



***caBIG &  
The Mobius Project***  
***<http://www.projectmobius.org/>***

**Scott Oster, Shannon Hastings, Stephen Langella, Tahsin Kurc, Joel Saltz**

**Ohio State University**

**Department of Biomedical Informatics**

**Multiscale Computing Laboratory**



- **Presentation (~ 45 minutes)**
  1. Problem Statement
  2. Overview of Mobius
  3. How does the Mobius framework fit in the CaBIG Initiative and Architecture?
- **Demos (~ 30 minutes)**
  1. Short demonstrations of applications that are implemented using Mobius
    - Integration of genomic and molecular databases.
    - Management and analysis of image data.
  2. Tour of the Mobius client APIs (time permitting)



- The introduction of caBIG will bring together numerous data sources, each with many different data types, some of which are overlapping
- Data types will evolve over time, and Grid services may require different versions of those data types
- Clients and Grid services must be able to enforce their data representations are compatible when they communicate
- **Formal mechanisms are required to manage the structural definition of data types, and the way data and its definitions are exchanged**



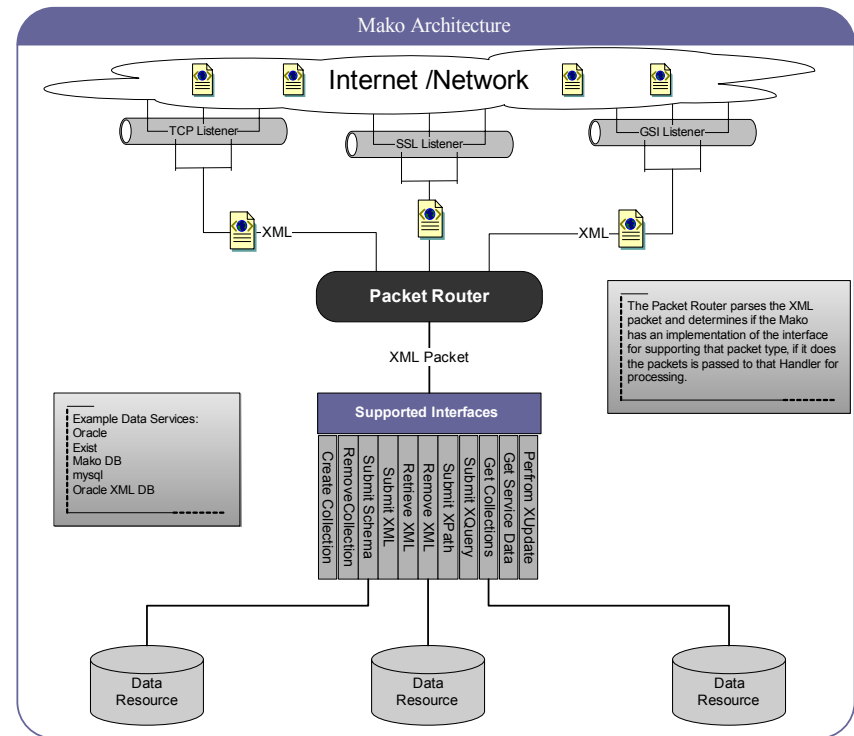
- **Identifies, defines, and builds a set of services and protocols enabling the management and integration of both data and data definitions.**
- **Features:**
  - distributed creation, versioning, management of data models and data instances
  - on demand creation of databases
  - federation of existing databases
  - querying of data in a distributed environment.
- **Consists of three main components:**
  - The protocol definitions.
  - The definition of service interfaces for utilizing the protocol.
  - Initial service implementation.



- **Mobius Core Services**
  - Mako -- Federated Ad hoc Storage Services
  - GME -- Global Model Exchange
  - DTS -- Data Translation Service
- **Mobius Extension Services**
  - VMako (Single virtual service view of a federation of Makos)
  - Other Higher level query services (semantic query, inference services etc.)
  - Data Transportation Service
- **Other Needed Grid Services**
  - Namespace Registration Management
  - Service Discovery
  - Service Naming
  - Data Replication
  - Security



- Exposes existing data services as XML data services through a set of well defined service interfaces based on the Mako protocol. (GGF/DAIS XML Realization Specification).
- Enables configuration file controllable binding of:
  - Network Listeners
  - Supported Interfaces
  - Protocol request implementation





- **Service Data**
  - Obtain metadata about Mako and its underlying data service
- **Administrative**
  - Allows the administration of the Mako and its underlying data services.
- **Security**
  - Enables management of accounts and access control.
- **Collection Creation/Deletion**
  - Data can be organized into collections and sub-collections.
- **Submit Data**
  - Data is submitted as XML, which is ingested by Mako and stored in the native format of the underlying data service.
- **Retrieve Data**
  - Data is obtained from underlying data service and returned as XML.
- **Update Data**
  - Uses XUpdate to update data.
- **Query Data**
  - Data can be queried by using XPath and XQuery.
- **Delete Data**
  - Data can be removed by specifying an identifier or XPath.





- **Mako DB**
  - In house XML database.
  - Optimized for federated ad hoc usage of XML.
  - Plugs into Mako framework and supports the full protocol.
- **XML DB Support**
  - Built in support for XML databases that support the XML DB API.
- **Exposing Relational Databases**
  - Partial support for exposing relational database via XQuark Bridge.
- **Other Data Resources**
  - Easily integrated, by implementing a small set of protocol handlers for them.
  - Any subset of handlers can be implemented (e.g. could be made Read Only)

