

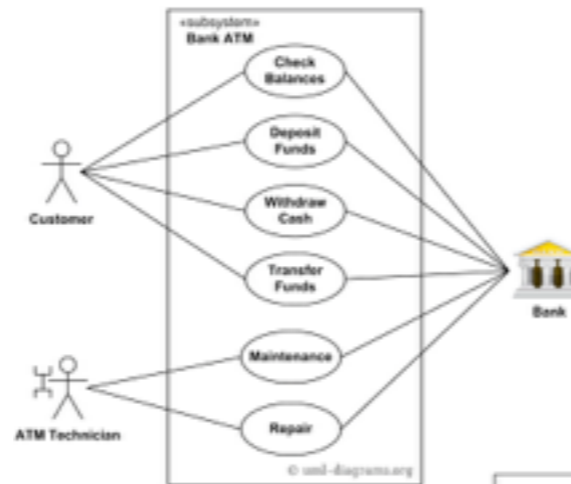
Outline

- Traceability
- Defining traceability links
- MaTraCa's prototype

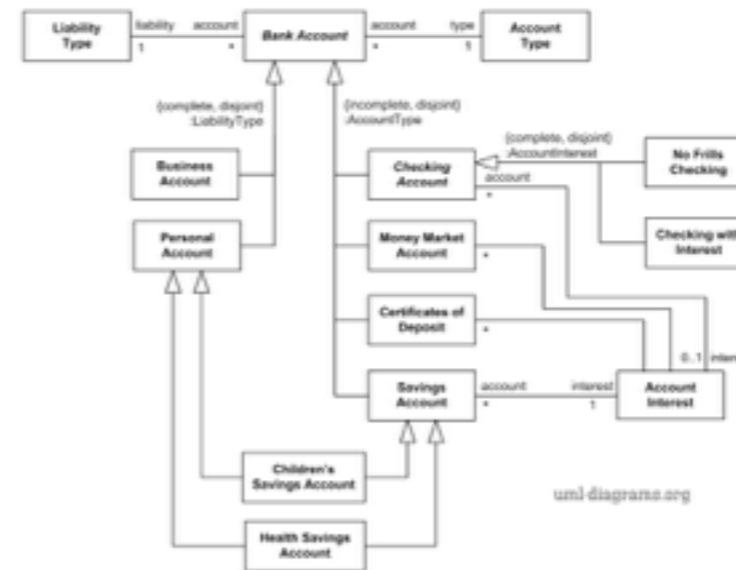
