BETA VERSION

Thoughts on Software Quality

and what can we learn from Apache Sling?

Bertrand Delacrétaz
@bdelacretaz, grep.codeconsult.ch

Principal Scientist, CQ/AEM R&D group, Adobe Basel
Apache Software Foundation Member and Director

June 2015
slides revision 2015-06-23

CONNECT WEB EXPERIENCE 2015

Adobe

how can we create a

culture of quality

in our teams?

created in 2007

powers “a few” major websites

still going strong

something must be right

Quality is usually visible from far away

Engineering culture?

Sling ?

Good tools for good craftsmen

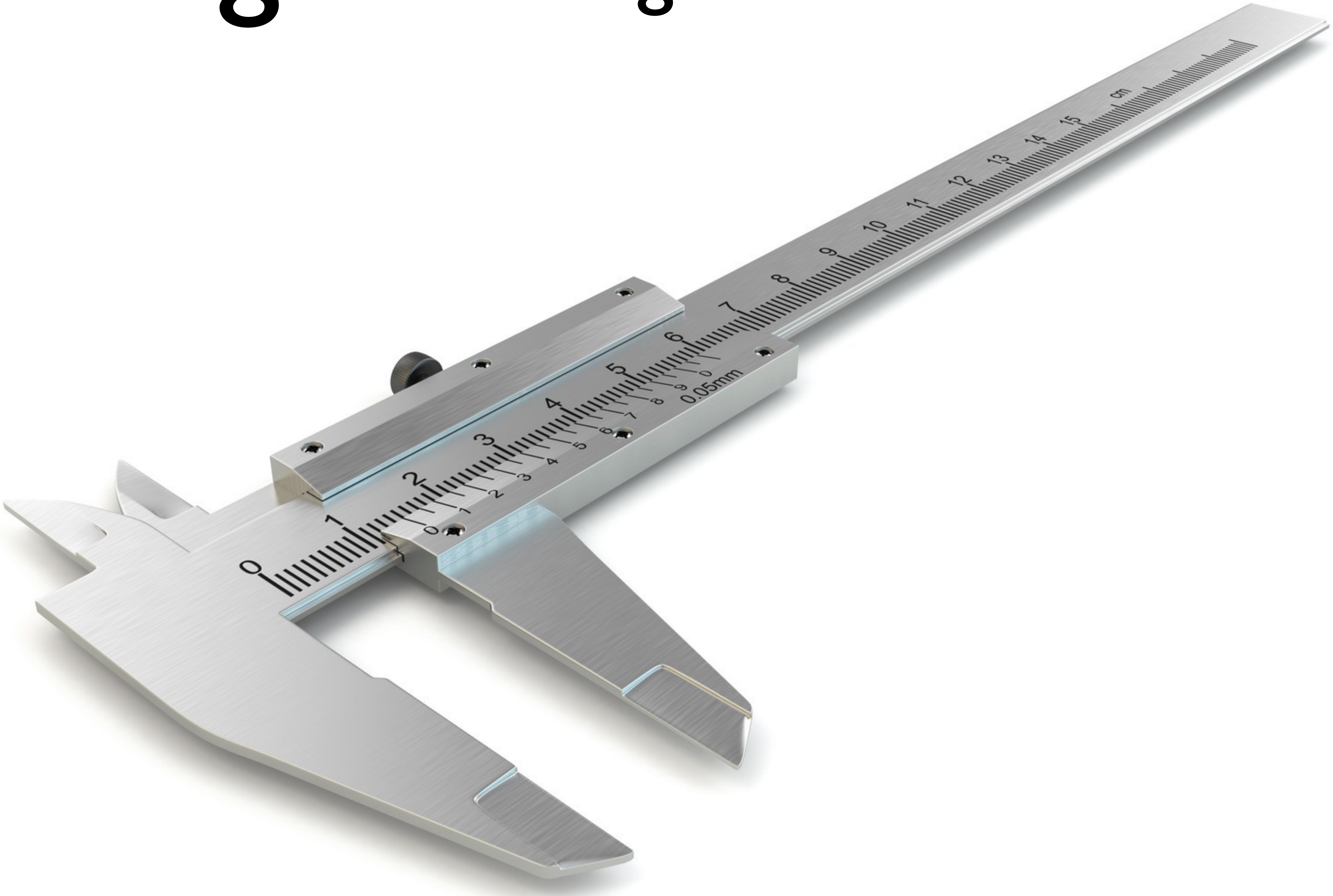
Sling?

Defensive & Conservative

Sling ?

Avoid broken windows

Do one thing and one thing well



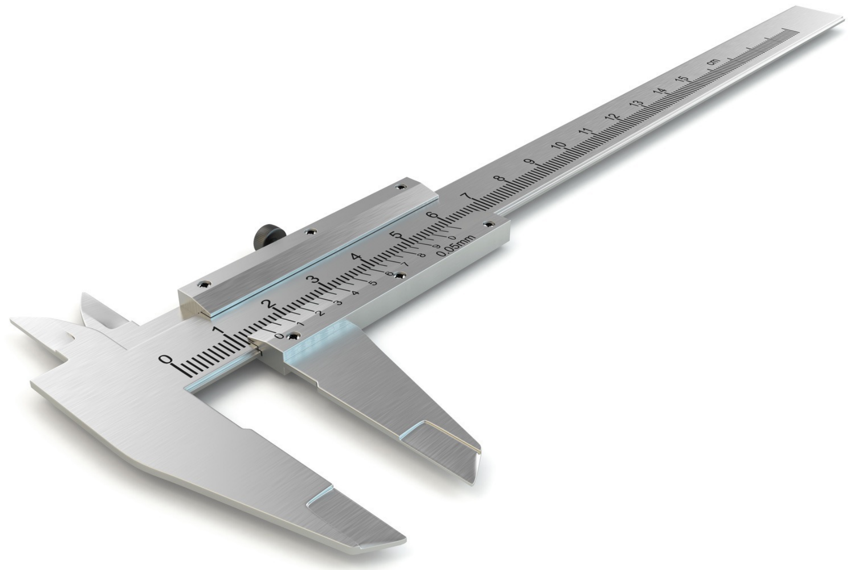
Sling ?

