# Usage stories of Authentication and Authorization

### DAX

The DAX APIs are REST APIs that allow a user or application to interface with MySQL (metaserv), qserv (dbserv), or retrieve images (imgserv). They are written in Python using the Flask library. qserv and imgserv will need to pass authentication down to dbserv and imgserv. Users can talk directly to DAX, or through an intermediary (Butler or FireFly). imgserv internally uses a Butler.

### SUI/FireFly

SUI/FireFly is, roughly speaking, a javascript Single-Page Application (SPA), a server which serves up the application, and a collection of APIs on the server which perform various functions, namely transforming data into something suitable for the SPA/javascript/http, eg. image tiling. In the case of DAX (qserv or metaserv), however, it's possible (and likely desirable) for FireFly to directly talk to the REST APIs, as they produce suitable data. If this was the case, these may very well be cross-origin requests (CORS), which don't play well with cookies.

There are other use cases when FireFly works best when running locally too. For example, when viewing a very large image, you may need quite a bit of memory (64GB+) for FireFly to work responsively, and this may be achieved best when FireFly is deployed to a beefy local machine (especially if you factor in client->server request latency when retiling images).

As a developer, it's desirable, to be able to run a FireFly servlet locally when developing. Workarounds for this include running locally but with a VPN, or continuously deploying to a development server at NCSA. If it's necessary to debug the application, these are often extremely slow and frustrating, especially with increase connection latency. It may be mandatory for a FireFly server to run at different "sites" during integration testing.

### Butler

There are Butler implementations which may talk directly to MySQL, or to a file system, or to the DAX APIs. In these are scenarios the Butler operates as a client API, possibly even communicating with services across several sites in order to determine the best site to fulfill a butler request.

### Qserv

MySQL/MariaDB internally uses roles for authorization. A custom PAM module may allow us to automatically setup additional attributes about a particular connection which could be useful for accounting, like QoS/rate-limiting. PAM modules are easy to write, so not too big of a problem there.

It is plausible Qserv will talk to servers outside of a given data center. There are two drivers of this: (1) recovering from failures after looking all replicas at a given site, and (2) moving queries to another data center during maintenance or to balance load.

### Usage Accounting

There hasn't been much talk about this. It's probably the case that a user-specifiable account attribute should be tied to the credential. (maybe a SAML attribute/JWT claim?). This is needed for resource management: consider a user that belongs to multiple groups, and need to "charge" different workloads against different quotas. This may merit custom PAM modules for MySQL/SSH, but maybe there's other solutions, like X-LSST-ACCOUNT header, export LSST_ACCOUNT, prepend qserv queries with: ALTER TABLE QservSession SET Account = 'desc-users'

### Level 3

There may be Level 3 applications, similar in function to FireFly (web or otherwise) which may interact with the DAX APIs or directly with qserv or other systems. As an example, DESC may write their own web portal, hosted at SLAC, which would act as an IAM client and need to pass on credentials to the DAX apis.