infobip

PUBLIC API INFRASTRUCTURE EVOLUTION

Denis Ćutić

Friday, October 19, 2018

Messaging Service Provider

API first

1000 HTTP endpoints

20 teams

100M request/day

Public API infrastructure evolution

Approaches

Pitfalls

Lessons learnt

Public API infrastructure evolution

REST-ish endpoints

Public API infrastructure evolution

Long term support

Public API infrastructure evolution

Backwards compatible

https://www.bu.edu/prlab/files/2017/11/2-1.png

Public API infrastructure evolution

Scaleable

Public API infrastructure evolution

Secure

Public API infrastructure evolution

@Infobip

~ since 2004 ~

Public API infrastructure evolution

Development

Security

Management

Monitoring

....

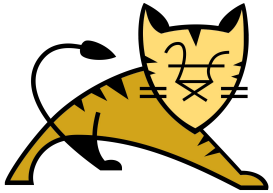
Public API infrastructure evolution

SMS API

Accounts API

...

Public API infrastructure evolution



maven

JUnit 5

Public API infrastructure evolution

maven
git
Java
JUnit 5
JFrog Artifactory
HAPROXY
docker
graylog
New Relic
Groovy
elastic
graphite
Grafana

The story...


https://c1.staticflickr.com/4/3696/33195910630_06350cc953_b.jpg

http://pngimg.com/uploads/stone/stone_PNG13612.png

https://upload.wikimedia.org/wikipedia/commons/d/db/Margaret_Hamilton_-_restoration.jpg

Breaking the monolith appart

Http API service

https://www.bu.edu/prlab/files/2017/11/2-1.png

Lesson #1

Don’t ever do this

```
if (this) {
    do something
} else if (that) {
    do something similar
} else if (friend) {
    be nice
} else if (foo) {
    be rude
} else if (boss) {
    be extra nice
}
```

Custom functionality
Custom endpoint

Company is growing...

