Purpose

Provide single sign-on functionality with centralized authentication and authorization capability for the RUcore infrastructure. Allow trusted applications access to a user's authorization roles/permissions once the user has been authenticated.

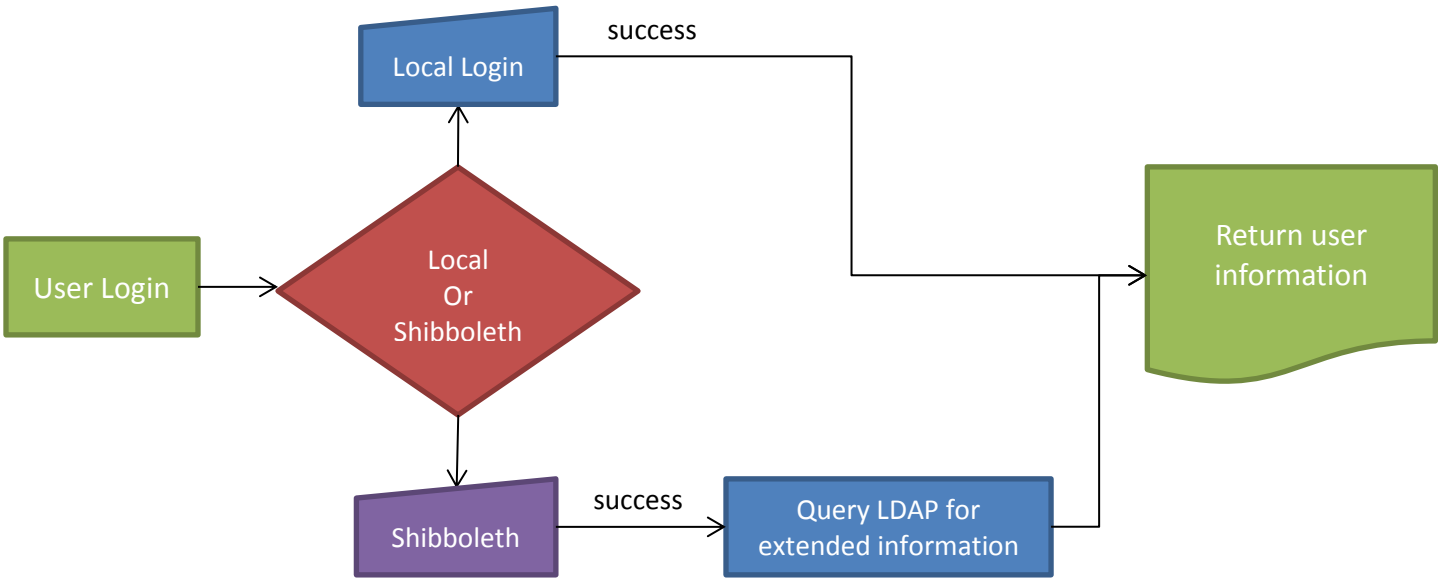
Initially, the applications that will use this functionality include the Deposit Module, RUanalytic, and User Account features.

Authentication – Methods, URL's, and HTTP Secure

Authentication

For authentication a central data source can be used to provide access to a majority of the RUcore user base. Ideally we should use either Rutgers LDAP or Shibboleth, via CAS. RUcore datastream access restriction currently uses Shibboleth as the authentication source. The deposit module offered by WMS uses LDAP for authentication. The RUanalytic Tool offers two options for users; Shibboleth or a local account controlled by a local database. Ideally authentication would be provided by a single mechanism; however since we are required to offer user accounts to non-Rutgers users a local account feature must exist.

A decision between using Rutgers LDAP and Shibboleth for Rutgers users must be made. LDAP is attractive because it offers a good deal of user information that applications can use to make decisions when the users interacts with the application. An example would be the department a faculty member is a part of. Shibboleth doesn't offer this fine grain user information. Shibboleth does offer the opportunity for RUcore to expose users from other trusted communities basic access to RUcore applications, this is something LDAP does not offer. A practical solution might be to explore using Shibboleth to authenticate the user and once authenticated query LDAP for extended information about the users, such as department. The simple flowchart below demonstrates how this might work.



URL's

A base URL will need to be defined for authenticating and delivering user account services. A directory simply named "account" is the suggestion of this specification.

Example - http://rucore.libraries.rutgers.edu/account/

When an un-authenticated user accesses this URL they will be prompted with two choices. One choice is login using your Rutgers NetID and the second choice is login using your RUcore/local account. Once authenticated, the user will be directed to the user account "dashboard."

