

Purpose:

- Extend Partner Portal customizability
- Enable search capability to be offered from a server other than the one hosting the search indexes, an option other than an iFrame

Mechanisms:

- Providing a **RESTful**¹ resource(API) that conforms to the design principles of Representational State Transfer (REST) would offer the capability to supply search forms and results to any requestor regardless of location.
- The current **iFrame** capability would remain with added customizability being offered to users who wish to use it. This would use a page generating agent that would access and process search forms and results from the RESTful API.
- Implementing a database that would provide customizability and filters for search displays and results.

Solutions Provided:

- Customizable search displays that would do away with coding intervention
- Capability to create truly unique search portals
- Ability to extend a repository search to other servers, direct service access.
- Open search access to the repository
- Provide collection hierarchies in a Partner Portal for those who choose it.
- Create pre-programmed searches for use in a browse link
- Allowing users to create more than one Partner Portal
- Compartmentalize the XML search engine and query protocol allowing for flexibility and future upgradeability
- Capability to provide a OpenURL² resolver by using the query translator
- Offer a true service providing future integration with other web applications, possibly courseware type applications.

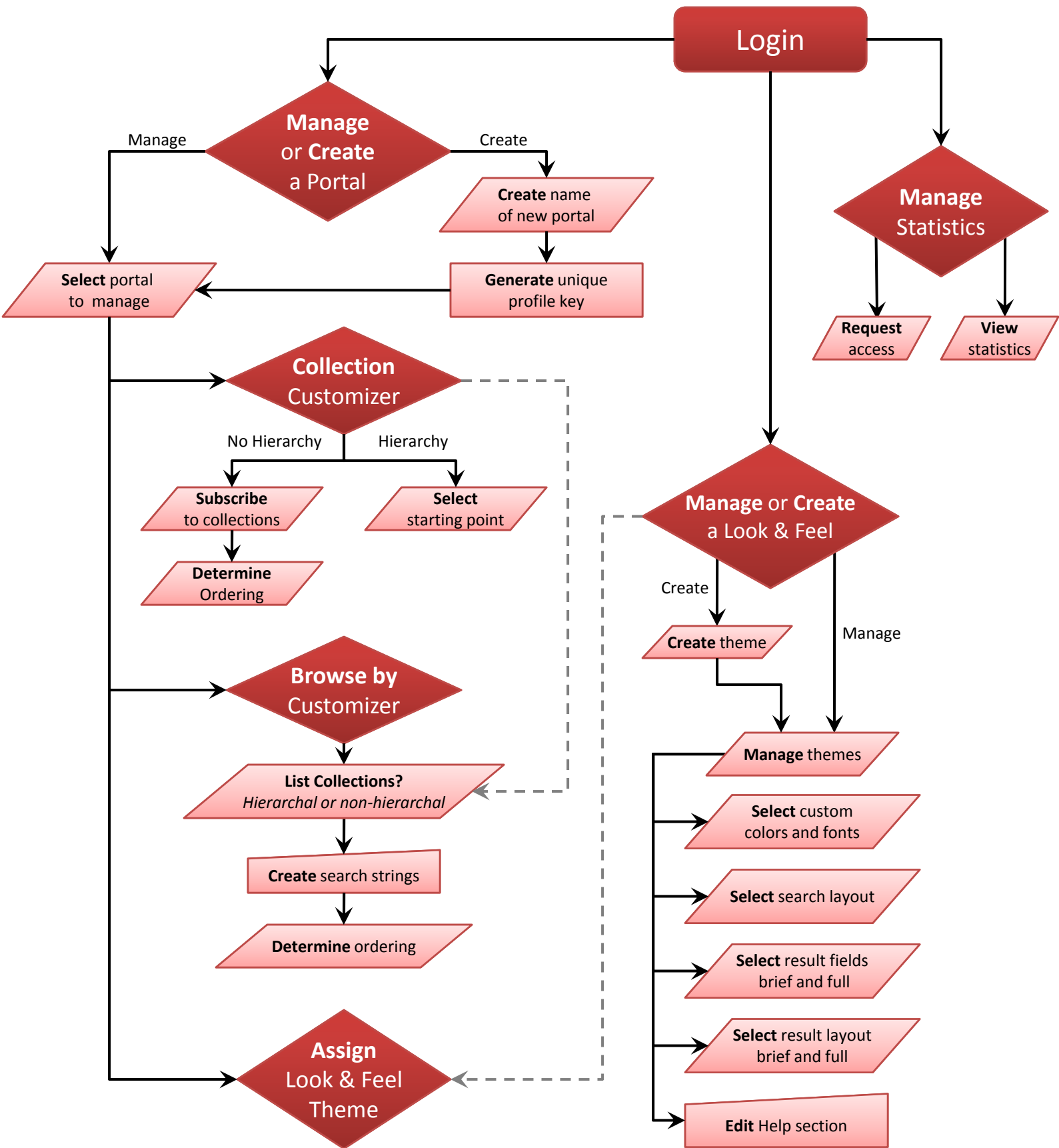
Frameworks:

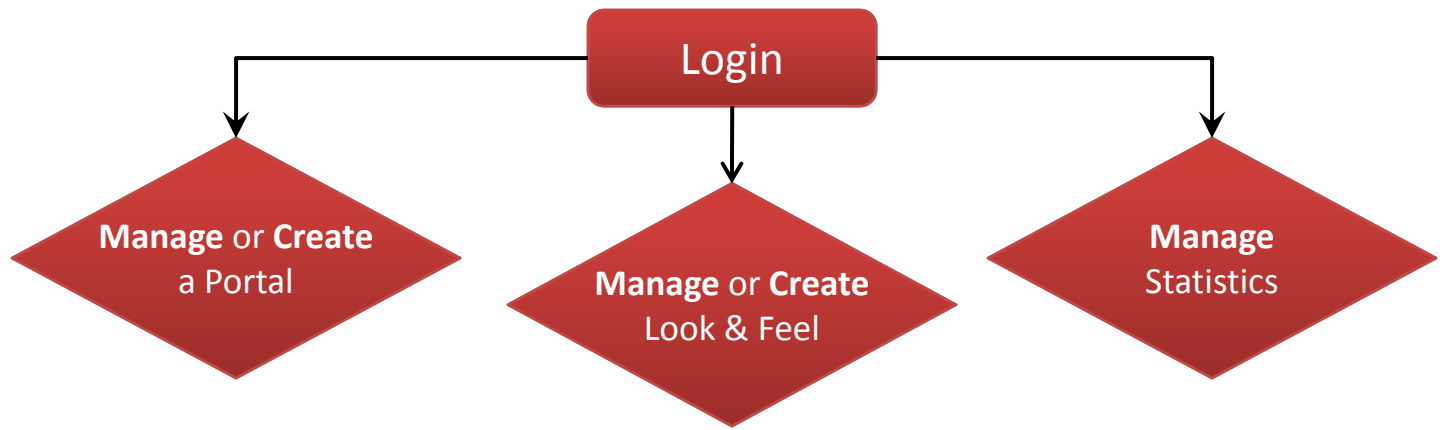
- The **Search Service Configuration Framework** provides the facility to create, store and manage customizable search and display criteria.
- The **Search Service Framework** provides the mechanism to process search requests and return results along with any custom display criteria.

¹ Representational State Transfer - <http://en.wikipedia.org/wiki/REST>

² OpenURL- <http://en.wikipedia.org/wiki/Openurl>

Overview: Search Service Configuration Framework





Login

Definition:

Facility that will allow for the management of search portals by their owners, also referred to as *logon*.

Mechanisms:

- Shibboleth
- LDAP
- Registration form for users without Shibboleth or LDAP access. Also allows for “Departmental” accounts.

User Levels:

- *Administrator* : Access to common user management, default themes, alerts to portal owners
- *Common User*: Access to creating portals with custom configurations

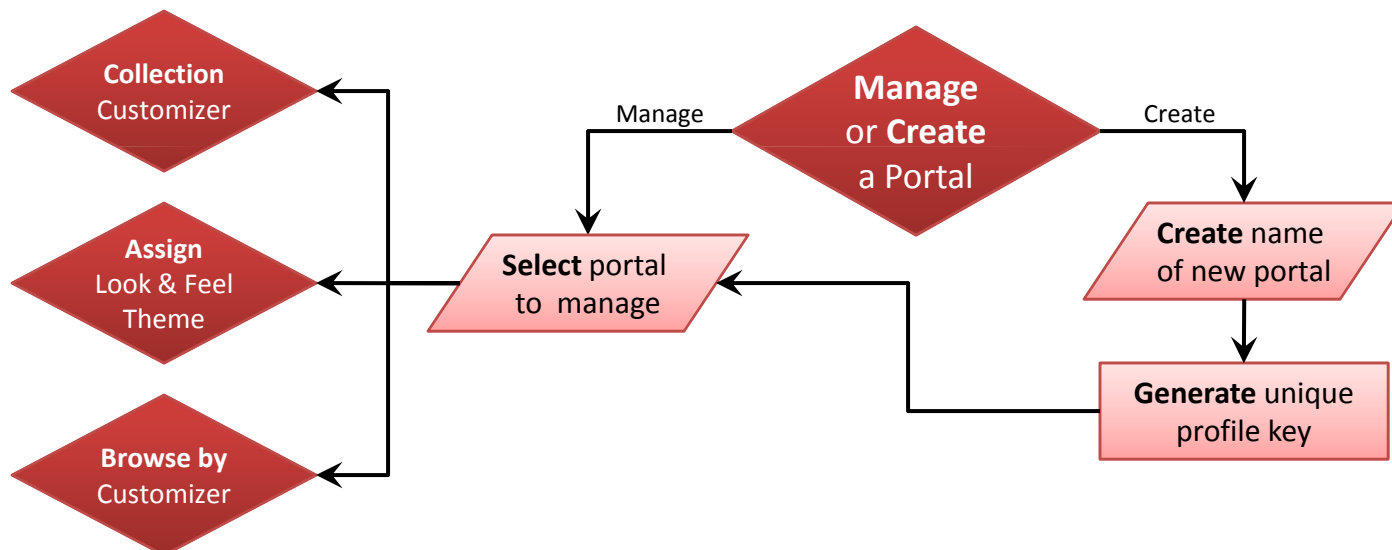
Questions:

- Will management need to be made available to multiple users for one portal configuration, i.e. surrogates?

