Website Self-Service Tools – Software Component Acquisition and "Death by a Thousand Cuts"

Paul D. Witman
pwitman@callutheran.edu
School of Business, California Lutheran University
Thousand Oaks, CA 91360

Abstract

This case provides a real-world study for students enrolled in a project management or systems development course. The case describes Online Banking Corporation (OBC), and its efforts to develop a new set of web hosting and maintenance tools for its clients. OBC began the project with the goal of modernizing and consolidating their web hosting platforms. The project leveraged vendor-built portal server software, a new area of technology for OBC, and required selection of a vendor for the server components. It also required specification and construction of software for the end users of the web site (bank customers), and for the site administrators (bank staff). Significant cross-organizational collaboration was required within OBC, along with careful vendor management to ensure that all required features were available, and that the product met the resource constraints required for profitable operation. Key concepts around technology selection, project management, and vendor management come to light as OBC's new product begins its lifecycle. Both business and technology issues are raised, and students will need to analyze the actions, inactions, and positions of the various players to determine both improved recovery approaches as well as ways to prevent such issues in the future.

Keywords: project management, vendor management, portal server, web site, self-service, new technology, Software as a Service, scalability, reliability, functional and non-functional requirements, case study.

1. BUSINESS BACKGROUND

Online Banking Corporation (OBC) was formed in the mid-1990s to capitalize on the opportunity created by emerging online commerce and the limited technical capacity of smaller banks and credit unions. Together, OBC referred to its client base as "financial institutions", or FIs.

OBC offered a wide range of e-banking services, including online banking tools for consumers and small businesses, as well as marketing tools for the FIs. It also offered a few different flavors of web site hosting, to provide the FIs' public faces on the Web.

All of these services were offered as just that – services. OBC believed that it could provide a significant advantage to its FIs by providing excellent operational service of its software, at a lower cost than the FI could operate on its own. It sought to achieve economies of scale by operating services for multiple FIs on each server in its data center, and leveraging such things as information security, disaster recovery, high availability, and ongoing maintenance across multiple clients. This model was known at the time as "Application Service Provider", or ASP. It has since been called Software as a Service (Vaquero, Rodero-Merino, Caceres, & Lindner, 2009).

OBC was a company of less than 1,000 employees, an amalgamation of several acquisitions since its founding. Its most recent acqui-

sition had been completely integrated a few months before. The software development and operations efforts required to enhance the products and migrate the acquired customers' data had been a significant extra effort beyond the day-to-day operations and development activities.

At the time, OBC offered two flavors of web hosting capabilities. The first was based on a Unix web server, and required OBC staff to make any substantive updates to an FI's web site. Each FI had the capability of editing some types of structured data – rates, account types, branch locations, hours, etc. – but not the ability to modify layout, unstructured text, images, or other such web content.

The second flavor of web hosting was more flexible and capable for the FI, and was based on Microsoft technology. For a variety of reasons, OBC had set its architecture on a Unix/Linux path and did not want to build out a new solution for web hosting on a Microsoft server platform.

A small number of FIs used HTML editing tools to create their own web pages and upload them to the web server. However, this created issues of support (FIs called when things didn't work, and OBC staff had to debug other people's HTML) and reliability (broken HTML prevented access to OBC's other products).

2. PROJECT ACTIVITIES

High-level requirements for the new web hosting product, dubbed WebService, were developed by OBC product managers led by Clint, and by web services staff. The requirements included functional elements, including required features of the end user and FI administrative views, as well as OBC's non-functional elements, including performance, reliability, scalability, and the like.

Key requirements included:

- Ability to make modifications to a web page and preview it before moving changes to production.
- Ability to allow a separate approver, independent of the person making the change. This met the industry's general rule for dual-control, or so-called "maker-checker" functions, to ensure against a single rogue employee.
- Ability to capture a snapshot of the entire site at any given point in time, for audit purposes. For example, auditing agencies would want to see that the FI's interest rates, as published on their web site, matched those offered to the new accounts opened while that version of the site was visible to customers.

In addition, OBC's business model required that the portal server (Wege, 2002), and WebService, be operated so that hundreds of clients shared the same computing hardware – referred to as *multi-tenancy* (Mietzner, Leymann, & Papazoglou, 2008). For load balancing, OBC further required the ability to readily move a single FI or group of FIs from one computing engine to another.

