



The O.K.I. Repository OSID and Sakai

Leveraging standards to provide broad markets for content

Summary

The Open Knowledge Initiative (O.K.I.) Repository OSID includes interfaces for integration among applications (consumers) and repository content (providers). Applications use the OSID to gain access to content in a manner that hides the technical detail by which that content is provided. This allows the application to manage integration with a wide range of content providers without including the complexity inherent in supporting heterogeneous means of communication and data exchange. The application is also insulated from technology changes made by providers which leads to a longer useful lifetime for an application and thereby a greater return on investment.

The Sakai project is building and deploying a collaboration and learning environment for higher education. Sakai is a community effort and the software has been developed under a community source model similar to open source.

Commercial vendors of learning object repositories, content management systems, learning management systems, and their variations have already invested in integration with popular commercial systems such as WebCT / BlackBoard. Now Sakai is an integration target for commercial vendors, many of whom have also been working with O.K.I.

Sakai will include a repository service that federates among Repository OSID implementations. The service is an implementation of the O.K.I. OSID's RepositoryManager – the interface that provides access to repositories and handles search requests.

With Sakai's incorporation of the Repository OSID as its repository service, there are some immediate benefits:

- Sakai tools can work with any Repository OSID implementation to access data sources.
- Sakai gains immediate access to data sources already provided via a Repository OSID implementation.
- Vendors that have already developed Repository OSID support need to make no further investment to have their data sources available in Sakai.
- Since the Repository OSID is maintained through the IMS Global Learning Consortium specification maintenance process, the community increases the leverage of standards.

Background

The Open Knowledge Initiative (O.K.I.TM) is a MIT-led, community effort to improve interoperability among applications and the enterprise system services on which they depend. The initial focus has been on higher education and eLearning, but the initiative's service-oriented architecture is applicable outside these domains. O.K.I. provides Open Services Interface Definitions (OSIDs): contracts between service consumers and providers. The OSIDs are well-defined integration boundaries that leave flexibility in the hands of developers. The OSIDs are

neutral with regard to programming language¹ and implementation detail. There are OSIDs for common services such as authentication, authorization, hierarchy, scheduling, and workflow and eLearning services such as repository, assessment, grading, and course management. The OSID currently witnessing the greatest level of interest and adoption is the Repository OSID. Repository defines objects such as a

¹ The programming language-neutral form of an OSID is called an XOSID. There are bindings of the XOSIDs for Java, Objective-C, and PHP. Other languages bindings are in development.

Repository Manager, Repositories, Assets, and their metadata as well as methods for managing object lifecycle, data maintenance, and searching.

For application developers, O.K.I. offers a standards-based repository content integration strategy. Supporting a technology-neutral OSID saves the developer from the complexity of understanding and tracking many disparate technologies. By implementing OSID consumer support, each developer gains immediate access to any content for which OSID provider implementations are available, subject to the content vendor's licensing requirements. There is local and remote content, content in databases accessible through a driver, web services, proprietary communications, etc. There are also numerous data formats, authentication schemes, and other detail. In addition, every time a content provider makes a technology change, applications that are bound to that means of integration must change as well. All this makes integration hard to deliver and expensive and fragile to maintain.

The Sakai Project began when several organizations: the University of Michigan, Indiana University, MIT, Stanford, the Open Knowledge Initiative (O.K.I.), and the uPortal consortium joined forces in a combined, open source effort. The goal was to build upon their respective in-house course management systems and related work. The project also received funding from the Mellon Foundation. The community of Sakai has grown to include partners and commercial affiliates, all of whom share an interest in providing services which provide basic infrastructure and tools which consume services to perform teaching or learning management functions.

Early on, Indiana University built a Sakai tool that searched several data sources. Results of the search could be pasted as links into other Sakai tools. This effort was named "Twin Peaks". In parallel, O.K.I. had been working with a variety of data source providers to help them build implementations of the Repository Open Service Interface Definition

(OSID). A family of applications that could work with any Repository OSID implementation was emerging, as was a set of implementations that offered access to content. Groups that were working with O.K.I. also wanted a Sakai integration strategy.

These separate threads of activity joined together under "Twin Peaks II" in May 2005. This effort was made by MIT. Starting from the Twin Peaks code, the effort added searching any Repository OSID implementation. The effort also added simple multi-source searching. Sakai 2.1 intends to offer the Repository OSID as a repository service. Rather than being an add-on, as Twin Peaks and Twin Peaks II were, the OSID will be offered as a service directly in Sakai.