Traceability

Software Artefacts

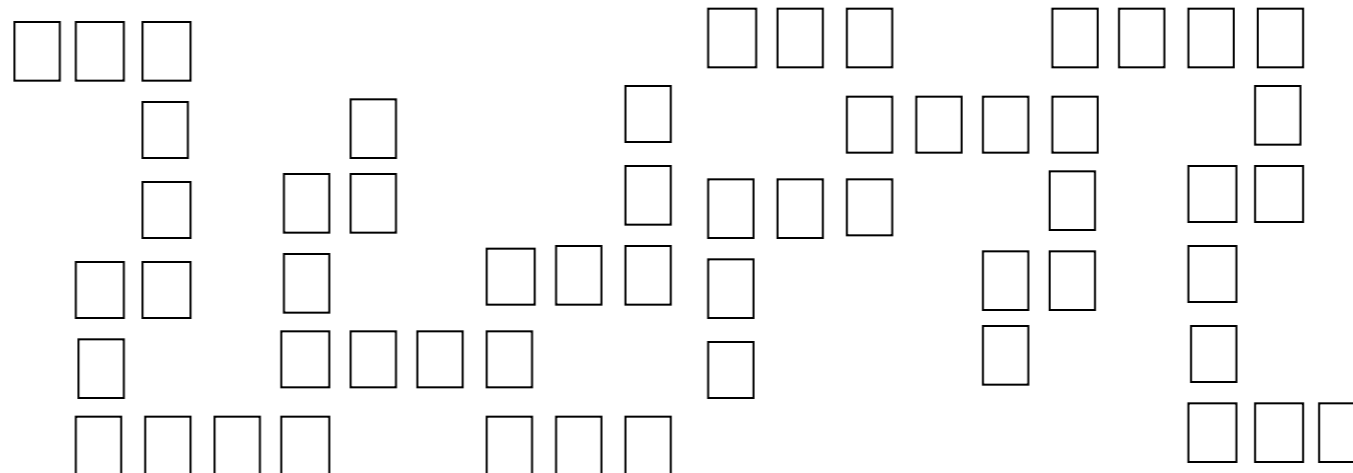
Requirements



Design



Code

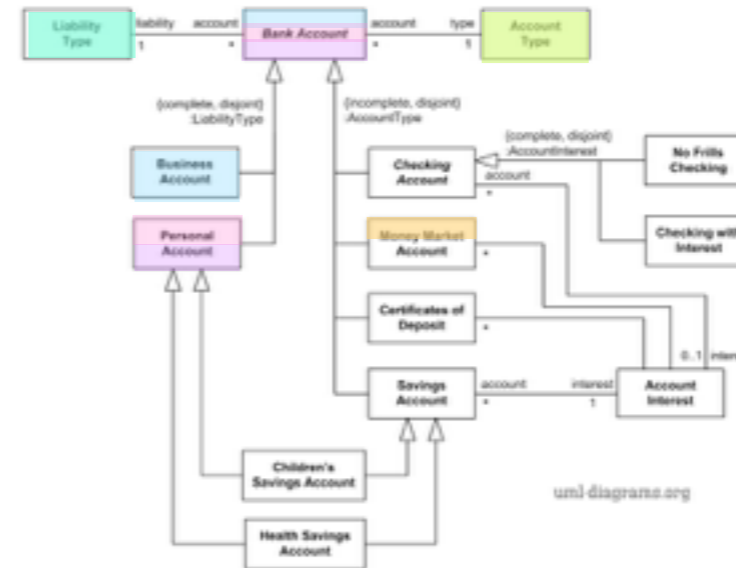