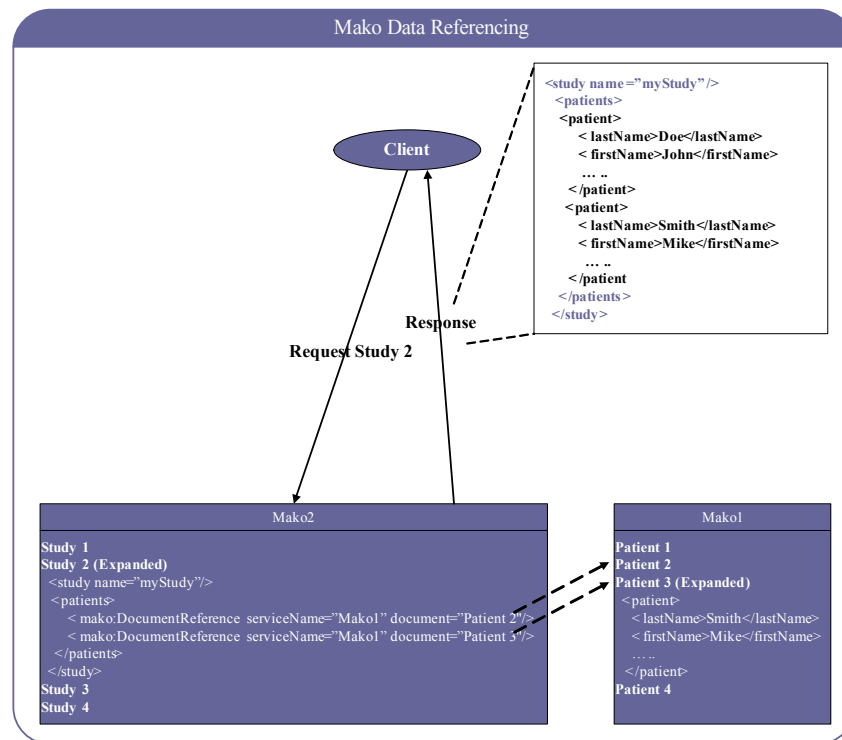




- **Security**
  - Will support grid security as set forth by the GGF.
- **Data Validation**
- **Element Referencing**
- **Lazy Retrieval**
- **Distributed Document Object Model (DOM)**

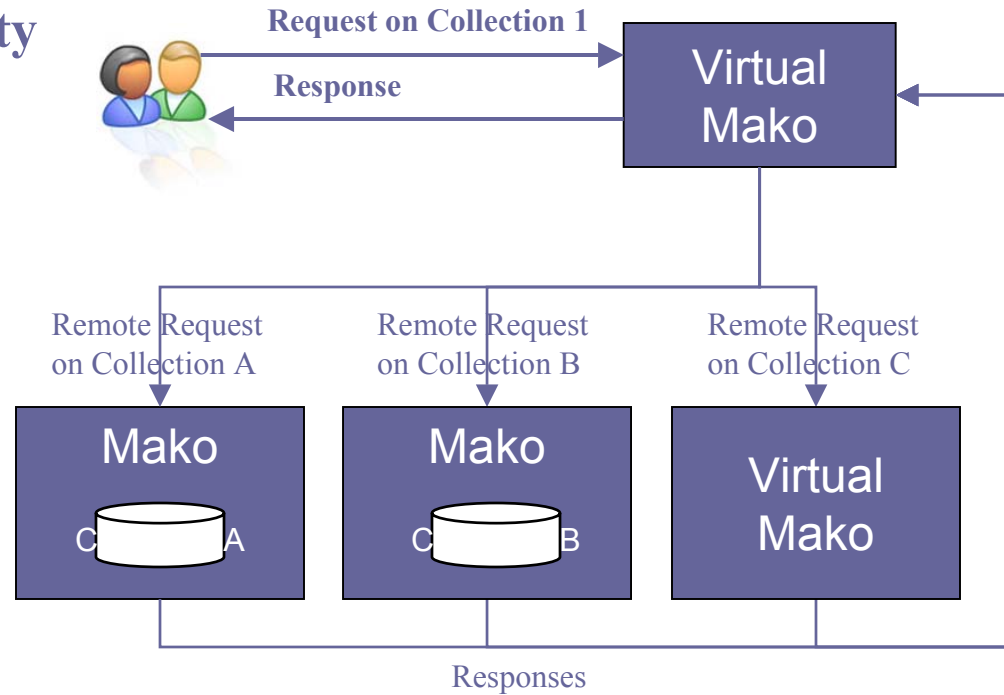


- The Mako Protocol allows pieces of data being referenced to be resolved at request time by the Mako retrieving the request, or it can be done lazily by the client
- Enables the federation of data across multiple Makos
- Enables partial result retrieval with ability to drill down later
- References can be submitted upon ingestion or created on retrieval



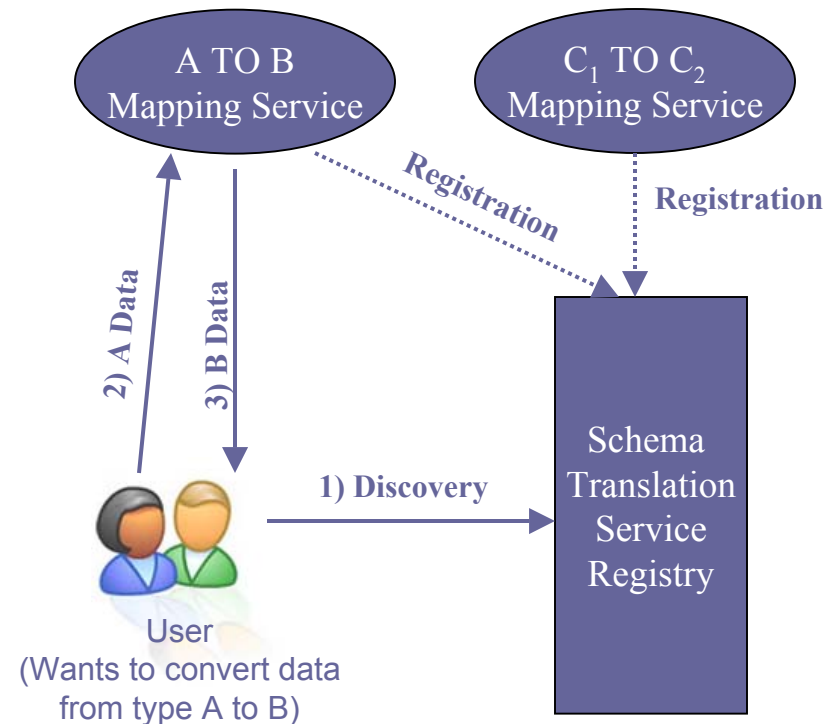


- **Simplifies client-side complexity of interfacing with multiple Makos by presenting a single virtualized interface to a collection of federated Makos.**
  - Acts as a data integration point for distributed queries
  - Pluggable algorithms for XML instance ingestion/distribution
  - Protocol request broadcast and response aggregation
  - Supports all services a standard Mako supports
  - Maps a Virtual Collection to a number of remote standard Collections or Virtual Collections