ANSWER – Management will be extended to allow multiple users control of a single portal. There will be a “portal creator” role who will manage all access to the management of the portal by the other “portal managers.”

- Do we wish to allow people to leave electronic contact information so a messages can be sent to them regarding downtime and outages?

ANSWER – This feature will be made available with an opt-in option for the users to decide what they wish to be contacted about. There will be an opt-in for outages and downtime and an opt-in for marketing.



Portal

Definition:

A portal is a configuration consisting of collections, browsing and Look & Feel criteria supplied by the portal owner, also referred to as a *Partner Portal*.

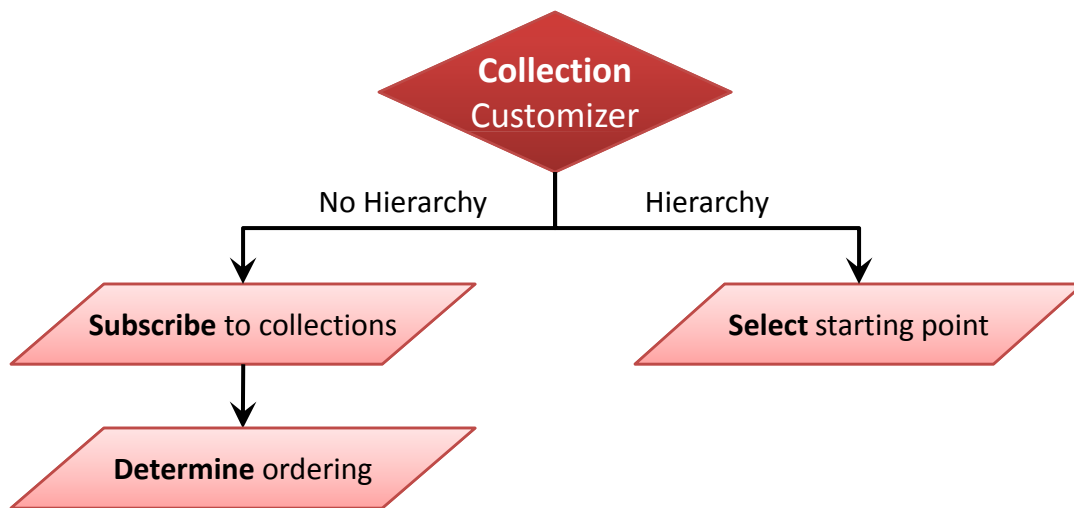
Operations:

- *Select portal to manage*: Management of an existing portal.
- *Create name of new portal*: Creation of a new portal, thus allowing one user to have multiple portals. This will require a user supplied name that will be used internally for identification by the portal owner.
- *Generate unique profile key*: A unique portal key will be generated that will be used by the system to identify which portal is being accessed.

Questions:

- What naming convention should be used for the portal keys?

ANSWER – A naming convention will be devised for the automatic generation of portal keys.



Collection

Definition:

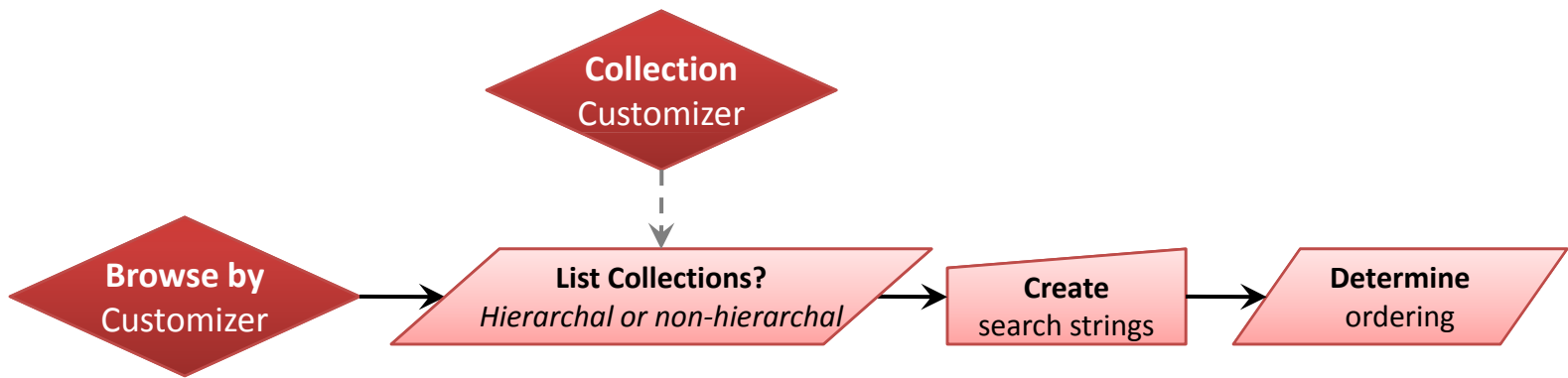
A collection is defined as being in the repository by having a resulting entry in the collections MySQL table. It is available for subscription regardless of active status or the number of objects its search index holds. These criteria will be enforced when displaying making the collection available for searching.

