

**adaptTo()**

APACHE SLING & FRIENDS TECH MEETUP  
BERLIN, 22-24 SEPTEMBER 2014



Bertrand Delacrétaz - Principal Scientist, Adobe - Apache Sling PMC member  
**Modern Operations with Apache Sling**

@bdelacretaz - [grep.codeconsult.ch](http://grep.codeconsult.ch) - slides revision 2014-09-23

# Modern Operations?



Is Sling DevOps and Cluster friendly?

11 messages in org.apache.incubator.sling-dev 

From	Sent On	Attachments
Ian Boston	Nov 16, 2013 1:58 am	
Chetan Mehrotra	Nov 16, 2013 7:18 am	
Felix Meschberger	Nov 16, 2013 1:06 pm	
Bertrand Delacretaz	Nov 20, 2013 1:07 am	
Ian Boston	Nov 20, 2013 2:06 am	
Felix Meschberger	Nov 20, 2013 2:49 am	

**Subject:** Is Sling DevOps and Cluster friendly?

**From:** Ian Boston (ie...@tfd.co.uk)

**Date:** Nov 16, 2013 1:58:23 am

**List:** org.apache.incubator.sling-dev

Hi,

Reading the thread on Lockback logging which raises configuration files on the filesystem and reading background information on Marathon[1] Borge, Omega[2], Mesos[3] as well as a bit of experience working on much older cluster management systems on MPP systems in the 1990's like Condor and IBM SP1/2s. Mix that with the smaller scale dev ops frameworks like puppet, chef, fabric etc, and it triggered the question in the subject. The references will also give context.





# Continuous Delivery of Apache Sling Applications

Master's Thesis

by

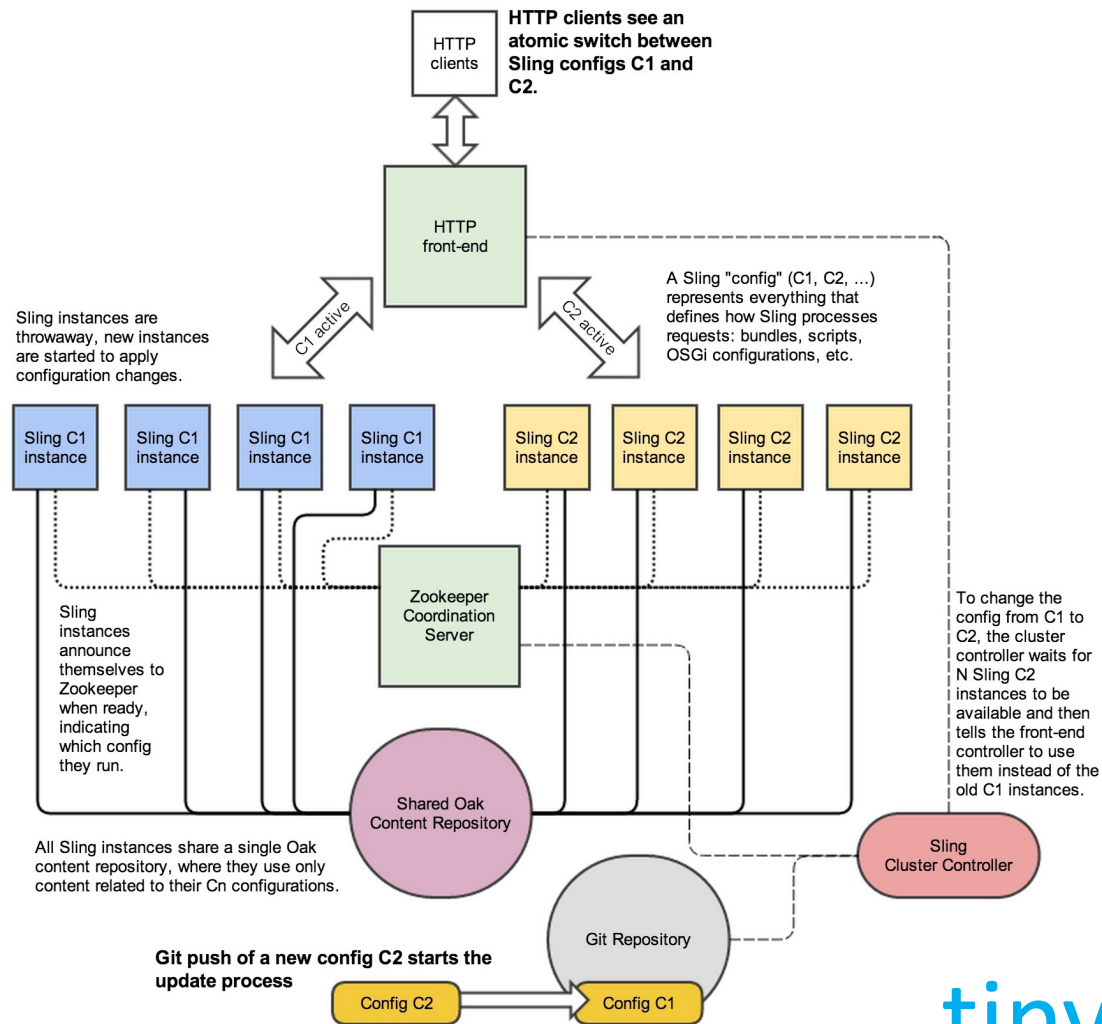
**Artyom Stetsenko**

Presented for the Degree of  
Master of Science  
in  
Computer Science

Supervisors:

**Prof. Willy Zwaenepoel**  
Operating Systems Laboratory  
École Polytechnique Fédérale de Lausanne

**Bertrand Delacrétaz**  
Principal Scientist  
Adobe Research Switzerland



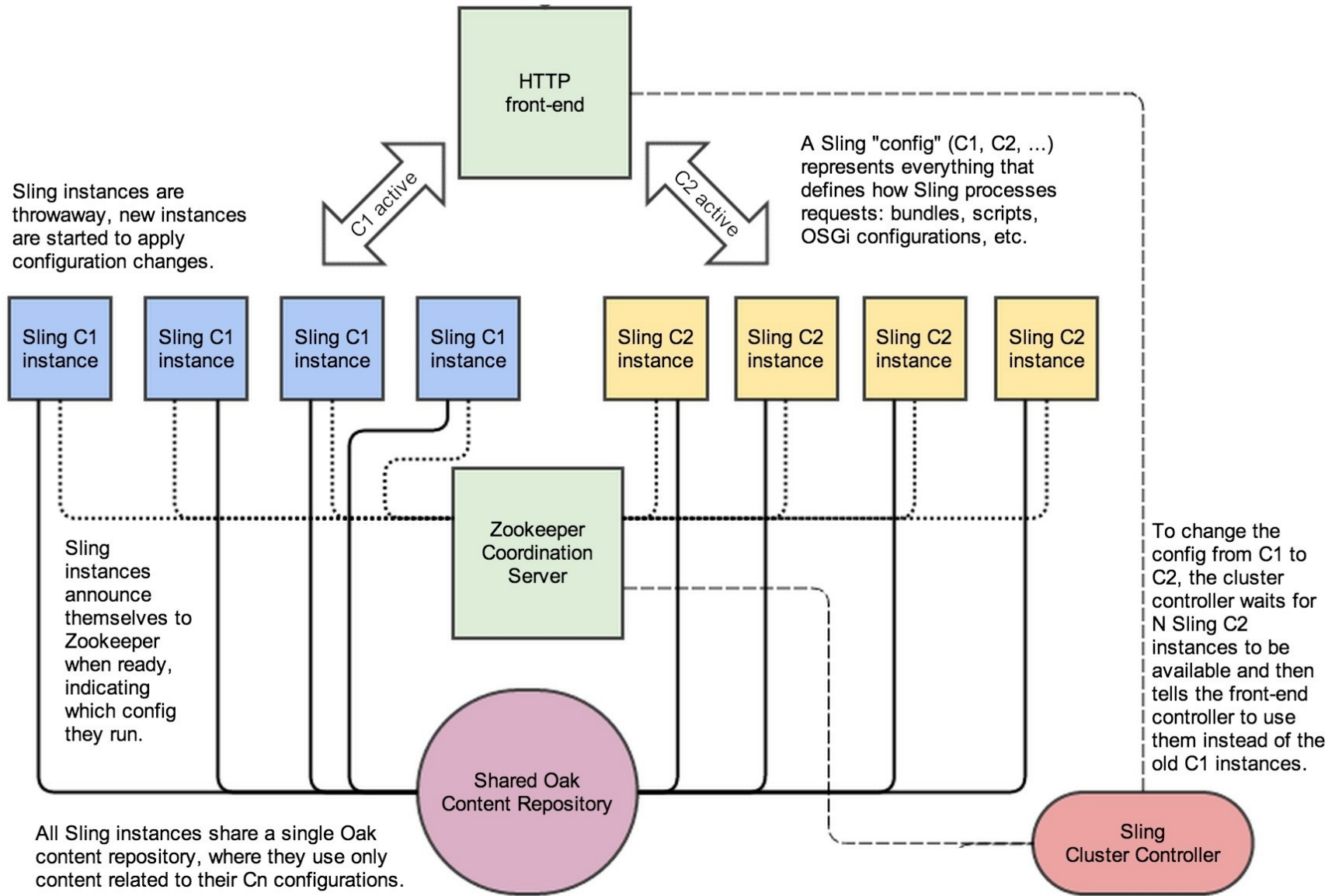
[tinyurl.com/slingops](http://tinyurl.com/slingops)



adapTo()

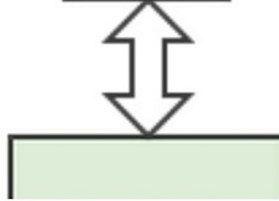
modern operations with Apache Sling - @bdelacretaz - September 2014





HTTP clients

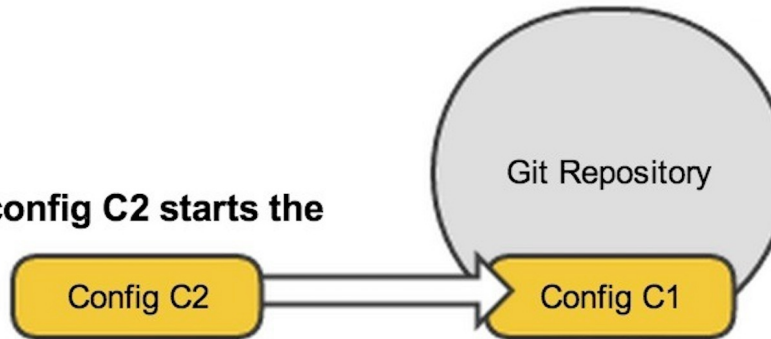
HTTP clients see an atomic switch between Sling configs C1 and C2.



