GenomeSpace Architecture

The primary services, or components, are shown in Figure 1, the high level GenomeSpace architecture. These include (1) an Authorization and Authentication service, (2) an analysis and tool manager service (ATM), (3) a provenance/history system, and a (4) data manager (DM).

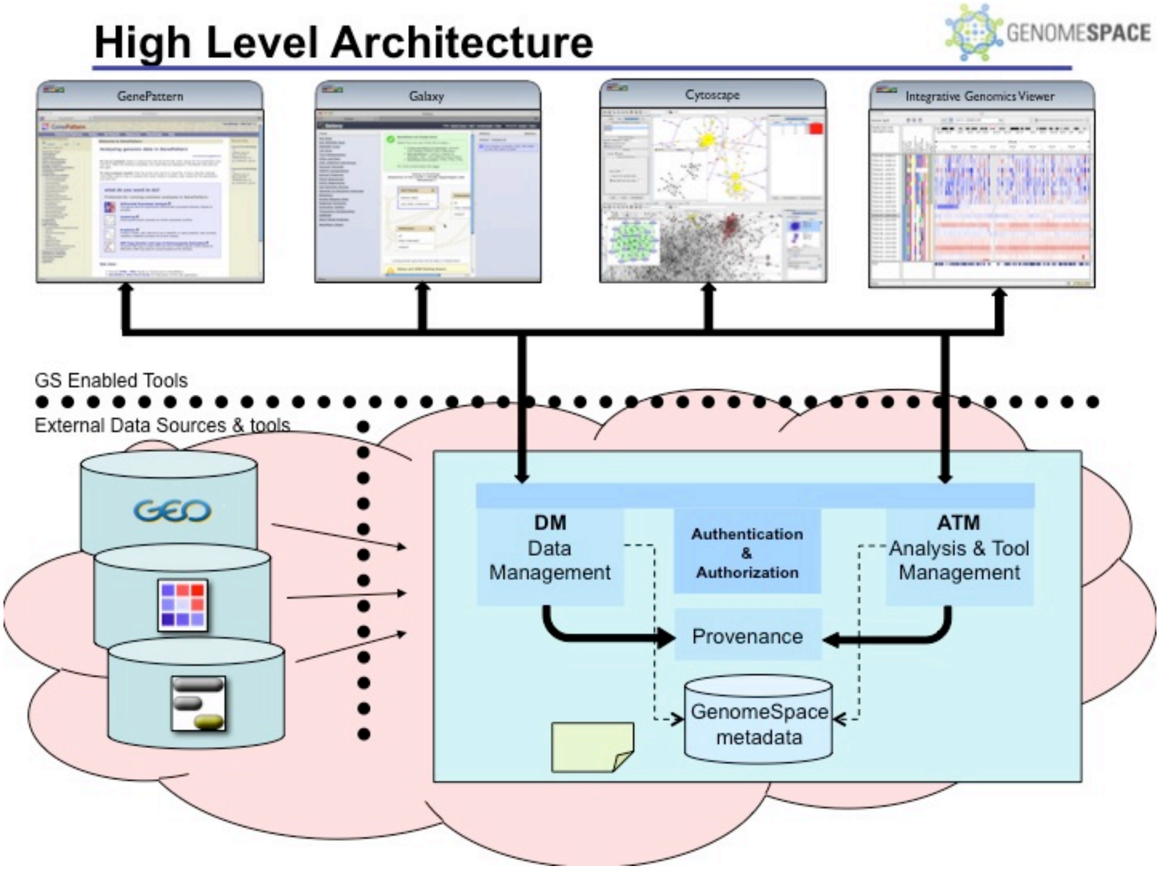

Figure 1. High Level GenomeSpace Architecture

Authentication and authorization service.

