National Cancer Institute
Center for Bioinformatics
(NCICB)

Vision

caArray

Last Revised:	5.8.2007
Produced By:	NCICB Development Team
Version:	0.8

Document Approvals

The list contains the name for the core project team and any key stakeholders who have an interest in the success of the project. An "S" identifies persons responsible for approval from the Business and Development. Sign off of the document would be required when a decision is made not to take action for defined gaps.

S	Name	Role	S	Name	Role
S	Anand Basu	NCICB Engineering Manager	S	Juli Klemm, Ph.D.	Product Manager
	Brent Gendleman	Development Project Manager		Jerry Eads	NCICB Project Manager
	Eric Tavela	Development Architect		Ron Keene	NCICB Independent QA
	Don Swan	NCICB Application Support			

Revision History

When you make a change to a document, you must add an entry to this Revision History table and you must manually type the Last Revised Date on the front cover.

Date	Version	Description	Revised by
03/12/2007	0.1	Initial Draft with a focus on Stakeholder descriptions and identification	Brent Gendleman
03/23/2007	0.2	Detailed the Problem Statement and Product Positioning	Brent Gendleman
04/06/2007	0.3	Updated from Internal Requirement Sessions	Brent Gendleman
04/11/2007	0.4	Updated through product positioning	Brent Gendleman
4/15/2007	0.5	Updated Business Opportunity and Product Positioning	Juli Klemm
4/18/07	0.6	Filled in Needs	Brent Gendleman
5/3/07	0.7	Flushed out Features based on in person cancer center interviews and documentation they provided	Brent Gendleman
5/8/07	0.8	Reviewed and updated needs and features per consultation with Product Manager	Brent Gendleman

Copyrights and Trademarks

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this document is to collect, analyze, and define high-level needs and features of the next generation of caArray. It focuses on the capabilities needed by the stakeholders and the target users, and why these needs exist. The details of how caArray fulfills these needs are detailed in the use-case and supplementary specifications.

1.2. Scope

Successful software naturally evolves to meet the changing needs of its user community and the "world" in which it exists. caArray has been deployed and in use since January 31, 2005, and the team has determined that several factors encourage us to review its first major revision since its original inception.

The factors we have considered include:

- Revisit the vision of caArray to ensure our evolved view of its raison d'etre is optimized going forward.
- Revisit of the user community to understand changing needs and current growth potential
- Review of other industry and competitive developments which should be considered for co-opting, collaboration, or capitulation.
- Introspective review of the 1.x architecture and features to ensure the best elements are elevated and the lesser elements are retired or replaced.
- Incorporate elements of caBIG-wide architecture that have matured since the creation of 1.x caArray (e.g., the Common Security Model).
- Re/Produce supporting software assets that may have received insufficient attention or grown out of sync with the software over time to ensure 2.0 is clearly communicated to both internal and external stakeholders.

An additional benefit of this activity is it will put essential assets in a form that our development team uses to schedule, prioritize, and reference as the new generation moves toward implementation. In short, all software applications require periodic consideration for major revision to ensure efficient and pertinent evolution. caArray is no exception. The team is enthusiastically anticipating driving the next generation of caArray!

1.3. Definitions, Acronyms, and Abbreviations

See caBIG Glossary: https://cabig.nci.nih.gov/glossary and NCI Dictionary of terminologies: http://nciterms.nci.nih.gov/NCIBrowser/Dictionary.do.

For any terms or acronyms not captured in the caBIG glossary, we will capture them in a caArray glossary and request their addition to the caBIG glossary.

To support understanding of this document, the following terms are defined here:

Program – a collection of projects that support a specific scientific goal

Project – a set of annotation and array data that describe the experimental design, the samples used and the array data produced.

Source – the material – typically tissue or cell lines - from which a sample is taken

Sample – the material used for extraction onto an array

Annotation – the project metadata that describe the "what" and "how" a project was executed

Array – the raw data that is produced as a result of an experiment

1.4. References

1.4.1. caArray sites of interest

NCICB Production Site: http://caarraydb.nci.nih.gov/caarray/ – the public instance of the caArray Portal with links to caAMEL, the MAGE-OM API, caAMEL and grid services.

Product Summary Site: https://cabig.nci.nih.gov/tools/caArray - the summary of caArray capabilities and direction

Public Information Site: http://caarray.nci.nih.gov/ – a public web site that allows anyone to download the latest version, access documentation, launch the portal and visit sites that provide analysis of the data contained in caArray.

caArray Work Group Site: https://cabig.nci.nih.gov/workspaces/ICR/caArray-wg/ – this public web site provides access to the schedule, monthly meeting notes and links to the listserv for the stakeholder community

Microarray Gene Expression Data Society - http://mged.org/ The providers and curators of microarray standards, software and models.

Welcome Trust Sanger Institute – http://www.sanger.ac.uk/Software/formats/GFF/GFF_Spec.shtml - for the General Feature Format Specifications

2. Positioning

2.1. Business Opportunity

The National Cancer Institute (NCI) has launched the caBIG™ (cancer Biomedical Informatics Grid™) initiative to accelerate research discoveries and improve patient outcomes by linking researchers, physicians, and patients throughout the cancer community. caBIG™ serves as the cornerstone of NCI's biomedical informatics efforts to transform cancer research into a more collaborative, efficient, and effective endeavor. In 2007, caBIG™ has moved from the pilot phase to the enterprise and is looking to evolve its set of maturing assets. A significant business opportunity exists in this evolution to make the day to day lives of the researchers who produce and consume data at the local level easier and in so doing, meet the needs of the global cancer community.

As the scientific community begins to better understand cancer at the molecular levels and personalized medicine is implemented in cancer patient care, researchers and clinicians will require more rapid access to—and easier methods to analyze—the multiple types of information involved. However, in many cases the systems needed to help translate the necessary data into better patient outcomes are either non-existent, disconnected, or underperforming. To this end, the vision of caBIG™ is a full cycle of integrated cancer research, extending from bench to bedside, and back again. caArray a committed following, years of user feedback and contribution on which to build from, sufficient resources, and an expansive vision to support integrative cancer research.

Expression profiling is now a standard tool set with which to interrogate biological systems. Parallel advances in computing and new array technology provide an opportunity for collaboration and discovery within the scientific community and across traditional boundaries to reach clinicians and ultimately patients. The insistence on open source development provides the community with the greatest opportunity to gain access to the tools they desperately need to execute their respective mission. caArray was initially developed with expression profiling in mind, using the caBIG Compatibility Guidelines, as well as the Microarray Gene Expression Data (MGED) society standards for microarray data. Compatibility with these standards and guidelines was and remains required. However, the ability to add new standards that are developing is also necessary to facilitate data exchange and analysis across domains. A number of analytical tools and services that connect to caArray are already available - including geWorkbench and GenePattern - that provide a variety of analysis, visualization and annotation functions for microarray data.

The primary goal of caArray is to further translational cancer research through acquisition, dissemination and aggregation of high quality array data to support subsequent analysis. Initially envisioned to be the de facto repository for cancer related expression data, other applications – primarily the Gene Expression Omnibus (GEO) – has assumed the role. However, the quality of the expression data and the ability to integrate the use of array technology into clinical research remains at issue. Further, the opportunity for caArray to evolve to handle other types of array data will provide greater impetus for use among the Cancer Centers and their collaborators which will ultimately benefit the cancer community. Data from expression, SNP, methylation, and protein arrays are anticipated for inclusion in caArray. Also under consideration are the burgeoning platforms of RNAi and CHip-chip data. Tissue microarrays represent another array-based technique that require a significantly different business process in their creation but are being considered for inclusion in the repository.

A significant challenge is to find common ground for the annotation, upload and extraction of these array data platforms to support meaningful analysis for cancer research. The need for logical and expedient approaches to storing, querying, retrieving and reporting on array–based data has increased over the last several years due to the contribution of several factors:

- a significant increase in array information (expression and others) available due to:
- decreases in the cost to generate this data
- improvements in the technology to generate this data
- advances in the approaches to analyze this data
- the need to increase the numbers of samples to enhance discovery and ensure validity of results once found;
- and the need to ease the administration of the data and results by giving the community a single point of reference to perform the most essential tasks of array-based research.

The initial generation of caArray (1.0) and subsequent point releases (currently 1.5) have provided the interface for annotating scientifically significant meta-data along with the independent ability to upload and download data through an applet or accessing the MAGE-OM API over the grid service provided by caBIG. caArray will improve upon this starting point by organizing the application around the natural workflow between investigators and the array labs that serve them, improve the user experience for storing and retrieving the data produced, query the data through an easier to comprehend API, and bridge the gap between the analysis tools in heavy use in the community today and the data they need to consume. A simplified organizational structure for basic project relationships that fundamentally represent "annotation" versus the resulting array "data" is shown in Figure 1. This is a definition we will use throughout this document and is an overall theme of the application.

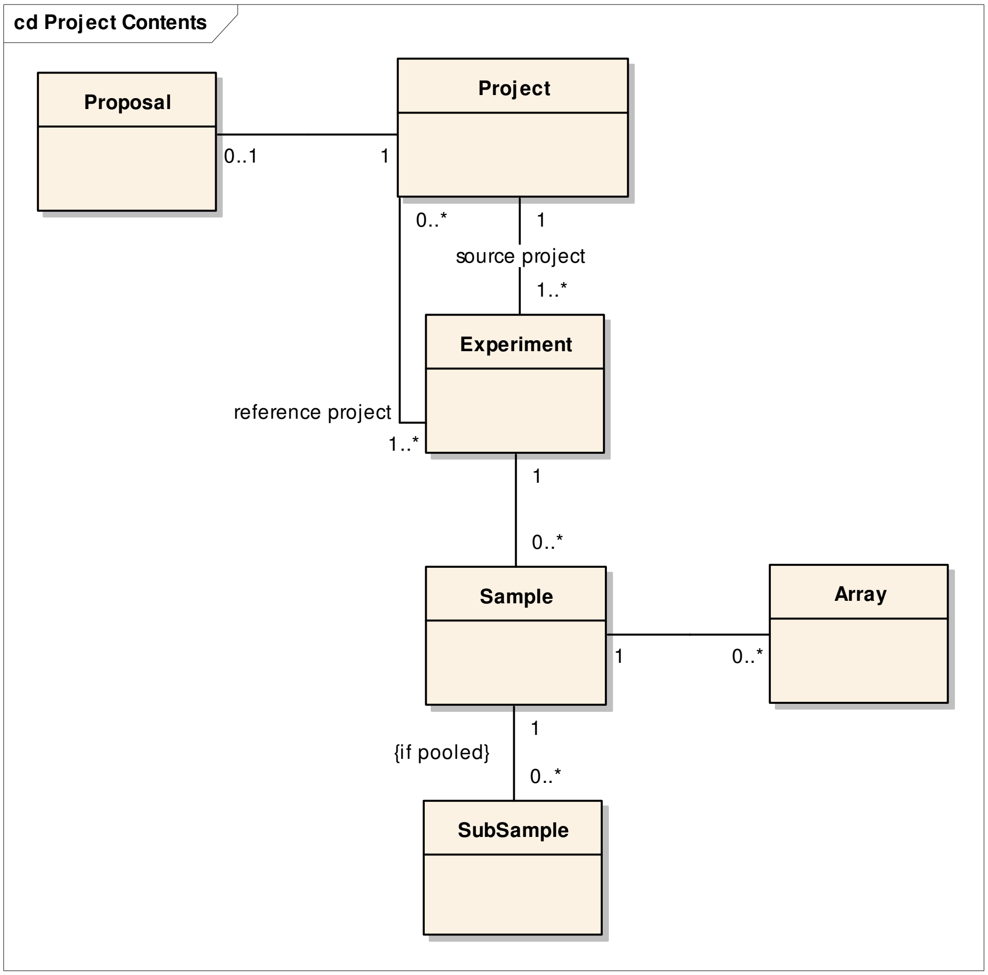


Figure 1: Basic Organizational Structure

caArray will look to perform intra-platform, intra-experiment queries for local adopters, via the NCICB instance and potentially across the grid collectively to increase the availability of quality data.

caArray will be built to scale with an open architecture and supportive documentation to allow for future enhancements, particularly with regard to interfacing with additional analysis tools, the ability to query across platforms, and poised to exploit web services and/or databases from other components of caBIG when available and prudent. The desire to create an extensible array system that is non-platform-specific and potentially customizable is a theme that should influence the building of caArray.

