

Use Case 1 - Register a Deployed Web Service

Purpose

The purpose of this use case is to describe the process of registering a deployed web service and to communicate why someone would want to do that. A deployed web service is a service that could actually be accessed by authorized users. A deployed service is different from a service that is available for deployment (like a web service that is an add-on feature of a vended student information system) or a web service that is specified but not implemented (like an IMS or PESC web service specification). The impetus behind this use case is the need for academic institutions to publish information about their web services, so that they may be accessed by authorized users such as other academic institutions, government agencies, clearing houses and other application service providers for individual end users, or individual end users like students, faculty, applicants, and staff.

The following use case could apply to just about any system at an academic institution. In order to keep the language simple and clear, we have selected one application and service to illustrate this use case. Any number of actors, applications, and specific web services could be substituted for this specific use case. We may decide to list more of these explicitly, if that is necessary to make this point.

Actors

[Application Administrators](#), [System Administrators](#)

Applications

[Student Information System](#)

High-level Story (Abstract)

A college or university deploys a web service such as the [Academic History Web Service](#) and wishes to publish it so it may be invoked by other colleges, universities, or service providers to facilitate course transfer articulation processes for its students.

Detailed Story (with illustrations and schematics as needed)

The [application administrator](#) or [system administrator](#) deploy and test the web service. Typically the web service is deployed in multiple environments for development, test, and production purposes. For example, the college or university must allow its trading partners some access to test integrations. Additionally, the academic institution and its trading partners will have test clients that use EdUnify infrastructure to locate its own services. All deployments of the web service in every environment that are available to applications using EdUnify are registered by the administrators in the directory.

The application or system administrator will deploys the web services at their institution and will then register all of these deployments in the EdUnify registry. The precise process for doing this will vary depending on the architecture of the EdUnify infrastructure. For example, if EdUnify is implemented as a distributed, federated directory, and the institution already has web service registry infrastructure that is federated with EdUnify the administrators may register these services in their existing directory infrastructure that is federated with EdUnify. If EdUnify is implemented in a more centralized model, then the administrators would use EdUnify web services and potentially EdUnify web applications to register these services.

For the sake of progress, let us assume that regardless of the architecture and topology of EdUnify there will be one set of EdUnify web services and one EdUnify web application for registering web services with EdUnify. Whether EdUnify is distributed, federated or distributed, centrally administered or any other variation need not have bearing on the fact that for ease of use, training, and communication reasons among the human users of the infrastructure, one common set of application programming interfaces and user interfaces will be provided and maintained by EdUnify.

The deploying administrators will use the EdUnify web services or web application to provide metadata about the web service to the registry. The type of metadata required will be determined by the web service search use cases, but for the present we assume this metadata will be in a standard format and include some annotation with agreed upon ontologies to enable semantic features of web service searches. Therefore, the EdUnify web services and web application must support the following high-level flow:

1. Authenticate the user
2. Allow the user to create, update, and delete entries for which the user is an authorized administrator
3. Validate the metadata provided by the user
4. Submit the user's changes to the underlying EdUnify directory infrastructure

To Do

These basic functions and much of the detail of their implementation are largely independent of the underlying technical implementation and architecture of the EdUnify registry. So we can work on more detailed requirements for the EdUnify web services and EdUnify web application based on the use case and the information we have at present. As we have more detailed about the preferred architecture and topology of the EdUnify registry, we should review and update this use case, adding more specifics about how the applications and system administrators interact with the EdUnify web services and web application.