A Unified Approach to **Architecture Conformance Checking**

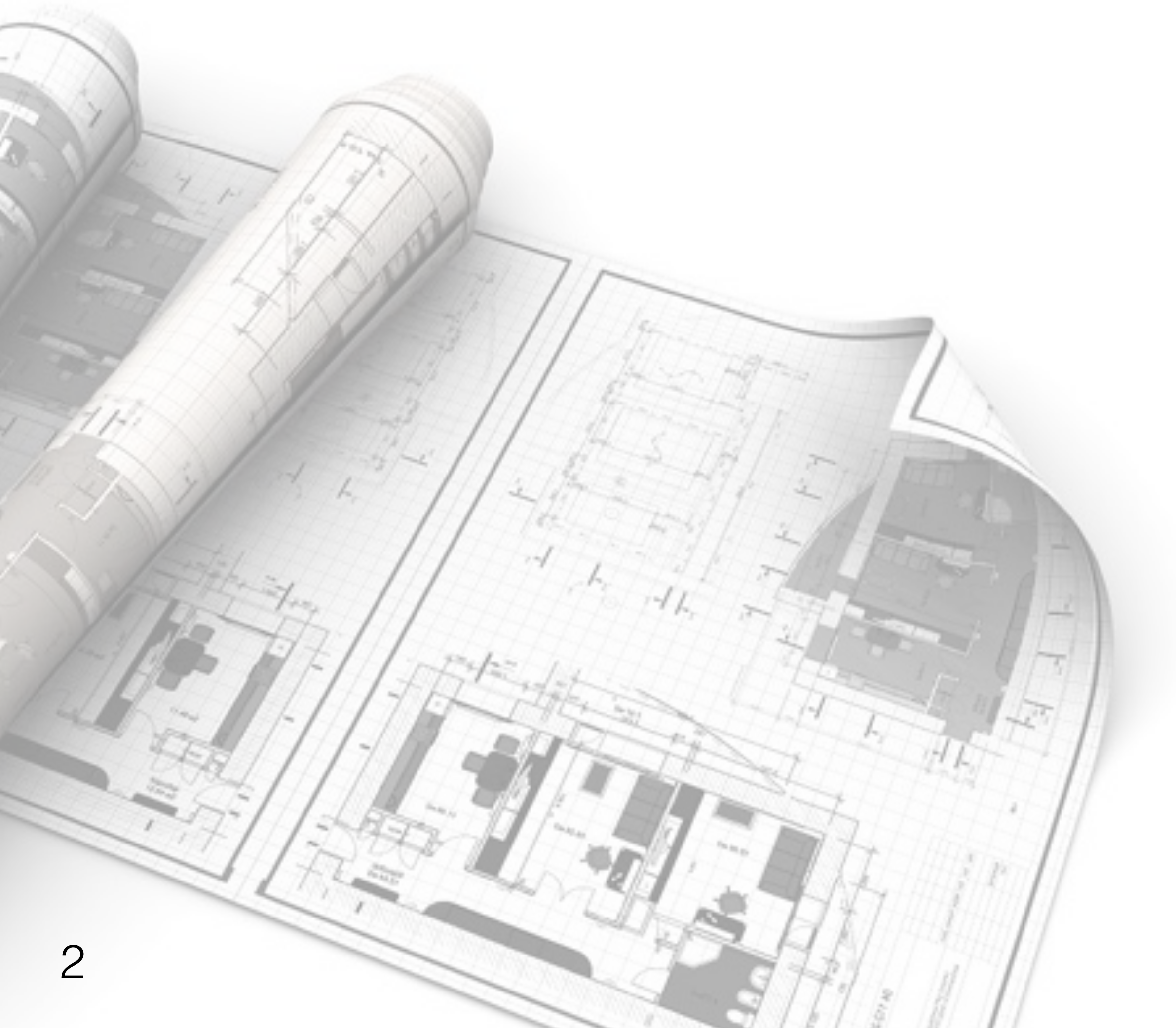
Andrea Caracciolo, Mircea Filip Lungu, Oscar Nierstrasz