2.2. Problem Statement

The problem of	Platform vendors and custom crafters not universally producing a single shareable format for the annotation of array experiments and the data that results; The lack of properly annotated high dimensionality molecular data that can be seamlessly integrated into the execution of clinical research; The amount of annotation needed to support the validation, retrieval and analysis of large sets of data; Data repositories that meet the publishing requirements of journals but not the scientific needs of geographically disparate researchers
Affects	The cancer research community, the life science community as a whole, clinicians and ultimately their patients access to the highest quality care
The impact of which is	A variety of incompatible tools and services disparately made available that limits the ability to share and benefit from others in the community who are generating otherwise useful array data,
A successful solution would be	One that permits the community to share and analyze array data regardless of the platform used to generate it, tied to meaningful sample annotations.

2.3. Product Position Statement

For	▪ The NCI and its affiliated institutions
Who	▪ Need to annotate, store, and share array annotation and data as part of cancer research and clinical trials in a caBIG compliant manner
caArray	▪ Is an open-source, user-driven, role-based, web and programmatically accessible data management system that guides the annotation and exchange of array data through a federated model of local and centralized installations
That	▪ provides browser-based and programmatic access to the data stored locally; ▪ enables mechanisms for accessing all local installation data over the caGrid; ▪ provides interoperability with a wide set of other data repositories available over the caGrid; ▪ supports silver compatibility with caBIG

	guidelines; ▪ promotes compatibility with the MIAME 1.1 guidelines and supports the import and export of MAGE-ML ▪ provides the data for caBIG analytical services
Unlike	▪ Gene Expression Omnibus ▪ Array Express ▪ Stanford Microarray Database ▪ And a variety of commercial proprietary and open-source solutions from microarray vendors and service providers
Our product	▪ is a completely open source (web application server and database) and funded by the NCI to continue its open-source evolution ▪ supports the natural workflow of the community across array approaches ▪ caGrid-Enabled ▪ is caBIG compliant at a Silver Level

3. Stakeholder and User Descriptions

3.1. Market Demographics

The overall market for caArray is expansive. caBIG™ casts a wide net across more than 50 Cancer Centers and a significant number of collaborators world wide. While each Cancer Center isn't necessarily a core facility for producing array data, it is readily apparent that the interrogation and analysis of array data is among the techniques that will drive discovery, trials and treatments and is therefore, a necessary tool for each center and their external collaborators. Two methods for accessing caArray have been established. The first is to use the centralized portal and repository hosted by the NCICB. The second is to download the application packaged with necessary external components to install locally. Currently there are 550 NCICB caArray Portal Users and roughly 6 known local installations across the community.

The market goes beyond the core set of users who will populate caArray to those who want to interrogate the data they produce in unique ways to support the translational model of discovery. This interrogation will come through manipulation of a web-based dialogue with a Graphical User Interface (GUI) or through accessing the data through a well-documented Application Protocol Interface (API) at the central NCICB repository or another other exposed data set across the grid.

3.2. Stakeholder Summary

3.2.1. External Stakeholders

These stakeholder types represent the breadth of the target audience that lie outside the core development team for the caArray application. At least one individual will be named as a representative for each Stakeholder type, though there is no restriction on the number of representatives or the ability for one individual to represent more than one stakeholder. In addition, not all stakeholders are end-users of the application.

Table 1. External Stakeholders

Type	Description	Responsibilities	Representative
NCICB Adopter	A user or group of users who have adopted the NCICB version of caArray but who haven't or don't plan to use local installations of caArray.	Requirements Specifier (excluding environment) Contributes to feature prioritization Potential Alpha Tester Beta Tester	Hanxin Lu (NCI/DCB)
Local Installer	A general term covering user groups who have installed at least one version of caArray (1.3, 1.4, or 1.5)	Requirements Specifer (including environment) Contributes to feature prioritization Potential Alpha Tester Beta Tester	University of Pittsburgh Georgetown University Thomas Jefferson Wistar
Code Contributor	Anyone who has or plans to provide complete modules, enhancements or fixes to any of the underlying code of caArray	Requirements Specifier API Reviewer Follower of practices and procedures for code contribution	Gerald Fontenay (LBNL)
caArray Integrator	Anyone who has or plans to access caArray's API to integrate into their own application either under the caBIG umbrella or outside of it.	Requirements Specifier API Reviewer API Alpha and/or Beta Tester	Subha Madhavan (NCICB) Aris Floratos (Columbia)
Desired Adopter	An individual or group of users who is targeted for adoption but for many reasons has chosen not to adopt caArray at the present time.	Requirements Specifier May contribute to feature prioritization Alpha/beta tester?	Cancer Center Community (e.g., Northwestern, Dana Farber, Washington University)
caGrid Team	The underlying service oriented infrastructure that supports caBIG™	Expose and maintain service definitions	Avinash Shanbhag
caCORE Team	Cancer Common Ontologic Representation Environment; open-source software that helps to streamline informatics development throughout the cancer community	Validate Common Data Elements (caDSR) Validate Controlled Vocabulary (EVS) Validate UML supporting a Model Driven Approach	Denise Warzel

Type	Description	Responsibilities	Representative
NCICB Systems Team	Provides system resources for development, test, and production	Contributes to Environment Requirements Supports all Environment Setup and Maintenance Provides Help Desk Support	Doug Kanoza
Satellite Environment Team	The IT team that supports local installations of caArray for our end user community	Contribute to Environment Requirements Support Installations and Updates	James Li (GTown), Carl Swanson (TJ)

3.2.2. Internal Stakeholders

These stakeholders are, at a minimum, focused on the evolution of caArray. For the purposes of this document and the caArray project, their perspective is squarely on caArray and its contribution to the success of caBIG™'s collective stakeholders.

Table 2. Internal Stakeholders

Name	Description	Responsibilities	Representative
caBIG Program Leader	The visionary across the caBIG enterprise.	Provide strategic input and validate the vision and scope of caArray.	▪ Ken Buetow
CBIIT Chief Operations Officer	Head of operations for CBIIT.	Provide input into the position of caArray in the broader scope of all CBIIT products.	▪ Peter Covitz
caArray Product Manager	Ensuring over time that caArray comfortably meets the needs of all stakeholders by continually monitoring and modifying the elements of how it is marketed and forecasting what caArray needs to provide to be successful.	Provide input into the direction for the evolution of the software leading the CCB. Monitor development, marketing and outreach Ensures that there will be a market demand for the product's features	▪ Juli Klemm
caArray Engineering Manager	Responsible for the ultimate success of the application both as an independent service and under the umbrella of caBIG	Monitors the project's progress Ensures that the system will be maintainable Monitors all facets of the project's progress Approves resources and funding	▪ Anand Basu

Table 3 User Depictions

Name	Description	Responsibilities	Stakeholder	Representative
Principal Investigator	Designers of experiments that will use array technology	Overall execution and quality of the science	NCICB Adopter, Local Installer, Desired Adopter	Monica Parenelli (Pitt)
Research Scientist	A support position in the Principal Investigator's department	Data entry and submission of projects for their PI	NCICB Adopter, Local Installer, Desired Adopter	TBD
Lab Administrator	Responsible for coordination and often reporting of the lab's day to day work.	Organizing and managing a lab's production, sample storage, and meeting the needs of the lab's researchers	NCICB Adopter, Local Installer, Desired Adopter	Paola Fortina and Jack London, Kathryn Scott (TJ, Local installer) Habtom Ressom (GWU, Local Installer)
Lab Scientist	Performs most of the lab's day to day scientific work with regard to preparing of samples, running chips and checking quality.	Sample receipt and storage Management of experiment runs Quality Control Raw Data Production	NCICB Adopter, Local Installer, Desired Adopter	TBD
Biostatistician/Informatician	Performs the analysis of the data derived from array experiments	Analysis of Raw Data Creation and Management of Algorithms and Scripts Reporting of Results	NCICB Adopter, Local Installer, Desired Adopter	Robert Gentleman (Fred Hutchinson) Baris Suzek (Gtown) Uma Chandran (UPitt) Louise Showe (Wistar) Abhijit Dasgupta (TJ) Simon Lin (NWU)
Curator	Reviews their own or their labs data entry and may change and contribute data terms to increase quality of submissions	Annotation Quality Control	NCICB Adopter, Local Installer, Desired Adopter	TBD
Anonymous User	Views the public projects on the NCICB instance of caArray	No responsibility	N/A – all stakeholders will dictate what is and what isn't available to an anonymous user	All

Name	Description	Responsibilities	Stakeholder	Representative
External System	Programmatically integrates with or interrogates the NCICB and/or a local instance of caArray	Use API to request and receive data	caBIG Integrator, TCGA, caBIO, caTissue, Analysis Tools	Subha Madhavan - Cancer Genome Atlas Aris Floratos (Columbia) WorkBench, caBIO Michael Becich-(Pitt) Honest Broker, caTissue

3.3. User Environment

In this section, we will cover vital elements that reflect the environment of our user community – the processes they have in place, the equipment and materials they use to conduct their experiments and the tools they commonly use to consume the data placed into caArray.

3.3.1. Institutional Structure of the Community

The array community caArray serves has a variety of organizational approaches to the set up of array based labs and the relationship to the research scientists they serve. The investigation into those structures has produced the following generic model that represents most of the institutional pattern for how people work within the cancer community. The model will be represented by the type of software roles we create, the assignment of individuals to those role or roles, allowances for collaboration within and across the institution those individuals are associated with, and the granularity of permissions when exposing projects as public.