Key participants in the WebService project are shown in Table 1.

Name	Firm	Role
Clint	OBC	Product Manager
Mitchell	OBC	Head of Architecture Team
Shawn	Vendor A	Salesperson
Jerry	OBC	Head of Operations and Development
Helen	OBC	Overall Program Manager
Hal	OBC	Lead Developer
Mary	OBC	Project Manager
Mike	Vendor A	Professional Services Director
Steve	OBC	Technical Project Manager

Table 1 - Key participants in the WebService project

Selection of Portal Server

OBC's architecture team, headed by Mitchell, was tasked with evaluating portal server products to serve as the foundation for WebService. Mitchell, a 15-year veteran of architectural roles for other companies, had come to OBC a year earlier and brought with him experience with the Rational Unified Process (RUP) (Gallagher & Brownsword, 2001; Kruchten, 2003) and its approach to software development.

Using RUP's concepts as a foundation, the team chose three vendors to participate in a comparison and test of portal server capabilities. The vendors are summarized in Table 2.

Vendor	Background	Portal offering
A	Relatively young company; provider of a variety of infrastructure software components	Reasonably mature product, but key feature (multi-tenancy) missing in current release.
B	Large vendor, offering wide range of hardware, software and services.	Portal offering relatively immature; required integration with 3rd party product to meet requirements
C	Large vendor of enterprise software; new to the infrastructure market	Relatively new to portal server market.

Table 2 - Vendors considered for portal server

Mitchell's team evaluated only commercial off-the-shelf (COTS) products, using a matrix based on the requirements attributable to the portal server software. Each product was scored based on its capabilities (some tested, some stated by the vendor). Open source products were not considered because OBC had little experience with the open source market.

Vendor C was eliminated early in the process due to its lack of depth in the portal server market, and the relative immaturity of its product. Mitchell's team tested Vendor B's product thoroughly, and it was a strong contender based on its product and existing relationship with OBC. However, Vendor B required use of a third party component built by a small start-up company to meet all of OBC's requirements. This was deemed to be too great of a risk to the overall project.

Vendor A ultimately prevailed, as it offered what appeared to be the best combination of product maturity, completeness of functional capabilities, and risk. Vendor A's product also integrated well with OBC's existing content management system, which managed elements of content across all of OBC's FIs. An overview of the initial architecture can be found in Figure 2(A).

Mitchell's team knew that some of OBC's requirements would not be satisfied until the next major release of the portal server was delivered but Vendor A promised that this version was well along in its development cycle and that it could meet OBC's stated development goals.

Business Relationship

Shawn, sales person for Vendor A, was interested in working with OBC not just for its portal server business, but also for its potentially much larger application server business. For comparison, OBC had forecast approximately 16 CPUs of capacity required for production portal server operations over a 24 month period, while its application server needs were likely to be in the dozens or hundreds of CPUs over that same period. (Server software products were often priced on a per-CPU basis.) Shawn was interested in winning the portal server business as an entrée into the application server business.

Discussions proceeded for about a month to identify a business model that would work for both organizations. Vendor A's normal model charged fees for all CPUs running their products. For reliability and disaster recovery purposes, OBC generally ran three times as many CPUs as were needed for their known peak loads, making Vendor A's proposed business model untenable for OBC. The amortization of the capital costs for the software licenses would dwarf the anticipated revenue for the product.

OBC and Vendor A ultimately agreed to a set of terms that balanced the risks and opportunities of the relationship. OBC agreed to buy portal and application server licenses for WebService if Vendor A met the functional and non-functional requirements for production. OBC agreed to review its broader application server needs and to consider Vendor A's products after successful production delivery of WebService to its clients.

Project Team

Following vendor selection but before finalizing the contract, Jerry, the head of OBC operations and development, appointed Helen as the overall program manager to form a team to build the WebService product. Previously, Helen had headed the Software Development group at OBC, and had recently transitioned into a new role supporting the OBC sales staff on large-customer relationships, as well as leading the technical integration of the recent acquisition. Helen was well-known and respected across the company for her technical

skills and ability to work with both business and technical staff. Helen had 20 years of software development and management experience at banks and 2½ years at OBC.