http://scg.unibe.ch

b
UNIVERSITÄT
BERN

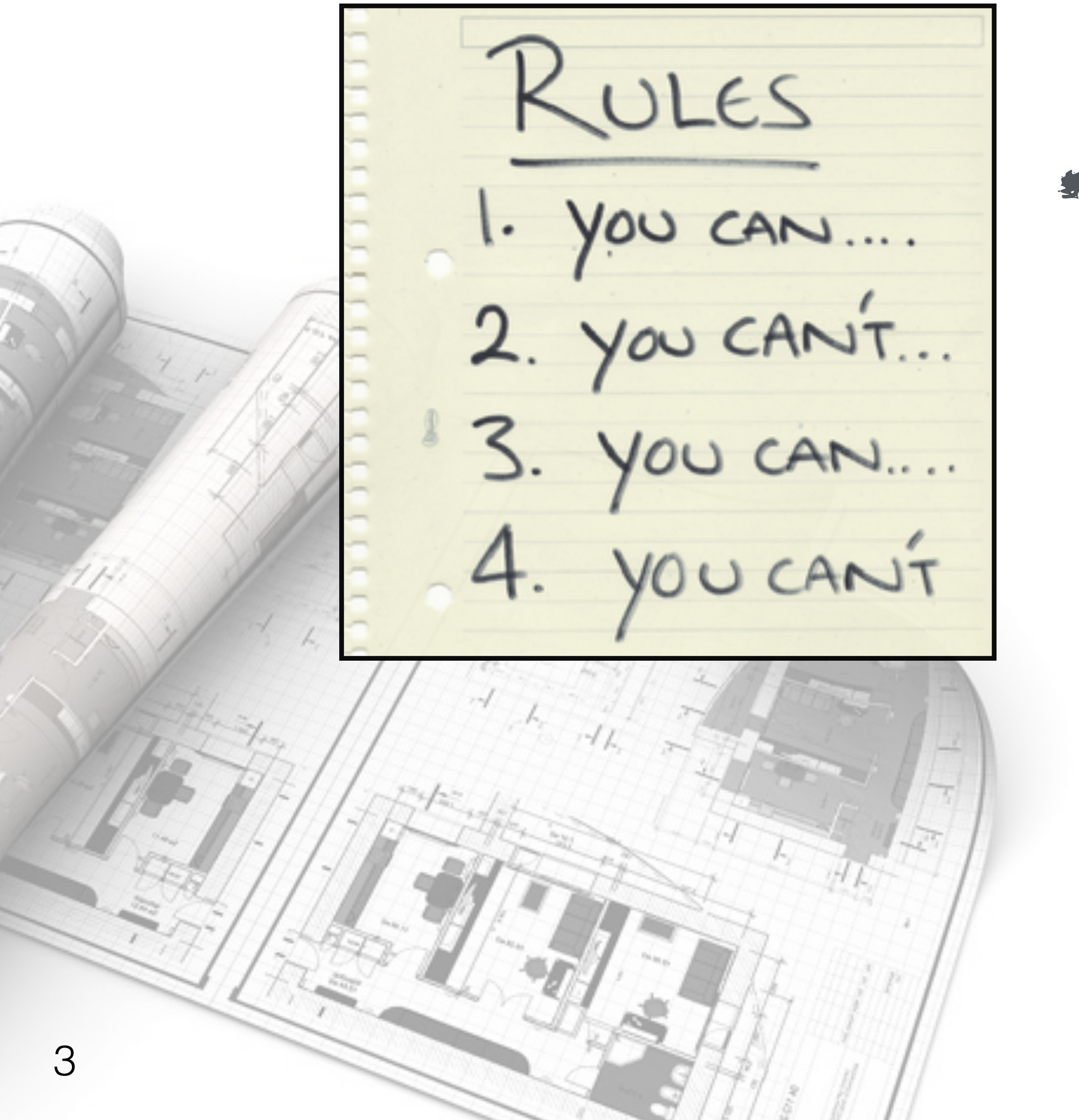
Architecture Erosion

Design ≠ Code



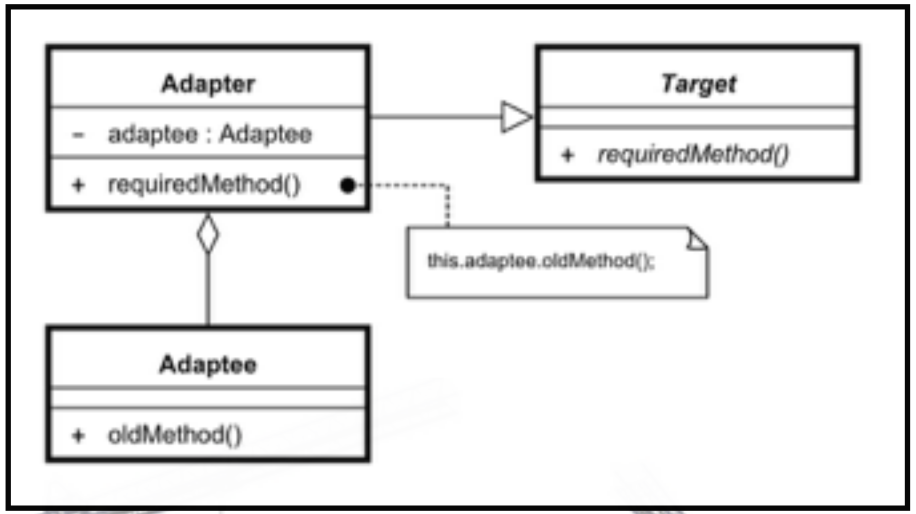
Architecture Erosion

Compliance Monitoring

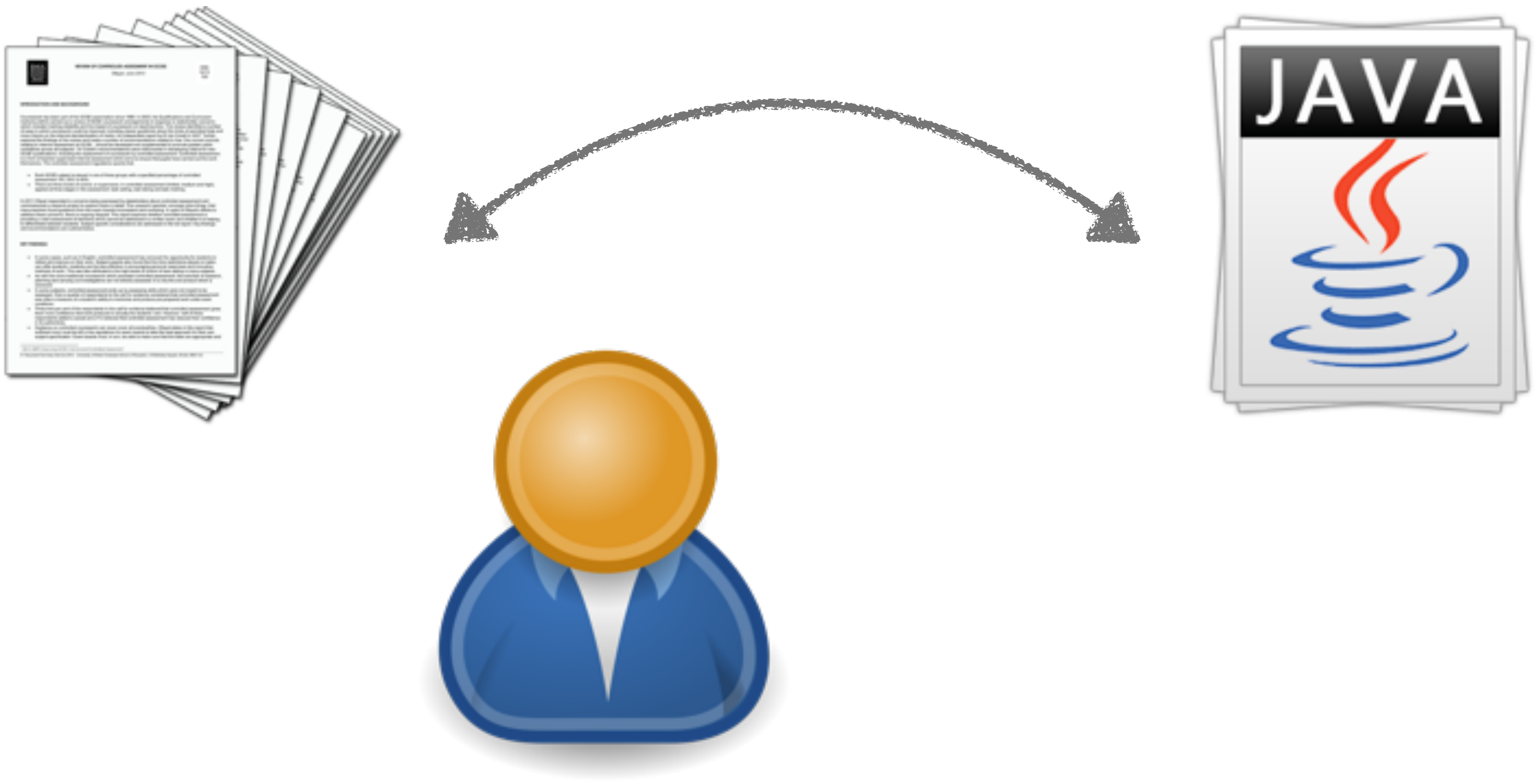


RULES

1. you can....
2. you can't...
3. you can....
4. you can't



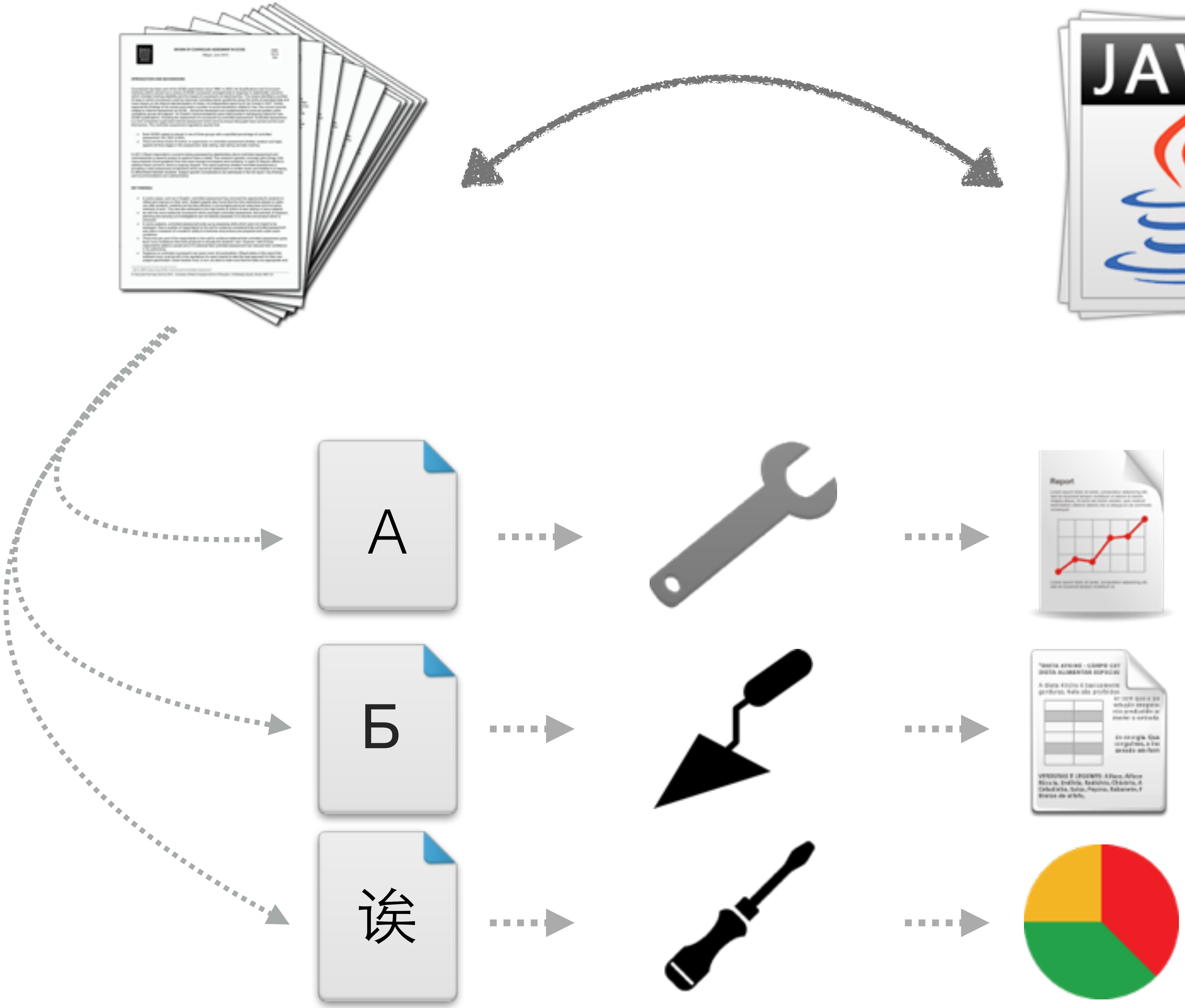
Compliance Monitoring



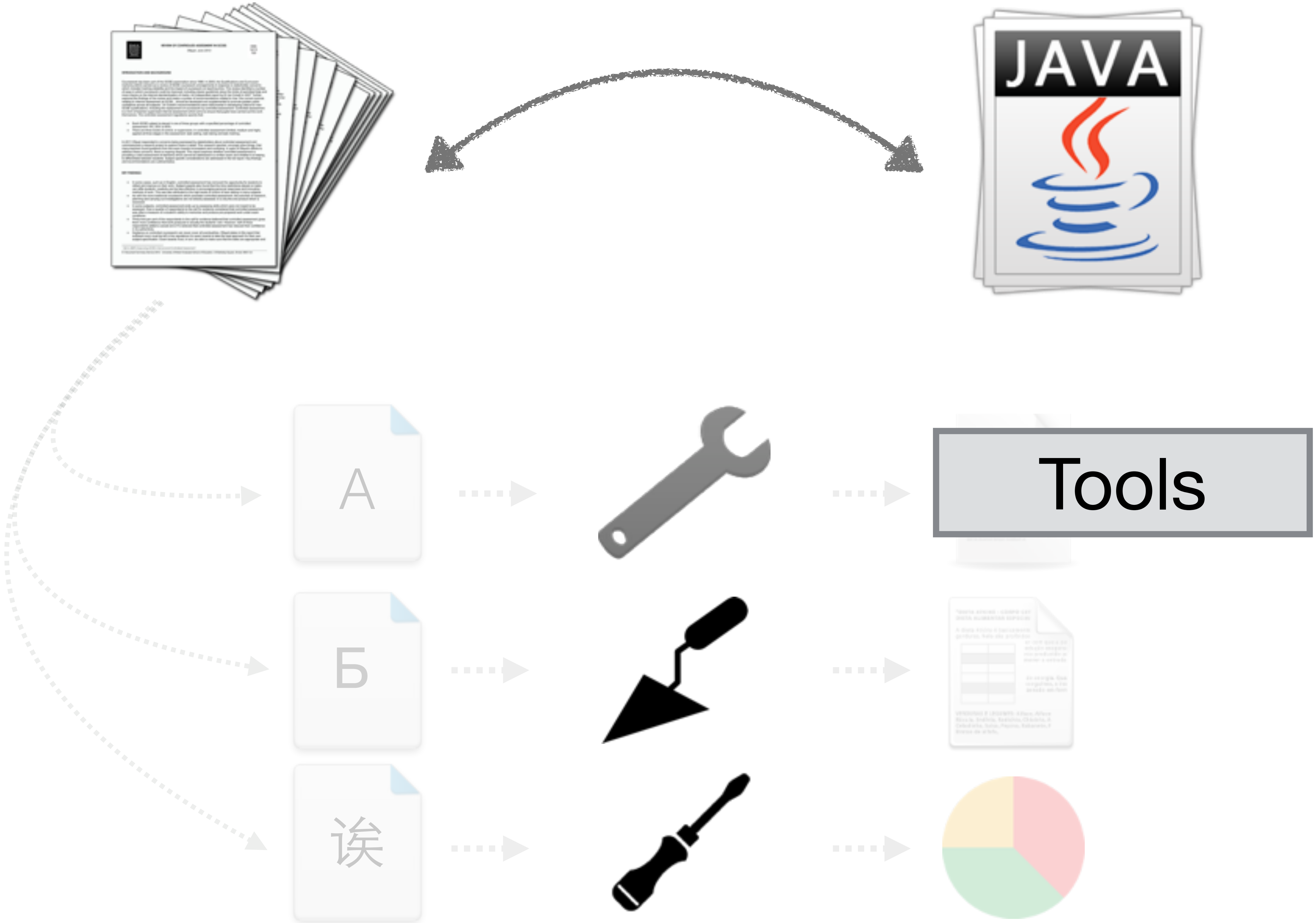
60% **non-automated** techniques *

* **How Do Software Architects Specify and Validate Quality Requirements?** (ECSA 2014)

