Abstract

This document proposes the development and deployment of an Enterprise Service Bus infrastructure to manage, integrate, transform, and route the information flowing among UCLA’s electronic systems. The proposal includes an early adoption plan to deploy the service bus in time to perform data integration functions in the UC Path project.

Status of This Document

This is a candidate draft. The document is ready for internal review and approval. It is suitable for IT Services internal circulation and limited stakeholder review.

document version:

Working Draft 0003, January 26, 2013

document identifier:

draft-itservices-eb-business-case-0003
# Table of Contents

1. **EXECUTIVE SUMMARY** .................................................................................................................. 3

2. **CURRENT STATE ASSESSMENT** ................................................................................................... 3

3. **PROJECT DESCRIPTION** .................................................................................................................. 4
   3.1.1 Phase 1 (now to July 2013) ....................................................................................................... 6
   3.1.2 Phase 2 (September 2012 to July 2013) .................................................................................. 6
   3.1.3 Phase 3 Objectives (July 2013 to December 2015) ................................................................. 6

4. **PROPOSED SOLUTION** .................................................................................................................... 6
   4.1 TECHNOLOGY ................................................................................................................................. Error! Bookmark not defined.
   4.2 DEVELOPMENT ............................................................................................................................... Error! Bookmark not defined.
   4.3 OPERATIONS .................................................................................................................................. Error! Bookmark not defined.

5. **BUDGET** ........................................................................................................................................ 8
   5.1 PROJECT COST DETAILS ............................................................................................................... 8
   5.1.1 Hosting Services ........................................................................................................................ 8
   5.1.2 ESB Software & Tools ............................................................................................................... 9
   5.1.3 Consulting Services .................................................................................................................... 9
   5.1.4 Training ..................................................................................................................................... 9
   5.1.5 Staffing ..................................................................................................................................... 9
   5.2 ANNUAL OPERATIONS .................................................................................................................. 9
   5.3 PROPOSED FUNDING MODEL ...................................................................................................... 9

6. **ASSUMPTIONS AND RISKS** ........................................................................................................... 9
   6.1 ASSUMPTION: THE UC PATH PROJECT IS ON TIME. ................................................................. 9
   6.2 RISK: THE PUSH TO DEPLOY SERVICE MIX FOR UC PATH MAY LEAD TO FUTURE CONVERSION. ................................................................. 10
   6.3 RISK: THE PROJECT’S FUNDING MODEL AND SOURCE HAS NOT BEEN FULLY IDENTIFIED. .................................................................................. 10

7. **APPENDIX** ..................................................................................................................................... 11
   7.1 WHAT IS AN ENTERPRISE SERVICE BUS? .................................................................................. Error! Bookmark not defined.
   7.2 HOW IS AN ENTERPRISE SERVICE BUS DIFFERENT FROM A DATA WAREHOUSE? Error! Bookmark not defined.

8. **DOCUMENT HISTORY** ..................................................................................................................... 12
1 Executive Summary

UC Path marks a major milestone in the UC’s desire to improve its administrative operations efficiencies. UC Path seeks to standardize and streamline the UC’s HR and Payroll business processes. To fully realize that vision, UCLA needs to update its own IT infrastructure to enable scale, real-time, modular, and efficient information exchange between UC Path and UCLA IT systems. The plan is a step in realigning resources and services to improve UCLA’s IT efficiency to better support the institution’s mission in research and education.

This document proposes the development and deployment of an Enterprise Service Bus infrastructure to manage, integrate, transform, and route the information flowing among UCLA’s administrative information systems, whether hosted locally or remotely. The proposal includes an early adoption plan to deploy the Service Bus infrastructure in time to perform data integration functions for the UC Path project.

2 Current State Assessment

UCLA takes pride in pioneering and using technology to help advance its missions. Over the past 40 years, a broad spectrum of information systems have sprung up and entered UCLA community members’ daily lives. These systems evolved organically and adapted themselves to solve specific problem domains’ needs. As the need to exchange information among systems emerged, we invented ways to exchange information 2 systems at a time. For example, the financial system exchanges data with payroll directly. Payroll exchanges data with billing directly. Billing exchanges data with purchasing. Purchasing exchanges data with financial system. The list goes on.

