

adaptTo()

APACHE SLING & FRIENDS TECH MEETUP
BERLIN, 23-25 SEPTEMBER 2013

Argeo Platform

Integrating OSGi, Spring, Eclipse and JCR

Argeo GmbH, Berlin

- Tailor-made enterprise systems since 2007, mostly for the finance industry
- Over time a generic basis has emerged
- Main packaged product based on the framework is **Argeo SLC**
- Coordinator/host of the “Enterprise Linux GIS” effort at the OSGeo foundation

Argeo Platform - Principles

- Consistent set of OSGi bundles (**Argeo Distribution**)
- Integrate the various security models and provide utilities (**Argeo Commons**)
- Prototyping is fast, while preparing evolution towards a complex application
- Linux and Java can fit well

Not THE framework, to cover all needs

Argeo – Core Technologies

- Modularity and provisioning: **OSGi**
- Assembly and configuration: **Spring**
- User interface: **Eclipse**
 - RCP (desktop)
 - RAP (web)
- Data layer: **JCR / Jackrabbit**

Argeo 1 vs. Argeo 2

- **Argeo 1 (stable)**
 - stable and backward compatible since 2009 (so, getting slowly outdated)
 - open source since the beginning but no effort was made to build a community around it
 - OSGi: Equinox only, v3.6
 - Spring v2.5
 - Eclipse v3.6 / RAP 1.3
 - Jackrabbit v2.2

