OSLC Consumer with Eclipse Lyo Project

Jean-Luc Johnson, AIRBUS Group Innovations
Gray Bachelor, IBM Rational
Samit Mehta, IBM Rational
Harry Reeder, IBM Rational

Your team today

- Jean-Luc Johnson (Airbus Group Innovations)

- Gray Bachelor (IBM Rational CTO Office)

- Samit Mehta (IBM Rational: ISV enablement and Ready for Rational)

- Harry Reeder (IBM Rational UK)

Crystal tool interoperability agenda

- 14:00-14:30 – Introduction and environment setup
- 14:30-15:00 – LAB 1
  - Bugzilla change request resource Java class
  - Discovery system to access the service
  - Summary
- 15:00-15:30 – LAB 2
  - Retrieve list of products in Bugzilla
- 15:30-17:00 – LAB 3
  - Deeper "Hands on" session with Lyo to continue
- 17:00-17:30 – Showcase
  - Java Swing GUI

OSLC Consumer workshop

Introduction and Overview

OSLC Consumer workshop

Initial setup

Initial setup

- Pre-requisites
  - Eclipse 4.2 : https://eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/keplersr2
    - Java JDK 1.6 + (Mandatory)
  - Stable version of Lyo: http://www.eclipse.org/downloads/download.php?file=/lyo/releases/2.1.0/org.eclipse.lyo.oslc4j-2.1.0.zip
    - Include in the labs
  - Follow the instruction at http://wiki.eclipse.org/Lyo/BuildOSLC4JBugzilla
    - Register yourself (Username/password) to https://landfill.bugzilla.org/bugzilla-4.2-branch/createaccount.cgi
    - Build the OSLC4JBugzilla service provider application
- **NOTES:**
  - We provide the code of the labs (Including the stable release of Eclipse Lyo)
  - We provide a bundle Tomcat + OSLC4JBugzilla war file

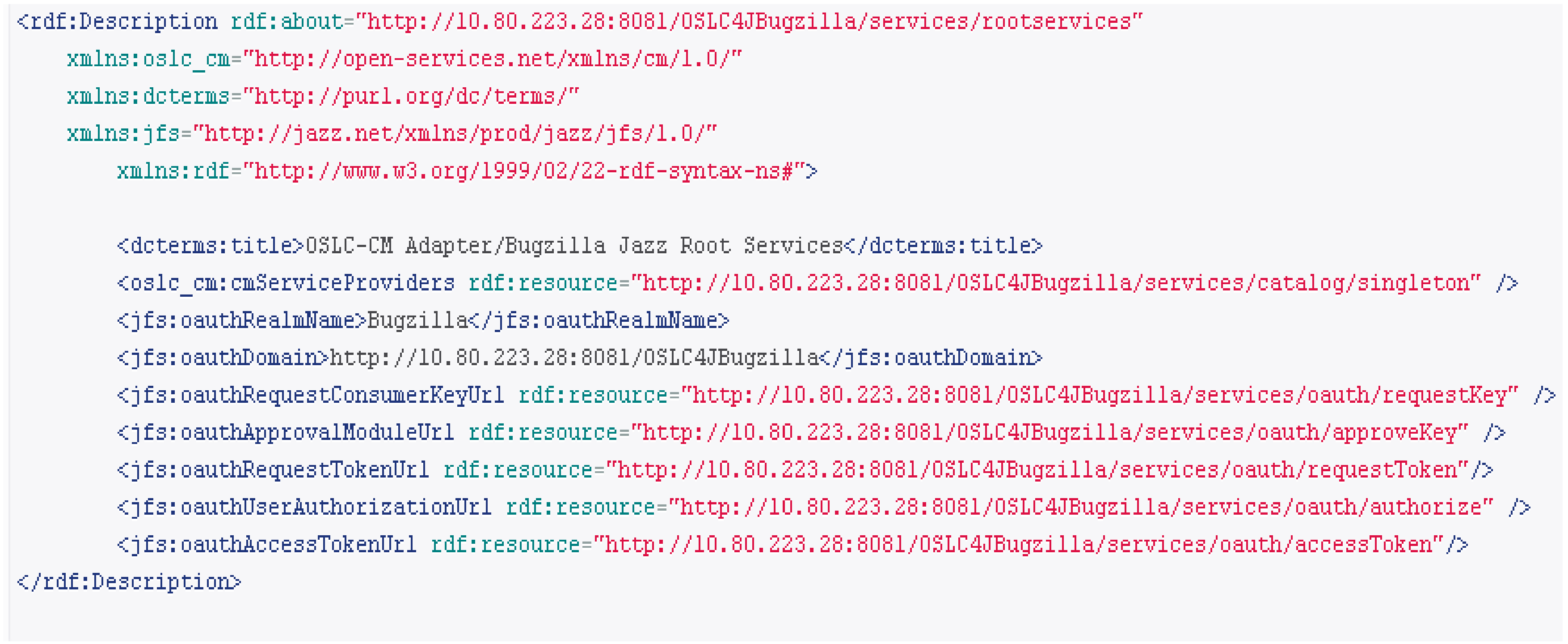
Check list

- ✓ Unzip the downloaded apache-tomcat-7.0.37.zip
  - ✓ Run on Port number 8080
  - ✓ Contains OSLC4JBugzilla.war
- ✓ Start Tomcat at [Tomcat-folder]\bin\startup.bat, startup.sh,….
  - ✓ Please select the option that applies to your Operating System
- ✓ Register at https://landfill.bugzilla.org/bugzilla-4.2-branch/createaccount.cgi
  - ✓ Username/password
- ✓ Start your Eclipse IDE
- ✓ Import the OSLC4JClientBugzilla.zip file into Eclipse
  - ✓ File > Import > Existing projects into Workspace (under general)
  - ✓ Option Select archive file
  - ✓ Browse to file
  - ✓ Finish

Note:
OSLC4JClientBugzilla includes the stable release Eclipse Lyo 2.1.0

Test environment setup

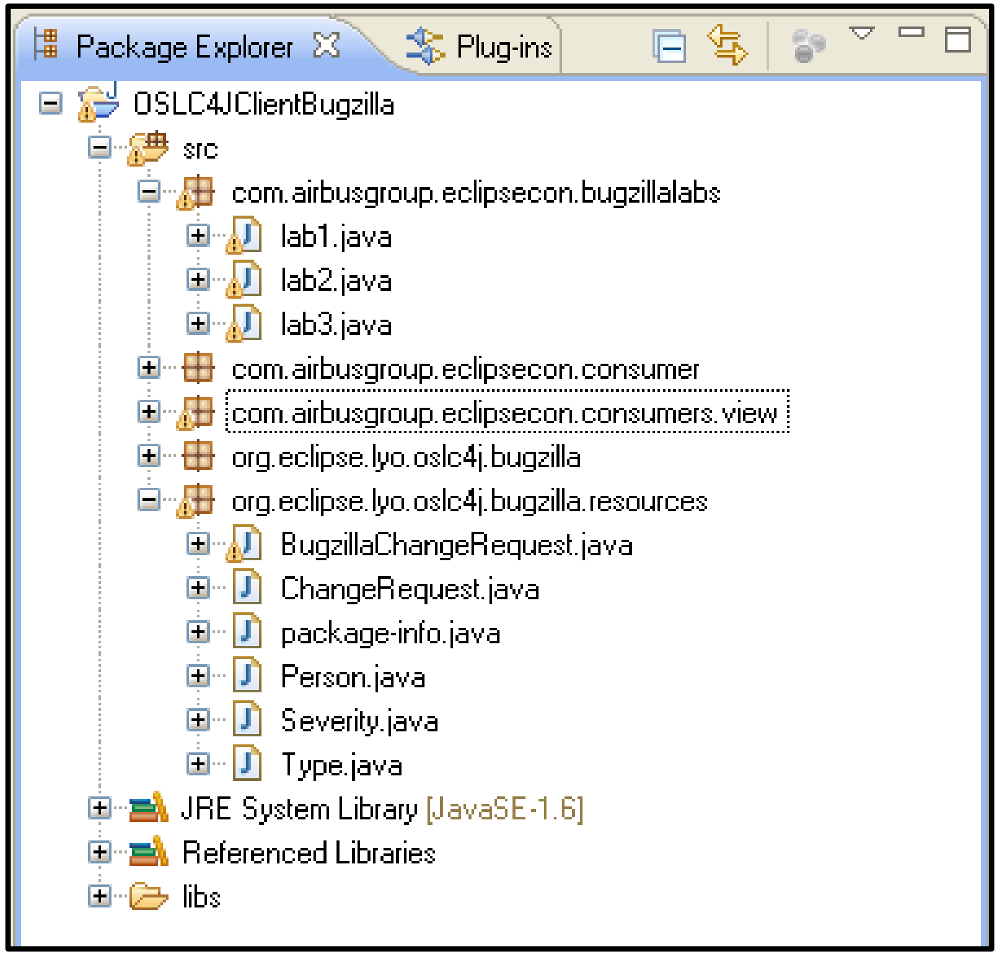
- Tomcat
  - Navigate to http://localhost:8080/OSLC4JBugzilla/rootservices



```
<rdf:Description rdf:about="http://10.80.223.28:8081/OSLC4JBugzilla/services/rootservices"
    xmlns:oslc_cm="http://open-services.net/xmlns/cm/1.0/"
    xmlns:dcterms="http://purl.org/dc/terms/"
    xmlns:jfs="http://jazz.net/xmlns/prod/jazz/jfs/1.0/"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

        <dcterms:title>OSLC-CM Adapter/Bugzilla Jazz Root Services</dcterms:title>
        <oslc_cm:cmServiceProviders rdf:resource="http://10.80.223.28:8081/OSLC4JBugzilla/services/catalog/singleton" />
        <jfs:oauthRealmName>Bugzilla</jfs:oauthRealmName>
        <jfs:oauthDomain>http://10.80.223.28:8081/OSLC4JBugzilla</jfs:oauthDomain>
        <jfs:oauthRequestConsumerKeyUrl rdf:resource="http://10.80.223.28:8081/OSLC4JBugzilla/services/oauth/requestKey" />
        <jfs:oauthApprovalModuleUrl rdf:resource="http://10.80.223.28:8081/OSLC4JBugzilla/services/oauth/approveKey" />
        <jfs:oauthRequestTokenUrl rdf:resource="http://10.80.223.28:8081/OSLC4JBugzilla/services/oauth/requestToken"/>
        <jfs:oauthUserAuthorizationUrl rdf:resource="http://10.80.223.28:8081/OSLC4JBugzilla/services/oauth/authorize" />
        <jfs:oauthAccessTokenUrl rdf:resource="http://10.80.223.28:8081/OSLC4JBugzilla/services/oauth/accessToken"/>
</rdf:Description>
```

