Getting Started with Serpens extension for Kepler 2.2.x

Working with UNICORE module

{marcinp, michalo, tzok}@man.poznan.pl

UNICORE module description covers:

Service discovery on UNICORE machines
Data management using UNICORE
Job submission and management on UNICORE

This paper covers Serpens 2.2.1 release

July 20, 2011

Getting Started with Serpens – UNICORE module

Getting Started with Serpens – UNICORE module guide is for scientists who would like to use Serpens suite in Kepler. It was created to assist Kepler users to manage UNICORE resources from the level of Kepler 2.2.x workflow.

Table of Contents

1 Introduction

This guide introduces the main components and functionality of Serpens UNICORE module. It is an add-on module for Kepler, a software application for the analysis and modeling of scientific data and it provides functionality for data and job management on UNICORE environment from the level of Kepler workflow.

1.1 What is UNICORE

From official website http://unicore.eu:

UNICORE (Uniform Interface to Computing Resources) offers a ready-to-run Grid system including client and server software. UNICORE makes distributed computing and data resources available in a seamless and secure way in intranets and the internet.

UNICORE has special characteristics that make it unique among Grid middleware systems. The UNICORE design is based on several guiding principles, that serve as key objectives for further enhancements.

Open source *under BSD license.*

Standards-based*, conforming to the latest standards from the Open Grid Forum (OGF), W3C, OASIS, and IETF, in particular the Open Grid Services Architecture (OGSA) and the Web Services Resource Framework (WS-RF 1.2).*

Open and extensible *realized with a modern Service-Oriented Architecture (SOA), which allows to easily replace particular components with others.*

Interoperable *with other Grid technologies to enable a coupling of Grid infrastructures or the users needs*

Seamless, secure, and intuitive *following a vertical, end-to-end approach and offering components at all levels of a modern Grid architecture from intuitive user interfaces down to the resource level. Like previous versions UNICORE 6 seamlessly integrates in existing environments.*

Mature security mechanisms *adequate for the use in supercomputing environments and Grid infrastructures. X.509 certificates form the basis for authentication and authorisation, enhanced with a support for proxy certificates and virtual organisations (VO) based access control.*

Application integration *mechanisms on the client, services and resource level for a tight integration of various types of applications from the scientific and industrial domain.*

Various operating and batch systems *are supported on all layers, i.e. clients, services and systems; Windows, MacOS, Linux, and Unix systems as well as different batch systems are supported such as LoadLeveler, Torque, SLURM, LSF, OpenCCS, etc.*

Implemented in Java *to achieve platform independence.*

1.2 User Certificate

In UNICORE environment, each user is authenticated and authorized with his certificate and private key. These are provided by local Certificate Authorities (CA). Please contact grid CA available in your country in order to obtain your pair of certificate and private key.

Each certificate is mapped to specific user account. When you have your credentials ready, please contact UNICORE administrators of machine of your interest. You will have your account created and certificate mapped to it.

Important!

Additional step required to work with Serpens UNICORE module is to add your certificate to the list of users allowed to connect with proxy. You need to contact UNICORE administrators and explicitly ask for this service. Kepler actors described in this user manual all use short-term proxy and this step is crucial for further working with UNICORE basing on Serpens suite.

1.3 What is Serpens UNICORE module?

Serpens UNICORE module provides a way for accessing HPC resources through the Kepler workflow. It allows users to:

- query UNICORE registry and obtain information about available resources,
- upload/download data to/from UNICORE storage,
- submit jobs into UNICORE enhanced HPC sites,
- check statuses of the running jobs,
- retrieve *Standard Output* and *Standard Error* of UNICORE job,
- retrieve UNICORE job output generated by application started at HPC machine.

Important, Serpens UNICORE module was tested and validated with UNICORE versions up to 6.3.1.

1.4 Prerequisites

In order to communicate with UNICORE registry you must have a grid proxy. You can generate it using available *gridproxygenerator.kar* workflow which requires that you provide an X509 public certificate and a private key.