Authentication and authorization will be implemented using Java as the core technology for both authentication (are users who they claim to be) and authorization (what are users allowed to do) for users entering the GenomeSpace system. The Authentication portion will be implemented with two interfaces for clients. First, a basic authentication (RFC 2617) interface which is available to any client. Second an OpenID protocol implementation, which will provide single-sign-on support for web applications. In general the OpenID implementation is preferred for web-based tools however it is not appropriate for desktop clients (such as IGV, Genomica and Cytoscape) as it relies on the requesting software application (IGV etc) to provide a URL that it can be called at to establish a secure channel to the opened server. For desktop applications this is often impossible due to firewall policies. Therefore we will also support the basic authentication protocol for desktop applications. The GenomeSpace services will be wrapped with common standardized servlet filters that ensure that no unauthorized access occurs.

GenomeSpace will launch tools, enable communication between them, serve them with data, and capture results.

The analysis and tool management system (ATM) will provide the core interoperability functionality to permit the analysis and visualization tools to be launched from the GenomeSpace environment. It will (1) manage the catalogue of GenomeSpace-enabled tools; (2) support tagging tools to allow users to find tools appropriate for specific tasks; (3) allow users to launch specific tools from the GenomeSpace environment; and (4) late in the project, provide the capability to create inter-tool analytic workflows.

We are developing an analysis engine for the ATM with the following features: 1. placement of tags (non-hierarchical keywords) on tools, and on specific parameters of these tools. GenomeSpace users will be able to add arbitrary tags to any GenomeSpace tool or tool parameter. To populate the initial tags on tools we will use keywords provided by the tool authors, and where possible with terms from the Ontology for Biomedical Investigations (OBI) Ontology for Data Transformation.

Launching of tools from the ATM will support running, or transferring to, Web applications. This is distinct from the functionality currently available in Galaxy and GenePattern which focus on the running of command line tools. Launching of tools in GenomeSpace will also permit the passing of contextual information to tools launched from GenomeSpace. Contextual information may include, but is not limited to, tokens defining user identity and a reference to the server that the launched tools can use to communicate back.

Transferring data files between tools or from the data manger to a requesting tool may involve some data format translation. Format Translation will be primarily implemented in the Data Manager (below) but this feature will be key to the launching of tools so it is discussed here. Data Transformations will be accomplished with format converters, which will be applied to data files as needed. Conceptually these are similar to shims in the Taverna environment. These format converters will be loaded into the GenomeSpace servers as OSGi bundles (or another similar mechanism that permits isolation of code and memory for each format converter to prevent classpath conflicts), allowing them to be dynamically updated as required, and catalogued in a format converter registry. Upon sending messages between two GenomeSpace-enabled applications, the metadata defining the data formats would be queried, and the registry consulted to find an appropriate transformation. If a transformation existed, it would be applied to the data before it was transferred to the receiving application.

Tools for data retrieval, formatting, and local storage.

We will develop tools to allow GenomeSpace users to retrieve data from the common publicly available sources for microarray data, genome tracks, SNP data, gene sets, and other data types as time and resources permit. We will develop tools to allow users to upload their own datasets to GenomeSpace and manage their private and remote data as GenomeSpace projects. We will develop a system to automatically convert data formats of different tools, removing the need for users to perform manual conversion when moving data from one tool to another

Support for these capabilities lies primarily with the Data Manger (DM). The DM will be a new implementation of a REST-ful Java server. The DM will use a plug-in architecture with a central registry of data source plug-ins. Specific implementation details for connecting to particular public data sources will be delegated to the individual plug-in for that source.

The data source registry will maintain metadata about data source properties including some metadata common to all, with a model that allows extensibility for specific data sources. The minimal metadata for a source will include data source name, connection information, data source plug-in class name, and details of the types and formats of data that may be retrieved. The registry will make this metadata available to the GenomeSpace user interface to allow it to present user interfaces for a data source to the end- user.

For many common publicly available data sources, we will implement data source plug-in implementations. Initially, we will provide plug-ins to retrieve microarray data, e.g., from GEO, ArrayExpress, and caArray, tracks (sequence, features, etc.) from the UCSC Genome Browser, SNP data from dbSNP and the Welcome Trust Case Control Consortium (WTCCC), gene sets from the Molecular Signatures Database (MSigDB), and others. For some of these data sources we will base the plug-ins on existing data retrieval modules in GenePattern. Other data source plug-ins will reuse existing implementations from Bioconductor. For the remaining data sources, we will investigate availability of Web services for data access and retrieval. If these do not exist, we will contact the data providers to investigate the best way to architect the plug-in. File format conversion of data retrieved from these data sources (if necessary) will be implemented as discussed above.