Test environment setup

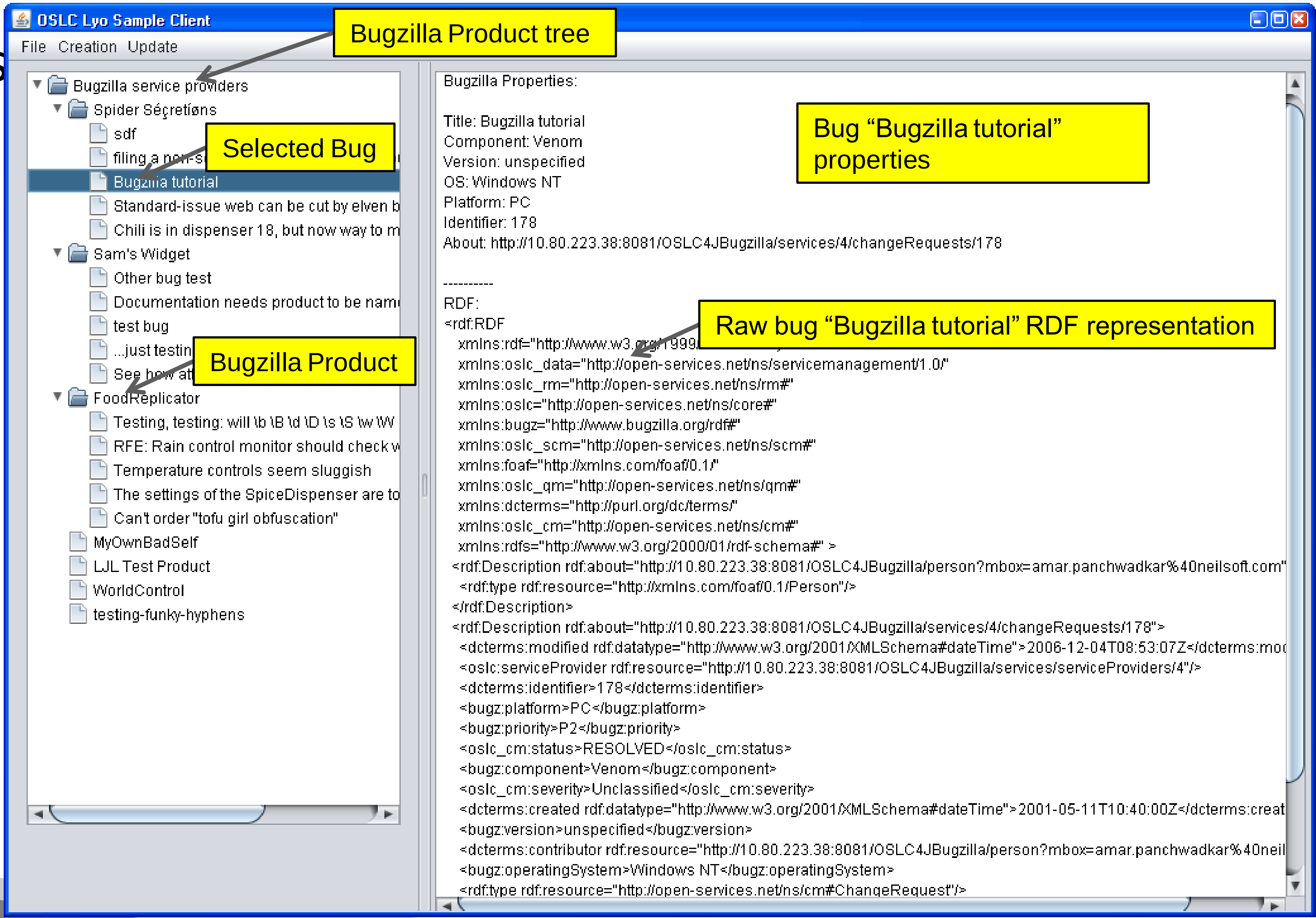
- Eclipse Package Explorer view
  - Labs
  - Swing client
  - Resource classes



```
OSLC4JClientBugzilla
  src
    com.airbusgroup.eclipsecon.bugzillalabs
      lab1.java
      lab2.java
      lab3.java
    com.airbusgroup.eclipsecon.consumer
    com.airbusgroup.eclipsecon.consumers.view
    org.eclipse.lyo.oslc4j.bugzilla
    org.eclipse.lyo.oslc4j.bugzilla.resources
      BugzillaChangeRequest.java
      ChangeRequest.java
      package-info.java
      Person.java
      Severity.java
      Type.java
  JRE System Library [JavaSE-1.6]
  Referenced Libraries
  libs
```

Test Bugzilla Java Swing consumer Application

- Run Consumer GUI class
  - Expand Package view
  - Open ConsumerGUI.java
  - Set your username/password
- Output:

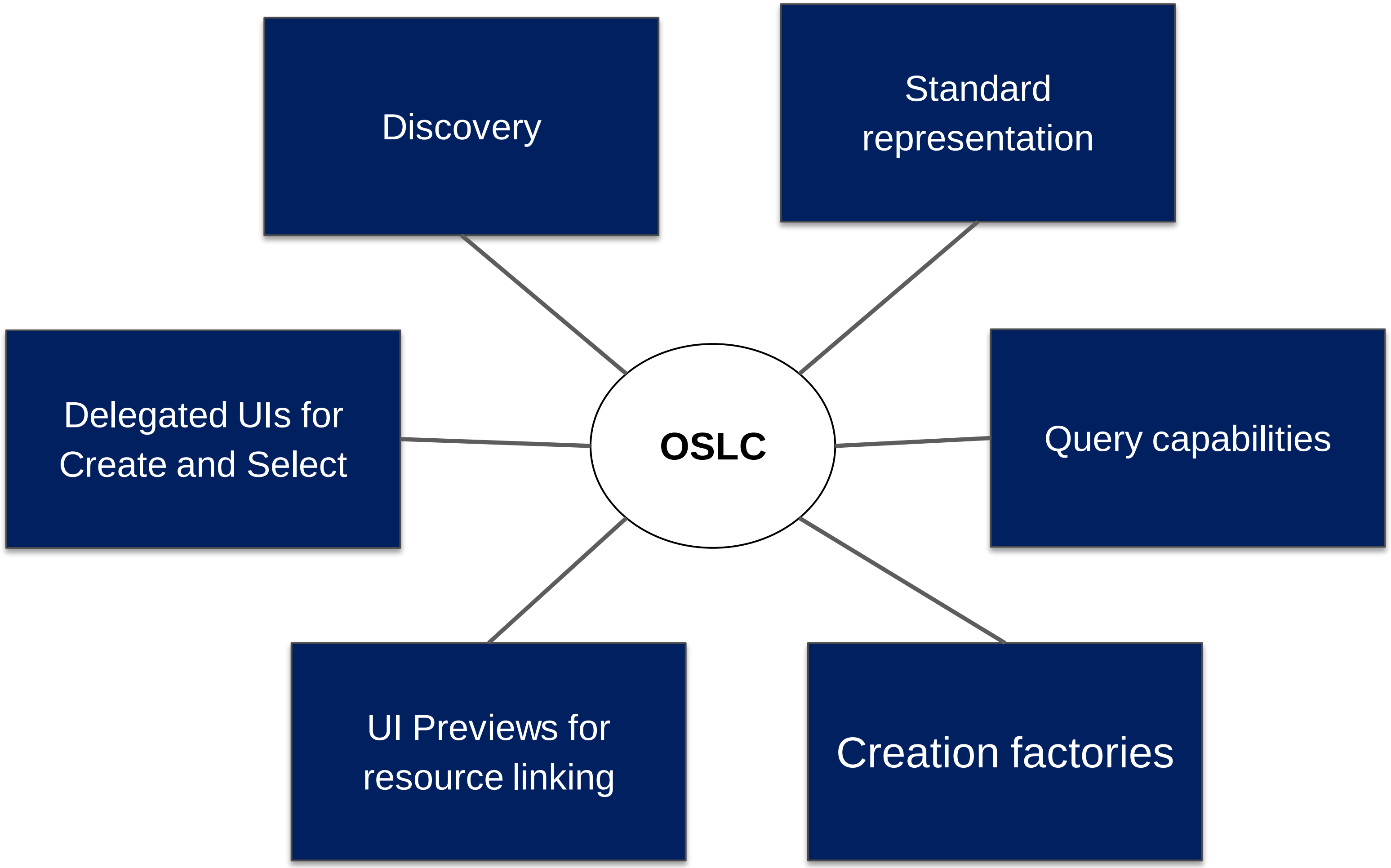


```
Bugzilla service providers
  Spider Sècretíøns
    sdf
    filing a non-s
    Bugzilla tutorial
    Standard-issue web can be cut by elven b
    Chili is in dispenser 18, but now way to m
  Sam's Widget
    Other bug test
    Documentation needs product to be nam
    test bug
    ...just testin
    See how at
  FoodReplicator
    Testing, testing: will \b \B \d \D \s \S \w \W
    RFE: Rain control monitor should check w
    Temperature controls seem sluggish
    The settings of the SpiceDispenser are to
    Can't order "tofu girl obfuscation"
  MyOwnBadSelf
  LJL Test Product
  WorldControl
  testing-funky-hyphens
```

```
Bugzilla Properties:

Title: Bugzilla tutorial
Component: Venom
Version: unspecified
OS: Windows NT
Platform: PC
Identifier: 178
About: http://10.80.223.38:8081/OSLC4JBugzilla/services/4/changeRequests/178

----------
RDF:
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999
  xmlns:oslc_data="http://open-services.net/ns/servicemanagement/1.0/"
  xmlns:oslc_rm="http://open-services.net/ns/rm#"
  xmlns:oslc="http://open-services.net/ns/core#"
  xmlns:bugz="http://www.bugzilla.org/rdf#"
  xmlns:oslc_scm="http://open-services.net/ns/scm#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:oslc_qm="http://open-services.net/ns/qm#"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:oslc_cm="http://open-services.net/ns/cm#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >
 <rdf:Description rdf:about="http://10.80.223.38:8081/OSLC4JBugzilla/person?mbox=amar.panchwadkar%40neilsoft.com"
  <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://10.80.223.38:8081/OSLC4JBugzilla/services/4/changeRequests/178">
  <dcterms:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2006-12-04T08:53:07Z</dcterms:mod
  <oslc:serviceProvider rdf:resource="http://10.80.223.38:8081/OSLC4JBugzilla/services/serviceProviders/4"/>
  <dcterms:identifier>178</dcterms:identifier>
  <bugz:platform>PC</bugz:platform>
  <bugz:priority>P2</bugz:priority>
  <oslc_cm:status>RESOLVED</oslc_cm:status>
  <bugz:component>Venom</bugz:component>
  <oslc_cm:severity>Unclassified</oslc_cm:severity>
  <dcterms:created rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2001-05-11T10:40:00Z</dcterms:creat
  <bugz:version>unspecified</bugz:version>
  <dcterms:contributor rdf:resource="http://10.80.223.38:8081/OSLC4JBugzilla/person?mbox=amar.panchwadkar%40neil
  <bugz:operatingSystem>Windows NT</bugz:operatingSystem>
  <rdf:type rdf:resource="http://open-services.net/ns/cm#ChangeRequest"/>
```

