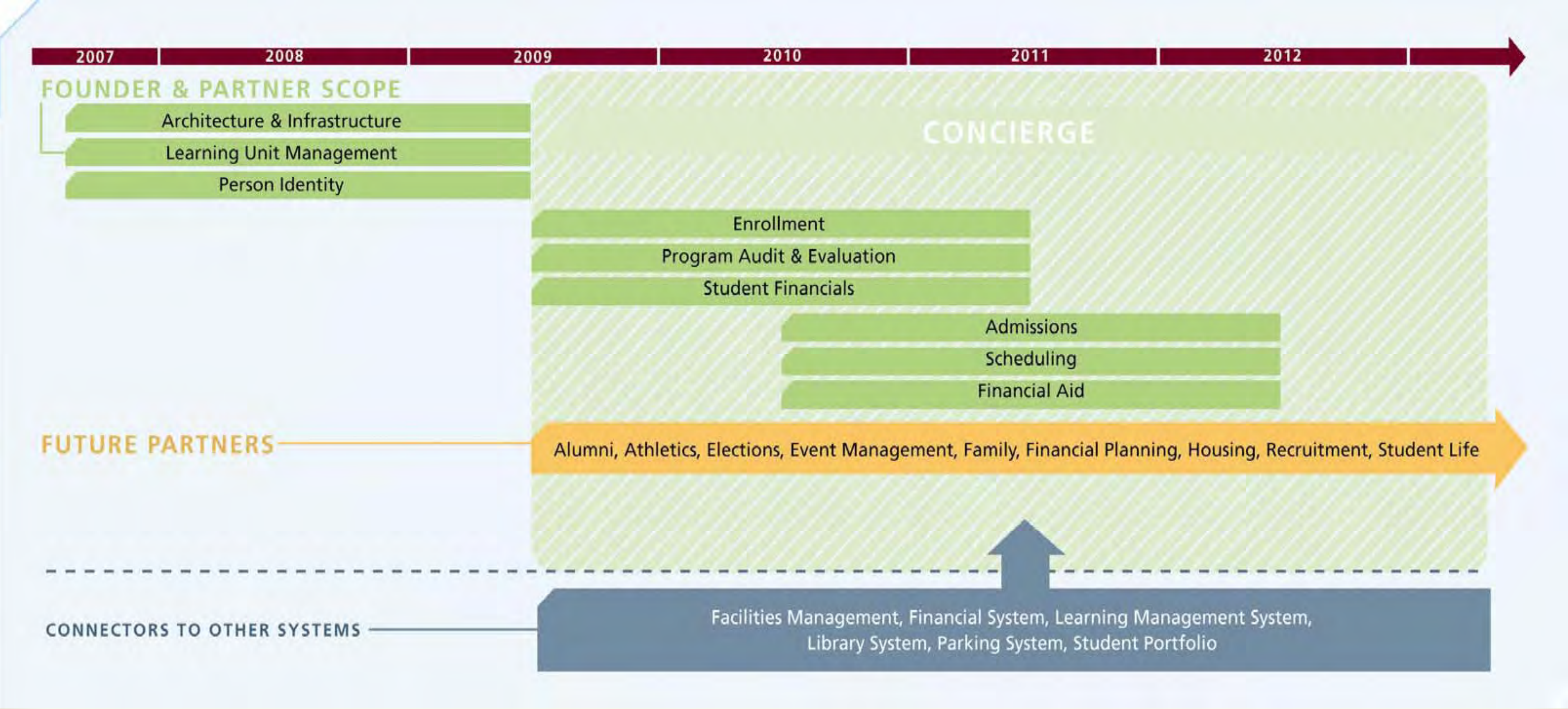
- Successfully implemented by the Founding Institutions
- Promote the adoption and implementation of Kuali Student
- Build a community of interest to sustain development
- Define product development and support
- Facilitate participation by vendors and service providers
- Evolve the technology and architecture

Where are we today?

- ✓ Legal agreements between Founders
- ✓ Partnership with Kuali Foundation
- ✓ Project charter approved
- ✓ $2.5 M Mellon grant awarded
- ✓ Project launch workshop July 30, 2007
- ✓ Technology architecture - recommendations completed
- ✓ Technology stack – proof of concept completed
- ✓ Application architecture - recommendations completed
- Service modelling & contract design
- Developers Workbench & Configuration Application
- Sustainment program - being developed with KF

kuali student®

Scope and Timeline


2007
2008
2009
2010
2011
2012
FOUNDER & PARTNER SCOPE
Architecture & Infrastructure
Learning Unit Management
Person Identity
CONCIERGE
Enrollment
Program Audit & Evaluation
Student Financials
Admissions
Scheduling
Financial Aid
FUTURE PARTNERS
Alumni, Athletics, Elections, Event Management, Family, Financial Planning, Housing, Recruitment, Student Life
CONNECTORS TO OTHER SYSTEMS
Facilities Management, Financial System, Learning Management System, Library System, Parking System, Student Portfolio

Information

Kuali.Org site:

http://student.kuali.org/

or Google: “Kuali Student”