API microservice

New technologies

Shut down legacy system

Lesson #2

Legacy public APIs (almost) never die

True legacy never dies

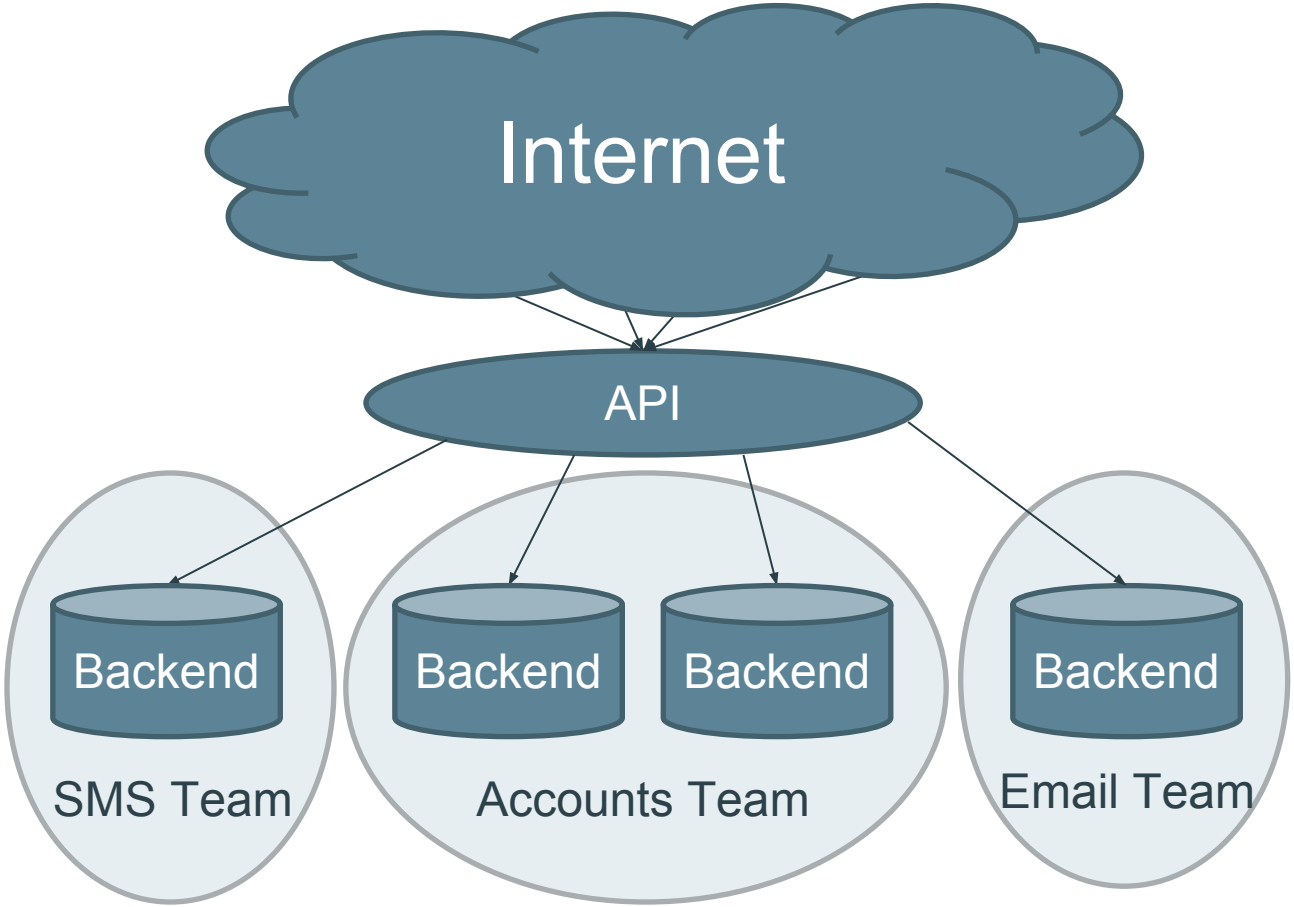
Lesson #3

Always look ahead

http://static.tvtropes.org/pmwiki/pub/images/rake-take_the-simpsons_951.jpg

Great service...

...that doesn’t scale


Internet
API
Backend
Backend
Backend
Backend
SMS Team
Accounts Team
Email Team

API team intervention

Slow

Does not scale

Easy deployment

Consistent API

```
POST /sendsms
{
    accountId: 1
    …
}
```

```
PUT /1/sms
{
    clientId: 1
    …
}
```

```
PUT /sms/1/send
{
    customerId: 1
    …
}
```