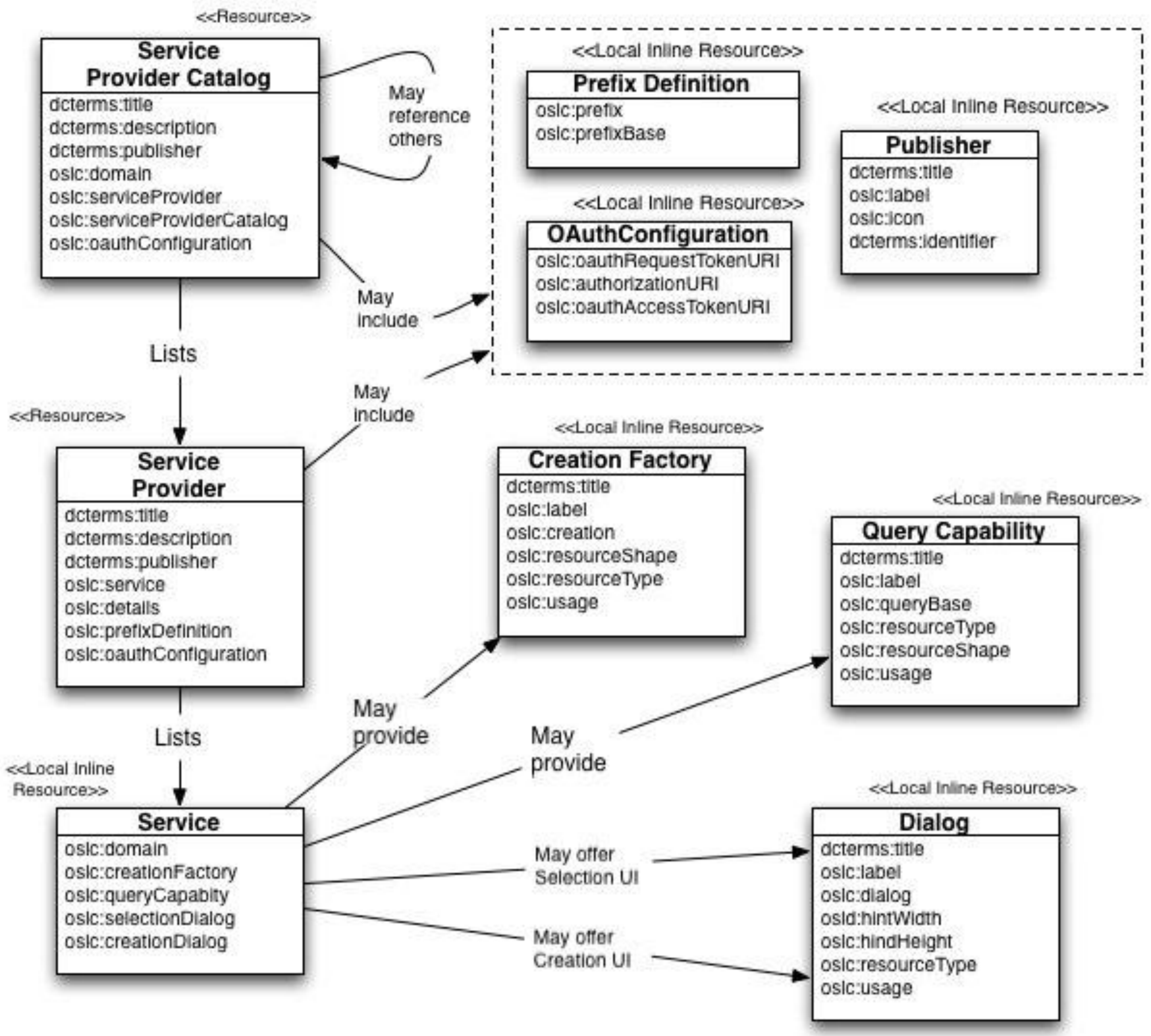
OSLC Capabilities

- Discovery
- Standard Resource representation
- HTTP C.R.U.D for resources
- Query capabilities
- Creation factory
- **Not covered**
  - UI preview for resource Links
  - Delegated UIs for create and Select



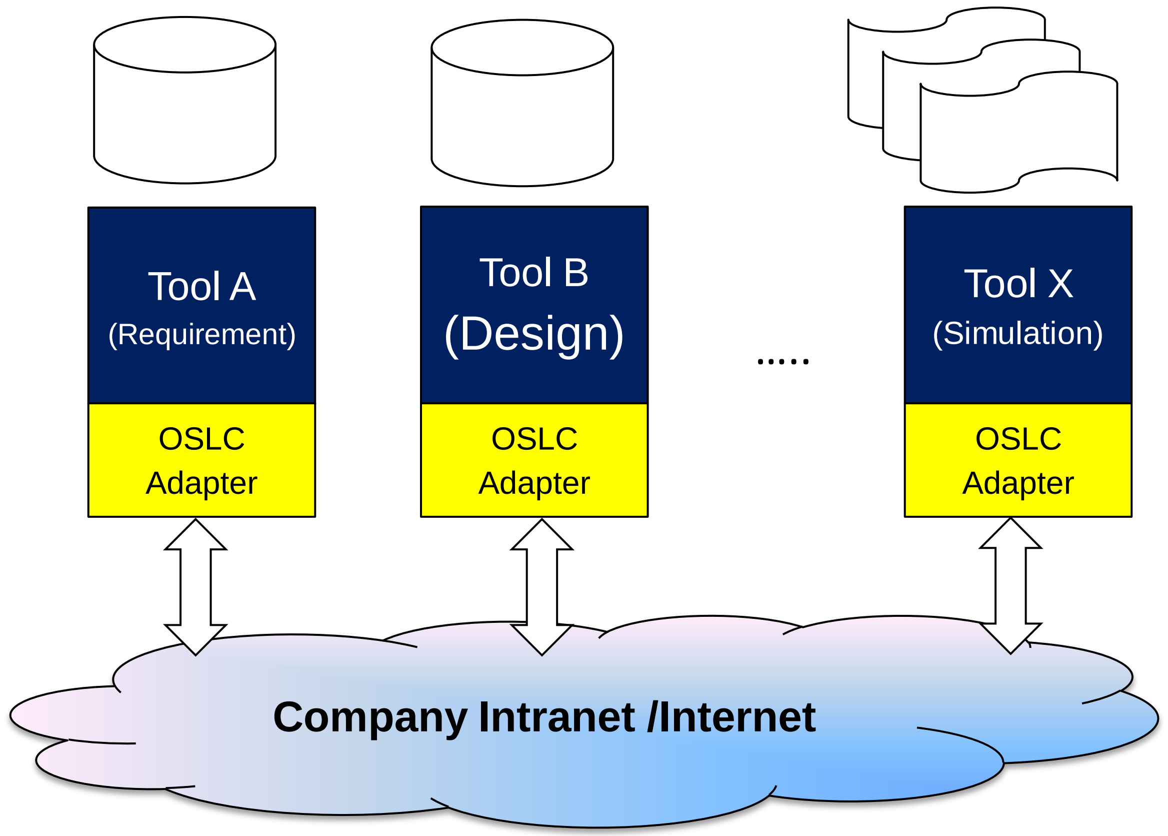
OSLC Architecture

- Service provider catalog
  - Service providers
    - Services
      - **Query capabilities**
      - **Creation Factory**
      - **Dialogs**



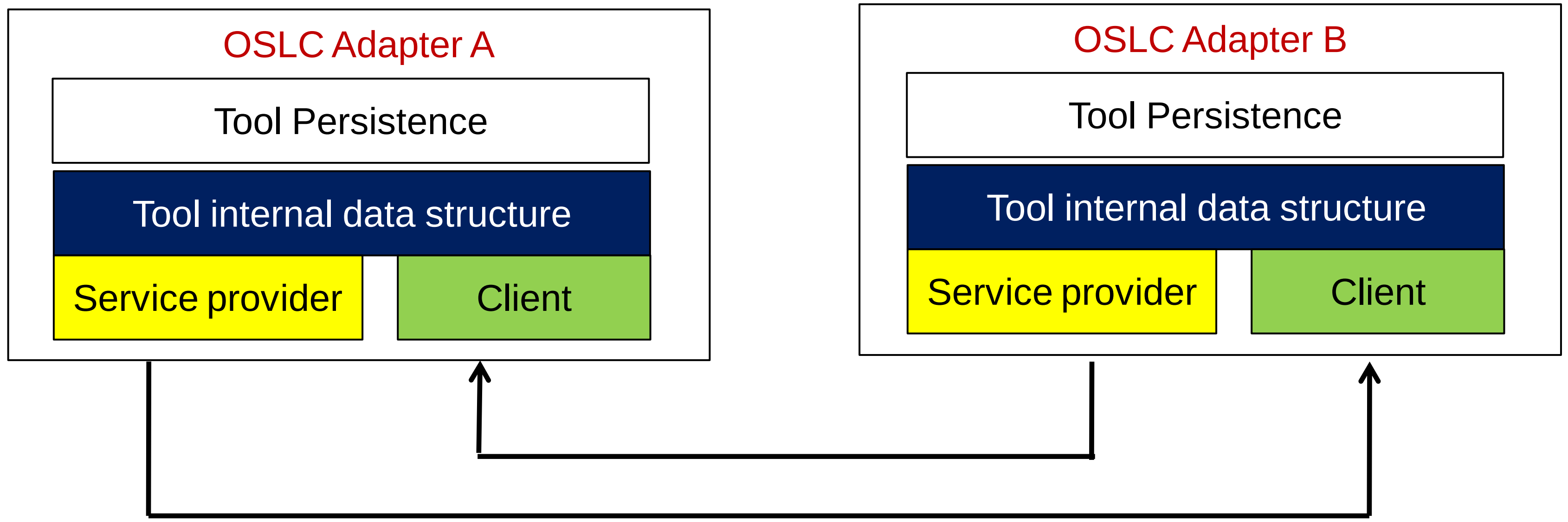
Integration based on OSLC

- An artefact within a tool is connected to another artefact based on the link data approach



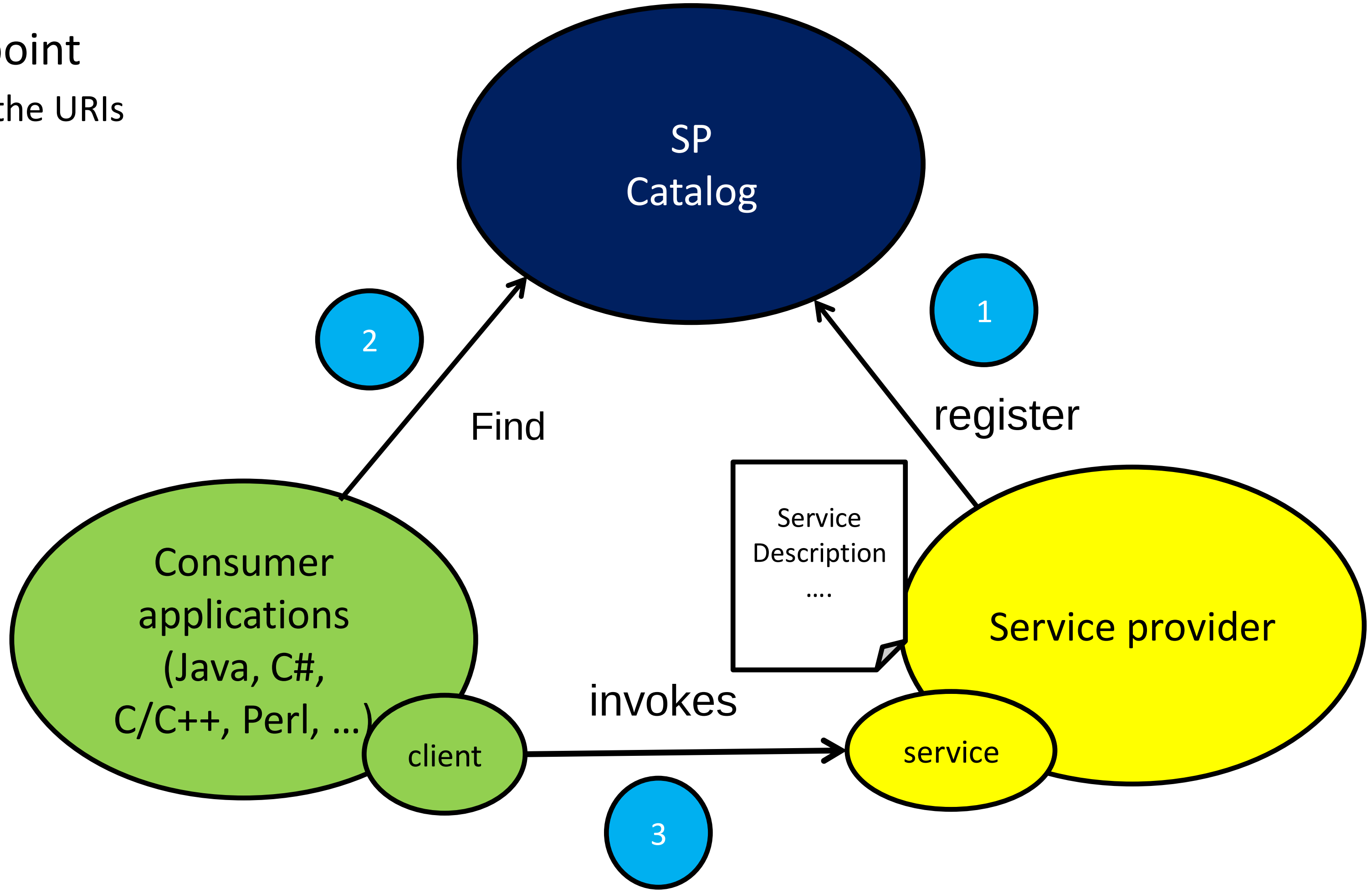
OSLC adapter concepts

- A tool may have both a service provider adapter and a consumer adapter
- But it is not mandatory