To provide users with the ability to maintain and manage their own datasets within GenomeSpace, we are first developing a GenomeSpace project data source. The project data source will use GenomeSpace storage in the Amazon cloud to provide users with read and write access to private project storage maintained on the GenomeSpace server. Computation and storage for this data source is being provided by Core A, Software Maintenance and Support. The security layer (see Overview above) will protect GenomeSpace projects and ensure privacy and confidentiality of user data. Unlike the public data sources (above) the GenomeSpace project data source permits not only reading but also write-back of data.

GenomeSpace user environment.

We will develop a user interface that will provide access to local and remote data and launching of analysis tools.

Researchers will work with GenomeSpace through the GenomeSpace user environment. The GenomeSpace user environment will be composed of two separate elements (Fig. 4). The first is the user interface for GenomeSpace (GS-UI), which provides access to the tools and data in the ATM and DM. The second is the integration of GenomeSpace user interface elements into GenomeSpace-enabled applications.

The second element of the GenomeSpace user environment is the integration of GenomeSpace menus and features into the individual tools and applications. This will provide users with the ability to

a. Send a dataset to another running GenomeSpace-enabled application,
b. Send a sub-selection of a dataset to another running GenomeSpace-enabled application,
c. Save a dataset (or sub-selection) back to a project folder on the DM,
d. Send a command request to another running GenomeSpace-enabled application

Support for GenomeSpace functionality will be packaged as a GenomeSpace Client Development

Kit (CDK) although it will be possible for clients to call the GenomeSpace services directly via their web interfaces.

System for capturing the history of GenomeSpace user sessions for reproducibility.

The provenance and session history system will be implemented as a separate service, which will receive communication from the DM and ATM services. It will deal with history capture for three separate cases: for the GenomeSpace servers themselves, for tools that support reproducible research and re-execution of analyses, and for tools lacking any internal history mechanism.

The Provenance server will track all tool invocations, including parameter settings, made from within the GenomeSpace environment. We will base development on the existing reproducible research capabilities of GenePattern which.

In a later development, for GenomeSpace-enabled interactive tools, which support reproducible research (e.g., GenePattern, Galaxy), we will work with their development teams to integrate those tools history capture and replay mechanisms. This development is expected to begin around the third year of development. For this we are evaluating two possible approaches: (1) record a reference the tool can use to replay a session, and (2) record a tool-provided replay-able form of the history that can be re- executed by the tool.

The Provenance and history system will support the recording and replay the steps of an analysis performed through GenomeSpace including the tools and parameters used, and data transfers between interactive GenomeSpace-enabled tools.

Support adaptation of tools to the GenomeSpace Environment.

We will provide the necessary capabilities and support to easily wrap and connect new tools to the GenomeSpace tool management, data retrieval, and history capturing systems.

Modifying tools for GenomeSpace-enablement. To participate as a GenomeSpace-enabled tool, an application must send and receive messages and data, maintain provenance of its analyses and other key actions, and save and restore its state as a “session”. We will specify and develop, packages that provide a system for developers to add these essential functions to their tools. We will implement the GenomeSpace-enabling packages in the GenomeSpace Client Development Kit (CDK) that provides simplified and controlled access to the web services interfaces The CDK software libraries will be developed on top of the various servers programmatic Web services interfaces. We will publish the specification for the CDK and the web services interfaces, and provide consultation and assistance, so that developers coding in other languages such as C++, Python, or Flash, can also integrate their tools.

Architectural Description

An early decision made in the design of the GenomeSpace services was to utilize readily available cloud-computing platforms and services in order to simplify development and provide a path to scalability. For this project we have chosen to use Amazon Web Services (AWS) as the platform upon which the GenomeSpace services will be developed. In particular we plan to use the following elements of the AWS technical stack:

Elastic Compute Cloud (EC2): We are using the EC2 system to provide virtual machines to host the publicly accessible GenomeSpace servers (Development versions are run locally within the Broad Institute).