Compliance Monitoring

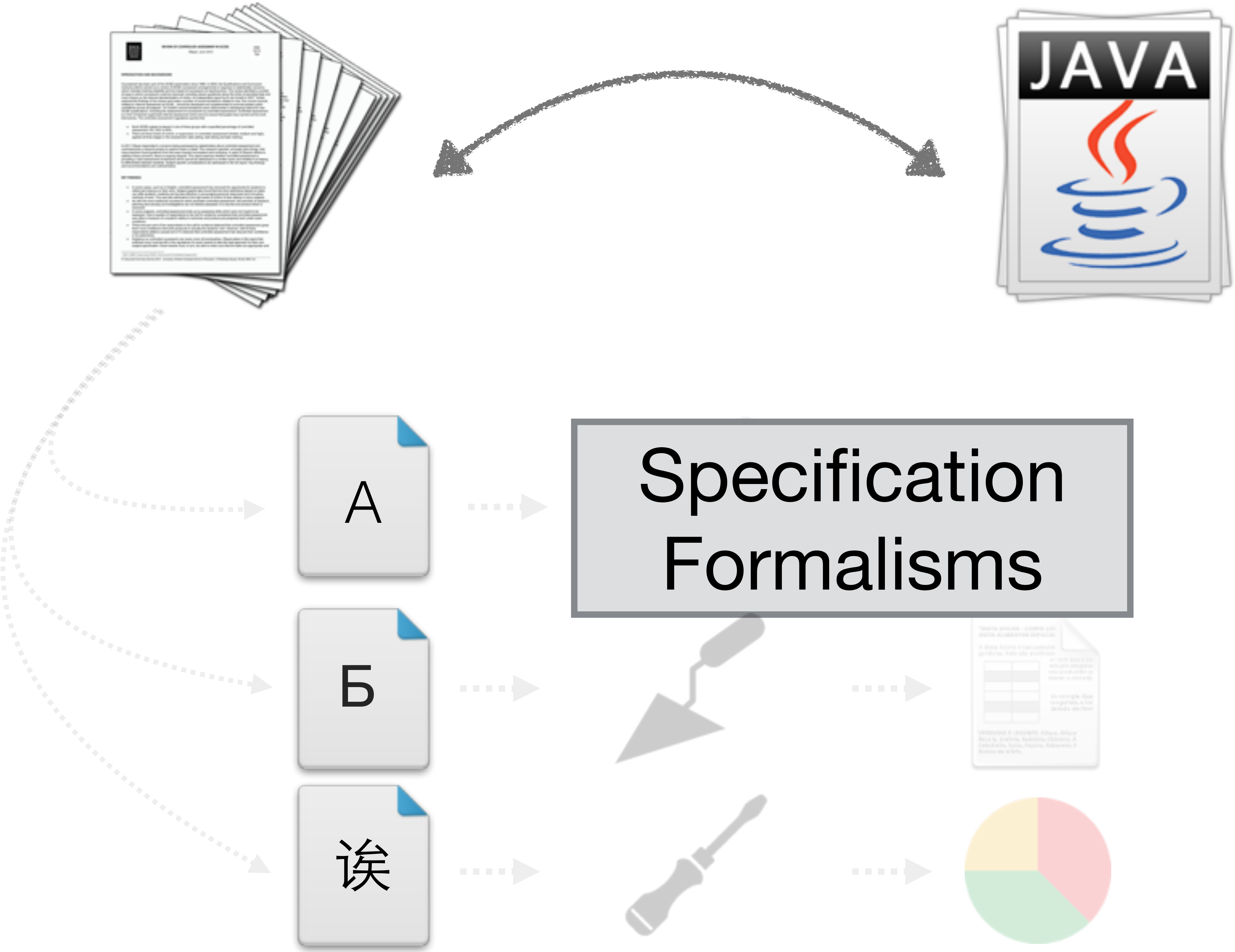


Compliance Monitoring



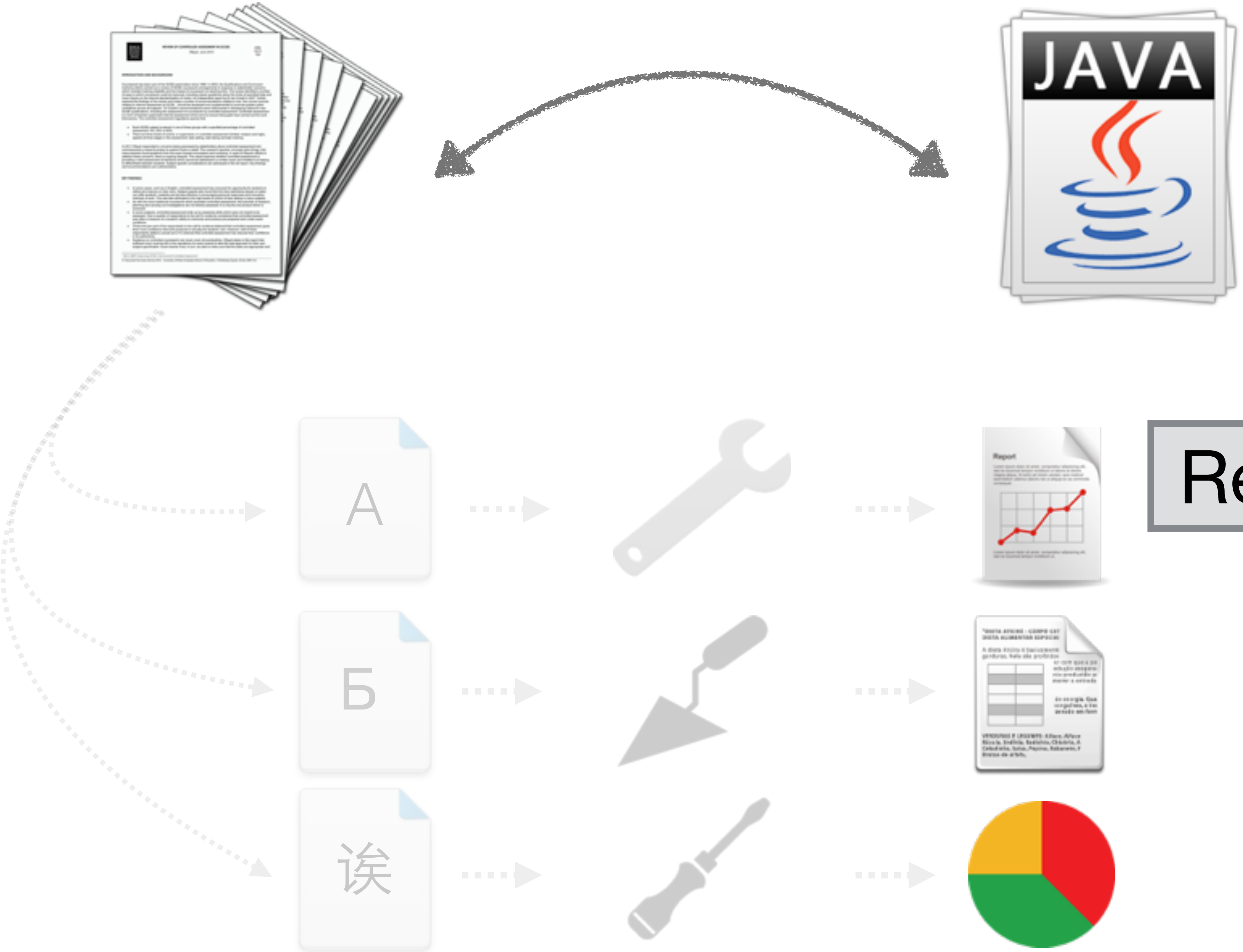
Tools

Compliance Monitoring



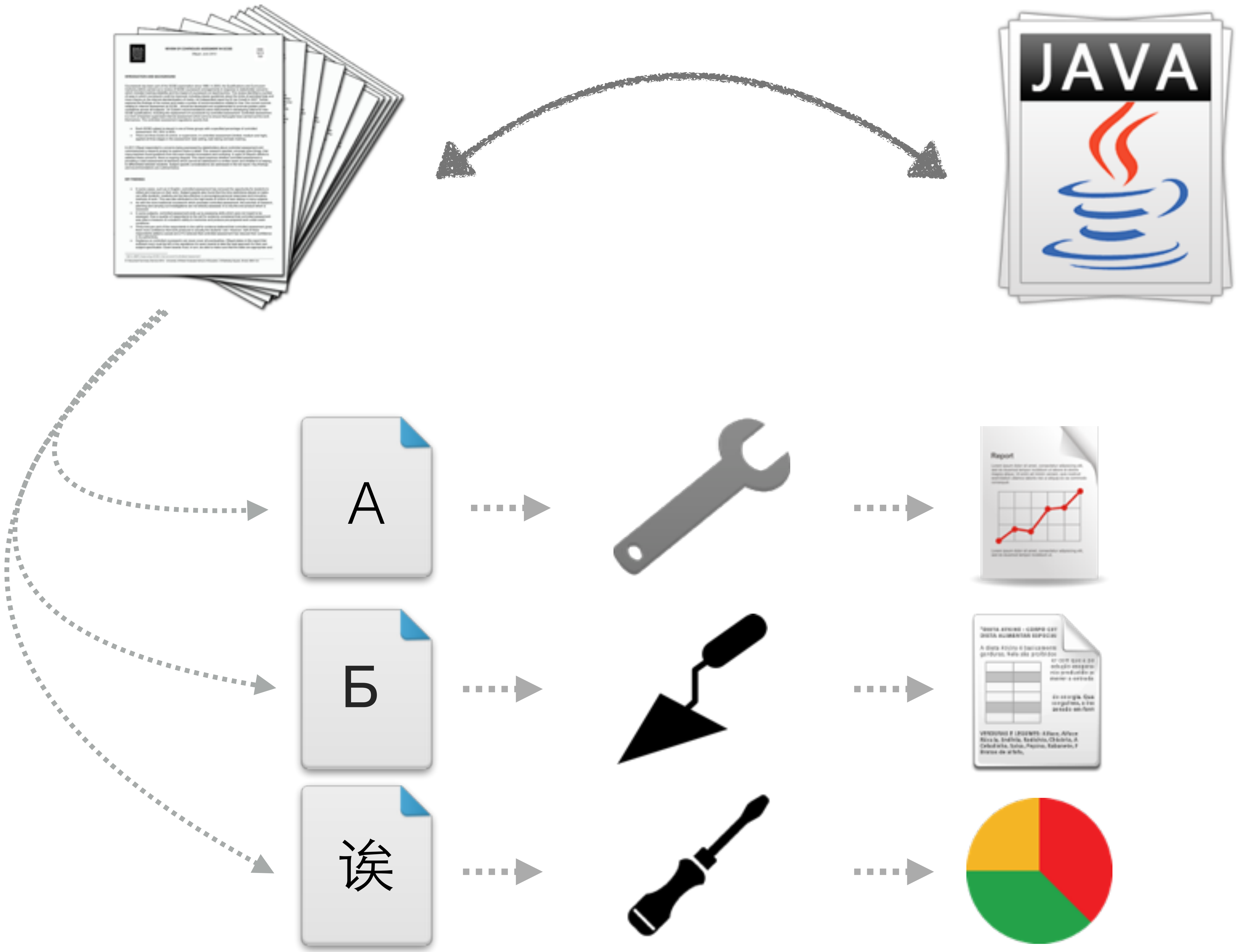
Specification Formalisms