OSLC adapter based on the SOA approach

- /Rootservice document as entry point
  - We don’t want the consumer to guess the URIs
- Service providers
  - Advertise themselves to the catalog
- Consumer applications
  - Discover service providers
  - Discover services available
- Consume applications
  - invoke services (CRUD)

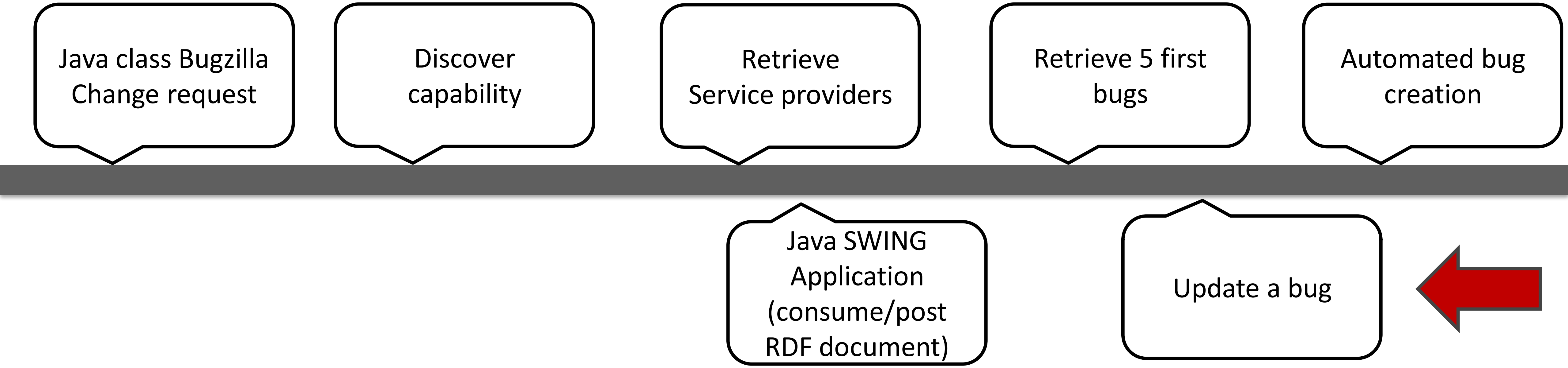


What examples do we see for OSLC interoperability ?

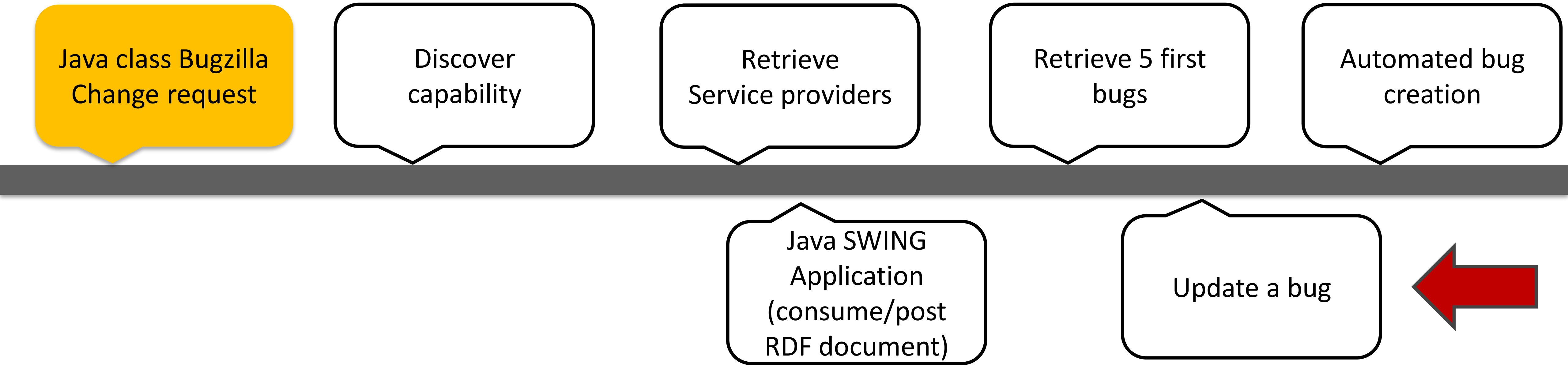
- Native integration examples
  - Jazz platform,
  - HP Quality centre,
  - Microsoft SharePoint, …
  - Salesforce
- Plug-in/Wrapper examples
  - Open Modelica
- Gateway examples
  - Bugzilla approach

More examples at http://open-services.net/software/

Our journey

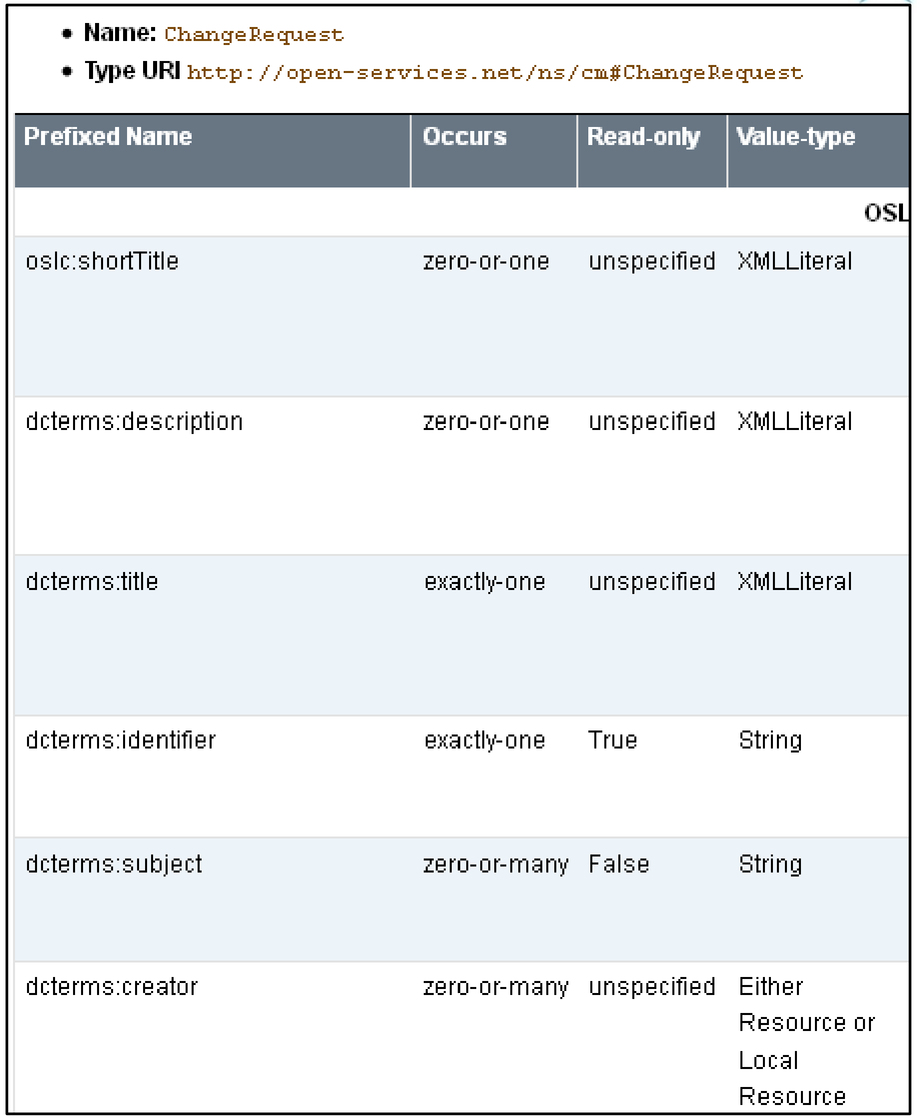

Java class Bugzilla Change request
Discover capability
Retrieve Service providers
Retrieve 5 first bugs
Automated bug creation
Java SWING Application (consume/post RDF document)
Update a bug

LAB 1: Java class bugzilla Change request.


Java class Bugzilla Change request
Discover capability
Retrieve Service providers
Retrieve 5 first bugs
Automated bug creation
Java SWING Application (consume/post RDF document)
Update a bug

Java classes ChangeRequest and BugzillaChangeRequest

- Look at the OSLC CM specification
- Open package org.eclipse.lyo.oslc4j.bugzilla.resources
  - Open ChangeRequest.java
- Open BugzillaChangeRequest.java
  - Extend ChangeRequest.java with
    - Product
    - Version
    - priority
    - component

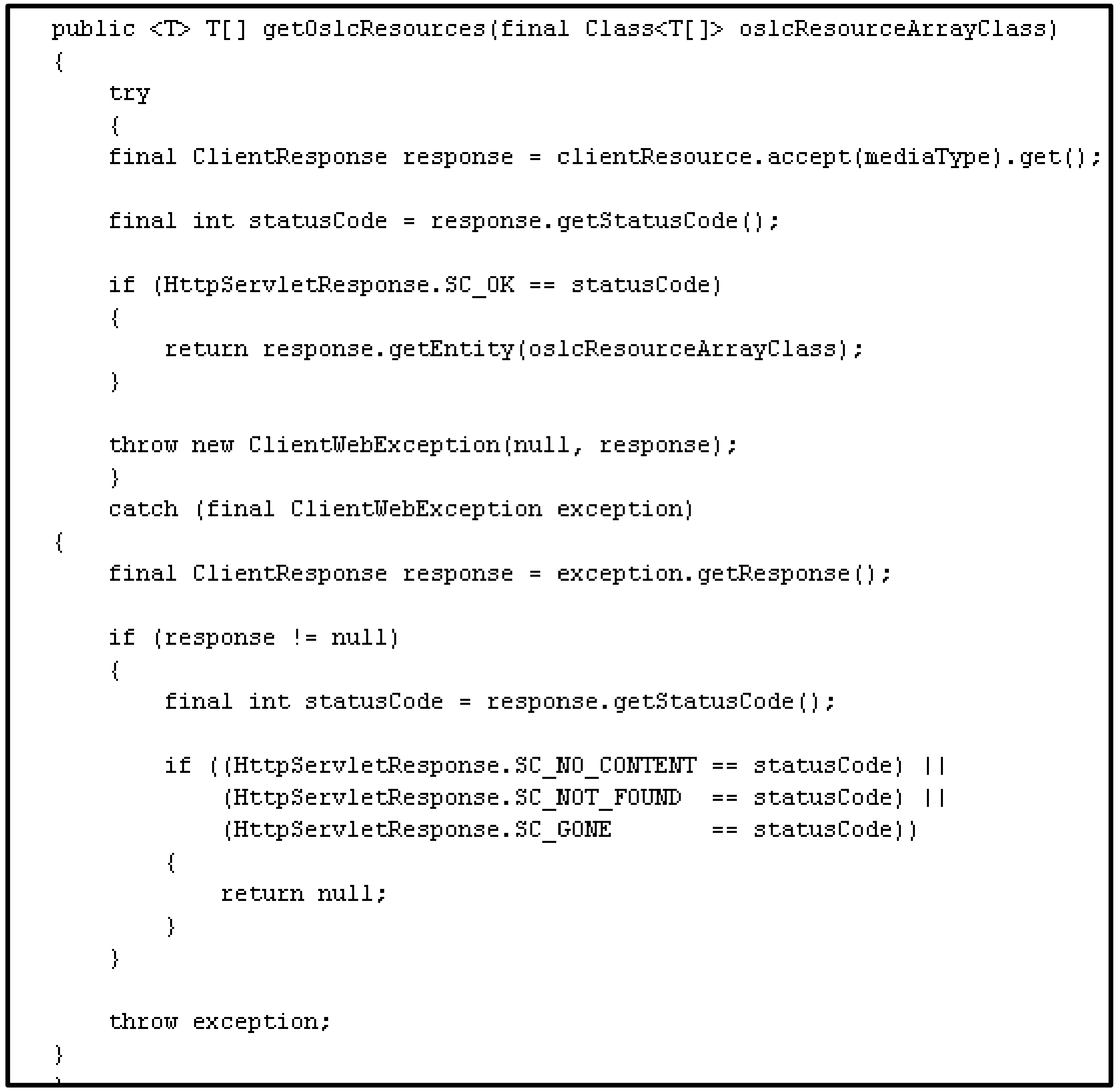


- **Name:** ChangeRequest
- **Type URI** http://open-services.net/ns/cm#ChangeRequest