Vertical Traceability

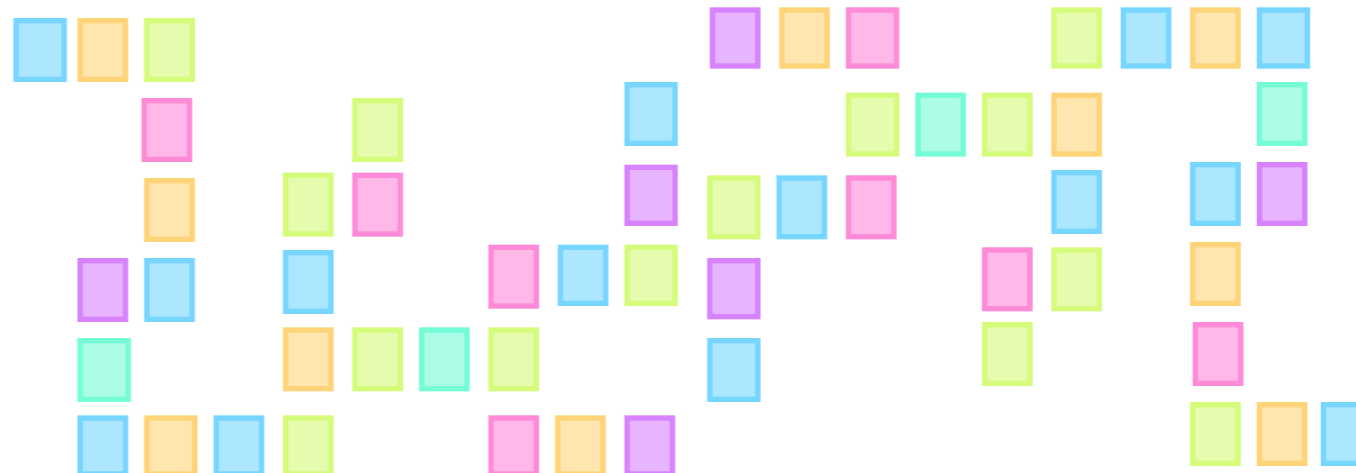
Requirements



Design

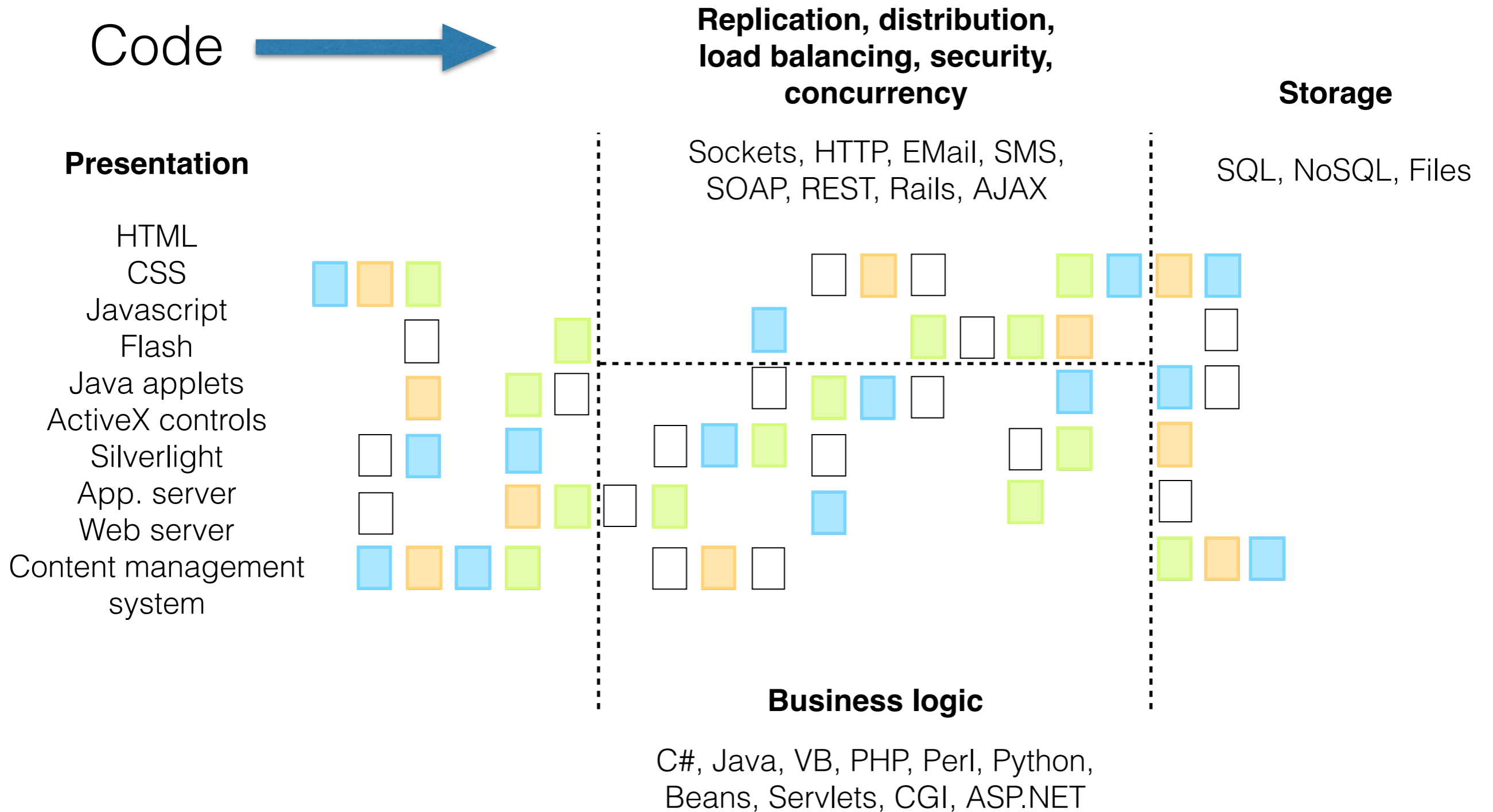