Compliance Monitoring

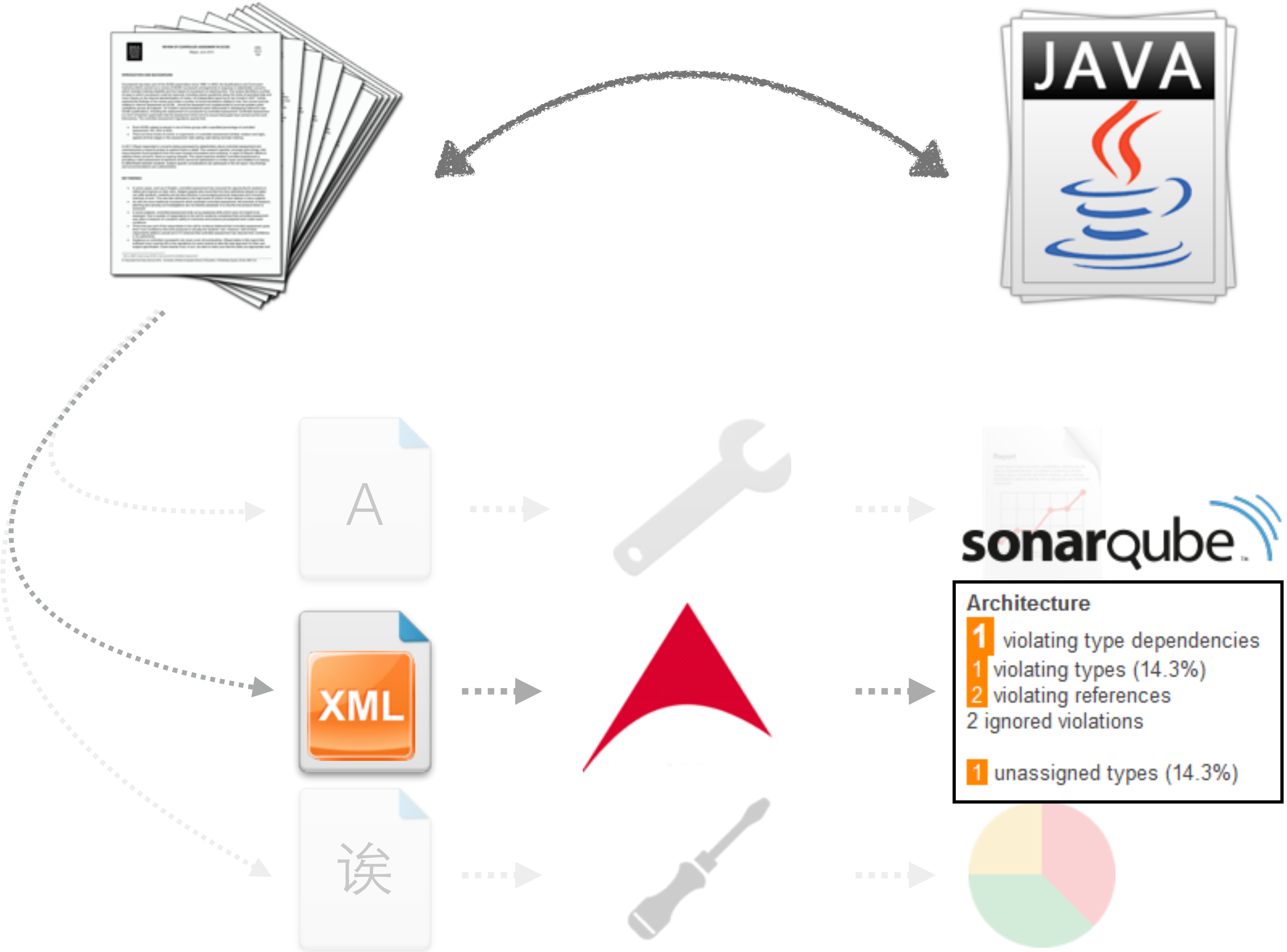


Reports

Compliance Monitoring



Compliance Monitoring



Architecture

1 violating type dependencies

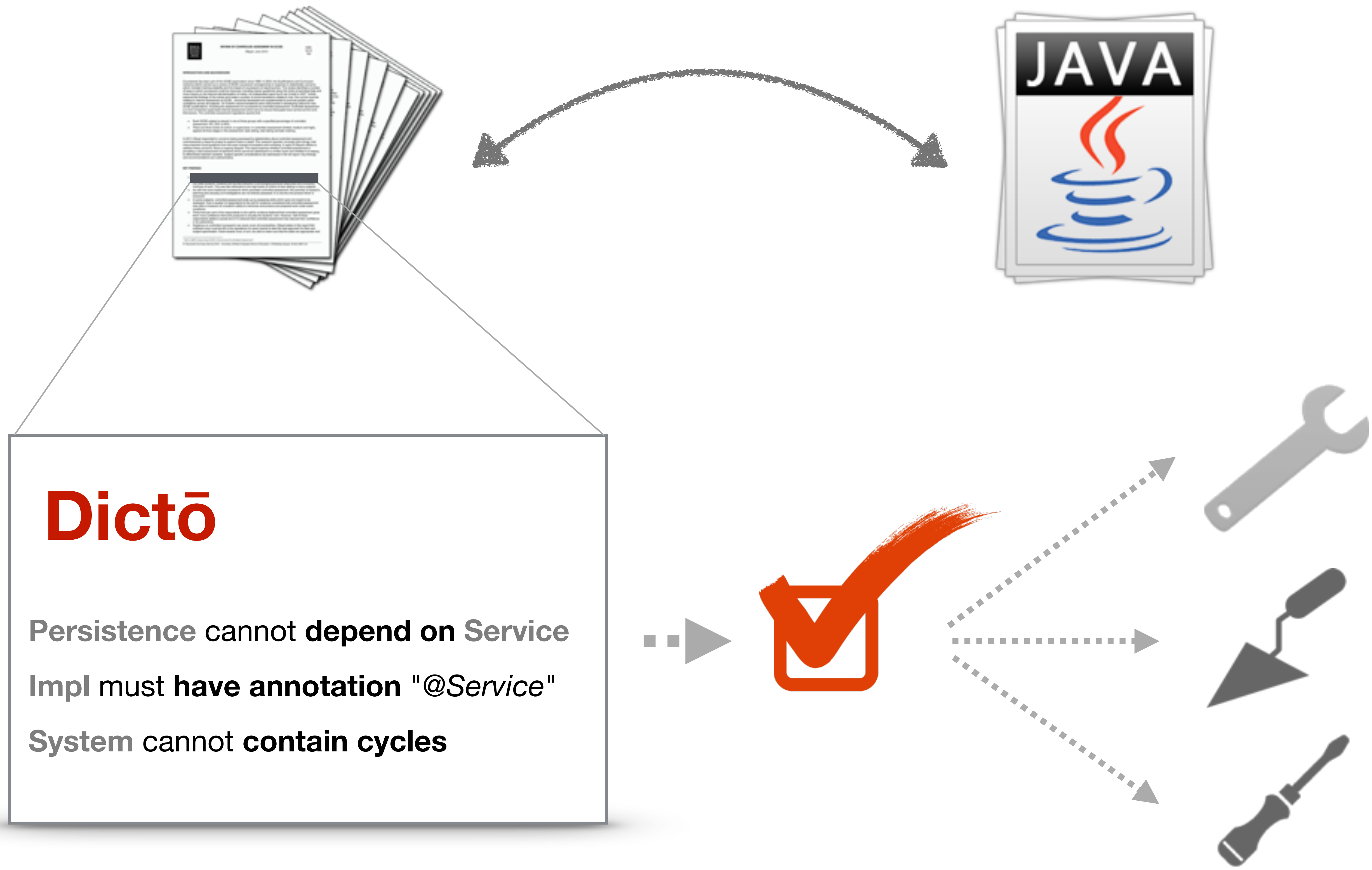
1 violating types (14.3%)

