



IP-FP6-015964

AEOLUS

Algorithmic Principles for Building Efficient Overlay Computers

Deliverable D6.4.3

Demo software for application scenario: Final release

Responsible Partner: Radiolabs (I)
Report Preparation Date: February 2010

Contract Start Date: 01/09/05 Duration: 54 months
Project Co-ordinator: University of Patras (EL)

Table of contents

1 Introduction.....	3
2 Release notes.....	3
3 AEOFORGE architecture.....	4
4 AEOLUS Project contributions included.....	5
4.1 FS.....	5
4.2 STA.....	5
4.3 SSOD.....	5
4.4 Scheduler.....	6
4.5 JXTACh.....	6
5 AEOFORGE setup.....	7
5.1 the testbed.....	7
5.2 the client.....	15
6 AEOFORGE APIs.....	15
6.1 Sequence Diagrams.....	19
6.1.1 add.....	20
6.1.2 ci.....	21
6.1.3 co.....	22
6.1.4 compile.....	22
6.1.5 del.....	23
6.1.6 describe.....	24
6.1.7 download.....	24
6.1.8 execute.....	25
6.1.9 find.....	25
6.1.10 noop.....	26
6.1.11 quit.....	26
6.1.12 Upload.....	27

Deliverable D6.4.3 concerns the final version of the demo application AEOFORGE. A demonstration will take place during the review meeting.

1 Introduction

AEOFORGE is a prototype of a „peer-to-peer sourceforge“: it allows users to share, control and manage software development. The final version of AEOFORGE adds further functionalities to those already implemented thanks to the AEOLUS' partners contribution: among them compilation and execution of java projects stored on the forge.

AEOFORGE provides its users with a simple command line interface to access a distributed file storage where software projects are hosted to be further retrieved, edited, compiled or executed.

The prototype demo software server component runs on the AEOLUS testbed, together with the integrated external services.

The client component comes as a zip file the user must unzip on his local machine

AEOFORGE-Client.zip

The 3rd year review demo showed how to upload a project tarball file, download it, edit a project's file and store it back in the forge avoiding concurrent modifications. This year the same activities will be performed working on the testbed together with the newly introduced ones.

2 Release notes

In this chapter we provide a description of all changes related to this final version of AEOFORGE prototype.

- All services integrated in AEOFORGE to act as the server-side component run on the AEOLUS testbed
- JxtaSocket are used instead of JxtaBiDiPipe to avoid limitations on messages size
- All communication streams among components are buffered (BufferedInputStream and BufferedOutputStream instead of InputStream and OutputStream) increasing efficiency
- Latencies on sockets and I/O channels closures have been removed
- New AEOLUS project partners' contributions integrated
 - SSOD
 - Scheduler
 - JXTACh
- New commands/functionalities added
 - del
 - find
 - compile a Java project
 - execute a Java project
- Project download and upload improved: all operations now run in parallel
- On upload failure projects partially uploaded are thoroughly removed from the storage

3 AEOFORGE architecture

The main improvement in AEOFORGE development has been the complete integration with the AEOLUS testbed. All services bound to AEOFORGE functionalities (exception made for the client component that runs on end users machines) are all testbed services and thus all serve, concurrently, the n user clients.

Figure 1 shows the actual setup of a testbed partition hosting services used by the AEOFORGE application:

- AEOFORGE service
 - the server-side component that, together with the client, makes up the core of the AEOFORGE application
- STA
 - the component in charge of assigning a unique and secure timestamp to all files under AEOFORGE control
 - is made of
 - 1 verifier node
 - 1 registry node
 - n authority nodes
- FS
 - the file storage component
 - made of
 - 1 coordinator node that „coordinates“ the
 - n storage nodes hosting users' files
- SSOD
 - Secure Searching on Outsourced Data, to search among AEOFORGE projects with a given research key
- Scheduler
 - the component in charge of assign the compilation and execution tasks (later described) to the less loaded dedicated testbed node
- n compiler nodes
 - testbed nodes dedicated to java sources compilation
- n executor nodes
 - testbed nodes dedicated to java programs execution

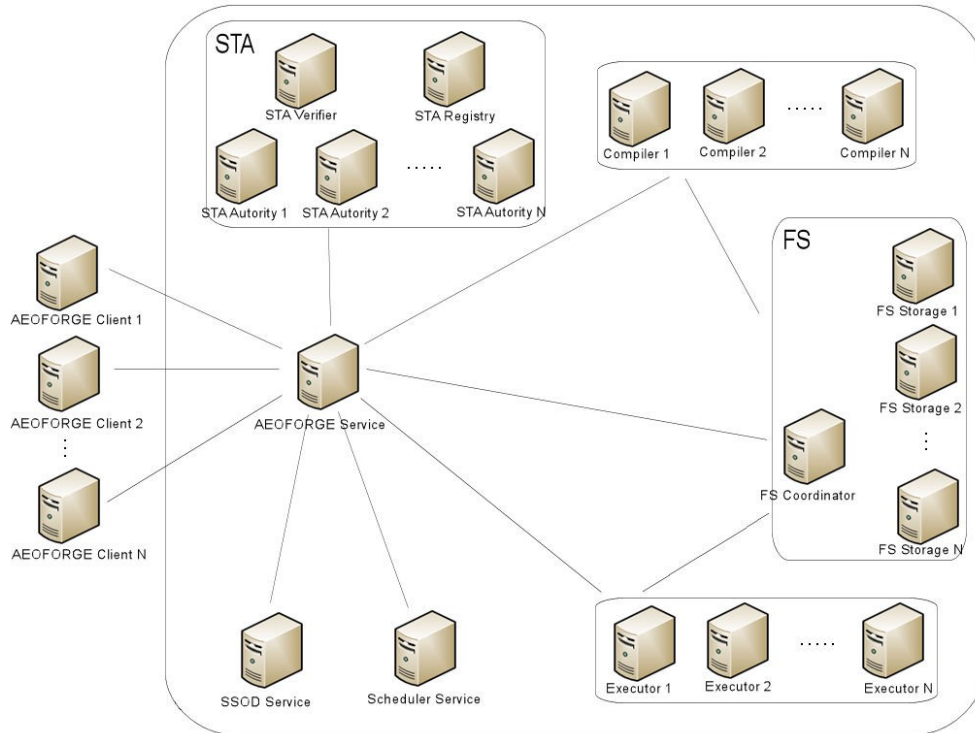


Figure 1: AEOFORGE architecture

4 AEOLUS Project contributions included

This section briefly describes the role of the services, running on the testbed and developed by AEOLUS project partners, that have been integrated in AEOFORGE.

4.1 FS

FS services offer filesystem-like functionalities to AEOFORGE application so that all the storage offered by the dedicated nodes (i.e. storage nodes) appear to the application as a cumulative and distributed filesystem. AEOFORGE service contacts FS Coordinator each time a read/write action must be performed not caring where the files are actually stored: the Coordinator service, being aware of all the nodes offering a storage area with a given capacity and duration, takes the weight of filesystem operations off the AEOFORGE service receiving files and routing them to the node it decides being the best in terms of available storage space.

4.2 STA

Secure Timestamping Authority services have been integrated in the AEOFORGE application to assign to each file stored on the forge a unique, and secure, timestamp. The timestamp is updated each time a file is modified to avoid concurrent modifications on files modified by different users.

4.3 SSOD

Secure Search on Outsourced Data services have been integrated to allow an AEOFORGE users to query the AEOFORGE projects database to search for a given project.

4.4 Scheduler

Scheduler services have been integrated to manage compilation and execution tasks of java projects stored on AEOFORGE.

Some nodes of the testbed have been setup to act as java compilers and executors; Scheduler is aware of the current load on those nodes and their actual load limits so that when AEOFORGE Service needs to compile or execute a project the best choice is made on the basis of the Scheduler decisions.

4.5 JXTACh

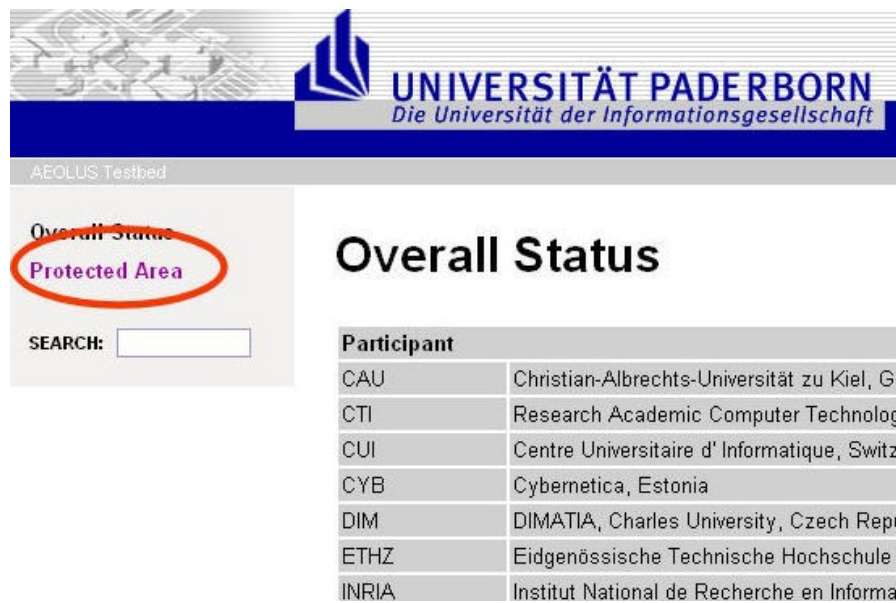
The JXTACh contribution is completely transparent to the end user as it consists in running the AEOFORGE service on a AEOLUS testbed partition replacing the JXTA rendezvous protocol implementation, namely the loosely consistent DHT, with the Chord algorithm to improve the JXTA rendezvous service performances.