Jerry gave Helen a target of early 2004 to have a version of the product available for sale. He asked Helen to put together a team and a plan to confirm her commitment to the schedule goals and raise any issues and concerns. An overview of the project timeline is provided in Figure 1.

Jerry assigned Helen the WebService project based on Helen's experience with the development and operations groups, and ability to work with other groups across the company. She had been involved only on a very limited basis with the portal server evaluation process.

Helen formed a team including Hal, a software engineer and experienced Web developer, along with numerous staff from data center operations, security, product management, and marketing, and a company-wide project manager, Mary. Mary was to keep track of high level schedule deliverables and plans from across the company. This included the technical team as well as marketing, product management, legal, and other functions. The cross-functional team was intended to collaborate and be able to rapidly assess progress and adjust plans as opportunities and issues arose.

Technology Challenges

The team encountered a number of challenges as it worked through the process of obtaining the vendor software, installing it, and creating the tools to edit web site content, save it in the content management system, approve content for production, and display pages to the end user.

First, there were issues with performance and memory usage. The team knew that the first version of the portal server, PS2003, did not support all of the scalability requirements. However, Shawn had stated that PS2004, due out early in the fourth quarter of 2003, would meet those requirements.

The team tried several configurations and installation options to reduce memory usage – with limited success. Some of the changes required changes to the FI administration tools, and affected the development of those tools. The interrelationship of the installation with the development created both bottlenecks and frustration among the team.

PS2004 arrived in early 2004, about 3 months later than it had been promised originally. After the OBC team had completed installation and testing and Hal had updated the software to work with PS2004, the team tested the new version for compliance with its requirements. The team worked hard to make up the lost time and get close to their early 2nd quarter delivery date.

Another key requirement was the ability to move FI web sites from one server to another in the OBC data center for load balancing. Shawn had stated that this would be possible with PS2004, but might require some additional work. Helen worked with Shawn to understand this before PS2004 shipped, but getting firm, specific details proved difficult.

After receiving PS2004, Helen and the operations team worked with Vendor A to understand how to migrate a site from one server to another. In addition to moving the site's object code, the portal server required that a collection of configuration data for the FI also be moved. The configuration data was stored in a proprietary database, and there were no tools for moving this data.

The configuration data included details about how the site was to operate, along with user preferences. The user preferences defined how each end user had chosen to set up their personal portal view. Loss of this data would mean that the end user's view would revert to the site's default view if it had to be moved to a different server. Helen stated OBC's position to Shawn that this data had to be moved with the site.

Helen's boss, Jerry, inquired regularly about the status of the project. Knowing the ongoing difficulties and issues, he asked Helen in early 2004, "What do you need to be successful?" At that time, the key known and unresolved issue was the inability to migrate the configuration data. Helen asked Jerry for funding to cover the cost of construction of tools to solve the problem, believing that this was the key obstacle to success.

Shawn proposed that Vendor A's professional services staff, headed by Mike, define and build a configuration migration (CM) tool to do the job. Helen's boss, Jerry, approved spending nearly $100,000 to build the tool, contingent upon its successful implementation. Mike's team at Vendor A defined and reviewed

requirements with OBC, and began building the tool.

After several months, missed deadlines, and reductions in requirements (to try to shorten the timeline), Mike's team acknowledged that the task was actually more challenging than they had thought. The expertise required for the task was in short supply. Some features that had been promised proved to be essentially impossible (e.g., migration of user preferences), and Vendor A's delivery time ultimately absorbed all the slack built into the OBC schedule.

Through each technical challenge, whether encountered in software development or operations, the project team collaborated to find solutions from within OBC. In general, each team member was willing to brainstorm potential solutions, to do extra work to test fixes, and offer compromises in the interests of doing what was best for the project. Vendor A's staff also did their best to support the project, within the constraints imposed by their organization.

People Challenges

Hal reported to Helen for the purposes of the WebService project, but still reported to another manager for his other projects, yearly performance reviews, etc. Due to Helen's other responsibilities for sales, Hal's line manager, Steve, regularly checked in with him. Steve learned in late 2003 that Hal was frustrated with the project's progress. He asked Hal what he could do to help.