This approach, while expedient and effective at a smaller scale, becomes problematic as the number of systems grows. First, as the point-to-point data connection increases, it becomes increasingly costly to manage the tangled web of data loading/transfer processes. Second, the processes are often developed to solve short term, tactical needs with little consideration for longer-term evolution. As business processes evolve, triggering data changes within systems, managing changes to these data transfer processes become major projects onto themselves.

These hard-coded, direct linked connections are often maintained separate with little coordination. Consequently, every application team winds up spending significant amount of time maintaining these connections. When data error occurs, troubleshooting often becomes a multi-team, multi-day, inefficient, and costly exercise.

UCLA hosts many different IT systems across different platforms, using a diverse mix of technologies, standards, protocols, developed over a period of time. There are multitude of point to point integrations which are costly to build and maintain. While these IT systems have served their purpose, it is not easy to build a comprehensive suite of services with legacy applications. ESB provides integration capabilities so that consumers can locate and consume services, regardless of the technology or hosting platform.

There are many IT groups in the campus, large and small, with heterogeneous IT environments. Sometimes they result in redundant services and applications.

Furthermore, a 21st century organization is increasingly depending on real-time flow of information to make timely business decisions. Our current data processes are largely built around daily or weekly batched

---

Figure 1: point-to-point data exchange is difficult to manage at scale
exchange. They lack the infrastructure support to enable real-time data exchange.

For example, the data integration analysis from the UC Path project reveals that the UCLA Personnel & Payroll System alone has more than 100 direct data integration points with other systems. That doesn’t include the several hundreds more indirect payroll data integrations through the data warehouse and downstream applications. As UC Path standardizes UC’s HR and Payroll business processes, data integration and migration is, in fact, one of the significant and costly challenges to implementing UC Path.

In addition, UCLA is already developing plans to replace its financial and purchasing systems with an ERP solution. The student information community is looking for better ways to improve student services quality and to better manage student information sharing among the dozens of UCLA student service. Furthermore, as UC strives to improve its operating efficiency by sharing technology and services, the UC applications community is eagerly looking for an organized, standardized way to access information across UC without having to build more one-off data extraction solutions.

It is time for UCLA to invest in an infrastructure to promote a healthy, secure, scalable, and cost effective way to manage information flow among systems. Deploying an Enterprise Service Bus is an industry proven approach to address the data management issues UCLA faces today.

**In layperson’s term: What is an Enterprise Service Bus?** (Section 6.1)

### 3 Project Description

The UCLA Enterprise Service Bus infrastructure (ESB) proposed in this document tackles several complementary needs for robust, cost efficient information management solutions:

In the near term, an ESB provides a streamlined data integration and transformation solution between UC Path and UCLA applications. By routing all UC Path data interfaces through a UCLA ESB, we vastly improve our ability to manage all data exchange between UC Path and UCLA applications. An ESB enables us to standardize and consolidate data transformations rules, alleviating the need to modify every PPS-connected application to adapt to the new UC Path data model. This layer of abstraction minimizes the project’s impact on each application. It also enables us to more easily evolve systems as data models and business processes change. In addition, by moving to an ESB, we reduce the number of UC Path to UCLA interfaces by shifting application-specific data parsing work to UCLA. Together, these streamlining measures lower UC Path’s data interface conversation cost.

Furthermore, the ESB provides a robust asynchronous, real-time data transmission platform. This allows us to
send and receive time-sensitive HR/Payroll changes in matters of seconds instead of hours or days. The real-time flow of information means that on-demand, HR data-driven service provisioning and de-provisioning can finally be a reality. With the ESB in place, we will, over time, able to enable real-time, HR event triggered, onboarding processes such as account creation, BruinCard printing, and Parking permit assignment.