Elastic IP: We are using Amazon's Elastic IP system to assign persistent Internet protocol addresses to our EC2 servers.

Simple Storage System (S3): Amazon Simple Storage Service provides a fully redundant data storage infrastructure for storing and retrieving any amount of data. We are using S3 as the basis for the GenomeSpace Project Folders data source and using S3 capabilities (for the moment) for upload and download of files into S3 and the Data Manager.

SimpleDB: Amazon SimpleDB is a highly available, scalable, and flexible non-relational data store. We are using it to store metadata for the GenomeSpace ATM, DM, Authorization and Provenance (eventually) servers.

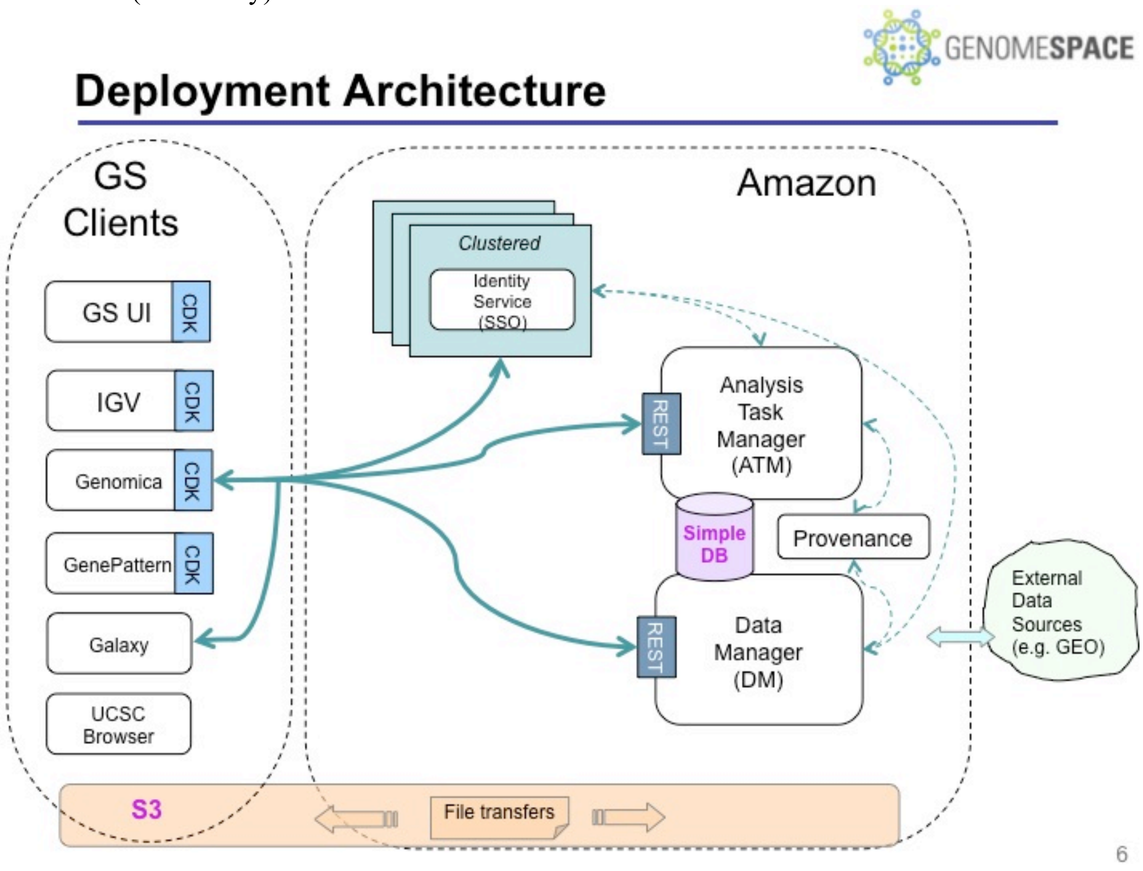

Figure 2. GenomeSpace Deployment Architecture

Server Deployment

The GenomeSpace servers (ATM, DM, Identity/Authorization and Provenance) are all being written to be deployed using a standard pattern.