Cultural traits? you can't measure those…

automate!

no broken windows

defensive



do one thing and one thing well

conservative

obvious quality, measurable where possible so, quality is a

cultural

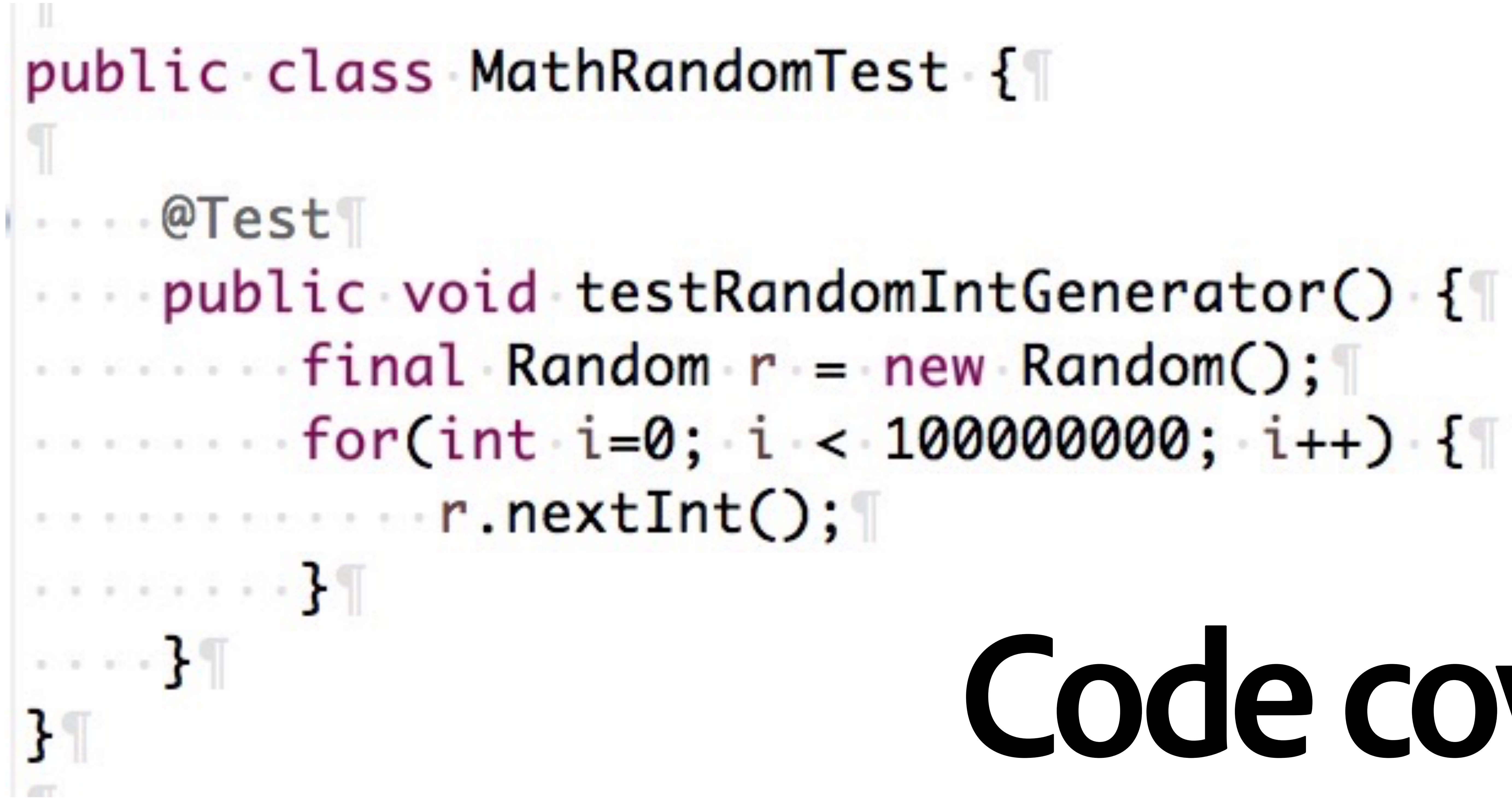
thing?

maybe a few

axes

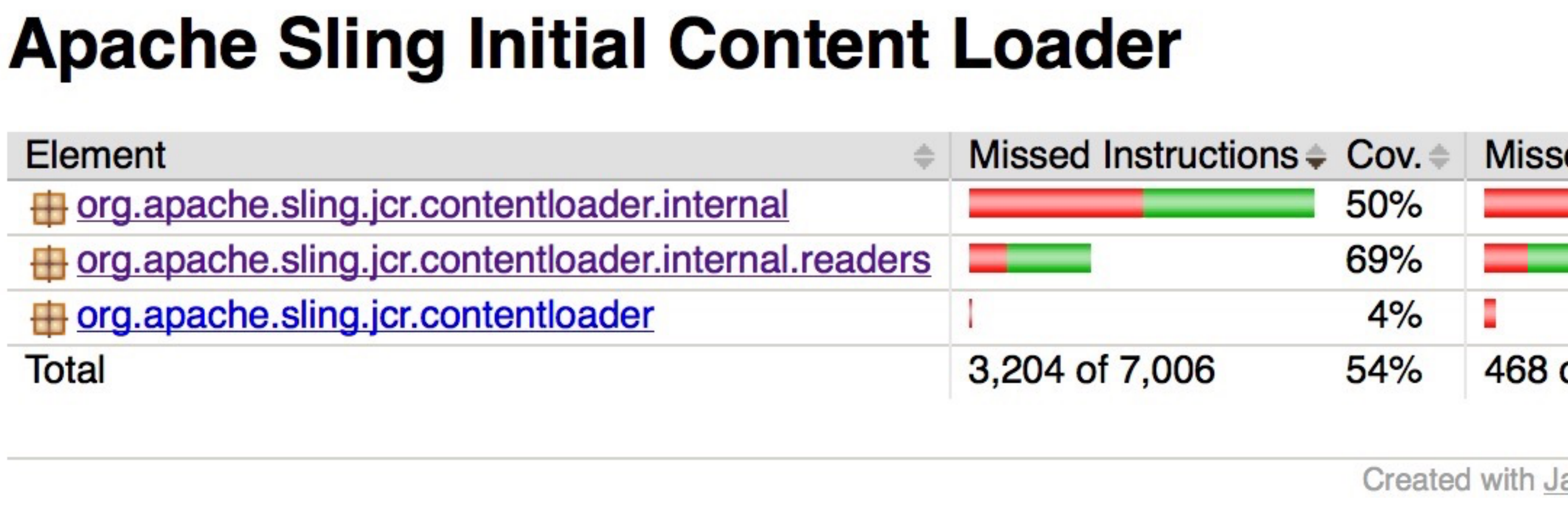
will help us measure

Code coverage? Not a silver bullet.



```
public class MathRandomTest {

    @Test
    public void testRandomIntGenerator() {
        final Random r = new Random();
        for(int i=0; i < 100000000; i++) {
            r.nextInt();
        }
    }
}
```



Apache Sling Initial Content Loader

Element	Missed Instructions	Cov.	Miss
org.apache.sling.jcr.contentloader.internal		50%	
org.apache.sling.jcr.contentloader.internal.readers		69%	
org.apache.sling.jcr.contentloader		4%	
Total	3,204 of 7,006	54%	468 o

Created with Ja

Code coverage: 100%
Semantic coverage: zero!

A generator with `return 42;` would pass.
Not useless…but just one tool among others.

quality is

much wider

than that

Happy users!

Sling ?