2 violating references

2 ignored violations

1 unassigned types (14.3%)

A Unified Approach



Dictō

Persistence cannot **depend on Service**

Impl must **have annotation** "*@Service*"

System cannot **contain cycles**

@ scg.unibe.ch/dicto

Rules

Persistence cannot **depend on Service**

Service must **have annotation** “@Service”

System cannot **contain cycles**

@ scg.unibe.ch/dicto

Rules

Persistence cannot **depend on** Service

Service must **have annotation** “@Service”

System cannot **contain cycles**

Entities

Persistence = Package with name:"*app.*persistence.**"

Service = Package with name:”*app.*service.**"

Rules

Persistence cannot **depend on Service**

only **Service** can **have annotation** “@Service”

System cannot **contain cycles**

Entities

Persistence = Package with name:"_app.*persistence.*_"

Service = Package with name:”_app.*service.*_"

Dictō

@ scg.unibe.ch/dicto

Rules

Persistence cannot **depend on** Service

only Service can **have annotation** “@Service”

System cannot **contain cycles**

Entities

Persistence = Package with name:"_app.*persistence.*_"

Service = Package with name:”_app.*service.*_”,
name!:"_*persistence*_"

Rules

Maintainability

Method can only **be named** "String"

only *Package* can **contain dead methods**

Compatibility

XMLTag must **have attribute** “String"

WebResource must **have content** "String"

Reliability

Method must **catch** Class

Performance

WebResource must **have latency <** *int* **ms**

WebResource must **handle load from** *int* **users**

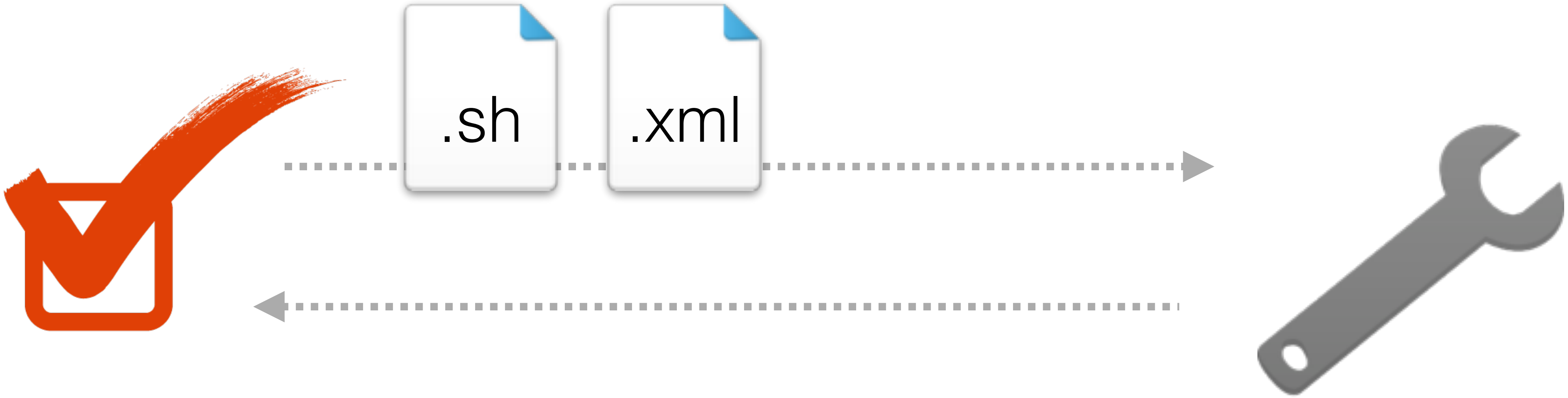
Analysis

Persistence cannot depend on Service

depend-on(app.xx.persistence.yy, app.xx.service)

depend-on(app.xx.persistence.yy2, app.xx.service)

depend-on(app.xx.persistence.yy3, app.xx.service)



