



MaTraca v.3: Maintaining Traceability Links

Presenter: Angela Lozano
alozanor@vub.ac.be

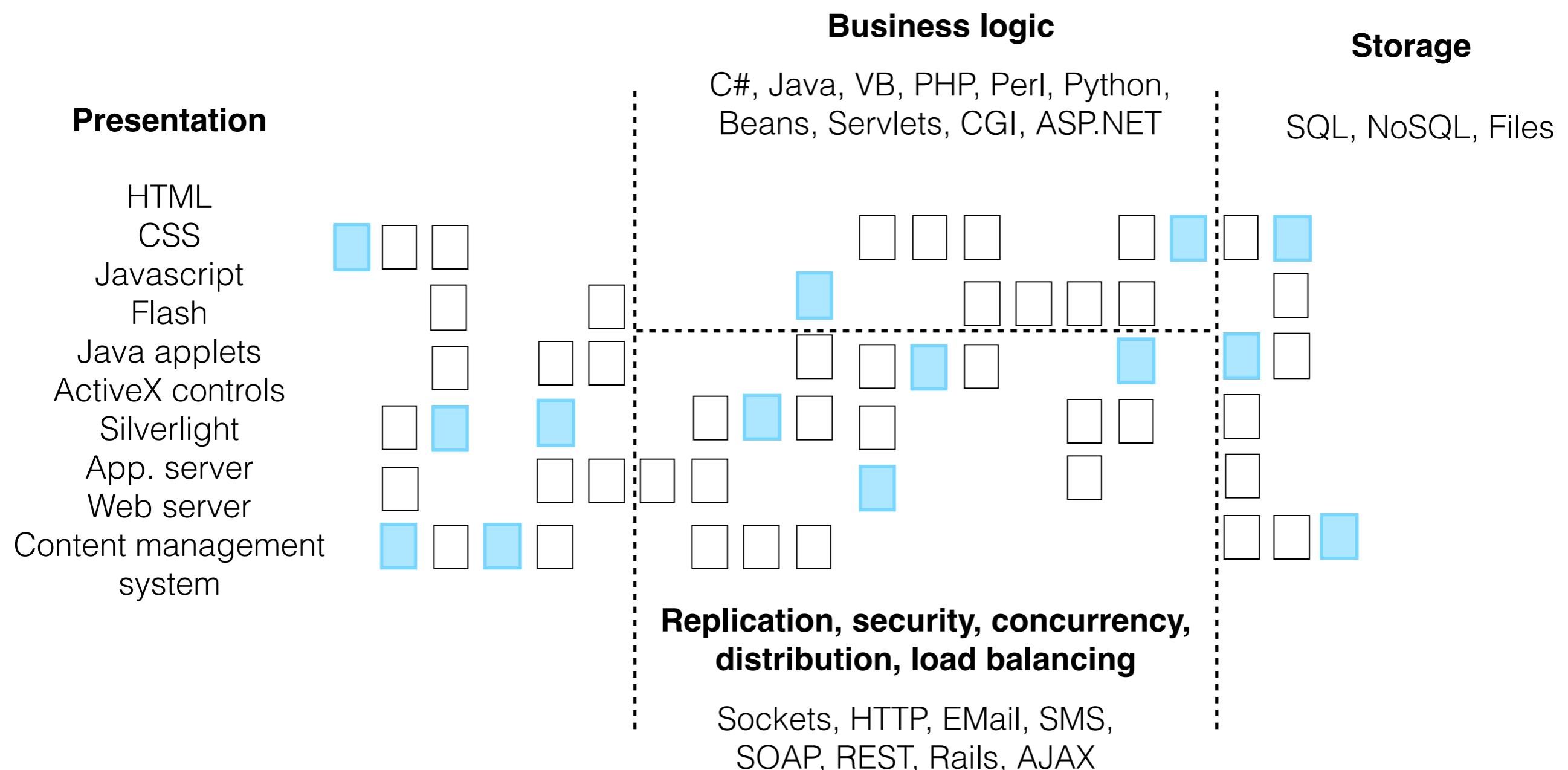
Managing Traceability Links With MaTraca