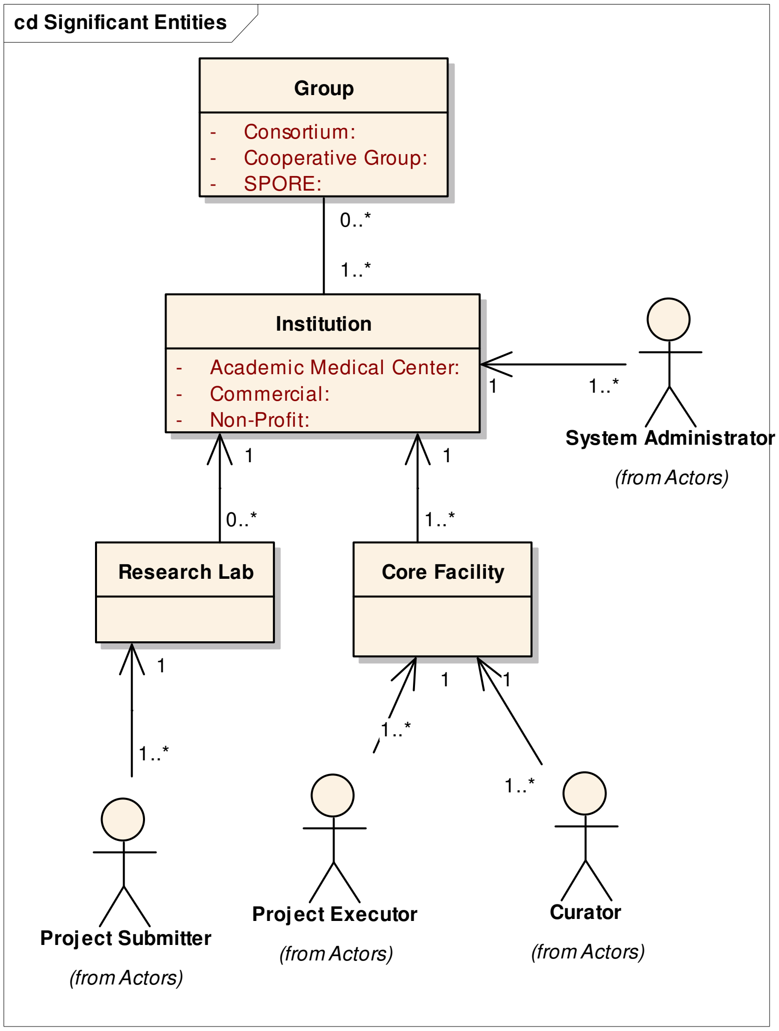


Figure 2. Institutional Organization and Generic Roles

At the top of the model is a general construct called “Groups” which serves to group 1 or more institutions and the underlying research and lab facilities that are associated to it. The most common types of groups in the community are **consortiums**, which primarily come together to share resources and expertise across institutions both inside and outside caArray’s cancer center focus. Examples of such groups include Cooperative Groups, which organize researchers, cancer centers and community doctors to support clinical trial activities and are officially recognized through grants; and the SPORE program, which are organized to promote interdisciplinary research and speed the bi-directional exchange of information between basic and clinical sciences.

caArray’s first priority are Institutions that are a part of the Cancer Center community. However, these centers’ array labs often support more than just cancer related inquiries and caArray will be generic enough to support non-cancer projects.

In addition, all cancer centers are not organized in the same manner. Many do have Core Facilities, which provide a consolidated, shared resource of expertise that is typically available to anyone inside the Institution and occasionally to scientists outside the home institution if a research agreement is in place between the respective institutions. Still others don’t have any one core facility, but rather several independent array facilities that offer different threads of expertise – such as genotyping, gene expression or proteomics. Regardless, there are core sets of roles that can be organized to support the array labs (project executors) and the researchers (project submitters) who request their services which are represented in Figure 2.

3.3.2. Project Lifecycle

The typical lifecycle ranges anywhere from 3-6 months from the point a project is initially discussed to the delivery of results by the lab to the requesting investigator. To this day, the most common method for the distribution of data back to the investigator is the creation of DVDs (to accommodate the larger file size that CDs can't store). Disconnection between the PI's and the Lab remains a consistent problem though our anecdotal evidence from the end users suggests that the state of collaboration is improving. In fact, some of the target institutions are requiring a minimal set of information to be shared by the investigator (vs. walking over to the lab and handing them samples to be run). A generalized depiction of the work flow from start to finish is described below in Figure 3.

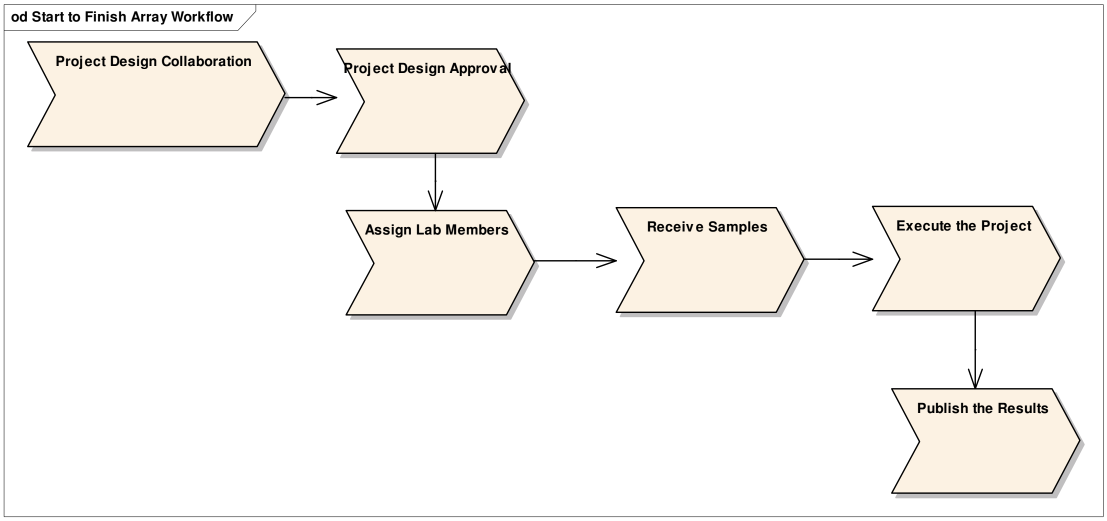


Figure 3. Overall Workflow

An additional item of importance is to allow for the establishment of how much a project will cost and what charge code will be used – for example, a purchase order or a grant number – as historically, payment has been an issue for doing work and it remains an issue regarding analysis. Most of the labs we spoke with won't begin a project without a known source of revenue to cover the costs.

3.3.2.1 Project Design Collaboration

Collaboration between the investigators and the statisticians is highly desired by the statisticians at a minimum is becoming known to be the beneficial to all. Figure 4 depicts the states of the "project" that support the collaboration between the Principal Investigator, the Lab Administrator and the Lab Scientists and Statisticians who will be assigned to execute the study's experimental design. Using the Overall Workflow (Figure 3) as a guide, these states represent the activities of collaboration, approval and assignment.

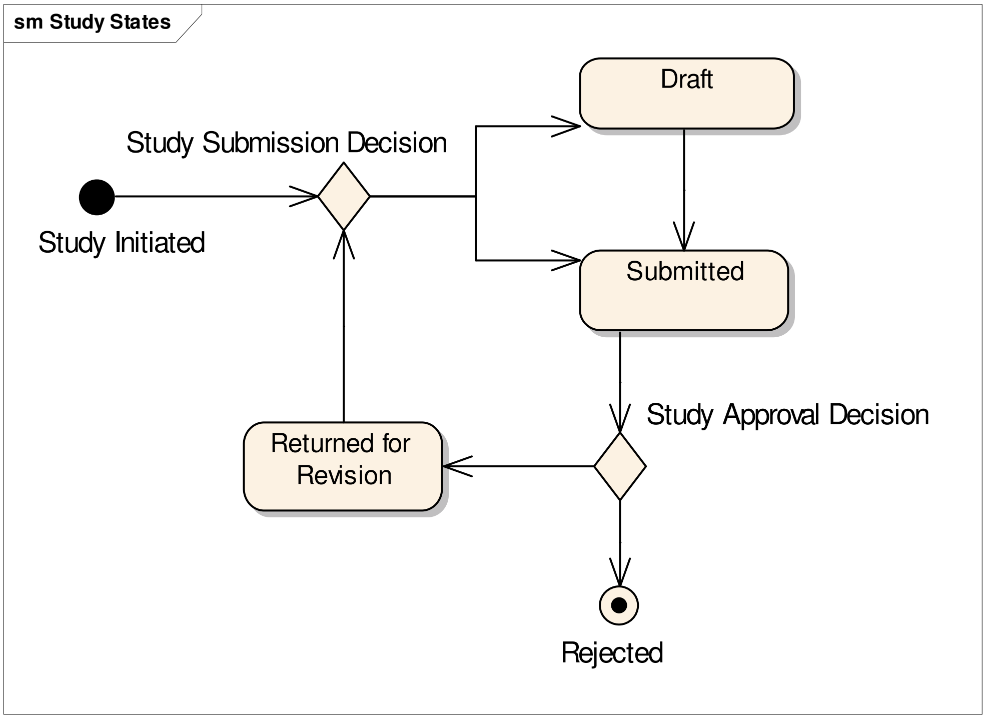


Figure 4. Pre-Approval Project States

3.3.2.2 Lab Workflow post-Acceptance

Acceptance indicates that the experiment is understood and assigned. The first activity is to actually receive the samples so the planning for the execution can begin. The samples may or may not come all at once and depending on the volume of work the lab is performing, the lab scientists may choose to start the experiment prior to receiving all the samples.

There are a significant number of activities to be taken against each sample in the experiment. They are described at a high-level below.