Code

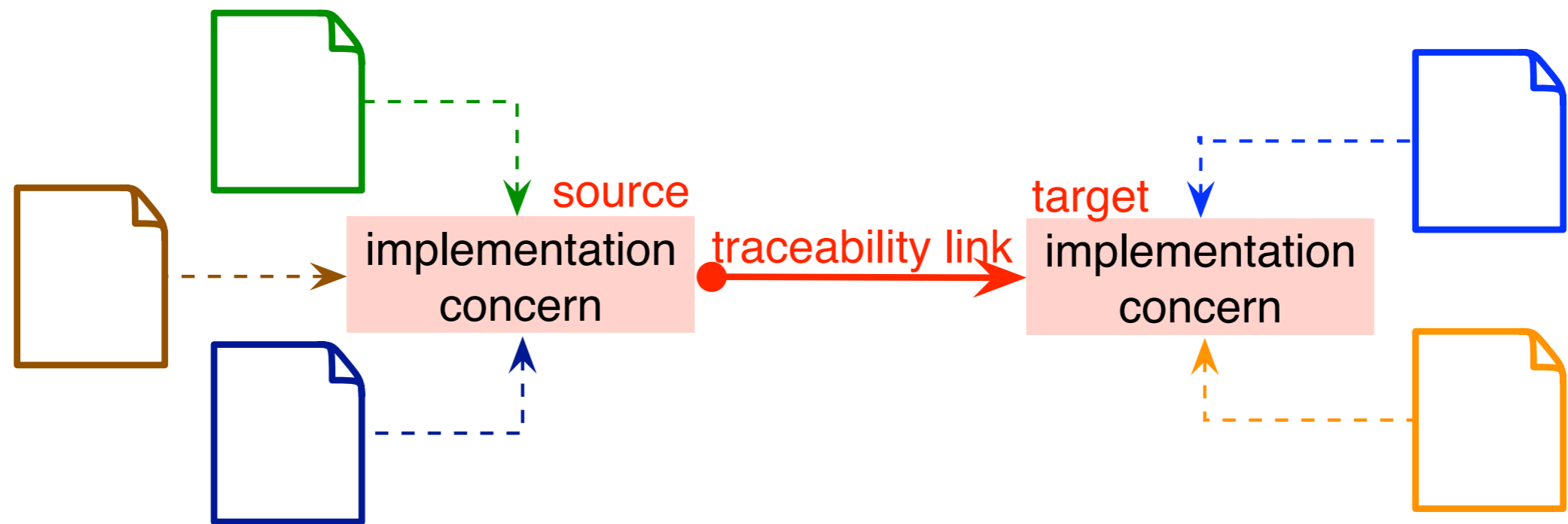


Horizontal Traceability

Code 

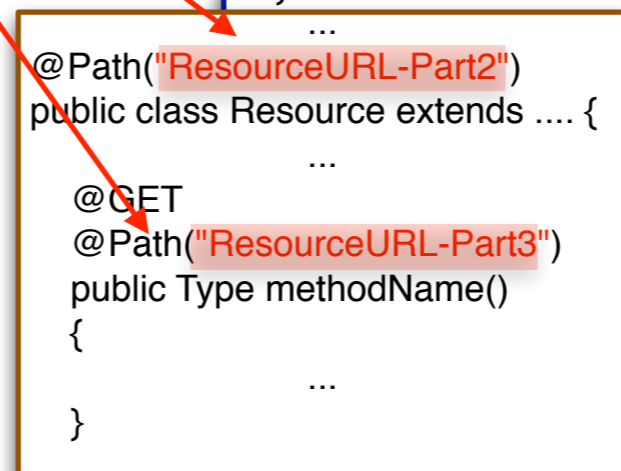