- **Use Cases**
  - How do I translate one data type to another?
  - How do I convert an old version of a data type to a newer one?
- **Protocol and service framework for handling the mapping of one data instance or data definition to another should exist.**
- **Allows two protocol-disjoint services to communicate**
- **Enables translating between changing data types.**
- *Not yet implemented*





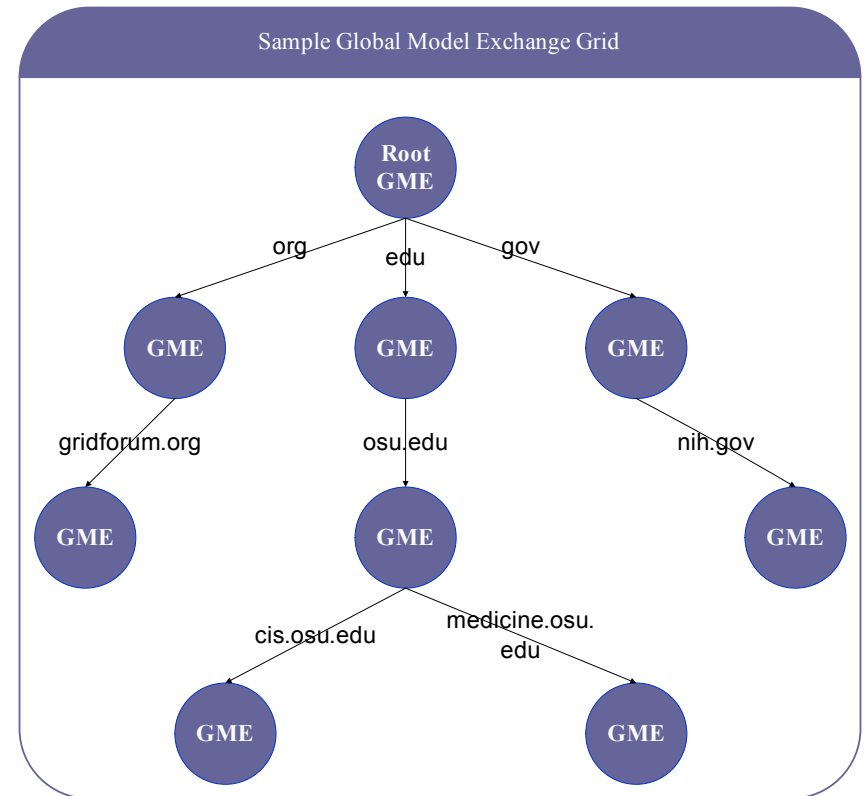
- **Need for a global data definition management!**
  - What is “global data definition” (Global Schema)?
  - Promote creation and evolution of standard definitions of data types.
  - For communication between multiple institutions they must agree on a common structure or a mapping between structures.
  - Allow for sharing and discovery of data definitions in a grid environment.



- **User/Organization defined entities**
  - e.g.: my “person” != your “person”
- **Changing schemas**
- **Schemas disappear**
- **Prevent conflicting schemas**
- **Discovering schemas**
- **Multiple definitions of similar schemas for different communities (syntactic / semantic mapping)**



- **Manages the Global Schema**
  - handles presented issues
- **Provides submission and discovery protocol**
- **Scale**
- **Replicate**
- **Cache**
- **DNS like architecture**
  - hierarchical parent child tree structure



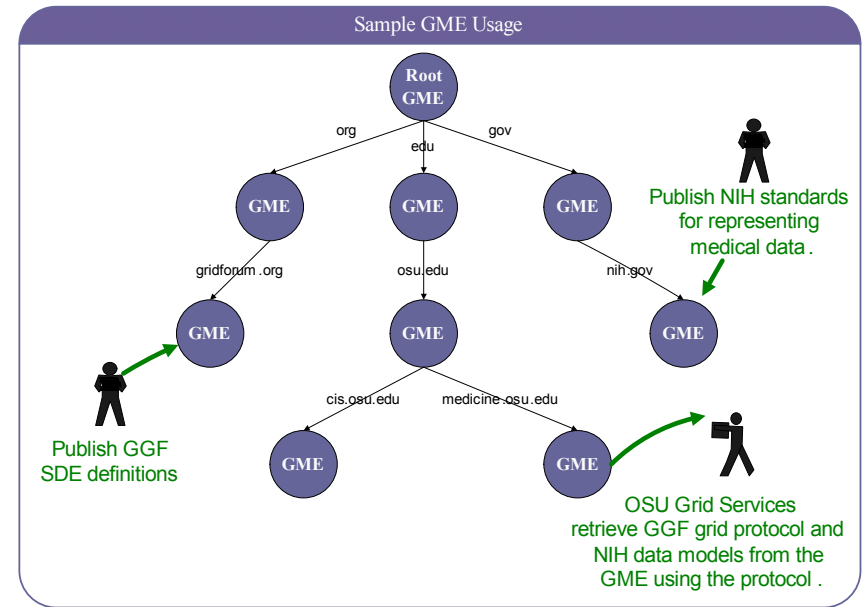




- **Publish Request**
  - Inserts a schema into an authoritative GME.
- **Retrieve Request**
  - Retrieve a schema from an authoritative or cache GME
- **Namespace Lookup Request**
  - Resolve a namespace to the authoritative GME.
- **Registration Request**
  - Registers a sub namespace GME to it's parent.



- Users/Services publish schemas to the authoritative GME of the schemas' respective namespaces.
- Any other Users/Services from similar or different organizations with the proper authority are able to reference, use, alter (version), etc the data definitions of that schema.