ESB provides a single platform for management to monitor service state and SLA. ESB enables us to extend legacy and mainframe applications without modifying the core, underlying services, by building wrappers and protocol transformations.

In addition, the Student Integrated Web Experience (IWE) project has also expressed interest to use the ESB as its real-time, asynchronous application-to-application message exchange mechanism. The ESB provides IWE with a robust, standard, and modular framework, enabling IWE to gradually and flexibly add additional student services applications into its environment.

In the longer term, we plan to work with campus stakeholders to assess campus needs and to deploy a robust, scalable, and adaptable campus information integration hub. This hub provides a common platform to securely and efficiently route, transform, and deliver information flowing across UCLA’s information systems. A common data hub vastly improves UCLA’s ability to monitor and manage information processing across the university. Routing services through a service bus also enables simpler, more gradual future UCLA system evolutions by encouraging applications to enforce a standard, modular, and loosely coupled system integration strategy.

To implement, we recommend a 3-phase deployment to:

a. Quickly implement a data integration solution to meet UCPath integration needs;

b. As much as possible, leverage UC Path’s technology investments for future endeavors, hence maximizing UCLA’s return on UC Path;
c. Scale the solution as the demand increases; 
d. Evolve the technology to match potentially shifting funding model and campus demands.

3.1 Objectives

The following are the objectives for each phase:

3.1.1 Phase 1 (now to July 2013)

Phase 1 is about meeting the demands of UC Path. Here, we focus on:

- Select ESB product
- Develop UCPath integration architecture and plan
- Deliver a working ESB solution by July 2013 to meet UCPath data integration needs

3.1.2 Phase 2 (September 2012 to July 2013)

Phase 2 focuses on campus engagement and planning for the eventual broad adoption of the ESB as a campus infrastructure. It is also where we intend to tackle IWE integration concerns. Phase 2 objectives include:

- Develop long term architecture, governance, and operating strategy for campus-wide adoption
- Develop detailed ESB funding and service model
- Conduct full ESB and tools evaluation for Phase 3 deployment

3.1.3 Phase 3 Objectives (July 2013 to December 2015)

Phase 3 is campus-wide deployment. Objectives include:

- If necessary, deploy new ESB and tools
- Integrate ESB with new services
- Update UCLA’s identity management infrastructure to fully adopt ESB
- Evaluate and select additional existing UCLA data processes for conversion/migration
- Collaborate with UC community, Develop and deploy UC-wide ESB-driven data routing architecture

3.2 Proposed Solution

3.2.1 Technology

A subset of the UC Path Technical team, in conjunction with middleware and SOA experts across UC, has evaluated Fuse ESB. Fuse ESB is from Radhat, built on top of the popular, open source product from the Apache Foundation - ServiceMix. The response was positive. Fuse ESB is a proven solution and can handle at least phase 1’s needs.

Due to the tight UC Path project schedule, we are unable to conduct a comprehensive evaluation of all viable ESB products in phase 1. To make sure we meet UC Path’s deadlines, we intend to deploy Fuse ESB as an “early adopter” ESB solution.

In the 2nd phase, to make sure the ESB infrastructure meets the broader campus needs, we plan to field
additional campus requirements, evaluate our lessons learned from adopting Fuse ESB, and to conduct a broader search for a long term ESB solution.

The project assumes that Fuse ESB will be hosted and operated from IT Services’ data center. Therefore, we prefer the solution to run on Redhat Linux 6.0 inside IT Services’ virtual server infrastructure.

The project team will publish detailed technical solution architecture in phase 1 of the project.

### 3.2.2 Development

For phase 1 of the project, all development work will focus on satisfying the integration requirements for the UC Path project. As a result, the development team will consist of largely the UC Path Technical team. Additional consulting resources will assist with the installation, configuration, and deployment of the Service Bus infrastructure.