5 AEOFORGE setup

Assuming we have a running instance of the AEOLUS testbed we must perform few steps to setup the environment that will serve AEOFORGE clients' requests

5.1 the testbed

- 1 With a web browser follow the link to the chosen testbed installation:
<http://aeolus.cs.upb.de>
- 2 Follow the „Protected Area“ link providing the user credentials (username, password)



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

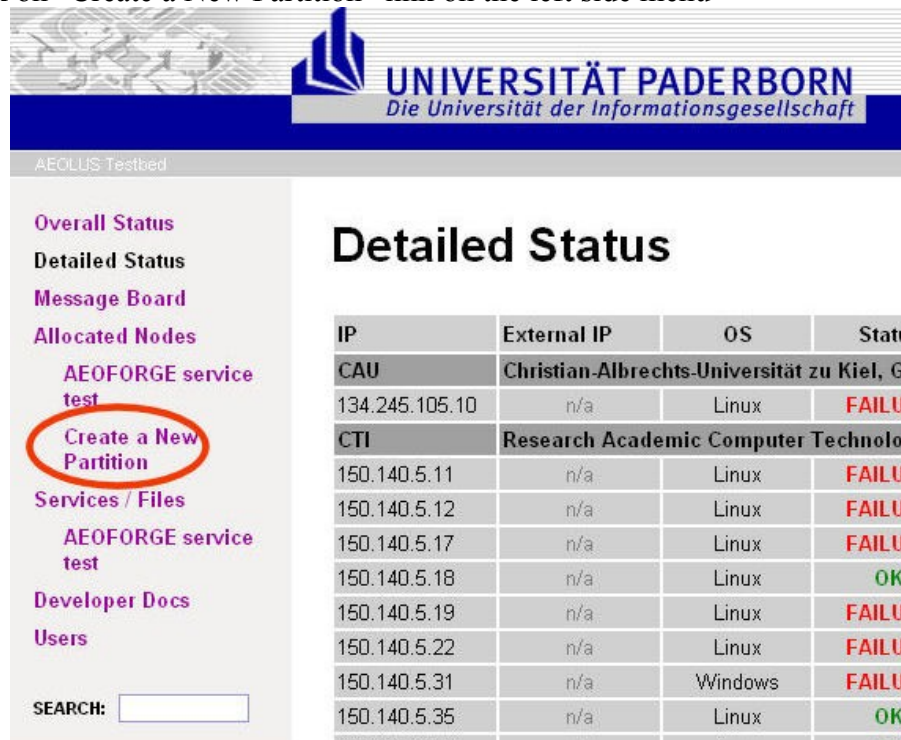
AEOLUS Testbed

Overall Status
Protected Area

SEARCH:

Participant	
CAU	Christian-Albrechts-Universität zu Kiel, G
CTI	Research Academic Computer Technolog
CUI	Centre Universitaire d' Informatique, Switz
CYB	Cybernetica, Estonia
DIM	DIMATIA, Charles University, Czech Repu
ETHZ	Eidgenössische Technische Hochschule
INRIA	Institut National de Recherche en Informa

- 3 Click on “Create a New Partition” link on the left side menu



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

AEOLUS Testbed

Overall Status
Detailed Status
Message Board
Allocated Nodes
AEOFORGE service test
Create a New Partition
Services / Files
AEOFORGE service test
Developer Docs
Users

SEARCH:

IP	External IP	OS	Statu
CAU Christian-Albrechts-Universität zu Kiel, G			
134.245.105.10	n/a	Linux	FAILU
CTI Research Academic Computer Technolo			
150.140.5.11	n/a	Linux	FAILU
150.140.5.12	n/a	Linux	FAILU
150.140.5.17	n/a	Linux	FAILU
150.140.5.18	n/a	Linux	OK
150.140.5.19	n/a	Linux	FAILU
150.140.5.22	n/a	Linux	FAILU
150.140.5.31	n/a	Windows	FAILU
150.140.5.35	n/a	Linux	OK

- 4 Allocate one physical node for each of the following services:
 - AEOFORGE Service
 - FS Coordinator
 - STA Verifier
 - STA Registry
 - SSOD Service
 - SCHEDULER Service
- 5 Allocate one node, at least, physical or virtual for the following services
 - AEOFORGE Compiler
 - AEOFORGE Executor
 - FS Storage
 - STA Authority

Create a New Partition

You can create a new testbed partition here. Once it is created, you can access it via the menu on the left (a submenu of "Allocated Nodes" will be created).

IP	External IP	OS	Status	Load ¹	CPU / Mem.	Allocate
CTI Research Academic Computer Technology Institute, Greece (ichatz@cti.gr , chita@cti.gr)						
150.140.5.11	n/a	Linux	FAILURE	n/a	667 MHz / 768 MB	don't allocate
150.140.5.18	n/a	Linux	OK	36.49 (13:32)	2.6 GHz / 256 MB	4 virtual instances
150.140.5.22	n/a	Linux	FAILURE	n/a	733 MHz / 1 GB	don't allocate
150.140.5.37	n/a	Windows	FAILURE	n/a	2x 1 GHz / 1.75 GB	don't allocate
ETHZ Eidgenössische Technische Hochschule Zürich, Switzerland (spyros@inf.ethz.ch , pautasso@inf.ethz.ch)						
129.132.43.177	n/a	Linux	FAILURE	n/a	2x 1.4 GHz / 1 GB	don't allocate
129.132.43.178	n/a	Linux	FAILURE	n/a	2x 1.4 GHz / 1 GB	don't allocate
129.132.43.179	n/a	Linux	FAILURE	n/a	2x 1.4 GHz / 1 GB	don't allocate
RAL Radio Labs, Italy (marco.leo@uniroma2.it , daniele.paiella@nexse.com)						
160.80.85.71	n/a	Linux	FAILURE	n/a	1.8 GHz / 1 GB	don't allocate
160.80.85.72	n/a	Linux	FAILURE	n/a	n/a	don't allocate
UDRTV Università degli Studi di Roma "Tor Vergata", Italy (rossig@mat.uniroma2.it)						
160.80.11.142	n/a	Windows	OK	1 % (13:15)	1.7 GHz / 768 MB	just 1 physical instance

- 6 Assign a name to the newly created partition and confirm your choices clicking on „Create the Partition“ button

Description of your partition:

public (tick this checkbox to make your partition accessible by all testbed users)

[Index A – Z](#) | [Imprint](#) | [Webmaster](#)

7 Manage the partition services with the left side menu

Services / Files

Services implementing the testbed API are automatically loaded if the fully qualified classname is checked below and the jar file containing the relevant classes is uploaded below. Optionally, you may upload configuration files (plain text) which will be stored under the `configs` subdirectory of the working directory of the edge peers. In these files, the following special variables are automatically replaced:

<code>__IP__</code>	IP address of the edge peer
<code>__EXTERNAL_IP__</code>	external IP address of the edge peer (or the same as <code>__IP__</code> if this is already a public address)
<code>__JXTA_PORT__</code>	the port used for JXTA
<code>__PUBWEB_PORT__</code>	the port used for PUB-Web
<code>__ABSOLUTE_PATH__</code>	absolute path name (may contain spaces!) of the working directory (with trailing (back-slash)
<code>__ABSOLUTE_PATH_ESC__</code>	the same as <code>__ABSOLUTE_PATH__</code> but with double backslashes on Windows Systems
<code>__JAVA__</code>	path (may contain spaces!) of the java executable
<code>__JAVA_ESC__</code>	the same as <code>__JAVA__</code> but with double backslashes on Windows Systems
<code>__L1_CACHE__</code>	L1 cache size in kB
<code>__L2_CACHE__</code>	L2 cache size in kB

Finally, you may upload arbitrary other files, such as, e.g., JNI native libraries. These files will not be modified and are stored in the `libs` subdirectory. Beside the `configs` and the `libs` subdirectories, the subdirectories `.jxta`, `tmp`, and `var` are the only subdirectories where you can read / modify / delete files.

Note: Once you've completed your service configuration and file uploads, it's necessary to restart your testbed partition.

8 Upload all libraries and resources needed following the “upload a new file” link

Libraries and Configuration Files:

To view the configuration files or download the library files, please follow the corresponding links. Please note that some browsers cache these downloads, i.e. if you see old contents when viewing a configuration file, please press your browser's reload button.

Filename	Actions	Upload to these hosts
upload a new file		

9 Upload the following, provided within the client zip file, libraries:

- `aeoforge.jar`
- `aeolus.jar`
- `aeolusShellExtension.jar`
- `bcprov-jdk14.jar`
- `cia.jar`
- `Cia_fat.jar`
- `ClientApi.jar`
- `commons-cli-1.1.jar`
- `commons-net-1.4.1.jar`
- `derby.jar`
- `derbytools.jar`
- `javax.servlet.jar`
- `jsch-0.1.41.jar`
- `junit-4.5.jar`
- `jxta.jar`
- `jxtashell.jar`
- `log4j-1.2.14.jar`
- `org.mortbay.jetty.jar`
- `ssod-aeolus-1.0.0.jar`
- `sta.jar`
- `TBAeolusScheduler.jar`

10 Upload the following, provided, configuration files:

- a_603_181.properties
- compiler_config.txt
- cramerShoup.parameters
- executor_config.txt
- log4j.xml
- mystorage_config.txt
- ssod.key
- sta0.staconfig
- staKeyStore.p12
- v0.stavconfig

Files

Upload a library (.jar) file:

 Sfoglia...

Upload

Upload a configuration file (plain text):

 Sfoglia...

Upload

Upload another type of file (e.g. a JNI shared library):

 Sfoglia...

Upload

Note: The maximum filesize is 20 MB. Files greater than this will be ignored.