As the primary interfaces are REST-ful web services, each server is initially developed as a Jersey web service. Jersey is the open source, production quality, JAX-RS (JSR 311) Reference Implementation. Each Jersey service implementation is built with a standard container request filter providing security protection and ensuring that all requests are authorized before the server sees them. The Jersey services are packaged in Java Web Application Archive (WAR) files for deployment. The war files are deployed into Apache Tomcat 6.0 servlet engines. It is expected that at some future date we may move to a different servlet engine to support scalability or better performance but for now, Tomcat is sufficient. The Tomcat servlet engines are deployed onto Amazon EC2 Amazon Machine Instances (AMIs) and this is where the services are run.

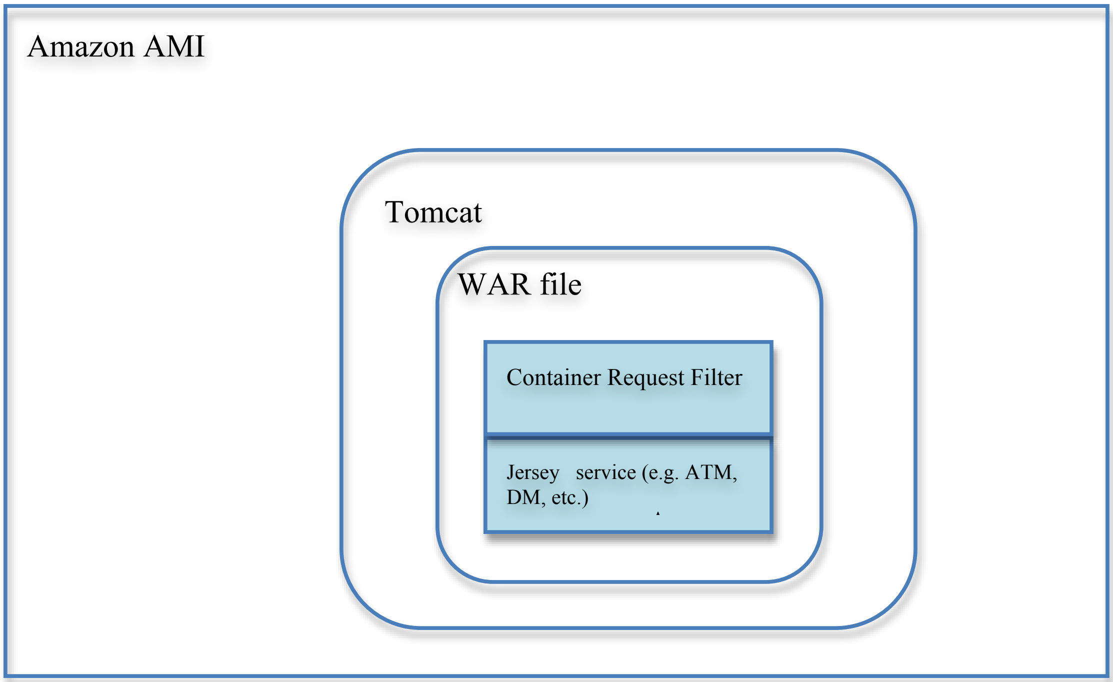

Figure 3. Server Deployment Layers

The server implementations (ATM, DM, etc) interact with the AWS services (S3, SimpleDB, etc) through the use of third-party java libraries. In particular we are using jets3t to interact with S3 and Typica.
At the current stage of development (January, 2011) there is still some variability, but this is being addressed on an ongoing basis.

Use Case 1: Launch GenePattern from Genomica

As an example of the GenomeSpace system in use, we will describe the behavior of the components for a typical use case, launching GenePattern from Genomica on a dataset

Description:

A researcher has been working on a gene expression dataset in Genomica. They wish to send the dataset to GenePattern to have GenePattern perform Hierarchical clustering on it.