Backend team

Faster

Consistency not guaranteed

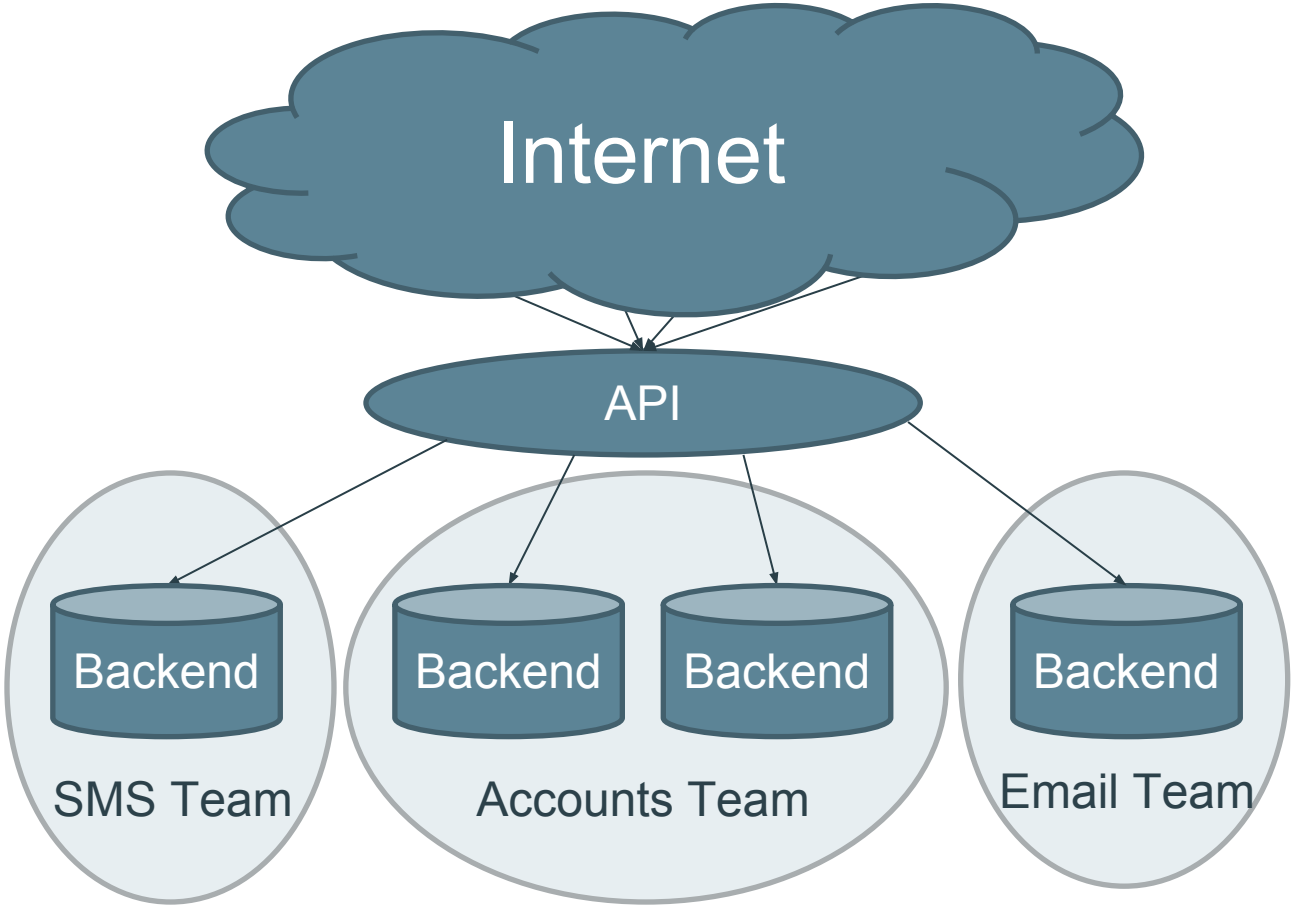
Deployment issues

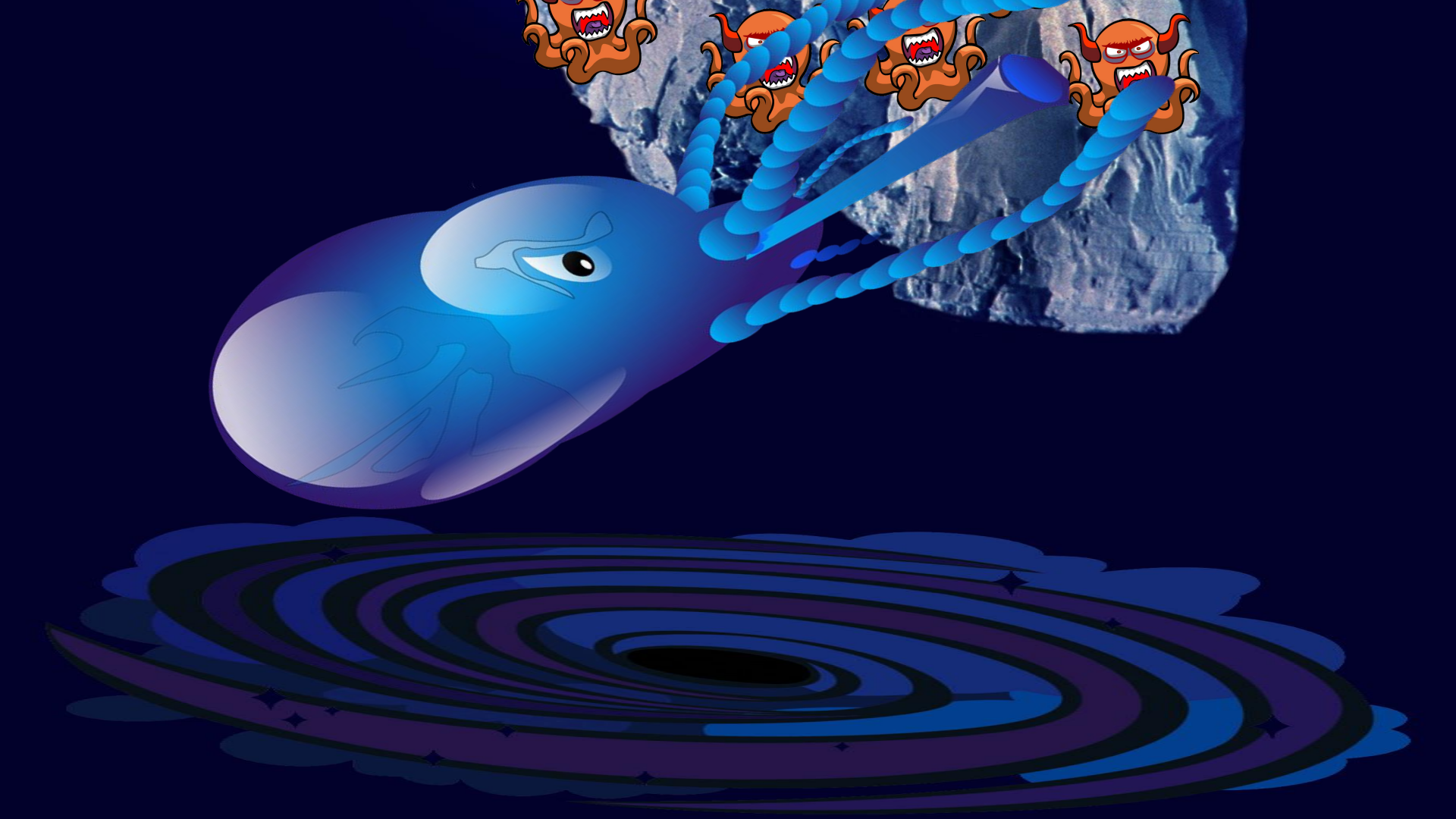
Team A deploys V5

Team B deploys V6

Team A reverts to V4

Team B is !@#$%&

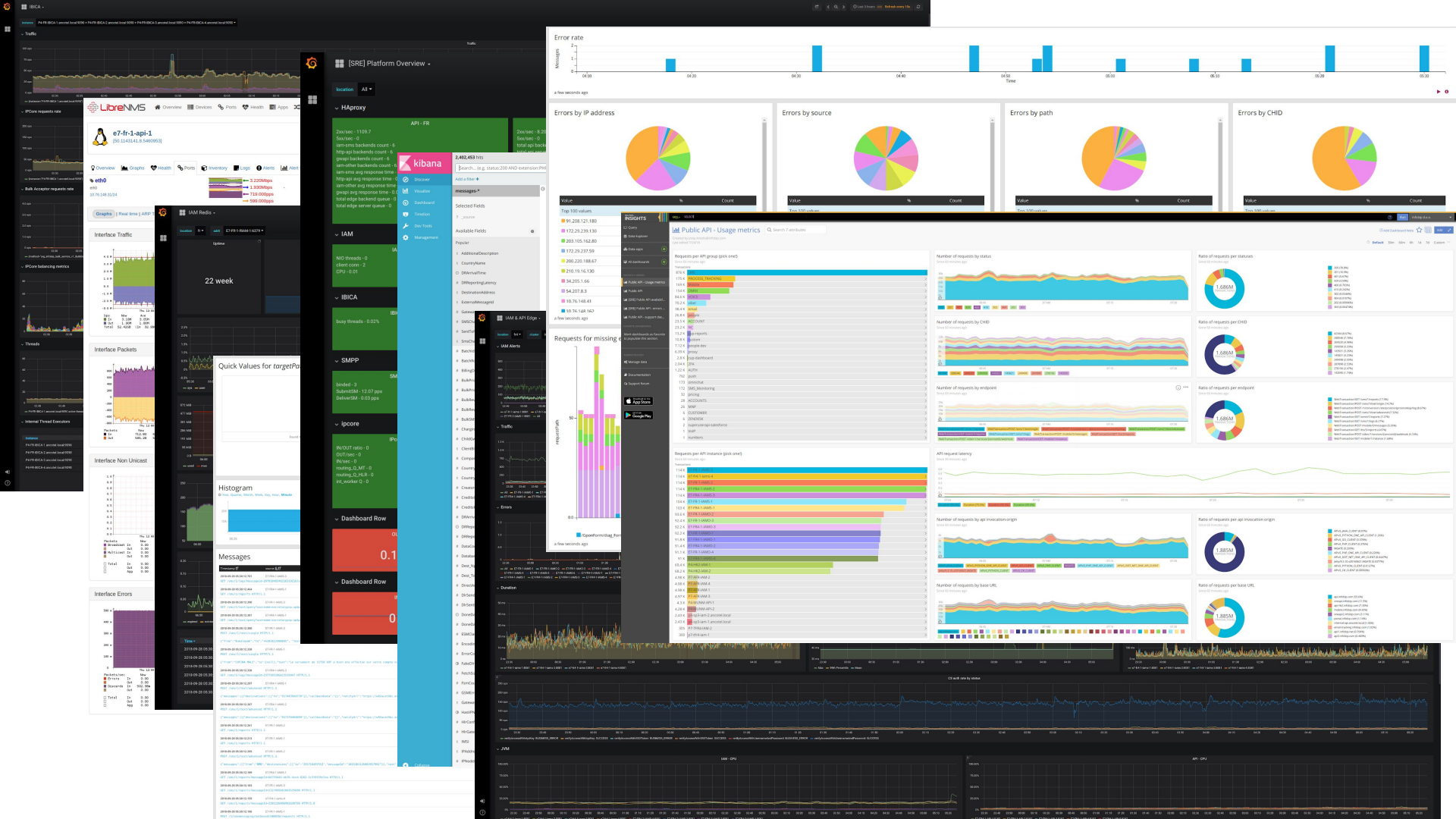

Internet
API
Backend
Backend
Backend
Backend
SMS Team
Accounts Team
Email Team



Survival

Lesson #4

Monitor everything, alert few


Error rate
Errors by IP address
Errors by source
Errors by path
Errors by CHID
[SRE] Platform Overview
HAproxy
IAM
IBICA
SMPP
ipcore
Dashboard Row
kibana
Discover
Visualize
Dashboard
Timelion
Dev Tools
Management
Public API - Usage metrics
Number of requests by status
Number of requests by CHID
Number of requests by endpoint
Number of requests by base URL
API request latency
Requests for missing e
Histogram
Messages
Interface Traffic
Interface Packets
Interface Non Unicast
Interface Errors
IAM Redis
IAM & API Edge
Quick Values for
22 week
LibreNMS
e7-fr-1-api-1
eth0
3.220Mbps
1.330Mbps
719.000pps
588.000pps
IAM - CPU
API - CPU
JVM

Request rate

Latency

Response status

...