Angela Lozano, Carlos Noguera, and Viviane Jonckers

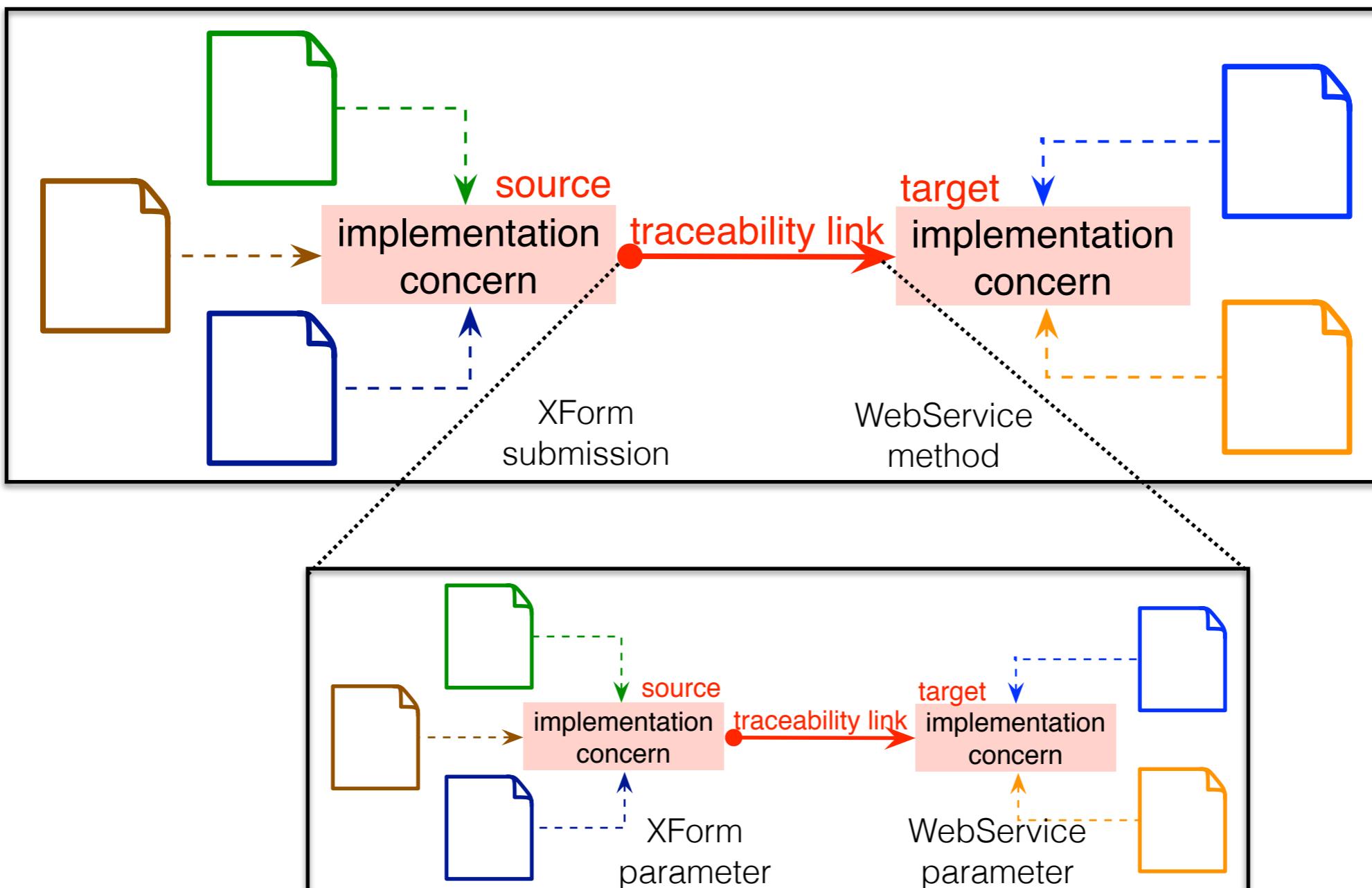
Tool track Proc. of IEEE Int'l International Conference on Software Analysis,
Evolution, and Reengineering SANER. 2016.