When Steve raised this issue with Helen, he volunteered to take on the task of technical project management, to ensure that the project's requirements of the vendor were clear, and that the vendor was accountable for their deliverables. Steve had noted that this aspect of the project seemed to be loosely managed, and Helen agreed. While Mary was responsible for overall project management, she wasn't technically skilled enough to assess and interpret both OBC's needs and Vendor A's commitments.

Steve worked with Vendor A's technical support and professional services staff to clarify both deliverables and timelines as well as support processes. One of the challenges the OBC team faced was that it was pushing the capabilities of the portal server further than other customers had. Therefore, their initial contacts with Vendor A technical support often proved fruitless and required escalation to their portal engineering team.

This was an issue for both firms, as it ate away at staff time needed to create the new functionality that OBC had requested. Vendor A's technical support staff members weren't familiar enough with the model that OBC was trying to implement. The process of escalating to the engineering staff also delayed response times to OBC for resolution of existing technical issues.

Deliveries

OBC announced its new product to potential customers in early 2004 – before the product was complete. While this was typical for the industry, the announcement was made before it was clear to OBC that there was an end in sight. There were still unresolved issues with the CM tool, as well as scalability issues with the portal server.

Ultimately, the OBC team decided to go live to a small number of customers before the CM tool was available, and with the remaining scalability issues outstanding. Given the small number of customers and the known size of the customer base, scalability issues could be predicted in advance and managed in a more labor-intensive fashion. (OBC's operations staff would have to manually reconfigure the portal servers if a client needed to be moved.) WebService was available only for very limited sales to minimize the impact on operations until the issues could be resolved.

For its next revision, the OBC project team determined that it needed to find a new path to achieve the scalability and operability it required. However, its FI admin tools were tightly coupled to the Vendor A portal server. At this time, standards for the creation of *portlets* (individual display and interaction modules within a portal page) were not yet available to ensure compatibility across portal servers. JSR168 would ultimately be adopted to resolve this issue, but this was not available at the time.

After researching alternative approaches with support from OBC's architecture team, Hal, OBC's lead developer, found a potential path to success (shown in Figure 2 (B)). He realized that he could separate the FI admin tools (which required Vendor A's product) from the

end user tools (which were the driver of the scalability and CM tool requirements).

The team found an open-source portal server and a lighter-weight application server combination that greatly reduced the memory required for each FI, resolving the scalability issue. It also stored its configuration information in a way that was readily movable between servers, resolving the migration issue.

This had the potential not only to resolve the issues with requirements, but due to the nature of open source licensing, the cost to OBC could be dramatically reduced. The components that scaled with user populations, the end user portal, could now be very low in cost, while Vendor A's more expensive COTS portal server served only FI administrators, requiring a very small number of CPUs.

The team now faced a dilemma – go forward with Vendor A or switch to the open source option? Steve and Helen laid out the options: Stay with Vendor A, with the known risks and existing relationship, and press to get open issues resolved or move the end user components to an open-source platform. Moving to open source would add complexity to the operations (two different application and portal servers), and introduce an untested set of components that the team would have to integrate. It would also disrupt OBC's already tenuous relationship with Vendor A.

Relationship with Vendor A

Both OBC and Vendor A believed that they made good faith efforts to engage productively. Shawn said that Vendor A was limited in the resources it could apply to the project because OBC had not committed to buy a large volume of Vendor A products. Jerry said that OBC could not in good faith commit to a large purchase without knowing that the products would meet its needs.

One key issue was the support of a standardized interface for portlets. A new standard, called JSR168 (Sun Microsystems, 2003), was in the process of being ratified. OBC was interested in building its code to comply with JSR168 (for future portability), but Vendor A would not commit to a schedule for delivery of this capability since the standard was not finalized.

This gave Vendor A the advantage since much of the code built by OBC was tightly coupled to the Vendor A portal server. OBC would have to engage in a significant re-coding effort if it dropped Vendor A's portal server entirely.

To attempt to resolve this and other issues, in the spring of 2004, Shawn invited key members of the OBC team, including Jerry, Helen, and Hal, to travel at Vendor A's expense to meet with the portal development team. The objective was to share Vendor A's roadmap for future development and for the portal team to hear firsthand OBC's concerns and questions.

Helen hoped that the meeting would provide stronger connections between the two companies. However, she noted that the development roadmap was still not firm enough to give OBC the confidence to go forward with Vendor A's product.