Service

Host

Network

Business

Retention

Organisation

Automate

Minimise false positives

Revise

Reasonable about service logs

Public communication logs

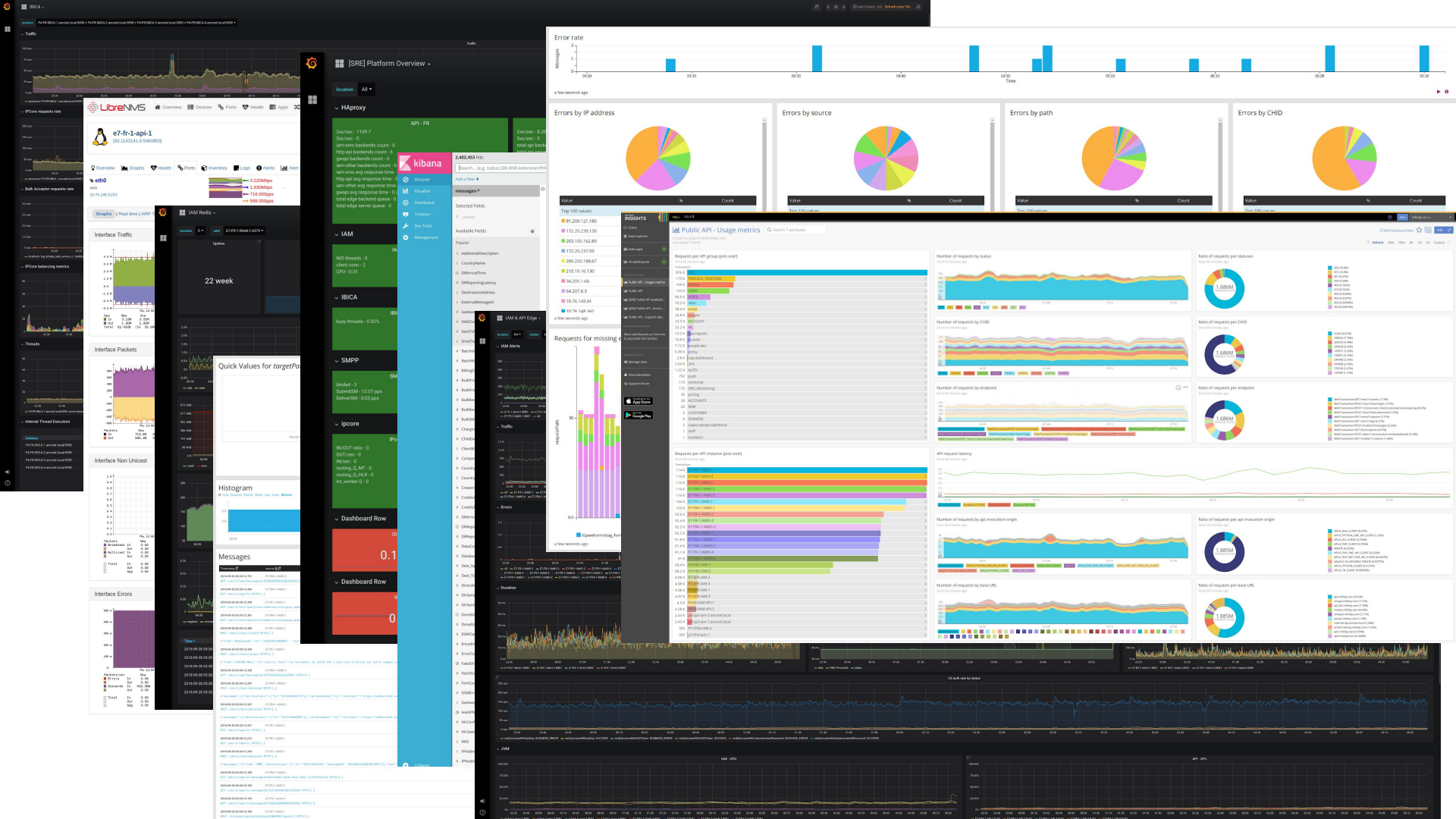

graylog


elastic


Grafana

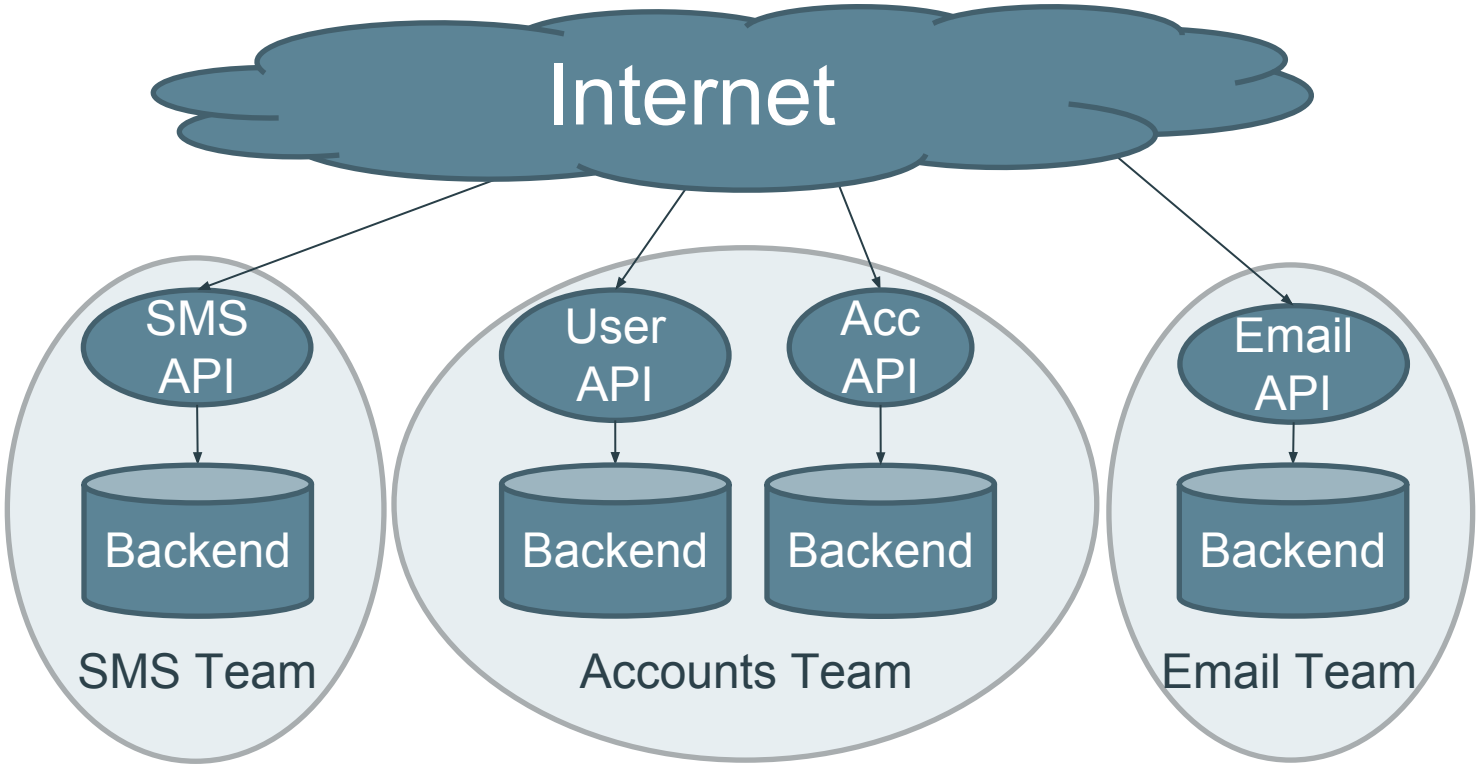

New Relic


logstash


Error rate
Errors by IP address
Errors by source
Errors by path
Errors by CHID
[SRE] Platform Overview
HAproxy
IAM
IBICA
SMPP
ipcore
Dashboard Row
kibana
Discover
Visualize
Dashboard
Timelion
Dev Tools
Management
Public API - Usage metrics
Number of requests by status
Number of requests by CHID
Number of requests by endpoint
Number of requests by base URL
API request latency
Requests for missing e
Histogram
Messages
Interface Traffic
Interface Packets
Interface Non Unicast
Interface Errors
IAM Redis
IAM & API Edge
Quick Values for
22 week
LibreNMS
e7-fr-1-api-1
eth0
3.220Mbps
1.330Mbps
719.000pps
588.000pps
IAM - CPU
API - CPU
JVM

How to scale?

Independent API development



Lesson #5

Can I copy your homework?

API gateways
API management systems

zuul

Kong API

...

Goal?

A solution that will
make it **simple** for **each team**
to **independently**
expose and manage **public endpoints**
with all the benefits of a **centralised system**

What do we need?

Definition

```
{
    path: /sms/send
    method: POST
    authentication: {...}
    serviceMapping: {...}
}
```

What do we need?