If an un-authenticated user attempts to access an application like RUanalytic the RUanalytic should redirect the user to the "account" URL. Once authenticated the account mechanism should redirect the user back to the application that the user initially tried to access. At that point the application needs to ascertain that the authenticated user has the proper authorization to use the application.

Unrelated to the single sign-on portion of this specification it might make sense at some point, now even, to move the deposit module under the rucore.libraries.rutgers.edu URL. Currently the deposit modules URL is https://facsub.libraries.rutgers.edu/rufd which redirects the user to a mss3.libraries.rutgers.edu URL. If the User Services Group thinks changing the URL makes sense we would need a new proposed URL, i.e. http://rucore.libraries.rutgers.edu/{submission application}/

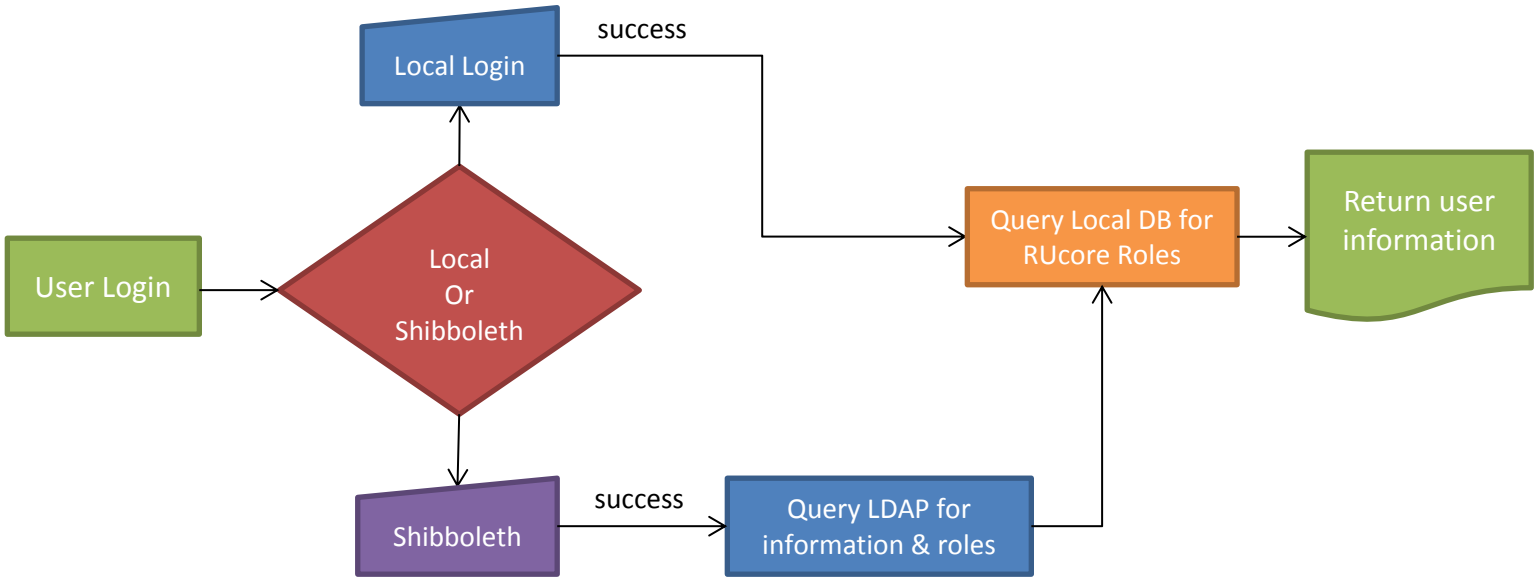
HTTP Secure

Currently the https://facsub.libraries.rutgers.edu/rufd site is configured for http secure. Users are prompted to submit a username/password combination at this site when authenticating. If Shibboleth is used the user is redirected to the Rutgers Centralized Authenticate Server (CAS) and a username and password are never submitted to the RUcore site. The only information passed back by CAS after a successful authentication is the username.

For RUcore/local accounts RUcore would be asking for a username/password combination. It might make sense to use https on the "account" pages, or possible site wide. It is this specifications recommendation that we implement HTTP Secure site wide, redirecting any non HTTP Secure traffic to HTTP Secure.