In a single OSID implementation scenario, say for SRW, the repository service works with one plug-in. This plug-in handles all the detail of sending a search via SRW and converting the responses to OSID Assets. The repository service consumes the Repository OSID (interfaces) that the SRW plug-in implements. This same OSID is implemented by all plug-ins, so there only needs to be a single piece of code in the repository service to handle any plug-in.

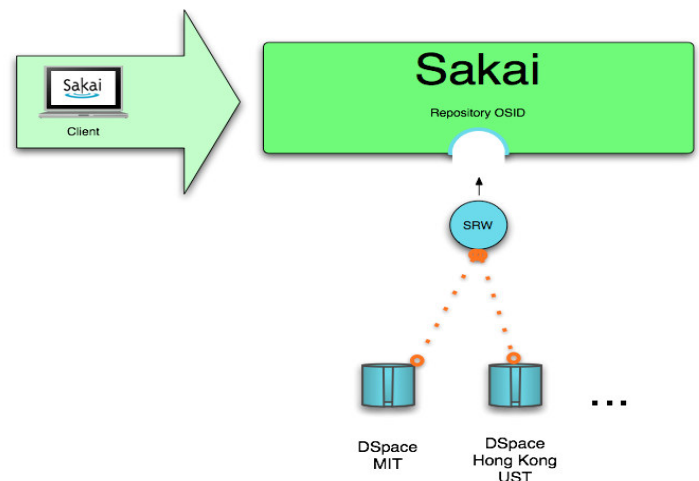


Figure 1: Sakai Accessing DSpace Content via OSID Plug-in atop SRW

The plug-in approach becomes more valuable with more than just a single plug-in. There is a “network” benefit that grows as adoption grows. OSIDs offer a further advantage because they are imple-

mentation-neutral technology. That technology can be SRW or something else and the consuming tool is no wiser and, more importantly, needs no additional plug-in-specific support code.

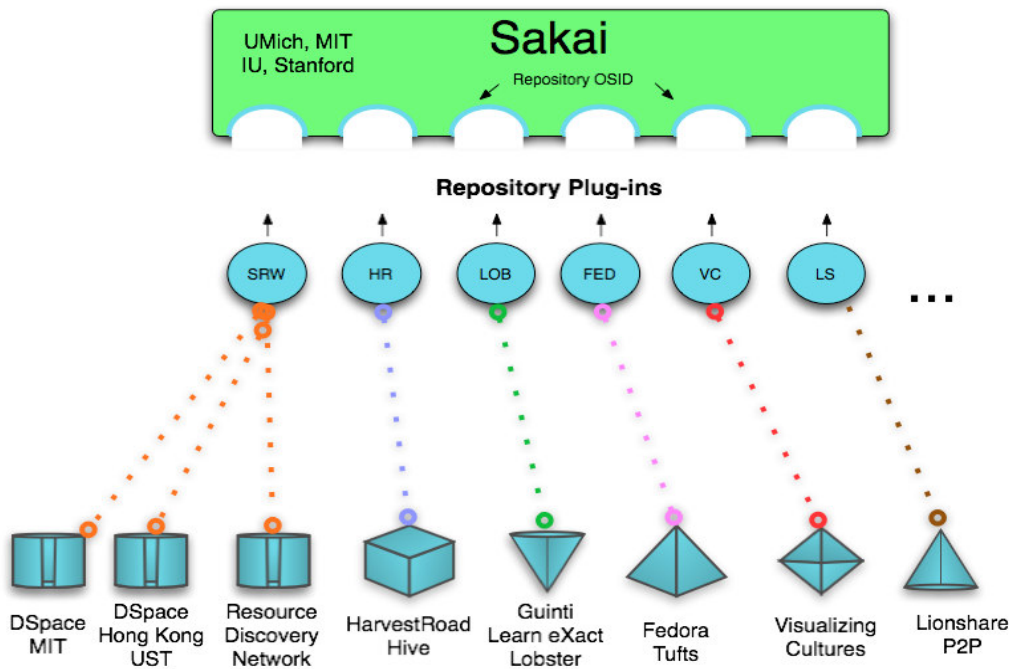


Figure 2: Sakai Accessing Varied Content via OSID Plug-ins

Keeping track of which plug-ins are available is an administrative task in its own right. A simple registration mechanism allows a Sakai installation administrator to list OSID implementations that are available. The registry is exposed via an interface. Sakai includes an implementation of this interface. This level of indirection allows the specifics of the registry mechanism to change without affecting consumers of the registry interface.