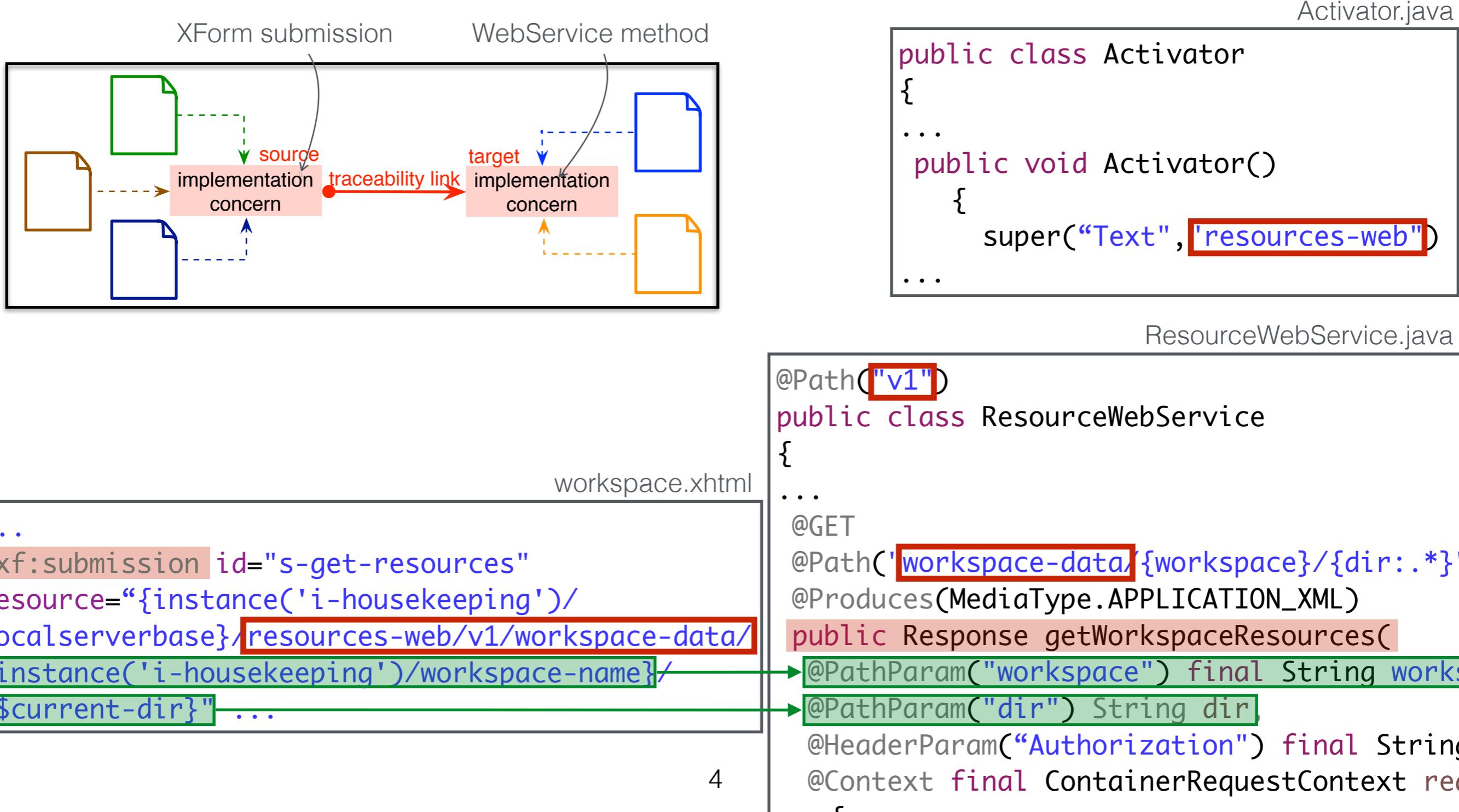
Horizontal Traceability



Defining traceability links



Defining traceability links

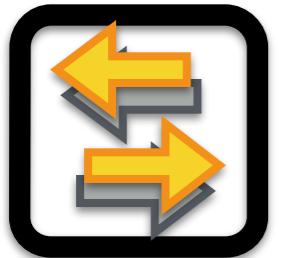




MaTraca's Features

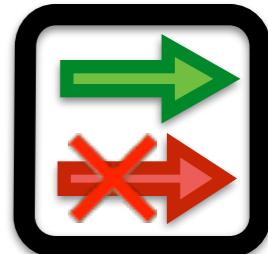


find all the entities on the source or target concerns



mark links / entities of interest

Source	Link status	Target
s-create-folder	-X->	
s-disconnect-repo	-X->	
s-delete-resource	-X->	com.id.webapp.resource.storage
s-delete-resource	-X->	com.id.webapp.resource.storage
s-get-resources	--->	com.id.webapp.resource.storage