- **GGF**
  - Chairs of Grid Metadata Management Research Group (GMMR-RG BOF at GGF 9 and 10)
  - Active members of Data Access and Integration Services Working Group (DAIS-WG, the specification side of OGSA-DAI)
  - Active members of Semantic Grid Research Group (SEM-RG)
    - Co-author of DAIS XML Realization Specification of which Mobius is a partial implementation.
- **Papers**
  - Shannon Hastings, Stephen Langella, Scott Oster, Joel Saltz  
"Distributed Data Management and Integration Framework: The Mobius Project"  
Proceedings of the Global Grid Forum 11 (GGF11) Semantic Grid Applications Workshop, June 2004, 20-38.
  - Stephen Langella, Shannon Hastings, Scott Oster, Tahsin Kurc, Umit Catalyurek, Joel Saltz  
"A Distributed Data Management Middleware for Data-Driven Application Systems"  
To be part of the Proceedings of Cluster 2004, Sept. 2004
- **Presentations**
  - BECON/BISTIC 2004 Symposium
  - GGF's 8 9 and 10
  - Grid Performance Workshop 2004
  - Semantic Grid Workshop 2004 (Held in conjunction with GGF 11 and HPDC04)
  - NASA Ames 2004
  - RSNA (Radiology Society of North America annual conference) 2003
  - IBM Almaden 2003
  - Supercomputing 2003
- **Demos**
  - BRTT (Biomedical Research Technology Transfer) Annual Site Review 2004
  - Supercomputing 2003



- **Protocol is XML with support for binary attachments**
  - Language independent
  - Platform independent
  - Grid communication protocol independent
- **Service Definitions and Initial Implementations are Java**
  - Platform Independent
  - Limited C++ client API has been implemented



# Potential Uses of Mobius in caBIG

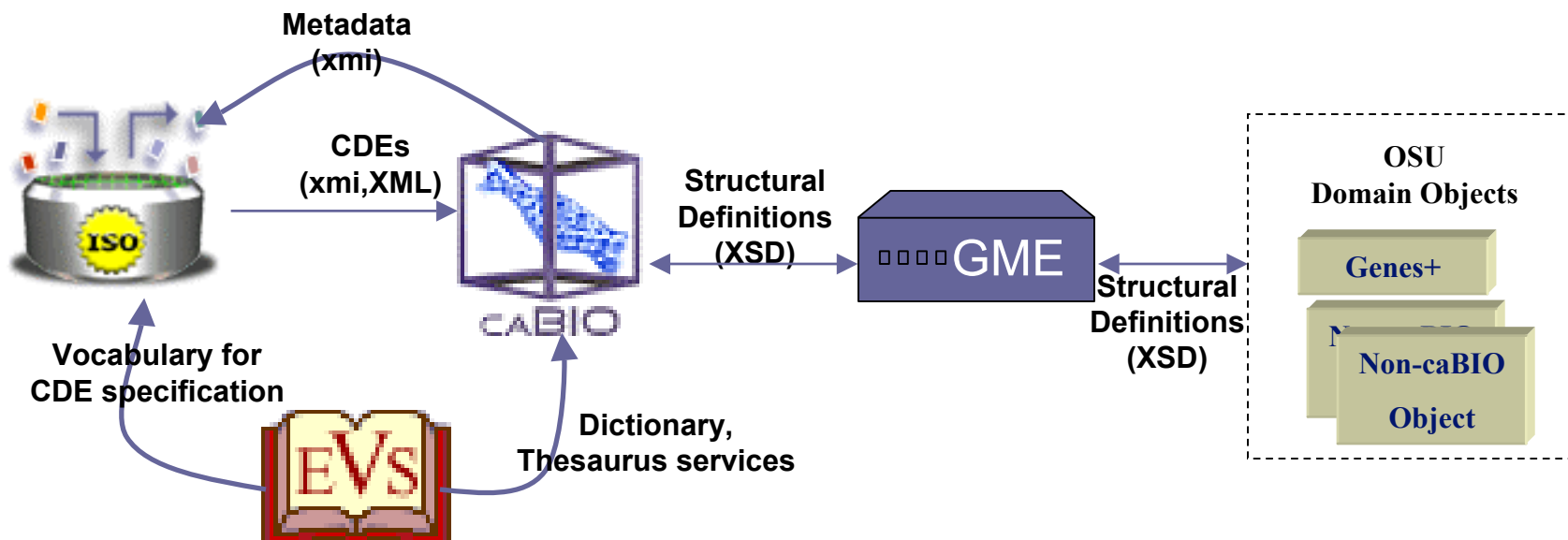


- **GME as a Structural Model Manager for caBIO**
  - Formal exchange of structural data definitions, and association of all data elements to their definitions
  - Enables interaction with non-caBIO services and new data elements not yet part of caBIO
  - Facilitates version evolution and seamless co-existence of different versions
  - Extends caCORE
    - **EVS** currently manages semantic information
    - **caDSR** currently manages controlled vocabulary
    - **GME** would manage syntactic and structural information
  - **ISSUES:** how to programmatically tie XMI, XSD, UML, OJB, etc. to generation of domain objects?



- Use Cases:**

- caBIO Object Managers validate Domain Objects against schemas in GME
- caBIO and non-caBIO clients publish schemas to GME and create data which validates against them
- Institutions are able to communicate about caBIO objects, extensions to caBIO objects, and objects not present in caBIO using the same mechanism