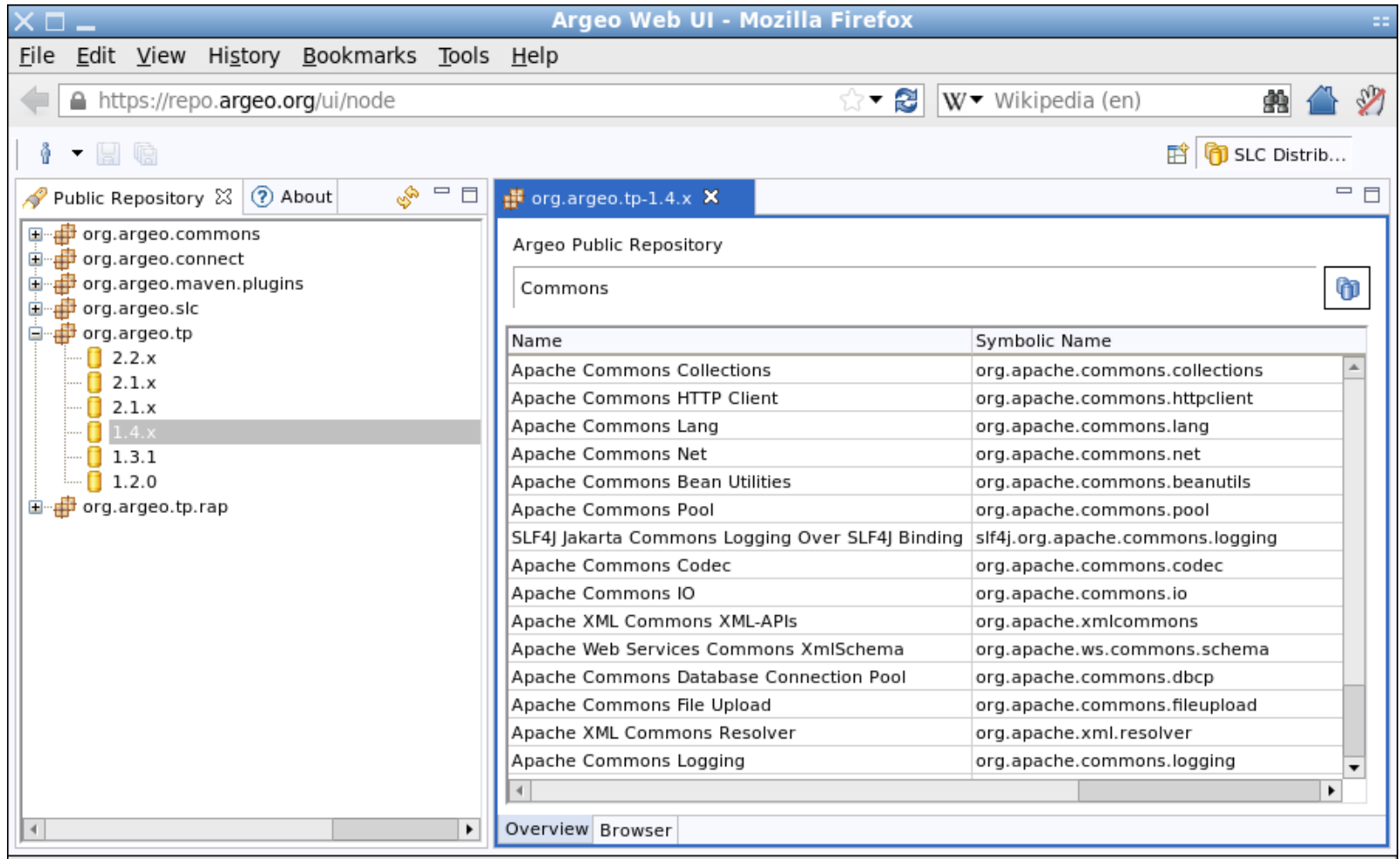
Argeo 1 vs. Argeo 2

- **Argeo 2 (under development)**
 - to be released early 2014
 - upgraded third parties
 - reusability by others (docs, SDK, community)
 - OSGi: support for Felix in addition to Equinox v3.9
 - Spring v3.2
 - Eclipse v4.3 / RAP 2.1
 - Jackrabbit v2.6

OSGi / Argeo Distribution

- 250+ open source modules, fully resolved in OSGi, available in maven-compatible repositories
- **[Argeo 2]** The factory used to maintain the distribution is easily forkable in order for others to create spins. Combined with **SLC Repo** it allows for a kind of 'git' for provisioning

OSGi / Argeo Distribution



Argeo Web UI - Mozilla Firefox

File Edit View History Bookmarks Tools Help

https://repo.argeo.org/ui/node

Public Repository About

- org.argeo.commons
- org.argeo.connect
- org.argeo.maven.plugins
- org.argeo.slc
- org.argeo.tp
 - 2.2.x
 - 2.1.x
 - 2.1.x
 - 1.4.x
 - 1.3.1
 - 1.2.0
- org.argeo.tp.rap