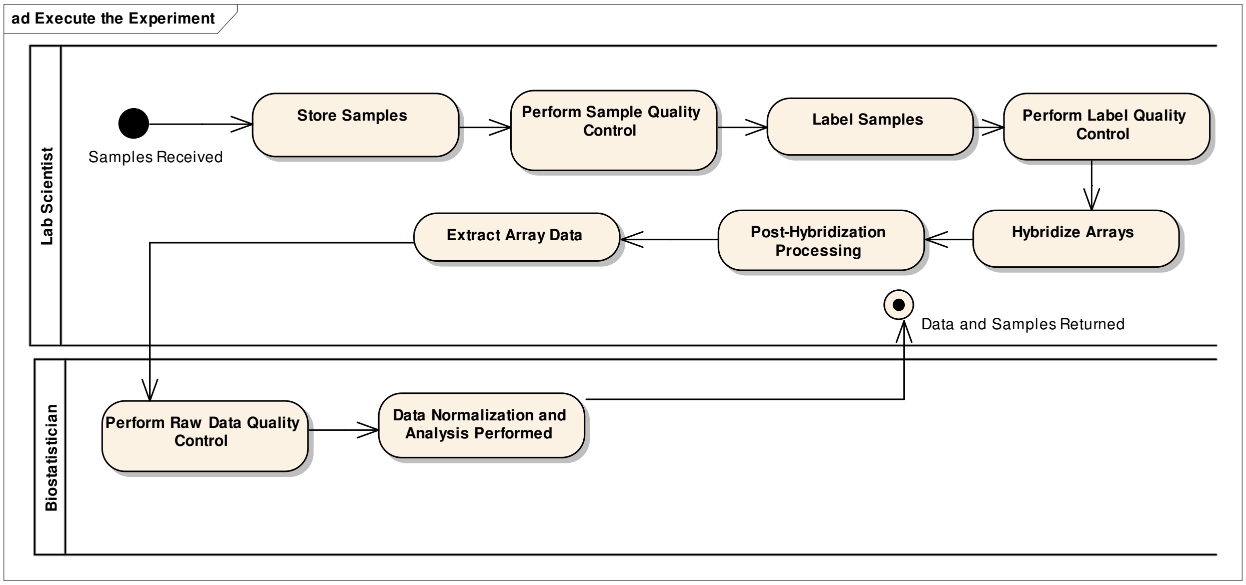

ad Execute the Experiment
Lab Scientist
Samples Received
Store Samples
Perform Sample Quality Control
Label Samples
Perform Label Quality Control
Hybridize Arrays
Post-Hybridization Processing
Extract Array Data
Data and Samples Returned
Biostatistician
Perform Raw Data Quality Control
Data Normalization and Analysis Performed

Figure 5. High-level Lab Workflow

3.3.2.3 Lab Workflow post-production

Once a project's data analysis has been completed, the priority becomes publication. Typically, there is a 6-9 month period between when the work is done and when a paper will be published, though longer time frames are common. In order to support publication, there is often a request for actual data to be made available to the journal considering publication and very commonly (though not absolutely) once a paper is published for the data to be publicly consumable.

As of this writing, there is no requirement across journals for making data available that is MIAME compliant. The MGED Society has a long history in the literature for making the case for MIAME and several significant journals have accepted the suggestion – with Nature being the most influential – but it isn't a fundamental requirement (see http://mged.org/Workgroups/MIAME/journals.html for current journals requiring MIAME). Much of the community has taken the step to submit the hybridization data to GEO but the cancer community widely regards the value of the data available to be limited due to the lack of detailed sample annotation.

The goal for caArray will be to support making the data publicly available as well as to allow for keeping data private.. The basic model for promotion of projects is shown in Figure 6 with the Principal Investigator responsible for assigning the visibility of the project as the "data owner". There are additional considerations to be explored governing what portion of a project is actually available – all of it or just portions – and the overall theme that when a project is made available at the "public access" level, the ability to see, find and download those projects across the grid is also available. This would allow for an anonymous user at the NCICB instance to search for projects of a certain type and find projects at any of the local installations.

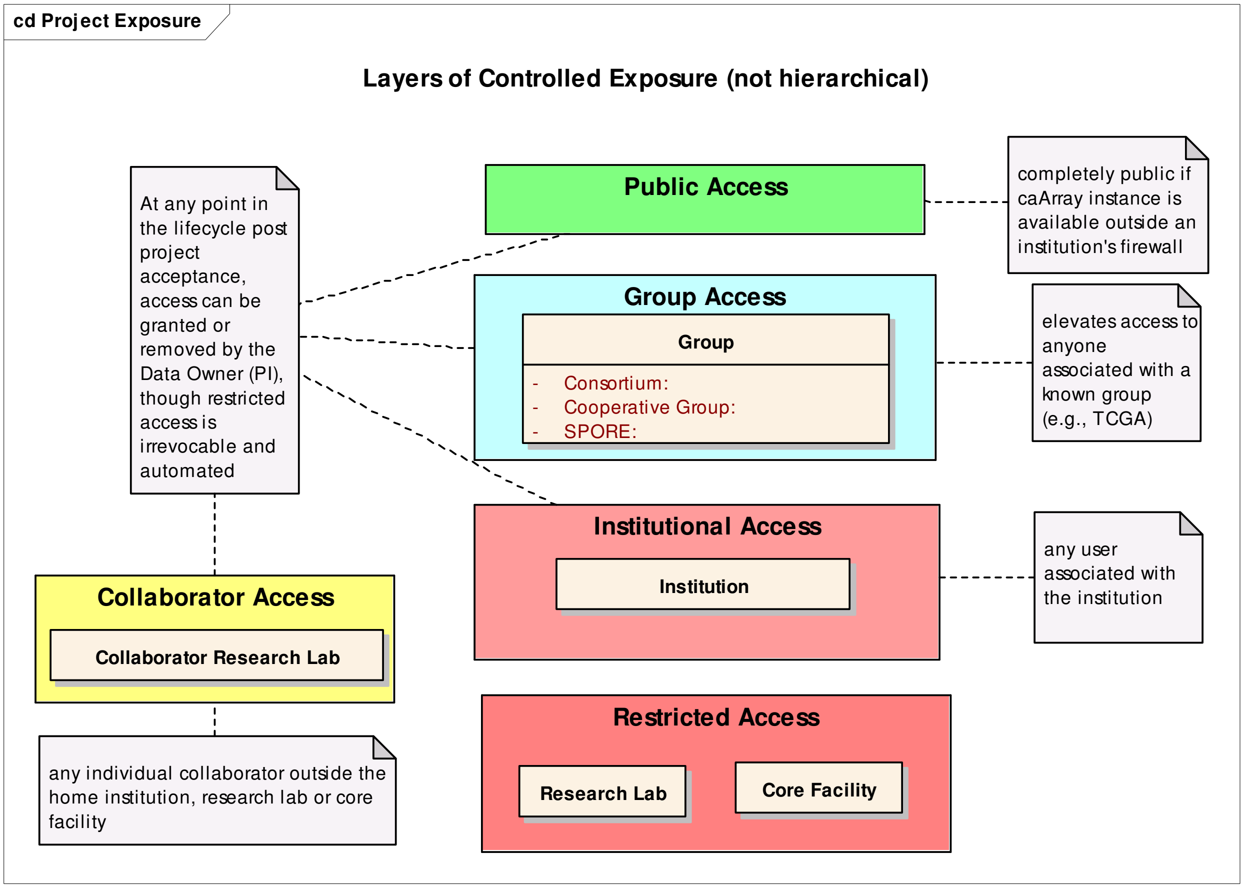

cd Project Exposure
Layers of Controlled Exposure (not hierarchical)
Public Access
completely public if caArray instance is available outside an institution's firewall
At any point in the lifecycle post project acceptance, access can be granted or removed by the Data Owner (PI), though restricted access is irrevocable and automated
Group Access
Group
- Consortium:
- Cooperative Group:
- SPORE:
elevates access to anyone associated with a known group (e.g., TCGA)
Institutional Access
Institution
any user associated with the institution
Collaborator Access
Collaborator Research Lab
any individual collaborator outside the home institution, research lab or core facility
Restricted Access
Research Lab
Core Facility

Figure 6 Graduated Degrees of Public Exposure

3.3.3. Array platforms

The following Array platforms are expected to be supported over the life of caArray. If an array platform is not supported, the expected release will be stated as either “tbd” which indicates that it is not known or “deferred” which indicates that a decision has been made for the release in question. It is also understood that multiple versions of an array design should be supported as well as the community is acutely aware of the cost of the technology and will want to use the last of a particular design even when a newer design is available. In addition, with the method of investigations and the supporting technology moving faster than the adoption, the support of additional manufacturers and experiment types is expected to grow. Table 4 indicates the current set of array methods, their description, providers, data produced known to be in use in the cancer community and our anticipated release for supporting them.

Table 4 Supported Array Types and Providers

Type	Description	Manufacturers	File Types	Approximate Size/Sample	Expected Release
DNA SNP/copy	A Single Nucleotide Poloymorphism is the most frequent	Affymetrix	.dat, .cel, .chp	500MB+	2.0
		Illumina bead	.idat (image),		2.0

Type	Description	Manufacturers	File Types	Approximate Size/Sample	Expected Release
number	type of variation in the genome. These studies look to scour a particular population's propensity for a given disease and potential therapy based on the SNP-linkage analysis	arrays	.csv or .dat (bead manifest), .xml (content)		
		Illumina Golden Gate			2.0
		Agilent	.txt		2.0
DNA aCGH	The ratio of the fluorescence intensity of the tumor to that of the reference DNA is calculated to measure the copy number changes for a particular location in the genome.	Agilent	.txt		TBD
		UCSF Spot			2.0
Gene Expression	Investigation focuses on what genes are expressed in a particular cell type of an organism, at a particular time, under particular condiditions	Affymetrix (including Exon)	.dat, .cel, .chp, .exp, .rpt, .txt, .cab	70MB	2.0
		GenePix	.gpr, .gal		2.0
		ImaGene	.gal, .gpr, .gps		2.0
		Agilent	.shp, .xml		2.0
Methylation	Attempts to establish patterns of methylation genome-wide or within targeted promoters or CpG islands	Illumina			
		Nimblegen	.tiff, gff version2.0		Tbd
Protein/antibody					Tbd
Reverse phase protein lysate					Tbd
RNAi	RNA Interference explores gene silencing affects	Nimblegen			Tbd
miRNA	By measuring activity among the	Ambion+Invitrogen			Tbd

Type	Description	Manufacturers	File Types	Approximate Size/Sample	Expected Release
	217 genes encoding miRNA, patterns of gene activity that can distinguish types of cancers can be discerned.	Exiquon			Tbd
Chip-chip	Map regulatory protein binding sites genome wide for transcription factors, polymerases and histones, or target promoter regions	Nimblegen	Raw data, variety of GFF image files		Tbd
Tissue Microarray	A fundamentally different approach, this method consists of paraffin blocks in which up to 1000 separate tissue cores are assembled in array fashion to allow simultaneous histological analysis.	Custom	Small alpha/numeric values	n/a	Tbd

3.3.4. Common Proprietary analysis tools

Commercial standard analytical tools such as SAS and S+ are expected to be a part of the statistician tool set.

3.3.5. Available caBIG analysis tools

Several tools are currently able to analyze data stored in caArray from local installations. These include:

geWorkbench - http://wiki.c2b2.columbia.edu/workbench/index.php/Home

Developed at the Columbia University, NY is an extendible and flexible desktop tool for microarray data analysis and visualization. The release 1.0 includes several data analysis and visualization functions, including hierarchical clustering, self organizing maps, color mosaic images and biological pathways.