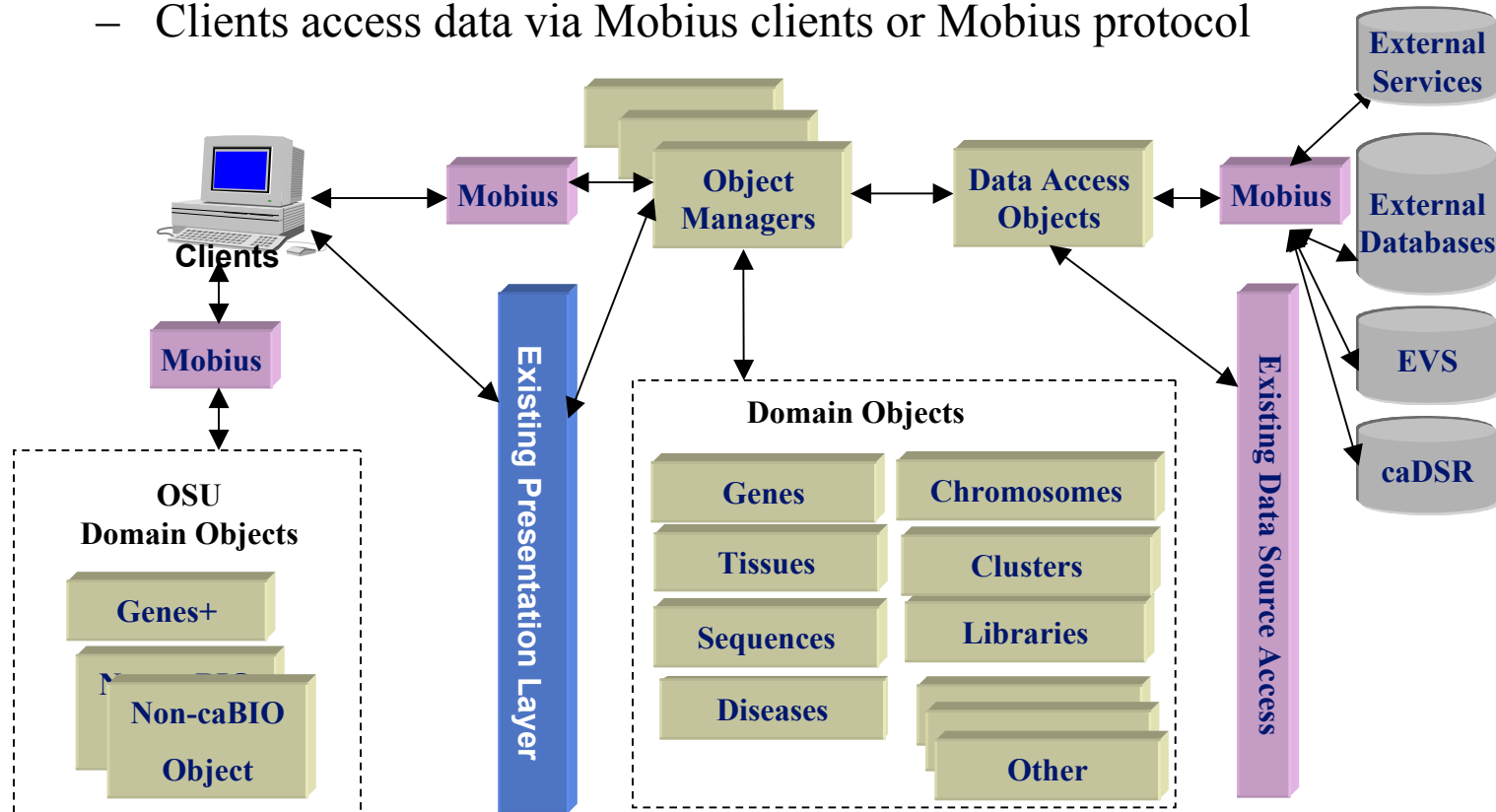


- **Leverage Mobius protocol for enable data exchange**
  - Formalizes data service interaction to be standard with both caBIO and non-caBIO services
  - XML would be similar to current caBIO XML but allows data to be associated with source (instead of getXML service), and to contain formal structural definition
  - **ISSUES:** co-existence with current getXML or replacement



- **Use Cases:**

- Existing caCORE Data Services and External Services communicate with each other using Mobius Protocol, when exchanging data or data definitions
- Clients access data via Mobius clients or Mobius protocol



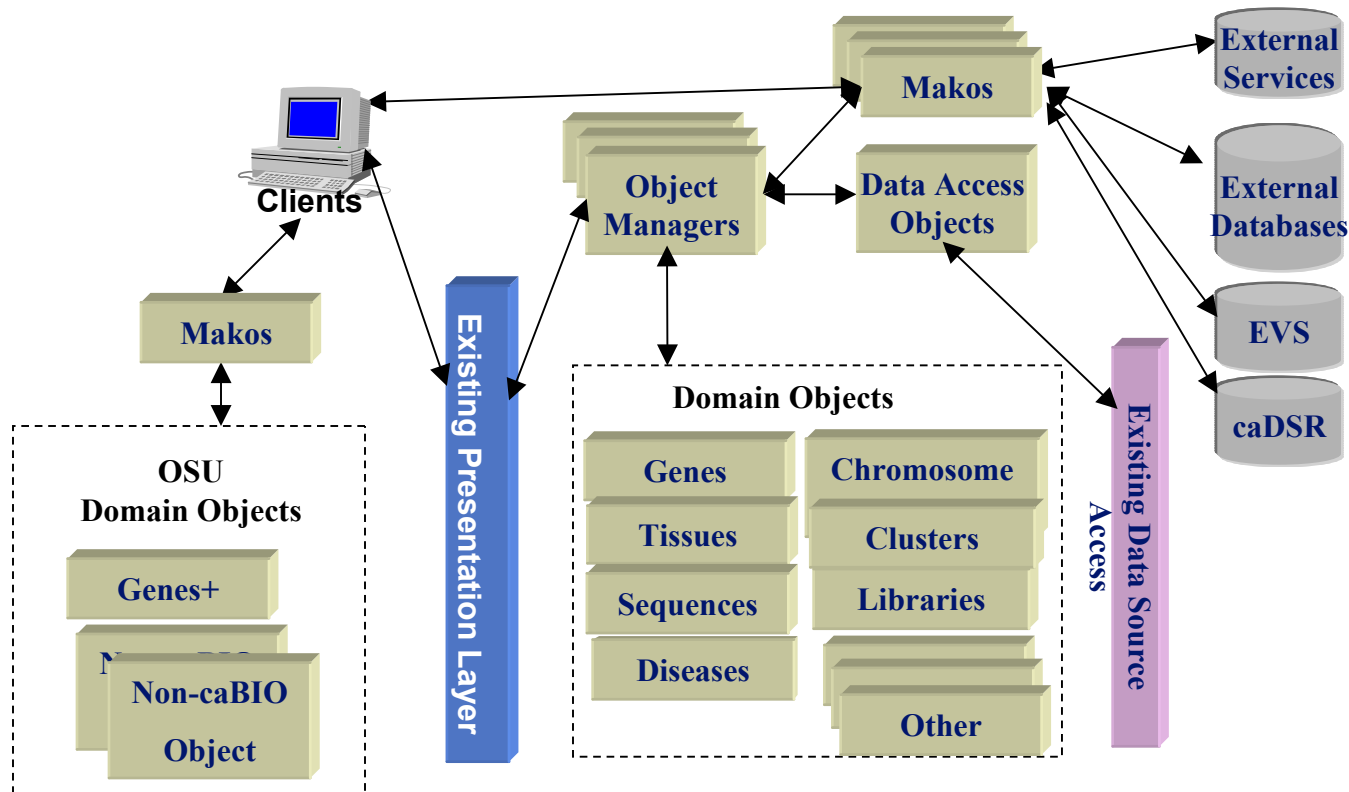


- Utilize Mako service to virtualize data services?
  - Expose data sources to caBIG Grid using Mako service
  - Similar to previous use case, but here the Mako Service is used to speak the Mobius protocol.
  - **ISSUES:** currently only supports XML virtualization (may not always be appropriate?)



