

GEODE: Technical Documentation (v0.2)

Koon Leai Larry Tan

University of Stirling

16th November 2007 [Edition 0.2]

GEODE Project Technical Paper No. 3

Technical Papers of the GEODE project: Grid Enabled Occupational Data Environment, www.geode.stir.ac.uk/publications.html. ESRC Small Grant in eSocial Science, Ref: RES-149-25-1015. The GEODE project is affiliated with National Centre for eSocial Science (www.ncess.ac.uk) and National eScience Centre (www.nesc.ac.uk).

Document Conventions

%<name>% refers to an environment variable in Windows. E.g.
 %CATALINA_HOME%=c:/tomcat

GT4 → Globus Toolkit 4

MDS → Monitoring and Discovery Services

1. System Architecture

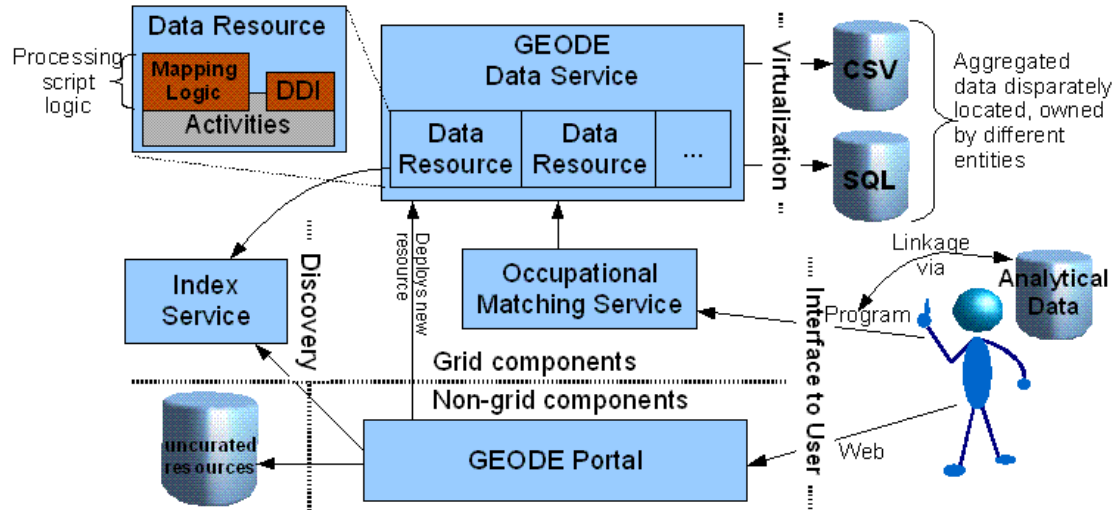


Figure 1. Overview of GEODE

Figure 1 categorizes the components, including clients, of the GEODE environment. Some of the major activities such as deploying new resources and performing occupational data matching are illustrated as “Deploys new resource” and “Linkage via → Program” respectively. The GEODE-customised configuration of the data resource is shown, with the Mapping Logic and DDI on top of customised Activities.