A special federating implementation of the Repository OSID works with this registry and presents a single interface to a Sakai tool. The Sakai tool calls the repository service. This service is an implementation of the O.K.I. Repository OSID. This implementation calls the registry interface implementation to find out what plug-ins are available. The repository service loads the plug-ins and federates requests and responses among them. The Sakai tool sees the union of all plug-ins as a single plug-in

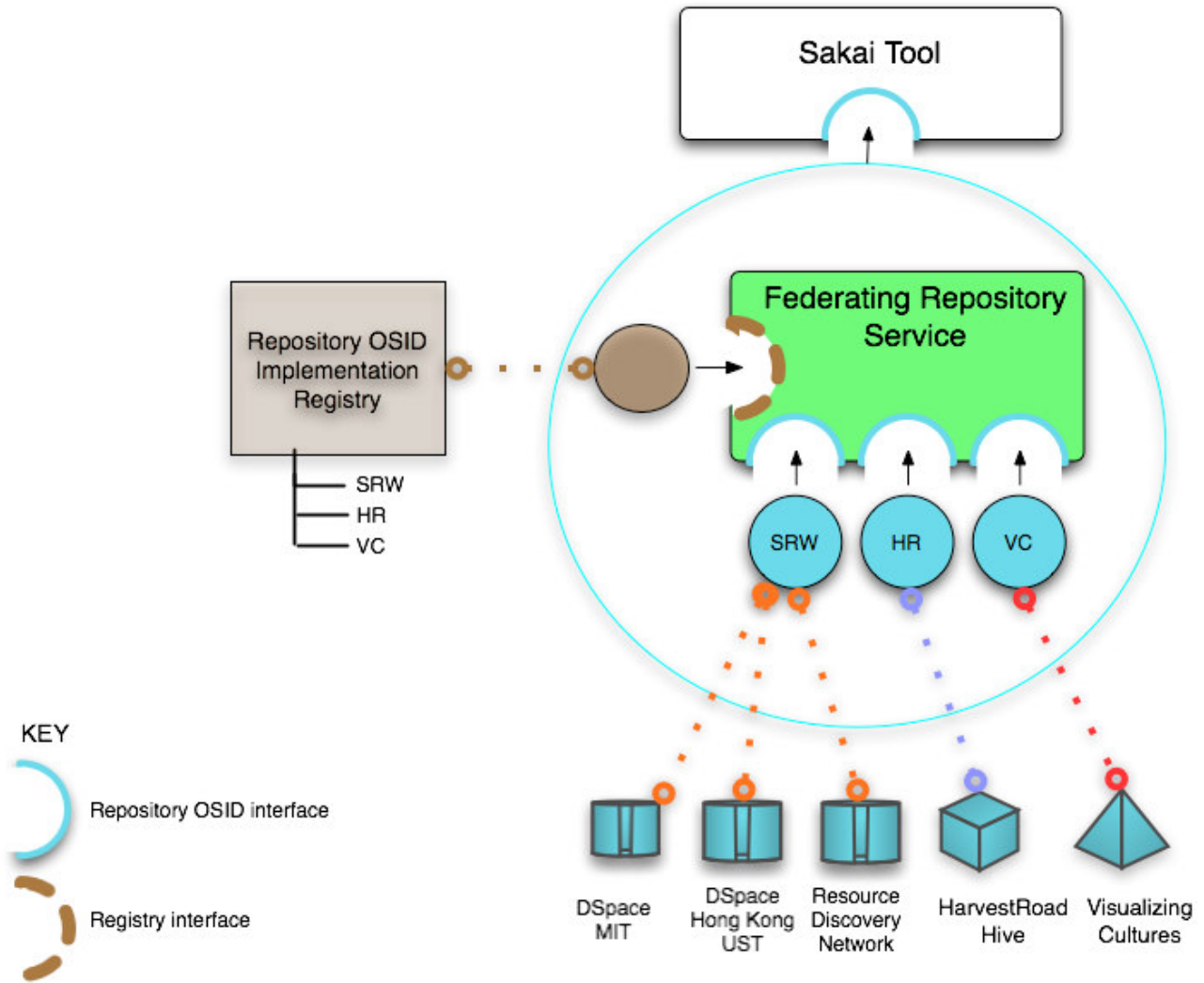


Figure 3: The Sakai Repository Service Architecture

In closing, it is important to remember that Sakai and its repository service is only one of a growing number of applications which use the O.K.I. Architecture.