- Use Cases:**

- Existing caCORE Data Services and External Services are exposed as Mako Services
- Clients access data by communicating with Mako Servers



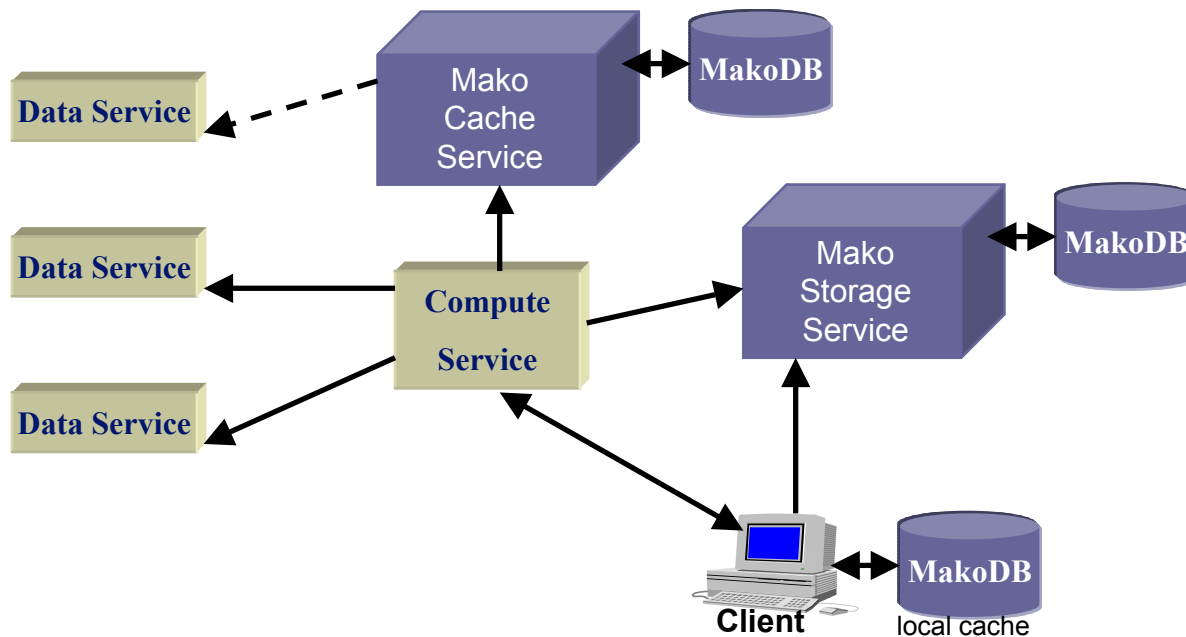


- **Provide data cache utilizing Mako and MakoDB**
  - Service interaction/collaboration for computation may require storage of temporary results and/or data cache
  - Utilize Mako's ability to generate on demand databases from schemas
  - Used locally by clients or as a Grid Service
  - **ISSUES:** schemas are required to create databases



- Use Cases:**

- Clients and Computational Grid Services utilize Makos to store and retrieve computational results
- Clients and Computational Grid Services utilize Makos as a data caches





- **“Common meta data structure and terminology is necessary to effectively describe services and data”**
  - Mobius provides a common protocol and service interface for addressing Data Services and Data Model Services
  - Mobius GME globally manages data structures
- **“A common query language is important to support federated queries”**
  - Mobius provides a protocol and service interface to request XML queries, and return their results
  - A protocol for communicating partial results of distributed joins is under development (DQP for XML)





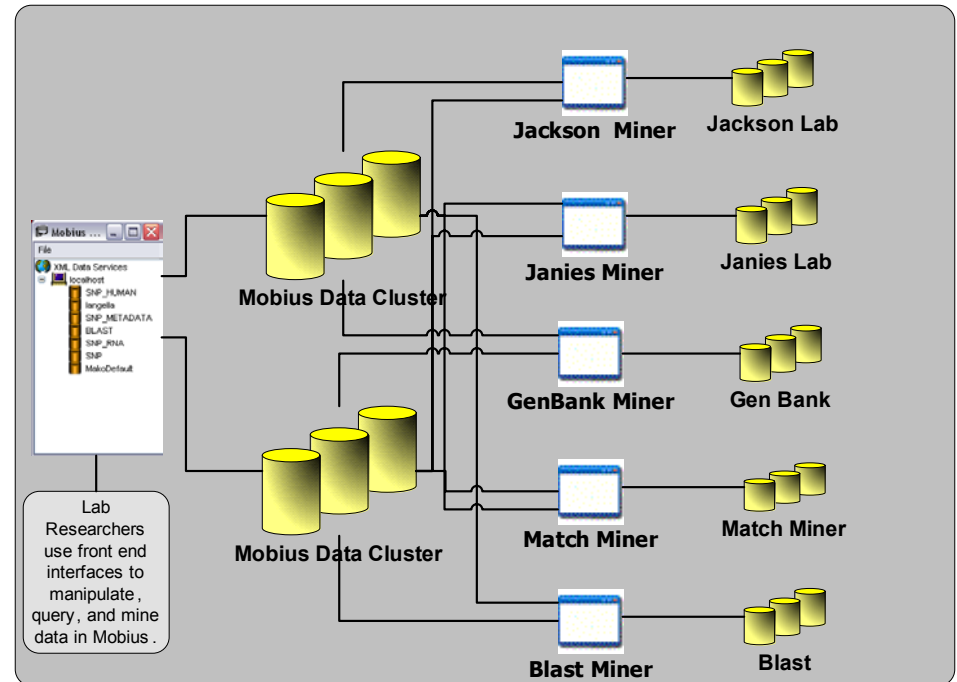
# Demos

(SNP, gPACS, client APIs and GUIs)