[Index A - Z](#) | [Imprint](#) | [Webmaster](#)

11 Start registering the services following the “Services/File” menu

Services

Add your services implementing the `aeolus.edge.Service` interface (either by extending the classes `aeolus.edge.BasicService` or `aeolus.edge.AdvancedService` together with the `aeolus.edge.ServiceImpl` interface, or using a custom implementation -- see [developer documentation](#) for further information) to the global list of services below and activate them in the sub-webpage for your testbed partition.

Global list of services:

Classname	User	Actions
aeolus.unipd.acg.testbed.BenchService	Paolo Bertasi	
helloWorld.HelloWorldServer	Stefan Schiffner	
it.unisa.aeolus.service.sfe.yao.SFEMultipartyMediatorImpl	Aniello del Sorbo	
it.unisa.aeolus.service.sta.STARegistryServiceImpl	Aniello del Sorbo	
it.unisa.aeolus.service.sta.STAServiceImpl	Aniello del Sorbo	
it.unisa.aeolus.service.sta.STAVerifierServiceImpl	Aniello del Sorbo	
it.unisa.cia.jxta.CertifiedDBOwner	Aniello del Sorbo	

12 Skipping the services that have been already registered follow the link “Add a new service”

sybilGuard.EmptyService	Stefan Schiffner	
sybilGuard.management.SgMmService	Joachim Gehweiler	
sybilGuard.management.SgMmSupernodeService	Joachim Gehweiler	
sybilGuard.SgService	Joachim Gehweiler	
udrtv.jxtachtest	Carlo Nocentini	
webdust.net.aeolus.discovery.DiscoveryListenerService	Ioannis Chatzigiannakis	
webdust.net.aeolus.discovery.DiscoveryService	Ioannis Chatzigiannakis	

[Add a new service](#)

and start adding the following services:

- `it.unisa.aeolus.service.sta.STARegistryServiceImpl`
- `it.unisa.aeolus.service.sta.STAServiceImpl`
- `it.unisa.aeolus.service.sta.STAVerifierServiceImpl`
- `it.unisa.dia.gas.ssod.aeolus.service.SearchableStoreServiceImpl`
- `net.jxta.impl.shell.bin.aeoforge.compiler`
- `net.jxta.impl.shell.bin.aeoforge.executor`
- `net.jxta.impl.shell.bin.aeoforge.service`
- `net.jxta.impl.shell.bin.fs.coordinator`
- `net.jxta.impl.shell.bin.fs.my_storage`
- `net.jxta.impl.shell.bin.sched.TBAeolusScheduler`.

Services

Create a New Service:

Save

[Index A – Z](#) | [Imprint](#) | [Webmaster](#)

- 13 Assign the services to the nodes defined in steps 4 and 5: AEOFORGE Service, FS Coordinator, STA Verifier, STA Registry, SSOD Service, SCHEDULER Service must be assigned to one physical node each while AEOFORGE Compiler, AEOFORGE Executor, FS Storage, STA Authority must be assigned to, at least, one physical or virtual node

it.unisa.dia.gas.ssod.aeolus.service.SearchableStoreServiceImpl	<input type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 147.162.96.123	<input type="checkbox"/> 131.234.63.24 <input type="checkbox"/> 147.162.96.48	<input type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 150.140.139.173	<input type="checkbox"/> 138.96.254.45 <input type="checkbox"/> 150.140.5.8	<input checked="" type="checkbox"/> 147.162.96.109 <input type="checkbox"/> 172.16.14.5	<input type="checkbox"/> 147.162.96.113 check all / none
it.unisa.aeolus.service.sta.STARegistryServiceImpl	<input type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 147.162.96.123	<input type="checkbox"/> 131.234.63.24 <input type="checkbox"/> 147.162.96.48	<input type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 150.140.139.173	<input type="checkbox"/> 138.96.254.45 <input checked="" type="checkbox"/> 150.140.5.8	<input type="checkbox"/> 147.162.96.109 <input type="checkbox"/> 172.16.14.5	<input type="checkbox"/> 147.162.96.113 check all / none
it.unisa.aeolus.service.sta.STAServletImpl	<input type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 147.162.96.123	<input type="checkbox"/> 131.234.63.24 <input checked="" type="checkbox"/> 147.162.96.48	<input type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 150.140.139.173	<input type="checkbox"/> 138.96.254.45 <input type="checkbox"/> 150.140.5.8	<input type="checkbox"/> 147.162.96.109 <input type="checkbox"/> 172.16.14.5	<input checked="" type="checkbox"/> 147.162.96.113 check all / none
it.unisa.aeolus.service.sta.STAVerifierServiceImpl	<input type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 147.162.96.123	<input type="checkbox"/> 131.234.63.24 <input type="checkbox"/> 147.162.96.48	<input type="checkbox"/> 138.96.254.42 <input checked="" type="checkbox"/> 150.140.139.173	<input type="checkbox"/> 138.96.254.45 <input type="checkbox"/> 150.140.5.8	<input type="checkbox"/> 147.162.96.109 <input type="checkbox"/> 172.16.14.5	<input type="checkbox"/> 147.162.96.113 check all / none
net.jxta.impl.shell.bin.aeoforge.compiler	<input type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 147.162.96.123	<input checked="" type="checkbox"/> 131.234.63.24 <input type="checkbox"/> 147.162.96.48	<input type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 150.140.139.173	<input type="checkbox"/> 138.96.254.45 <input type="checkbox"/> 150.140.5.8	<input type="checkbox"/> 147.162.96.109 <input type="checkbox"/> 172.16.14.5	<input type="checkbox"/> 147.162.96.113 check all / none
net.jxta.impl.shell.bin.aeoforge.executor	<input checked="" type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 147.162.96.123	<input type="checkbox"/> 131.234.63.24 <input type="checkbox"/> 147.162.96.48	<input type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 150.140.139.173	<input type="checkbox"/> 138.96.254.45 <input type="checkbox"/> 150.140.5.8	<input type="checkbox"/> 147.162.96.109 <input type="checkbox"/> 172.16.14.5	<input type="checkbox"/> 147.162.96.113 check all / none
net.jxta.impl.shell.bin.aeoforge.service	<input type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 147.162.96.123	<input type="checkbox"/> 131.234.63.24 <input type="checkbox"/> 147.162.96.48	<input checked="" type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 150.140.139.173	<input type="checkbox"/> 138.96.254.45 <input type="checkbox"/> 150.140.5.8	<input type="checkbox"/> 147.162.96.109 <input type="checkbox"/> 172.16.14.5	<input type="checkbox"/> 147.162.96.113 check all / none
net.jxta.impl.shell.bin.fs.coordinator	<input type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 147.162.96.123	<input type="checkbox"/> 131.234.63.24 <input type="checkbox"/> 147.162.96.48	<input type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 150.140.139.173	<input checked="" type="checkbox"/> 138.96.254.45 <input type="checkbox"/> 150.140.5.8	<input type="checkbox"/> 147.162.96.109 <input type="checkbox"/> 172.16.14.5	<input type="checkbox"/> 147.162.96.113 check all / none
net.jxta.impl.shell.bin.fs.my_storage	<input type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 147.162.96.123	<input type="checkbox"/> 131.234.63.24 <input type="checkbox"/> 147.162.96.48	<input type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 150.140.139.173	<input type="checkbox"/> 138.96.254.45 <input type="checkbox"/> 150.140.5.8	<input checked="" type="checkbox"/> 172.16.14.5	<input type="checkbox"/> 147.162.96.113 check all / none
net.jxta.impl.shell.bin.sched.TBAeolusScheduler	<input type="checkbox"/> 131.234.63.111 <input checked="" type="checkbox"/> 147.162.96.123	<input type="checkbox"/> 131.234.63.24 <input type="checkbox"/> 147.162.96.48	<input type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 150.140.139.173	<input type="checkbox"/> 138.96.254.45 <input type="checkbox"/> 150.140.5.8	<input type="checkbox"/> 147.162.96.109 <input type="checkbox"/> 172.16.14.5	<input type="checkbox"/> 147.162.96.113 check all / none

- 14 Assign uploaded libraries and configuration files to nodes:

- all nodes:
 - aeolus.jar
 - bcprov-jdk14.jar
 - cia.jar
 - Cia_fat.jar
 - commons-cli-1.1.jar
 - commons-net-1.4.1.jar
 - javax.servlet.jar
 - jsch-0.1.41.jar
 - junit-4.5.jar
 - jxta.jar
 - jxtashell.jar
 - log4j-1.2.14.jar
 - org.mortbay.jetty.jar
- aeoforge.jar
 - AEOFORGE Service
 - Compiler
 - Executor
- aeolusShellExtension.jar
 - FS Coordinator
 - FS Storage
- ClientApi.jar
 - AEOFORGE Service
- derby.jar, derbytools.jar
 - SSOD Service
 - STA Registry
 - STA Authority
 - STA Verifier
- ssod-aeolus-1.0.0.jar
 - AEOFORGE Service

- SSOD Service
- sta.jar
 - AEOFORGE Service
 - STA Registry
 - STA Authority
 - STA Verifier
- TBAeolusScheduler.jar
 - SCHEDULER
- a_603_181.properties
 - AEOFORGE Service
 - SSOD Service
- compiler_config.txt
 - Compiler
- cramerShoup.parameters
 - AEOFORGE Service
 - SSOD Service
- executor_config.txt
 - AEOFORGE Executor
- log4j.xml
 - SSOD Service
 - STA Registry
 - STA Authority
 - STA Verifier
- mystorage_config.txt
 - FS Storage
- ssod.key
 - AEOFORGE Service
 - SSOD Service
- staKeyStore.p12
 - STA Registry
 - STA Authority
 - STA Verifier
- v0.stavconfig
 - STA Verifier
- sta0.staconfig (sta1.staconfig, 2, 3 etc...)
 - STA Authority

Libraries and Configuration Files:

To view the configuration files or download the library files, please follow the corresponding links. Please note that some browsers cache these downloads, i.e. if you see old contents when viewing a configuration file, please press your browser's reload button.