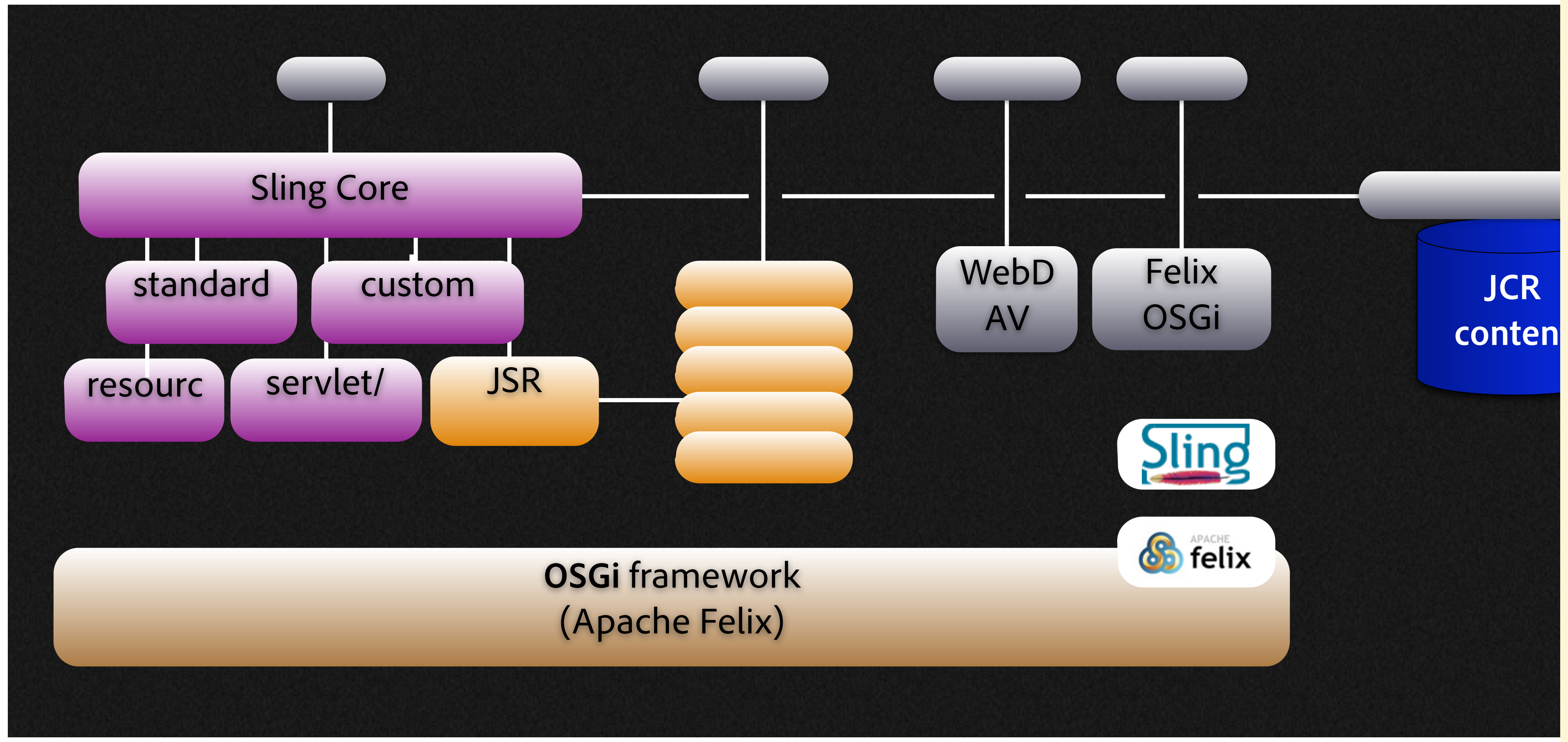
Happy Developers!

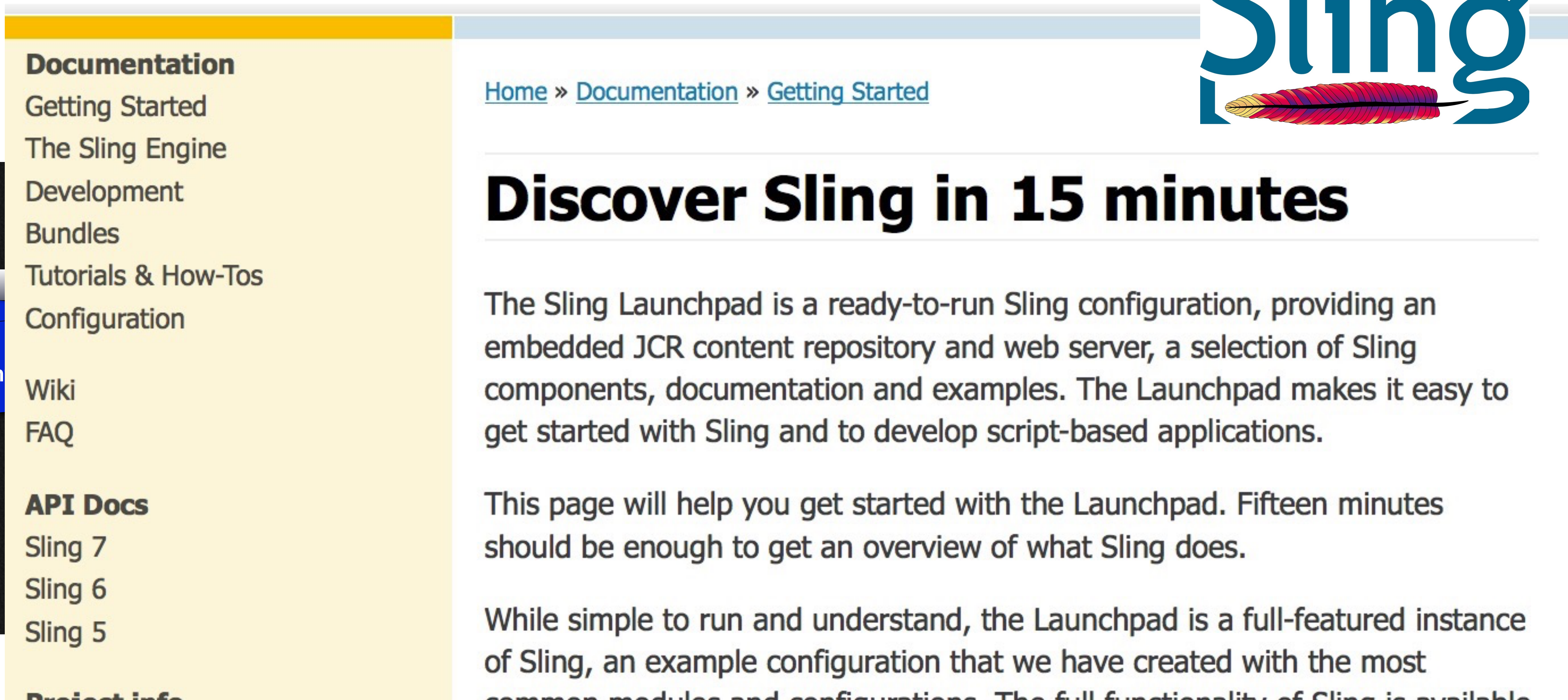
Sling?