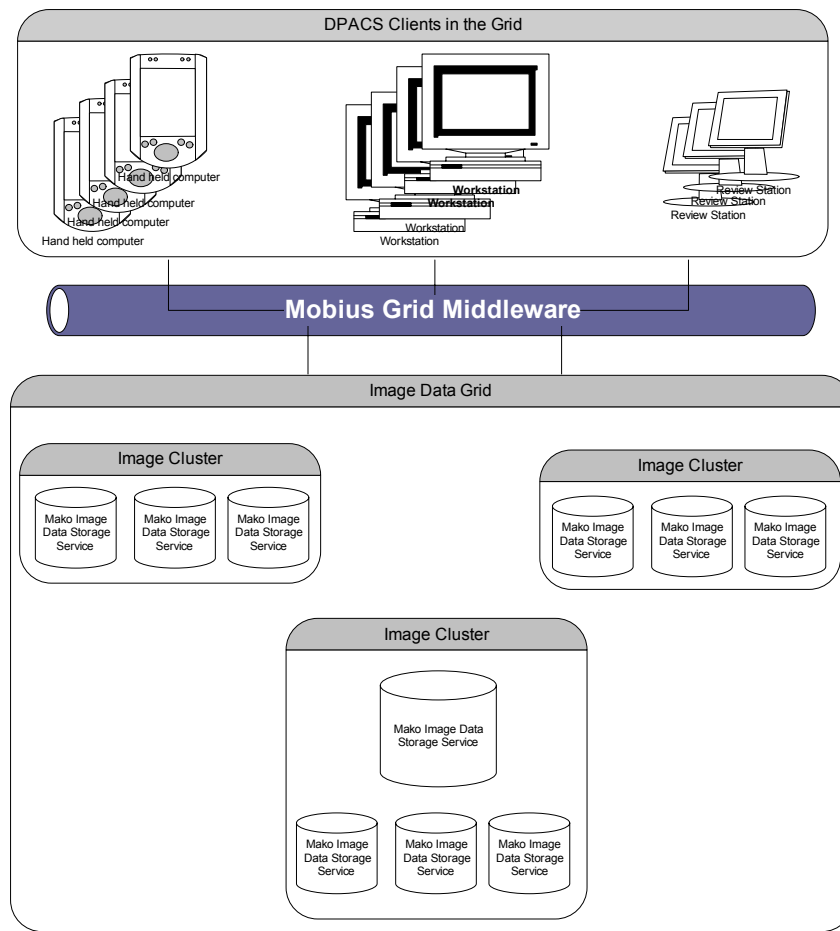
## • Finding Candidate Genes

- Overarching GOAL: Link phenotypes (traits) to genotypes
  - Complex, multi-factorial diseases e.g. Coronary artery disease (CAD),
- Long candidate lists of suspects. Much medical research is work done on one candidate gene at a time.
- We are using evolutionary variations among mouse genomes in order to search for sets of multiple genes that correlate with disease traits.





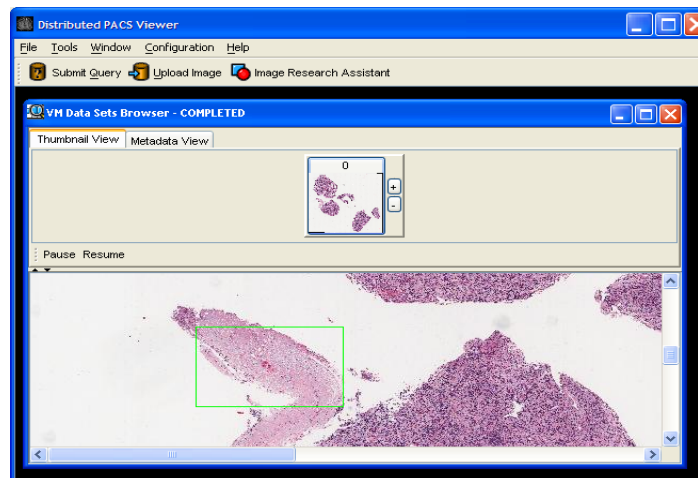
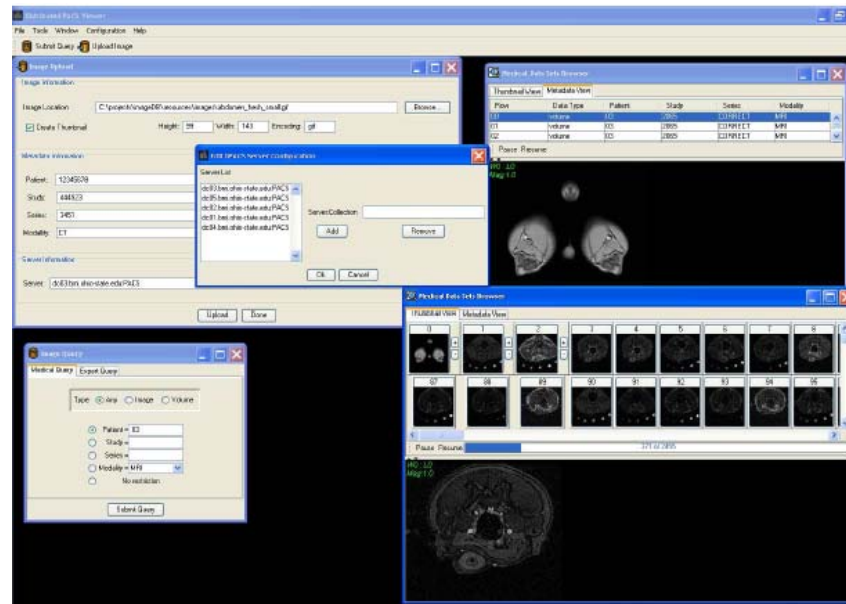
- Designed to address the storage, querying, and processing requirements of large-scale image databases in a grid wide environment.
- “Model-centric” application, majority of backend implemented by simply submitting schemas to a number of Makos
- Enables modeling and execution of image processing workflows





- **Relies heavily on the Mobius Infrastructure**

- Data Referencing – metadata and chunks of data distributed across grid via references
- Partial Retrieval – data retrieved on demand
- Distributed DOM – emulates local data environment
- VMako – query broadcast and aggregation
- Model-driven data storage – On demand creation of schema-based metadata and image storage collections on Makos