Every actor has two parameters to specify path to TrustStore and its password. By default these are empty fields, which is interpreted as not using TrustStore. However optionally if you wish to authenticate UNICORE registry before using its services, you can prepare a TrustStore containing trusted certificates. To do this and to obtain UNICORE server CA you can either contact its administrator or get it using OpenSSL package: (NOTE: the command below must be pasted in one line)

```
openssl s_client -connect HOST:PORT | awk '/-----BEGIN CERTIFICATE-----/,/-----END CERTIFICATE-----/
{ print }' > ca.crt
```

Where HOST:PORT represents the UNICORE registry address. The certificate will be stored in a file “ca.crt”.

After you obtain UNICORE server CA certificate you can import it into truststore with the following command:

```
keytool -importcert -file ca.crt -keystore truststore.jks
```

2 Using Serpens UNICORE module

Once Serpens UNICORE module is installed in Kepler, you will notice new actors available at the Component list.

Actor name	Purpose
UnicoreListSites	Lists sites available at specified UNICORE registry.
UnicoreListStorages	Lists storage services available for given site.
UnicoreUploadFile	Uploads file into UNICORE storage.
UnicoreDownloadFile	Downloads file from the UNICORE storage.
UnicoreSubmitJobJsdl	Submits a job defined in JSDL (Job Submission Description Language).
UnicoreSubmitJob	Submits a job defined using actor's input ports.
UnicoreGetStatus	Returns status of a job.
UnicoreGetOutput	Retrieves output files of a job.

2.1 Listing available sites

UNICORE registry gathers information about one or more sites. Each site is a specific machine available to the user. Knowledge of site names and their availability is crucial in every step of working with UNICORE, from data to job management.

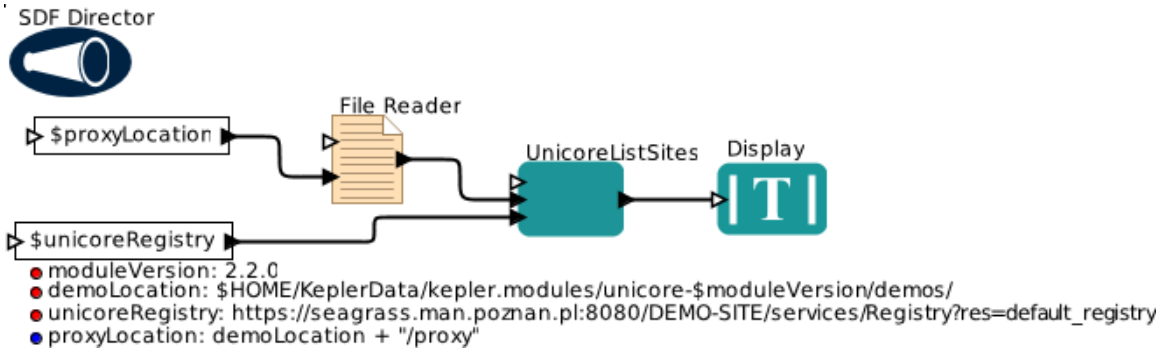
Serpens suite provides an actor UnicoreListSites, which connects to UNICORE registry and retrieves information about available sites. In order to use it, one needs to provide the following data:

Input port/parameter	Description
`proxy`	Mandatory port to receive contents of your proxy file.
`registry`	Mandatory port to receive UNICORE registry identifying a machine to use.
`trustStorePath`	Optional parameter specifying location of your TrustStore.
`trustStorePassword`	Optional parameter specifying TrustStore password.

Actor has one output port:

Output port	Description
`sites`	Port to send an array of available UNICORE sites.

An example workflow using this actor looks like the following:



2.2 Listing available storages

Each UNICORE site may provides a set of storage locations. There user can store its files. Most of the tasks one can do using UNICORE machines require some input and generate some output files. Thus the need to list available storages.