Filename	Actions	Upload to these hosts
configs/a_603_181.properties	delete	<input type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 131.234.63.24 <input checked="" type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 138.96.254.45 <input checked="" type="checkbox"/> 147.162.96.109 <input type="checkbox"/> 147.162.96.113 <input type="checkbox"/> 147.162.96.123 <input type="checkbox"/> 147.162.96.48 <input type="checkbox"/> 150.140.139.173 <input type="checkbox"/> 150.140.5.8 <input type="checkbox"/> 172.16.14.5 check all / none
configs/compiler_config.txt	delete	<input type="checkbox"/> 131.234.63.111 <input checked="" type="checkbox"/> 131.234.63.24 <input type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 138.96.254.45 <input type="checkbox"/> 147.162.96.109 <input type="checkbox"/> 147.162.96.113 <input type="checkbox"/> 147.162.96.123 <input type="checkbox"/> 147.162.96.48 <input type="checkbox"/> 150.140.139.173 <input type="checkbox"/> 150.140.5.8 <input type="checkbox"/> 172.16.14.5 check all / none
configs/cramerShoup.parameters	delete	<input type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 131.234.63.24 <input checked="" type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 138.96.254.45 <input checked="" type="checkbox"/> 147.162.96.109 <input type="checkbox"/> 147.162.96.113 <input type="checkbox"/> 147.162.96.123 <input type="checkbox"/> 147.162.96.48 <input type="checkbox"/> 150.140.139.173 <input type="checkbox"/> 150.140.5.8 <input type="checkbox"/> 172.16.14.5 check all / none
configs/executor_config.txt	delete	<input checked="" type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 131.234.63.24 <input type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 138.96.254.45 <input type="checkbox"/> 147.162.96.109 <input type="checkbox"/> 147.162.96.113 <input type="checkbox"/> 147.162.96.123 <input type="checkbox"/> 147.162.96.48 <input type="checkbox"/> 150.140.139.173 <input type="checkbox"/> 150.140.5.8 <input type="checkbox"/> 172.16.14.5 check all / none
configs/log4j.xml	delete	<input type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 131.234.63.24 <input type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 138.96.254.45 <input checked="" type="checkbox"/> 147.162.96.109 <input checked="" type="checkbox"/> 147.162.96.113 <input type="checkbox"/> 147.162.96.123 <input checked="" type="checkbox"/> 147.162.96.48 <input checked="" type="checkbox"/> 150.140.139.173 <input checked="" type="checkbox"/> 150.140.5.8 <input type="checkbox"/> 172.16.14.5 check all / none
configs/myoffer_config.txt	delete	<input type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 131.234.63.24 <input type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 138.96.254.45 <input type="checkbox"/> 147.162.96.109 <input type="checkbox"/> 147.162.96.113 <input type="checkbox"/> 147.162.96.123 <input type="checkbox"/> 147.162.96.48 <input type="checkbox"/> 150.140.139.173 <input type="checkbox"/> 150.140.5.8 <input checked="" type="checkbox"/> 172.16.14.5 check all / none
configs/ssod.key	delete	<input type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 131.234.63.24 <input checked="" type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 138.96.254.45 <input checked="" type="checkbox"/> 147.162.96.109 <input type="checkbox"/> 147.162.96.113 <input type="checkbox"/> 147.162.96.123 <input type="checkbox"/> 147.162.96.48 <input type="checkbox"/> 150.140.139.173 <input type="checkbox"/> 150.140.5.8 <input type="checkbox"/> 172.16.14.5 check all / none
configs/sta0.staconfig	delete	<input type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 131.234.63.24 <input type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 138.96.254.45 <input type="checkbox"/> 147.162.96.109 <input type="checkbox"/> 147.162.96.113 <input type="checkbox"/> 147.162.96.123 <input checked="" type="checkbox"/> 147.162.96.48 <input type="checkbox"/> 150.140.139.173 <input type="checkbox"/> 150.140.5.8 <input type="checkbox"/> 172.16.14.5 check all / none
configs/sta1.staconfig	delete	<input type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 131.234.63.24 <input type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 138.96.254.45 <input type="checkbox"/> 147.162.96.109 <input checked="" type="checkbox"/> 147.162.96.113 <input type="checkbox"/> 147.162.96.123 <input type="checkbox"/> 147.162.96.48 <input type="checkbox"/> 150.140.139.173 <input type="checkbox"/> 150.140.5.8 <input type="checkbox"/> 172.16.14.5 check all / none
configs/staKeyStore.p12	delete	<input type="checkbox"/> 131.234.63.111 <input type="checkbox"/> 131.234.63.24 <input type="checkbox"/> 138.96.254.42 <input type="checkbox"/> 138.96.254.45 <input type="checkbox"/> 147.162.96.109 <input checked="" type="checkbox"/> 147.162.96.113 <input type="checkbox"/> 147.162.96.123 <input checked="" type="checkbox"/> 147.162.96.48 <input checked="" type="checkbox"/> 150.140.139.173 <input checked="" type="checkbox"/> 150.140.5.8 <input type="checkbox"/> 172.16.14.5 check all / none

15 Apply changes

User management:

Partition owner	Stefano Caliano		
Admitted users	<input type="checkbox"/> Andrea Pietracaprina	<input type="checkbox"/> Angelo De Caro	<input type="checkbox"/> Aniello del Sorbo
	<input type="checkbox"/> Benjamin Potyka	<input type="checkbox"/> Carlo Nocentini	<input type="checkbox"/> Christos Kaklamanis
	<input type="checkbox"/> Christos Koninis	<input checked="" type="checkbox"/> Daniele Paiella	<input type="checkbox"/> Dmitri Akatov
	<input type="checkbox"/> Domenico Gambelli	<input type="checkbox"/> Efi Chita	<input type="checkbox"/> Emiliano Casalichio
	<input type="checkbox"/> Florian Diedrich	<input type="checkbox"/> Geppino Pucci	<input type="checkbox"/> Giovanni Cortese
	<input type="checkbox"/> Gunnar Schomaker	<input type="checkbox"/> Ioannis Caragiannis	<input type="checkbox"/> Ioannis Chatzigiannakis
	<input type="checkbox"/> Joachim Gehweiler	<input type="checkbox"/> Luis Javier de la Garza Trevino	<input type="checkbox"/> Marina Drosou
	<input type="checkbox"/> Mauro Sozio	<input type="checkbox"/> Michel Syska	<input type="checkbox"/> Nikos Karanikolas
	<input type="checkbox"/> Panagiotis Kanellopoulos	<input type="checkbox"/> Paolo Bertasi	<input type="checkbox"/> Remi Vannier
	<input type="checkbox"/> Silvano Riz	<input type="checkbox"/> Stefan Schiffner	<input type="checkbox"/> Stefano Braghin
	<input type="checkbox"/> Tomas Chudlarsky	<input type="checkbox"/> Ugo Colesanti	<input type="checkbox"/> UlfPeter Schröder
	<input type="checkbox"/> Ulrich Ahlers		check all / none
	<input type="checkbox"/> Provide to all testbed users (read only)		

[Apply Changes](#)

16 Restart the partition in the node management section

Rendezvous Peer:

IP	Port	Actions
131.234.65.73	9710	logfile

In case you are using additional peers outside the testbed, please remember to delete your local `.jxta` folder when restarting.

[\(Re-\)start this Partition!](#) using original JXTA JXTACH

[Stop this Partition!](#) [Delete this Partition!](#)

[latest logfile](#)

[Index: A – Z](#) | [Imprint](#) | [Webmaster](#)

5.2 the client

The **AEOFORGE-Client.zip** file contains all libraries needed to run an extended JXTA Shell that acts as a Command Line Interface towards the AEOFORGE application. The shell is started executing a .bat file in Windows based machines or a .sh file in unix-like machines.

To configure the client, the end-user must:

- set its own environment defining the variable JAVA_HOME or editing the .bat (or.sh) file setting the corresponding script variable
- set in the rdvs.txt file, located at the same path of the previously described files, the testbed Rendezvous Peer TCP address (e.g.: tcp://ipaddress:aport)

6 AEOFORGE APIs

This chapter provides a description for the AEOFORGE commands a user can issue from the AEOFORGE command line interface.

