# SobekCM 2015 : Architecture and Code ReOrganization

Architectural Changes

Over the next six months we will be transitioning the SobekCM architecture to a more modular, separated architecture built upon a set of microservices exposed via a REST API. We will generally be building the API on JSON, although some native support for XML and SOAP will be added as well.

The first set of REST APIs should be available for comment within the next 30 days. This will include the APIs for searching and browsing items in an aggregation, retrieving a single item for display, and retrieving information about a single aggregation in the library. These API's will be made available for discussion and we ask for feedback from all interested parties.

In addition to these fairly obvious APIs, we expect to have a large number of reporting APIs, administrative APIs, and task-oriented APIs also available over the next six months. These will be rolled out on a monthly basis, to allow for feedback.

While the APIs will be made available for comment first, they will ultimately become the backbone of a separated system, as shown below:
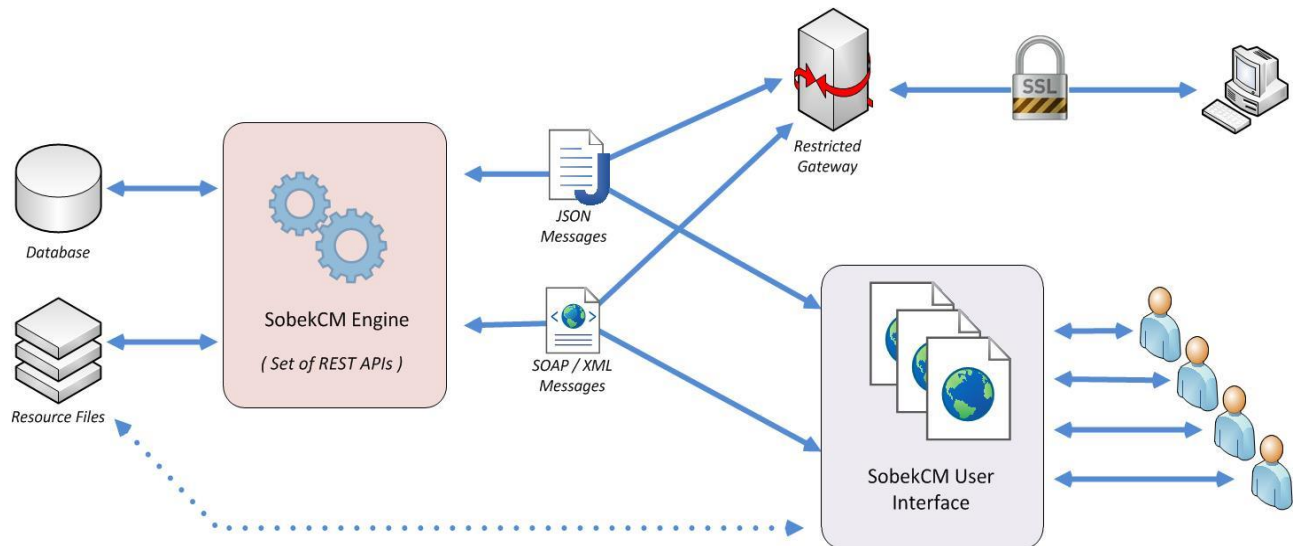
In this architecture, the *SobekCM Engine* is the data layer providing all access to the database. In addition, all changes to the resource files must be done through the *SobekCM Engine*. Each of the services are exposed via the set of REST APIs to the *SobekCM User Interface* web application. It is from this web application that your users will be served their content and interact with your library.

In addition to providing the services to the now separated user interface portion, the services can also be exposed via a restricted gateway, utilizing SSL, for you to consume in any other way you would like. This should provide controlled, but open access to your data and the ability to perform basic automation and integration tasks with other systems.

This architecture will allow institutions to fully customize their user interface and more easily control code rollouts and upgrades of each individual portion of their archive/library.

Changes

The database and the organization of resource files should change very little with these changes. But, during this transition, many of the class libraries will be drastically altered or split between new libraries. This process will mirror the rollout of the new APIs for comment.