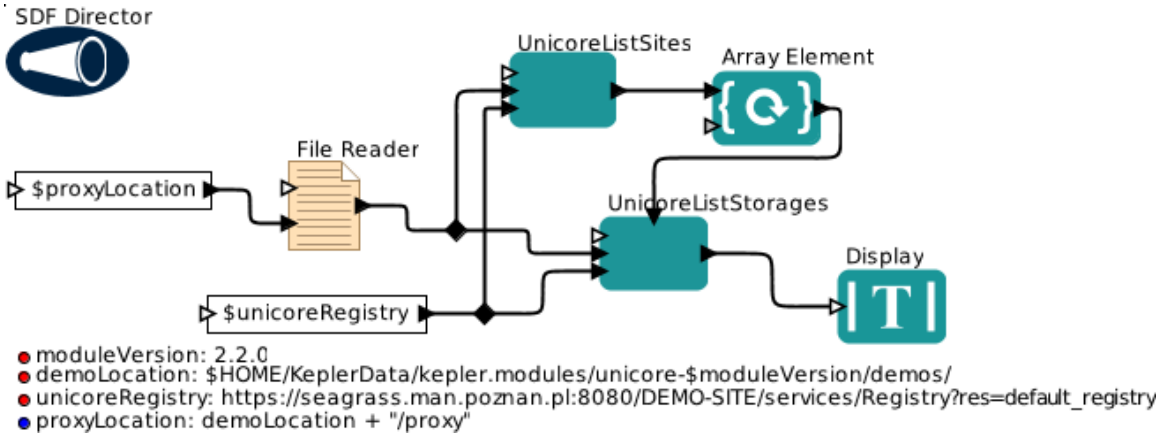
An actor responsible for this task is UnicoreListStorages. To use it, you must provide the actor with the following data:

Input port/parameter	Description
`site`	Mandatory port to receive UNICORE site name with the storage service.
`proxy`	Mandatory port to receive contents of your proxy file.
`registry`	Mandatory port to receive UNICORE registry identifying a machine to use.
`trustStorePath`	Optional parameter specifying location of your TrustStore.
`trustStorePassword`	Optional parameter specifying TrustStore password.

Actor has one output port:

Output port	Description
`storages`	Port to send an array of available UNICORE storages.

An example workflow, which lists storages for the first site on the list, is the following:



2.3 Uploading files

In order to use external data during job execution, one must upload data into UNICORE storage in the first place. This can be done using UnicoreUploadFile actor. It allows to upload local input files into UNICORE storage specified by the user.
To use UnicoreUploadFile one must provide the following data:

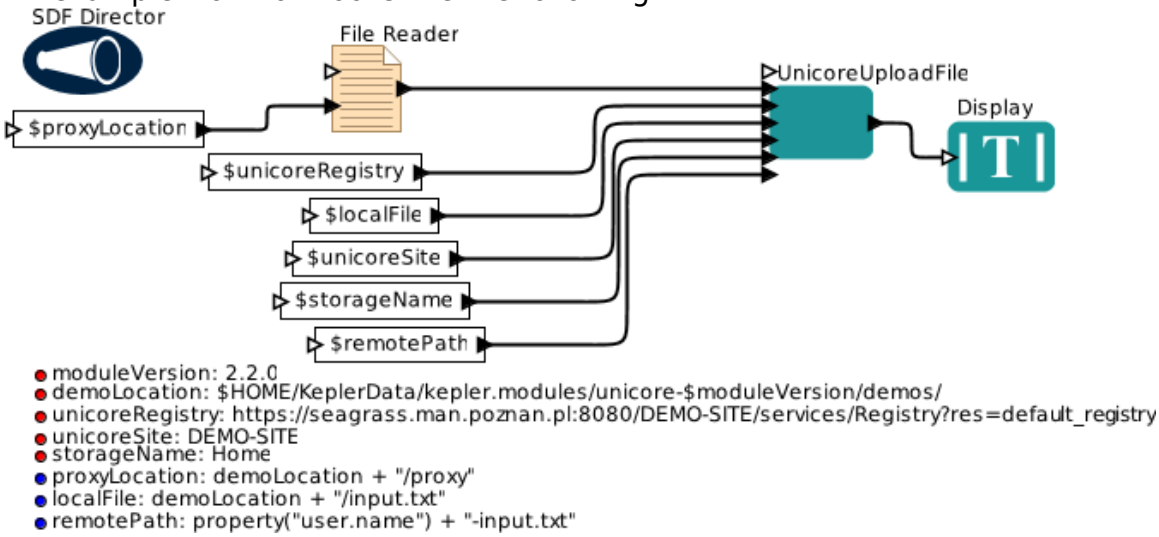
Input port/parameter	Description
`source`	Mandatory port to receive path of local file which will be uploaded.
`site`	Mandatory port to receive UNICORE site name with the storage service.