Operations:

- *Subscribe to collections*: Collections chosen through subscription will result in a non-hierarchical view to the end user. The collections will be subscribed through a pick list and can be unsubscribed as well.
- *Determine ordering*: Determine ordering of how the collections in the non-hierarchical view, defaults to alphabetical A-Z.
- *Select starting point*: Collection chosen to have a hierarchical view will be selected through a pull down as only one collection can be selected in this mode.

Questions:

- For hierarchical choices should we:
 - Require the depth of the hierarchical tree be defined by the portal owner?
ANSWER – We will provide a pull down with values to choose (1-10) from, defaulting to the value 3 if no user intervention occurs. The default value 3 and the range of choices (1-10) will be configurable by the Portal Administrator.
 - Have a default collection depth?
ANSWER – Initially the value will be 3, however the user can change this value if they wish with the maximum number provided.
 - Dynamically find the maximum depth during each request and use that?
ANSWER – We will not, there is too much overhead involved in adding this feature at this time.



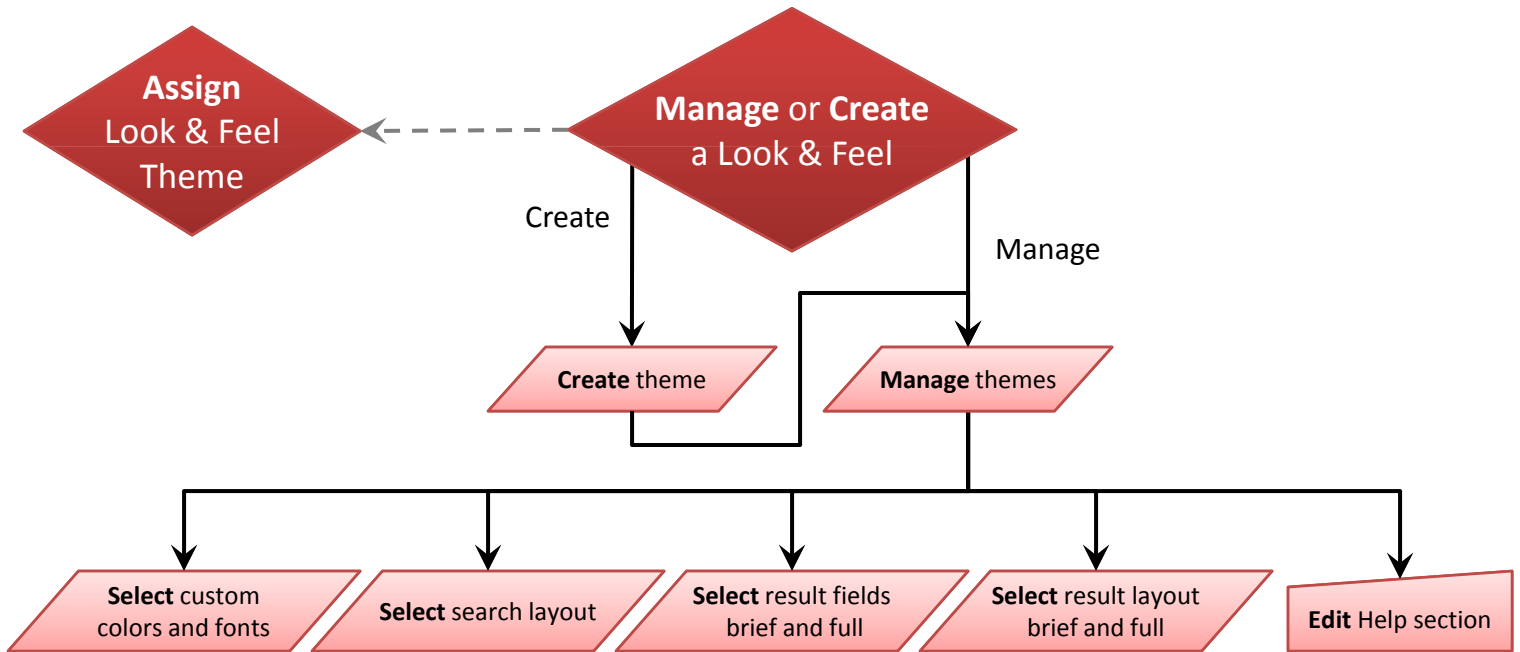
Browse by

Definition:

Browsing is accomplished through a series of links provided to the user in the Browse tab in the search display. These links are constructed from search criteria defined by the portal owner.