Prefixed Name	Occurs	Read-only	Value-type
			OSL
oslc:shortTitle	zero-or-one	unspecified	XMLLiteral
dcterms:description	zero-or-one	unspecified	XMLLiteral
dcterms:title	exactly-one	unspecified	XMLLiteral
dcterms:identifier	exactly-one	True	String
dcterms:subject	zero-or-many	False	String
dcterms:creator	zero-or-many	unspecified	Either Resource or Local Resource

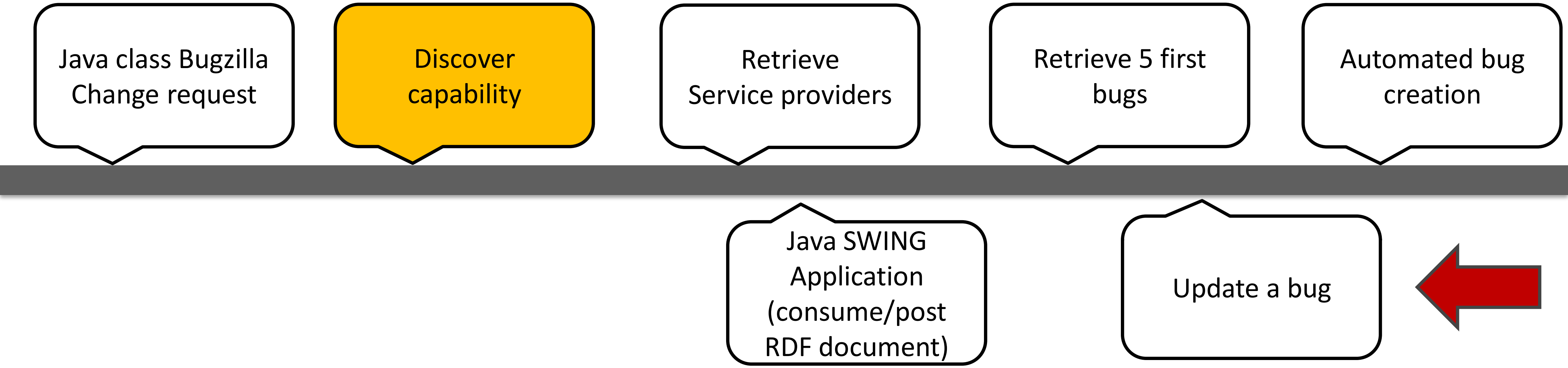
Java OSLC Rest Client

- Identify the class OSLCRestClient
  - Open Lyo wink project
    - Src/org/eclipse/lyo/oslc4j/OsclRestClient.java
  - OslcRestClient, key java class of the labs



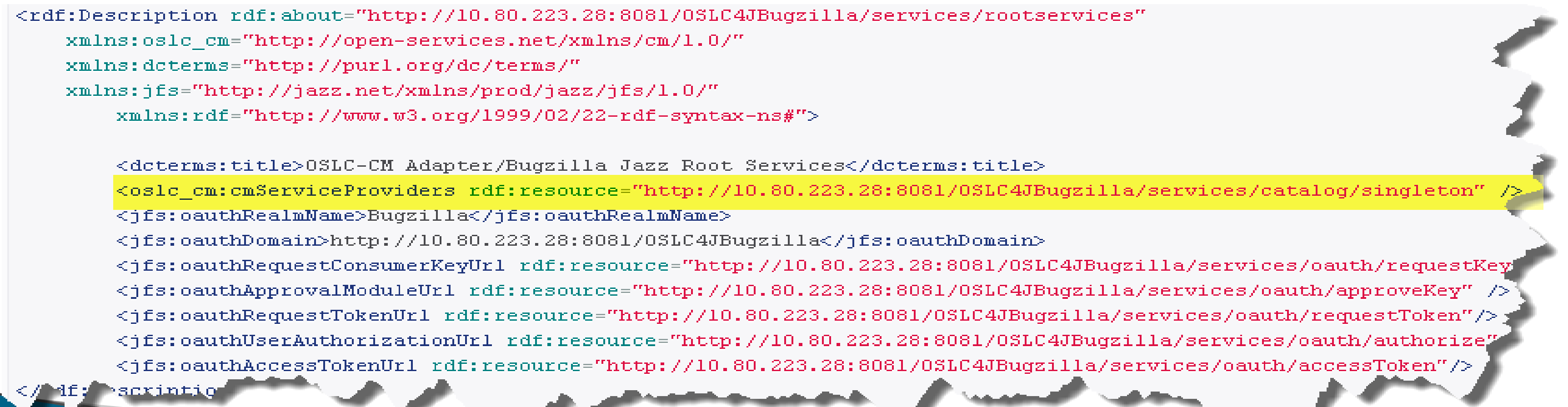
```
public <T> T[] getOslcResources(final Class<T[]> oslcResourceArrayClass)
{
    try
    {
    final ClientResponse response = clientResource.accept(mediaType).get();

    final int statusCode = response.getStatusCode();

    if (HttpServletResponse.SC_OK == statusCode)
    {
        return response.getEntity(oslcResourceArrayClass);
    }

    throw new ClientWebException(null, response);
    }
    catch (final ClientWebException exception)
{
    final ClientResponse response = exception.getResponse();

    if (response != null)
    {
        final int statusCode = response.getStatusCode();

        if ((HttpServletResponse.SC_NO_CONTENT == statusCode) ||
            (HttpServletResponse.SC_NOT_FOUND  == statusCode) ||
            (HttpServletResponse.SC_GONE       == statusCode))
        {
            return null;
        }
    }

    throw exception;
}
```

LAB 1 cont'd: Discovery capability- Rootservices as entry point


Java class Bugzilla Change request
Discover capability
Retrieve Service providers
Retrieve 5 first bugs
Automated bug creation
Java SWING Application (consume/post RDF document)
Update a bug

Lab 1: resource class overview - Rootservices document

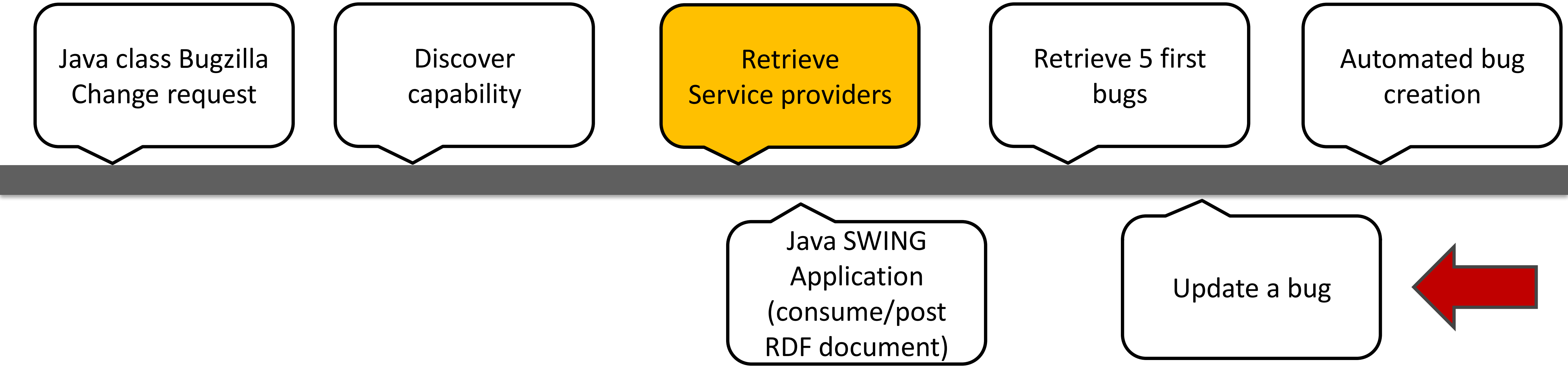
- Actions:
  - Locate Rootservices URL (entry point)
    - http://localhost:8080/OSLC4JBugzilla/rootservices
  - Write a java code to parse Rootservices document to extract catalog URI
    - Use Java class JazzRootServicesHelper
- Output: URL of the Service Provider Catalog
  - http://localhost:8080/OSLC4JBugzilla/services/catalog/singleton



```
<rdf:Description rdf:about="http://10.80.223.28:8081/OSLC4JBugzilla/services/rootservices"
    xmlns:oslc_cm="http://open-services.net/xmlns/cm/1.0/"
    xmlns:dcterms="http://purl.org/dc/terms/"
    xmlns:jfs="http://jazz.net/xmlns/prod/jazz/jfs/1.0/"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

        <dcterms:title>OSLC-CM Adapter/Bugzilla Jazz Root Services</dcterms:title>
        <oslc_cm:cmServiceProviders rdf:resource="http://10.80.223.28:8081/OSLC4JBugzilla/services/catalog/singleton" />
        <jfs:oauthRealmName>Bugzilla</jfs:oauthRealmName>
        <jfs:oauthDomain>http://10.80.223.28:8081/OSLC4JBugzilla</jfs:oauthDomain>
        <jfs:oauthRequestConsumerKeyUrl rdf:resource="http://10.80.223.28:8081/OSLC4JBugzilla/services/oauth/requestKey
        <jfs:oauthApprovalModuleUrl rdf:resource="http://10.80.223.28:8081/OSLC4JBugzilla/services/oauth/approveKey" />
        <jfs:oauthRequestTokenUrl rdf:resource="http://10.80.223.28:8081/OSLC4JBugzilla/services/oauth/requestToken"/>
        <jfs:oauthUserAuthorizationUrl rdf:resource="http://10.80.223.28:8081/OSLC4JBugzilla/services/oauth/authorize"
        <jfs:oauthAccessTokenUrl rdf:resource="http://10.80.223.28:8081/OSLC4JBugzilla/services/oauth/accessToken"/>
</ df: scriptio
```