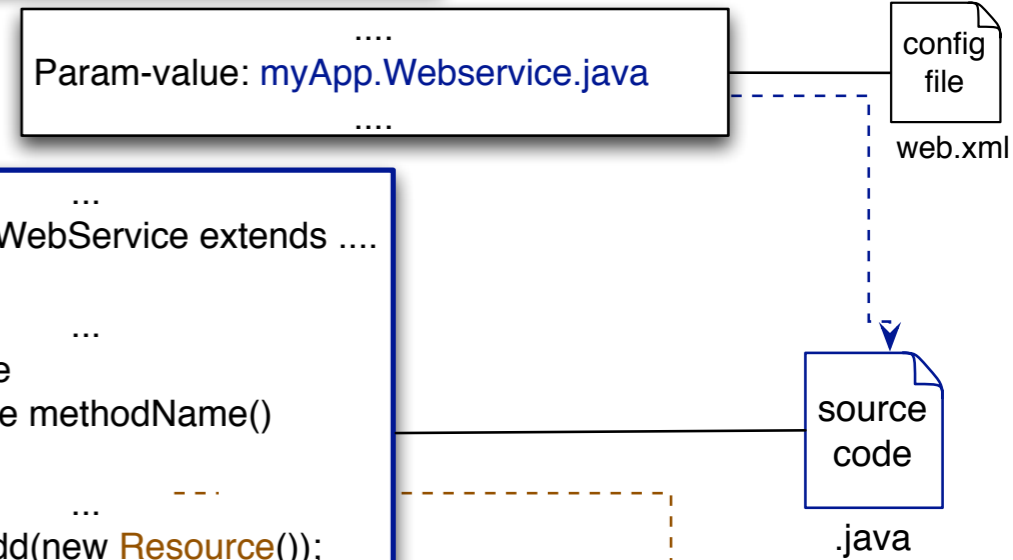
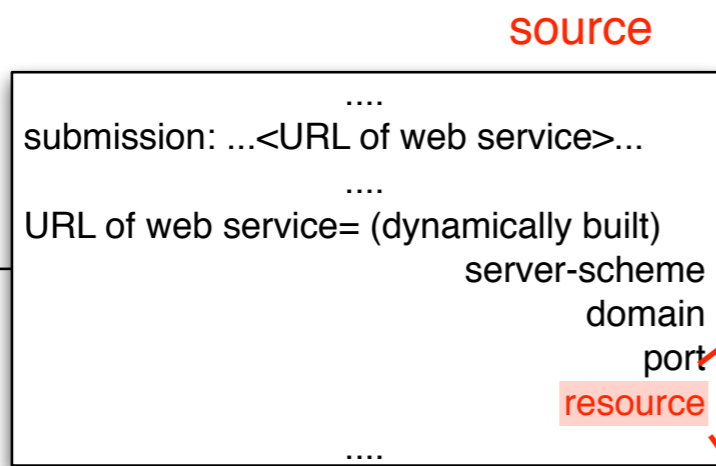
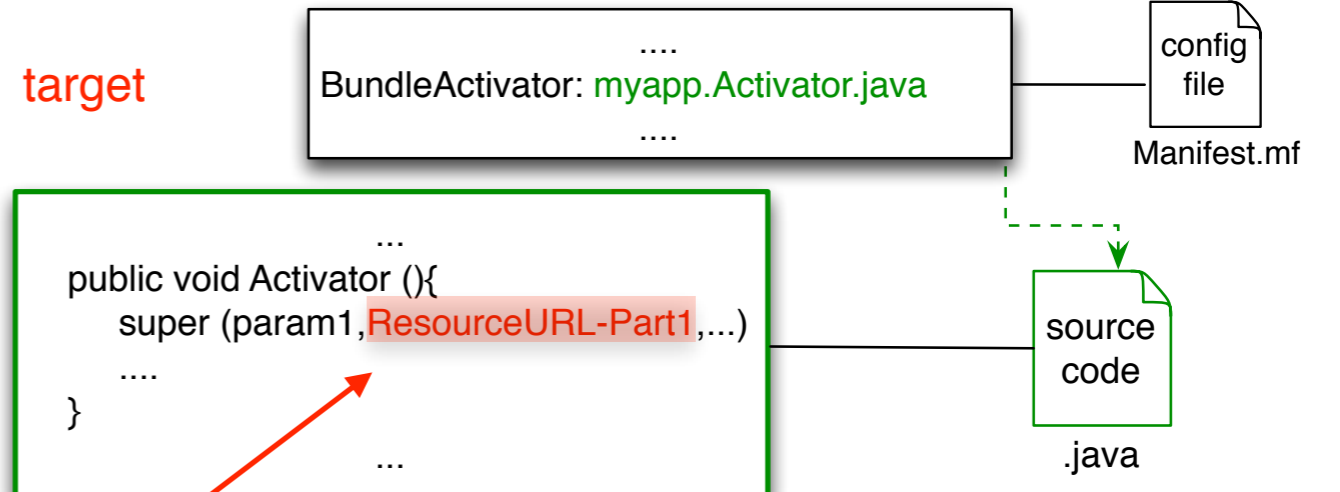
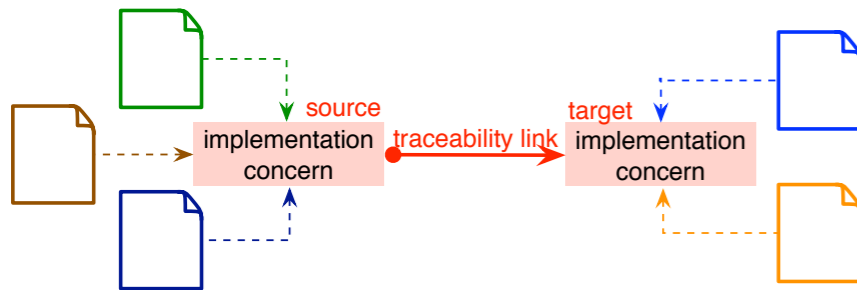