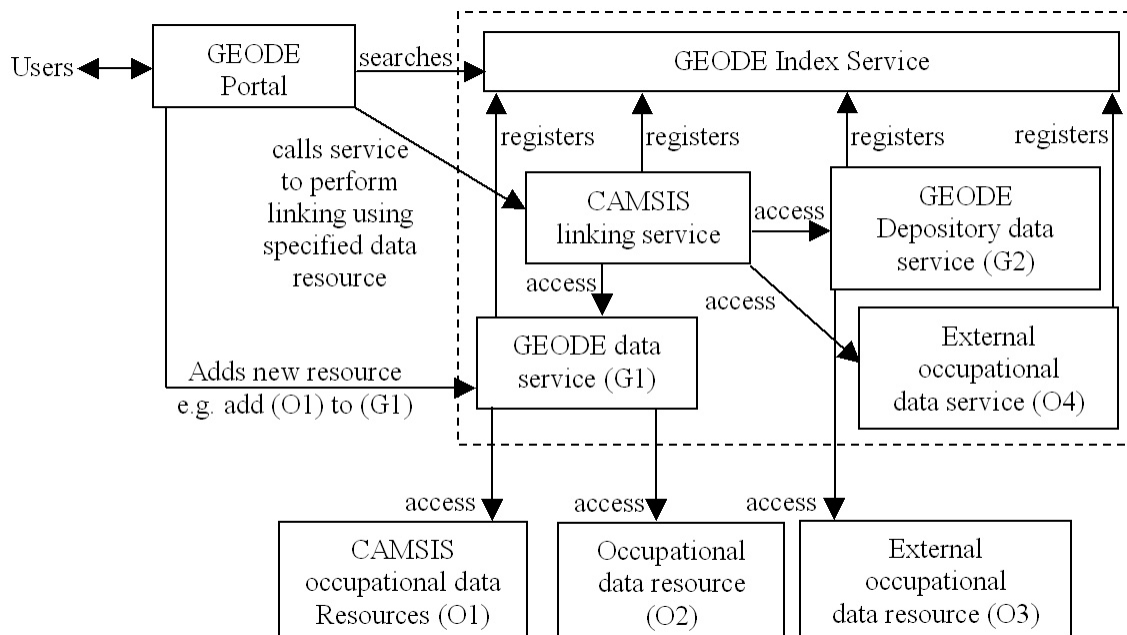


Figure 2. Functional relationships between GEODE core components

Figure 2 illustrates to a certain extent the scope

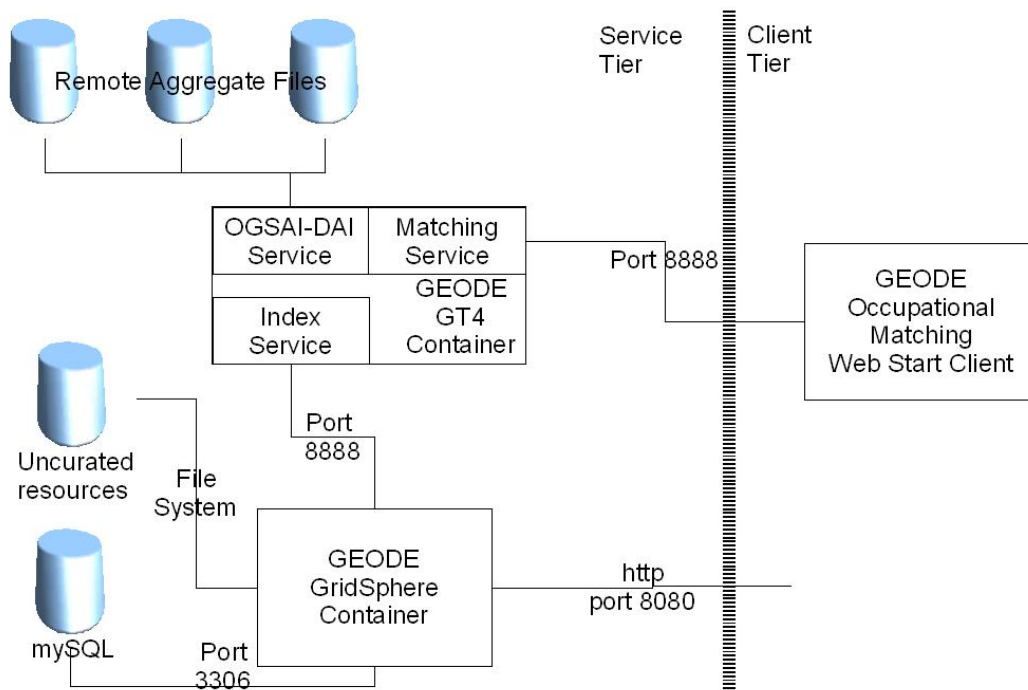


Figure 3. Actual Connections between components and resources

GEODE project develops/deploys the following software:

1. Web Portal (develop)
 - a. Searching for occupational data resources
 - b. Managing occupational data resources
 - c. Searching for occupational matching resources
 - d. Launching Occupational Matching Web Start Client
2. GEODE Index Service (deploy)
 - a. DefaultIndexService (from MDS)
3. GEODE data service
 - a. Factory resource that facilitate creation of data resources
4. GEODE Occupational Matching Services
 - a. Factory service that creates occupational matching resources
 - b. Instance service that manages occupational matching resources

Web/Grid Portal

The GEODE portal is a web portal developed using the GridSphere portal development framework, deployed in Tomcat 5. Registered GEODE users are allowed to do the following

1. Searching for occupational data resources
2. Managing occupational data resources
3. Searching for occupational matching resources
4. Launching Occupational Matching Web Start Client
5. Submit feedbacks to administrators

The portal services interfaces with the GEODE grid/data services in the following ways:

1. Querying GEODE Index Service
2. Creating/deleting occupational matching resources
3. Deploying/un-deploying data resources

There (JAR) libraries of the GEODE services and GT4 are required for the portal to operate.