Sling Sling Sling Sling Sling Sling

1..N Sling instances

Git push of a new config C2 starts the update process



# How can Sling help?



Distributed  
workers

Immutable  
services

Git-driven  
deployments

Programmable  
Infrastructure

Stateless  
services

# Modern Operations

First-class  
logs

Metrics

Elastic  
clusters

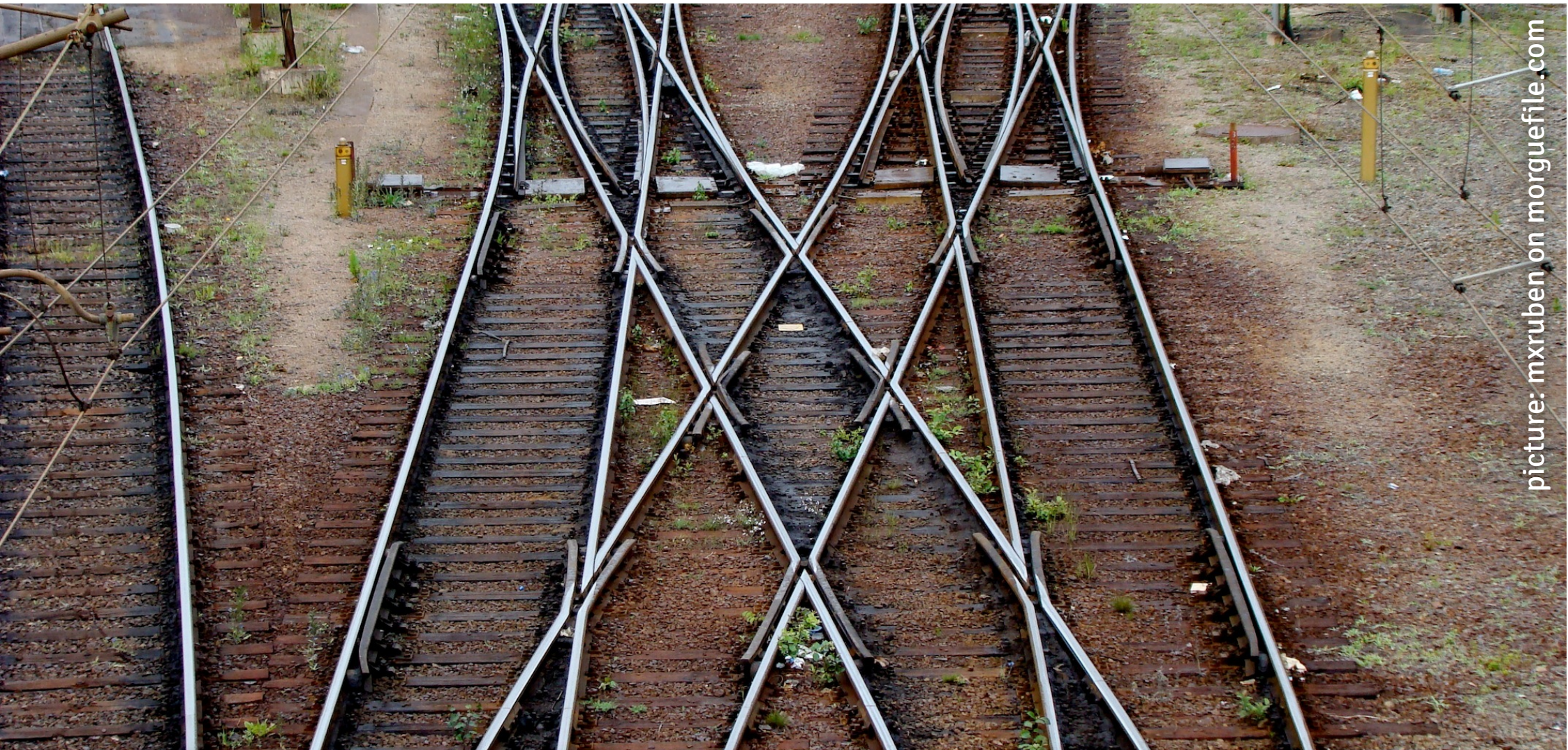
Service  
discovery

Self-healing  
systems

<http://12factor.net/>







picture: mxruben on morguefile.com

We're still exploring...

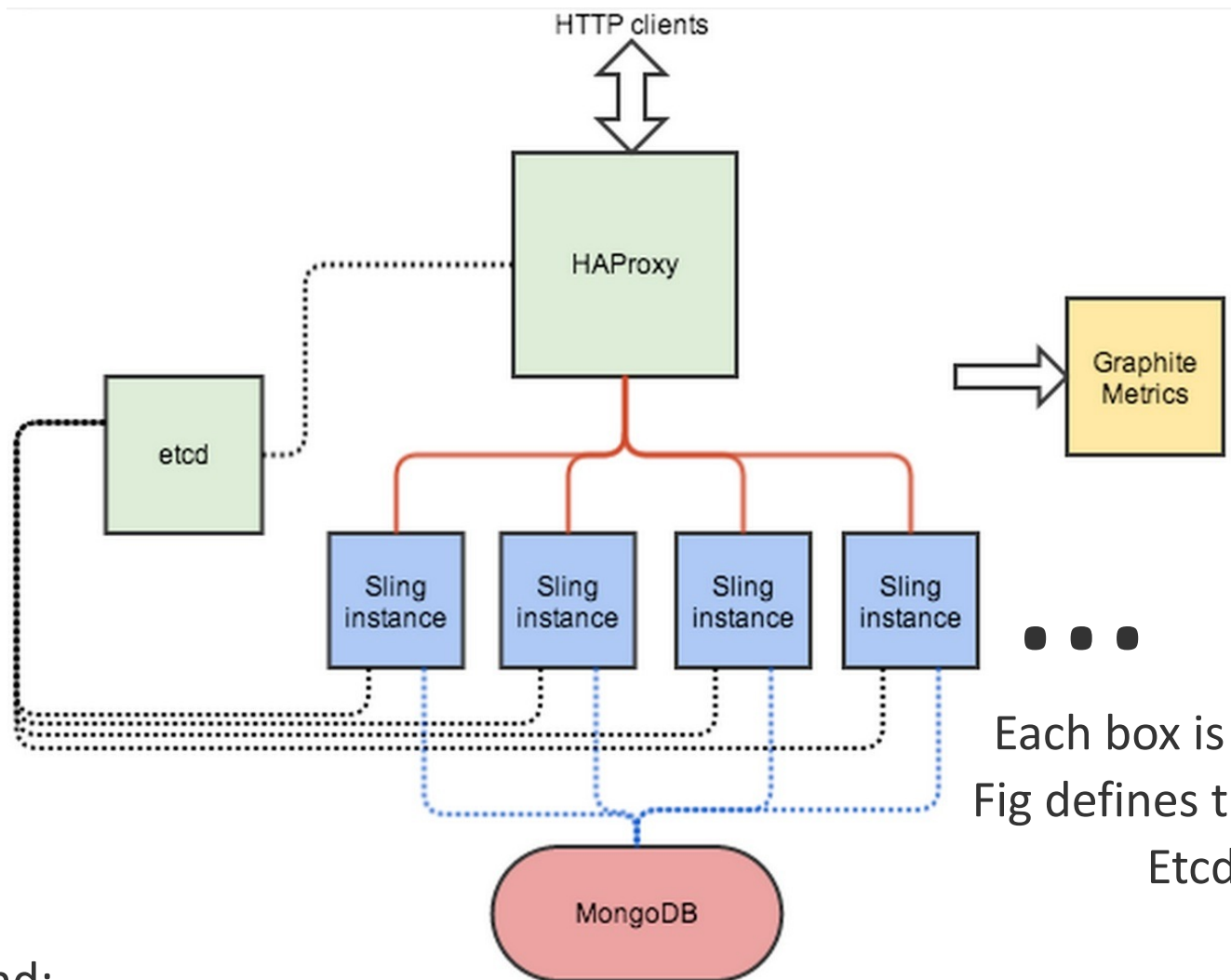


# Let's play with this stuff



**adaptTo()**





Each box is a Docker container  
 Fig defines the cluster topology  
 Etcd used for discovery

Playground:

<https://github.com/bdelacretaz/docker-sling-cluster/>



# Git-driven Sling setup



adaptTo()



```

# Test the crankstart launcher by setting up an HTTP
# server with a few servlets that require specific OSGi configurations

# Default values for our variables
defaults single.path /single
defaults felix.http.jetty.version 2.2.0

# Bootstrap classpath (variables are not supported here)
classpath mvn:org.apache.felix/org.apache.felix.framework/4.4.0
classpath mvn:org.slf4j/org.slf4j-api/1.6.2
classpath mvn:org.apache.sling/org.apache.sling.crankstart.core/0.0.1-SNAPSHOT
classpath mvn:org.apache.sling/org.apache.sling.crankstart.api/0.0.1-SNAPSHOT

# OSGi properties
osgi.property org.osgi.service.http.port ${http.port}
osgi.property org.osgi.framework.storage ${osgi.storage.path}

# Start the framework
start.framework

# Start ConfigAdmin, HTTP service and SCR
bundle mvn:org.apache.felix/org.apache.felix.http.jetty/${felix.http.jetty.version}
bundle mvn:org.apache.felix/org.apache.felix.eventadmin/1.3.2
bundle mvn:org.apache.felix/org.apache.felix.scr/1.8.2
bundle mvn:org.apache.felix/org.apache.felix.metatype/1.0.10
bundle mvn:org.apache.sling/org.apache.sling.commons.osgi/2.2.1-SNAPSHOT
bundle mvn:org.apache.sling/org.apache.sling.commons.log/2.1.2
bundle mvn:org.apache.felix/org.apache.felix.configadmin/1.6.0
bundle mvn:org.apache.felix/org.apache.felix.webconsole/3.1.6

# The crankstart.api.fragment bundle makes the crankstart.api package available
# to bundles, required for bundles to provide crankstart extension commands like
# the test.system.property command below
bundle mvn:org.apache.sling/org.apache.sling.crankstart.api.fragment/0.0.1-SNAPSHOT
bundle mvn:org.apache.sling/org.apache.sling.crankstart.test.services/0.0.1-SNAPSHOT

# Test our Sling extension commands, that add a bundle via the Sling installer
# (which requires commons.json and jcr-wrapper)
bundle mvn:org.apache.sling/org.apache.sling.installer.core/3.5.0
bundle mvn:org.apache.sling/org.apache.sling.commons.json/2.0.6
bundle mvn:org.apache.sling/org.apache.sling.jcr.jcr-wrapper/2.0.0
bundle mvn:org.apache.sling/org.apache.sling.sling.extensions/0.0.1-SNAPSHOT
bundle mvn:commons-io/commons-io/2.4

# Now start our bundles
start.all.bundles

# OSGi configs that activate our test servlets
config org.apache.sling.crankstart.testservices.SingleConfigServlet
path=${single.path}
message=doesn't matter

config.factory org.apache.sling.crankstart.testservices.ConfigFactoryServlet
CRANKSTART_CONFIG_ID=some.unique.ID
path=/foo
message=Not used

config.factory org.apache.sling.crankstart.testservices.ConfigFactoryServlet
path=/bar/test
message=Not used

# Test Felix format configs
config felix.format.test FORMAT:felix.config
mongouri="mongodb://localhost:27017"
service.ranking.launcher.test="54321"
array=["foo", "bar.from.launcher.test"]

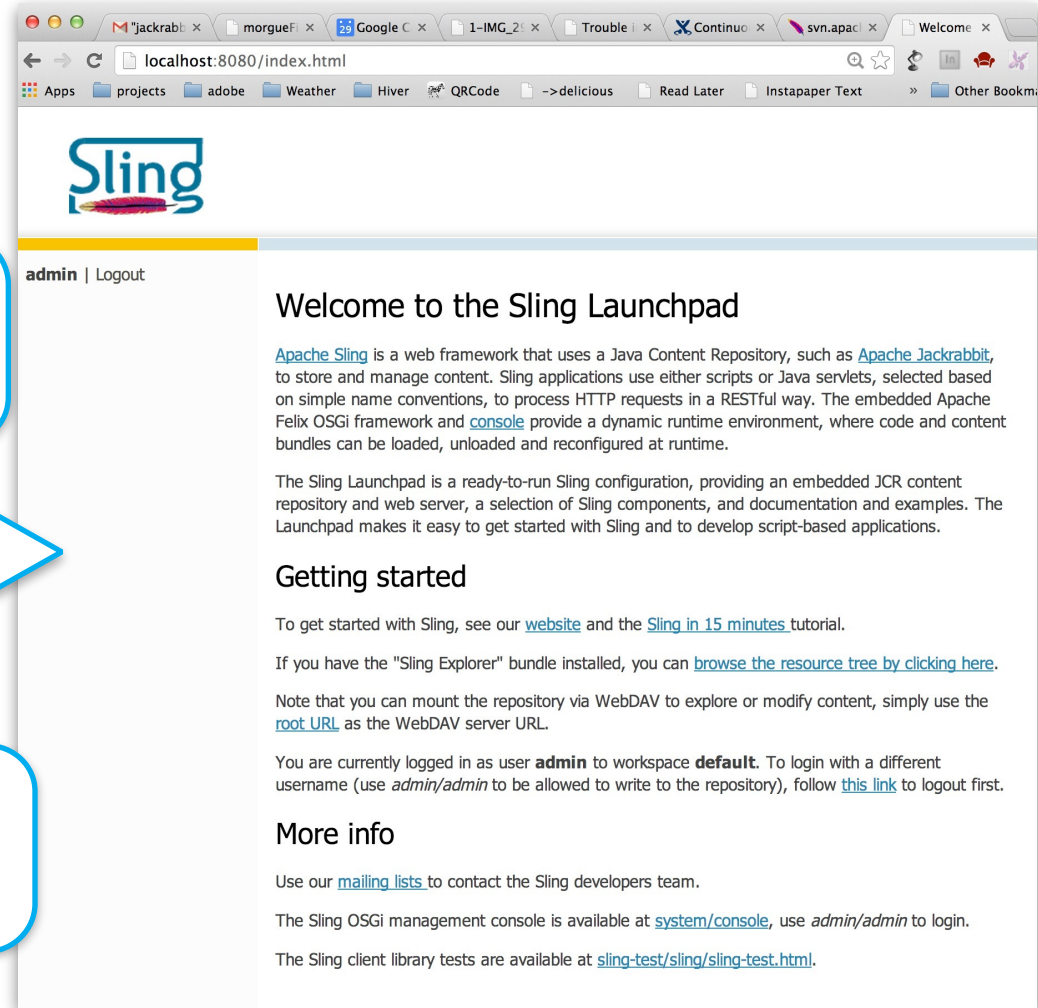
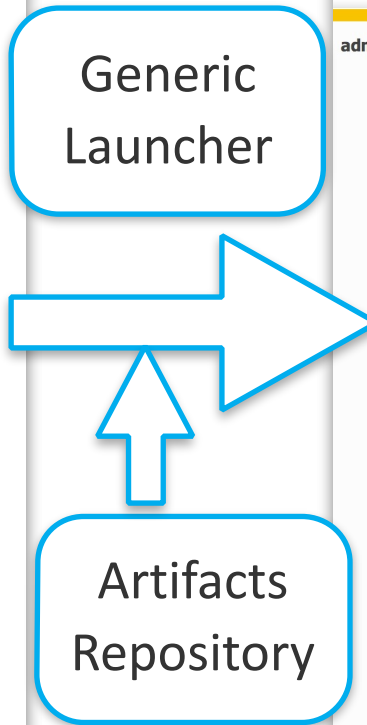
config empty.config.should.work FORMAT:felix.config

# Test an extension command provided by our test-services bundle
test.system.property the.test.system.property was set by test-services bundle

# Prepare additional resources for the Sling installer
sling.installer.resource mvn:org.apache.sling/org.apache.sling.junit.core/1.0.8
sling.installer.resource mvn:org.apache.sling/org.apache.sling.commons.mime/2.1.4
sling.installer.resource mvn:org.apache.sling/org.apache.sling.settings/1.3.0

# And register the installer resources
sling.installer.register crankstart