Input port/parameter	Description
`storage`	Mandatory port to receive storage name.
`target`	Mandatory port to receive remote path on UNICORE storage (NOTE: this path must contain destination file name as well; it is insufficient to set it to directory name only).
`proxy`	Mandatory port to receive proxy needed to authenticate you on UNICORE machine.
`registry`	Mandatory port to receive a URL of UNICORE's registry.
`trustStorePath`	Optional parameter specifying location of your TrustStore.
`trustStorePassword`	Optional parameter specifying TrustStore password.

Actor has two output ports:

Output port	Description
`outputURI`	Port to receive URI of uploaded file.

An example workflow looks like the following:



2.4 Downloading files

Jobs produce new files during execution. So as well as to upload input files, it is crucial to download output ones.

To achieve this, you can use UnicoreDownloadFile. There are two possible ways to reference remote file:

1. Provide file URI (Unique Resource Identifier) which already consists of host, port and path.

2. Point to file on a specific UNICORE site, storage and path. You must thus provide these three arguments.

UnicoreDownloadFile allows you to use both of these methods. First, it checks if URI is given and if it is, then site, storage and source path are not read at all. Otherwise these three values must be present as tokens on input ports.

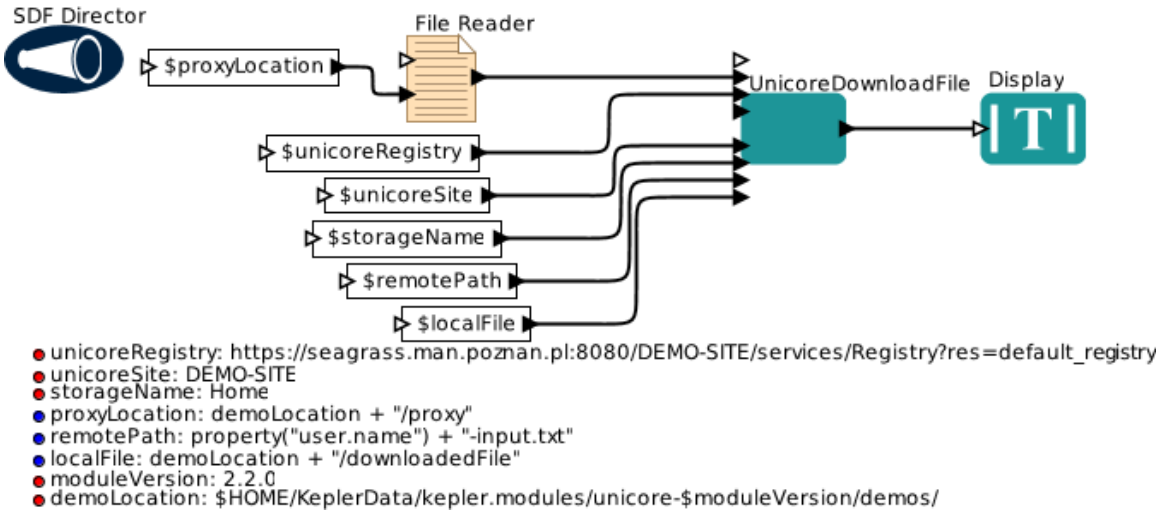
To use this actor please provide it with the following data:

Input port/parameter	Description
`uri`	Mandatory port to receive URI of UNICORE resource to download. NOTE: when you specify an URI, then site, storage and source are not read!
`site`	Mandatory port to receive UNICORE site name with the storage service. NOTE: when you specify an URI, then site, storage and source are not read!
`storage`	Mandatory port to receive storage name. NOTE: when you specify an URI, then site, storage and source are not read!
`source`	Mandatory port to receive remote path on UNICORE storage. NOTE: when you specify an URI, then site, storage and source are not read!
`target`	Mandatory port to receive a local path.
`proxy`	Mandatory port to receive proxy needed to authenticate you on UNICORE machine.
`registry`	Mandatory port to receive a URL of UNICORE's registry.
`trustStorePath`	Optional parameter specifying location of your TrustStore.
`trustStorePassword`	Optional parameter specifying TrustStore password.

Actor has one output port:

Output port	Description
isSuccessful	Port to send status of file download process.

An example workflow using UnicoreDownloadFile looks like this:



2.5 Job submission

Serpens suite for Kepler 2.x allows you to use two ways of job submission:

1. Job is defined in JSDL (Job Submission Description Language) which is a standard XML-based format. It allows to define what resources are needed and what job will be run.
2. Job is defined using a set of input ports.

The first method is more flexible, because you can use all the fields of JSDL and describe a job in every detail. The second way is suitable for most of usual cases and allows to submit very complex jobs with many requirements and options.

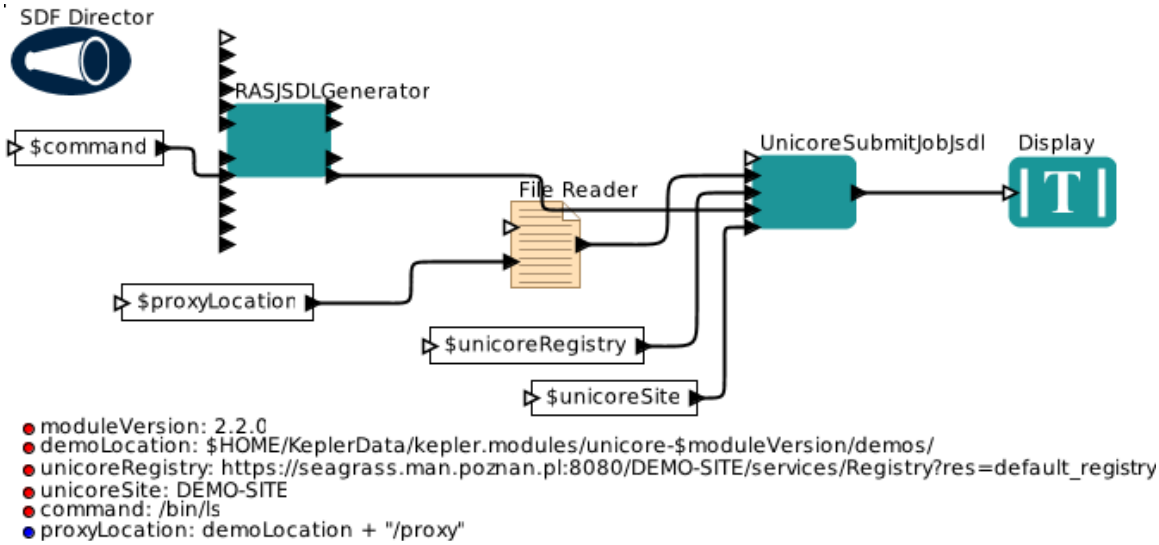
To submit job the first way, you can use UnicoreSubmitJobJsdl with the following input ports/parameters:

Input port/parameter	Description
`jsdl`	Mandatory port to receive JSDL description of a UNICORE job.
`proxy`	Mandatory port to receive contents of your proxy file.
`registry`	Mandatory port to receive UNICORE registry identifying a machine to use.
`siteName`	Optional port to receive name of site to use on chosen machine.
`trustStorePath`	Optional parameter specifying location of your TrustStore.
`trustStorePassword`	Optional parameter specifying TrustStore password.

Actor has one output port:

Output port	Description
status	Port to receive ID of the submitted job.