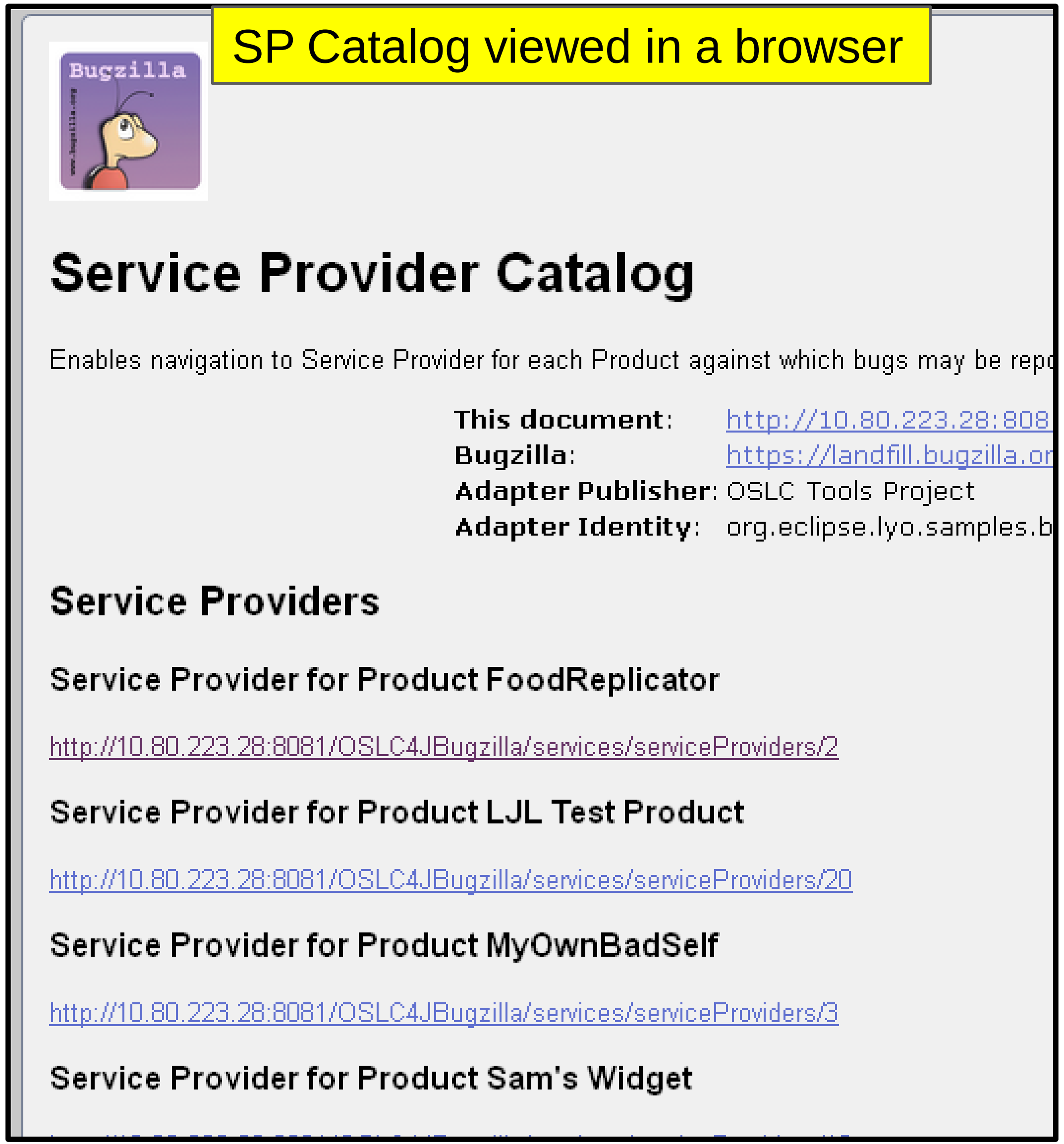
Key info::
- Lyo enables discovery capability
- Consumers don't have to guess Resources URIs

LAB 2: Retrieve service providers (Bugzilla products)


Java class Bugzilla Change request
Discover capability
Retrieve Service providers
Retrieve 5 first bugs
Automated bug creation
Java SWING Application (consume/post RDF document)
Update a bug

LAB2 : Discovery capability – Service provider and service list

- Retrieve list of service providers (SP)
  - SP mapped to Bugzilla product
  - Other applications may map SP to
    - Project area, systems or subsystem
- Actions
  - Consume Service provider catalog document
    - http://localhost:8080/OSLC4JBugzilla/services/catalog/singleton
  - List Services for Service provider 2.
- Outputs:
  - List of Service providers
  - List of Services for Service provider 2



SP Catalog viewed in a browser

Bugzilla

Service Provider Catalog

Enables navigation to Service Provider for each Product against which bugs may be repo

This document:	http://10.80.223.28:808
Bugzilla:	https://landfill.bugzilla.or
Adapter Publisher:	OSLC Tools Project
Adapter Identity:	org.eclipse.lyo.samples.b

Service Providers

Service Provider for Product FoodReplicator

http://10.80.223.28:8081/OSLC4JBugzilla/services/serviceProviders/2

Service Provider for Product LJL Test Product

http://10.80.223.28:8081/OSLC4JBugzilla/services/serviceProviders/20

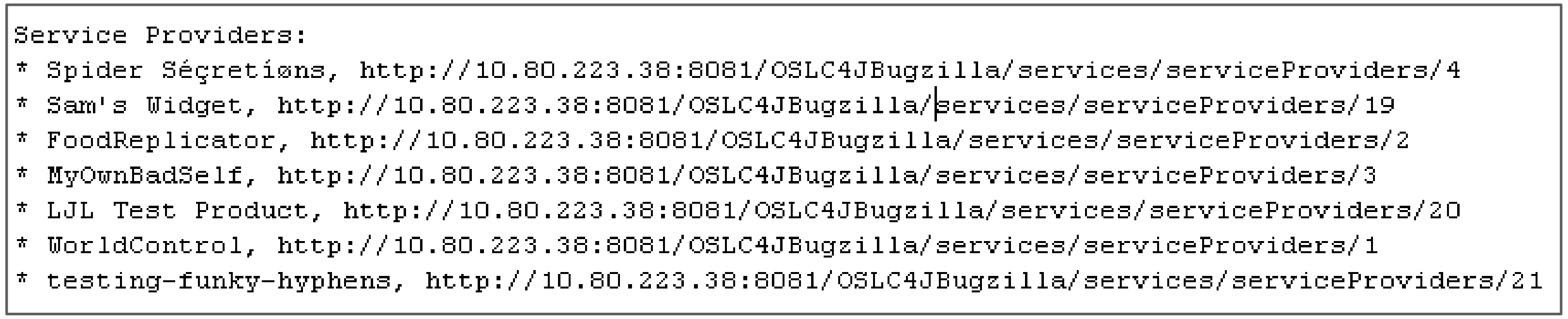
Service Provider for Product MyOwnBadSelf

http://10.80.223.28:8081/OSLC4JBugzilla/services/serviceProviders/3

Service Provider for Product Sam's Widget

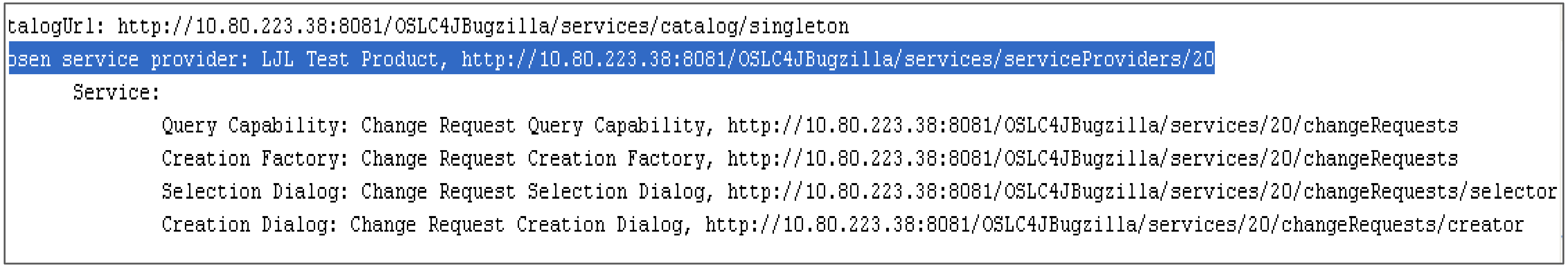
LAB2 : outputs

- Outputs:
  - List of Service providers



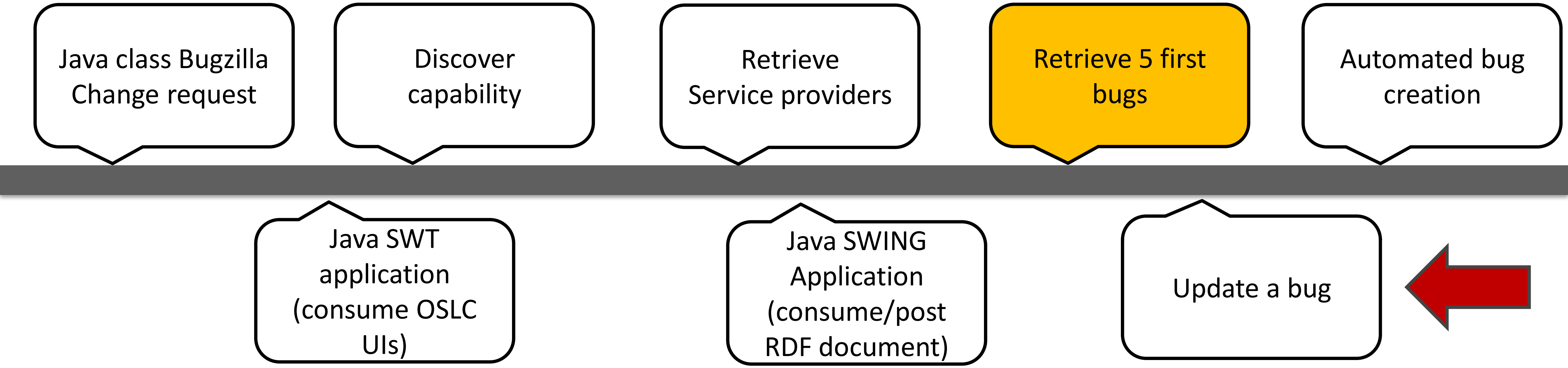
```
Service Providers:
* Spider Séçretíøns, http://10.80.223.38:8081/OSLC4JBugzilla/services/serviceProviders/4
* Sam's Widget, http://10.80.223.38:8081/OSLC4JBugzilla/services/serviceProviders/19
* FoodReplicator, http://10.80.223.38:8081/OSLC4JBugzilla/services/serviceProviders/2
* MyOwnBadSelf, http://10.80.223.38:8081/OSLC4JBugzilla/services/serviceProviders/3
* LJL Test Product, http://10.80.223.38:8081/OSLC4JBugzilla/services/serviceProviders/20
* WorldControl, http://10.80.223.38:8081/OSLC4JBugzilla/services/serviceProviders/1
* testing-funky-hyphens, http://10.80.223.38:8081/OSLC4JBugzilla/services/serviceProviders/21
```

  - List of Services for Service provider 2



```
talogUrl: http://10.80.223.38:8081/OSLC4JBugzilla/services/catalog/singleton
osen service provider: LJL Test Product, http://10.80.223.38:8081/OSLC4JBugzilla/services/serviceProviders/20
        Service:
                Query Capability: Change Request Query Capability, http://10.80.223.38:8081/OSLC4JBugzilla/services/20/changeRequests
                Creation Factory: Change Request Creation Factory, http://10.80.223.38:8081/OSLC4JBugzilla/services/20/changeRequests
                Selection Dialog: Change Request Selection Dialog, http://10.80.223.38:8081/OSLC4JBugzilla/services/20/changeRequests/selector
                Creation Dialog: Change Request Creation Dialog, http://10.80.223.38:8081/OSLC4JBugzilla/services/20/changeRequests/creator
```

LAB 3: Consume the services


Java class Bugzilla Change request
Discover capability
Retrieve Service providers
Retrieve 5 first bugs
Automated bug creation
Java SWT application (consume OSLC UIs)
Java SWING Application (consume/post RDF document)
Update a bug

LAB 3 : query capability

Find out more at http://open-services.net/bin/view/Main/OSLCCoreSpecQuery

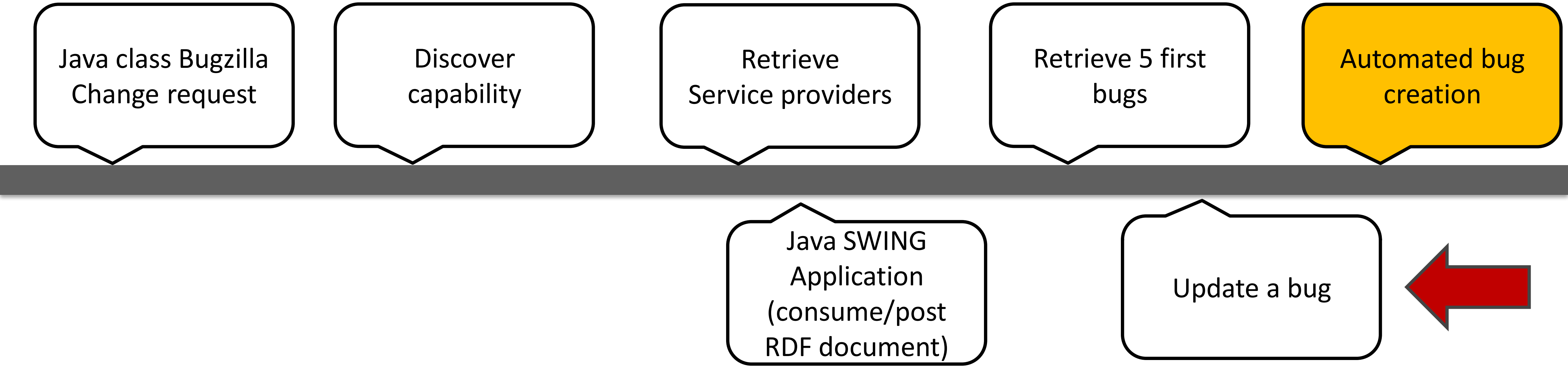
- Consume services available
  - Query capability
    - OSLC query properties:
      - OSLC.where, oslc.paging, oslc.pagesize, oslc.prefix, …