3. CONCLUSIONS

OBC ultimately brought its new web toolset live but only after significant delays and acceptance of limitations to the product. Bringing WebService v1 to market helped get customer feedback for refining the product, but did not meet the expectation of general availability for sales.

The project did not suffer a single large delay, but rather a series of small delays. Helen never felt justified in proposing to "pull the plug" on the project and start with a new architecture – not until the original target date had passed.

What mistakes did Helen and her team make in the course of the project? What did they do well? How did Vendor A contribute to the problems?

OBC still needs to put out a version of WebService that can be sold to a wide range of customers. What path should OBC take now?

4. ACKNOWLEDGEMENTS

The author thanks the OBC management team for their support, and the whole WebService team for their willingness to collaborate.

5. REFERENCES

Gallagher, B., & Brownsword, L. (2001). The Rational Unified Process and the Capability Maturity Model - Integrated Systems/Software Engineeringpp. 169). Available from www.sei.cmu.edu/library/assets/rup.pdf

Kruchten, P. (2003). The Rational Unified Process: An Introduction (3 ed.): Addison-Wesley Professional.

Mietzner, R., Leymann, F., & Papazoglou, M. P. (2008). Defining Composite Configurable SaaS Application Packages Using SCA, Variability Descriptors and Multi-tenancy Patterns. Paper presented at the Internet and Web Applications and Services, International Conference on. Retrieved July 15, 2010, from http://www.computer.org/portal/web/csdl/doi/10.1109/ICIW.2008.68

Sun Microsystems (2003). Introduction to JSR 168—The Java Portlet Specification. Santa Clara, CA: Sun Microsystems.

Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2009). A Break in the Clouds: Towards a Cloud Definition. ACM SIGCOMM Computer Communication Review, 39(1), 50-55.

Wege, C. (2002). Portal Server Technology. IEEE Internet Computing, 6(3), 73-77.