<p>aeoforge.add</p>	<p>The user adds a file <filepath> to the project <projectname>. When the AEOFORGE server receives this command it obtains the file content from the client issuing the command, retrieves the project descriptor stored on the forge, obtains a secure timestamp from the STA service, stores the file on the forge, updates the file descriptor, sends back an ack message to the client, registers a new task with the SCHEDULER service, asks the SCHEDULER to assign a compiling task to one of the registered COMPILER nodes, asks the chosen COMPILER node to compile the project, on compilation success it removes the task from the SCHEDULER service and quits.</p> <p>If the project doesn't exist or the file is already part of the project a message is sent back to the client. If an error occurs on the server a message describing the cause is sent back to the client.</p> <table border="1" data-bbox="534 1435 1348 1626"> <thead> <tr> <th colspan="2">PARAMETERS</th> </tr> </thead> <tbody> <tr> <td><projectname></td> <td>unique name of the project</td> </tr> <tr> <td><filepath></td> <td>relative path to the file (e.g.: "/myDir/MyClass.java")</td> </tr> </tbody> </table>	PARAMETERS		<projectname>	unique name of the project	<filepath>	relative path to the file (e.g.: "/myDir/MyClass.java")
PARAMETERS							
<projectname>	unique name of the project						
<filepath>	relative path to the file (e.g.: "/myDir/MyClass.java")						
<p>aeoforge.ci</p>	<p>The user commits changes to a file <filepath> modified on the client machine. The file is part of the project <projectname>. When the AEOFORGE server receives this command it obtains from the client the file content and its certificate containing the secure timestamp previously issued by the STA (e.g.: upload/add), retrieves the project descriptor file, verifies the certificate via the STA service, obtains a new secure timestamp to update the file certificate, stores the file on the forge, updates the project descriptor and stores it on the forge,</p>						

	<p>sends back an ack message to the client, registers a new task with the SCHEDULER service, asks the SCHEDULER to assign a compiling task to one of the registered COMPILER nodes, asks the chosen COMPILER node to compile the project, on compilation success it removes the task from the SCHEDULER service and quits.</p> <p>If the project doesn't exist or the file is not part of the project or an old version of the file has been modified (e.g.: concurrent modification) a message is sent back to the client. If an error occurs on the server a message describing the cause is sent back to the client.</p> <table border="1" data-bbox="536 640 1347 831"> <tr> <th colspan="2">PARAMETERS</th> </tr> <tr> <td><i><projectname></i></td> <td>unique name of the project</td> </tr> <tr> <td><i><filepath></i></td> <td>relative path to the file (e.g.: <code>"/myDir/MyClass.java"</code>)</td> </tr> </table>	PARAMETERS		<i><projectname></i>	unique name of the project	<i><filepath></i>	relative path to the file (e.g.: <code>"/myDir/MyClass.java"</code>)
PARAMETERS							
<i><projectname></i>	unique name of the project						
<i><filepath></i>	relative path to the file (e.g.: <code>"/myDir/MyClass.java"</code>)						
aeoforge.co	<p>The user checks-out, for modifications, a file <i><filepath></i>. The file is part of the project <i><projectname></i>.</p> <p>When the AEOFORGE server receives this command it obtains from the forge the file content and its certificate containing the secure timestamp previously issued by the STA (e.g.: upload/add) and sends both to the client issuing the command.</p> <p>If the project doesn't exist or the file is not part of the project a message is sent back to the client. If an error occurs on the server a message describing the cause is sent back to the client.</p> <table border="1" data-bbox="536 1323 1347 1514"> <tr> <th colspan="2">PARAMETERS</th> </tr> <tr> <td><i><projectname></i></td> <td>unique name of the project</td> </tr> <tr> <td><i><filepath></i></td> <td>relative path to the file (e.g.: <code>"/myDir/MyClass.java"</code>)</td> </tr> </table>	PARAMETERS		<i><projectname></i>	unique name of the project	<i><filepath></i>	relative path to the file (e.g.: <code>"/myDir/MyClass.java"</code>)
PARAMETERS							
<i><projectname></i>	unique name of the project						
<i><filepath></i>	relative path to the file (e.g.: <code>"/myDir/MyClass.java"</code>)						
aeoforge.compile	<p>The user downloads a .jar file containing the compiled java project <i><projectname></i>. The file will be downloaded on the [destDir] folder if provided, else it will be stored in the current client folder.</p> <p>When the AEOFORGE server receives this command, it retrieves the project descriptor from the forge, retrieves the jar file and sends it to the client where the file is stored.</p> <p>If the jar file is not available due to errors in the compilation phase a message is sent back to the client describing the error. If the project has not been compiled after the last modification, but an old version of the jar file exists both the jar and a message describing the error are sent back to the client. If the</p>						

	<p>project doesn't exist a message is sent back to the client. If an error occurs on the server a message describing the cause is sent back to the client.</p> <table border="1" data-bbox="536 347 1342 501"> <tr> <th colspan="2" data-bbox="536 347 1342 398">PARAMETERS</th> </tr> <tr> <td data-bbox="536 398 759 450"><projectname></td> <td data-bbox="759 398 1342 450">unique name of the project</td> </tr> <tr> <td data-bbox="536 450 759 501">[destDir]</td> <td data-bbox="759 450 1342 501">destination folder, optional</td> </tr> </table>	PARAMETERS		<projectname>	unique name of the project	[destDir]	destination folder, optional
PARAMETERS							
<projectname>	unique name of the project						
[destDir]	destination folder, optional						
aeoforge.del	<p>The user deletes the file <filepath> from the project <projectname>. When the AEOFORGE server receives this command it obtains from the forge the project descriptor, deletes the file from the forge, updates the file descriptor, stores it back on the forge, sends back an ack message to the client, registers a new task with the SCHEDULER service, asks the SCHEDULER to assign a compiling task to one of the registered COMPILER nodes, asks the chosen COMPILER node to compile the project, on compilation success it removes the task from the SCHEDULER service and quits</p> <p>If the project doesn't exist or the file is not part of the project a message is sent back to the client. If an error occurs on the server a message describing the cause is sent back to the client.</p> <table border="1" data-bbox="536 1140 1342 1330"> <tr> <th colspan="2" data-bbox="536 1140 1342 1191">PARAMETERS</th> </tr> <tr> <td data-bbox="536 1191 759 1243"><projectname></td> <td data-bbox="759 1191 1342 1243">unique name of the project</td> </tr> <tr> <td data-bbox="536 1243 759 1330"><filepath></td> <td data-bbox="759 1243 1342 1330">relative path to the file (e.g.: "./myDir/MyClass.java")</td> </tr> </table>	PARAMETERS		<projectname>	unique name of the project	<filepath>	relative path to the file (e.g.: "./myDir/MyClass.java")
PARAMETERS							
<projectname>	unique name of the project						
<filepath>	relative path to the file (e.g.: "./myDir/MyClass.java")						
aeoforge.describe	<p>The user asks the project <projectname> description. When the AEOFORGE server receives this command, it retrieves the project descriptor from the forge and sends back to the client the list of the files together with their current timestamps. If the project doesn't exist a message is sent back to the client. If an error occurs on the server a message describing the cause is sent back to the client.</p> <table border="1" data-bbox="536 1677 1342 1778"> <tr> <th colspan="2" data-bbox="536 1677 1342 1729">PARAMETERS</th> </tr> <tr> <td data-bbox="536 1729 759 1778"><projectname></td> <td data-bbox="759 1729 1342 1778">unique name of the project</td> </tr> </table>	PARAMETERS		<projectname>	unique name of the project		
PARAMETERS							
<projectname>	unique name of the project						
aeoforge.download	<p>The user downloads the whole project <projectname>, as a .zip file in the [destDir] folder, if provided, or in the current client folder.</p> <p>When the AEOFORGE server receives this command, it retrieves the project descriptor from the forge, reads the</p>						

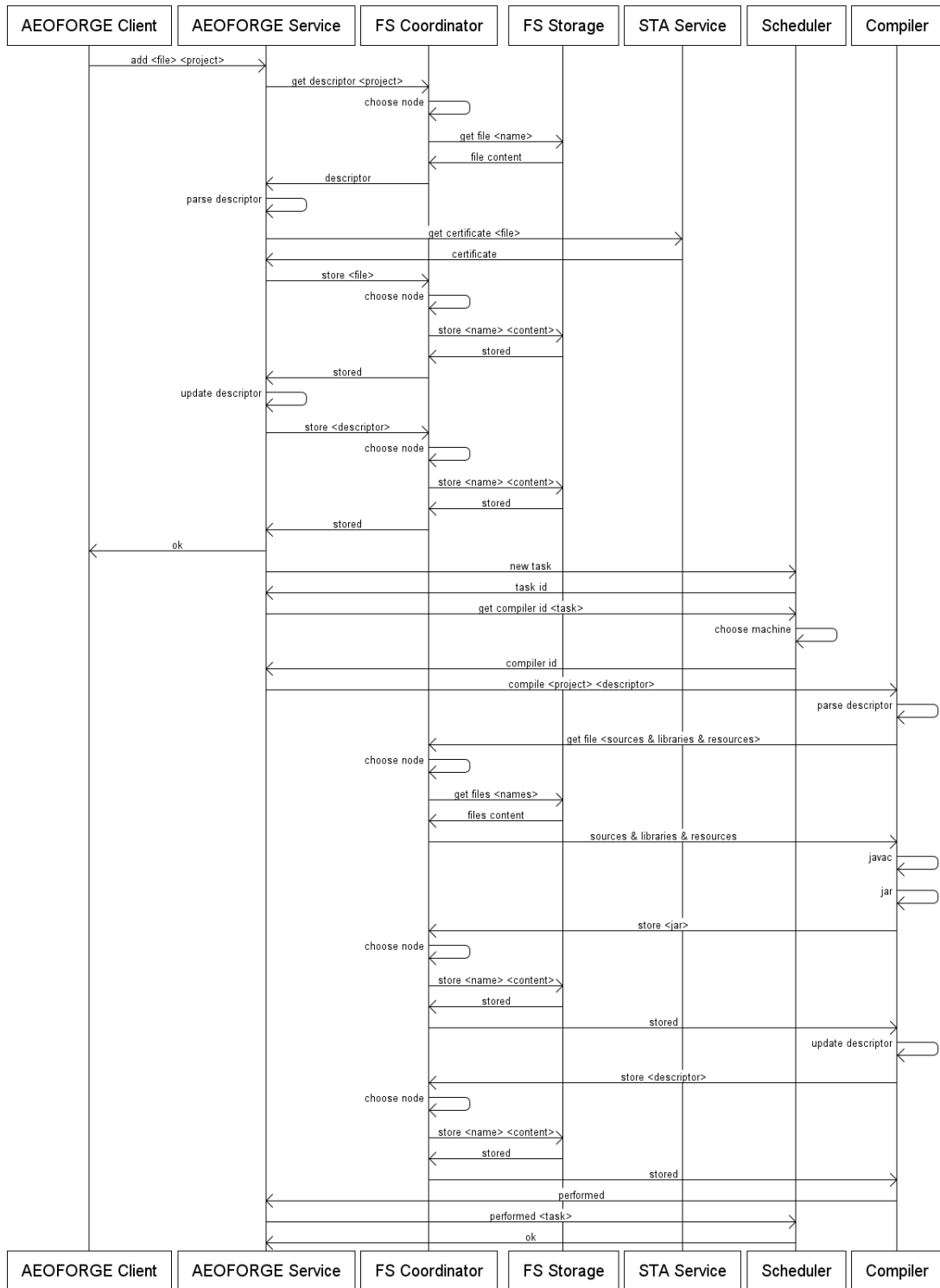
	<p>descriptor retrieving all the project files from the forge, creates the .zip file maintaining the original project structure and sends the archive content to the client. If the project doesn't exist a message is sent back to the client. If an error occurs on the server a message describing the cause is sent back to the client.</p> <table border="1"> <tr> <th colspan="2">PARAMETERS</th> </tr> <tr> <td><projectname></td> <td>unique name of the project</td> </tr> <tr> <td>[destDir]</td> <td>destination folder, optional</td> </tr> </table>	PARAMETERS		<projectname>	unique name of the project	[destDir]	destination folder, optional
PARAMETERS							
<projectname>	unique name of the project						
[destDir]	destination folder, optional						
aeoforge.execute	<p>The user requests the execution of the Java project <projectname> providing the full name of the Class <mainclass> containing the main() method. The AEOFORGE server retrieves the project descriptor from the forge, registers a new task to the SCHEDULER, asks the SCHEDULER to assign the task to an Executor node, sends the chosen Executor the project descriptor and all the parameters specified by the client. At the end of the execution it removes the task from the scheduler and quits. If the project doesn't exist a message is sent back to the client.</p> <table border="1"> <tr> <th colspan="2">PARAMETERS</th> </tr> <tr> <td><projectname></td> <td>unique name of the project</td> </tr> <tr> <td><mainclass></td> <td>full name of the class containing the main() method (e.g.:"mypackage.mysubpackage.MyClass")</td> </tr> </table>	PARAMETERS		<projectname>	unique name of the project	<mainclass>	full name of the class containing the main() method (e.g.:"mypackage.mysubpackage.MyClass")
PARAMETERS							
<projectname>	unique name of the project						
<mainclass>	full name of the class containing the main() method (e.g.:"mypackage.mysubpackage.MyClass")						
aeoforge.find	<p>The user requests the list of all projects loaded on AEOFORGE. When the AEOFORGE server receives this command, searches for all the projects stored on the forge via SSOD service and sends back the client the list of all the project names. If an error occurs on the server a message describing the cause is sent back to the client.</p>						
aeoforge.noop	<p>The user checks whether the AEOFORGE service is running. When the AEOFORGE service receives this message it sends back an ack message.</p>						
aeoforge.quit	<p>The user requests the AEOFORGE service shutdown. When the AEOFORGE service receives this message it sends back an ack message and shuts down.</p>						
aeoforge.upload	<p>The user uploads a new project, named <projectname> on AEOFORGE. When the AEOFORGE service receives this command, it</p>						