Simplicity & adequation

Simplicity & adequation

```
$ cat $(find bundles/engine -name *.java) | wc -l
   13841
$ cat $(find bundles/resourceresolver -name *.java) | wc -l
   18273
$ cat $(find bundles/servlets -name *.java) | wc -l
   19404
$ cat $(find bundles/api -name *.java) | wc -l
   13938
$ █
```





Home » Documentation » Getting Started

Discover Sling in 15 minutes

The Sling Launchpad is a ready-to-run Sling configuration, providing an embedded JCR content repository and web server, a selection of Sling components, documentation and examples. The Launchpad makes it easy to get started with Sling and to develop script-based applications.

This page will help you get started with the Launchpad. Fifteen minutes should be enough to get an overview of what Sling does.

While simple to run and understand, the Launchpad is a full-featured instance of Sling, an example configuration that we have created with the most common modules and configurations. The full functionality of Sling is available

Robustness

Robustness

```
[INFO] ------------------------------------------------------------------------
[INFO] Building Apache Sling Launchpad Testing 7-SNAPSHOT
[INFO] ------------------------------------------------------------------------
...
Tests run: 529, Failures: 0, Errors: 0, Skipped: 1
...
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 02:08 min
[INFO] ------------------------------------------------------------------------
```

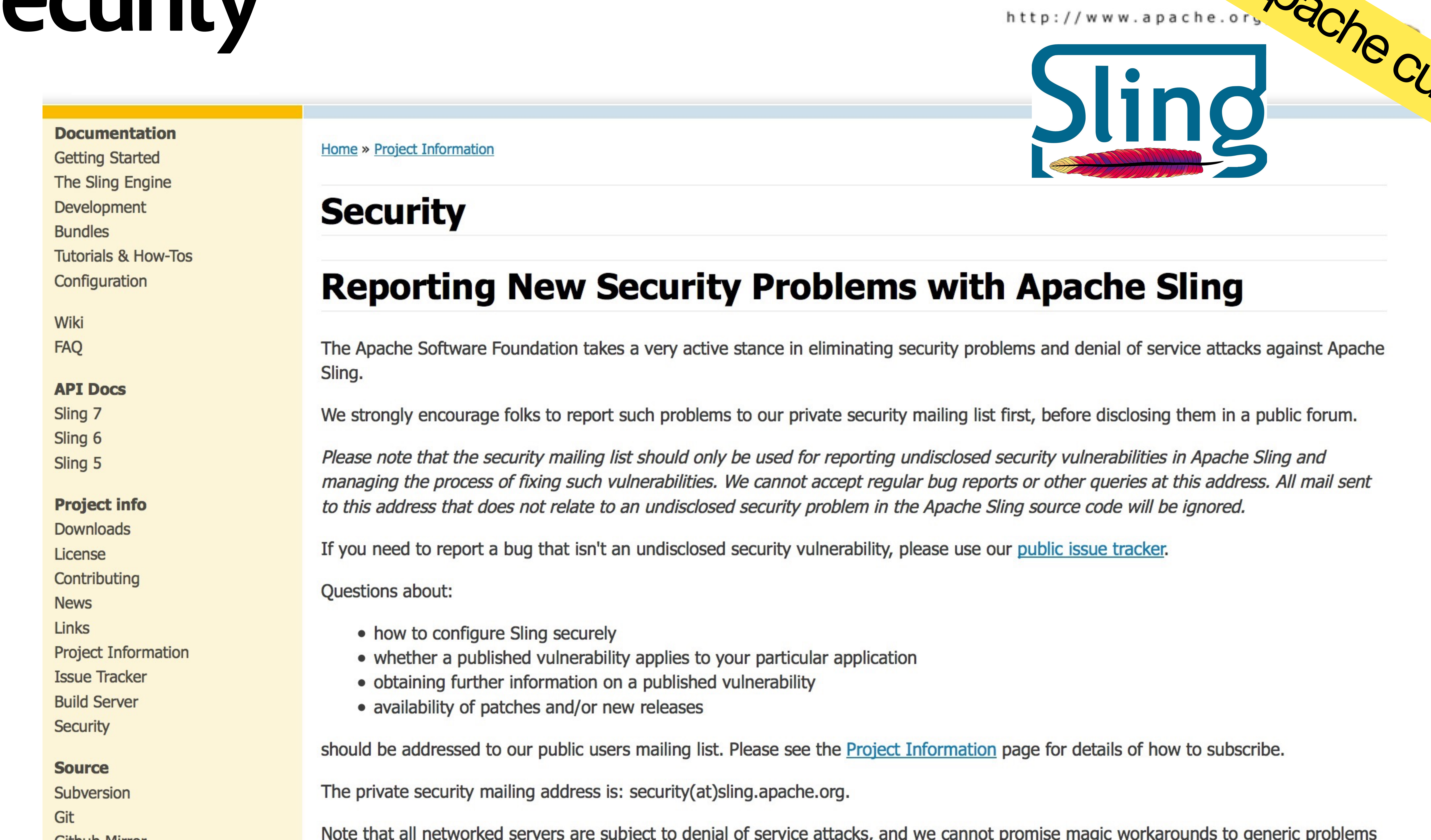
Sling?

Security

the best way to avoid risk
is to do nothing, of
course. Nobody will steal
this picture, but it’s
quite a boring slide.

Security

Apache culture!



Documentation
- Getting Started
- The Sling Engine
- Development
- Bundles
- Tutorials & How-Tos
- Configuration

- Wiki
- FAQ

API Docs
- Sling 7
- Sling 6
- Sling 5

Project info
- Downloads
- License
- Contributing
- News
- Links
- Project Information
- Issue Tracker
- Build Server
- Security

Source
- Subversion
- Git
- Github Mirror

Home » Project Information

Security

Reporting New Security Problems with Apache Sling

The Apache Software Foundation takes a very active stance in eliminating security problems and denial of service attacks against Apache Sling.

We strongly encourage folks to report such problems to our private security mailing list first, before disclosing them in a public forum.

Please note that the security mailing list should only be used for reporting undisclosed security vulnerabilities in Apache Sling and managing the process of fixing such vulnerabilities. We cannot accept regular bug reports or other queries at this address. All mail sent to this address that does not relate to an undisclosed security problem in the Apache Sling source code will be ignored.

If you need to report a bug that isn't an undisclosed security vulnerability, please use our public issue tracker.