The portal is developed to be configurable at deployment/start-up time. It is done via properties files in the %CATALINA_HOME%/webapps/geode-portal/WEB-INF/classes. The portal at present uses MySQL databases for storing information.

NB: The portal is also the web server at present. There is not much static graphics or textual context yet to justify the need for the httpd server.

GEODE Services

There are 3 types of grid services in GEODE: index, data, and occupational matching. All these services runs in the same container.

Index Service

The index service is the DefaultIndexService of MDS modules in GT4. It aggregates registration information from the data resources as well as occupational matching resources

Data Service

OGSA-DAI dynamic data resources are configured and deployed in the GT4 runtime container. Customised OGSA-DAI activities are implemented to facilitate the deployment of new data resources within the dynamic data resource configuration. Furnishing metadata and managing the state of deployed data resources are done through these customised activities.

Each data resource registers with the index service upon deployment. Information such as the data activities supported, metadata, and schemas are used as part of the registration information.

Occupational Matching Services

There are 2 types of such services: the factory and instance. The factory service is responsible for creating occupational matching resources. The instance service provides the operations to manipulate and invoke matching functions of the resources.

The resources register to the index service upon creation. Occupational matching resources are also persistent even when the container shuts down. Configuration is required to define the location for persistency.

An occupational matching service/resource performs the translation of the client's data, based on the logic they are setup with.

2. Technical and Software Requirements

J2SE 1.5.0_07-b03

Apache ANT 1.6.5
 GT4.0.1 WS Core + MDS modules
 MySQL server 5.0.17
 StelsCSV JDBC driver. (Licensed)
 MySQL JDBC driver (mysql-connector-java-3.1.12-bin.jar)
 OGSA-DAI 2.2
 Jakarta Tomcat 5.5.14
 GridSphere 2.1.2

Installation/Migration

Run the setup MySQL backup script
 Migrate the directory where uncured resources are stored, and the persisted occupational matching resources.

3. Configuration

3.1 Default Configuration

Ports

Component	Port
GridSphere Portal (Tomcat)	8080
GT4 Container (Self-contained)	8888
mySQL server (Windows service)	3306

Java Heap Size

Component	Configuration
GT4 Container (Self-contained)	(environment variable) GLOBUS_OPTION=-Xmx768M
GridSphere Portal (Tomcat)	(environment variable) CATALINA_OPTS=-client -server -Xms512M -Xmx512M

Please refer to Section 3.2 Custom Configuration Information should there be requirements to run otherwise.

3.2 Custom Configuration Information

Web/Grid Portal

MIME for JNLP

Edit file: %CATALINA_HOME%/conf/web.xml

Add in MIME type

```

<mime-mapping>
  <extension>jnlp</extension>
  <mime-type>application/x-java-jnlp-file</mime-type>
</mime-mapping>
  
```

Port

%CATALINA_HOME%/conf/server.xml
Configure to desired port to run on. E.g. 8080 (default)

Property files configuration for application

Edit the following files:

1. feedbackParams.properties
 - a. jdbcURL → The URL to the location of MySQL database that stores feedback table
 - b. jdbcUser → User id for the MySQL database (must have access to schema/table)
 - c. jdbcPassword → User password
2. indexConfig.properties
 - a. indexURI → Change to URL of the DefaultIndexService
3. occMatchConfig.properties
 - a. jdbcURL → The URL of the MySQL database that stores occmatchresources table
 - b. jdbcUser → The User id that can access the table
 - c. jdbcPassword → User password
 - d. jnlpReplaceArgValue → Change to the host that runs GEODEDownloadEPRServlet (geode-portal)
 - e. jnlpLaunchWebStartURL → Change to the host that runs GEODELaunchWebStartServlet (geode-portal)
 - f. occMatchFactoryServiceURL → Change to the host that runs the GEODE grid services (GEODEOccMatchFactoryService)
4. searchParams.properties
 - a. saveAccessURL → Change to the host where GEODEDownloadServlet is hosted (geode-portal)
 - b. savePath → The path where the uncurated resources files are stored
 - c. jdbcURL → The URL to the MySQL database
 - d. jdbcUser → User ID
 - e. jdbcPassword → Password
 - f. indexURI → URL to the DefaultIndexService
 - g. uncuratedResourceInfoURL → The URL to the servlet GEODEViewUncuratedDetailsServlet.
5. uncuratedParams.properties
 - a. saveAccessURL → Change to the host where GEODEDownloadServlet is hosted
 - b. savePath → The path where the uncurated resources files are stored
 - c. jdbcURL → Th URL to the MySQL database
 - d. jdbcUser → User ID
 - e. jdbcPassword → Password