Operations:

- *List Collections?:* Provides a block list of bulleted links to browse the contents of collections chosen for that portal in the *Collection Customizer*. In the hierarchal view the list of links will mimic collection hierarchy using indents. In non-hierarchal view the links will be a list not indented. This would be a Yes/No option.
- *Create search strings:* Portal owners will be able to pre-program search results into link that will be made available to end-users. This will require not only search criteria but also a user friendly label to display the link to the search string with in this block.
- *Determine ordering:* Ordering of browse by link blocks, collection and search strings, links would be made available to the portal owner. Also the ability to order links inside each block would be available. Default ordering in each block would be alphabetical, A-Z.



Look & Feel

Definition:

The visual representation of the search interface and results to the end user defined by themes. A default Look & Feel theme will be available and this section is considered optional and for advanced portal owners.

Operations:

- *Create theme*: Creation of a new theme for Look & Feel.
- *Manage themes*: Management of existing Look & Feel themes.
- *Select custom colors and fonts*: Portal owners would be given the opportunity to choose the colors of the pages, information boxes etc. along with borders, fonts, font colors, link styles etc.
- *Select search layout*: Selection from different search layouts, advanced, quick etc would be made here. Also the capability to customize search page pull down menus and presence.
- *Select result fields brief and full*: The choice of which fields to display and what user friendly labels to be used will be made here for both brief and full result displays.
- *Select result layout brief and full*: The layout of brief and full result pages would be configured here. This includes what order to display the fields in.
- *Edit Help Section*: Portal owners will have the opportunity to edit the Help tab of the search display.

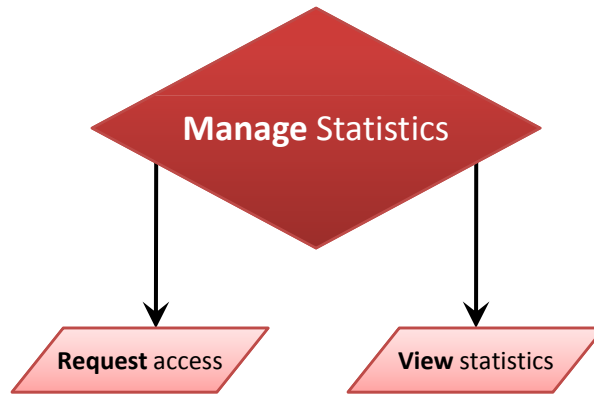
Questions:

- Can all of this customization be made available to portal owners?

ANSWER – This is to be determined during the development cycle.

- Should all of this customization be made available to portal owners?

ANSWER – We will start with a small set of customization options and roll out additional customization options through future software releases after gauging response and reaction from the previous set of customization options.



Statistics

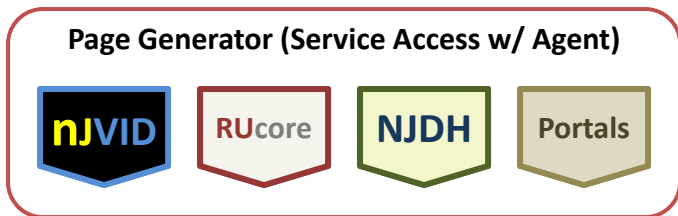
Definition:

Statistical information pertaining to searches and management against a collection. The user will be required to request access to collection search statistics. This access will only be granted to collection owners. By default no access will be granted and only requesting access will be available.

Operations:

- *Request access*: User will request access to viewing collection statistics. User will select collection(s) they wish to have access to from a list of available collections. Once access has been granted the collection will be available in the *View Statistics* section. Access will be controlled by an administrator who will determine if the request is coming from the actual collection owner. This feature will be performed through emailed forms exclusively.
- *View Statistics*: A bulleted list of hyperlinks correlating to the collections that the user has been granted access to. The hyperlinks will direct the user to the Statistics feature of the repository, similar to how the Faculty Deposit application does this.