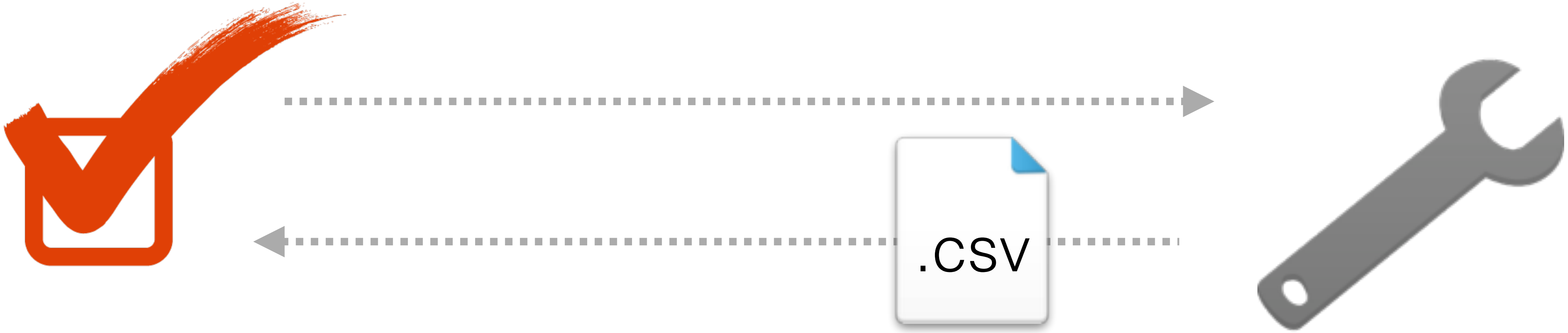
Analysis

Persistence cannot depend on Service

🚫 ***depend-on(app.xx.persistence.yy, app.xx.service) = True***

depend-on(app.xx.persistence.yy2, app.xx.service) = False

depend-on(app.xx.persistence.yy3, app.xx.service) = False



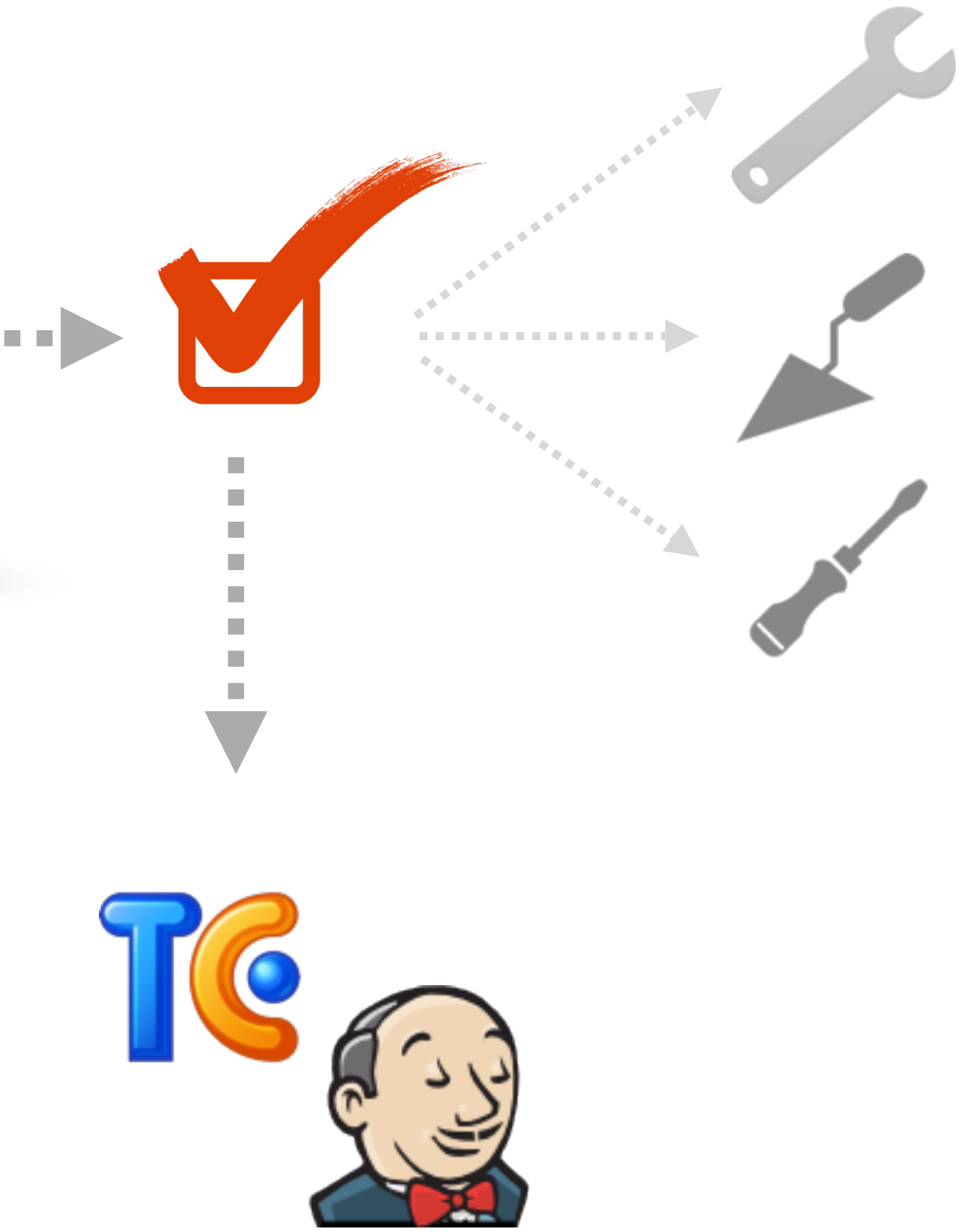
Reporting

Dictō

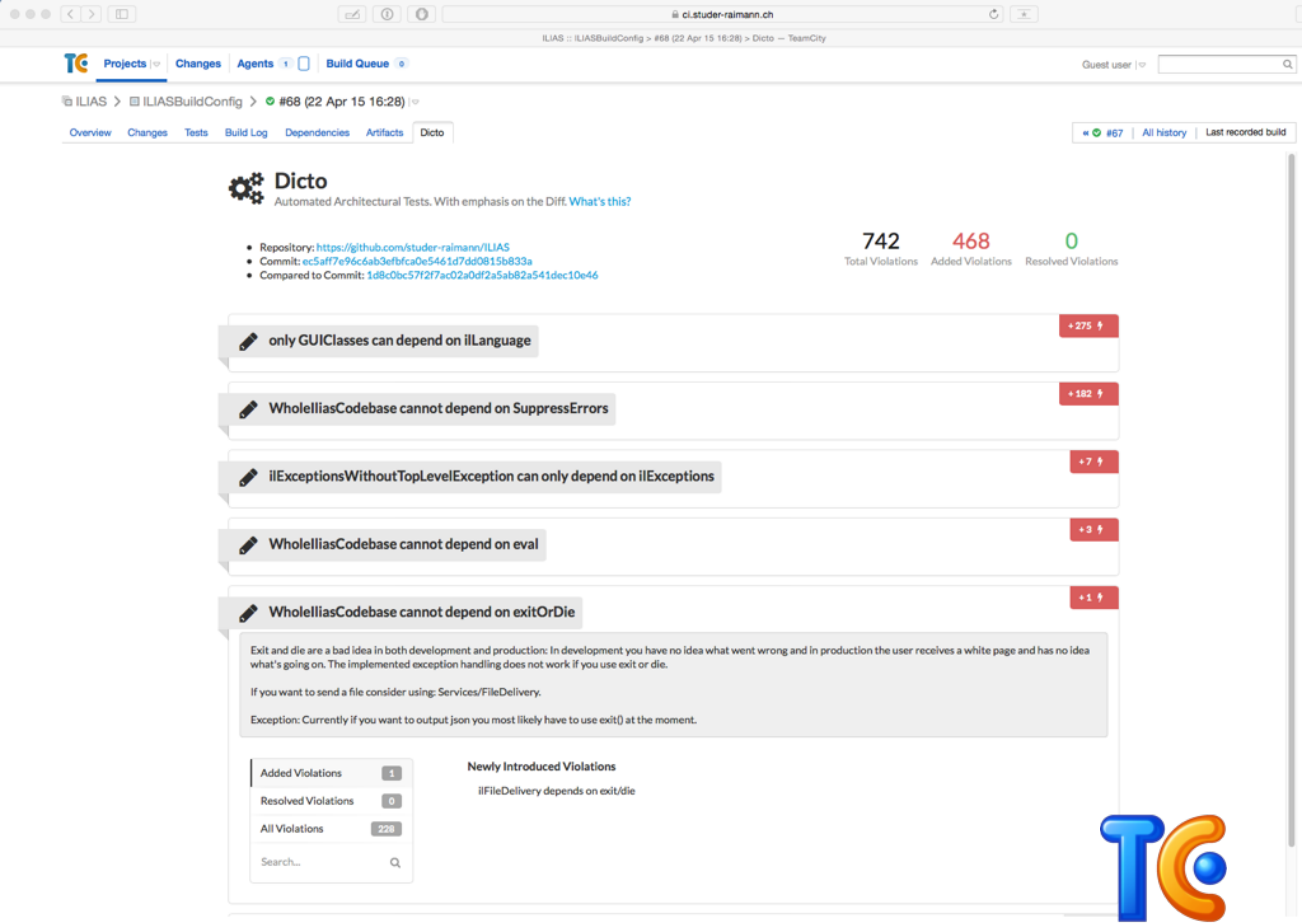
Persistence cannot **depend on Service**

Service must **have annotation** “@Service”

System cannot **contain cycles**



Statistics



Dicto

Automated Architectural Tests. With emphasis on the Diff. What's this?

- Repository: https://github.com/studer-raimann/ILIAS
- Commit: ec5aff7e96c6ab3efbfca0e5461d7dd0815b833a
- Compared to Commit: 1d8c0bc57f2f7ac02a0df2a5ab82a541dec10e46

742	468	0
Total Violations	Added Violations	Resolved Violations

only GUIClasses can depend on ilLanguage

+275

WholeIliasCodebase cannot depend on SuppressErrors

+182

ilExceptionsWithoutTopLevelException can only depend on ilExceptions

+7

WholeIliasCodebase cannot depend on eval

+3

WholeIliasCodebase cannot depend on exitOrDie

+1

Exit and die are a bad idea in both development and production: In development you have no idea what went wrong and in production the user receives a white page and has no idea what's going on. The implemented exception handling does not work if you use exit or die.

If you want to send a file consider using: Services/FileDelivery.

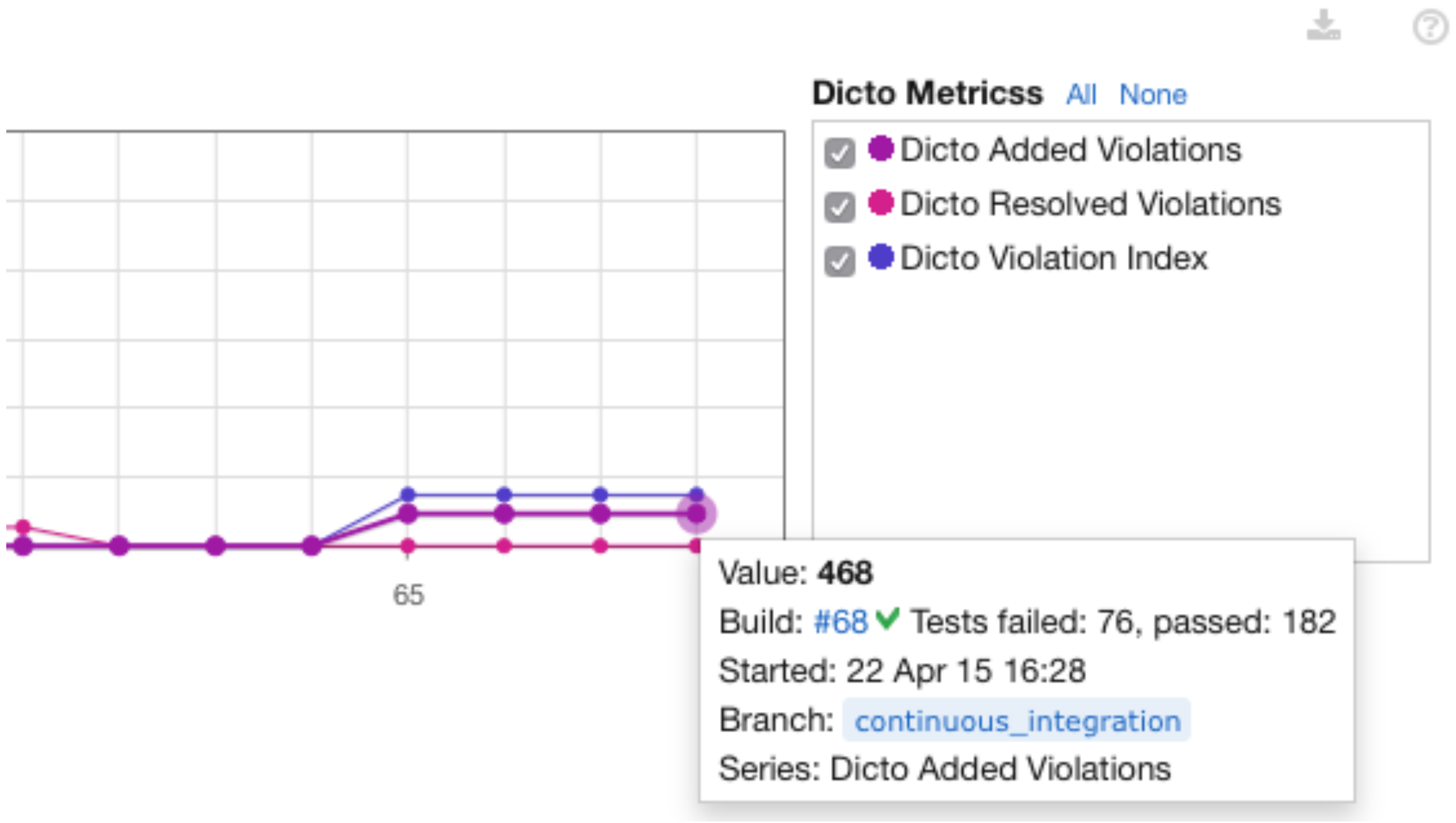
Exception: Currently if you want to output json you most likely have to use exit() at the moment.

Added Violations	1
Resolved Violations	0
All Violations	228

Search...