Set environment variable

CATALINA_OPTS=-client -server -Xms512M -Xmx512M

Web Start

Web start activated in the following manner:

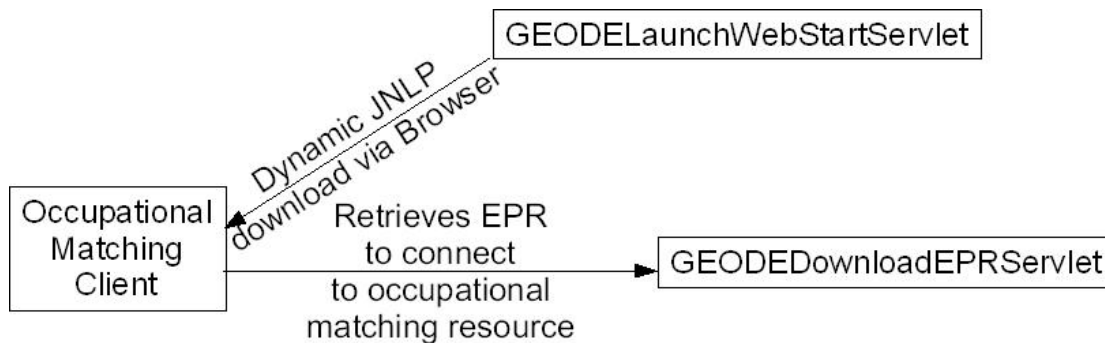


Figure 4. Flow of GEODE Web Start - Occupational Matching Client

Upon clicking a hyperlink to launch the Occupational Matching Client to a designated occupational matching resource, it actually activate a JNLP download from the GEODELaunchWebStartServlet. The GEODELaunchWebStartServlet replaces placeholders in a template JNLP file which consist of configuration to launch the application. When the application is launched on the user's

Directory where the Java Web Start JAR files are located is:

- %CATALINA_HOME%/webapps/ROOT/occmatch-webstart

Replace all the full qualified host names to the name of the server that hosts the GEODE grid services

Files to be edited:

1. java-activation.jnlp
2. javamail.jnlp
3. jce-jdk13-125.jnlp
4. occmatch.jnlp

- %CATALINA_HOME%/webapps/geode-portal/WEB-INF/classes

Replace all the full qualified host names to the name of the server that hosts the GEODE grid services

Files to be edited:

1. occmatch.jnlp

Servlets in GEODE

There are 3 servlets, all are configured in geode-portal context which is a portlet:

1. GEODELaunchWebStartServlet
2. GEODEDownloadEPRServlet
3. GEODEDownloadServlet

The purpose for GEODELaunchWebStartServlet and GEODEDownloadEPRServlet has been described above. GEODEDownloadServlet handles downloads of uncurated data resources (as uploaded by users, e.g. zip, spss, dat etc). These servlets are found in %CATALINA_HOME%/webapps/geode-portal/WEB-INF/lib/geode-portal-services.jar.

The GEODELaunchWebStartServlet uses the %CATALINA_HOME%/webapps/geode-portal/WEB-INF/classes/occmatchConfig.properties for configuration (see above).

GEODEDownloadEPRServlet uses the JDBC properties in %CATALINA_HOME%/webapps/geode-portal/WEB-INF/classes/occMatchConfig.properties for configuration. It reaches the mySQL database and retrieves the respective EPR using the given resourceID.

The GEODEDownloadServlet configuration uses %CATALINA_HOME%/webapps/geode-portal/WEB-INF/classes/uncuratedParams.properties (see above).