Defining traceability links



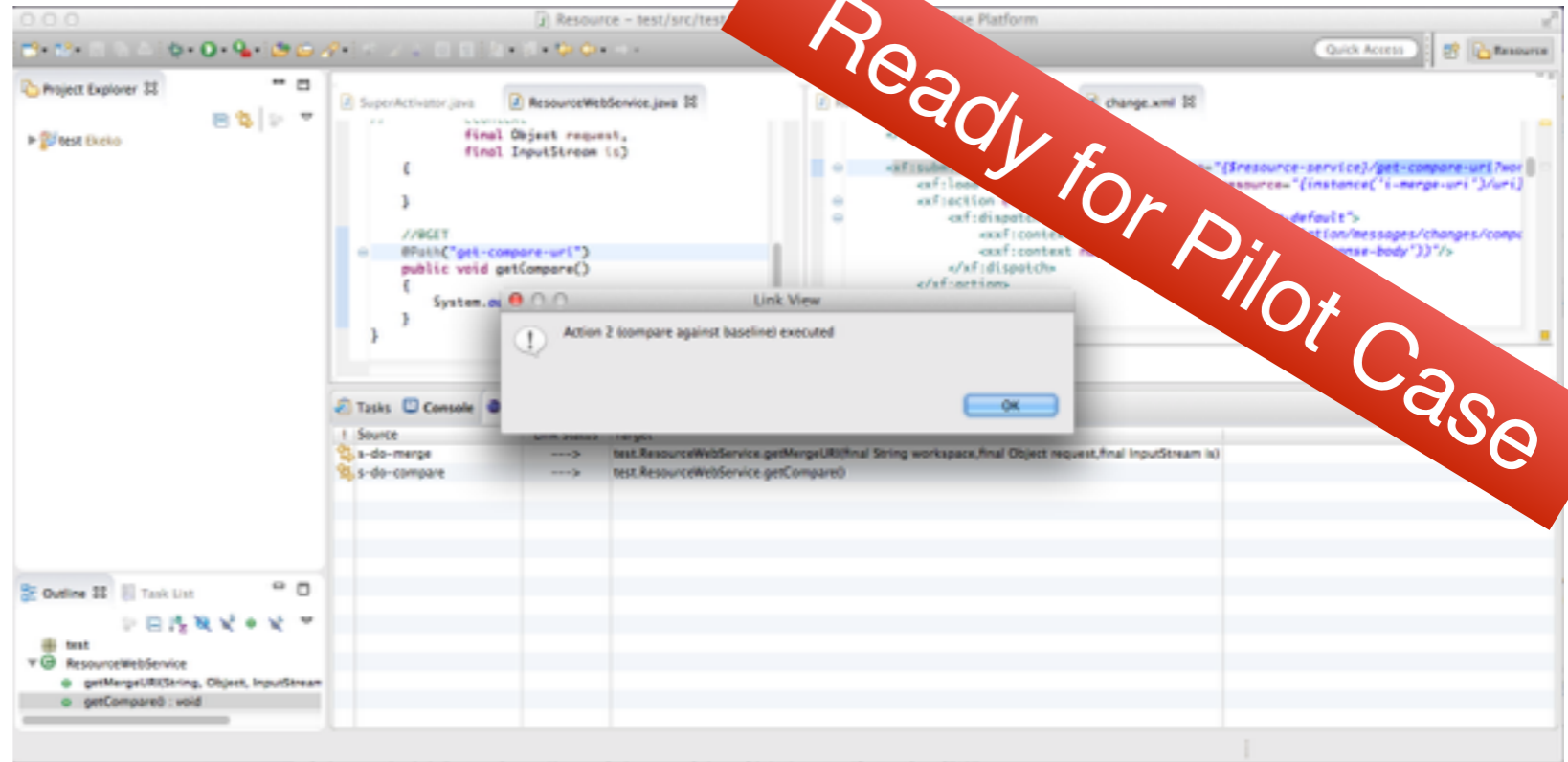
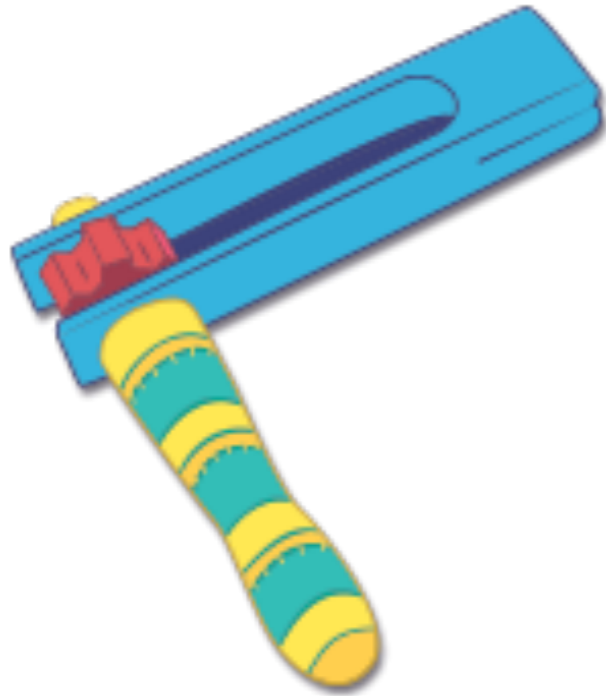
Example

simple name of the class
 Fully qualified name of the class
 the class with the package
 path application