Below is a diagram of the final hierarchy of class libraries that will be employed by the new architecture. A full description of each library and the philosophy between these organization will be forthcoming in the next several weeks.
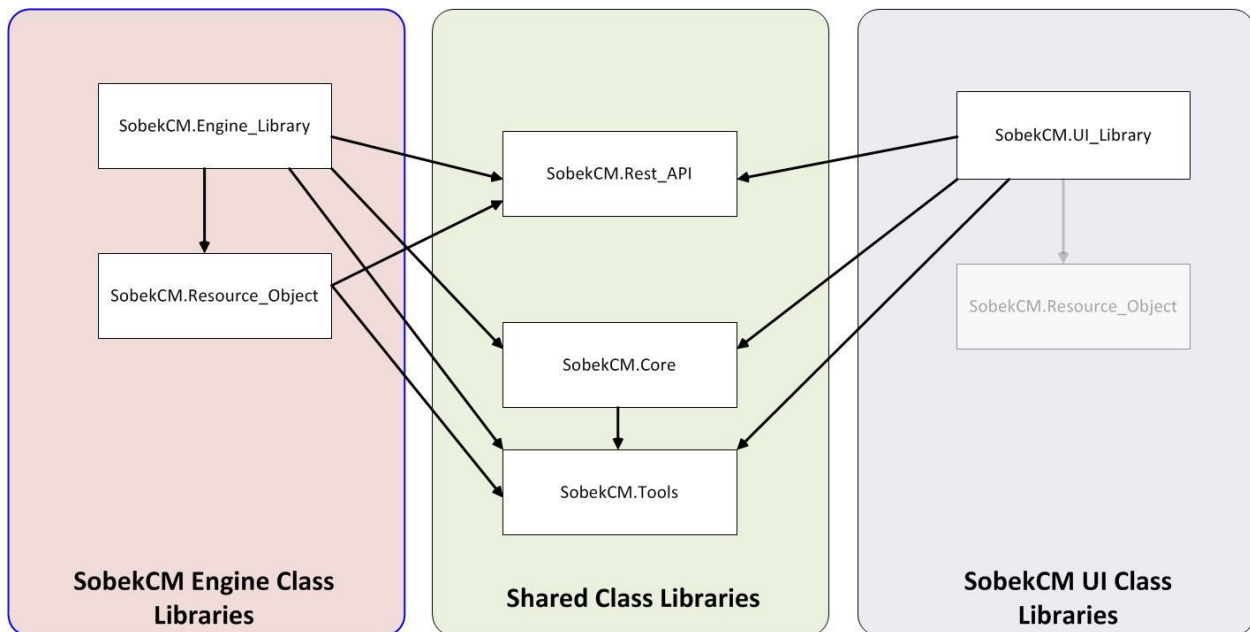

Hosted Solution

This architecture also allows for a variety of hosting configurations. A digital archive/library can be housed completely within a private cloud, as seen in the figure below:

Sobek Digital Private Cloud
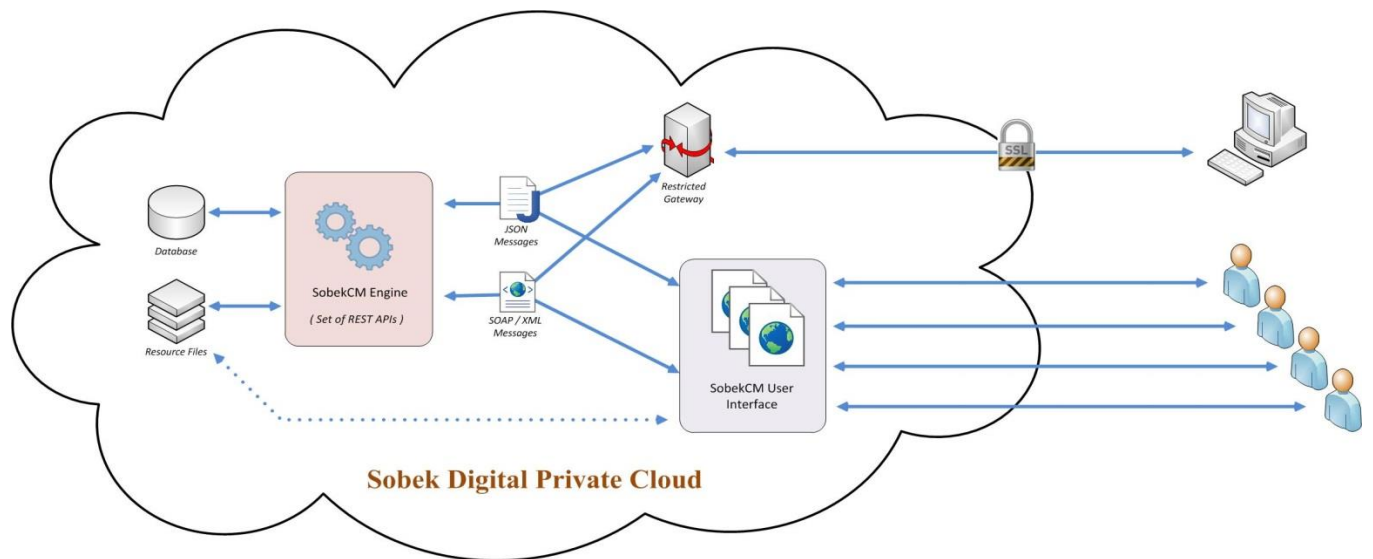
Alternatively, just the engine could be hosted within a private cloud, freeing institutions to only concern themselves with the user interface portion of the library, either modifying the SobekCM User Interface application, or creating their own interface in any language, such as Java, PHP, or C#.

Sobek Digital Private Cloud
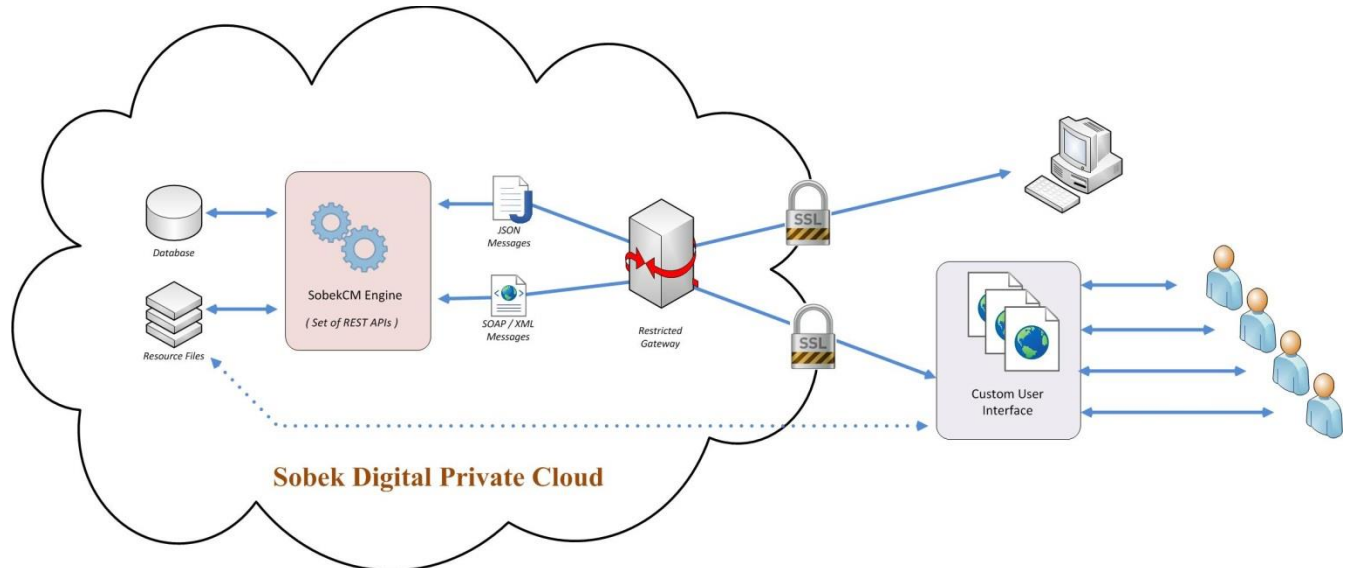
Stay tuned for announcements and watch GitHub over the next weeks to see our progress!