- Actions:
  - Use query properties to retrieve bugs from Bugzilla
    - queryCapability.getQueryBase().toString() + "?oslc.paging=true&oslc.pageSize=3";
    - queryCapability.getQueryBase().toString() + "?oslc.where=dcterms:identifier=" + resourceId;

- Output:
  - RDF document of bugs that match the criteria

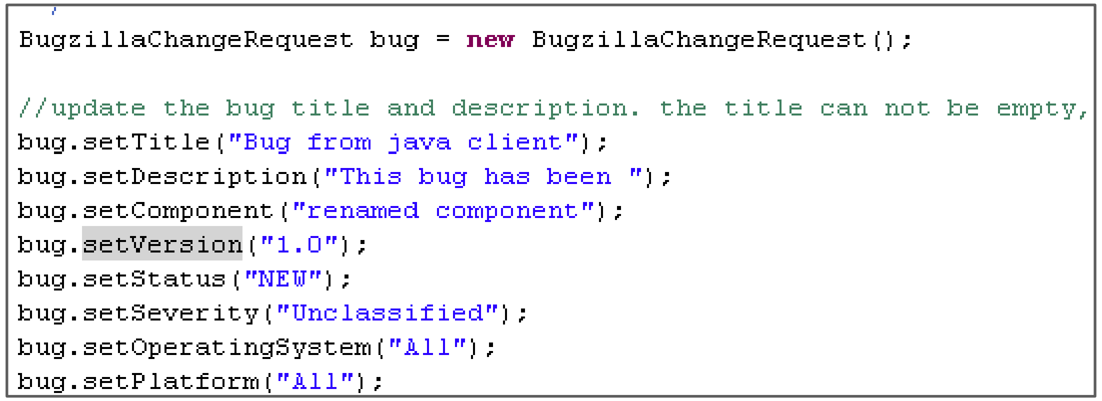
Note:
Assign “resourceId “ to a bug id in the selected Bugzilla product

LAB 3: creation factory


Java class Bugzilla Change request
Discover capability
Retrieve Service providers
Retrieve 5 first bugs
Automated bug creation
Java SWING Application (consume/post RDF document)
Update a bug

LAB 3 : Creation factory

- Consume services available
  - Creation factory
    - Post raw RDF resources to the service provider URL
- Actions
  - Identify the service URL
  - Create a bug



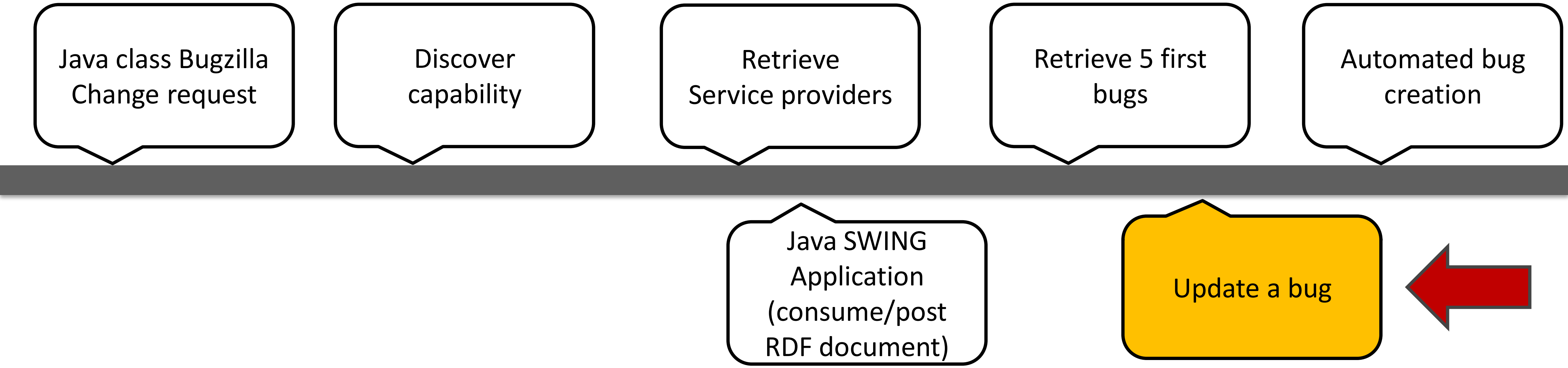
```
BugzillaChangeRequest bug = new BugzillaChangeRequest();

//update the bug title and description. the title can not be empty,
bug.setTitle("Bug from java client");
bug.setDescription("This bug has been ");
bug.setComponent("renamed component");
bug.setVersion("1.0");
bug.setStatus("NEW");
bug.setSeverity("Unclassified");
bug.setOperatingSystem("All");
bug.setPlatform("All");
```

- Output
  - Display the bug Id
  - See the bug on Bugzilla website

Note:
Make sure the property Component matches with a value in the product selected.

Key message:
OSLC4J handles automatically Serialisation from Java to RDF

LAB 3: Update a bug info


Java class Bugzilla Change request
Discover capability
Retrieve Service providers
Retrieve 5 first bugs
Automated bug creation
Java SWING Application (consume/post RDF document)
Update a bug

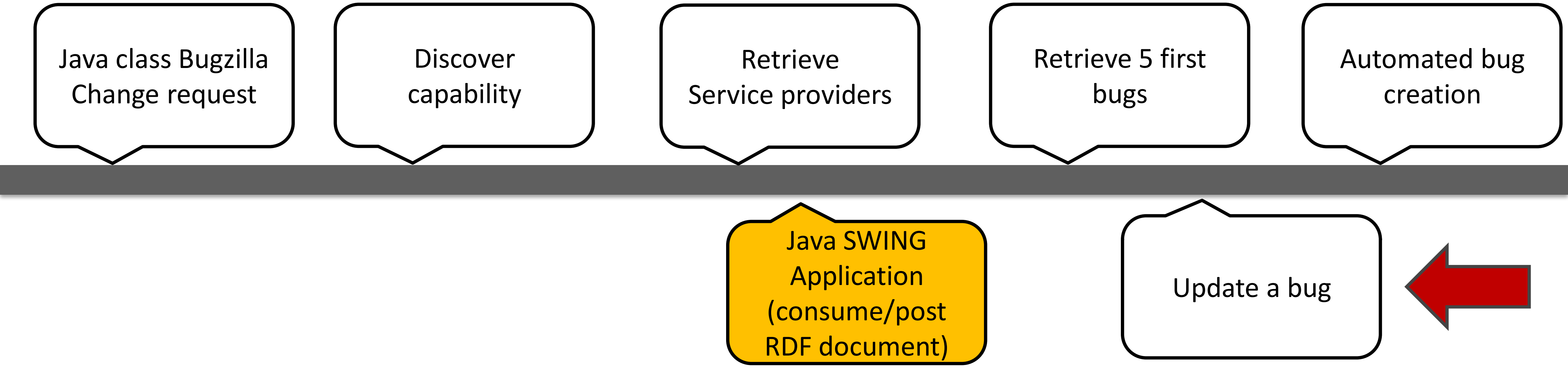
LAB 3 : Update a bug info

- Objective
  - Update an existing bug
- Actions
  - Use OSLCRestClient class to retrieve a bug
  - Add a new comment
  - Send an Update request to the service provider
- Output
  - Check the bug on the bugzilla website
  - A new comment has been added on Internet

Limitations:
OSLC4JBugzilla does not provide the list of comments in the Bug structure

Key Message:
At the server side, there is a programmatic lock system to prevent resource override.
Check header Etag

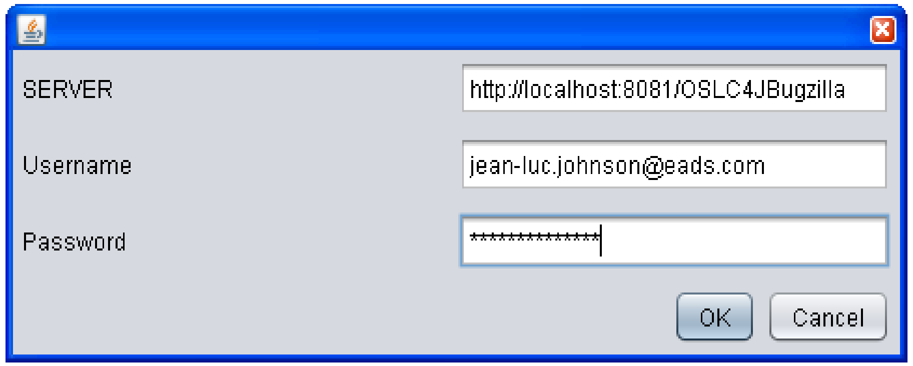
Final Demo : Java Swing GUI consumer of Bugzilla bugs


Java class Bugzilla Change request
Discover capability
Retrieve Service providers
Retrieve 5 first bugs
Automated bug creation
Java SWING Application (consume/post RDF document)
Update a bug

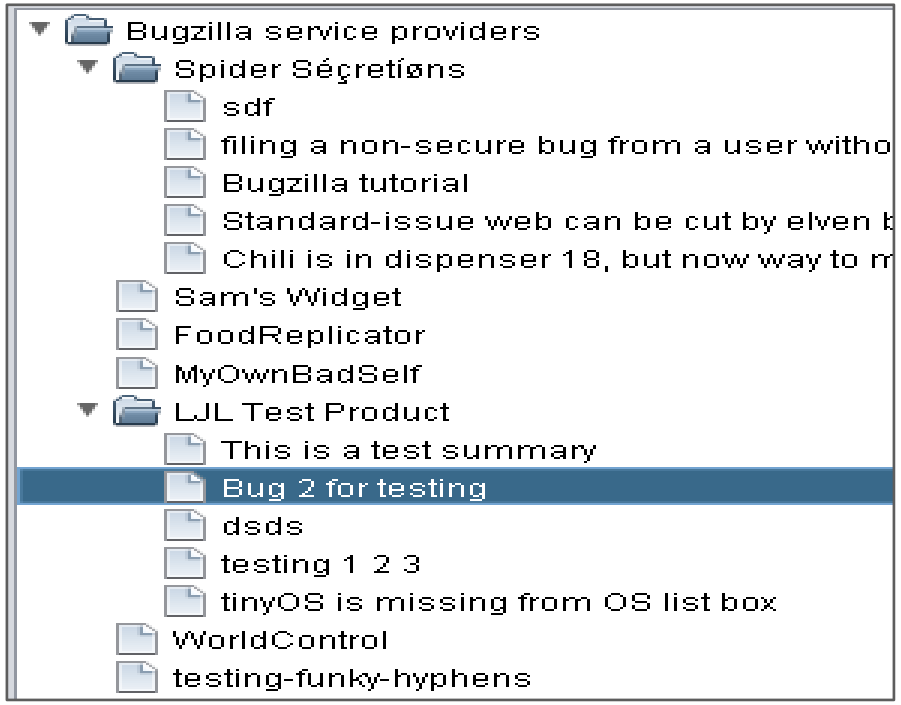
Client Swing demo based on the OSLC Eclipse Lyo SDK