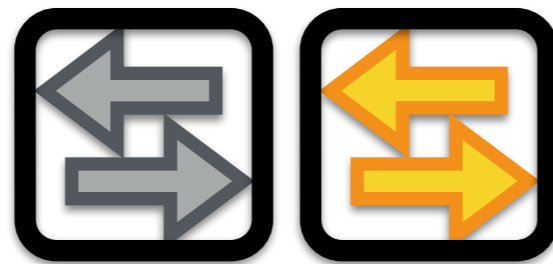


Prototype

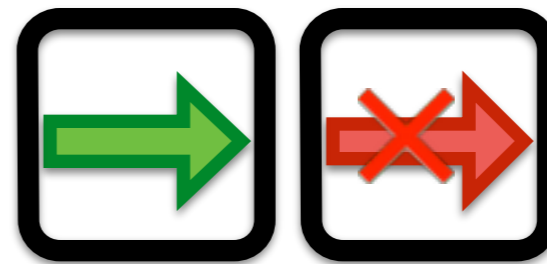
MaTraCa



find all the entities on the source or target concerns



mark links / entities of interest



evaluate compliance to trace in marked links / entities



propose other types of entities to trace

MaTraCa

Resource - test/change.xml - Eclipse Platform

Project Explorer: test Ekeko

Console | Link View

!	Source	Link status	Target
↕	s-do-merge	--->	test.ResourceWebService.getMergeURI(final String work...
↕	s-get-repos	-X->	
↕	s-get-workspaces	-X->	
↕	s-revert	-X->	
↕	s-git-commit-push	-X->	
↕	s-git-use-theirs	-X->	
↕	s-abort-merge	-X->	
↕	s-git-update	-X->	
↕	s-git-use-ours	-X->	
↕	s-do-compare	-X->	
↕	s-get-workspace-name	-X->	test.ResourceWebService.getCompare()
		-X->	test.ResourceWebService.getCompare()

Outline: Tas

Code Editor: xhtml:html/xhtml:head/xf:model/xf:submission/resource

Status Bar: Writable | Smart Insert | 140 : 91 | [4805-4820]

MaTraCa

ResourceWebService.java

```
package test;

import java.io.InputStream;

@Path("v1")
public class ResourceWebService {
    @Path("s-do-compare")
    public void getCompare()
    {
        System.out.println("dummy test r
    }

    @Path("get-merge-uri")
    public void getMergeURI(
        final String workspace,
        final Object request,
        final InputStream is)
    {
    }
}
```

ResourceWebServiceApplication.java

change.xml

```
<xf:dispatch name="e-refresh" target="m-default"/>
</xf:action>
</xf:submission>

<xf:submission id="s-do-compare" resource="{ $resource-service }/get-compare-uri?w
<xf:load ev:event="xforms-submit-done" resource="{instance('i-merge-uri')/ur
<xf:action ev:event="xforms-submit-error">
    <xf:dispatch name="e-add-message" target="m-default">
        <xxf:context name="message" value="$translation/messages/changes/com
        <xxf:context name="params" value="(event('response-body'))"/>
    </xf:dispatch>
</xf:action>
</xf:submission>

<xf:submission id="s-get-repos" resource="{ $resource-service }/repositories-overv
<xf:action ev:event="xforms-submit-done">
    <xf:setvalue ref="instance('i-housekeeping')/current-repository-id" valu
</xf:action>
```

Console Link View

!	Source	Link status	Target
↕	s-do-merge	--->	test.ResourceWebService.getMergeURI(final String work...
↕	s-get-repos	-X->	
↕	s-get-workspaces	-X->	
↕	s-revert	-X->	
↕	s-git-commit-push	-X->	
↕	s-git-use-theirs	-X->	
↕	s-abort-merge	-X->	
↕	s-git-update	-X->	
↕	s-git-use-ours	-X->	
↕	s-do-compare	-X->	
↕	s-get-workspace-name	-X->	
↕		-X->	test.ResourceWebService.getCompare()

MaTraCa

```
ResourceWebService.java ⌘
package test;

import java.io.InputStream;

@Path("v1")
public class ResourceWebService {
    @Path("s-do-compare")
    public void getCompare()
    {
        System.out.println("dummy test r
    }

    @Path("get-merge-uri")
    public void getMergeURI(
        final String workspace,
        final Object request,
        final InputStream is)
    {
```

```
ResourceWebServiceApplication.java  change.xml ⌘
<xf:variable name="resource-service"
    value="concat(instance('i-housekeeping')/server-scheme,
        '://localhost:',
        instance('i-housekeeping')/server-port,
        '/resources-web/v1')"/>

<xf:variable name="translation" value="instance('i-translation')/resources/translation[@l

<!-- Submissions -->
<xf:submission id="s-do-merge" resource="{ $resource-service }/get-merge-uri?workspace={ins
    <xf:load ev:event="xforms-submit-done" resource="{instance('i-merge-uri')/uri}"/>
    <xf:action ev:event="xforms-submit-error">
        <xf:dispatch name="e-add-message" target="m-default">
            <xxf:context name="message" value="$translation/messages/changes/merge-nok"/>
            <xxf:context name="params" value="(event('response-body'))"/>
        </xf:dispatch>
    </xf:action>
</xf:submission>
```