Figures

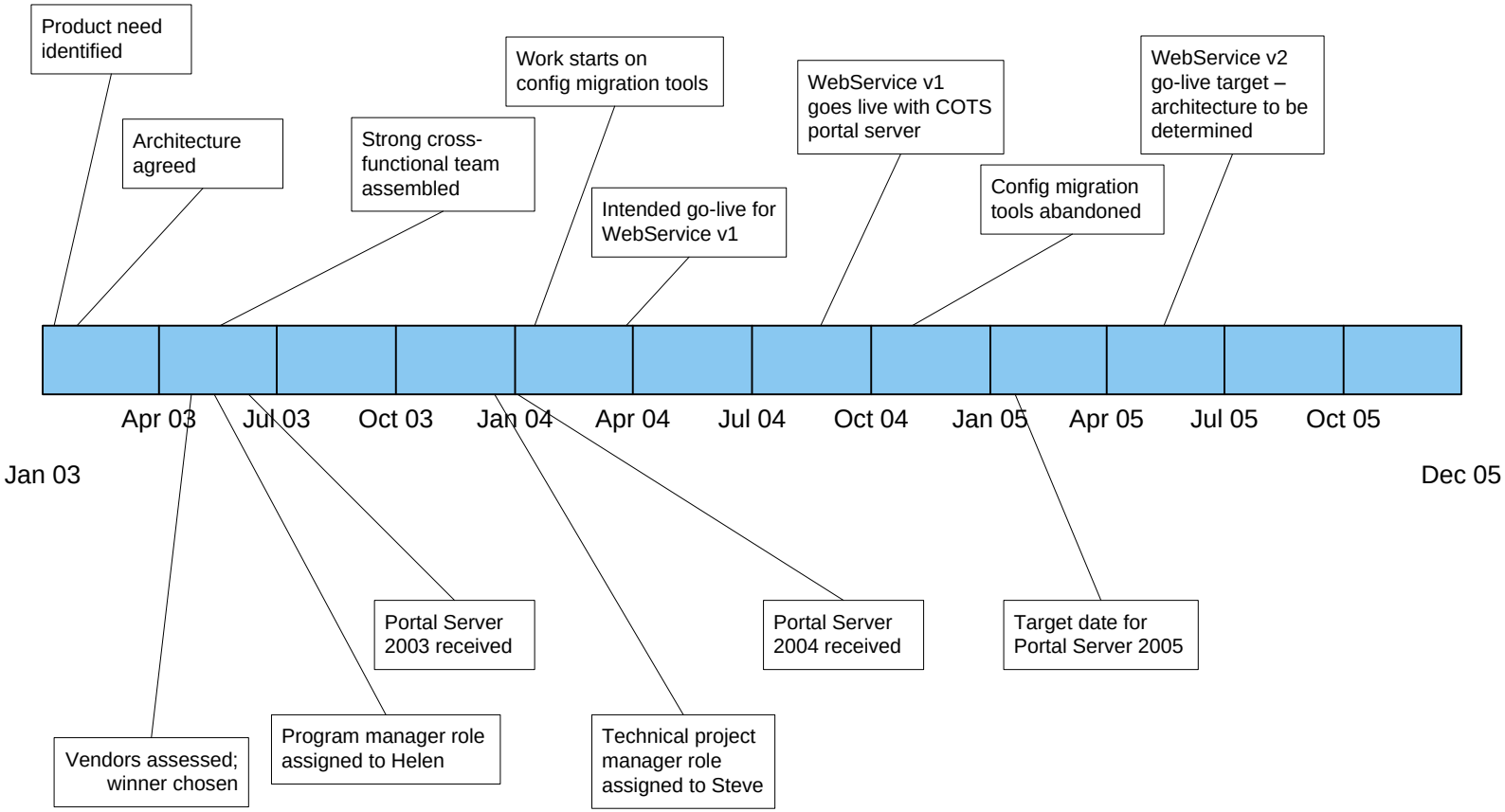

Figure 1 - WebService timeline

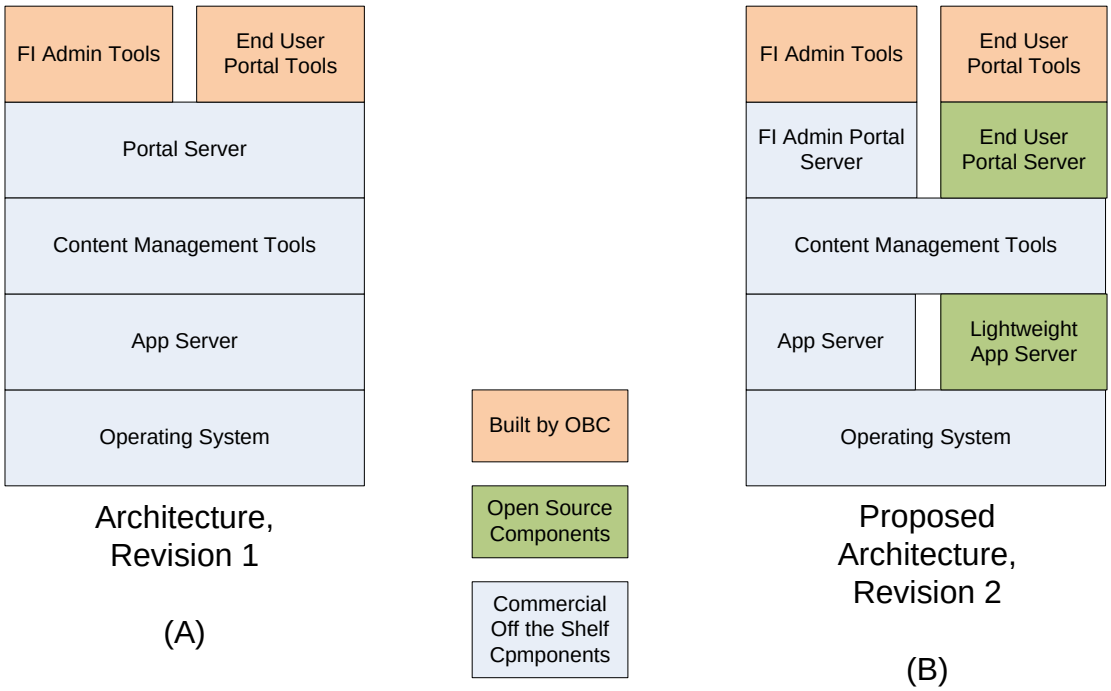

Figure 2 - Architecture of WebService, Revisions 1 (final) and 2 (proposed)