Gene Pattern – http://www.broad.mit.edu/cancer/software/genepattern/

GenePattern puts sophisticated computational methods into the hands of the biomedical research community. A simple application interface gives a broad audience access to a growing repository of analytic tools for genomic data, while an API supports computational biologists. GenePattern is a powerful analysis workflow tool developed to support multidisciplinary genomic research programs and designed to encourage rapid integration of new techniques. It is developed by the Broad Institute.

Bioconductor – http://www.bioconductor.org/

Bioconductor is an open source and open development software project for the analysis and comprehension of genomic data. The project was started in the Fall of 2001. The Bioconductor core team is based primarily at the Fred Hutchinson Cancer Research Center. Other members come from various US and international institutions. Bioconductor is primarily based on the R programming language but we do accept contributions in any programming language.

webGenome – link here

VISDA - https://cabig.nci.nih.gov/tools/VISDA

VISDA (VIsual Statistical Data Analyzer) is an analytical tool for cluster modeling, visualization, and discovery.

3.4. Key Stakeholder or User Needs

The following table describes high level needs elicited from the stakeholder and user representatives above and prioritized by the Product Manager with their assistance. These needs frame the context for deriving features and a basic traceability should exist.

Table 5. Stakeholder and User Needs

Need	Priority	Concerns	Current Solution	Proposed Solutions
Improve ability to upload fully annotated array data in bulk by any owner of an experiment	**High**	Distinctions between arrays don't lend itself immediately to one common interface and underlying compliance requirements (for importing, exporting MAGE-ML) may be problematic	Often experiments are sent to NCICB for uploading. The process is tedious for them as well but the desire is to get the data into	Provide an intuitive interface for organizing the array data and annotation that resembles concepts that are commonly understood across arrays, rather than an unfamiliar and intimidating model meant more for computers than human consumption.
Conveniently add new array types and designs as the field progresses	**High**	The ability to add custom array types and designs may be a significant effort. Commercially available designs seem more prudent. Need to determine whether to distribute the designs or indicate where and how to load them.	A new design is loaded into the database through a backend process.	NCICB will provide new array types and load the designs which can be downloadable by all local installations. This could be a choice – where NCICB informs user of the existence or automated by way of a check caArray performs periodically. In addition, local centers may be able to make public a type and a design if they are first to use it.

Need	Priority	Concerns	Current Solution	Proposed Solutions
Dramatically decrease the turnaround time for releases of new features and bug fixes	**High**	The matrix of teams, management and processes and lack of automation. Balancing the needs of functional testing and scientific/quality control	While thorough, manual, redundant effort is employed throughout the process and is prone to inadvertent error. Months go by between the release of candidates to their eventual promotion to production	Automate wherever prudent and possible that supports the quality and timeline of delivery. Configuration and Testing are the likely candidates for automation.
Improve turnaround time on acquiring user accounts	**High**	Distinction might need to be made for acquiring NCICB accounts versus local accounts	Current process is up to 72 hours; need to ensure that at NCICB all users are known	Register through the application web portal and automate requests to NCICB
Secure transactions and data exchange	**High**	Applet signature has been problematic and it has not been tested outside of NCICB	SSL has been introduced as an option for general browsing in addition to the upload applet being made secure	Determine if an applet is required and ensure extremely clear instructions on how to install the certificate to allow for full coverage of the server and the applet
Need for an accession code or URI to unambiguously identify an experiment	**High**	Needs to be recognized as acceptable for being publicly available by MGED and scientific journals	For caArray, long URL pointing to Experiment	A simple, human understandable study identifier consisting of a concatenated description of the study (studytype_platform_pilastname_number)
caArray MUST be able to easily pass the data to analytical tools to get usable results out.	**High**	Formats and other constraints of the analytical tools need to be established up front and be careful implemented. Backward compatibility is also a concern	Different tools are running different methods and even caArray versions to access caArray data	Establish THE method for accessing caArray data that removes the need for re-implementation on the part of tools that use it
A structure that supports MIAME-compliant experimental annotations in a user-firendly manner	**High**	Users dislike adding the amount of data necessary to ensure MIAME compliance, potential cause for retrenchment of projects; application that forces user to input MIAME is not designed yet	Some solutions such as maxdLoad from EBI offer a Java app that forces the user to input all MIAME fields. ArrayAnalyzer from Insightful does not allow analysis to continue until the user inputs all MIAME fields	Ensure MIAME compliance through the use of an adaptive, intelligent, flexible tool that enforces data compliance, makes uploading less error-prone, and provides the user with enough feedback to make the process painless BUT NOT required.
Link array data to clinical sample Attributes	**High**	High variability in the quality, accessibility and format of the data	Hand entry without restriction	Design an attribute set that is suggested, but not required, for any sample

Need	Priority	Concerns	Current Solution	Proposed Solutions
Balance the need for privacy of project data with the need for expansive collaboration	**High**	Need to be representative of diverse methods of institutional organization and ensure that the PI's know what data is being exposed	Organization of public versus private experiment by way of group assignments	Support data ownership by Principal Investigator and have granular, ever-expanding levels of public exposure
Support multiple types of projects including those that have already done most of the experiment outside of the Array lab and just bring the lab chips to be analyzed	**High**	Determining how to incorporate and validate the data they do have and degree of difficulty vs. value of working to allow this "pick your service" portion of the workflow	Various – from paper to home grown LIMS	Don't support the service variations or do support it but make it a configurable items that would allow for Core labs to set up service arrangements that would roughly increase or decrease the amount of data to be entered.
Store ancillary information such as images	**High**	Increases the amount of data to be stored, not particularly valuable outside of completeness of the data produced by the experiments	Upload of individual files (i.e., .dat)	These files can be stored without parsing so they are available for downloading
Be informed of state of the project's execution	**Medium**	Workflow may not be common across labs or techniques. This is a new feature that should be validated.	A balance between looking at a lab notebook and manually inspecting a series of web pages related to an experiment	Genericize a workflow that can be extended. Worth investigating whether the creation of an experiment should begin with the type of experiment it is which would drive the unique thread(s) of the otherwise generic workflow
Scientists require a series of points in the process where array annotation data can be downloaded at the current state.	**Medium**	May need to support multiple formats to make this effective for importing into other systems (MAGE-ML, comma-delimited txt files)	No ability in caArray - ranges from printing out data and re-entering it into additional software packages, placing data on ftp sites or burning cds	Allow for the project sample structure to be downloaded (as placeholders for the deposition of array data); allow for the extraction and download of project structure and annotation in the format of the user's choice.
Store analysis results	**Medium**	annotation of the analysis factors used needs to be provided	Desktops, LAN's, FTP sites and the "other" category of current caArray applet upload	Enable the annotation of analysis parameters and offer the ability to upload files on a per-parameter set basis. Potentially set a size limit on a "project/study" based on the platform and number of samples to control size and don't allow versioning of the data

Need	Priority	Concerns	Current Solution	Proposed Solutions
There is currently no way to enforce the integrity of data in caArray	**Medium**	Corrupt data can be loaded into caArray, compromising the overall integrity of the database	None	"Quarantine" data when it is submitted to caArray. A curator must review and approve the data before it is persisted.
Document Quality Control measures on a sample by sample basis to support data analysis and reuse	**Medium**	Will need a UI to support QC annotations for small and large numbers of samples in a given project	Not done, by hand in a lab notebook, excel	Provide features for sorting, global selections, all
Store Data in the appropriate application that corresponds to the kind of data it is (e.g., tissue data belongs in caTissue)	**Medium**	Applications are in various stages of maturity with a high degree of divergence	caBIG is the attempt	When possible and practical, look to access vs acquire data and harmonize the data elements at key touchpoints
Make Analysis Accessible to a larger portion of the user community	**Medium**	A little knowledge can be dangerous and cause less productive discussions than a show and tell performed by the biostatistician	µAD-B, commercial products for simple analysis	Increase the ability to organize and format the data to be exposed to the analysis tools and seek to transmit just the data necessary to increase speed. Potentially integrate less sophisticated analysis tools directly into caArray.
Ability to maintain a high quality data repository at NCICB	**Medium**	A significant amount of compromised data is submitted to NCICB	No solution exists	Quarantine data upon initial upload and inspect through curation before making it available
Access Current Gene Annotation and understand the difference between when the project's data was produced and the time it is being accessed	**Low**	Issues surrounding versioning, storing the gene annotation in multiple places, usability burdens on representing the distinct annotations	No solution exists	Store the gene annotation ids from caBIO

3.5. Alternatives and Competition

Since the initial deployment of caArray in January 2003, several applications have extended their reach and capabilities serving a very similar, if not the same audience. Tbale 7 outlines how caArray will be positioned relative to its non-commercial competitors.

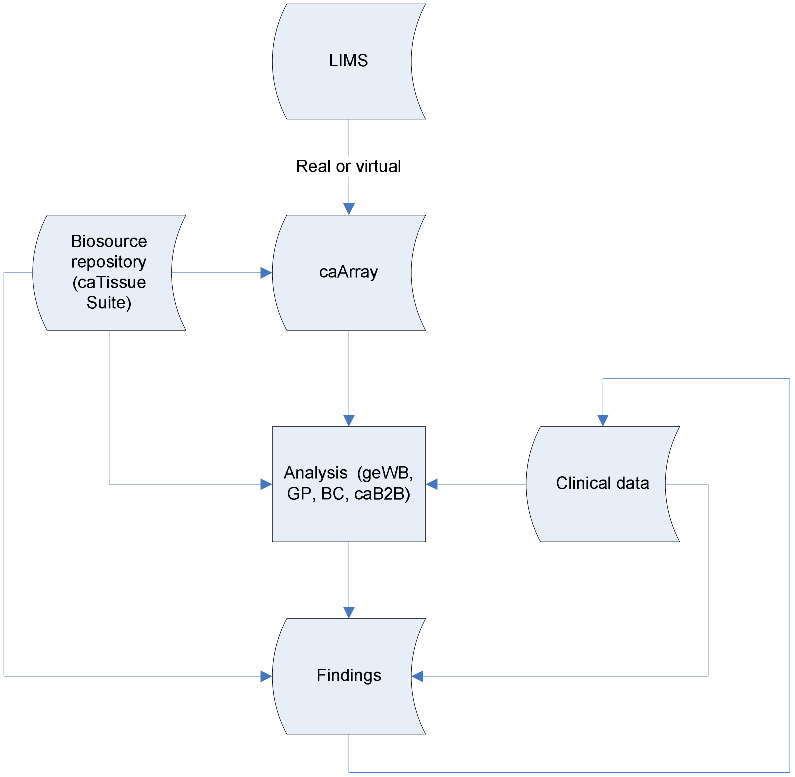
Table 6 Alternate Solutions

Alternative	Why caArray	What can we benefit from that they do well
Gene Expression Omnibus (GEO): http://www.ncbi.nlm.nih.gov/geo/	Initially, quality over quantity. While the amount of data in GEO is unsurpassed, the ability to find quality data (including the attribution needed to determine its value) is difficult. Next, is the transparent access to analysis tools and retrieval of meaningful results.	Easy interface to upload data leading to their current status as the defacto repository for microarray data. Provides unique but short accession numbers for each sample which is used to prove data has been placed in a public repository for publication purposes.
ArrayExpress: http://www.ebi.ac.uk/arrayexpress/	caArray is open-source and caBIG-compliant.	Simple query interface against a small number of concepts on the home page, RSS feeds, indications of how many experiments are available, can query against a gene name
Stanford Microarray Database (SMD: http://genome-www5.stanford.edu//)	Not an open-development, open source project, not caBIG compliant	Statistical analysis available directly in the systems; storage of gene lists, query by gene

4. Product Overview

caArray is associated with the Integrative Cancer Research Workspace (ICR) that falls under the caBIG™ umbrella.