	<p>obtains the .zip file content, certifies each file via the STA service, stores each file on the forge, builds a project descriptor file, stores it on the forge, registers the project with the SSOD service, sends back an ack message to the client, registers a new task with the SCHEDULER, asks the SCHEDULER to assign a compilation task to a Compiler node, asks the compilation to the chosen Compiler node, if compilation succeeds removes the task from the SCHEDULER and quits.</p> <p>If an error occurs on the server, all file stored are removed from the forge and an error message is sent back to the client.</p>						
	<table border="1"> <tr> <th colspan="2" data-bbox="526 638 1355 689">PARAMETERS</th> </tr> <tr> <td data-bbox="526 689 842 813"><tarballfilename.zip></td> <td data-bbox="842 689 1355 813">name of the zip file containing the project files to be uploaded (.zip extension is mandatory)</td> </tr> <tr> <td data-bbox="526 813 842 864"><projectname></td> <td data-bbox="842 813 1355 864">unique name of the project</td> </tr> </table>	PARAMETERS		<tarballfilename.zip>	name of the zip file containing the project files to be uploaded (.zip extension is mandatory)	<projectname>	unique name of the project
PARAMETERS							
<tarballfilename.zip>	name of the zip file containing the project files to be uploaded (.zip extension is mandatory)						
<projectname>	unique name of the project						

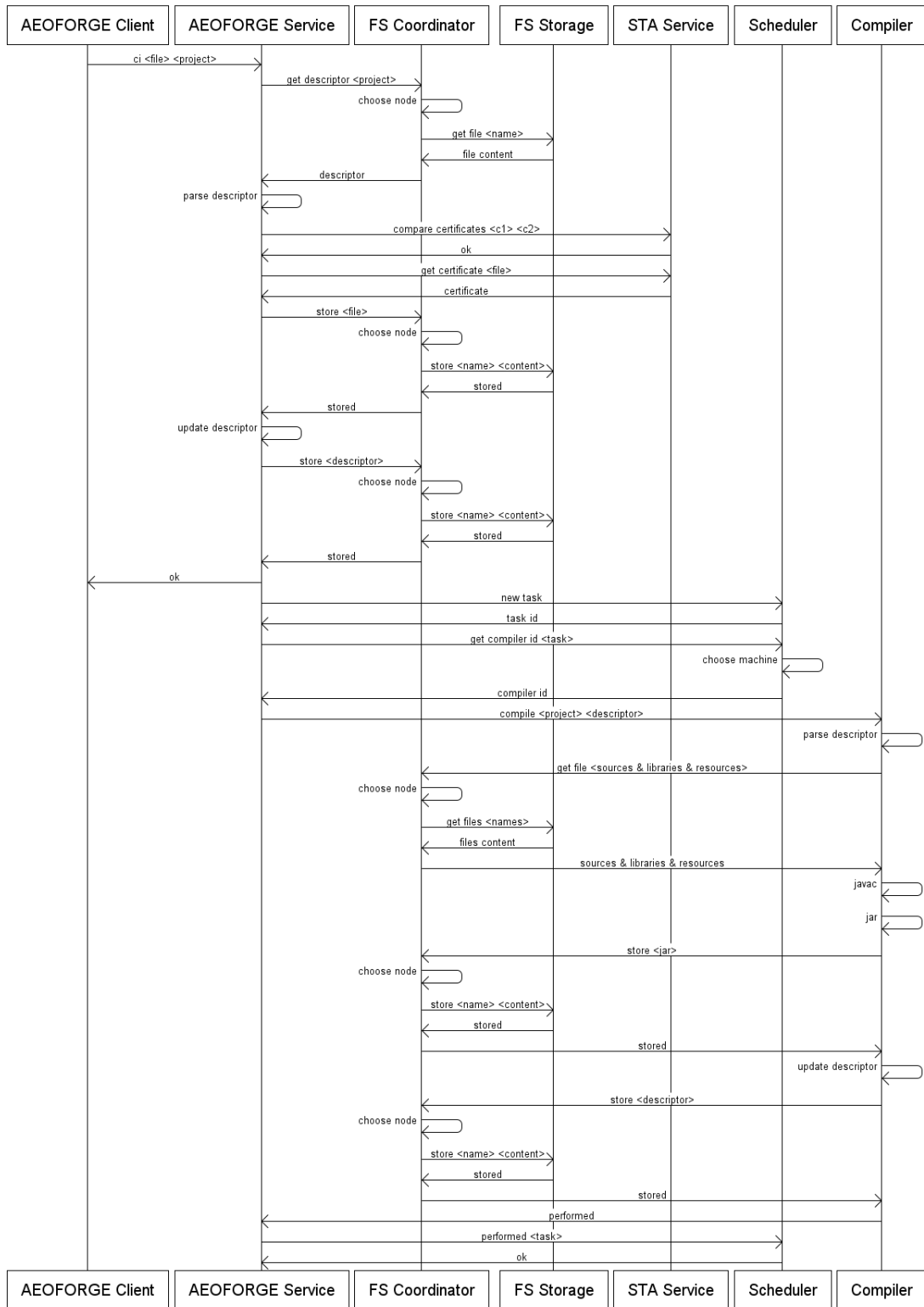
6.1 Sequence Diagrams

In this section we provide UML sequence diagrams for the API described in the section above. In this diagrams the interactions among all the components involved are clearly depicted.