In phase 2, recruitment for the eventual permanent ESB team (slated to be part of the IT Services Information Management Services) begins. We also switch focus from rapid system development to preparing the service for production operation. In addition, the team also will begin engaging the campus to assess campus needs and prepare for a broader launch in phase 3.

In phase 3, the ESB enters production service. The team begins work to scale the service and to integrate additional applications. Depending on campus demands and market trend collected in phase 2, we may reassess whether to continue with Fuse ESB or look for a different vendor product.

We intend to deploy the ESB using an application-as-a-service model. IMS team builds and operates the ESB platform. The application teams consume the platform by developing integration components to expose and to consume services in its applications. IMS team builds and exposes monitoring, logging, dashboard to consumers, within the scope of consumer applications.

### 3.2.3 Operations

The project assumes that IT Services will host and manage the deployed solution. It will operate as a part of IT Services’ service offering. Specifically:

- The Information Management Services team will have program management and ongoing product development responsibilities for the Service Bus infrastructure. It will also provide consulting resources to assist project teams with integration work.
- The Infrastructure Services team will provide system hosting and data center services according to its data center services offering.
- The Customer Services team has the front line service desk support role.
4   Budget

The following table summarizes the projected project cost. Note: the bulk of the initial project development staffing cost is funded through UC Path’s data integration project.

<table>
<thead>
<tr>
<th></th>
<th>FY 2012-13</th>
<th>FY 2013-14</th>
<th>FY 2014-15</th>
<th>Project Total</th>
<th>Permanent (Annual)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Phase I &amp; II Cost</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hosting Services</td>
<td>144,790</td>
<td>-</td>
<td>-</td>
<td>130,000</td>
<td></td>
</tr>
<tr>
<td>ESB Software &amp; Tools*</td>
<td>15,000</td>
<td>-</td>
<td>-</td>
<td>15,000</td>
<td></td>
</tr>
<tr>
<td>Consulting Services</td>
<td>75,000</td>
<td>-</td>
<td>-</td>
<td>75,000</td>
<td></td>
</tr>
<tr>
<td>Training</td>
<td>25,000</td>
<td>-</td>
<td>-</td>
<td>25,000</td>
<td></td>
</tr>
<tr>
<td>Staffing</td>
<td>611,600</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td><strong>Phase III Cost</strong></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hosting Services</td>
<td>-</td>
<td>144,790</td>
<td>144,790</td>
<td>289,580</td>
<td>144,790</td>
</tr>
<tr>
<td>ESB Software &amp; Tools*</td>
<td>-</td>
<td>15,000</td>
<td>15,000</td>
<td>30,000</td>
<td>15,000</td>
</tr>
<tr>
<td>Consulting Services</td>
<td>-</td>
<td>50,000</td>
<td>25,000</td>
<td>75,000</td>
<td>10,000</td>
</tr>
<tr>
<td>Training</td>
<td>-</td>
<td>10,000</td>
<td>-</td>
<td>10,000</td>
<td></td>
</tr>
<tr>
<td>Staffing</td>
<td>-</td>
<td>629,948</td>
<td>648,847</td>
<td>1,278,795</td>
<td>668,312</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>871,390</td>
<td>849,738</td>
<td>818,637</td>
<td>2,539,765</td>
<td>838,102</td>
</tr>
<tr>
<td><strong>Funding Source</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC Path Project</td>
<td>611,600</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Core Investment**</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
</tr>
<tr>
<td>Direct Charge**</td>
<td>-</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
</tr>
</tbody>
</table>

* the ESB software and tool cost are based on selection of ServiceMix.