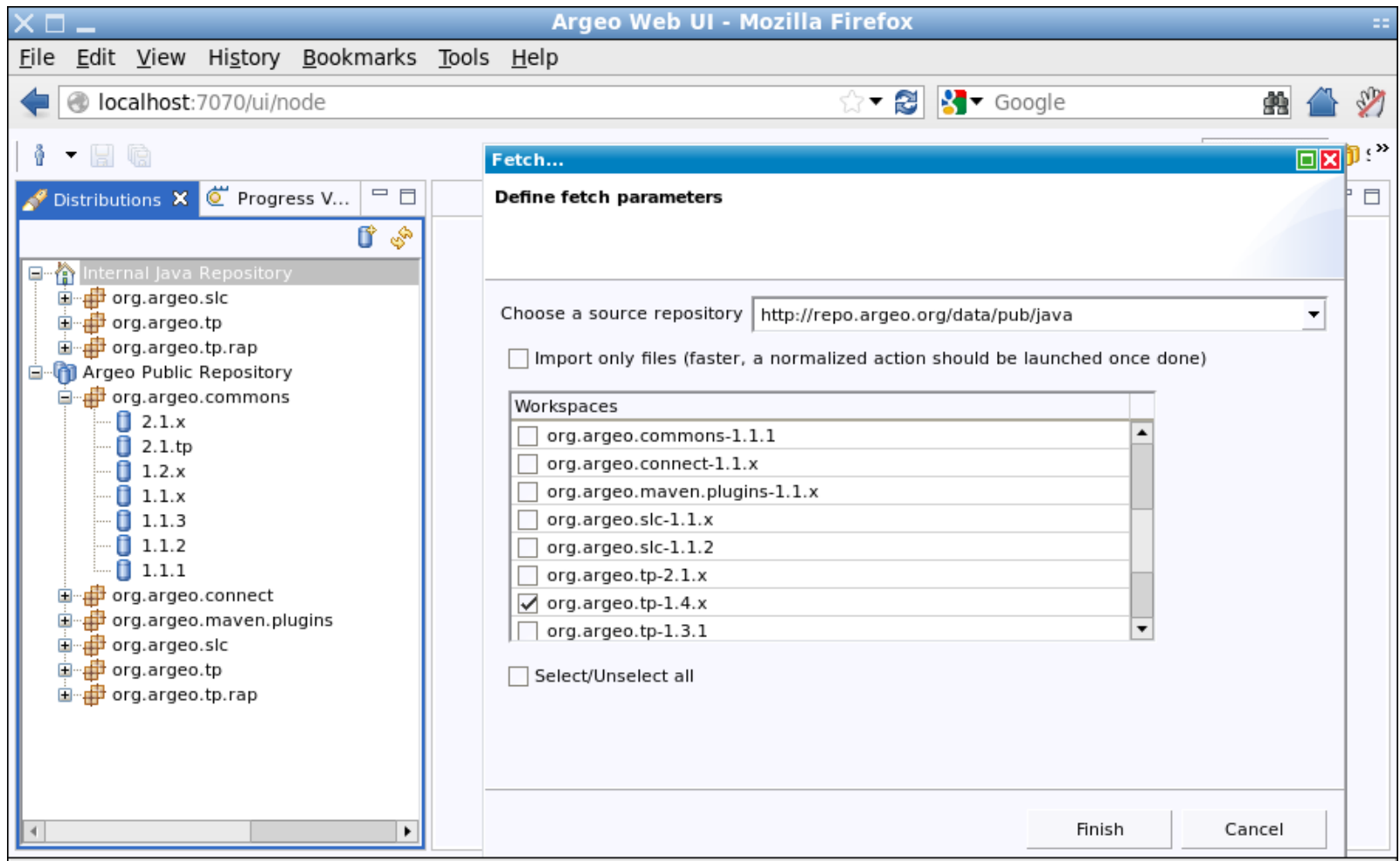
org.argeo.tp-1.4.x

Argeo Public Repository

Commons

Name	Symbolic Name
Apache Commons Collections	org.apache.commons.collections
Apache Commons HTTP Client	org.apache.commons.httpclient
Apache Commons Lang	org.apache.commons.lang
Apache Commons Net	org.apache.commons.net
Apache Commons Bean Utilities	org.apache.commons.beanutils
Apache Commons Pool	org.apache.commons.pool
SLF4J Jakarta Commons Logging Over SLF4J Binding	slf4j.org.apache.commons.logging
Apache Commons Codec	org.apache.commons.codec
Apache Commons IO	org.apache.commons.io
Apache XML Commons XML-APIs	org.apache.xmlcommons
Apache Web Services Commons XmlSchema	org.apache.ws.commons.schema
Apache Commons Database Connection Pool	org.apache.commons.dbcp
Apache Commons File Upload	org.apache.commons.fileupload
Apache XML Commons Resolver	org.apache.xml.resolver
Apache Commons Logging	org.apache.commons.logging

Overview Browser



- Spring-friendly containers for Tomcat, Jackrabbit, Apache Directory Server (LDAP), etc.
- Spring massively used and pushed to the limits in Argeo SLC
- **[Argeo 2]** Upgrade to Gemini Blueprint from Spring OSGi, expression language, REST support

```

<!-- Templates -->
<bean id="template.jcrRepository" abstract="true"
      class="org.argeo.jackrabbit.JackrabbitContainer" init-method="init"
      destroy-method="destroy">
  <property name="variables" value="osgibundle:/repo.properties" />
  <property name="cndFiles">
    <list>
      <value>/org/argeo/jcr/argeo.cnd</value>
      <value>/org/argeo/slc/jcr/slc.cnd</value>
      <value>/org/argeo/slc/repo/repo.cnd</value>
    </list>
  </property>
  <property name="bundleContext" ref="bundleContext" />
  <property name="forceCndImport" value="${slc.repo.jcr.forceCndImport}" />
</bean>

<!-- JCR repositories -->
<bean id="javaRepository" parent="template.jcrRepository">
  <property name="homeDirectory" value="${slc.repo.jcr.reposBase}/java" />
  <property name="configuration" value="${slc.repo.jcr.configuration.java}" />
</bean>
<bean id="distRepository" parent="template.jcrRepository">
  <property name="homeDirectory" value="${slc.repo.jcr.reposBase}/dist" />
  <property name="configuration" value="${slc.repo.jcr.configuration.dist}" />
</bean>

```

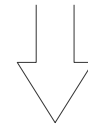
```

<!-- SERVICES -->
<service ref="javaRepository" interface="javax.jcr.Repository">
  <service-properties>
    <beans:entry key="argeo.jcr.repository.alias" value="java" />
  </service-properties>
</service>
<service ref="distRepository" interface="javax.jcr.Repository">
  <service-properties>
    <beans:entry key="argeo.jcr.repository.alias" value="dist" />
  </service-properties>
</service>