evaluate compliance in marked links / entities



filter results

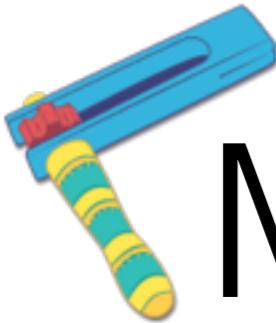
Source	Link status	Target
test.TesetErrorMessages.main(java.lang.String[]):void @ 16	---	MSG000001 @ /chaqTe
test.TesetErrorMessages.main(java.lang.String[]):void @ 18	---	MSG000001 @ /chaqTe
test.TesetErrorMessages.main(java.lang.String[]):void @ 12	-X->	MSG000001 @ /chaqTe

Source	Link status	Target
String	---	String
String	---	String
Integer	-X->	Long

Source	Link status	Target
String	---	String
String	---	String
Integer	-X->	Long

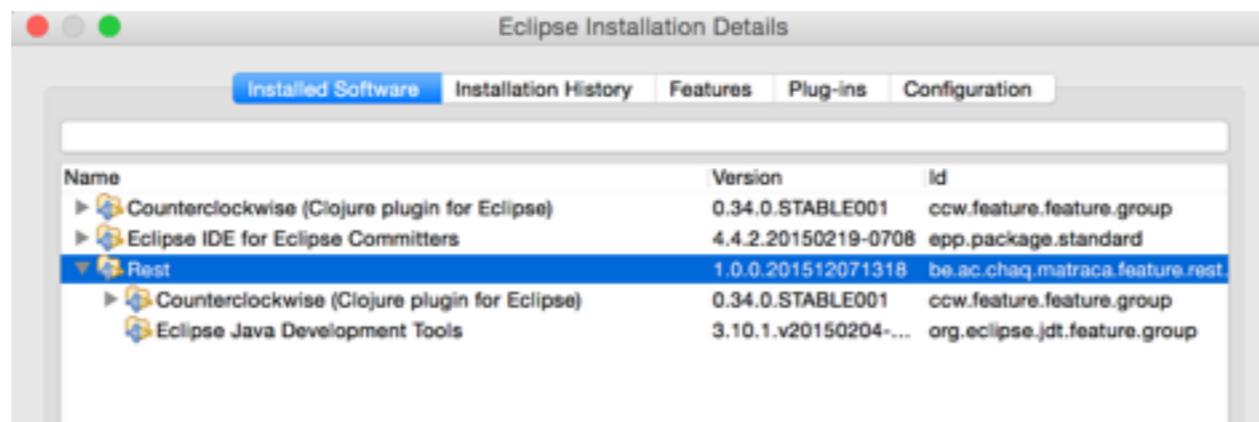


check matching of parameters

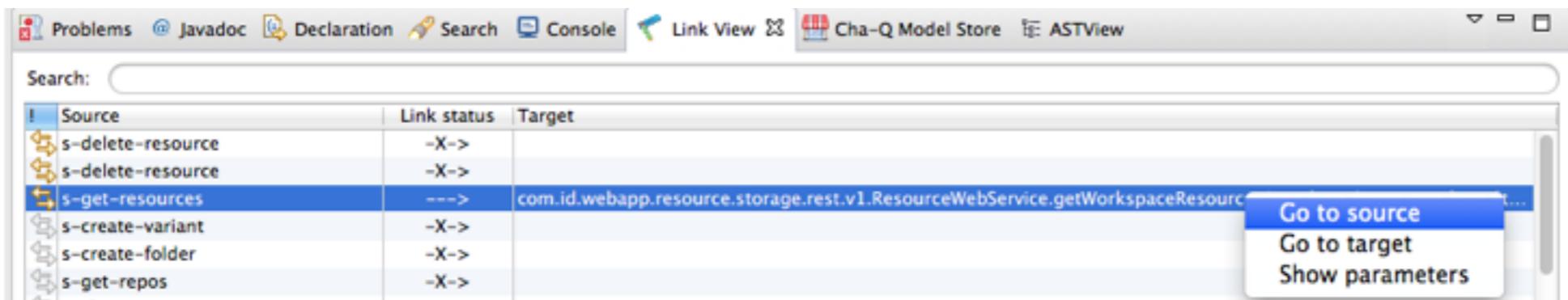


Matraca 3.0 (1/5)