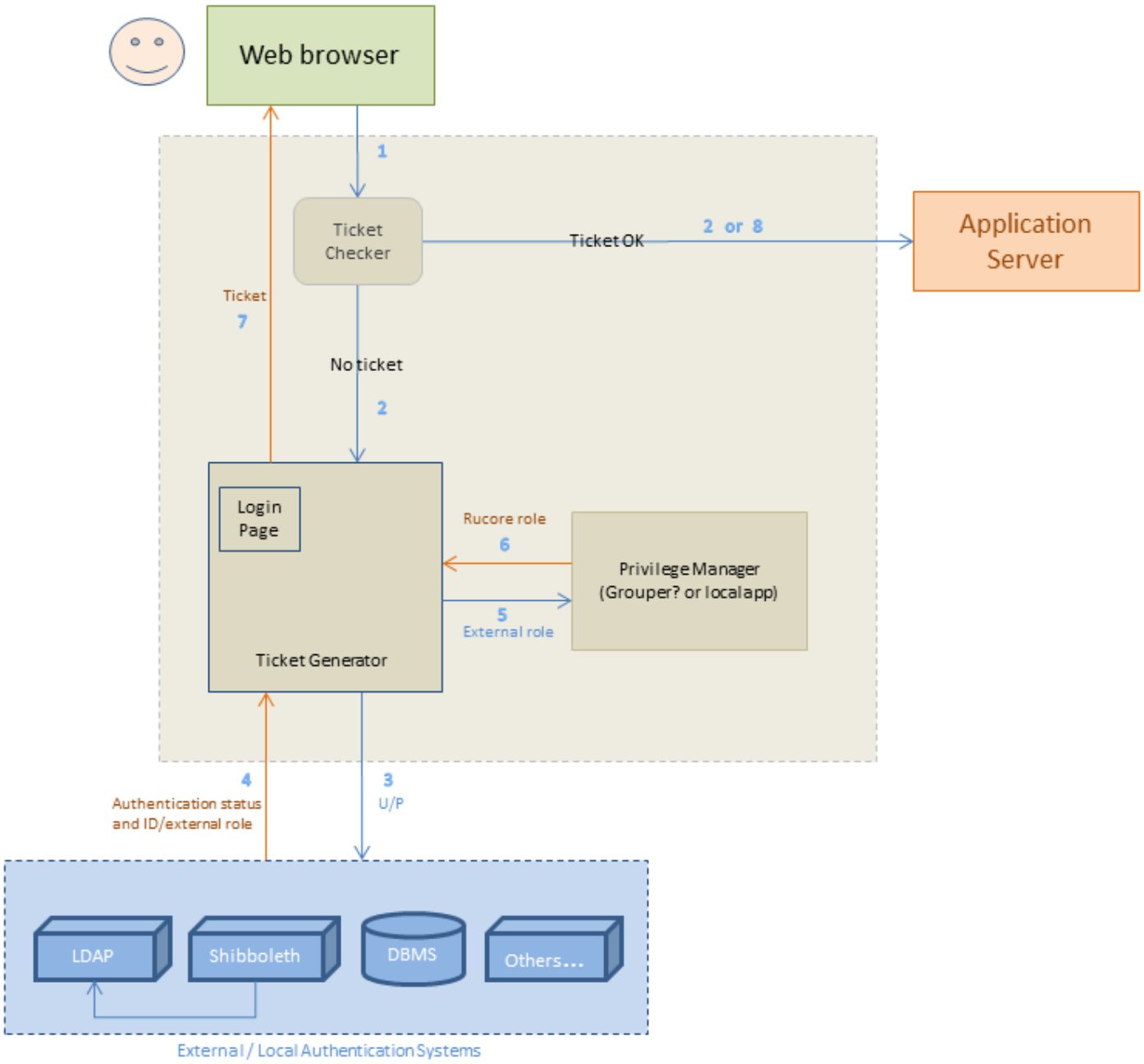
Authorization

Once a user has authenticated it will be necessary to retrieve role assignments that are associated with the authenticated user. These roles assignments might include general role settings for the user; staff, faculty, etc. or very granular application specific roles; e.g. administrator of the RUanalytic Tool. For shibboleth authenticated users that are part of Rutgers a query to the Rutgers LDAP server could provide general role assignments. RUCore/local user and very granular roles assignments could be stored and managed in an authorization database that is queried. Below is a diagram demonstrating how this might work.



Interfacing Applications with Authentication/Authorization

Application such as deposit, RUanalytic, and account "dashboard" will need to be provided an interface to validate authenticated users and receive authorization roles. Such an interface should be designed and implemented in a way that shields all of the shibboleth, LDAP and other local database queries from those applications and provide a simple discrete set of methods. Whether this is accomplished using a RESTful API with protected access to its methods or a local class is an item for discussion. Attached is a diagram demonstrating this layered architecture approach.


Web browser
1
Ticket Checker
Ticket OK
2 or 8
Application Server
Ticket
7
No ticket
2
Login Page
Rucore role
6
Privilege Manager (Grouper? or localapp)
5
External role
Ticket Generator
4
Authentication status and ID/external role
3
U/P
LDAP
Shibboleth
DBMS
Others...
External / Local Authentication Systems