Prerequisites:

Genomica has the GenomeSpace CDK installed and configured into the UI.
GenePattern has been loaded as a web tool into the ATM.
Hierarchical Clustering in GenePattern has been loaded as a web analysis tool in the ATM.
The user has already registered for a GenomeSpace user account.
The DM has the code necessary for a transformation from Genomica's tab or gmt format into GenePattern's gct format.

Actors:

Researcher – a human working on a computer to analyze gene expression data
GenePattern – a web application
Genomica - a java desktop application
GS Identity server – a GenomeSpace web service
ATM – the GenomeSpace Analysis and Task Manager web service running on AWS.
DM - the GenomeSpace Data Manager web service running on AWS.
S3 – the Amazon Simple Storage Service
Browser – the Researcher's web browser

User Steps:

These steps describe the behavior of the system from the user's (Researcher's) perspective.

1. Researcher opens Genomica.
2. Researcher loads a dataset.
3. Researcher does stuff in Genomica …
4. Researcher selects GenomeSpace menu in Genomica toolbar, login
5. Resercher enters Genomespace user/pass to login
6. Researcher selects GenomeSpace menu, 'send to', 'GenePattern', 'Hierarchical Clustering'
7. Browser opens on Researcher's screen showing GenePattern with Hierarchical Clustering module loaded and a GenomeSpace URL to their dataset in the input filename field.
8. Researcher hits 'run' button to start the analysis in Genepattern
9. Researcher does other stuff in Genepattern…

System Steps:

These steps describe the behavior of the system from the perspective of the software components (or a software developer).

1. Researcher opens Genomica.
   i. Genomica starts up, initializes UI, adds GenomeSpace menu to toolbar.
2. Researcher loads a dataset.
   i. Genomica loads a tab or gmt formatted dataset
3. Researcher does stuff in Genomica …
   i. Genomica responds to user input normally

4. Researcher selects GenomeSpace menu in Genomica toolbar, login
    i. Genomica presents login dialog to user.
5. Resercher enters Genomespace user/pass to login
    i. Genomica gives user/pass to CDK
    ii. CDK sends user/pass to IdentityServer
    iii. IdentityServer sends back token/cookie to CDK
    iv. CDK saves token/cookie to send along on subsequent GenomeSpace calls
    v. CDK contacts ATM, requests list of web tools and web analyses
    vi. CDK returns control to Genomica
    vii. Genomica takes list of ATM tools and analyses and constructs send-to menus
6. Researcher selects GenomeSpace menu, 'send to', 'GenePattern', 'Hierarchical Clustering'
    i. Genomica prepares current dataset as a gmt/tab formatted file
    ii. Genomica tells CDK to send file to GenePattern/HC analysis
    iii. CDK sends tab/gmt file to DM
    iv. CDK gets file descriptor for uploaded file
    v. CDK calls ATM, tells it to launch GenePattern/HC on file descriptor
    vi. ATM reviews parameters for HC on GenePattern, assigns file descriptor to input.filename parameter
    vii. ATM notes tab/gmt to gct/res transformation required
        i. ATM calls DM, requests URL for transformed file
        ii. DM launches asynch transformation of gmt/tab file to a gct file
        iii. DM returns URL to transformed file to ATM (before transformation is complete)
    viii. ATM generates URL to launch GP/HC with URL the DM gave it
    ix. CDK launches Browser on Researcher's computer
7. Browser opens on Researcher's screen showing GenePattern with Hierarchical Clustering module loaded and a GenomeSpace URL to their dataset in the input filename field.
    i. GenePattern behaves normally.
8. Researcher hits 'run' button to start the analysis in Genepattern
    i. GenePattern begins launch of task. Tries to download URL
        i. If transformation is complete, files is downloaded in gct format
        ii. If transformation incomplete, download blocks until ready
        iii. Analysis in GenePattern continues normally.
9. Researcher does other stuff in Genepattern…
    i. GenePattern responds to user input normally.