Existing Data Resources

Configuration for data resources (location of files)

Edit ogsadai_wsrf/server-config.wsdd

1. dai.config.files.root (edit accordingly
%GLOBUS_LOCATION%/etc/ogsadai_wsrf)
2. dai.schema.files.root (edit accordingly
%GLOBUS_LOCATION%/share/schema/ogsadai)

For each deployed resource, edit the *.files

Search replace e.g. “C:” → Change accordingly to the location where the schema is held

1. sessionConfig.xml
2. dataResourceConfig.xml
3. dataResourceClassConfig.xml
4. DatabaseRoles.xml
5. activityConfig.xml

Data Resource Templates/Skeletons

Configuration for data resources (location of files)

Any absolute locations are to be changed in *.xml

There are 2 kinds of data (OGSA-DAI) resources in GEODE, access via 1 service: GEODEFactoryResource, database or csv data resources. GEODEFactoryResource is a static-deployed resource configured with the capability for deploying dynamic database or csv data resources in the same service. The dynamic data resources are the resources that virtualise databases such as mySQL and CSV format files.

For GEODEFactoryResource

File/Directory	Purpose
activityConfig.xml	Configured with 1 activity (geodeCreateResource) for deploying/creating dynamic data resource. The activity schema is based on geode_create_db.xsd found in %GLOBUS_LOCATION%/share/schema/ogsadai/xsd/activities/geode_create_db.xsd. It is configured with the class uk.ac.stir.geode.activities.GEODECreateDBResourceActivity found in geode.jar
dataResourceClassConfig.xml	Configured with uk.ac.stir.geode.dataresource.GEODECreateDBDataResource

	founding geode.jar
sessionConfig.xml	Default OGSA-DAI configuration
geodeResourceConfig.xml	Configured with custom information describing the factory resource. It is according to the schema found in %GLOBUS_LOCATION%/share/schema/ogsadai/xsd/geode_resource_config.xsd
(dir) instance_DB	Contains skeleton configuration templates for deploying relational databases such as MySQL. See next table
(dir) instance_DB_TXT	Contains skeleton configuration templates for deploying CSV formatted files. See next table.

For instance_DB and instance_DB_TXT directories

	File Name	Purpose
1.	sessionConfig.xml	Default OGSA-DAI configuration
2.	dataResourceConfig.xml	Contains template (with placeholders) used for deploying data resources
3.	DatabaseRoles.xml	Configure accordingly to OGSA-DAI documentation
4.	activityConfig.xml	Contains the GEODE customised activities for functions such as accessing DDI, updating matching logic. Custom activities are prefix with “geode”. The activity Java classes are found in geode.jar Activity configuration for instance_DB and instance_DB_TXT differ.
5.	mds-registration.xml	Configure accordingly to OGSA-DAI documentation. Added the registration of DDI of data resource
6.	ddi.xml	It will contain the DDI XML schema for the data resource it is representing
7.	csvFileList.xml	It stores the list of CSV files the data resource represents. Useful for CSV files available via HTTP. This is because it is not known how many files are under that HTTP location. So this configuration list semi-statically the files so that the database schema can be worked out
8.	addIndexFileConfig.xml	Default OGSA-DAI configuration

A occ_match_logic.xml file will be created upon using the geodeUpdateOccMatchLogic activity of data resources.

Globus Container

JVM Heap Size of GT4 container

Set GLOBUS_OPTION=-Xmx768M

GEODE Grid services WSDD

Edit server-config.wsdd of geode_wsrf_services_occ_match

- Under GEODE/ GEODEOccMatchService
GEODEInstanceRegistration → Configure full path to registration.xml
- Under GEODE/GEODEOccMatchFactoryService
GEODEFactoryRegistration → Configure full path to registration.xml
GEODEInstanceRegistration → Configure full path to instance_registration.xml

Application properties

Edit GEODEOccMatchResourceProp.properties in
GEODEOccMatchResourceProp.jar

1. GEODEOccMatchPersistDirectory → The location where occupational matching resources are persisted

Edit MyServiceHostProp.properties in MyServiceHostProp.jar

This file is for MyServiceHost to overwrite the default protocol and port when attempting to work out protocol and port when deployed in Tomcat. This is because the “transport.url” property in the MessageContext is absent when in Tomcat.

1. protocol → Configure the protocol for the grid services. HTTP
2. host → Machine name of the grid services
3. port → Port number

If WSRF deployed into Tomcat

There is a requirement to overwrite the attributes in the container that determines the protocol and the port number (ServiceHost class) if the Tomcat container is to start with a non-https and not 8443 port.

5. Tested

Platform(s):
Windows XP

JRE
1.5.0_11

No JUNIT test suite yet