Overview: Search Service Framework

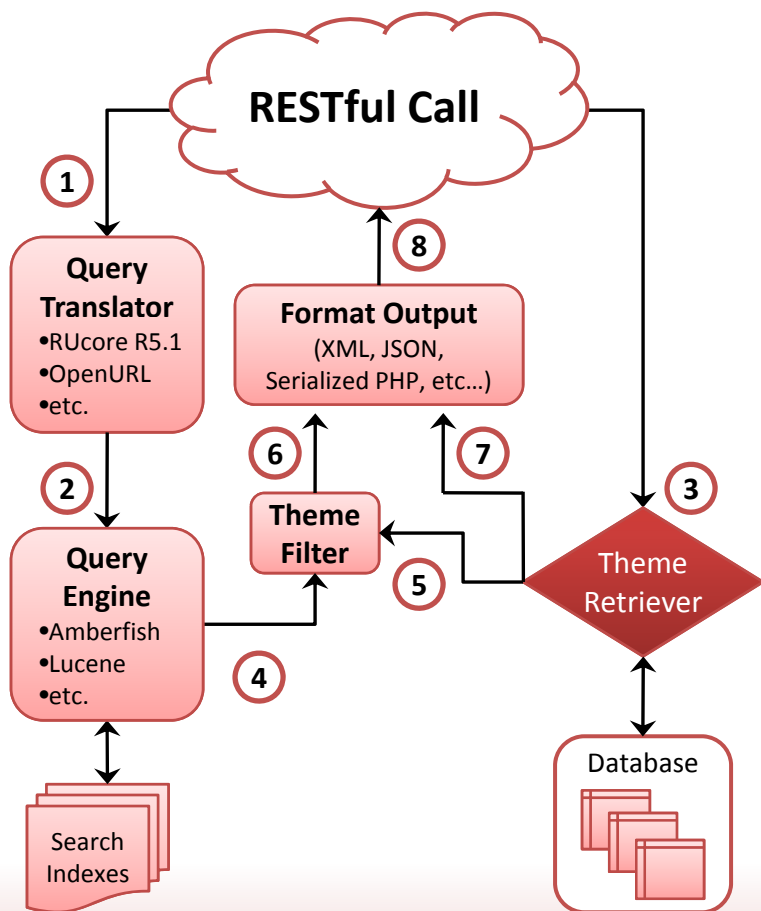


Direct Service Access

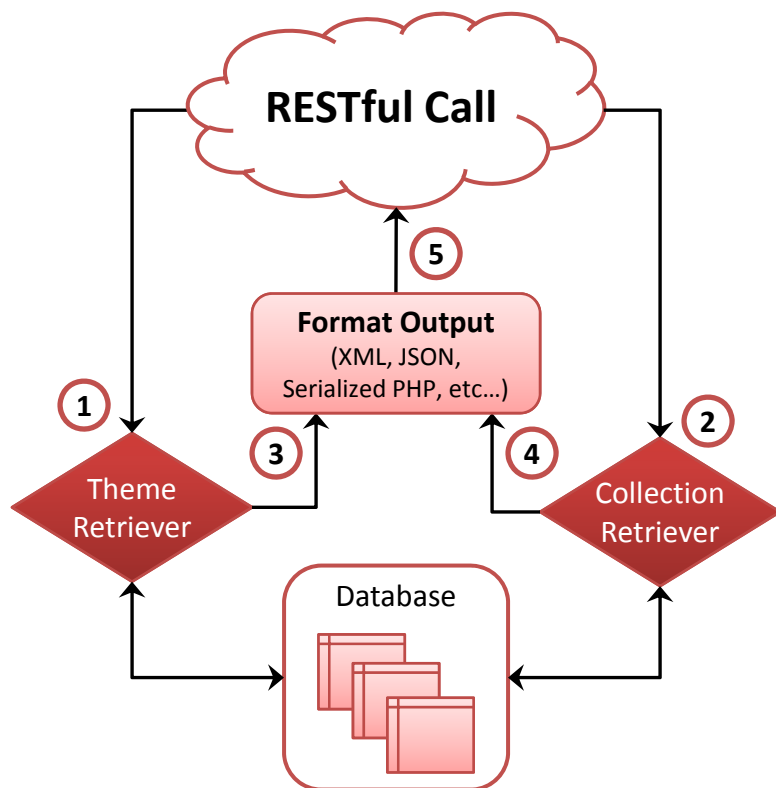


Service Layer

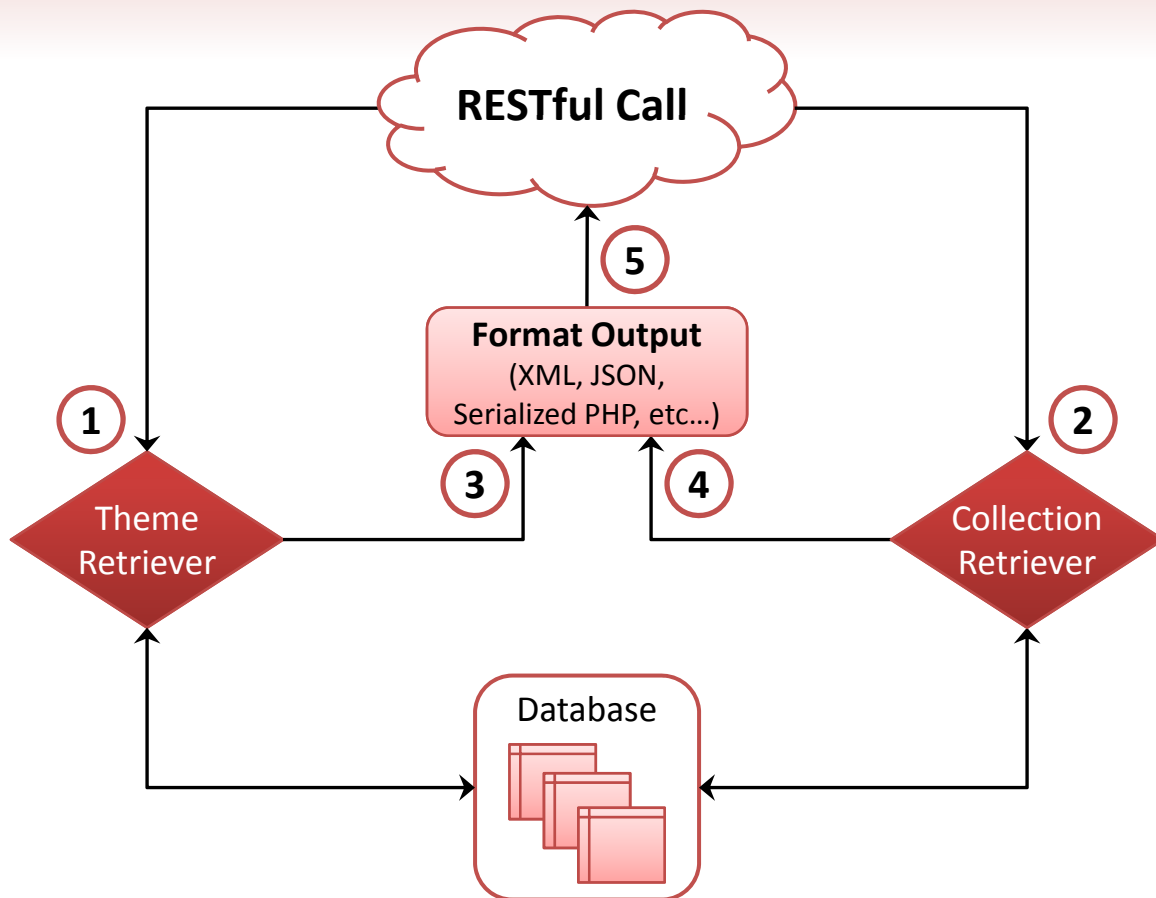