```



## Sling Crankstart Launcher: Single text file fully defines a Sling instance



```
# Set default values for our variables
# Must set these three on command line
defaults port 80
defaults sling_home /tmp/SLING-HOME
defaults mongo mongodb://MISSING:0
```

```
# Set OSGi framework properties
osgi.property org.osgi.service.http.port ${port}
osgi.property sling.home ${sling_home}
osgi.property org.osgi.framework.storage ${sling_home}/osgi.framework.s
osgi.property org.apache.sling.commons.log.level INFO
```





```
# Once OSGi properties are set, start the framework
start.framework

# Load the Configuration Admin bundle and start it
bundle mvn:org.apache.felix/org.apache.felix.configadmin/1.6.0
start.all.bundles

# Once Configuration Admin is loaded, we can set configurations
config org.apache.sling.installer.provider.jcr.impl.JcrInstaller
    sling.jcrinstall.search.path = /sling-cfg/${config}/apps:200
    sling.jcrinstall.search.path = /sling-cfg/${config}/libs:100
```



```
# Install bundles
# mvn: protocol can be used.
bundle mvn:org.apache.felix/org.apache.felix.http.jetty/2.2.2
bundle mvn:org.slf4j/slf4j-api/1.7.6
bundle mvn:org.apache.sling/org.apache.sling.commons.log/4.0.0
bundle mvn:org.apache.sling/org.apache.sling.commons.logservice/1.0.2
bundle mvn:org.slf4j/jcl-over-slf4j/1.7.6
bundle mvn:org.slf4j/log4j-over-slf4j/1.7.6

# If this is a warmup run, exit
log warmup=${warmup}
exit.iftrue ${warmup}
```





```
# Test the crankstart launcher by setting up an HTTP
# server with a few servlets that require specific OSGi configurations

# Default values for our variables
defaults single.path /single
defaults felix.http.jetty.version 2.2.0

# Bootstrap classpath (variables are not supported here)
classpath mvn:org.apache.felix/org.apache.felix.framework/4.4.0
classpath mvn:org.slf4j/org.slf4j-api/1.6.2
classpath mvn:org.apache.sling/org.apache.sling.crankstart.core/0.0.1-SNAPSHOT
classpath mvn:org.apache.sling/org.apache.sling.crankstart.api/0.0.1-SNAPSHOT

# OSGi properties
osgi.property org.osgi.service.http.port ${http.port}
osgi.property org.osgi.framework.storage ${osgi.storage.path}

# Start the framework
start.framework

# Start ConfigAdmin, HTTP service and SCR
bundle mvn:org.apache.felix/org.apache.felix.http.jetty/${felix.http.jetty.version}
bundle mvn:org.apache.felix/org.apache.felix.eventadmin/1.3.2
bundle mvn:org.apache.felix/org.apache.felix.scr/1.8.2
bundle mvn:org.apache.felix/org.apache.felix.metatype/1.0.10
bundle mvn:org.apache.sling/org.apache.sling.commons.osgi/2.2.1-SNAPSHOT
bundle mvn:org.apache.sling/org.apache.sling.commons.log/2.1.2
bundle mvn:org.apache.felix/org.apache.felix.configadmin/1.6.0
bundle mvn:org.apache.felix/org.apache.felix.webconsole/3.1.6

# The crankstart.api.fragment bundle makes the crankstart.api package available
# to bundles, required for bundles to provide crankstart extension commands like
# the test.system.property command below
bundle mvn:org.apache.sling/org.apache.sling.crankstart.api.fragment/0.0.1-SNAPSHOT
bundle mvn:org.apache.sling/org.apache.sling.crankstart.test.services/0.0.1-SNAPSHOT

# Test our Sling extension commands, that add a bundle via the Sling installer
# (which requires commons.json and jcr-wrapper)
bundle mvn:org.apache.sling/org.apache.sling.installer.core/3.5.0
bundle mvn:org.apache.sling/org.apache.sling.commons.json/2.0.6
bundle mvn:org.apache.sling/org.apache.sling.jcr.jcr-wrapper/2.0.0
bundle mvn:org.apache.sling/org.apache.sling.sling.extensions/0.0.1-SNAPSHOT
bundle mvn:commons-io/commons-io/2.4

# Now start our bundles
start.all.bundles

# OSGi configs that activate our test servlets
config org.apache.sling.crankstart.testservices.SingleConfigServlet
path=${single.path}
message=doesn't matter

config.factory org.apache.sling.crankstart.testservices.ConfigFactoryServlet
CRANKSTART_CONFIG_ID=some.unique.ID
path=/foo
message=Not used

config.factory org.apache.sling.crankstart.testservices.ConfigFactoryServlet
path=/bar/test
message=Not used

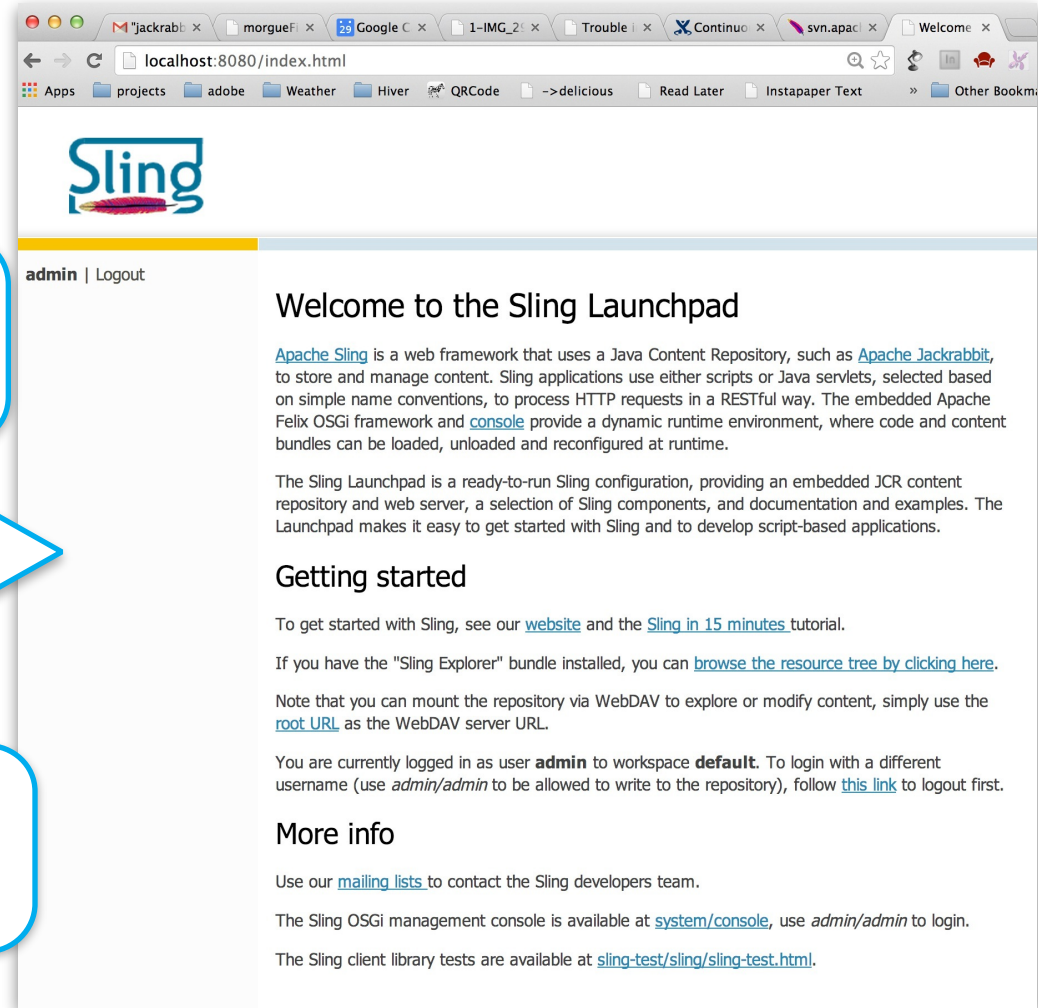
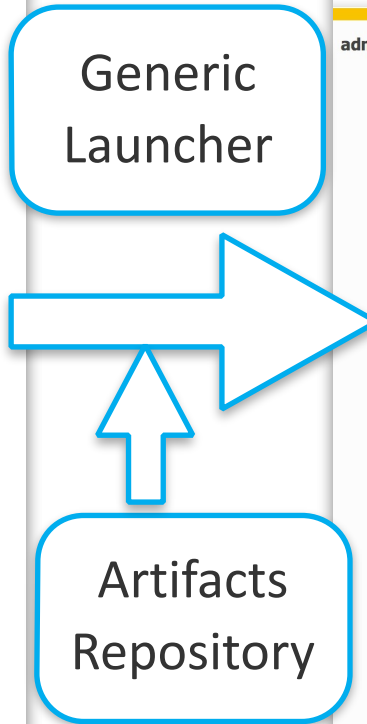
# Test Felix format configs
config felix.format.test FORMAT:felix.config
mongouri="mongodb://localhost:27017"
service.ranking.launcher.test="I54321"
array=["foo", "bar.from.launcher.test"]

config empty.config.should.work FORMAT:felix.config

# Test an extension command provided by our test-services bundle
test.system.property the.test.system.property was set by test-services bundle

# Prepare additional resources for the Sling installer
sling.installer.resource mvn:org.apache.sling/org.apache.sling.junit.core/1.0.8
sling.installer.resource mvn:org.apache.sling/org.apache.sling.commons.mime/2.1.4
sling.installer.resource mvn:org.apache.sling/org.apache.sling.settings/1.3.0

# And register the installer resources
sling.installer.register crankstart
```