Newly Introduced Violations

ilFileDelivery depends on exit/die


Dicto Metricss
All
None
☑ ● Dicto Added Violations
☑ ● Dicto Resolved Violations
☑ ● Dicto Violation Index
65

Value: **468**

Build: #68 ✓ Tests failed: 76, passed: 182

Started: 22 Apr 15 16:28

Branch: continuous_integration

Series: Dicto Added Violations

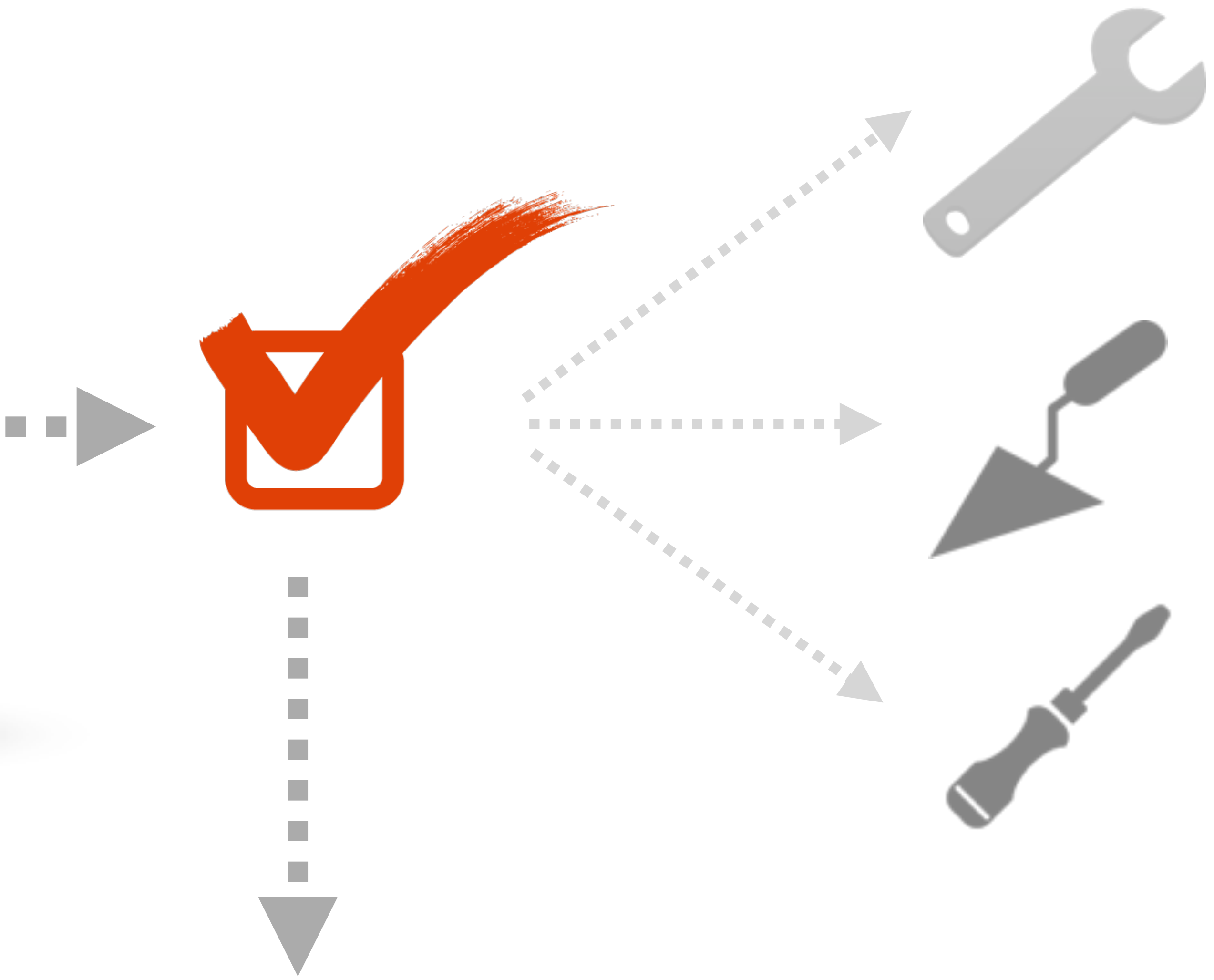
Reporting

Dictō

Persistence cannot **depend on Service**

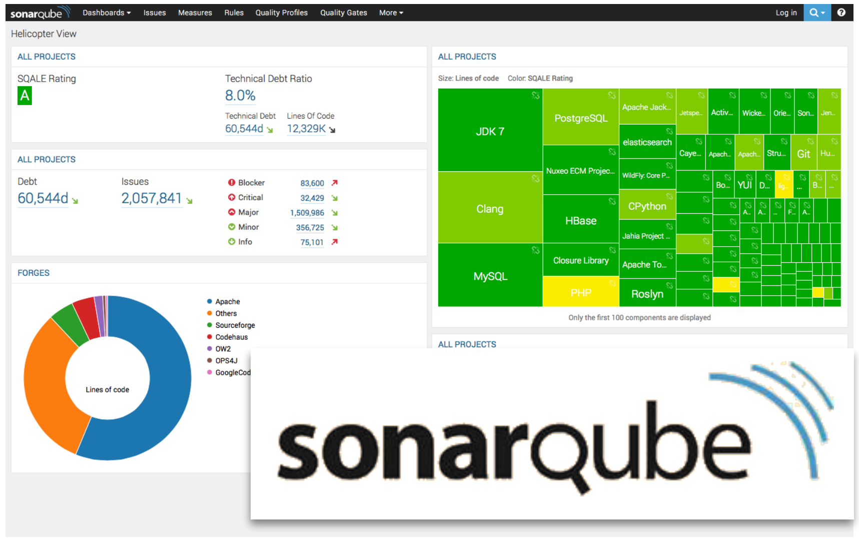
Service must **have annotation** "@Service"

System cannot **contain cycles**



New Issue

Statistics



Helicopter View

ALL PROJECTS

SQALE Rating A

Technical Debt Ratio 8.0%

Technical Debt 60,544d

Lines Of Code 12,329K

ALL PROJECTS

Debt 60,544d

Issues 2,057,841

Blocker	83,600
Critical	32,429
Major	1,509,986
Minor	356,725
Info	75,101

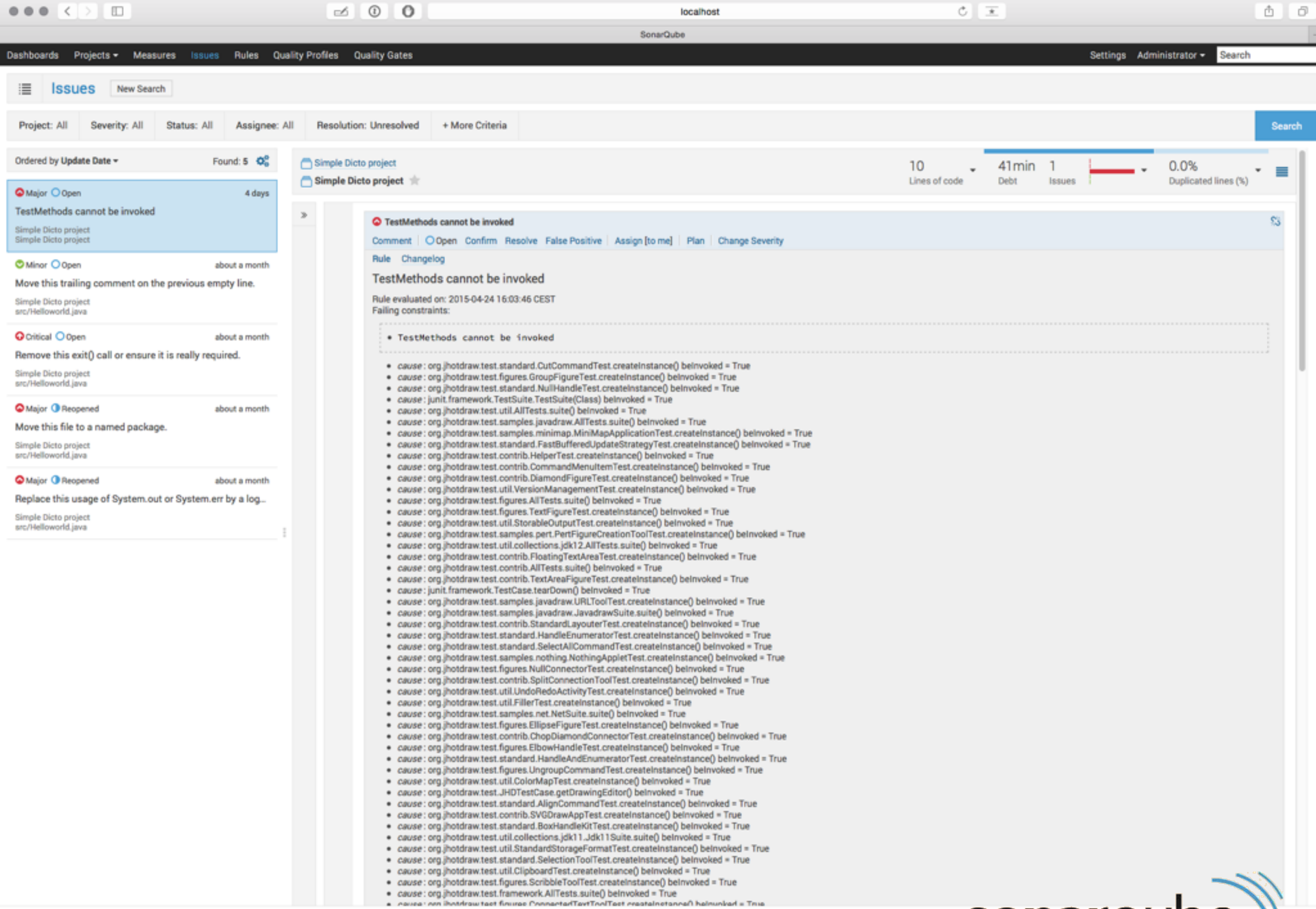
FORGES

ALL PROJECTS

Size: Lines of code Color: SQALE Rating

Only the first 100 components are displayed

ALL PROJECTS



Major Open 4 days
TestMethods cannot be invoked
Simple Dicto project
Simple Dicto project