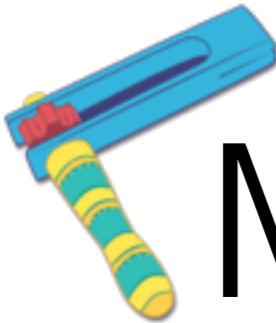
- Installer for eclipse (as an update site)



- Immediate jump to source



- BugFix: non-local code markers (i.e., not only files locally at the workspace are recognised but also git files)



Matraca 3.0 (2/5)

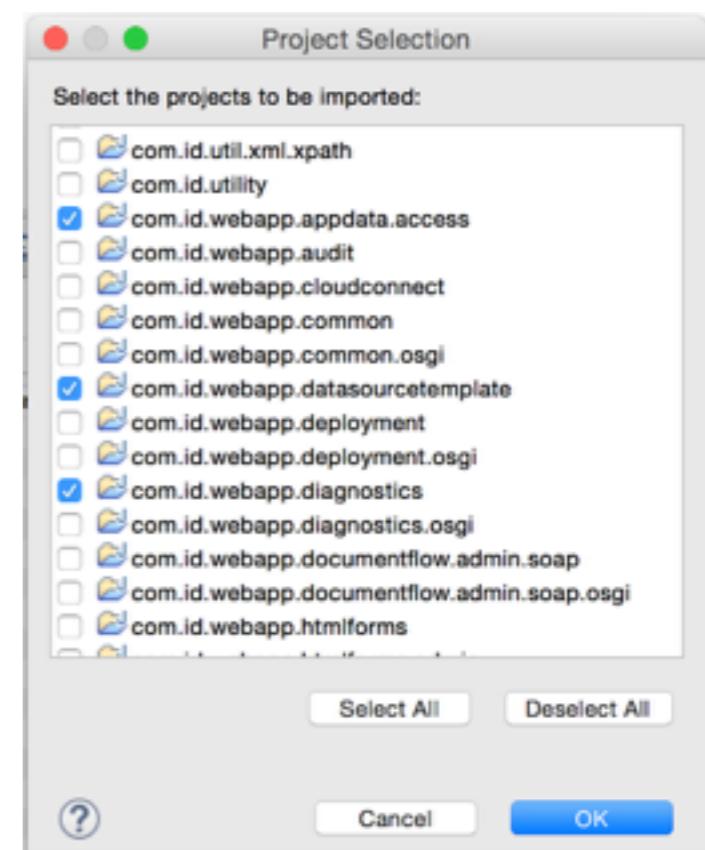
- Add location to the entity identified in the GUI

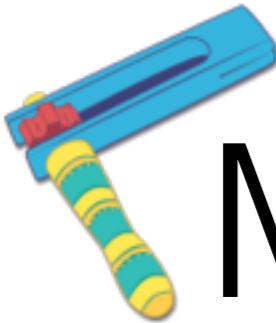
The image displays two side-by-side screenshots of the Matraca software interface. Both screenshots show a 'Link View' window with a table of links between entities. The left screenshot shows links for a 'com.id.webapp.resources' entity, while the right screenshot shows links for a 'changes.xhtml' entity. The tables have columns for Source, Link status, and Target.

Source	Link status	Target
s-delete-resource	-X->	
s-delete-resource	-X->	
s-get-resources	--->	com.id.webapp.resources
s-create-variant	-X->	
s-create-folder	-X->	
s-get-repos	-X->	

Source	Link status	Target
s-git-commit-push @ changes.xhtml	--->	gitCommitPush(java.lang.String,java.lang.String,java.lang.String)
s-git-use-ours @ changes.xhtml	--->	gitUseOursVersion(java.lang.String,java.lang.String,java.lang.String)
s-do-merge @ changes.xhtml	--->	getMergeURI(java.lang.String, javax.ws.rs.container.ContainerRequest)
s-git-use-theirs @ changes.xhtml	--->	gitUseTheirsVersion(java.lang.String,java.lang.String,java.lang.String)
s-revert @ changes.xhtml	--->	gitUseTheirsVersion(java.lang.String,java.lang.String,java.lang.String)
s-git-update @ changes.xhtml	--->	gitUpdateWorkspace(java.lang.String,java.lang.String,javax.ws.rs.core.UriBuilder)