# Git-driven Sling setup!



Distributed workers  
Immutable services  
Git-driven deployments  
Programmable Infrastructure  
Stateless services  
Elastic clusters  
Service discovery  
Self-healing systems  
First-class logs  
Metrics  
Experimenting with

# Modern Operations

<http://12factor.net/>



# Docker and Fig

## start.sh (excerpts)

```
# Create etcd announcer config
MY_IP=$(grep $HOSTNAME /etc/hosts | cut -f1)
cat > /tmp/sling-configs/ch.x42.sling.etcd.EtcdAn
etcd.url=http://{ETCD}/v2/keys/http/backends/sli
sling.host=${MY_IP}
sling.port=80
interval.seconds=10
ttl.seconds=30
EOF
```

Service discovery  
via Docker links  
(env. variables)

```
java \  
-Dmongo=$MONGO \  
-Dwarmup=$WARMUP \  
-Dorg.ops4j.pax.url.mvn.localRepository=./tmp/mave  
$REPO \  
-jar /sling/org.apache.sling.crankstart.launcher.jar \  
/sling/crankstart.txt
```

## Dockerfile

```
FROM centos:centos6
MAINTAINER bdelacretaz@apache.org

RUN yum clean all
RUN yum -y update
RUN yum install -y java-1.7.0-openjdk-devel.x86_64

ADD fsroot /

# Create the Sling state during Docker image build
# (TODO remove Sling id file?)
RUN /bin/bash /start.sh true

# Run Sling from the state created during image build
CMD /bin/bash /start.sh false
```



mongo:

image: "mongo:2.6"

entrypoint: /usr/local/bin/mongod

# TODO getting "Insufficient free space for journal files" sometimes

command: --nojournal

ports:

- "27017"

sling:

build: sling

ports:

- "80"

links:

- mongo:mongo

- etcd:etcd

- graphite:graphite

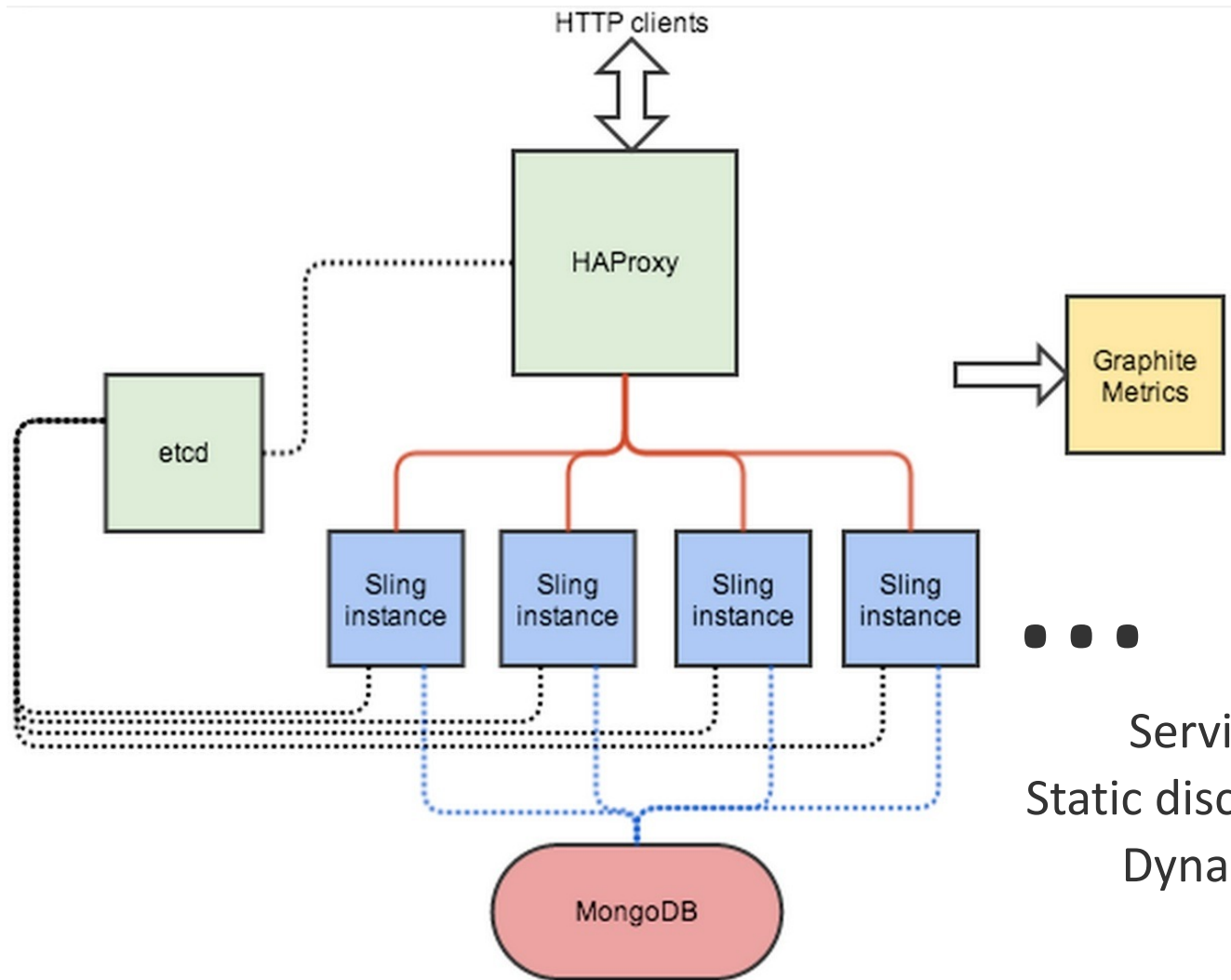
fig.yml excerpts

fig up to start

fig scale sling=4 to scale

fig.yml defines the cluster





Services launched by fig  
 Static discovery: Docker links  
 Dynamic discovery: etcd



<https://github.com/bdelacretaz/docker-sling-cluster/>



Distributed workers  
Immutable services  
Git-driven deployments  
Programmable Infrastructure  
Stateless services  
Elastic clusters  
Service discovery  
Self-healing systems  
First-class logs  
Metrics  
Experimenting with

# Modern Operations

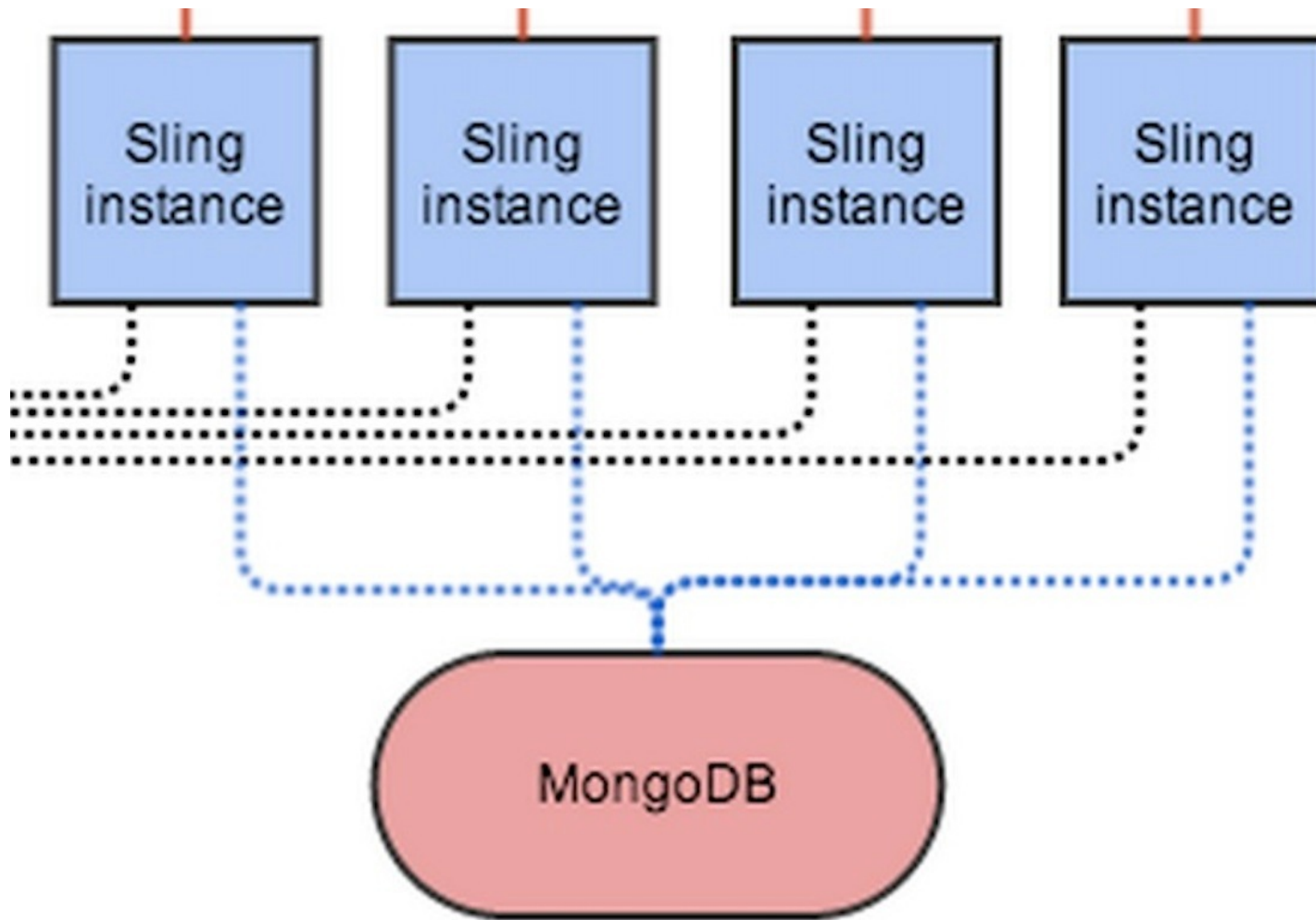
<http://12factor.net/>



# So, what's missing?







Get configs from environment variables.  
Reconnect without interruption if backend changes.