Minor Open about a month
Move this trailing comment on the previous empty line.
Simple Dicto project
src/Helloworld.java

Critical Open about a month
Remove this exit() call or ensure it is really required.
Simple Dicto project
src/Helloworld.java

Major Reopened about a month
Move this file to a named package.
Simple Dicto project
src/Helloworld.java

Major Reopened about a month
Replace this usage of System.out or System.err by a log...
Simple Dicto project
src/Helloworld.java

TestMethods cannot be invoked

TestMethods cannot be invoked

Rule evaluated on: 2015-04-24 16:03:46 CEST
Failing constraints:

```
• TestMethods cannot be invoked
```

- *cause* : org.jhotdraw.test.standard.CutCommandTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.figures.GroupFigureTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.standard.NullHandleTest.createInstance() beInvoked = True
- *cause* : junit.framework.TestSuite.TestSuite(Class) beInvoked = True
- *cause* : org.jhotdraw.test.util.AllTests.suite() beInvoked = True
- *cause* : org.jhotdraw.test.samples.javadraw.AllTests.suite() beInvoked = True
- *cause* : org.jhotdraw.test.samples.minimap.MiniMapApplicationTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.standard.FastBufferedUpdateStrategyTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.contrib.HelperTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.contrib.CommandMenuItemTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.contrib.DiamondFigureTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.util.VersionManagementTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.figures.AllTests.suite() beInvoked = True
- *cause* : org.jhotdraw.test.figures.TextFigureTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.util.StorableOutputTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.samples.pert.PertFigureCreationToolTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.util.collections.jdk12.AllTests.suite() beInvoked = True
- *cause* : org.jhotdraw.test.contrib.FloatingTextAreaTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.contrib.AllTests.suite() beInvoked = True
- *cause* : org.jhotdraw.test.contrib.TextAreaFigureTest.createInstance() beInvoked = True
- *cause* : junit.framework.TestCase.tearDown() beInvoked = True
- *cause* : org.jhotdraw.test.samples.javadraw.URLToolTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.samples.javadraw.JavadrawSuite.suite() beInvoked = True
- *cause* : org.jhotdraw.test.contrib.StandardLayouterTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.standard.HandleEnumeratorTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.standard.SelectAllCommandTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.samples.nothing.NothingAppletTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.figures.NullConnectorTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.contrib.SplitConnectionToolTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.util.UndoRedoActivityTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.util.FillerTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.samples.net.NetSuite.suite() beInvoked = True
- *cause* : org.jhotdraw.test.figures.EllipseFigureTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.contrib.ChopDiamondConnectorTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.figures.ElbowHandleTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.standard.HandleAndEnumeratorTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.figures.UngroupCommandTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.util.ColorMapTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.JHDTestCase.getDrawingEditor() beInvoked = True
- *cause* : org.jhotdraw.test.standard.AlignCommandTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.contrib.SVGDrawAppTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.standard.BoxHandleKitTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.util.collections.jdk11.Jdk11Suite.suite() beInvoked = True
- *cause* : org.jhotdraw.test.util.StandardStorageFormatTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.standard.SelectionToolTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.util.ClipboardTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.figures.ScribbleToolTest.createInstance() beInvoked = True
- *cause* : org.jhotdraw.test.framework.AllTests.suite() beInvoked = True
- *cause* : org.jhotdraw.test.figures.ConnectedTextToolTest.createInstance() beInvoked = True

sonarqube

Evaluation

Open source project

LMS - PHP (1.8M LOC)
12 service providers, 900’000+ users

Large size company

B2B - Java EE (50K LOC)
1’000 employees

Medium size company

various - Java EE / .NET
100 employees

Evaluation

Impact

Process? Culture?

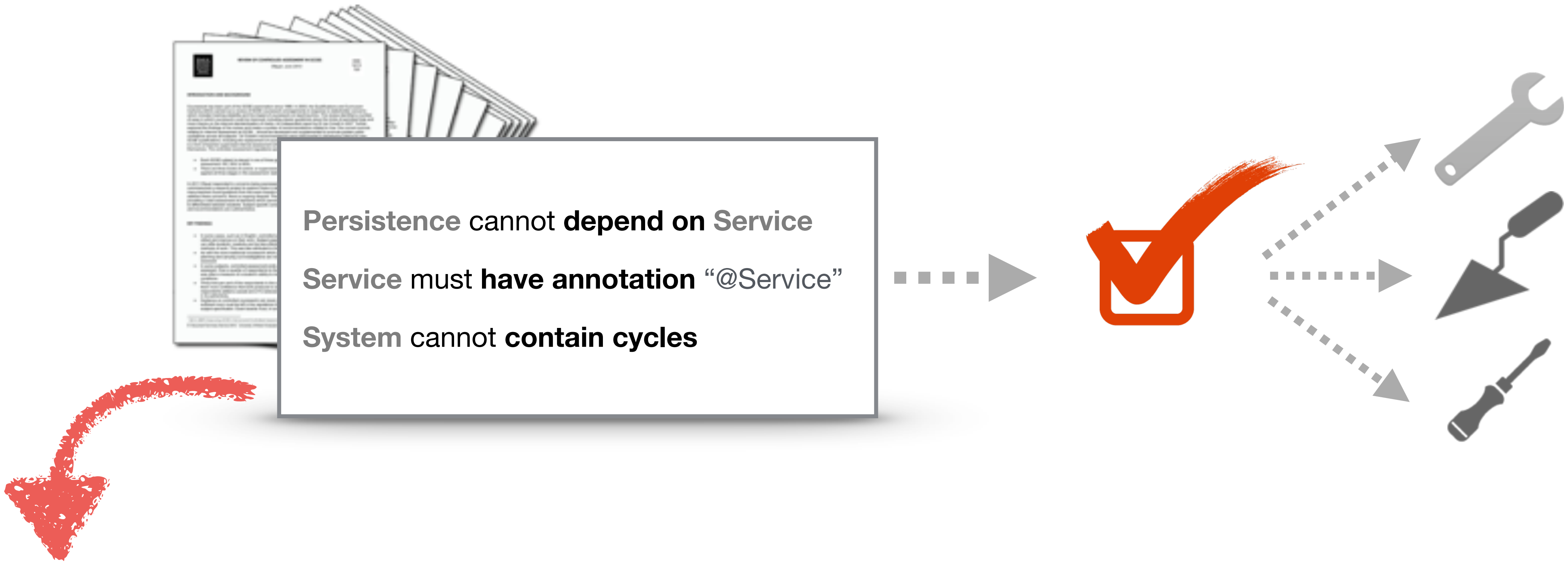
Value

Cost–effectiveness? Quality?

Applicability

Expressivity? Usability?

Summary



Persistence cannot **depend on** Service

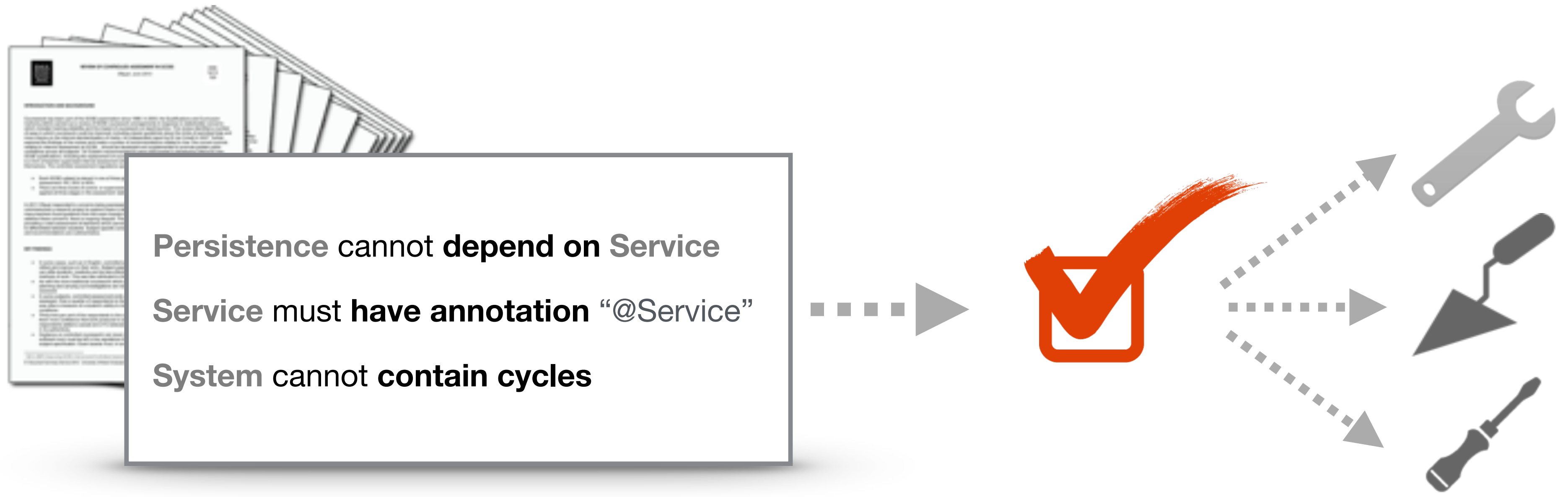
Service must **have annotation** “@Service”

System cannot **contain cycles**

Dictō

A **uniform**, **readable**, **executable** DSL for specifying architectural constraints.

Discussion Topics



Persistence cannot depend on Service

Service must have annotation “@Service”

System cannot contain cycles

- how to streamline/incentivize compliance monitoring?
- which are the common obstacles?