Sakaibrary Terminology Glossary

**Asset** or **Citation Asset.** A *Reference* as returned by an OSID *AssetIterator*, whose methods, parts and record structures are defined in the Repository OSID Implementation.

*AssetIterator.* What the *Repository Object* returns to the consumer; calling *nextAsset()* on it returns a Citation Asset.

**Citation** or **Citation Object.** A *Reference* as it appears internally to Sakai – what Jim’s code transforms a Citation Asset into.

**Citation Collection.** A collection of *citation objects* inside Sakai; a bunch of java objects that represent citations. Not user-visible; see Citation List.

**Citation Editor.** A user-visible form in Sakai that allows a person to change data in a Citation Object.

**Citation List.** A user-visible document in Sakai that shows the renderings of a Citation Collection (in APA or MLA or whatever), along with a description of the list.

**Citation Service.** The code inside of Sakai that is responsible for pulling Citation Objects out of an OSID, manipulating them, editing them, and storing them somewhere.

**Consumer** or **Client.** The code that is going to be sucking data (Assets) out of an OSID. In our case, Jim’s Citation Service within Sakai.

**OSID** or **OSID Implementation.** An implementation of the OKI Repository OSID (the OSID Specification), specific to a particular backend (e.g., Metalib). More specifically, one might say *The Metlib O Sid implementation* or even *Gaurav’s Metalib O Sid implementation.*

**OSID Specification or O Sid Interface.** The OKI Repository O SID specification, an agreement about how objects that call themselves “OSID Implementations” will behave. This is a document that defines how a (Java) object is dealt with by other code, although parts of the specification are “mushy,” not defining the format of data and potentially allow all sorts of data to be passed back and forth. See Out of Band Agreement.

**Out of Band Agreement.** A document describing of how a particular implementation is using underspecified elements of the OSID interface. These include the format a search string should take, how to specify what to search, how to know when an asynchronous search is waiting for more data, etc.

**Produc er.** The code that produces Assets – in this instance, an OSID talking to a metasearch engine, which is *producing Asset Citations* to be consumed.

**Reference.** The abstract name for a set of title, author, etc. values that uniquely identify a citable object.

**Repository** or **Repository Object.** The Repository object from a particular OSID.

**Sakaibrary.** The overall name of the project.

**Synchronous and Asynchronous searching.** Synchronous searches make you wait until we have all the search results – perhaps up to a minute or two. Asynchronous searches allow the consumer to pass in data as it becomes available, but opens up various cans of worms regarding how to know when/how long to wait for more data, how many results are available, etc. We’re shooting for asynchronous searching.

**Triple-Store.** A specific method of storing sets key-value pairs (like those that make up a reference). Only Jim needs to worry about how this works; the rest of us can just think of it abstractly as *storage.*