Questions about:

- how to configure Sling securely
- whether a published vulnerability applies to your particular application
- obtaining further information on a published vulnerability
- availability of patches and/or new releases

should be addressed to our public users mailing list. Please see the Project Information page for details of how to subscribe.

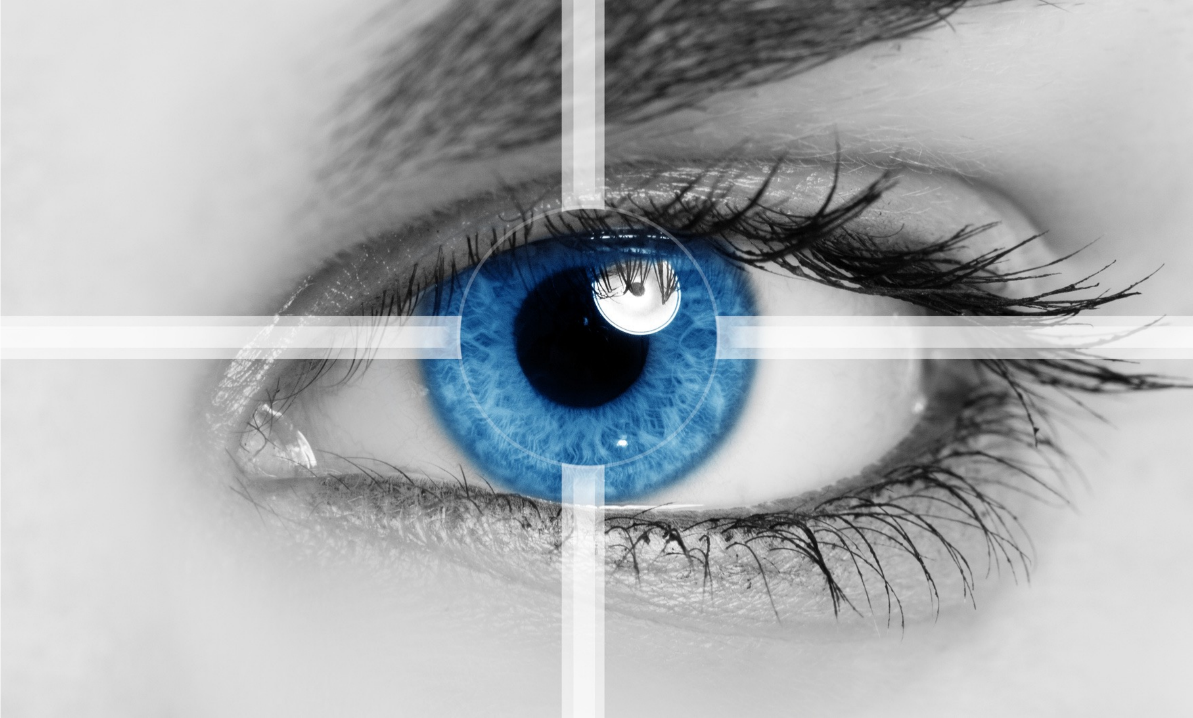
The private security mailing address is: security(at)sling.apache.org.

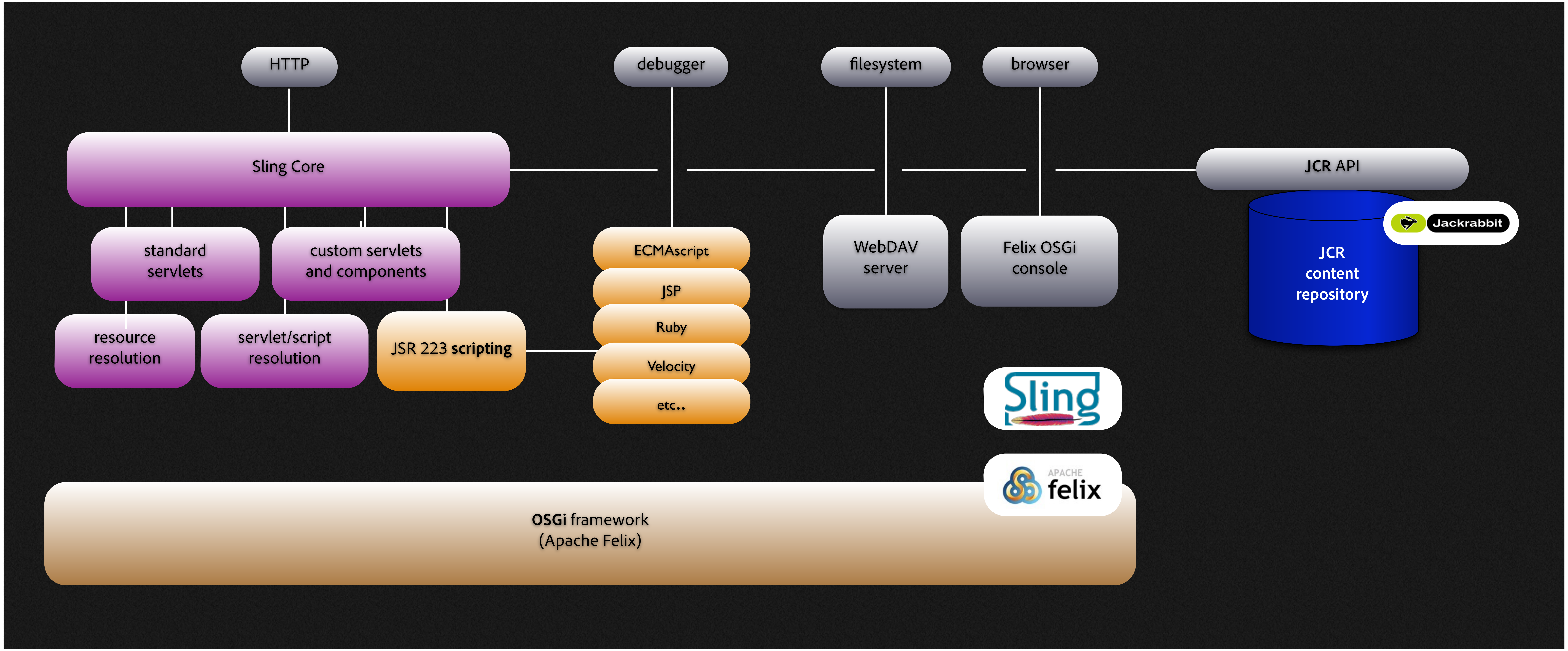
Note that all networked servers are subject to denial of service attacks, and we cannot promise magic workarounds to generic problems

Focus

Sling ?

Focus





scarce resources help to focus!