** placeholder line items for to be determined funding sources.

4.1 Project Cost Details

4.1.1 Hosting Services

The hosting services line item includes the cost of hosting and operating the infrastructure required to host the production, testing, and development systems. The hosting budget is estimated based on the IT Services Data Center Hosting Services rates, and includes the following items:

<table>
<thead>
<tr>
<th></th>
<th>Unit</th>
<th>Monthly Unit Cost</th>
<th>Annual Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Production VM's</td>
<td>6</td>
<td>62</td>
<td>4,464.00</td>
</tr>
<tr>
<td>Development &amp; Test VM's</td>
<td>12</td>
<td>62</td>
<td>8,928.00</td>
</tr>
<tr>
<td>Network Services</td>
<td>36</td>
<td>10</td>
<td>4,320.00</td>
</tr>
<tr>
<td>Storage (GB)</td>
<td>1024</td>
<td>0.40</td>
<td>4,915.20</td>
</tr>
<tr>
<td>Back Up/Recovery (GB)</td>
<td>1024</td>
<td>0.24</td>
<td>2,949.12</td>
</tr>
<tr>
<td>Disaster Recovery - VM</td>
<td>24</td>
<td>15</td>
<td>4,320.00</td>
</tr>
<tr>
<td>DR – Storage (GB)</td>
<td>1,024</td>
<td>1</td>
<td>12,288.00</td>
</tr>
<tr>
<td>System Administration</td>
<td>18</td>
<td>475</td>
<td>102,600.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td><strong>144,784.32</strong></td>
</tr>
</tbody>
</table>
4.1.2 ESB Software & Tools

The ESB software evaluation and selection process is still underway. Given that it is likely the project will adopt open source solutions, the current software estimate includes the cost of subscribing commercial support for the software only. This may change significantly if the project adopts commercial solutions.

4.1.3 Consulting Services

The project intends to retain commercial consulting resources to help with the initial installation, configuration and deployment of the ESB infrastructure. The cost estimate is based on an hourly rate of $150.

4.1.4 Training

The training cost includes the cost of training UCLA staff to manage/maintain the deployed ESB service.

4.1.5 Staffing

To support UCLA’s demands, we project that a fully assembled ESB team will include 2 senior SOA engineers, 1 support engineer, and 1 ESB Program Manager, totaling 4 FTE.

Staffing cost is calculated based on the average annual salary of a senior enterprise SOA developer/engineer ($110,000) multiplied by the 39% benefits adjustment. For FY2012-13, the average FTE rate is $152,900. The table also reflects an annual 3% salary adjustment to account for merit and inflation.

4.2 Annual Operations

Once fully deployed, the ESB becomes a mission critical middleware component spanning all major UCLA applications. Where possible, we intend to leverage existing IAMUCLA resources to provide design and logistics support. Although to properly support its ongoing operation and development, we require an addition of 3 permanent FTE.

4.3 Proposed Funding Model

The Enterprise Service Bus is a common infrastructure that we want to encourage applications and departments to take advantage of. In fact, it derives most of its value from ubiquitous adoption, similar to UCLA’s Single Sign-On Service. To encourage adoption, we recommend funding the infrastructure centrally, to allow campus applications to adopt the basic use of the technology without incurring heavy up front cost. The annual operating cost outlined in this document reflects the projected cost to operate such basic service.

We can augment the central funding with additional direct charge services to meet the needs of larger, more demanding integration/usage scenarios.

5 Assumptions and Risks

5.1 Assumption: The UC Path project is on time.

Priority: High
Description

Much of the proposed implementation plan is predicated on the condition that UCLA needs to complete its UC Path implementation by July 2013. This constraint is the major reason we are pushing ahead with a product without first conducting a thorough product evaluation. If the UC Path project delays significantly (we will need to find out in the next 30 days), we should switch project priority to perform a more complete evaluation first.

5.2 Risk: The push to deploy ServiceMix for UC Path may lead to future conversion.

Priority: Medium

Description

While ServiceMix meets the needs of UC Path project, it is possible that we may choose a different product when we conduct the campus needs assessment and the subsequent product evaluation in phase 2. If that happens, we face conversion work to migrate UC Path integration work to the selected campus ESB infrastructure.

Mitigation

First, we believe that our exposure here is low. ServiceMix is a widely deployed and well-regarded product. UCLA already uses it for the BruinBills consolidated billing application. UC Berkeley has also deployed it as its campus service bus. If anything, there is mounting support within the UC middleware community to standardize on ServiceMix.