Empower developers

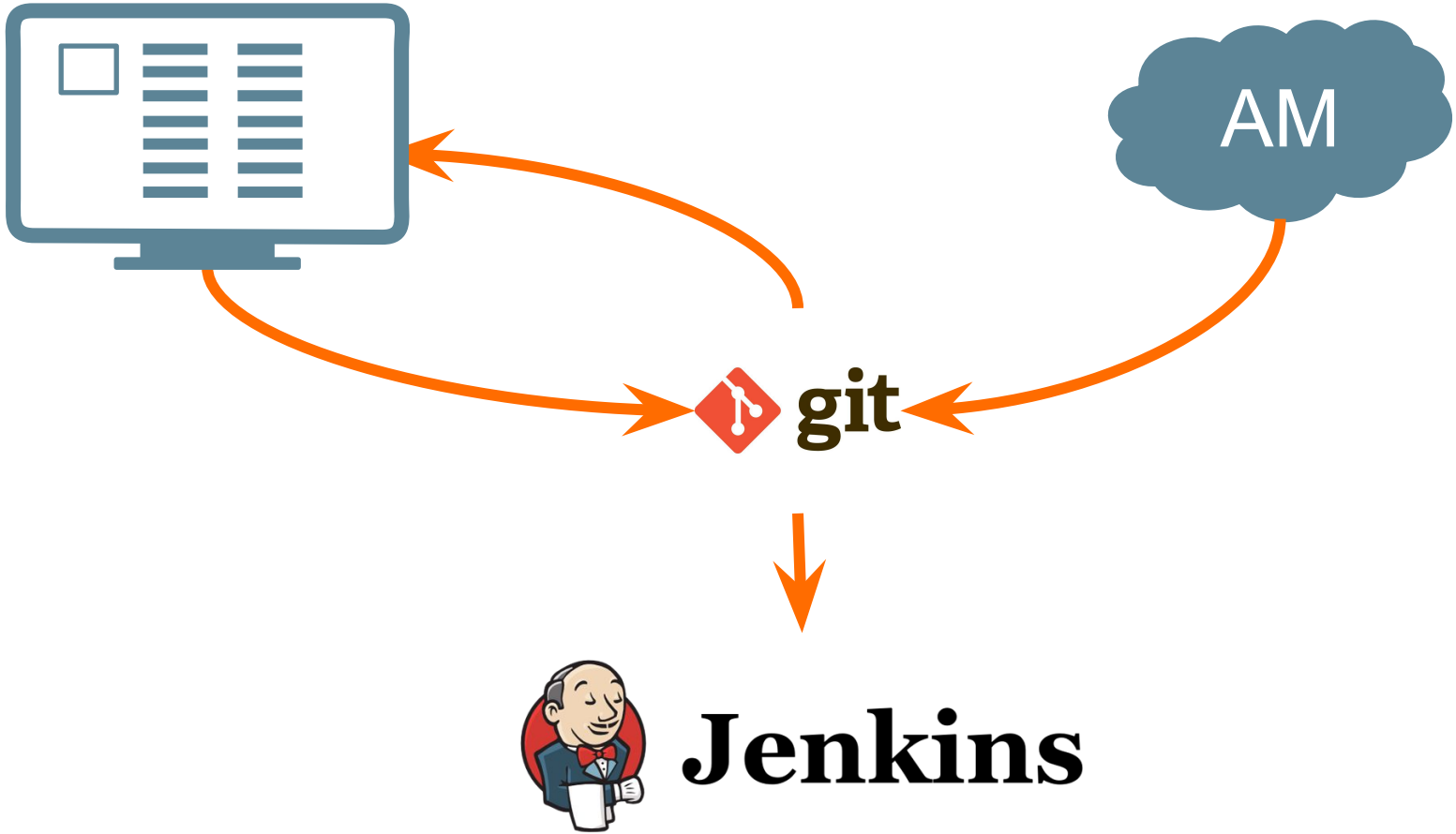
```
new GroovyScriptEngine(“/path”, classLoader)
    .loadScriptByName(“easy.groovy”);
```

What do we need?

