**Vision/Roadmap for caBIG iPhone Application(s)**

## Vision

Create a set of iPhone apps that highlight caBIG applications and provide the ability to quickly see available services and data on the grid. These applications should operate like other iPhone applications (follow Apple’s design and operation guidelines) and should respond quickly and handle situations where the iPhone data connection may be slow or unavailable.

## Features

The following functionality will be incrementally implemented in one or more iPhone applications:

1. Obtain a list of available grid **data** services and some basic data about each, to include:
   1. Name and description of service (caArray, caTissue, etc.)
   2. Main types of data exposed (Experiment, Sample, Specimen, etc.)
   3. Version of the application/model
   4. Hosting research center and POC information
2. Allow the user to select a list of favorites or bookmark specific services they are interested in
3. Track status changes in “bookmarked” services and push these as alerts to the user, and allow user to view current status on any service. Service status information will include:
   1. Changes in up/down status of service
   2. New services registered on the Grid
   3. Changes in versions of a service (upgrade)
   4. New data on a service or a count of available items in the main data types for a service
   5. Up/down time percentages for the service
4. Explorer for grid data service that allows a drill down into the data in that service. Types of drilldowns include:
   1. Count of items of a certain data type available
   2. List of basic information about each (name, id, description)
   3. Details on one item at a time within the list
   4. Simple search within the single data type
5. Simple search and form-based search service powered by caB2B. Will expose all caB2B web application saved searches and keyword searches.
   1. Should lead to a screen similar to 4(a) and 4(b) above
   2. Export and more extensive B2B filtering capabilities will not be supported

Priorities and planned release schedules for these capabilities will be outlined in the Roadmap section below.

## Architecture

The planned architecture for the iPhone applications is a client/server model where the iPhone app functions as a presentation layer – with some caching of data – and the majority of the business functionality is performed on a separate server application to be hosted at NCI.

This architecture is the most flexible and eliminates several hurdles that exist in other potential architectures outlined below:

1. Directly calling Grid services from the iPhone

While directly accessing Grid data services from the iPhone may be desirable as a demonstration of technology, several difficult problems exist. Grid data/analytical services (including the Grid Index Service) are implemented on top of Globus, which itself is based on a nonstandard Apache Axis implementation. To start, there is no real support for SOAP-based services in the iPhone SDK. In addition, the non-standard Axis implementation used by Globus would require significant hacking to any SOAP libraries anyway. There are also questions about the bandwidth requirements for Grid queries across the iPhone 3G network.

By using a separate “proxy” service, it is possible to control both ends of the iPhone to service communication. This allows the iPhone to use a REST API to access the service (which is well-supported within the SDK) and then have the service handle communication to grid services. In addition, the bandwidth used will be far less than receiving index service and CQL query responses directly to the iPhone.

It makes sense to continue to look at ways to directly call grid services directly from the iPhone, if only for the technical appeal of such a solution. As support for Java and SOAP improve in the iPhone SDK, we can prototype other implementations to see how feasible direct grid calls are going forward.

1. Using the caB2B server directly from the iPhone

On the surface, this seems like the right level of abstraction. Both the caB2B thick client and the caB2B web application use an identical interface to caB2B server, and it is logical that the iPhone application (as just another client) should use the same. However, the caB2B client-to-server interface is currently done through remote EJB calls, which are not supported in Apple’s Objective C SDK.

Using a separate service, again the iPhone can speak to the service in a simple RESTful protocol, while the Java service can communicate directly with the caB2B server behind the scenes using the EJB interface. This also isolates the B2B interface in such a way, that if we later want to use another B2B interface (e.g. a caB2B-provided portlet), the iPhone application itself does not need to change.

The client-server model for the iPhone application also allows us to centrally maintain information about the state of the grid and its services without requiring each iPhone client to ping each service (including the grid index service) for updates. The iPhone proxy service can collect new data from each grid service once an hour and can respond to all iPhone clients as needed without generating additional load on the grid index or the services themselves.

The caching capability on the iPhone itself will be limited to small queries and is intended to provide some sort of reduced/minimal functionality if the iPhone is off the data network. This will allow some form of download of results to be browsed later and updated when the iPhone is back on the network.

## Roadmap

The proposed roadmap for this application is based around an Agile incremental development process with one month iterations. The functionality is planned as follows:

Iteration 1: Develop Feature 1 above as an initial demonstrable proof of concept. This will result in a iPhone client that can obtain a list of available grid services and allow the user to browse through some basic data about each. It will also produce the initial version of the server application and validate the architecture and interface between the two.

Iteration 2: Develop Features 2 & 3 above. This will include additional server-side development to ping grid services for availability and count/types of data available. The iPhone client will grow to allow bookmarking of grid services and a push/pull feature to receive updates on these services. The architecture will remain flexible so that additional data on grid services can be tracked and queried by the iPhone app as more requirements are uncovered.

Iterations 3 and up: Develop Features 4 & 5 and/or other requirements that have developed. We anticipate by this point that new requirements will have been identified based on demonstrations and early use of the iPhone app. Search and browse are certainly high on the priority list at this point, but identifying specifics of what will be implemented in iterations beyond #2 should be delayed until the end of iteration 1 at the earliest.