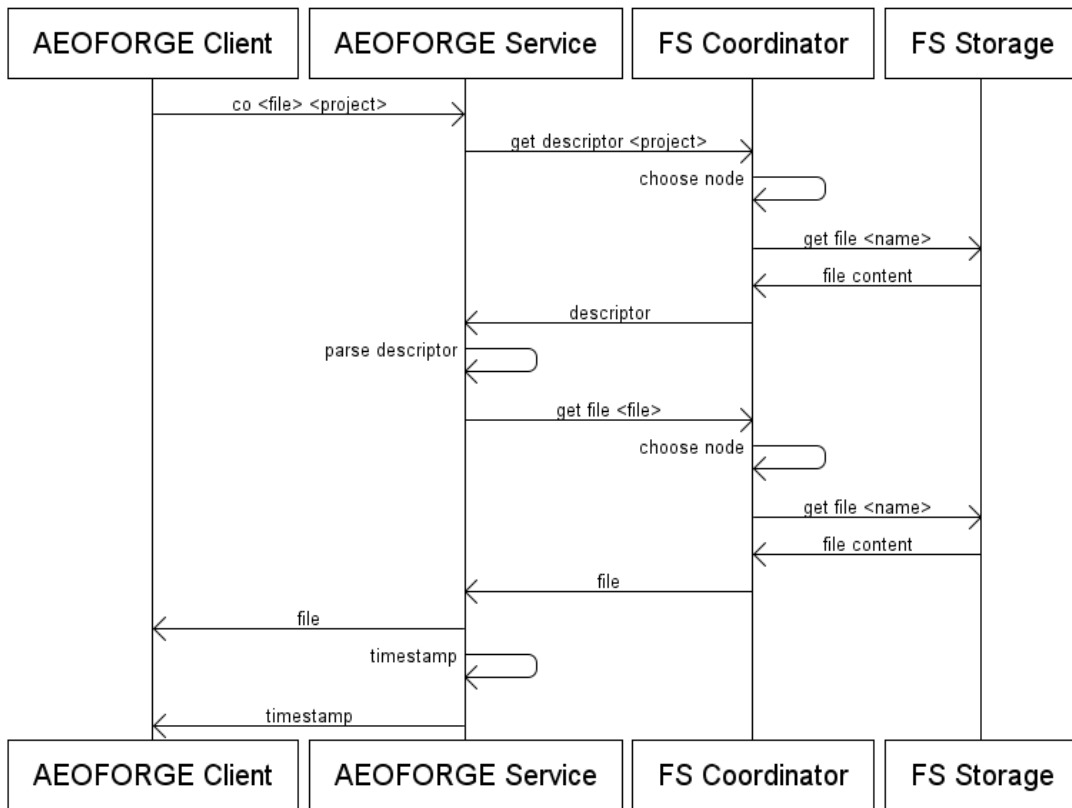
6.1.1 add



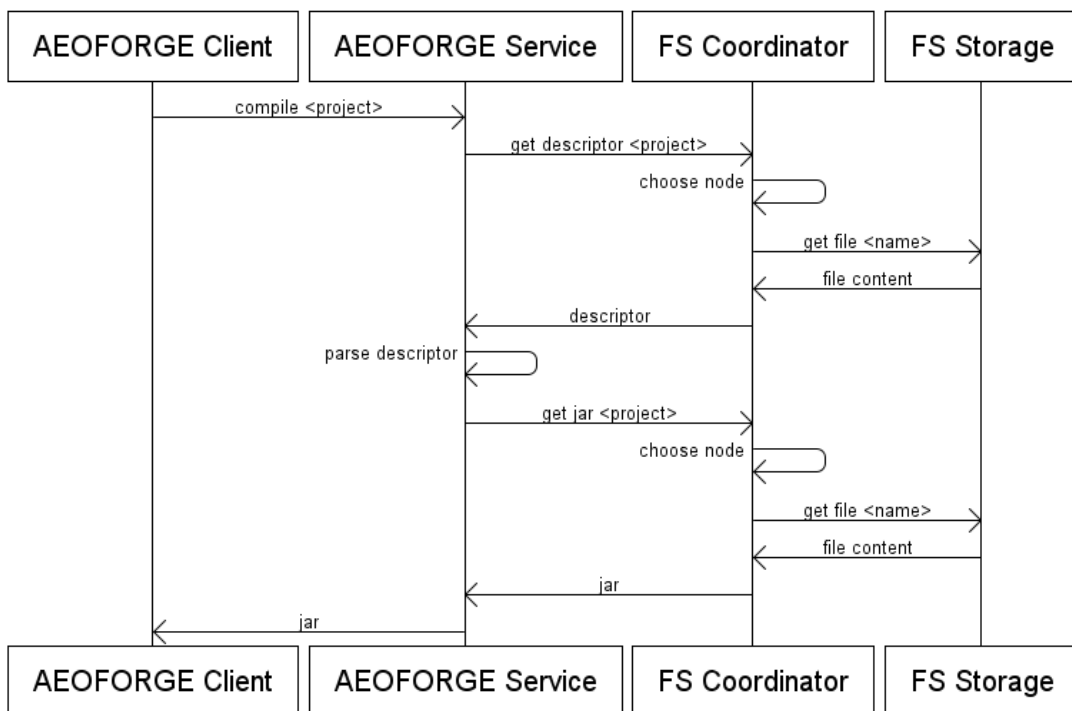
6.1.2 ci



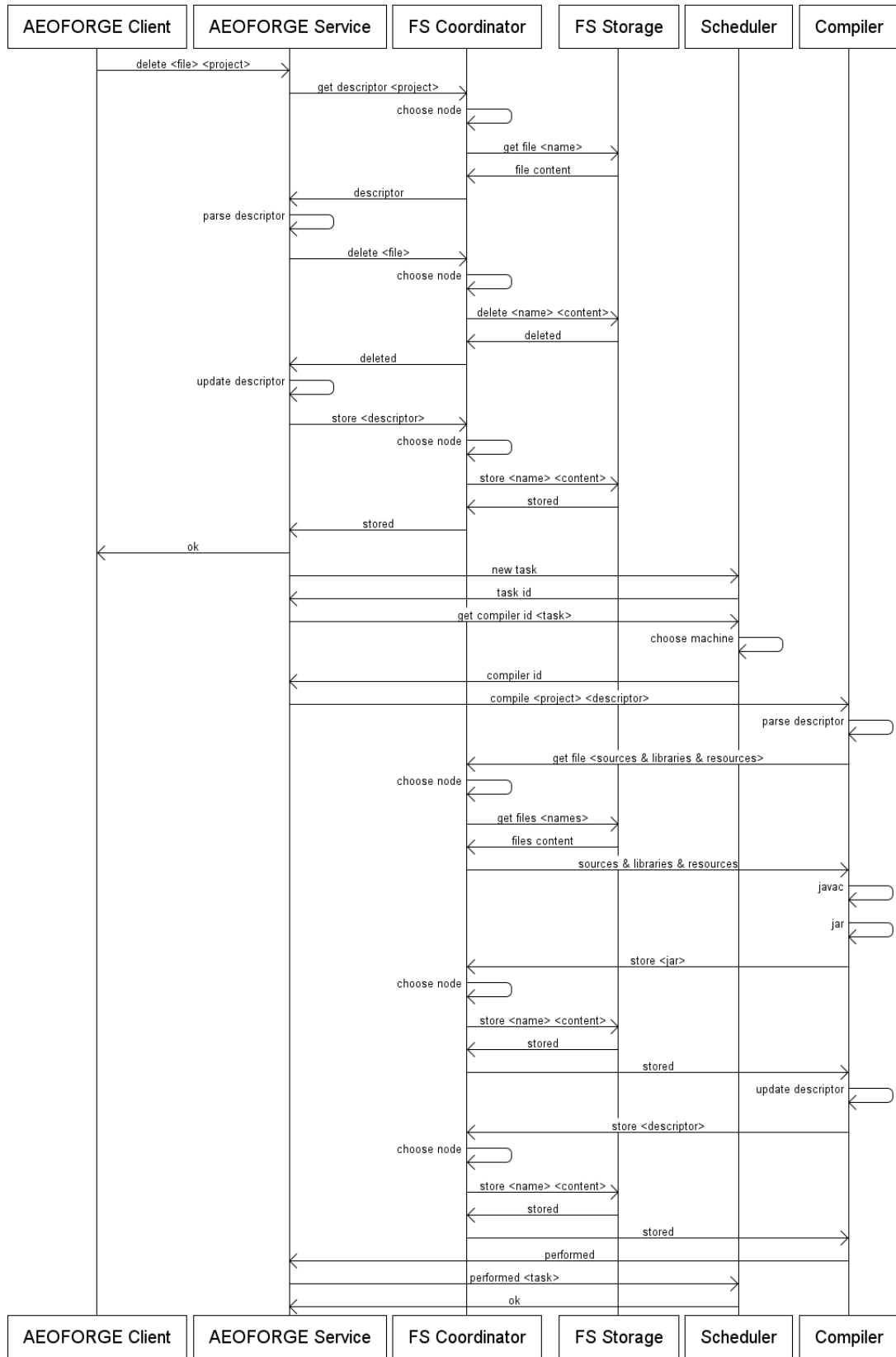
6.1.3 co



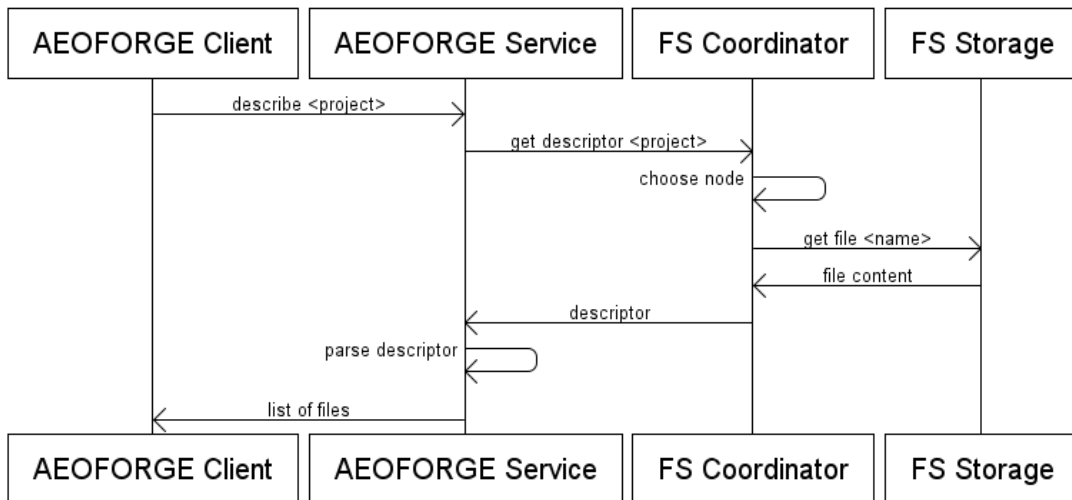
6.1.4 compile



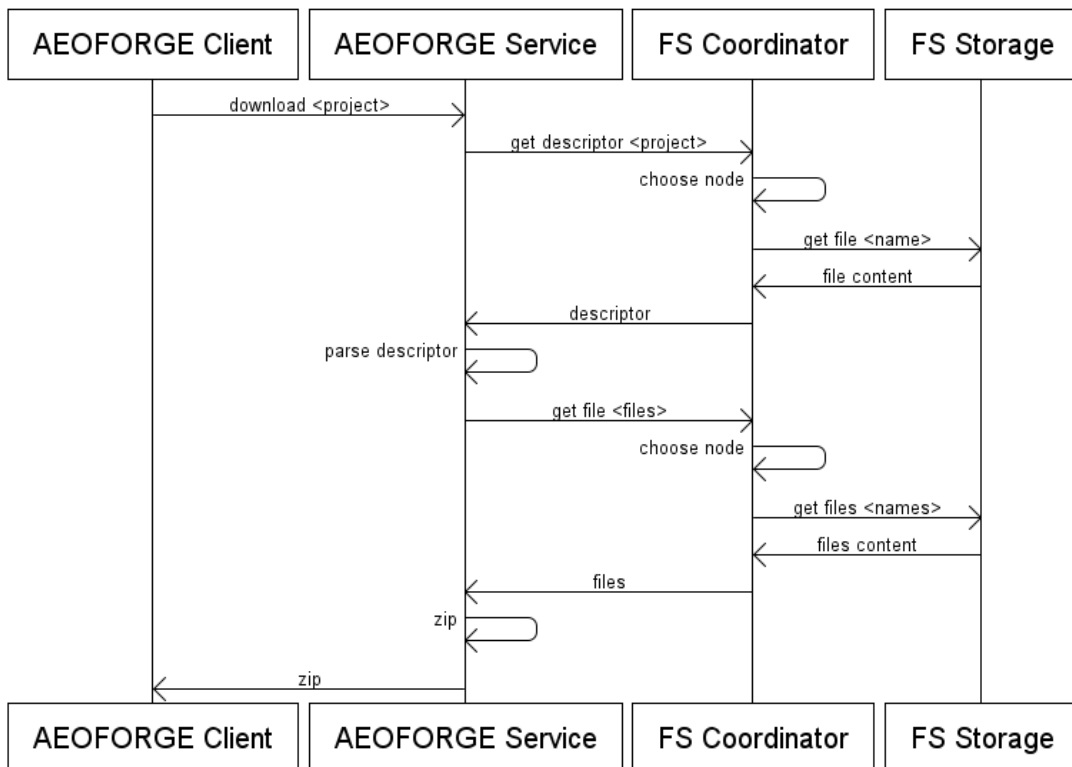
6.1.5 del