- Objective
  - List products/List bugs
  - Post a new bug/Update a bug

- Actions
  - Configure the GUI with your credentials
    - Click Menu Item File > Options
  - Click on the root node in navigation panel
    - Expand a product node
    - Select a bug to display RDF document
  - Post a new bug
    - Click Menu Item Creation > New bug
    - Click Menu Item Update > Bug update



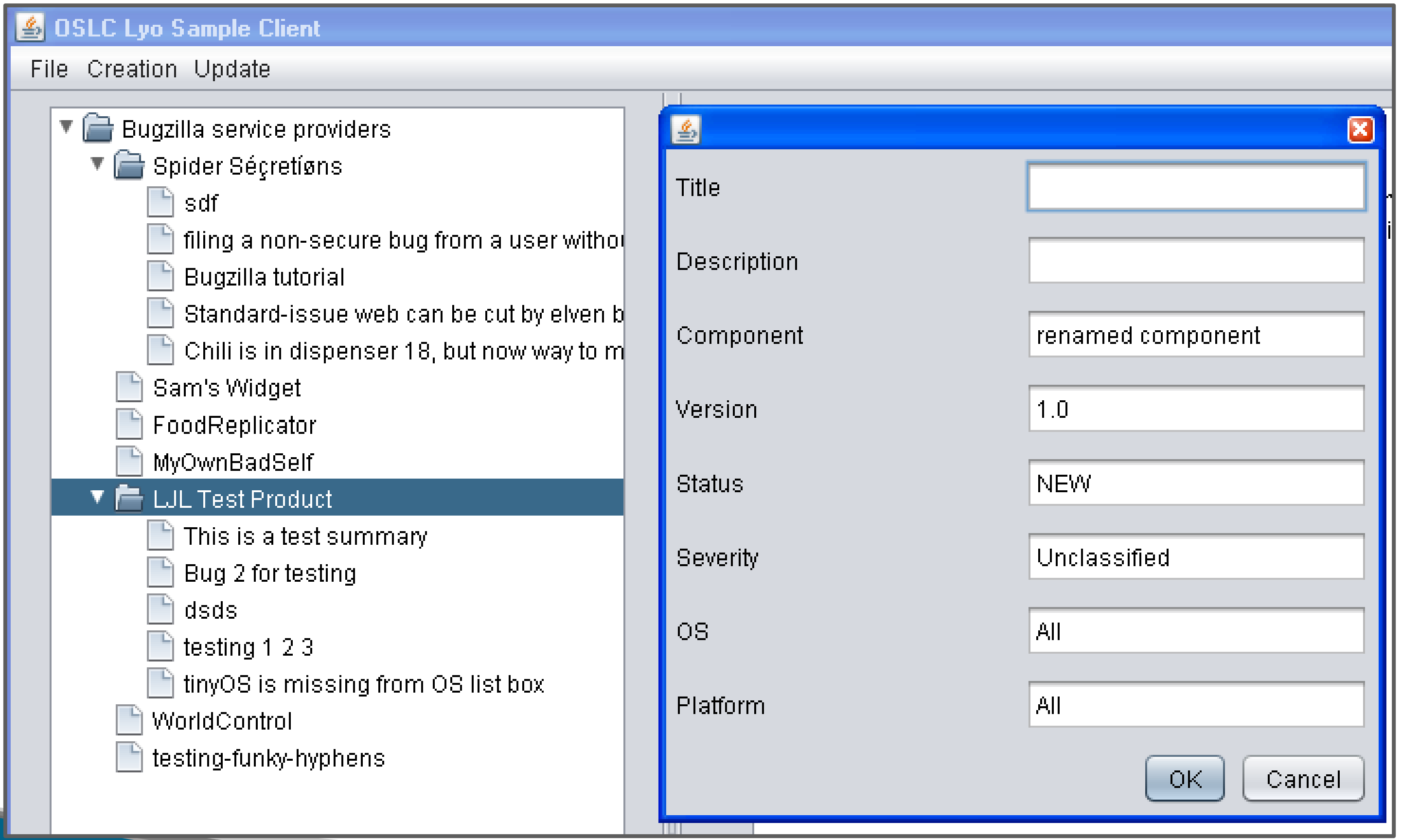
SERVER	http://localhost:8081/OSLC4JBugzilla
Username	jean-luc.johnson@eads.com
Password	**************



- Bugzilla service providers
  - Spider Sécretíøns
    - sdf
    - filing a non-secure bug from a user witho
    - Bugzilla tutorial
    - Standard-issue web can be cut by elven b
    - Chili is in dispenser 18, but now way to m
  - Sam's Widget
  - FoodReplicator
  - MyOwnBadSelf
  - LJL Test Product
    - This is a test summary
    - Bug 2 for testing
    - dsds
    - testing 1 2 3
    - tinyOS is missing from OS list box
  - WorldControl
  - testing-funky-hyphens

Client Swing demo based on the OSLC Eclipse Lyo SDK

- Actions
  - Post a new bug
    - Click Menu Item Creation > New bug

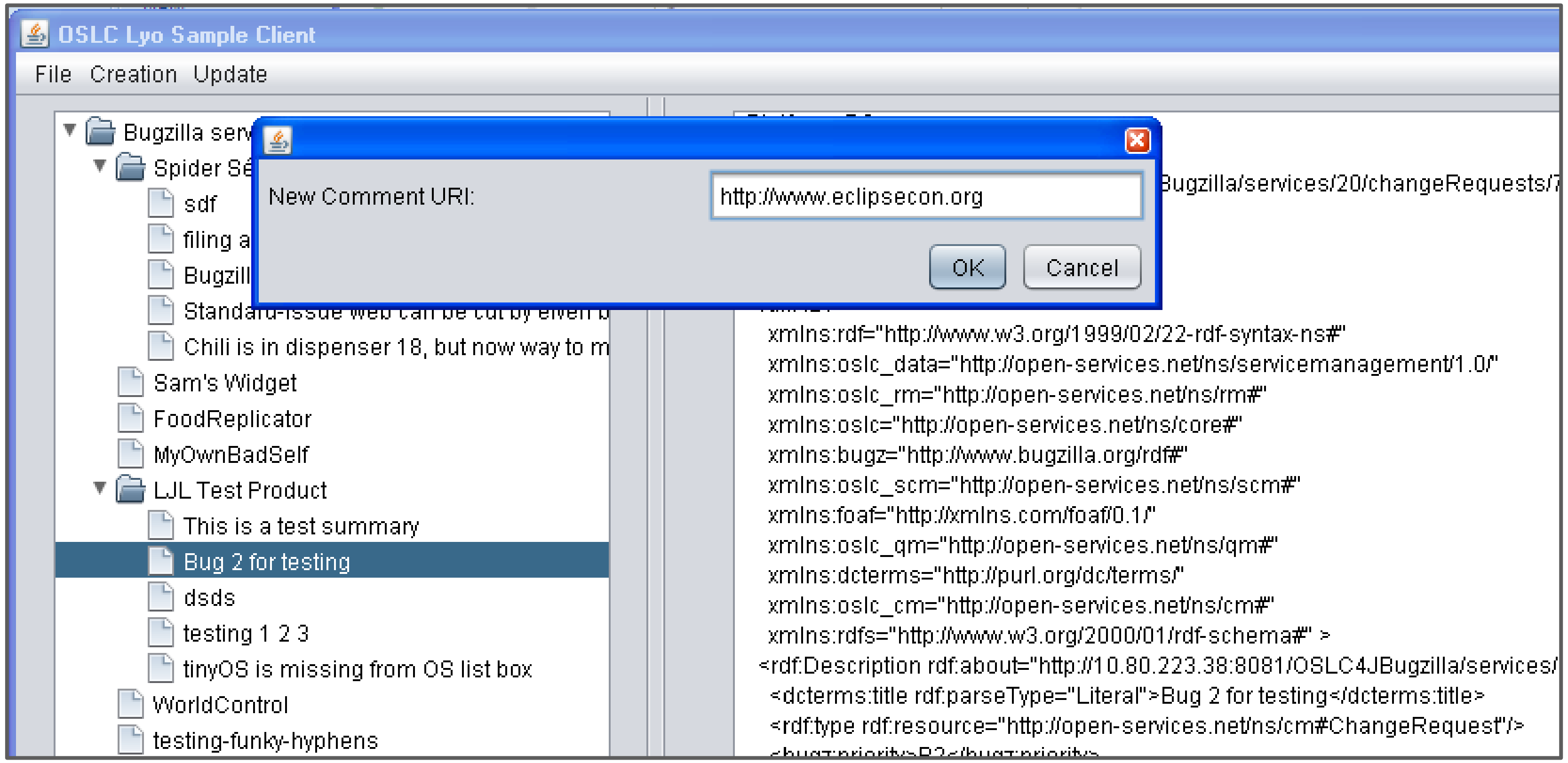


- Bugzilla service providers
  - Spider Sęçretiøns
    - sdf
    - filing a non-secure bug from a user witho
    - Bugzilla tutorial
    - Standard-issue web can be cut by elven b
    - Chili is in dispenser 18, but now way to m
  - Sam's Widget
  - FoodReplicator
  - MyOwnBadSelf
  - LJL Test Product
    - This is a test summary
    - Bug 2 for testing
    - dsds
    - testing 1 2 3
    - tinyOS is missing from OS list box
  - WorldControl
  - testing-funky-hyphens

Title	
Description	
Component	renamed component
Version	1.0
Status	NEW
Severity	Unclassified
OS	All
Platform	All

Client Swing demo based on the OSLC Eclipse Lyo SDK

- Actions
  - Update the selected bug
    - Click Menu Item Update > Bug update



OSLC Lyo Sample Client

File Creation Update

- Bugzilla serv
  - Spider Se
    - sdf
    - filing a
    - Bugzill
    - Standard-issue web can be cut by elven b
    - Chili is in dispenser 18, but now way to m
  - Sam's Widget
  - FoodReplicator
  - MyOwnBadSelf
  - LJL Test Product
    - This is a test summary
    - Bug 2 for testing
    - dsds
    - testing 1 2 3
    - tinyOS is missing from OS list box
  - WorldControl
  - testing-funky-hyphens

New Comment URI: http://www.eclipsecon.org

OK Cancel

```
Bugzilla/services/20/changeRequests/7
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:oslc_data="http://open-services.net/ns/servicemanagement/1.0/"
 xmlns:oslc_rm="http://open-services.net/ns/rm#"
 xmlns:oslc="http://open-services.net/ns/core#"
 xmlns:bugz="http://www.bugzilla.org/rdf#"
 xmlns:oslc_scm="http://open-services.net/ns/scm#"
 xmlns:foaf="http://xmlns.com/foaf/0.1/"
 xmlns:oslc_qm="http://open-services.net/ns/qm#"
 xmlns:dcterms="http://purl.org/dc/terms/"
 xmlns:oslc_cm="http://open-services.net/ns/cm#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >
<rdf:Description rdf:about="http://10.80.223.38:8081/OSLC4JBugzilla/services/
 <dcterms:title rdf:parseType="Literal">Bug 2 for testing</dcterms:title>
 <rdf:type rdf:resource="http://open-services.net/ns/cm#ChangeRequest"/>
 <bugz:priority>P2</bugz:priority>
```

Let’s recap

We described:
- the OSLC capabilities
- the discovery mechanism
- how to extend the ChangeRequest resource

We used OSLCRestClient:
-To process the rootservice
- to get the list of products
-To get the list of services
-To consume the services

We showed a java GUI:
-to navigate the products
- to display a bug
-To post a new bug
-To update a bug

You should be able now to build your own OSLC consumer application based on the Eclipse lyo project.

Questions ?

- Contacts
  - Jean-Luc Johnson (AGI): jean-luc.johnson@eads.com
  - Gray Bachelor (IBM): gray.bachelor@uk.ibm.com
  - Samit Mehta (IBM): samit.mehta@us.ibm.com
  - Harry Reeder (IBM)