```



OSGi Services



```

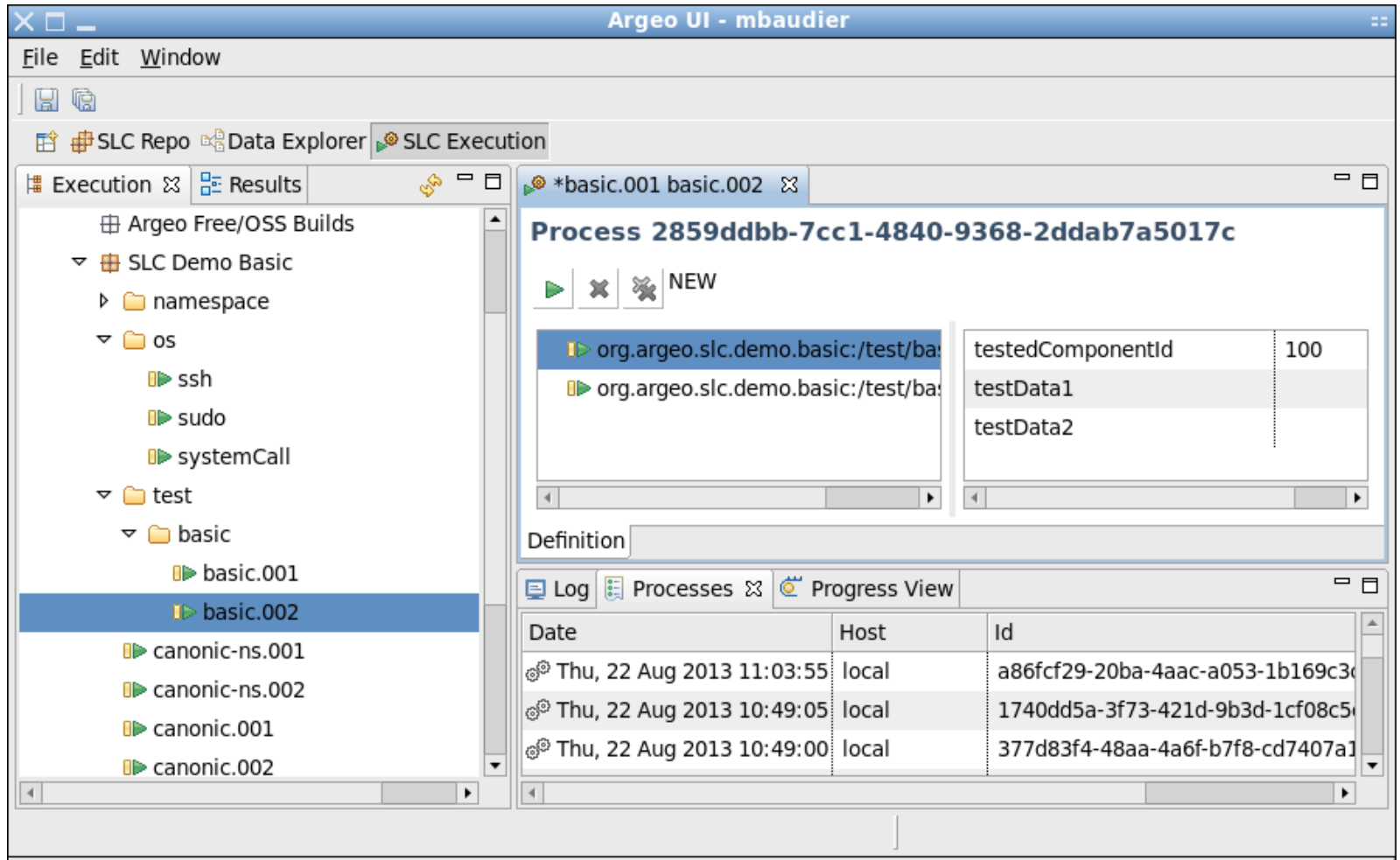
<!-- REFERENCES -->
<reference id="javaRepository" interface="javax.jcr.Repository"
  filter="(argeo.jcr.repository.alias=java)" />
<reference id="distRepository" interface="javax.jcr.Repository"
  filter="(argeo.jcr.repository.alias=dist)" />

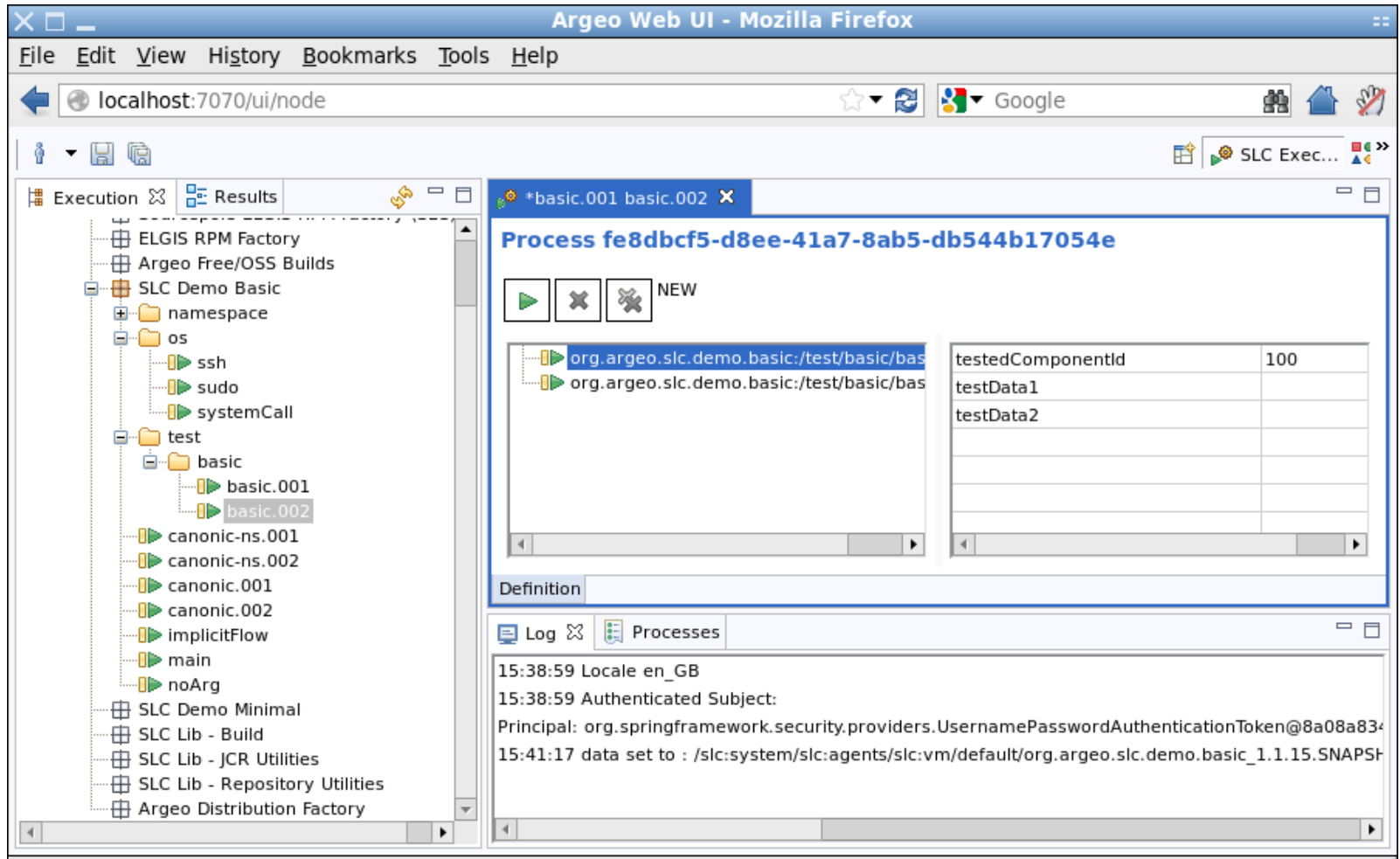
<!-- Example of injection into a POJO bean -->
<bean id="osgiFactory" class="org.argeo.slc.repo.osgi.OsgiFactoryImpl"
  init-method="init" destroy-method="destroy">
  <property name="workspace" value="org.argeo.tp-2.1.x" />
  <property name="nodeIndexers">
    <list>
      <bean class="org.argeo.slc.repo.ArtifactIndexer" />
      <bean class="org.argeo.slc.repo.JarFileIndexer" />
    </list>
  </property>
  <property name="javaRepository" ref="javaRepository" />
  <property name="distRepository" ref="distRepository" />
</bean>