- **API walkthrough (command line and GUIs) :**
  - Show Mako configuration; startup Mako
  - Show GME configuration; startup GME
  - Add Authoritative Namespace to GME
  - Submit schema to GME
  - Create Mako collection
  - Submit XML to Mako collection (Mako will contact GME to retrieve schema)
  - Retrieve, Query, Update, Delete XML



```

<mobius>
  <resource name="makoConfig" class="org.projectmobius.services.mako.MakoConfiguration">
    <mako-configuration>
      <MobiusNetworkServiceDescriptor serviceType="MAKO" hostname="localhost" id="localhost">
        <ports>
          <port protocol="TCP" portNumber="3940"/>
        </ports>
        <aliases/>
      </MobiusNetworkServiceDescriptor>
      <handlers>
        <handler name="SubmitSchemaRequest" class="org.projectmobius.makodb.handlers.SubmitSchemaHandlerImpl"/>
        <handler name="SubmitXMLRequest" class="org.projectmobius.makodb.handlers.SubmitXMLHandlerImpl"/>
        <handler name="XMLElementRequest" class="org.projectmobius.makodb.handlers.XMLElementHandlerImpl"/>
        <handler name="RetrieveXMLRequest" class="org.projectmobius.makodb.handlers.RetrieveXMLHandlerImpl"/>
        <handler name="XPathRequest" class="org.projectmobius.makodb.handlers.XPathRequestHandlerImpl"/>
        <handler name="CreateCollectionRequest" class="org.projectmobius.makodb.handlers.CreateCollectionHandlerImpl"/>
        <handler name="RemoveCollectionRequest" class="org.projectmobius.makodb.handlers.RemoveCollectionHandlerImpl"/>
        <handler name="RemoveXMLRequest" class="org.projectmobius.makodb.handlers.RemoveXMLHandlerImpl"/>
        <handler name="XPathRemoveRequest" class="org.projectmobius.makodb.handlers.XPathRemoveHandlerImpl"/>
        <handler name="StatusRequest" class="org.projectmobius.services.mako.handlers.DefaultStatusRequestHandler"/>
        <handler name="StopRequest" class="org.projectmobius.services.common.handlers.DefaultStopRequestHandler"/>
        <handler name="CollectionListRequest" class="org.projectmobius.makodb.handlers.CollectionListHandlerImpl"/>
        <handler name="SchemaListRequest" class="org.projectmobius.makodb.handlers.SchemaListHandlerImpl"/>
        <handler name="XUpdateRequest" class="org.projectmobius.makodb.handlers.XUpdateRequestHandlerImpl"/>
      </handlers>
      <factories>
        <xmlDataServiceFactory class="org.projectmobius.client.mako.DefaultXMLDataServiceFactory"/>
        <xmlDataModelServiceFactory class="org.projectmobius.client.gme.DefaultXMLDataModelServiceFactory"/>
        <xmlCollectionFactory class="org.projectmobius.client.mako.DefaultXMLCollectionFactory"/>
      </factories>
      <gme-client-configuration>
        <gme-identifier identifier="localhost"/>
      </gme-client-configuration>
    </mako-configuration>
  </resource>
  <resource name="makoDBConfig" class="org.projectmobius.makodb.MakoDBConfiguration">
    <makoDB-configuration>
      <idManager maxId="2147483647" elementIdBatchSize="100" mapIdBatchSize="100"/>
      <insertionManager bufferSize="1000000"/>
      <attachments directory="attachments" directorySize="500"/>
      <upgrade auto="true"/>
      <databases>
        <database name="ROOT">
          <name/>
          <driver>com.mysql.jdbc.Driver</driver>
          <urlPrefix>jdbc:mysql:</urlPrefix>
          <host>localhost</host>
          <port>3306</port>
          <username>root</username>
          <password/>
          <pool>1</pool>
        </database>
      </databases>
    </makoDB-configuration>
  </resource>
</mobius>

```





```

<mobius>
  <resource name="gmeDatabaseManager" class="org.projectmobius.services.gme.GMEDatabaseManager">
    <gme-configuration id="localhost" hostname="localhost">
      <root-database-name>ROOT</root-database-name>
      <registry-database-name>GME_REGISTRY</registry-database-name>
      <schema-store-database-name>GME_SCHEMA_STORE</schema-store-database-name>
      <root-database>
        +
      </root-database>
      <databases>
        +
      </databases>
    </gme-configuration>
  </resource>
  <resource name="gmeConfig" class="org.projectmobius.services.gme.GMEConfiguration">
    <gme-configuration id="localhost" hostname="localhost">
      <gme-communication-protocol>TCP</gme-communication-protocol>
      <MobiusNetworkServiceDescriptor serviceType="GME" hostname="localhost" id="localhost">
        <ports>
          <port protocol="TCP" portNumber="1111"/>
        </ports>
        <namespaces>
          <namespace name="bmi.osu.edu"/>
        </namespaces>
      </MobiusNetworkServiceDescriptor>
      <handlers>
        <handler name="StopGMERequest" class="org.projectmobius.services.gme.handlers.StopRequestHandler"/>
        +
        <handler name="StatusResponse" class="org.projectmobius.protocol.StatusResponse"/>
      </handlers>
      <policies>
        <performance-caching>
          <namespace-caching max-records="5" time-to-live="4:00"/>
          <schema-caching max-schemas="1000"/>
          <schema-lookup time-to-live="4:00"/>
        </performance-caching>
        <quality-caching>
          <namespace name="osu" method="notification"/>
        </quality-caching>
        <notification-policy support="specified">
          <notification-list>
            <gme hostname="" id=""/>
          </notification-list>
        </notification-policy>
      </policies>
    </gme-configuration>
  </resource>
  <resource name="authorityManager" class="org.projectmobius.services.gme.AuthorityManager">
    <gme-configuration id="localhost" hostname="localhost">
      <authorities>
        +
      </authorities>
    </gme-configuration>
  </resource>
  <resource name="subordinateManager" class="org.projectmobius.services.gme.SubordinateManager">
    <gme-configuration id="localhost" hostname="localhost">
      <subordinates>
        +
      </subordinates>
    </gme-configuration>
  </resource>
</mobius>

```





- **Integration with caGRID prototype?**
- **Investigation of potential caBIO/Mobius workflow**
- **Investigate how XMI models could be used with GME**
- **Others?**



***caBIG &  
The Mobius Project***  
***<http://www.projectmobius.org/>***

**Scott Oster, Shannon Hastings, Stephen Langella, Tahsin Kurc, Joel Saltz**

**Ohio State University**

**Department of Biomedical Informatics**

**Multiscale Computing Laboratory**

The  
**MOBIUS**  
PROJECT