An example workflow using UnicoreSubmitJobJsdl using JSDL generator looks like the following:



To submit job the second way, you can use UnicoreSubmitJob. It has many ports, however a big part of them is optional. The full list is the following:

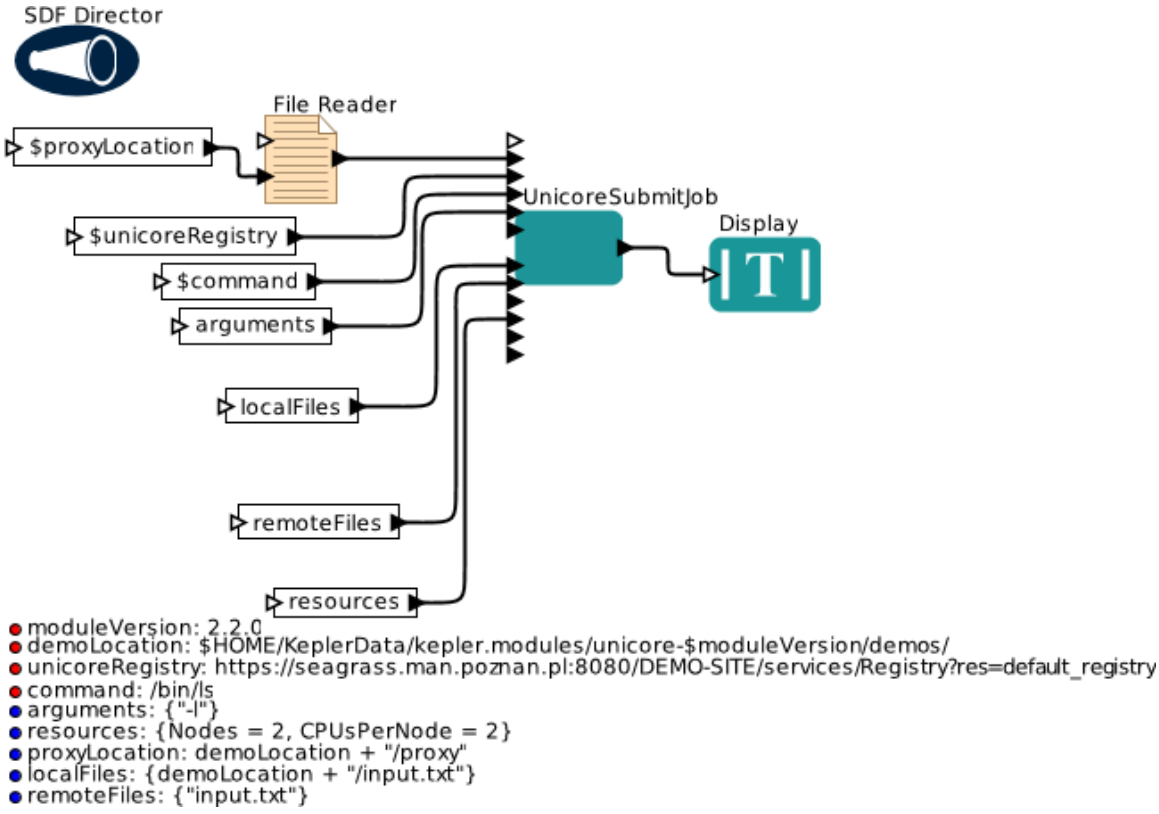
Input port/parameter	Description
proxy	Mandatory port to receive contents of your proxy file.
registry	Mandatory port to receive UNICORE registry identifying a machine to use.
executable	Mandatory port to receive path to executable.
arguments	Optional port to receive an array of arguments to executable.
environment	Optional port to receive an array of environment variables.
imports	Optional port to receive an array of job's input files.
exports	Optional port to receive an array of job's output files.
siteName	Optional port to receive name of site to use on chosen machine.
resources	Optional port to receive a record token with names and values of res.
executionEnvironment	Optional port to receive a record token with execution environment.