- Save projects previously analyzed





Matraca 3.0 (3/5)

- Take into account annotations other than @Path to identify the matching method: GET, PUT, POST, DELETE, HEAD

```
@Path("v1")
public class ResourceWebService
{
    @GET
    @Produces(MediaType.APPLICATION_XML)
    public Response getActiveWorkspace(@HeaderParam("If-Match") final Set<String>
    {
        return Response.ok().build();
    }

    @POST
    @Path("get-compare-uri")
    @Produces(MediaType.APPLICATION_XML)
    public Response getCompareURI(@QueryParam("workspace") final String workspace)
    {
        return Response.ok().build();
    }
}
```

```
<xf:submission id="s-get-workspace-name" resource="{instance('i-
housekeeping')/localserverbase}/resources-web/v1/" method="get"...>
```

```
<xf:submission id="s-do-compare" resource="{$resource-service}/get-compare-uri?
workspace={instance('i-housekeeping')/workspace-name}"
... method="post" ... >
```

```
public class ResourceWebServiceApplicationActivator extends AbstractWebApplicationActivator
{

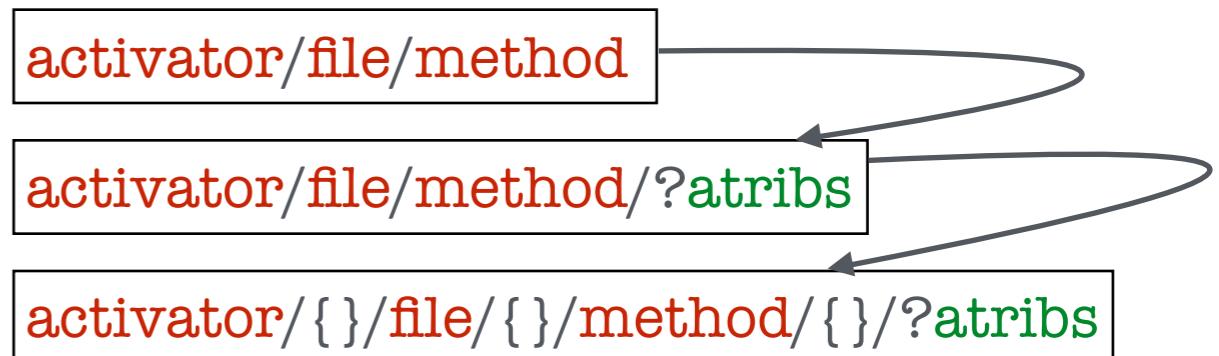
    public ResourceWebServiceApplicationActivator()
    {
        super("com.id.webapp.resource", "resources-web", "resources-web", true, "frontend");
    }
}
```

```
public class ResourceWebServiceApplicationActivator extends AbstractWebApplicationActivator
{
    static final String aux = "resources-web";
    public ResourceWebServiceApplicationActivator()
    {
        super("com.id.webapp.resource", aux, "resources-web", true, "frontend");
    }
}
```



Matraca 3.0 (4/5)

- Improve the matching algorithm so it can handle variables in source and target URLs



```
<xf:submission id="s-do-compare"
resource="{$resource-service}/get-compare-uri?
workspace={instance('i-housekeeping')/workspace-name}..."...
method="post" ... >
```

```
@Path("v1/{gitRepo}")
public class AnotherWebService
{
...
@GET
@Path("get-compare-uri")
public Response getWorkspaceResources(
@Context final ContainerRequestContext request)
{
...
}
```

```
workspace.xhtml
```

```
...
<xf:submission id="s-get-resources"
resource="{instance('i-housekeeping')/
localserverbase}/resources-web/v1/workspace-data/
{instance('i-housekeeping')/workspace-name}...
${$current-dir}..."
```

```
Activator.java
```

```
public class Activator
{
...
public void Activator()
{
super("Text", "resources-web")
...
}
```

```
ResourceWebService.java
```