```

Eclipse

- Integration with Spring: views and editors are defined as Spring beans so that OSGi services can be easily injected
- Single sourcing and JCR utilities
- **[Argeo 2]** Leverage RAP 2.x improvements to have more flexible design and templating and go beyond the workbench paradigm





The screenshot shows the Argeo Web UI interface. On the left, a tree view lists various components under 'SLC Demo Basic', including 'test/basic/basic.001' and 'basic.002'. The right pane shows the details for a process with ID 'fe8dbcf5-d8ee-41a7-8ab5-db544b17054e'. It includes a table of test data and a log window.

testedComponentId	100
testData1	
testData2	

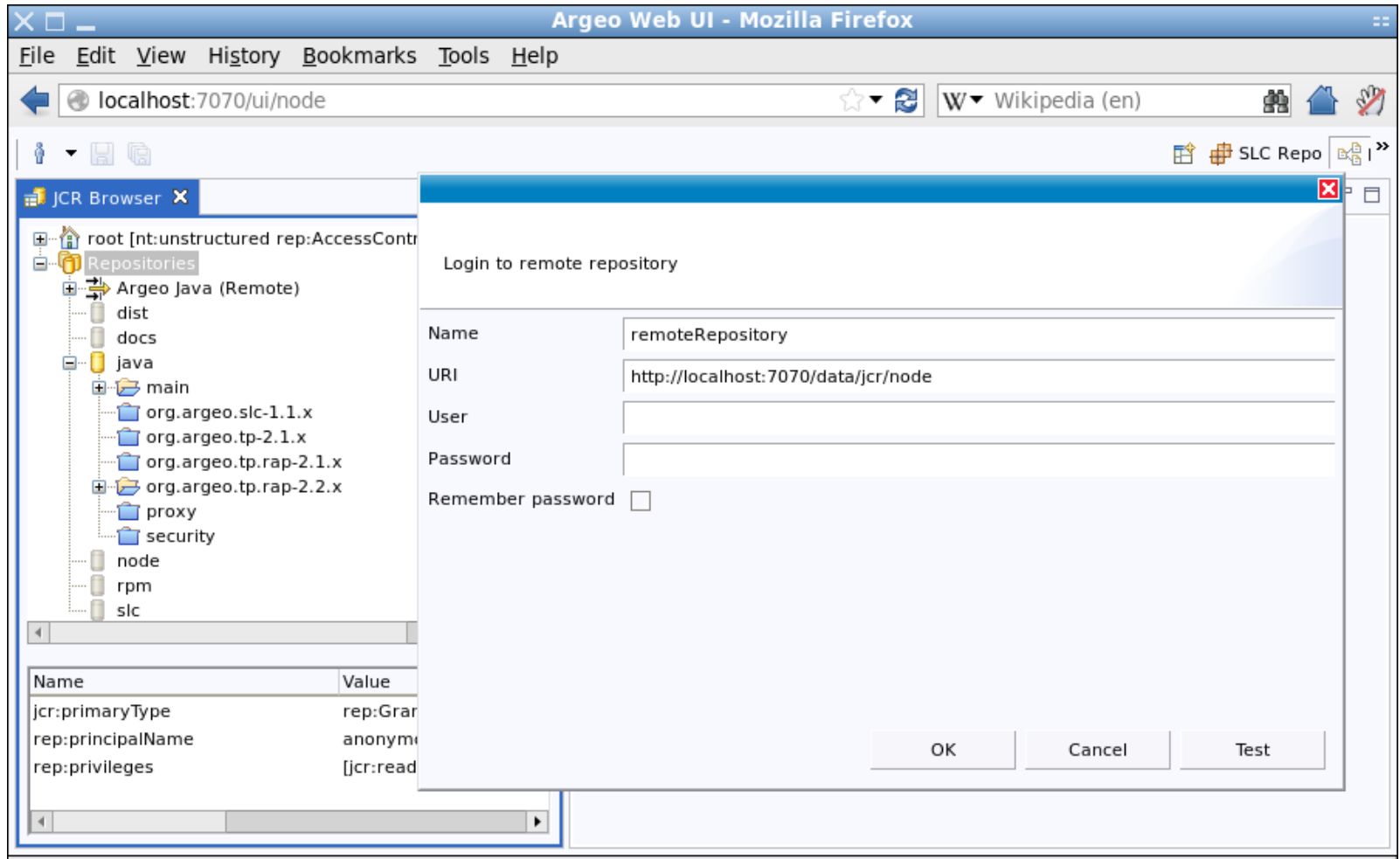
```

15:38:59 Locale en_GB
15:38:59 Authenticated Subject:
Principal: org.springframework.security.providers.UsernamePasswordAuthenticationToken@8a08a834
15:41:17 data set to : /slc:system/slc:agents/slc:vm/default/org.argeo.slc.demo.basic_1.1.15.SNAPSH
  