Even if we did have to move, the nature of Service Bus architecture means that we can gradually migrate services across buses without creating service disruption. These migrations can be done within each program’s natural enhancement cycle, eliminating the need to trigger massive conversion projects.

5.3 Risk: The project’s funding model and source has not been fully identified.

Priority: High

Description

The urgency of UC Path calls for this project to begin development before we can identify permanent funding model and source to support the ongoing operations of the Service Bus. Existing IT Services teams cannot sustain the infrastructure without additional resource.

Mitigation

Developing the proper long term funding model for the ESB service is a key objective in phase 2 of the project. It determines whether the project can proceed to phase 3 – campus adoption.

For components developed and deployed in Phase 1, a default, albeit not ideal, model would be to recharge each application requiring HR/PPS data from UC Path for the cost of maintaining the bus.
6 Appendix

6.1 What is an Enterprise Service Bus?

Imagine the information we exchange across electronically each day as cargos shipped across oceans, and that the systems or services sending and receiving that information are the businesses sending and receiving the cargo packages. An Enterprise Service Bus (ESB) would be a port facility.

A shipping port facility is a critical transportation hub designed to facilitate the sorting, inspection, routing, and delivery of massive amount of cargo. An effective port allows businesses to streamline shipping by offloading logistical details such as customs inspection and transferring cargo between transports (e.g. ships to train, to trucks) to a dedicated, optimized facility. It also reduces shipping cost by aggregating smaller shipments into larger, more economical shipping vessels.

An ESB performs similar tasks for information. It provides a common hub to aggregate information from a variety of information sources, aggregate (or parse) the information into digestible chunks, manages information access security, and route the processed information to its intended target on transports suitable for the target.

Just as port facilities enable a nation to effectively manage its commerce at a massive scale, an ESB enables UCLA to effectively manage its information flow across the enterprise and beyond.

Here is a more technical explanation of what an ESB is, can do:

ESB is a services gateway that acts as a central coordinator of a variety of services. It acts as a central hub where services are mediated, monitored, operated. Provides a clear picture of service state and SLA to management.

ESB implements SOA through middleware that offers virtualization and management of service interactions between communication participants. Thus, this flexible connectivity layer could help connect and integrate an organization's IT infrastructure across many differing systems and locations reliably and securely while reducing the number, size and complexity of application interfaces.

ESB facilitates modular, standards-based applications that are dynamic and require high scalability, high availability, tight security, platform heterogeneity, monitoring capability.

An ESB is an integration technology that...

- delivers a Service Oriented Architecture (SOA) through a fabric of service end points
- uses rich messaging and transformation capabilities for reliable, any-to-any delivery of events and data
- supports service virtualization to abstract the location and internal workings of providers from consumers
- enables orchestration of lower-level sub-services into higher order services
- supports ongoing system changes via configuration rather than reprogramming

6.2 How is an Enterprise Service Bus different from a Data Warehouse?
An Enterprise Service Bus (ESB) is a hub for information flow. It is designed primarily to effectively manage the flow of information from one system to another. A data warehouse is an information library, where data is aggregated, sorted, transformed, and stored, primarily for human consumption.

An ESB and a data warehouse are in fact complimentary technologies. An ESB does a great job at managing complex information flow. A data warehouse excels at collecting and transforming that massive amount of information into useful reports and analyses for human use.

6.3 ESB and SOA

SOA is an architectural style, a pattern, not a product. ESB is a software product. ESB is a core component of a service-oriented architecture (SOA). SOA provides the ability to decouple the links between business functions and specific applications by isolating service definition and usage from the underlying service implementation. ESB offers an incremental approach to SOA with the goal of extending the access of existing applications from department-wide to enterprise-wide.

7 Document History

Revision 0001 ...6/10/2012 ................................................................. initial draft document
Revision 0002 ...7/6/2012 ................................................................. candidate draft
Revision 0003 ...1/26/2013 ................................................................. release candidate