4.1. Product Perspective


LIMS
Real or virtual
Biosource repository (caTissue Suite)
caArray
Analysis (geWB, GP, BC, caB2B)
Clinical data
Findings

Conceptual positioning of caArray in integrative cancer research

4.2. Assumptions and Dependencies

4.2.1. Only terms from Managed Ontology systems will be available

Whenever a controlled vocabulary is appropriate for user-based entry, the system will allow for the use of terms from the NCI Enterprise Vocabulary Service. If terms are not found, then the data can be saved but will be placed in the queue of the installation's curator for submission to the EVS.

4.3. Cost and Pricing

There is no cost to the user community in terms of purchasing a license for caArray. However, there is a cost in terms of time and energy spent in supporting the product and its evolution, the server(s) and potentially the use of commercial tools such as Oracle to replace the open source database of choice - MySQL.

4.4. Licensing and Installation

Following the principles of caBIG of Federation, Open-Development, Open-Access, and Open Source, the source code is available to view, alter, and redistribute however a consumer chooses to do so. The license agreement is available from the NCICB download center: http://ncicb.nci.nih.gov/download/caarraylicenseagreement.jsp.

5. Product Features

With priority set by the Product Manager, with input from all stakeholders, the following features represent the functional aspects of the caArray. All features are not created equal, particularly when having to deal with constraints such as time, budget and level of effort. Ranking requirements by their relative benefit to the end user opens a dialogue with customers, analysts and members of the development team. This feature set and the relative priority will be used to manage scope and determining development priority.

Table 7. Feature Priority Attributes

High	Essential features. Failure to implement means the system will not meet customer needs. All critical features must be implemented in the release or the schedule will slip.
Medium	Features important to the effectiveness and efficiency of the system for most applications. The functionality cannot be easily provided in some other way. Lack of inclusion of a required feature may affect customer or user satisfaction, but release will not be delayed due to lack of any required feature.
Low	Useful features that are anticipated to be used less frequently, or for which reasonably efficient workarounds can be achieved. No significant customer satisfaction impact can be expected if such an item is not included in a release.
Deferred	Future and nice to have features.

5.1. Project Management

5.1.1. Propose a Project

To support the collaboration between a Principal Investigator and the lab that serves them, the ability to propose a project that will contain a set of required, suggested (e.g., strongly encouraged), and truly optional annotation data that describes in sufficient detail the experiment desired, the platform to be used and the samples to be sent.

5.1.1.1 Save as Draft

The ability to save the work entered during a given session by an investigator as long as a minimum set of information has been entered and is unique enough to be stored discreetly.

5.1.2. Review a Project

Once a project has been submitted, the ability to take action and communicate with the submitting investigator is expected. The action can take three forms:

Accept a Project – moves the project forward and includes the ability to establish a price and assign staff (scientists and biostatisticians)

Reject a Project – an irrevocable action that places the project back in the queue of the submitter, who may then choose to hide the rejected project.

Return a Project for Revision – sends the project back in the investigator's queue

5.1.2.1 Inform Project Submitter of Activity

In addition, the ability to enter, automatically transmit via email from the application and retain project notes submitted should be easily navigable from the project's "home page. If a researcher acts on behalf of a Principal Investigator, both users will be informed.

5.1.2.2 Allow Principal Investigators or their designees to submit Projects

Any user with a Principal Investigator role will be able to submit projects and become the logical "owner" of the project. The Principal Investigator will also be able to associate users to their research lab to allow other users to submit projects on their behalf.

5.1.2.3 Restrict Project Submission based on Required Project Annotation

The ability to submit a proposed project must be restricted by requiring the following set of information to be entered using an interactive web form by a Principal Investigator or their designee:

The **Principal Investigator** and the contact information

A **Project Synopsis** that details the high level aspects of the project that will drive succeeding data entry and/or upload options. This will include the kind of project it is (genotyping, gene expression, etc.) array platform of choice, the source of funding, the species, organ/tissue type, organ region, cell type, disease or condition being studied, the number of replicates, whether pooled samples are being used, the type of samples being submitted (e.g., total RNA), and the total number of samples expected.

The **Experimental Design** that will include the experiment type (time series, pharmacogenomic, subclassification, etc.), the relevance of the study, hypothesis, specific aim, experimental procedures and factors.

The **sources** which will be a derived from the synopsis but with additional characteristics that will correspond to the type of project it is (e.g., human will require sex). Common properties would be a name and description, strain, developmental stage, genetic variation, genotype or mutation status, and age.

The **Samples**, which will also be derived from the synopsis but will require a bio-source and additional characteristics such as a name and description, the amount (volume or weight), and any treatment information.

Additional information that covers extractions and array designs would be made available (to support those investigators who will perform most activity in their own lab and use the Core Facility to run the labeled extracts or just do analysis) but not required.

5.1.2.4 Copy a Previous Project's Annotation

The ability for an Investigator to select one of their projects for copying into a new project will be made available. The copy feature will only apply to the annotation for the investigator, project synopsis and experimental design.

5.1.3. Transfer Project Ownership

Projects and the objects that make them up in caArray are owned by individual users. To prevent a situation where a user is no longer active, the current owner or a system administrator can transfer the ownership to an active member.

5.1.4. Cascading Control of Project or Sample Visibility

The ability to control the public visibility of a project must be controlled by the Principal Investigator or their designee. The actions can be taken against a project as a whole (with a cascading affect down all project elements) or an individual or set of samples within a project (with no cascading affect). Depending on the organization of an Institution, the ability to set the visibility to all within the institution, a self created group of collaborators, a known group (such as a SPORE) or completely public (allowing for anonymous browsing) should be available.

5.1.5. Control Project or Sample Permissions

The ability to control the permissions at a project or sample level is expected. Similar to the visibility feature – which makes the project or samples read-only, this feature will allow for read, write and no permissions to be extended to labs, institutions, groups or the public. Editing will not be allowed when making a project or samples public at the highest level. Otherwise, all permissions are allowable.

5.1.6. Provide an Legible and Accessible Project name and URL

Each project, as soon as it is saved as a draft, must have an automated name and reflective URL associated to the project. A potential pattern for name generation is as follows:

Principal Investigator Last Name-Platform-Species-Autonumber which will be truncated to the first 5 characters of each particular attribute.

An example would be: klemm-affym-human-89765. The name produced would be directly represented in the url. No matter the state of the project's visibility, if a user has permission (unless it's public – where permission is unecessary) the url can be directly entered into a web browser and the project pulled up. This is particularly useful when trying to indicate where the data is and may suffice for publication if the installation is outside of the firewall. The brevity and legibility will also help to discern what the project is, while at the same time, the auto-generation of the number helps offset concerns of hacking the url.

5.1.7. Support Multiple Types of Array Services

Across the community, there are three main kinds of services requested and are categorized as complete, partial, or analysis services. The challenge with the latter two is to ensure that the requisite annotations are properly acquired and uploaded into caArray.

5.1.7.1 Complete Service

This service supports the complete set of activities from the receipt of source material (samples) where the lab prepares the RNA, runs the experiment and performs the data analysis

5.1.7.2 Partial Service

This service allows for the submission of labeled extracts to be run through hybridization and data analysis

5.1.7.3 Analysis Service

This service is invoked when the requesting researcher has already performed all the effort to produce the raw data and just wants to have the data analyzed.

5.1.8. Associate Publications to Projects

The ability to create, edit and delete publications, indicate their state (published, in press, in preparation) and associate them to specific projects is desired. Additional publication types – abstracts, talks and reference materials – would also be available. The standard format for citations would be followed with the ability to create hyperlinks to the paper and open the project from a listing of publications as well as open the publication from the project. In addition, the ability to auto-populate citation information (volume, page number, etc.) based on PubMed ID would be available.

5.2. Array Annotation and Data Management

These features describe the essential functions required to populate and maintain array project annotation and data – an essential and primary purpose for the system. To support finding data of interest for any user, the ability to annotate the data clearly and consistently is required. Any descriptive project elements that support and/or identify arrays are considered "annotations." "Arrays" are the data produced by the experiment. Annotations, therefore, are used to describe the following:

Project

Person

Organization

Protocol

Array Design

Source

Sample

Experiment

Details of these elements will be found in the Domain Model in the Use Case Summary and the Data Model associated with the Software Architecture Document.

5.2.1. Source and Sample Annotation

5.2.1.1 Acquire Source Annotation from External Source

The general ability to acquire tissue annotation from an external source is desired. Whether through navigation of caTissue directly in caArray or The minimum set of annotation to support MIAME will be exposed so that the data pulled in will conform to the standard and allow for expedient searching of caArray as a portion of the data will be moved. Non-caTissue repositories that contain the information could also be used – though their integration is not directly supported by caArray through the user interface.

5.2.1.2 Allow for an Institutional Id to be entered

When annotating a source or sample, allow for an institutional id to be entered to support relationships back to de-identified clinical data. While a unique id is expected to be generated by caArray, this additional id allows for the opportunity to connect to other repositories of additional information that local users would have access to – directly or indirectly from caArray.

5.2.1.3 Allow for Clinical Annotation to be entered

Although there is a set minimum of information to be entered, the system should support the addition of clinical annotation such as treatments and outcome data to the desired degree on the part of the investigator. In addition, the ability to do so programmatically is also desired, though any editing of data consumed by caArray is uni-directional and not expected to update the other system. This would include details such as the removal of the stated amount of the sample from the source in the institutional tissue bank.

5.2.1.4 Share Controls across Projects

There are occasions where a control source or sample will be used for multiple experiments. Therefore, the ability to "re-use" a control should be available across projects.

5.2.1.5 Generate Bio-Source and Sample Placeholders

When a project is created, the users will have the option to enter the number of sources and samples expected to be in the study and have the system produce the requisite number of placeholders for data entry. The characteristics of each will be predicated on the high level project depictions (experiment type, species, tissue type) and will annotate the name field of each source or sample and distinguishing one from another by incrementing the number (e.g., source1, source2).