Input port/parameter	Description
`preferredProtocols`	Optional port to receive preferred protocols (a string with names separated with space).
`trustStorePath`	Optional parameter specifying location of your TrustStore.
`trustStorePassword`	Optional parameter specifying TrustStore password.

Actor has one output port:

Output port	Description
`epr`	Port to send generated EPR, which is UNICORE's job ID.

From the listing, one can notice that job can be minimally defined by executable only. All other fields are considered optional. They allow to define input/output files, environment variables, and other features. An example workflow using many of these capabilities looks like the following:



2.6 Job management

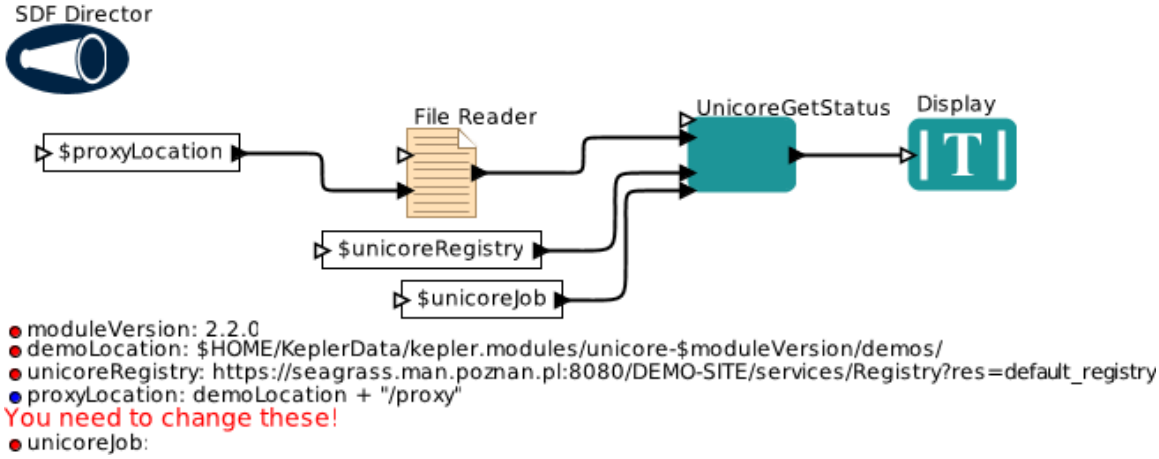
By job management, we understand both status inquiry and output retrieval. The first task can be done with UnicoreGetStatus actor, which has the following input ports:

Input port/parameter	Description
epr	Mandatory port to receive an EPR (id) of a UNICORE job.
proxy	Mandatory port to receive contents of your proxy file.
registry	Mandatory port to receive UNICORE registry identifying a machine to use.
trustStorePath	Optional parameter specifying location of your TrustStore.
trustStorePassword	Optional parameter specifying TrustStore password.

Actor has one output port:

Output port	Description
status	Port to send status of a UNICORE job.

An example workflow checking status of a single job looks like this:



When the job is done, its output files can be downloaded using UnicoreGetOutput. Please note that standard output and standard error are both treated as job output, in consequence, this actor will retrieve them by default. To use it, you must fill its input ports:

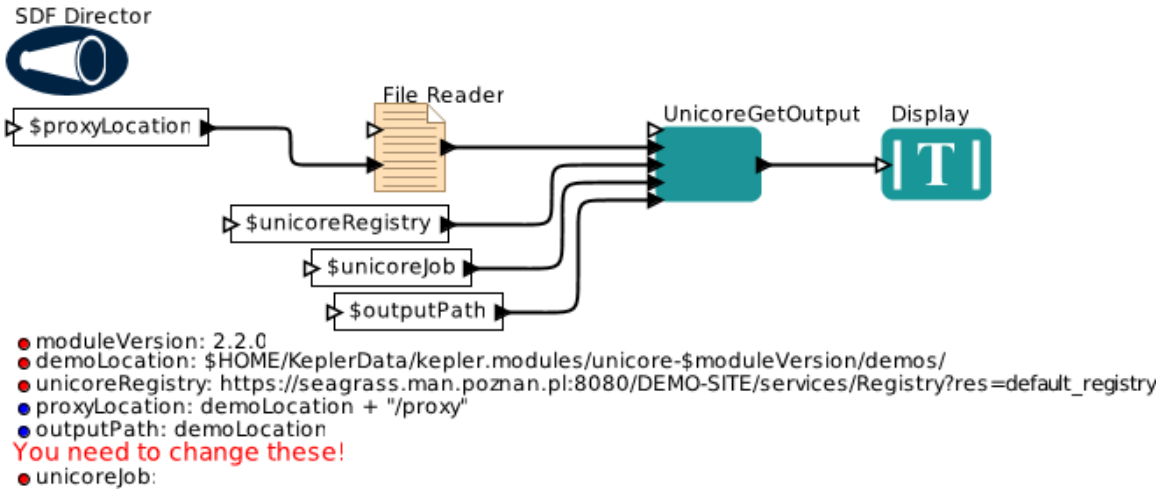
Input port/parameter	Description
epr	Mandatory port to receive an EPR (id) of a UNICORE job.
outputPath	Mandatory port to receive output path where files will be downloaded.
proxy	Mandatory port to receive contents of your proxy file.
registry	Mandatory port to receive UNICORE registry identifying a machine to use.

Input port/parameter	Description
`trustStorePath`	Optional parameter specifying location of your TrustStore.
`trustStorePassword`	Optional parameter specifying TrustStore password.

Actor has one output port:

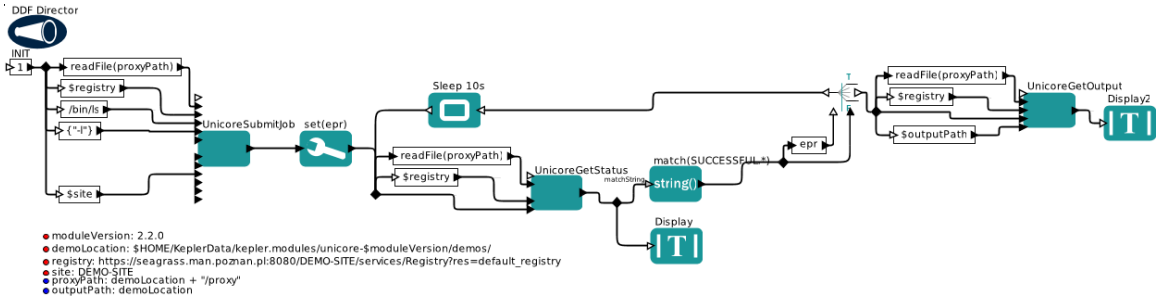
Output port	Description
`status`	Port to send an array of downloaded files.

An example workflow retrieving output of UNICORE job looks like this:



2.7 Full workflow: job submission and management

All previously mentioned actors are sufficient to create a full workflow capable of managing a UNICORE job. Actors from Serpens UNICORE module allow to submit a job, check its status and retrieve the output upon finishing. These building blocks together may lead to a workflow like this example one:



3 Glossary

JSDL – Job Submission Description Language, a standard format of job description.

Serpens – a Kepler 2.x suite containing modules to work with grid and HPC resources from the level of Kepler workflow. Currently available modules provide actors for gLite, UNICORE and Vine Toolkit.

Proxy – short for Proxy Certificate is a short-term (typically 12 hours) Digital Certificate designed to act remotely on behalf of a user.

Trust store – contains trusted CA certificates.

Key store – contains user's private key.

UNICORE registry – A UNICORE registry is an URL published for clients to use UNICORE services. Each UNICORE server should have at least one, publicly available registry as long as it is exposed in public. You should refer to your UNICORE instance manual, or contact your local UNICORE administrator in order to get registry address.

4 References

1. Serpens – http://scilla.man.poznan.pl/serpens
2. UNICORE – http://www.unicore.eu/
3. Kepler-2.0 – https://kepler-project.org/
4. JSDL – http://www.gridforum.org/documents/GFD.56.pdf

5 Acknowledgments

This research has also received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n°211804 (EUFORIA).