Outline

- Motivation
- Approach
- Supported Services
- Landscape at JSC
- Future Work
Motivation

- Large number of parameters in any UNICORE configuration
- Several interacting services
- Simple write-down of infrastructure
- Automated installation and service deployment
- Certificate setup
Approach

Puppet

- Automate as much of the infrastructure as possible
- Achieve > 90% of configuration in little to no time and focus on real issues
Supported Services

- Gateway
- Registry
- UNICORE/X
- XUUDB
- Service Orchestrator
- Workflow Engine
- Portal
- UFTP AuthServer
- [Unity]
General Approach

1. Download .tar.gz from SF.net
2. Ensure presence of required packages, e.g. Java or Grid certificates
3. Unpack in /opt/unicore-source
4. Create version and machine specific directory, e.g. /opt/unicore/unicorex-7.10.2-fzj_jureca
5. “Static” files via symbolic links to unpacked “sources”
6. Configuration files created from templates in Puppet module
7. Certificates created acc. to template
8. Service configuration

```
/opt/unicore/unicorex-7.10.2-fzj_jureca
|-- bin -> /opt/unicore-source/...
|-- certs
| |-- conf
| | |-- unicore-idb
| | | -- deep
| | | | -- simpleidb
| | | -- idb
| | | | -- Bash:4.3.30.idb
| | | -- ...
| | -- wsrflite.xml
| | -- xacml2.config -> /opt/unicore-source/...
| | -- xacml2Policies -> /opt/unicore-source/...
| | -- xnjs_legacy.xml
|-- data
|-- LAST_PID
|-- lib -> /opt/unicore-source/...
|-- logs
| |-- startup.log
| |-- unicorex.log
```
Configuration File Templates

- Templates support lists
- Useful if you have several
  - registries
  - authenticators
  - attribute sources
  - storages

```ruby
# Registry URL:
container.externalregistry.url=<%= @registry_url %>

<% if @registry_urls -%>
<% urlNo = 2 -%>
# additional URL(s):
<% @registry_urls.each do |regUrl| -%>
  container.externalregistry.url=<%= urlNo -%>='<%= regUrl %>
<% urlNo +=1 -%>
<% end -%>
<% end -%>

<% if @storages -%>
<% storageCount = 0 -%>
<% @storages.each do |storageName, storageParams| -%>
  # <%= storageName %> Storage
  coreServices.targetsystem.storage.<%= storageCount -%>.name='<%= storageName %>'
  <% storageParams.each do |param, value| -%>
    coreServices.targetsystem.storage.<%= storageCount -%>.<%= param -%>='<%= value %>
  <% end -%>
  <% storageCount +=1 -%>
<% end -%>
<% end -%>
<% end -%>
```
UNICORE/X

An Example

- Infrastructure written as YAML files
- An (incomplete) example

```
unicore::unicorex::instances:
  jureca:
    version: '7.11.0'
    listen_address: 'unic-integration.fz-juelich.de'
    port: 7777
    java_heap_size: '512'
    sitename: 'JURECA'
    registry_url: 'https://unic-integration.fz-juelich.de'
    attribute_sources:
      XUUDB:
        class: 'eu.unicore.uas.security.xuudb.XUUDBAuthoriser'
        xuudbHost: 'https://unic-integration.fz-juelich.de'
    use_mysql_persistence: true
    persistence_database: 'data_demo_site'
    persistence_password: '**************************'
    use_default_simpleidb: true
    trusted_assertion_issuers: ['/opt/unicore/certs/trusted-unity-cert/*.pem']
```
Landscape at JSC

- Central persistence DB
- Single U/X per HPC machine
- Multiple U/X per UNICORE host
- Dedicated Gateway and Registry host
- Single Gateway for multiple infrastructures
Future Work

- Publish Puppet module
  - requires removing dependency on internal module
- Create decent documentation to get administrators started