## Dynamic service discovery needs some work

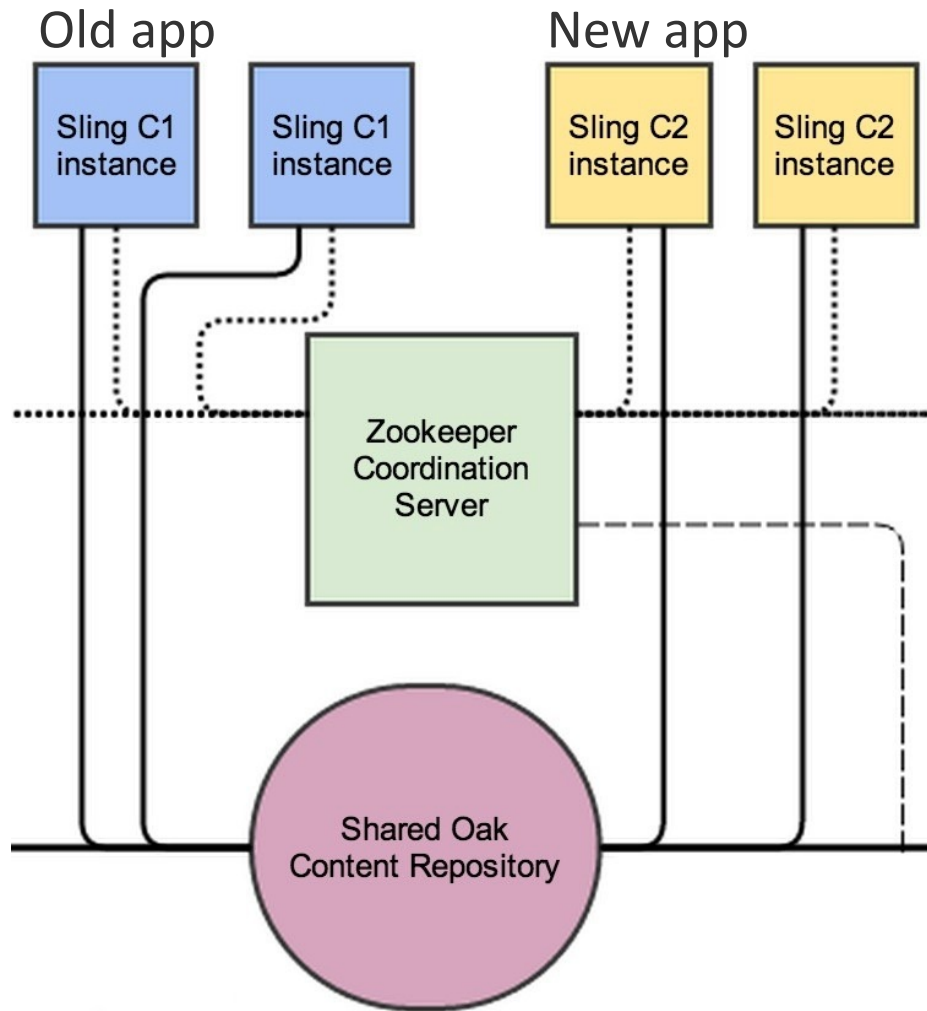


```
// Code from Sling's NodeTypeBasedRenderingTest
uploadTestScript(scriptPath, "nodetype-and-path.esp", "html.esp");

// next request might hit a different Sling instance which
// does not see the changed content yet
String content = getContent(testNodeUrl + ".html", CONTENT_TYPE_HTML);
```

## Eventual consistency impacts applications





While moving from C1 to C2, the same content might be exposed to two different versions of your application.

## Continuous deployment requires interoperability

modern operations with Apache Sling - @bdelacretaz - September 2014



adaptTo()



Distributed workers  
Immutable services  
Git-driven deployments  
Programmable Infrastructure  
Stateless services  
Elastic clusters  
Service discovery  
Self-healing systems  
First-class logs  
Metrics  
Experimenting with

# Modern Operations

<http://12factor.net/>

Not too bad...the devil is in the details!

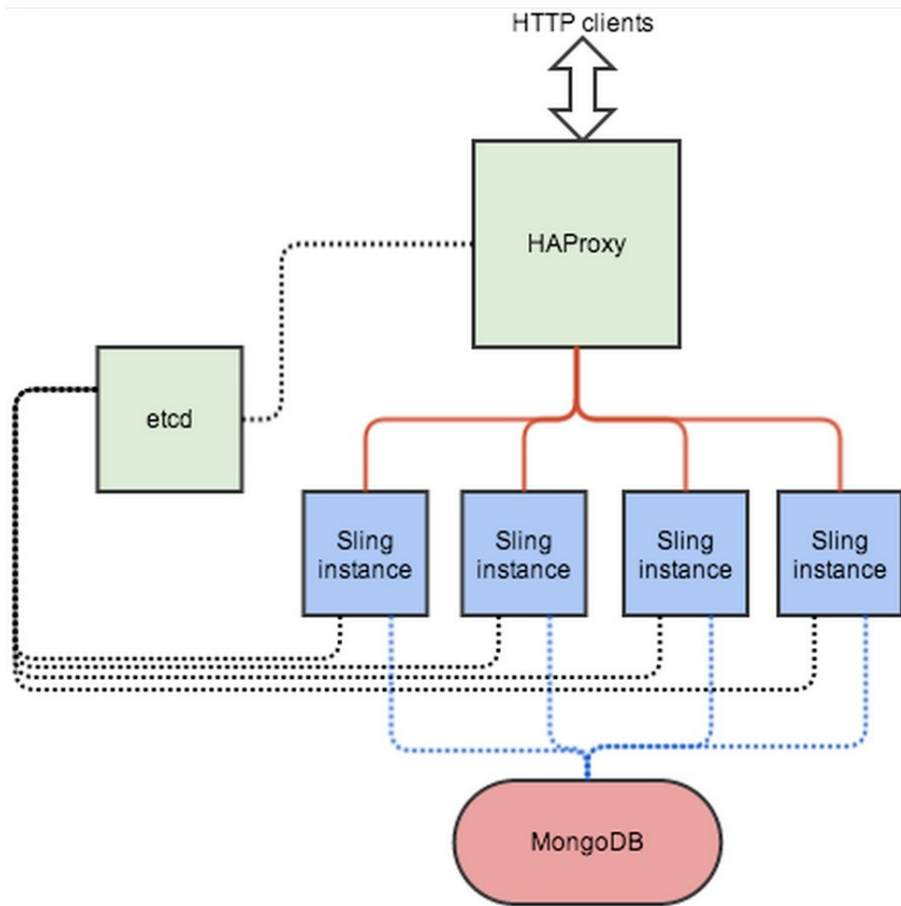


adaptTo()

modern operations with Apache Sling - @bdelacretaz - September 2014



# Conclusions



No major problems with Sling in modern operations.

Runtime assembly might help.

Dynamic backend discovery needs testing, OSGi helps.

Programmable, elastic, dynamic deployments become the norm.

Applications might need some work: eventual consistency, interoperability of versions.

<https://github.com/bdelacretaz/docker-sling-cluster/>

[tinyurl.com/slingops](http://tinyurl.com/slingops)