Backwards compatibility

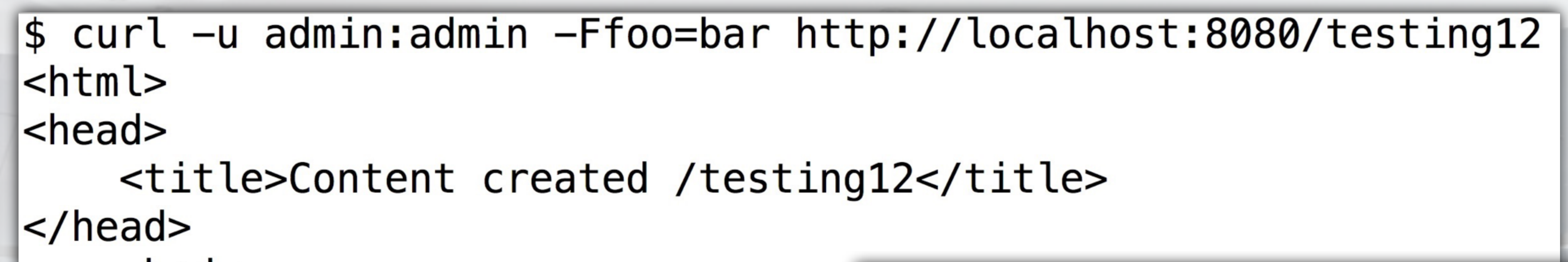
Apache culture!

Transparency

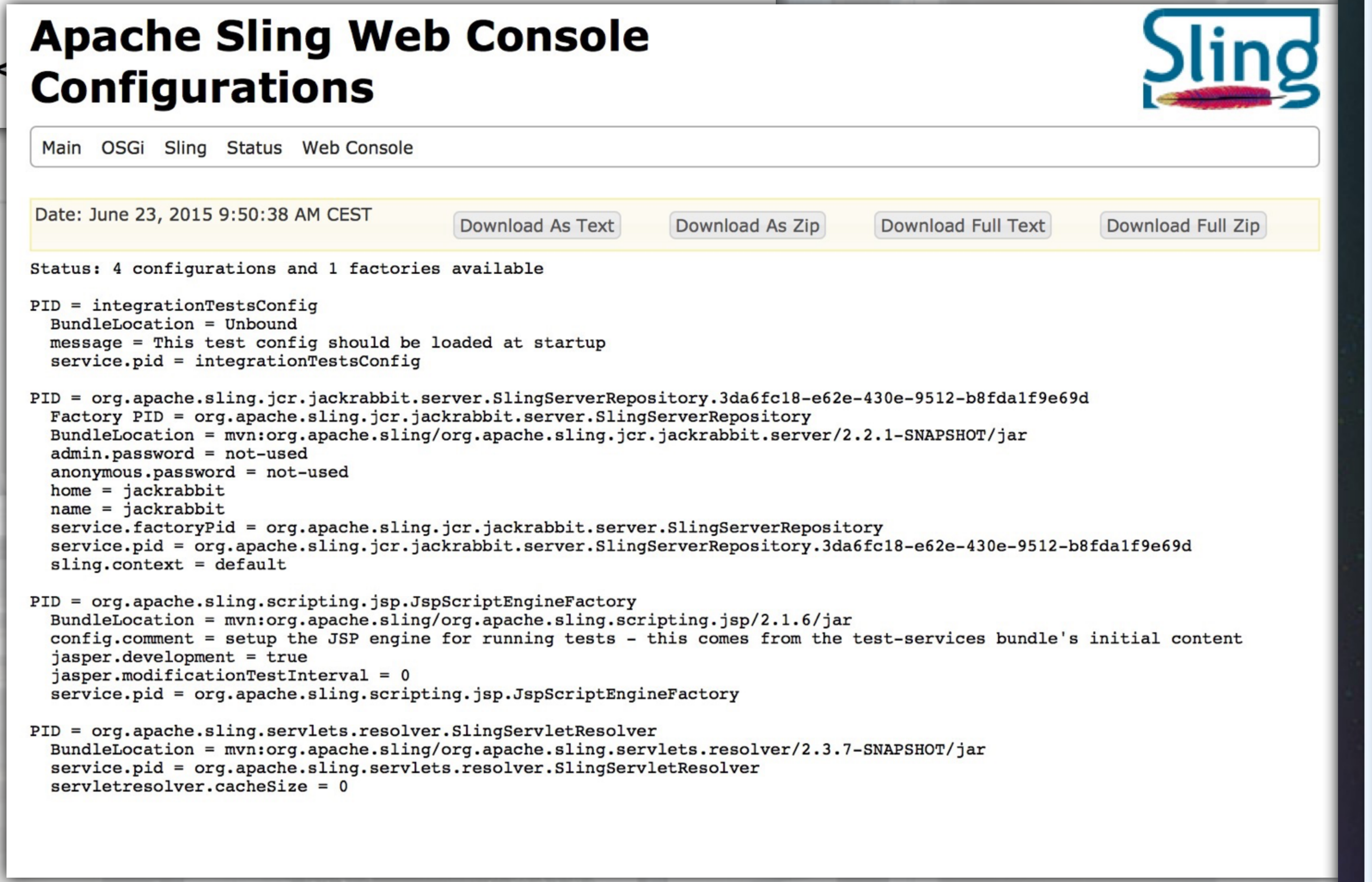
REST culture!

Transparency

REST -> naturally transparent
Good names
Helpful messages
Introspection tools



```
$ curl -u admin:admin -Ffoo=bar http://localhost:8080/testing12
<html>
<head>
    <title>Content created /testing12</title>
</head>
    <body>
    <h1>Content created /testing12<
    <table>
```