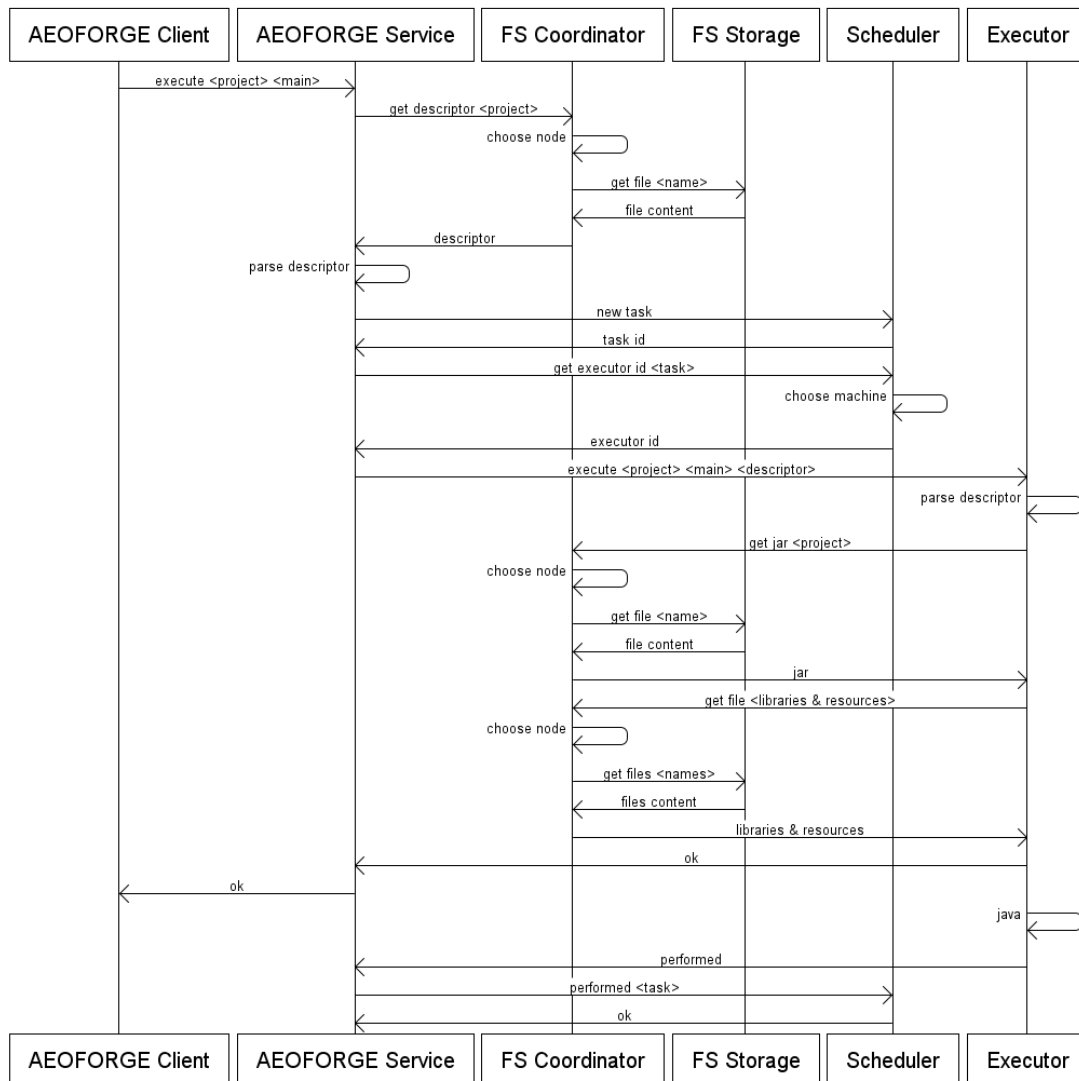
6.1.6 describe



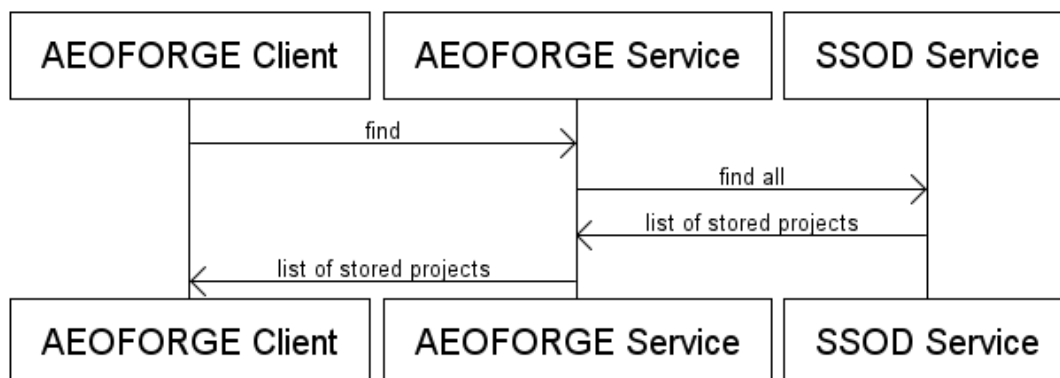
6.1.7 download



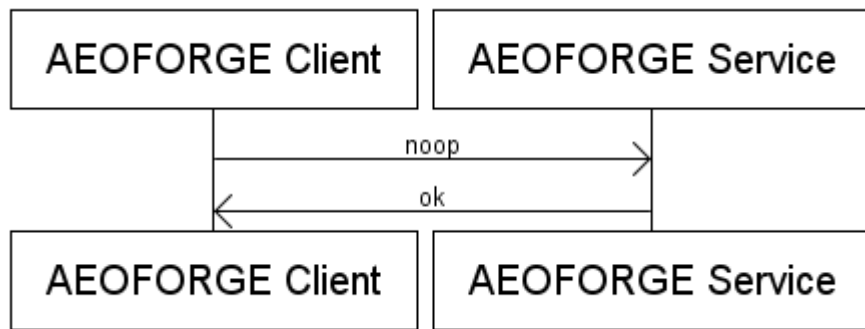
6.1.8 execute



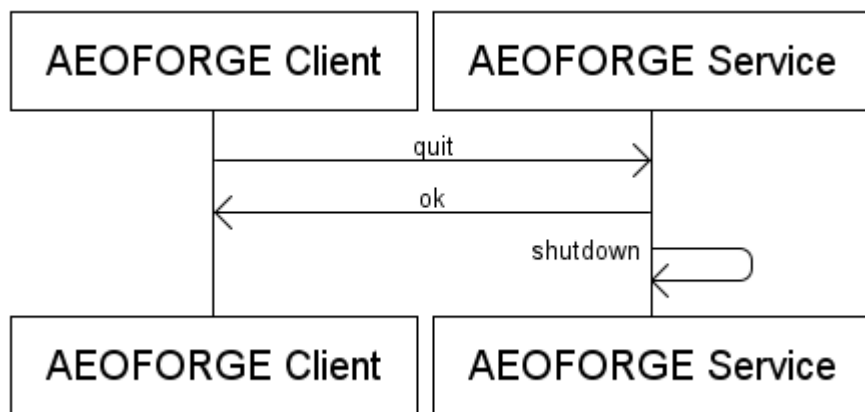
6.1.9 find



6.1.10 *noop*



6.1.11 *quit*



6.1.12 Upload

