Collaborative Tools Strategy
University of California, Berkeley

Spotlight: Service Blending IT Skills

In the Campus Collaborative Tools Strategy for UC Berkeley, Goal 5 is to "Train our workforce to work with and support new collaborative technologies and architectures, while protecting privacy and keeping data secure." Goal 5b focuses on enriching the campus workforce's skills in blending information technology services, in order to better support the deployment of collaborative tools. This Spotlight document provides more in-depth discussion regarding those recommendations.

The collection of collaborative tools and other IT services used by members of higher education institutions is expected to increasingly consist of a blend of services. More than ever before, the services with which end-users come in contact may be sourced from a diverse set of providers and provision models.

Some examples of providers from which collaborative services can potentially be sourced include:

- Campus units that have developed special in-house expertise.
- IST, as it continues to construct increasingly rich and capable service layers on which campus and departmental service developers can build solutions.
- Consortia of UC schools or other higher education institutions.
- Outside providers.

Furthermore, there are multiple provision models – on both technological and business sides – through which services may be obtained from any of these providers. For instance, services may be run on a server computer under a department's or campus's control, or accessed over the Internet via the Software as a Service (SaaS) model. Those services, in turn, can be licensed in various ways, ranging from flat or tiered rates to per-user or per-transaction fees, and may even be wholly or partly subsidized by advertising displayed to users.

We are already witnessing several indications of the diversity of providers and provision models available, both here and at other higher education institutions. For instance, some universities and colleges have partnered with Microsoft or Google to offer email to some of their populations, and the UC Office of the President has recently signed a contract with Google, which permits UC campuses to offer Google's online application suite to their students. Here at UC Berkeley, initiatives are underway to investigate hosted service providers like Salesforce.com, to allow service units to better track and coordinate their contacts with customers; and to explore whether to run instances of our Event Calendar Network system in the campus Data Center, to be offered to several other UC campuses to power their own departmental and campus event calendars.

So far, the discussion above has focused on instances where multiple providers and provision models are either being considered, or are being used today, to provide monolithic services. However, there is another trend that presents great opportunities – and challenges – to IT providers: the growing availability of services providing some of the fundamental, lower-level building blocks from which systems can be constructed:

- Sourceable infrastructure.

  Many startup companies today 'hit the ground running' by sourcing various infrastructure-layer elements, including storage, compute cycles, server administration, hosted databases, web hosting, authentication

and account management – or offerings that integrate several of these – from service providers in the Internet "cloud." IST also provides similar infrastructural pieces as service offerings, which are used by many campus and departmental application developers.

If, as we expect, campus and departmental IT will over time become more agile and entrepreneurial, we might expect to see a trend – if at times haltingly – in that same direction here. Growing numbers of system developers will likely source these infrastructural components from both IST and cloud-based providers, rather than acquiring, installing, and administering servers, storage, databases, and web servers.

- Composable services.

  Beyond the infrastructure layers, we expect to see a growing number of services accessible via web services architectures, such as REST and SOAP and the WS-* stack. Some services may also, or instead, be provided via messaging that passes over enterprise service buses (ESBs). Many of these services are capable of being composed, together with domain-specific programming code, into systems.

  These services may be provided as standalone, single-purpose services, such as the campus web service that validates General Ledger chart of account strings ("COAs"); consist of discrete functionality of otherwise monolithic applications that have been exposed as services; or consist of frameworks or federated sets of services, such as those currently being designed by the Kuali Student Systems project, a consortia of higher education institutions.

In this emerging paradigm, a campus system – such as a major administrative application – may run on servers, storage, operating systems, and web server and application container software provided and maintained on IST's Application Farm or Webfarm platform; have an Oracle database sourced from another IST unit; use second-tier storage purchased from Amazon's S3 service; and have code that interacts with multiple web services. The latter may be running as part of the Kuali Student Systems framework, exposed from parts of other campus administrative applications; or even offered by Microsoft, Google, or another "cloud" provider.

In this environment, it will be crucial to continue to build skills among IT staff in evaluating, sourcing, and managing relationships with multiple providers. These skills include building a deep understanding of when it is appropriate to source services from different types of providers, and in managing the end-to-end risks involved, where some of these services may reside outside of the campus Data Center, or alternately, outside of a department's direct control.

While all IT services have a finite lifetime, some of the compelling services that one can now source from outside one's IT organization may reasonably be expected to turn out to be mutable or short-lived. Avoiding these risks where possible, and conducting careful planning to mitigate the impacts of inevitable change, will be a critically-needed skill. Some examples of mitigation techniques include:

- Choosing service providers that use standards-based network transport protocols, data access protocols, and data formats. This helps avoid provider 'lock-in', and makes it more feasible to migrate to new providers when necessary.
- Having access to source code, whether via open source licenses or, less desirably, via contractual escrow mechanisms. This can make it possible for the campus to run service instances in-house or to migrate them rapidly to new hosting services, if required.
- Regularly taking time to review the offerings of alternative providers for particular services, and developing contingency plans for switching to a different provider.
- Developing or building upon existing programming frameworks that provide "abstraction layers" – layers of code that generalize interactions with specific providers and service – that make it easier to switch to different back-end service providers.

As may be evident from this discussion, contracting with outside service providers will not release the campus or department from responsibility for those services. Retaining, and in some cases, deepening, in-house technical and

managerial expertise within the campus IT community will be increasingly essential to effective sourcing; to managing technical integration and working relationships with a mix of service providers; to managing the end-to-end risks involved when combining services; and to blending multiple back-end services into coherent, unified services as experienced by their customers.

As just one representative example, having deep expertise in storage technologies available within the campus IT community will be essential to analyze when, if ever, it might be appropriate to source some amount of storage from a "cloud" provider like Amazon Web Services, and if so, how best to integrate that storage into a campus-provided system and to monitor its performance and availability over time.