Apache Sling Web Console
Configurations

Date: June 23, 2015 9:50:38 AM CEST

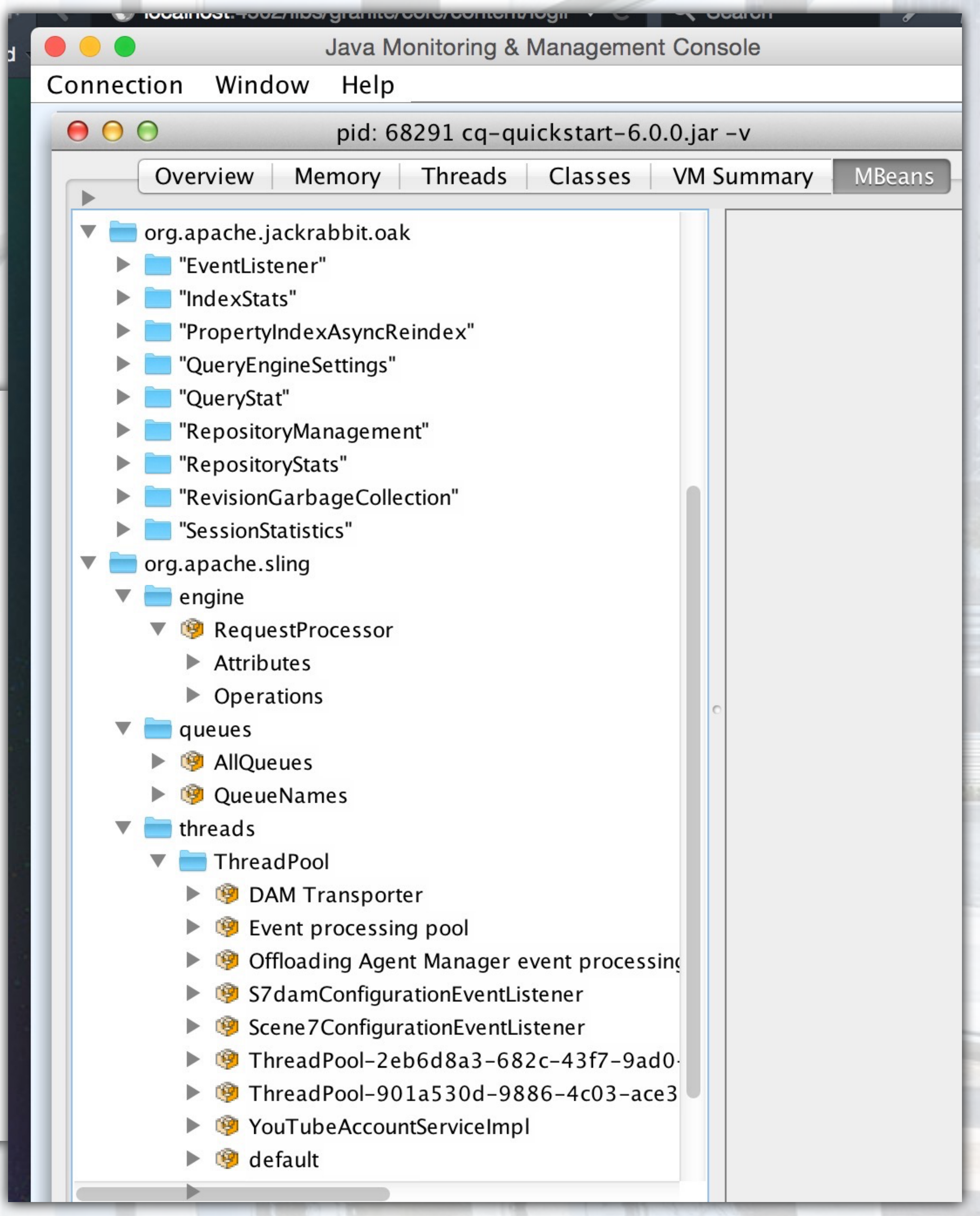
```
Status: 4 configurations and 1 factories available

PID = integrationTestsConfig
  BundleLocation = Unbound
  message = This test config should be loaded at startup
  service.pid = integrationTestsConfig

PID = org.apache.sling.jcr.jackrabbit.server.SlingServerRepository.3da6fc18-e62e-430e-9512-b8fda1f9e69d
  Factory PID = org.apache.sling.jcr.jackrabbit.server.SlingServerRepository
  BundleLocation = mvn:org.apache.sling/org.apache.sling.jcr.jackrabbit.server/2.2.1-SNAPSHOT/jar
  admin.password = not-used
  anonymous.password = not-used
  home = jackrabbit
  name = jackrabbit
  service.factoryPid = org.apache.sling.jcr.jackrabbit.server.SlingServerRepository
  service.pid = org.apache.sling.jcr.jackrabbit.server.SlingServerRepository.3da6fc18-e62e-430e-9512-b8fda1f9e69d
  sling.context = default

PID = org.apache.sling.scripting.jsp.JspScriptEngineFactory
  BundleLocation = mvn:org.apache.sling/org.apache.sling.scripting.jsp/2.1.6/jar
  config.comment = setup the JSP engine for running tests - this comes from the test-services bundle's initial content
  jasper.development = true
  jasper.modificationTestInterval = 0
  service.pid = org.apache.sling.scripting.jsp.JspScriptEngineFactory

PID = org.apache.sling.servlets.resolver.SlingServletResolver
  BundleLocation = mvn:org.apache.sling/org.apache.sling.servlets.resolver/2.3.7-SNAPSHOT/jar
  service.pid = org.apache.sling.servlets.resolver.SlingServletResolver
  servletresolver.cacheSize = 0
```



Java Monitoring & Management Console
Connection
Window
Help
pid: 68291 cq-quickstart-6.0.0.jar -v
Overview
Memory
Threads
Classes
VM Summary
MBeans

```
org.apache.jackrabbit.oak
  "EventListener"
  "IndexStats"
  "PropertyIndexAsyncReindex"
  "QueryEngineSettings"
  "QueryStat"
  "RepositoryManagement"
  "RepositoryStats"
  "RevisionGarbageCollection"
  "SessionStatistics"
org.apache.sling
  engine
    RequestProcessor
      Attributes
      Operations
  queues
    AllQueues
    QueueNames
  threads
    ThreadPool
      DAM Transporter
      Event processing pool
      Offloading Agent Manager event processing
      S7damConfigurationEventListener
      Scene7ConfigurationEventListener
      ThreadPool-2eb6d8a3-682c-43f7-9ad0
      ThreadPool-901a530d-9886-4c03-ace3
      YouTubeAccountServiceImpl
      default
```

Sling ?