```

JCR - Multiple Repositories

- Multiple Jackrabbit repositories in one runtime referenced as OSGi services. Always a 'node' repository available
- All repositories exposed via remoting / webdav thanks to a Spring MVC layer integrating the Jackrabbit servlets
- **[Argeo 2]** Modeshape support. Generalized usage of remoting thanks to improvements in Jackrabbit.

JCR - Multiple Repositories



The screenshot shows the Argeo Web UI in Mozilla Firefox. The browser address bar displays `localhost:7070/ui/node`. The JCR Browser on the left shows a tree structure with a 'Repositories' folder containing 'Argeo Java (Remote)'. Underneath, there are folders for 'dist', 'docs', 'java', 'main', 'org.argeo.slc-1.1.x', 'org.argeo.tp-2.1.x', 'org.argeo.tp.rap-2.1.x', 'org.argeo.tp.rap-2.2.x', 'proxy', and 'security'. Below the tree is a table with the following data:

Name	Value
jcr:primaryType	rep:Gran
rep:principalName	anonym
rep:privileges	[jcr:read

A dialog box titled 'Login to remote repository' is open in the foreground. It contains the following fields and options:

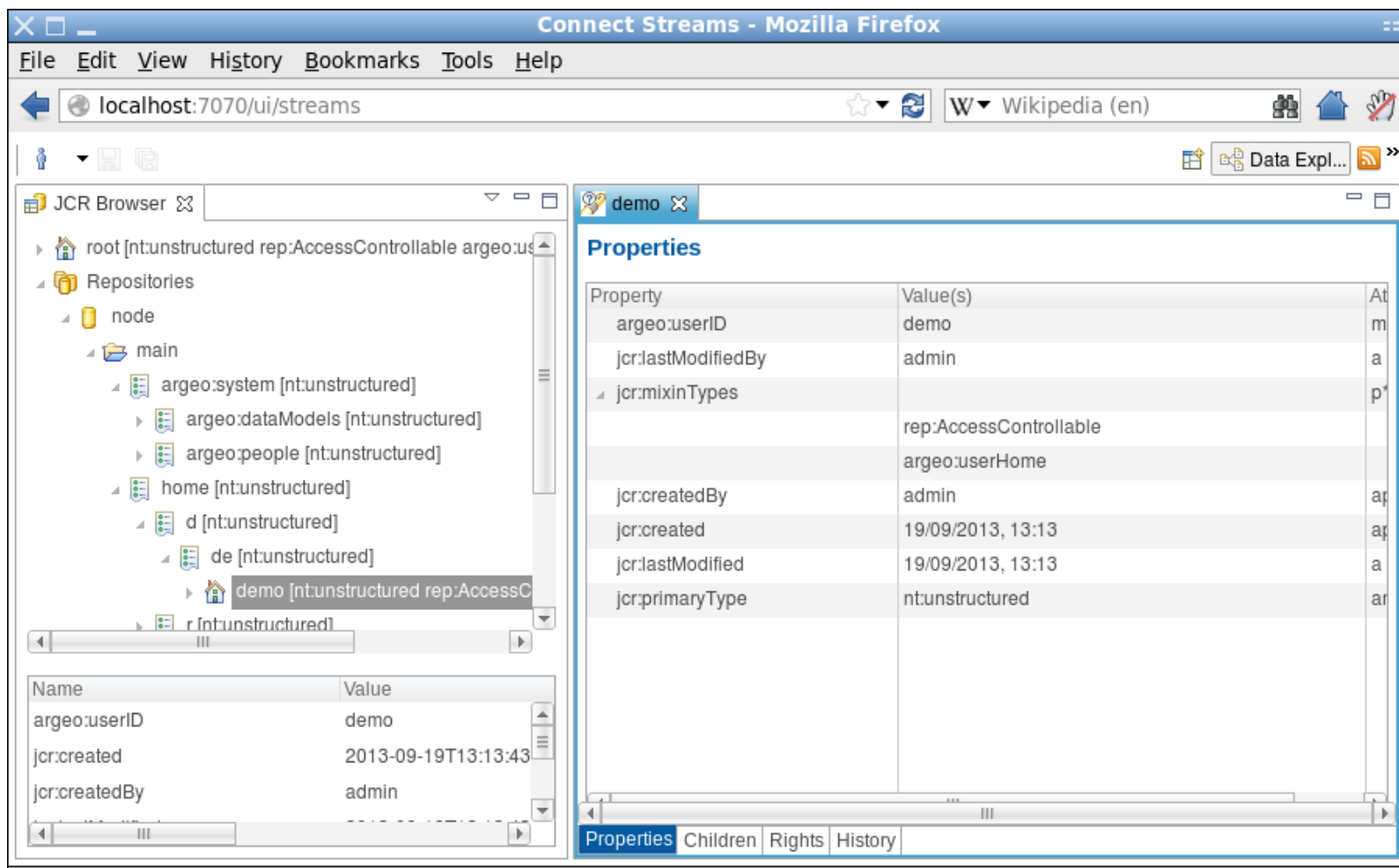
- Name: `remoteRepository`
- URI: `http://localhost:7070/data/jcr/node`
- User: (empty)
- Password: (empty)
- Remember password:

Buttons for 'OK', 'Cancel', and 'Test' are located at the bottom right of the dialog box.

JCR – Data Explorer

- Browse the multiple repositories in a runtime, or remote repositories
- Very useful for debugging and during development
- **[Argeo 2]** Finer management of security and versioning. Management of Jackrabbit specific features

JCR - Data Explorer



The screenshot shows the JCR Browser interface in Mozilla Firefox. The browser window title is "Connect Streams - Mozilla Firefox". The address bar shows "localhost:7070/ui/streams". The JCR Browser interface is split into two main panes.

The left pane shows a tree view of the JCR repository structure. The tree is expanded to show the "demo" node. The tree structure is as follows:

- root [nt:unstructured rep:AccessControllable argeo:us]
 - Repositories
 - node
 - main
 - argeo:system [nt:unstructured]
 - argeo:dataModels [nt:unstructured]
 - argeo:people [nt:unstructured]
 - home [nt:unstructured]
 - d [nt:unstructured]
 - de [nt:unstructured]
 - demo [nt:unstructured rep:AccessC] (selected)

The right pane shows the "Properties" view for the selected node. The properties are listed in a table:

Property	Value(s)	At
argeo:userID	demo	m
jcr:lastModifiedBy	admin	a
jcr:mixinTypes	rep:AccessControllable	p
jcr:createdBy	admin	ap
jcr:created	19/09/2013, 13:13	ap
jcr:lastModified	19/09/2013, 13:13	a
jcr:primaryType	nt:unstructured	ar

At the bottom of the right pane, there are tabs for "Properties", "Children", "Rights", and "History".

Security Integration 1/2

- Security integrated between JAAS, Eclipse and Jackrabbit with Spring Security as pivot
- A given thread has consistent username and authorities through all layers
- Init/destroy methods of beans can be configured to have admin/system rights

Security Integration 2/2

- Each user has a home JCR node in the 'node' repository (similar to a UNIX home directory) where modules can write
- Integrated with
 - LDAP for server applications
 - OS authentication for desktop
 - Remote Jackrabbit for thin clients

Connect Streams - Mozilla Firefox

File Edit View History Bookmarks Tools Help

localhost:7070/ui/streams

Wikipedia (en)

Security

Users Roles

Type filter criterion separated by a space (on user ID, name and E-mail)

User ID	Name	E-mail	Description
demo	demo User	demo@localhost	Demo user
root	demo User	root@localhost	Superuser

demo User

General

Displayed Name

First name

Last name

Email

Description

Password

New password

Repeat password

Main Roles

Demo - SLC Runtime

- Based on Argeo 1, Desktop
- Execute processes defined as standard Java Runnable via Spring
- Used for builds, tests, configuration management, building RPMs, etc.
- Argeo Distribution factory based on it

Demo – RSS Reader 'Streams'

- Based on Argeo 2, Web
- As Google Reader has been retired, we wanted to illustrate that free, distributed, innovative alternatives are possible
- Prototyped in 3 man-days
- Note improvements in styling and instant search with [\[Argeo 2\]](#)



<http://www.argeo.org>

Mathieu Baudier
mbaudier@argeo.org

Argeo GmbH, Weigandufer 45, D-12059 Berlin