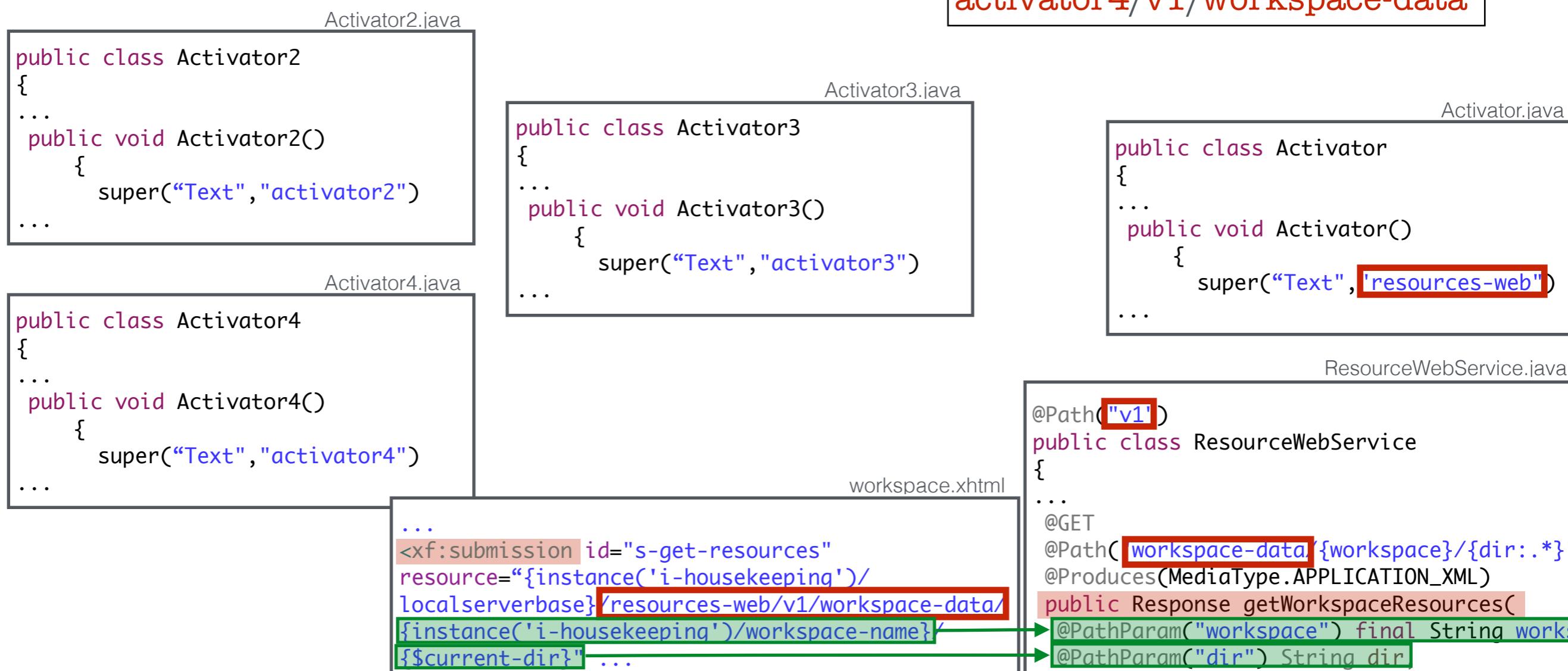
```
@Path("v1")
public class ResourceWebService
{
...
@GET
@Path("workspace-data/{workspace}/{dir:.*}")
@Produces(MediaType.APPLICATION_XML)
public Response getWorkspaceResources(
@PathParam("workspace") final String works...
```

```
@PathParam("dir") String dir
@HeaderParam("Authorization") final String
```



Matraca 3.0 (5/5)

- Verify that activator's data are only used to build links on the methods implemented in the same project



Does Matraca facilitate the maintenance of traceability links?

- 4 tasks: from the issue repository (total 4 hours)
- 4 developers: (equivalent expertise: d1-d2 & d3-d4)

	t1	t2	t3	t4
Baseline	dbd	dbd	dbd	dbd
without Matraca	d2+d4	d2+d4	d1+d3	d1+d3
with Matraca	d1+d3	d1+d3	d2+d4	d2+d4

To do list

VUB

Create new version of Matraca

Select tasks

Prepare manual of Matraca

Prepare evaluation questionnaires

Prepare correctness assessment points per task

Summarisation of results

Industrial partner

Arrange logistics to access data

Arrange logistics to prepare tasks

Prepare workspace/infrastructure

Trial tests with real data

Allow time for experiment:
familiarise with Matraca, perform
tasks and answer questions

Evaluation of results