Search Request



Form Request



Numbers indicate the order internal operations would be performed.



Form Request

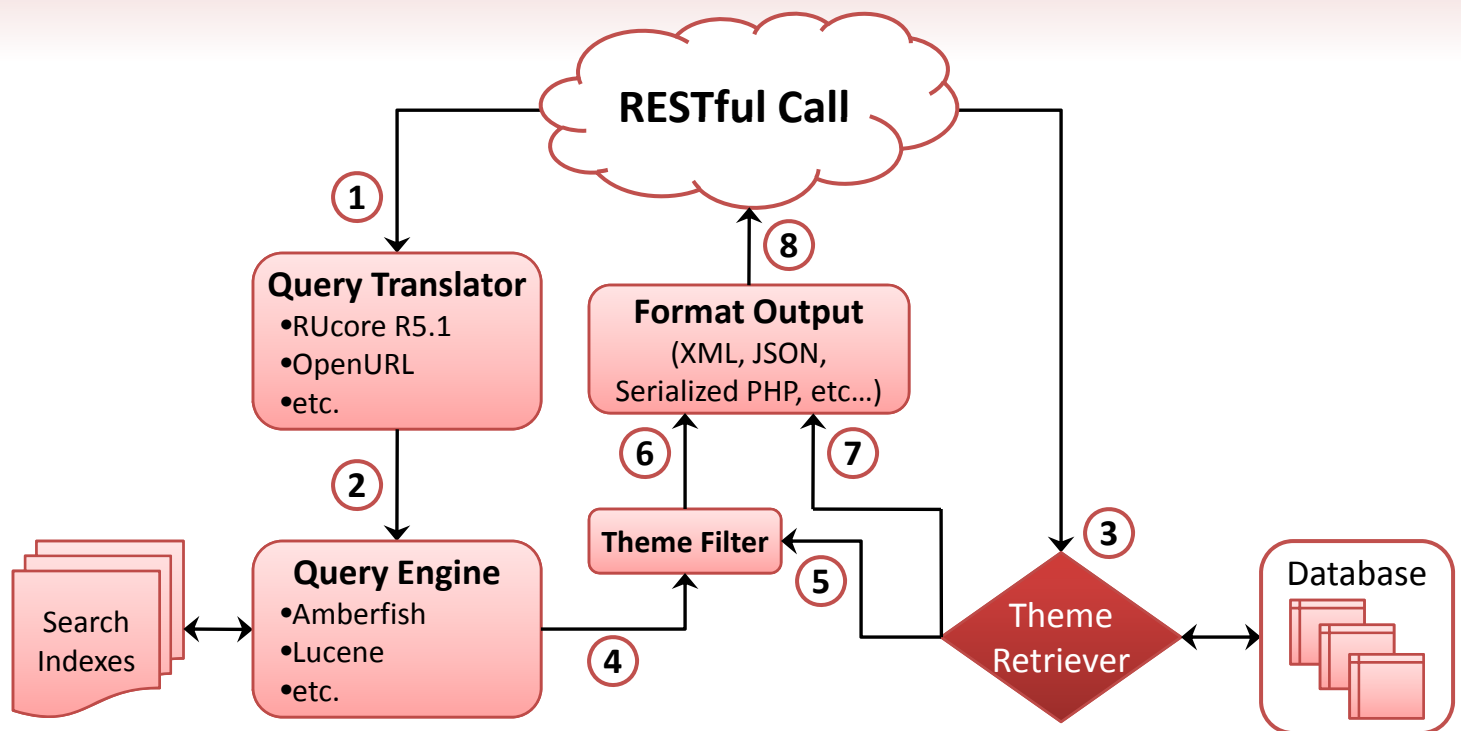
Definition:

A service call that would return defined information regarding Look & Feel and collection assignment made through the Search Service Configuration. Intended to be used to create a search form.

Operations:

- *Theme Retriever (1)*: Retrieves theme information based on profile key passed in RESTful call. This profile key matches the theme selected when using the Search Service Configuration.
- *Collection Retriever (2)*: Retrieves collections chosen in the Search Service Configuration and returns result information. In the case of non-hierarchical a flat list of collections would be returned. For hierarchical it would be represented in a hierarchical model.
- *Format Output (3, 4, 5)*: Resulting output from request would be formatted based on request parameter. Possible formats include XML, JSON and Serialized PHP.

Numbers indicate the order internal operations would be performed.



Search Request

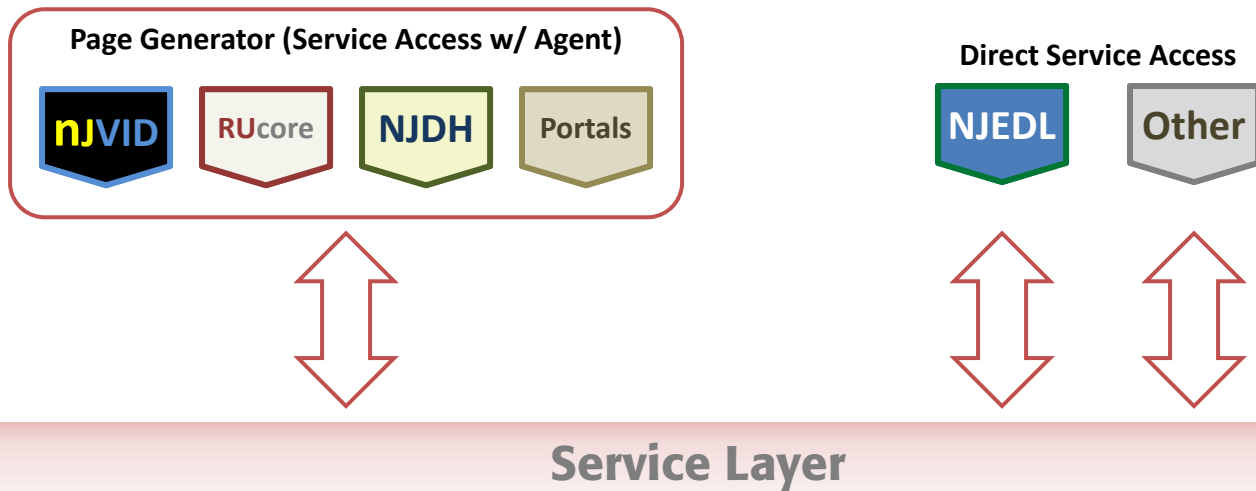
Definition:

A service call that would return search results and defined Look & Feel information made through the Search Service Configuration. Intended to be used to query for search results. Content will not be streamed through this mechanism, only result sets from the Query Engine.

Operations:

- *Query Translator (1)*: Translates submitted query parameters into an internal structure for querying for records. Source URI language is defined as a parameter in the RESTful call. When URI Language is not defined a best guess based on passed parameters would be made.
- *Query Engine (2)*: Creates query, submits and parses results of the applicable XML search engine being used.
- *Theme Retriever (3)*: Retrieves theme information based on profile key passed in RESTful call. This profile key matches the theme selected when using the Search Service Configuration.
- *Theme Filter (4, 5)*: Based on prescribed theme definitions results and filtered to return only requested information.
- *Format Output (6, 7, 8)*: Resulting output from request would be formatted based on request parameter. Possible formats include XML, JSON, RSS, Atom and Serialized PHP. Content will not be streamed.

Numbers indicate the order internal operations would be performed.



Service Access

Definition:

Communication between an outside agent and the service layer of the Search Service.

Operations:

- *Page Generator (Service Access w/ Agent)*: Performs request and response communication with the service layer. Generates display of search forms and results to the end user. This is accomplished through the transformation of Format Output, which is returned by the request.
- *Direct Service Access*: Performs request and response communication with the service layer.
- *Service Layer*: The layer that provides request and response.