5.2.2. Annotate Projects Interactively Using Web Forms

The most common method for project annotation is expected to be through a web-based series of forms. In addition to the project definitions, the ability to manage the project's bio-source, samples, extracts and array designs is expected. While the data characteristics are dependent on the element in question, each significant element should be enterable through a form.

5.2.2.1 Evaluate MIAME Compliance Per Form

When entering data in any form or reviewing imported or uploaded data annotations, the system will provide an option to evaluate the compliance of the data that has been entered. If the evaluation fails, the indication of why will include:

- The element(s) of the MIAME guidelines that produced the failure
- A link to the full MIAME description
- And Each instance of what is wrong

However, the system will not enforce the user to fix the error. Rather, this is simply informational to help users understand what they need to do in order to fairly state that project is supportive of MIAME compliance. Further, there are some variations of MIAME which should be adopted for particular project types: CGH and ChIP-on-chip. Finally, Illumina has offered an extension to the MIAME guidelines which they would like to see adopted - http://mged.org/Workgroups/MIAME/Illumina_MIAME.pdf.

5.2.2.2 Evaluate MIAME Compliance Per Project

When viewing a project's initial view, the ability to see at a glance what elements of the project are in compliance with the MIAME guidelines is required. The summary should include:

- Array Design Details
- Experiment Details
- Sample Detail
- Hybridization Detail

- Measurement Detail

A basic indication of whether it is valid or not should be exposed and a link to the exact cause, a full description and each instance of error should be provided.

5.2.2.3 Support Extension of MIAME as the guidelines evolve

The ability to seamlessly introduce new MIAME elements and wholesale additions or concepts to the MIAME guidelines should be supported. As the field expands and new technologies mature, the ability to add new validation routines should be transparent.

5.2.3. Import Array Annotations and Data

Users will be able to submit array data by importing data files in formats known to caArray. Annotations associated with this data will be submitted though file-based methods or through a form-based graphical user interface as described above. The submission process can either be incremental (e.g., a file at a time) or a complete set of data.

5.2.3.1 Import existing array data on a sample by sample basis

The ability to select one or more pre-existing data files located on any user's local hard-drive, local access network or locally available media (e.g., a CD, DVD, or external hard drive) for system-specified array designs needs be supported. The array data will correlate to a pre-existing sample available in caArray. The array design plus the version of the design needs to be selected in order for this to succeed.

5.2.3.2 Import existing array data across samples

The ability to select a **set** of one or more pre-existing files located on any user's local hard-drive, local access network or locally available media (e.g., a CD, DVD, or external hard drive) for system-specified array designs needs be supported. The array data will correlate to existing samples and the ability to choose which pre-existing files go with each sample must be available. The array design plus the version of the design needs to be selected in order for this to succeed.

5.2.3.3 Import existing array data without existing samples

The ability to upload a set or a single array data file into a project without pre-existing sample definitions needs to be available. The system will automatically create sample definitions or "placeholders" based on the names of the files, however, the source of each sample may need to be manually entered and the sample array data uploaded will need to be associated to the source.

5.2.3.4 Import existing annotations from files

Array annotation is available in a variety of independent files. MAGE-TAB is one source of annotation that is gaining popularity. Support for MAGE-TAB files is under consideration. In addition, other file formats contain annotations, particularly from commercial vendors (e.g., Affymetrix's .chp file). The list of supported annotation files are found in the Supplemental Requirements: Annotation Files.

5.2.3.5 Upload Supplemental and Private Documents

In addition to annotations and data, the ability to manage (e.g., add, replace, delete) documents such as spreadsheets, images, and word documents should be available. These documents would be attached at the project level and could be made public or private (only available to a lab, institution, group or collaborator). The initial assumption is that they are private to the Lab and the PI.

5.2.3.6 Import array annotation and array data from other repositories

The ability to select complete project annotation and data from other repositories is desired. The initial target is the EBI's Array Express whose data is considered of high quality.

5.2.3.7 Establish Integration with other repositories

The ability to exchange data with other repositories, particularly EBI's ArrayExpress, is desired. This would allow the automatic transmission of data from one repository to another. An alternative, also deferred, approach, would be to allow the searching of one repository's data from another.

5.2.4. Validate Data

The ability to validate the format and completeness (not the scientific contents) of the array files being imported is required. The expected formats vary per manufacturer (e.g., Affymetrix vs. Illumina) or standard (e.g., MAGE-ML, MAGE-TAB). In order to validate, the user must select the array design that the file corresponds to with the design and the system should indicate if the file or set of files meet the expectations of the platform and design.

5.2.5. Parse Data

The ability to parse particular data files that contain values of interest – for example, expression levels on a per gene basis – is required to support granular searching of data and to serve out data in a platform-independent format. Which file and what values are dependent on the experiment type and manufacturer. The relationship between manufacturer, data file and values will be described in the Supplemental Specifications: Data Parsing and the actual data model.

5.2.6. Maintain Annotations and Array Data

Maintenance is defined as the ability to edit, replace and delete annotations or data as distinguished from Management, which explicitly allows for the ability to create in addition to the actions of maintenance.

5.2.6.1 Edit Annotations

For any annotation element in a project, the ability to edit those annotations by the data owner or those associated with the data owner is required. While the Principal Investigator is the ultimate data owner, the elements of the execution workflow that are directed to distinct roles will allow those assigned to the task to edit the annotations.

5.2.6.2 Restrict Project editing once a Paper has been published

A project will become read only once a manuscript has been published and associated with the project. Additions can be made, such as new samples or uploading of additional supplemental files.

5.2.6.3 Replace Array Data

The ability to change array data by importing/uploading a new file in place of an existing file is required. Versioning of the files, however, is not expected or required.

5.2.6.4 Delete Annotations and Array Data

Any unassociated annotation can be deleted. This would allow for samples not associated with array data, for example, to be removed, or to delete array data files if uploaded in error.

5.2.6.5 Copy, Edit and Delete Project Annotations at a Glance

Entering source and particularly sample annotations are often time consuming. To offset that cumbersome activity, the ability to enter one and copy it as many times as desired should be supported. This should be allowed to be done for any one significant entity – source, sample, extracts and arrays. In addition, any one of the entities could be edited from the initial page and provide for more extensive editing in a separate page if necessary. Finally, the ability to delete an entity should be efficiently done directly on screen, rather than having to drill down several layers to get to the element desired.

5.3. Array Design Management

5.3.1. Pre-load Array Designs

At the time of any release, the ability to provide array designs pre-loaded and annotated into the system is desired. The list of array designs supported is maintained in the Supplementary Specifications: Array Designs and will be visible within the application itself. A validation that the array design doesn't already exist in a local installation's library will need to be present in the upgrade script.

5.3.2. Import Array Design

For any array design that isn't already available, the ability to add an array design to the system is required. Once the design has been uploaded, the ability to load data files corresponding to the design will be available and allowed to be parsed.

5.4. Extraction

One of the critical values that caArray must bring to the community is the ability to find and use the content of the repository to serve multiple, meaningful functions.

5.4.1. Export Project to Standard Formats

Export of project data to standard formats will be supported - current candidates include MAGE-ML, MAGE-TAB, and additional standards that become recognized in the community. The standards decided upon will be explicitly added in the Supplementary Specification: Export Formats.

5.4.2. Submit Project to GEO or ArrayExpress

The ability to select a project and choose the repository for data submission is required. When the project is intended for GEO, the system needs to create an appropriate SOFT file for the user to upload to GEO. Array Express can consume a MAGE-ML file so that ability to produce a MAGE-ML File representing an entire experiment is also expected.

5.4.3. Retrieve Complete Data Matrix as a comma or tag delimited file

Essentially the desire is to acquire a spreadsheet of data from the repository that represents the expression data for a project (rows=reporter values; columns=samples). This would include references to the raw file via a system-generated URL.

5.4.4. Retrieve Partial Data Matrix as a file

There is occasion for the user community to only want to select a subset of the data for a given project(s). Tied to the general Search features, this specifically refers to the ability to:

- select from a list of arrays (with a “select all” or “select some” option)
- selecting the interesting quantitation type for each array; example – signal, S/N, probeid
- Searching for a sample of interest

5.4.5. Retrieve data in native (manufacturer) format

The ability to find and download all raw project data or a subset of available array data is required. The search mechanism would allow for common selection methods that support the selection of all items found or the de-selection of particular items in the list and the choice of data format to be downloaded by file extension.

5.4.6. Retrieve annotations and array data through an API

A key feature of caArray is the ability to access data through a caBIG-compliant application programming interface. The existing caArray MAGE-OM API provides this capability but is highly complex and requires users to have an intimate knowledge of the MAGE model. It is a requirement that a less complex API be provided for access to data within caArray. For the NCICB instance of caArray, this API must be grid-enabled. The requirement for also maintaining a MAGE-OM API is TBD.

5.5. System Administration

5.5.1. Add a User

The ability to add a unique user, assign a role or multiple roles, and the institution they belong to should be made available to the System Administrator. Common validation checks should be performed to thwart potential duplicates being introduced.

5.5.2. Edit a User

An individual user or a system administrator should be allowed to edit their demographic characteristics. If a user changes Institutions and still has active projects, the System Administrator can decide if the user can continue to have access to the local installation.

5.5.3. Delete user

The System Administrator can delete users who have not been associated with a project. The user is completely removed and will no longer have any restricted access priveleges.

5.5.4. Disable a User

The System Administrator, Lab Administrator or Principal Investigator can disable a user’s privileges. They can reinstate the user’s privileges at a later time but when disabled, the user is no longer able to neither view restricted portions of the site nor be assigned to workflow tasks.

5.5.5. Manage Collaboration Groups

Any registered user can create, edit, or delete groups outside or within their home institution to allow for other registered users to participate with projects that are otherwise restricted to their lab, institution, or group.

5.5.6. Provide Configurable, System Generated Email

Using a set of templates, allow for the customization of generic emails at given points in the process. This would include a welcome message for newly registered users, the project approval decisions, and indications when a project has moved from additional state progressions.

5.5.7. Provide for Customization of Installations

The ability to easily indicate the location of a particular caArray installation by adding a text or graphic logo is desired. In addition, the ability to add html pages into the framework of caArray to support custom instructions and other information should be possible.

5.6. Curate Annotation

5.6.1. Edit Ontological Values through a Single View

The ability to view the lists of common terms and definitions, from protocols to file types to array designs from a single page and be able to edit or add to the values from a single view should be supported. While the NCI's Enterprise Vocabulary System is the vocabulary authority and will provide the categorical definition, the values or details of a particular definition will be different from lab to lab. To curate the data definitions, the ability to edit an entry performed by a user and have the update cascade throughout the system is required.

5.6.2. Edit Ontological Values during Annotation of a Project

In addition to the single view held by a curator, the ability to select controlled vocabulary terms and definitions should be supported. If a user edits the details of a term, the ability to save it and have it cascade throughout the local instance is required.