Console Link View ⌘

!	Source	Link status	Target
↕	s-do-merge	--->	test.ResourceWebService.getMergeURI(final String work...
↕	s-get-repos	-X->	
↕	s-get-workspaces	-X->	
↕	s-revert	-X->	
↕	s-git-commit-push	-X->	
↕	s-git-use-theirs	-X->	
↕	s-abort-merge	-X->	
↕	s-git-update	-X->	
↕	s-git-use-ours	-X->	
↕	s-do-compare	-X->	
↕	s-get-workspace-name	-X->	
↕		-X->	test.ResourceWebService.getCompare()

MaTraCa

The screenshot shows the Eclipse IDE interface with the MaTraCa Link View open. The Link View displays a table of links between source code elements and their targets. The table has three columns: Source, Link status, and Target. The source elements are various methods from the ResourceWebService class, and the targets are specific method calls within the same class.

Source	Link status	Target
s-do-merge	--->	test.ResourceWebService.getMergeURI(final String work...
s-get-repos	-X->	
s-revert	-X->	
s-git-commit-push	-X->	
s-git-use-theirs	-X->	
s-abort-merge	-X->	
s-git-update	-X->	
s-git-use-ours	-X->	
s-get-workspace-name	-X->	
s-do-compare	-X->	
s-get-workspaces	-X->	test.ResourceWebService.getCompare()
s-abort-merge	-X->	
s-get-workspaces	-X->	
s-git-update	-X->	
s-revert	-X->	
s-do-compare	-X->	test.ResourceWebService.getCompare()

The Outline view on the left shows the project structure with the ResourceWebService class selected. The Console view at the top shows the execution of the code, and the Link View at the bottom provides a detailed view of the links between source code elements.

MaTraCa

The screenshot displays an IDE interface with several components:

- Project Explorer:** Shows a project named 'test Ekeko'.
- Code Editor:** Contains three files: `SuperActivator.java`, `ResourceWebService.java`, and `change.xml`.
 - `ResourceWebService.java` snippet:

```
@Path("v1")
public class ResourceWebService {

    // @POST
    // @Path("get-merge-uri")
    // @Produces(MediaType.APPLICATION_XML)
    public void getMergeURI(
        // @QueryParam("workspace")
        final String workspace,
        // @Context
        final ...
    ) {
```
 - `change.xml` snippet:

```
<!-- Submissions -->
<xf:submission id="s-do-merge" resource="{resource-service}/get-merge-uri"
  <xf:load ev:event="xforms-submit-done" resource="{instance('i-merge-uri'
  <xf:action ev:event="xforms-submit-error">
    <xf:dispatch name="e-add-message" target="m-default">
      <xxf:context name="message" value="$translation/messages/change
      <xxf:context name="params" value="(event('response-body'))"/>
    </xf:dispatch>
  </xf:actions>
```
- Link View Dialog:** A modal dialog box titled 'Link View' with a warning icon and the text 'Action 2 (compare against baseline) executed'. It has an 'OK' button.
- Tasks Console:** A table showing task execution details.
- Outline:** Shows a file named 'xhtml:html' with the namespace `xmlns:xhtml=http://www.w3.or`.

Source	Link Status	Target
s-do-merge	--->	test.ResourceWebService...
s-get-workspaces	-X->	
s-do-compare	-X->	
	-X->	test.ResourceWebService...

MaTraCa

The screenshot displays the Eclipse IDE interface with the following components:

- Project Explorer:** Shows a project named 'test Ekeko'.
- Code Editors:** Opened files include 'SuperActivator.java', 'ResourceWebService.java', 'ResourceWebServiceApplication.java', and 'change.xml'. The 'ResourceWebService.java' editor shows a method signature: `public void getCompare()`.
- Link View:** A dialog box titled 'Link View' is open, displaying the message: 'Action 2 (compare against baseline) executed'. It has an 'OK' button.
- Tasks Console:** A table showing the execution of tasks. The visible part of the table is as follows:

Source	Link Status	Target
s-do-merge	---	test.ResourceWebService.getMergeURI(final String workspace,final Object request,final InputStream is)
s-do-compare	---	test.ResourceWebService.getCompare()

At the bottom left, the **Outline** view shows the class structure for 'ResourceWebService' with methods: `getMergeURI(String, Object, InputStream)` and `getCompare() : void`.