

# Sakai

## Respondent

Ray Davis, University of California, Berkeley

## Goal/Problem Space

Sakai is a pluggable open source Learning Management System / Collaborative Learning Environment framework and suite, particularly targeting centralized installations in large universities.

## Features

- Active widely-distributed developer community.
- 100+ contributed applications and services, including standard collaborative tools such as forums, wikis, and file storage, and education-targeted tools such as assessment engines and gradebooks.
- A single installation can support thousands of independent workspaces (or "sites"), each with their own application instances and configurations, and some with a thousand or more members.

## Technology Stack

Sakai's widely distributed development and component architecture promote a variety of stacks.

Sakai 2 kernel services are written in Java and based on Spring's component and transaction capabilities. Any list of key application dependencies would include Hibernate, Velocity, and JSF, but many applications and components use alternatives.

The next major generation of Sakai is under active development and represents a significant break with the past. The new kernel runs on Apache Sling, which is in turn based on Apache Jackrabbit (a Java Content Repository), OSGi, and scripting languages such as JavaScript and Groovy.

## Identity Services

"Sakai" refers to both a framework and to the software suite built on that framework. Here, I'll focus on Sakai 2's core services, including typical integrations with external systems. Consumers of "produced" services would then be the deployment's plug-in components and applications. I'll take "Broker/Convey" to mean "convey to non-Sakai software."

Managed Information	Consume?	Produce?	Broker /Convey?
Privileges	X (from plug-ins)	X	
Roles	X (from optional integrations)	X	
Groups	X (from optional integrations)	X	
Attributes	X (from optional integrations)	X	
Identification	X (from optional integrations)	X	
Defined Interfaces	Consume?	Produce?	Broker /Convey?
Authentication	X	X	
Attributes	X	X	
Permissions	X	X	
Provisioning	X	X	
Authorization	X	X	
Subjects	X	X	
Other	Consume?	Produce?	Broker /Convey?
Course Management Data	X		

## Standards and Interfaces

Sakai 2 is designed to allow integration with external authentication services and user and group provision. However, it lacks out-of-the-box support to set up and configure such integrations, which forces most large universities into more-or-less customized builds. (CAS and LDAP integrations may require the least actual code editing, thanks to stable "sample" code.)

EduPerson support is included, but not as part of the kernel.

Integration with course management data (AKA "SIS") is handled via a Sakai-specific interface designed when existing standards were found lacking. We hope that the forthcoming IMS LIS standard will meet its motivating use cases and eventually serve as a replacement.

A stated goal of Sakai 3 is to rely less on home-grown services and more on well-established open source libraries and standards. What this means in terms of integration with external authentication, authorization, and user provisioning sources is very much a work in progress.

## Issues and Challenges

On the implementation side, our federated authentication and user provision can certainly be improved, but the basic concepts seem well understood and manageable. Our greatest challenges lie in federated authorization.

A single individual may play an instructor in a classroom, a student in a seminar, a leader in a discussion group, a grader in a lab section, and an editor on a research project. Some of these roles and contexts (but not all) are typically determined outside Sakai. These context-dependent roles imply different application roles and permissions. The set of possible roles and their implied permissions vary from institution to institution, and even within an institution. The applications (and their own ideas of user functions) are developed independently and deployed over time.

In short, a pluggable open source CLE/LMS like Sakai must cleanly and efficiently integrate end-user-managed groups, non-static installation-defined roles, non-static application-determined function sets, and multiple externally-managed role and group assignments.

## More Information

- [Sakai](#)
- [Sakai 3 Kernel](#)
- [Federated authorization requirements](#)
- [Sakai 2 integration issues](#)