5.6.3. Add Vocabulary Term

The ability to add an entry to the vocabulary at any local instance should be supported though the process for getting it accepted in EVS is distinct. This feature will support timely execution of a project but it should be an infrequent event where a new term is necessary. All new terms will need to have a category, value and description, potentially specific properties and database data.

5.6.4. Manage Terms Added by Users

Across the local installation, the ability to see any new terms that have been entered should be available to support the acceptance, rejection or re-assignment of those terms along with the ability to mark the action taken. For new terms that fall outside of the EVS, the ability to submit the terms to the EVS for approval should be supported.

5.7. Navigation and Search

5.7.1. Inventory of Summary Repository Statistics

The ability to see from the home page of caArray, a dashboard-like summary of overall statistics for any installation is required. This would be broken out by a local installation and all installations available on the Grid, allowing the user to switch between views. The "home" institution will be shown first, with the ability to switch to any installation that is available across the grid. This would support quick navigation to projects by project state, platforms, experiment types, species and publications by choosing the numeric link corresponding to the item of interest which would produce a list of respective projects. Graphical representation of summary data is also desired. All data should be represented, though only public data can be searched. This is the only place in the application that goes against the principle of showing only what a given user has rights to see.

5.7.2. Search Public Data

In addition to the inventory summary, any project data that has been made public should have a finer-grained search capability. High level searches for projects, hardware, software, arrays, array designs, sources, samples, and labeled extracts would be available on a per installation basis, and have content appropriate values related to the type of search to be executed. In addition, the search should lead the user to results by only showing data attributes that will produce a result as much as possible.

5.7.2.1 Anonymous Browsing

The ability to search public data should be available without requiring the individual user or external application to have a registered account.

5.7.3. Search Restricted Data

The same search mechanism that applies to public data will also apply to project data that has restricted visibility Only the project data that a user has access to will be made available, following the general principal of show users what they have rights to, not what they don't.

5.7.4. Search Categories and Result Actions

The ability to query the repository for data one has access to can occur in the following manner:

- Natural text search against a discrete set of commonly referenced annotation values. (NOTE: analogous the iTunes search.)
- May also use appropriate ontology categories (where appropriate) to support searches by:
  - Disease
  - Tissue type
  - Species
  - Array design
  - Investigator

- Submission date
- “Accession number” of a project
- Identifiers of sources, samples, array design, investigators, species

All search results should allow user to export associated data or other actions corresponding to the search executed. For example, a search against a disease type will produce a set of projects referencing the disease type. The ability to download all or some of the projects data should be available as well as independent inspection of any given project selected.

5.7.5. Query Genes Across the Repository

Once the data is stored and annotated appropriately, the ability to query a single gene across common tissues or treatments within one’s own project or against available projects. The intent would be to retrieve projects or samples that contain arrays that contain information about the gene or genes of interest given a particular value.

5.7.6. Actionable Search Results

For any search performed, the ability to perform actions in context of the search should be available. These will range from links to more information, downloading or exporting of the results, saving the result or portions of the result to virtual projects or other organizing mechanisms.

5.7.7. Project Workspace

For those users involved in any aspect of the workflow of a project, a workspace that shows all projects they are associated with to be immediately available (a two-click maximum. Whether they propose the project as a Principal Investigator, or are assigned work by a Lab Administrator, the project will be placed in their queue with the state of the project exposed and the respective actions available. Once a project is public, it is removed from their queue but still available to the user. This feature is provided automatically based on the role the user has and can not be altered by the user.

5.7.8. User-Directed Organization of Repository Information

All registered users should have the ability to place information they care to access frequently and easily into logical folders of their choosing. For example, a folder can represent a category of its contents (for example, “My Queries”, “My Virtual Projects”). Additionally, folders can contain other folders (leading to a hierarchical organization). It is assumed that putting something in a folder does not physically copy the item, but instead simply refers to it. This will permit inexpensive organization and also reduce fears of mistakenly deleting something of value.

5.7.8.1 Default Organization

The default organization of ‘folders” should support a project workspace, virtual projects, and independent organization of favorite samples, queries and genes.

5.7.8.2 Manage Virtual Projects

The ability to create, edit or delete a “virtual” project that will allow for sources and samples to be pulled from real project data should be supported. The population of the project will likely come from search results, but may also be derived from actions taken on an open project, where, for example, a particular sample is of interest to the user. A limited set of actions available to a real project will be available to the user – primarily the ability to download the data or send to the analysis tools.

5.7.8.3 Save Queries

The ability for any registered user who creates a query with at least one attribute/parameter selected and/or entered to save the query for future use. The query must be named and within a user's set of queries, be unique.

5.7.8.4 Update Queries

Supports the ability to select a user's saved queries, change the parameters and save it again for future use. There is no expectation of versioning of queries though a user may choose to save a query with a new name and the system should show when the query of interest was last run.

5.7.8.5 Re-Run Queries and represent what's new

The user would be able to choose to run a saved query and indicate that the results of interest should be restricted to only those that have appeared since the last time the query was run by the user (or not).

5.7.8.6 Share Queries

The ability to expose a query to another user and/or set of users is desired. Some of the queries may take significant time to develop/curate, therefore this feature would help share the results of that burden. It is TBD where there should be ownership of the query or referential integrity.

5.7.8.7 Paginate Query Results

For result sets that span 25 or more items, the ability to show the total number of results and paginate the results in 25 entity increments should be supported. In addition, the ability to navigate to any one page of results should be available.

6. Constraints

Since caArray is one of several applications made under the caBIG umbrella, all design constraints such as open source platforms, J2EE, JBoss, etc. must be adhered to implicitly. Additionally, we feel it is important to note the following constraints.

6.1. Operating System and Browser Support

Clients accessing caArray must be able to be supported form both of the following platform/browser combinations:

PC's running Microsoft XP (other OS TBD), and Internet Explorer 6.0+

Apple Macs running OS X (TBD) and Firefox 2.0+

Screen Resolution: 1024 x 768 pixels

Because Firefox is the most used cross-platform browser, it should be considered as the standard for testing. This is not to suggest testing shouldn't occur for IE, rather that its lack of adherence to GUI standards may hide poor behavior. In addition, the web browsers Safari and Opera can be considered as well but testing against those, particularly for UI consistency is not required.

6.2. Open Source Database Support

The application must support MySQL Version 5.0. While version 5.1 is in beta and 6.0 is in alpha, the certainty is 5.0. In addition, it is understood that some institutions may prefer Oracle (the current database for caArray) or Postgres (version 8.2 is the current version), the target is MySQL.

6.3. Required caCORE and Infrastructure Components

The following caCore components must be incorporated into caArray's architecture:

CSM – The security module must implement and possibly extend CSM

EVS – The EVS must be used to provide controlled vocabulary

caDSR – Must be used to register meta data for exposed API(s)

caGRID – Must be used to allow access to the caArray data through the API

6.4. caBIG Compatibility

caArray must be compatible with caBIG standards at the Silver level and will look to achieve Gold if compatibility requirements are defined in time for the initial release of 2.0.

6.5. caCORE Horizontal Search

caArray 2.0 intends to use the caCORE's Horizontal Search capabilities. If mature and high performing enough by the close of elaboration, it is the team's intention to include this component to fulfill our natural text search requirement.

6.6. Application and Data Migration

For every caArray instance across the federation that contains data from a previous version, the ability to migrate the application to the next version (e.g., version 1.5 to 2.0) is required. However, data curation is not to be an activity the application is responsible for, so while a path for data migration and validation needs to be accomplished by the application, "fixing" the data is not supported. In cases where previously imported and parsed data was performed by versions prior to 1.3, expectations for smooth integration should be very low.

Going forward, however, the expectation is to support both application and data migration.

7. Precedence and Priority

Will be set once initial priority is validated by the User Community between May 9th and May 15th, 2007.

8. Other Product Requirements

8.1. Applicable Standards

8.1.1. Microarray Standards

8.1.1.1 MAGE v1.1

8.1.1.2 MIAME v1.1

8.1.2. Controlled Vocabulary

8.1.3. Usability – Section 508 Compliance

Each application provided by the National Cancer Institute must meet the Section 508 standards, in particularly being compatible with screen readers and other assistive technologies. Given the high degree of complexity in this application and the intensity of the data, this is quite a challenge. The actions must

8.2. System Requirements

8.2.1. Operating Systems

The platforms the application is expected to be rolled out to are:

Linux

Windows

8.2.2. Required Server Memory

The minimum amount of memory to be available to the application should be 2GB of memory.

8.2.3. Encryption

The community this application serves is often very sensitive about their data. It may be useful to make the entire restricted area encrypted using a Secure Socket Layer (SSL) of 128-bit encryption. This would encourage comfort among the users that their data and project descriptions are protected from their scientific competitors, interlopers and would support HIPAA guidelines for any patient data that is uploaded.

8.3. Performance Requirements

8.3.1. Simultaneous Uploads

The raw data size of the files supplied to caArray are extremely large. The speed of the upload will depend not only on the connection speed of the user but how many people are using it at the same time. However, the application should be unaffected by how many people are uploading.

8.3.2. Concurrent Users

The number of concurrent users will vary from installation to installation but a minimum number of 100 should not impact the performance of the application.

8.3.3. File Download Compression

The ability to compress in .zip format should be supported. The current maximum for zipping of data is 4GB. While that is excessively large for typical downloads, no more than 4GB should be expected to succeed and 2GB is a more likely user driven action.

8.3.4. Parsing Time

Need to determine a metric here.

9. Documentation Requirements

9.1. Installation Guide

9.2. User Guide

[Describe the purpose and contents of the User Manual. Discuss desired length, level of detail, need for index, glossary of terms, tutorial versus reference manual strategy, and so on. Formatting and printing constraints must also be identified.]

9.3. Release Notes

9.4. Use of G-Forge

9.5. Online Help

9.5.1. For the GUI Community

Significant effort should be spent on taking the use case documentation and translating it into the User Guide and subsequently to online help. The current online help system at NCICB is located here: http://ncicbsupport.nci.nih.gov/sw/bin/5/supportwizard.cgi though very little is populated by caArray (6 total responses). This is expected to go up considerable with the release of 2.0

9.5.1.1 User Forum

The caArray user forum is the most common method for asking questions. Email is sent to the forum and when appropriate, the developer community is expected to respond within 4 business hours of receiving the message. Typical response times are much less.

9.5.1.2 Web-based Training

Application Overviews and Hands-on training are expected to continue to be delivered over the web by NCICB Support (http://ncicb.nci.nih.gov/training/caarray/). The overviews occur weekly and the hands-on sessions are by request.

9.5.2. For the API Community

The most important aspect of online help for the API community is a well annotated model (required by caDSR), as well as logical and actual examples of how to get to particular kinds of data. Providing this as a part of the application is vital.