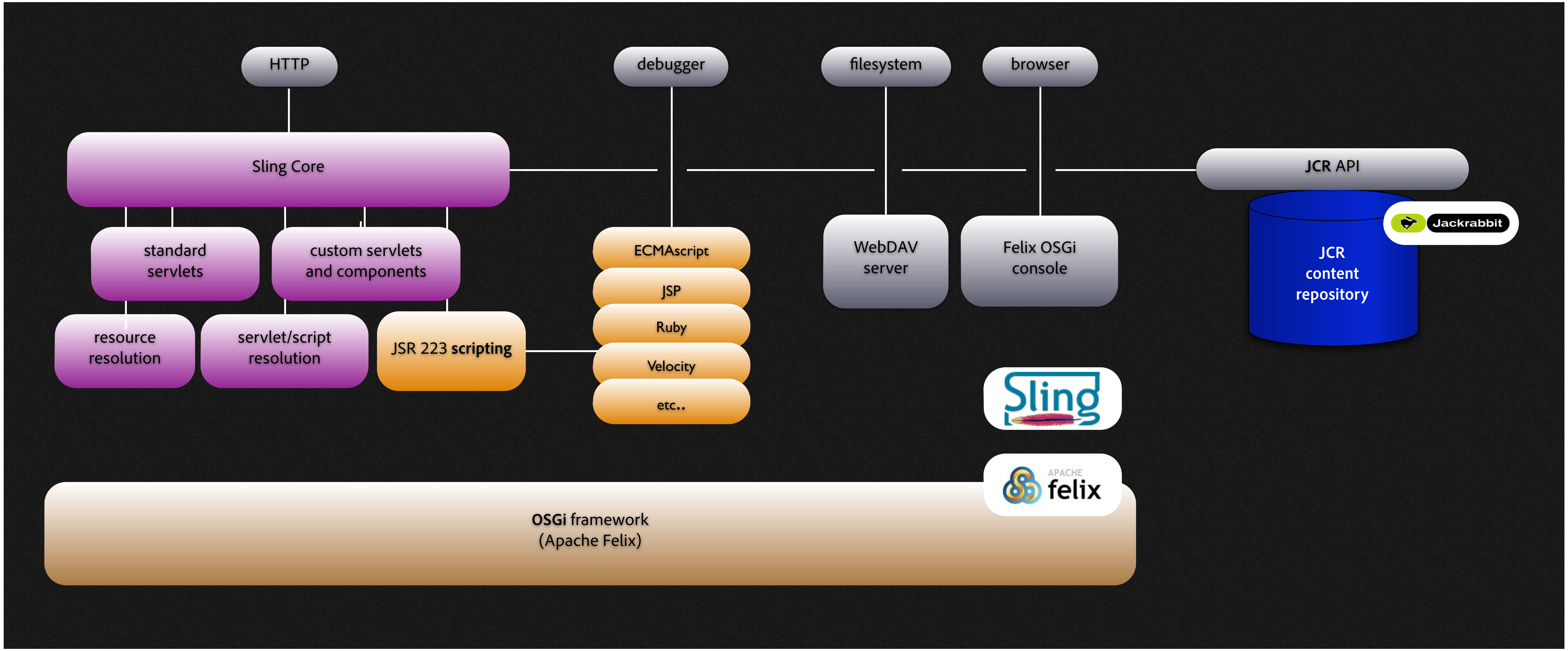
Good names?

```
AbstractResource
AbstractResourceVisitor
AccessSecurityException
Adaptable
AdapterFactory
AdapterManager
AttributableResourceProvider
Authenticator
CompositeValueMap
DeepReadValueMapDecorator
DynamicResourceProvider
ExternalResourceListener
HtmlResponse
SlingAllMethodsServlet
SlingBindings
SlingConstants
SlingException
SlingHttpServletRequest
```

```
LoginException
ModifiableValueMap
ModifiableValueMapDecorator
ModifyingResourceProvider
NoAuthenticationHandlerException
NonExistingResource
ObservationReporter
ObserverConfiguration
OptingServlet
ParametrizableResourceProvider
PersistableValueMap
PersistenceException
PropertyBuilder
ProviderContext
RefreshableResourceProvider
RequestDispatcherOptions
RequestParameter
RequestParameterMap
RequestPathInfo
RequestProgressTracker
RequestUtil
```

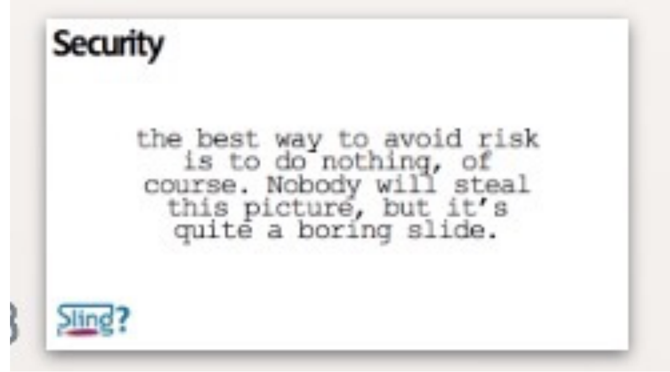
Low coupling

Low coupling



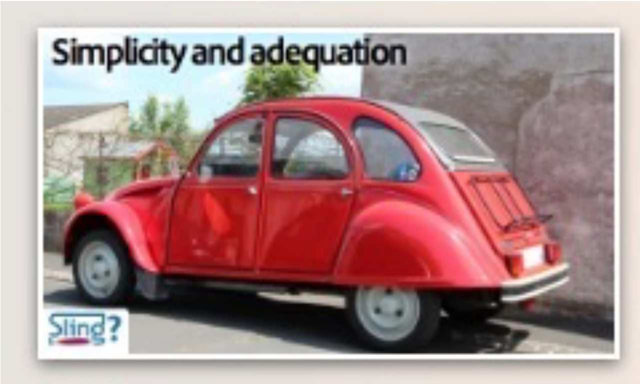
sailboats are very cool but we shouldn't take their high coupling as an example...

quality has
many
dimensions
often hard to measure


Security

the best way to avoid risk is to do nothing, of course. Nobody will steal this picture, but it's quite a boring slide.

Backwards compatibility


Simplicity and adequation

Low coupling

Focus

Robustness

Transparency

Happy Developers!

so, how is

Sling

doing?

I'd say

not bad

thanks to a strong community with a **culture of quality** in design, implementation and testing.

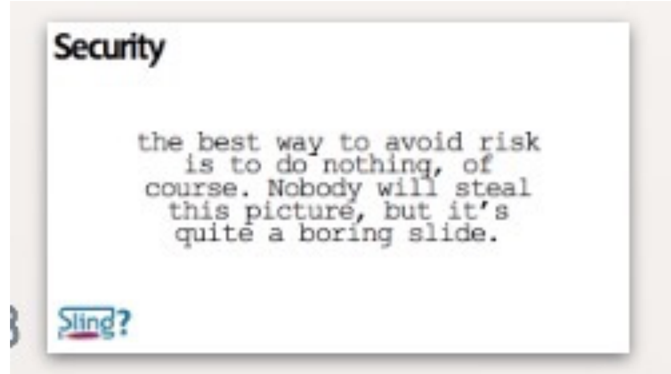
"many eyeballs"

Sling?

Coda

You can't measure it all: quality is multi-dimensional and needs to be *part of your team's culture.*

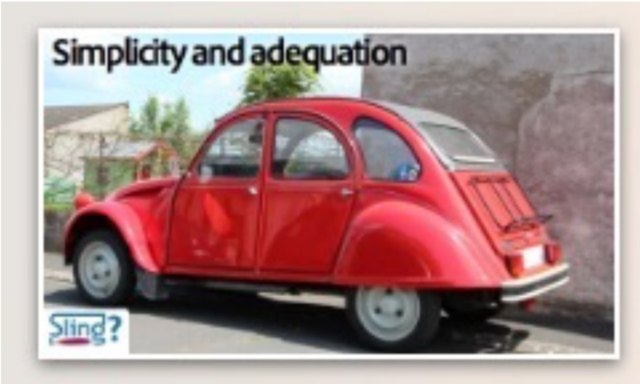
Thinking and reviewing along those axes can help build such a culture.



Security

the best way to avoid risk is to do nothing, of course. Nobody will steal this picture, but it's quite a boring slide.

Backwards compatibility



Simplicity and adequation

Low coupling

Focus

Robustness

Transparency

Happy Developers!

And in the end the only thing that *really* matters is happy users…

"Thoughts on Software Quality" - Bertrand Delacretaz, Principal Scientist, Adobe