Git

Maven

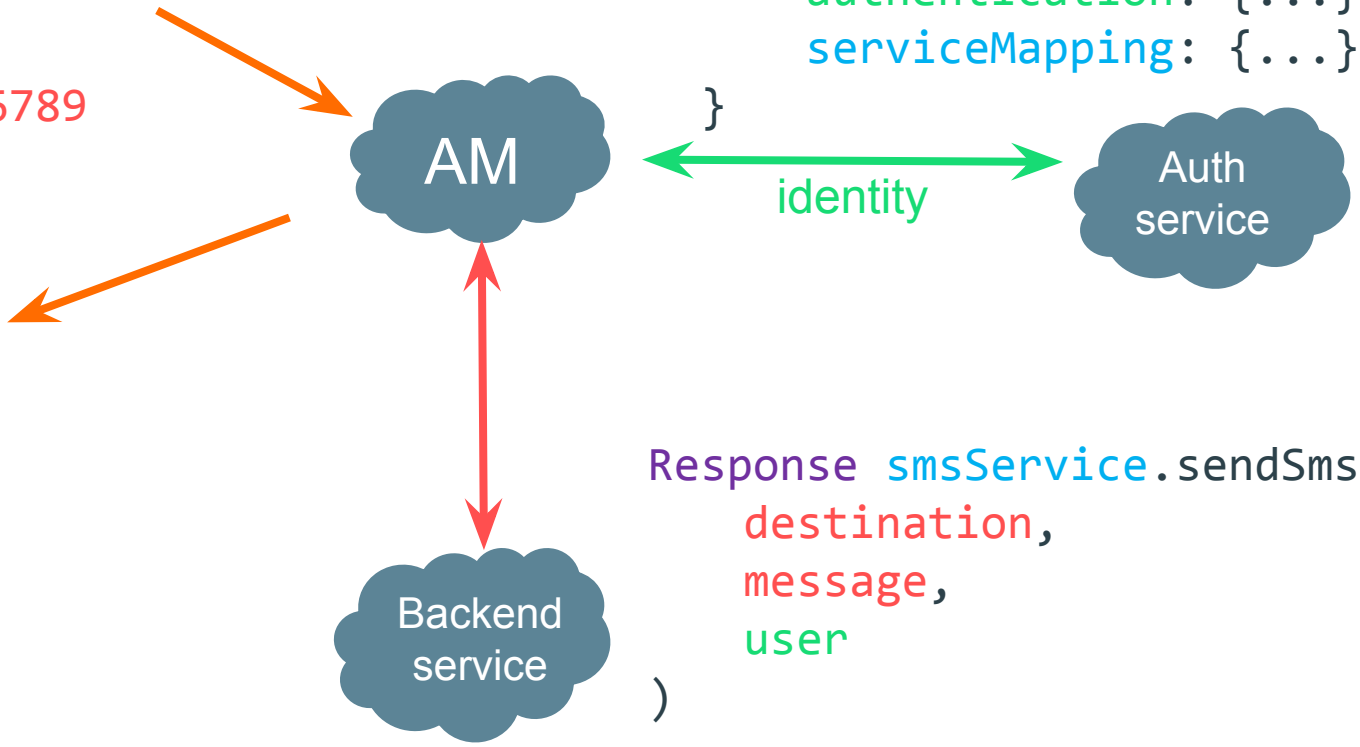
CI / CD


AM
git
Jenkins

```
POST /sms/send HTTP/1.1
Content-Type: application/json
Authorization: Basic 4asdf234

{
    to: 385123456789
    text: Hello
}
```

```
200 OK
{
    status: SENT
}
```




```
{
    path: /sms/send
    method: POST
    authentication: {...}
    serviceMapping: {...}
}
```

```
Response smsService.sendSms(
    destination,
    message,
    user
)
```

Considerations

Lesson #6

Make it async

Slide unavailable

Http error: 503 - slide currently unavailable

Lesson #7

Async all the way

CompletableFuture Callbacks

Backend trouble

Balancers

Lesson #8

Even more async

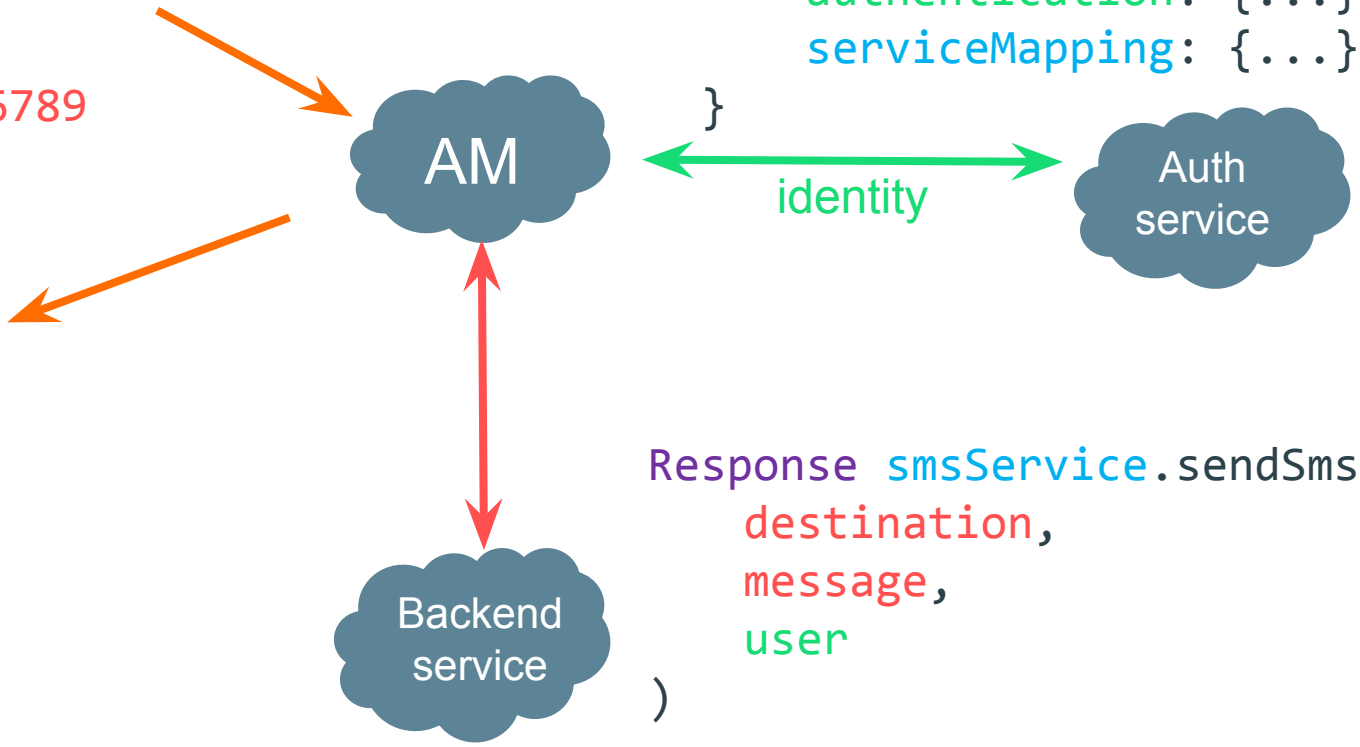
Queueing

Design for async

```
POST /sms/send HTTP/1.1
Content-Type: application/json
Authorization: Basic 4asdf234

{
    to: 385123456789
    text: Hello
}
```

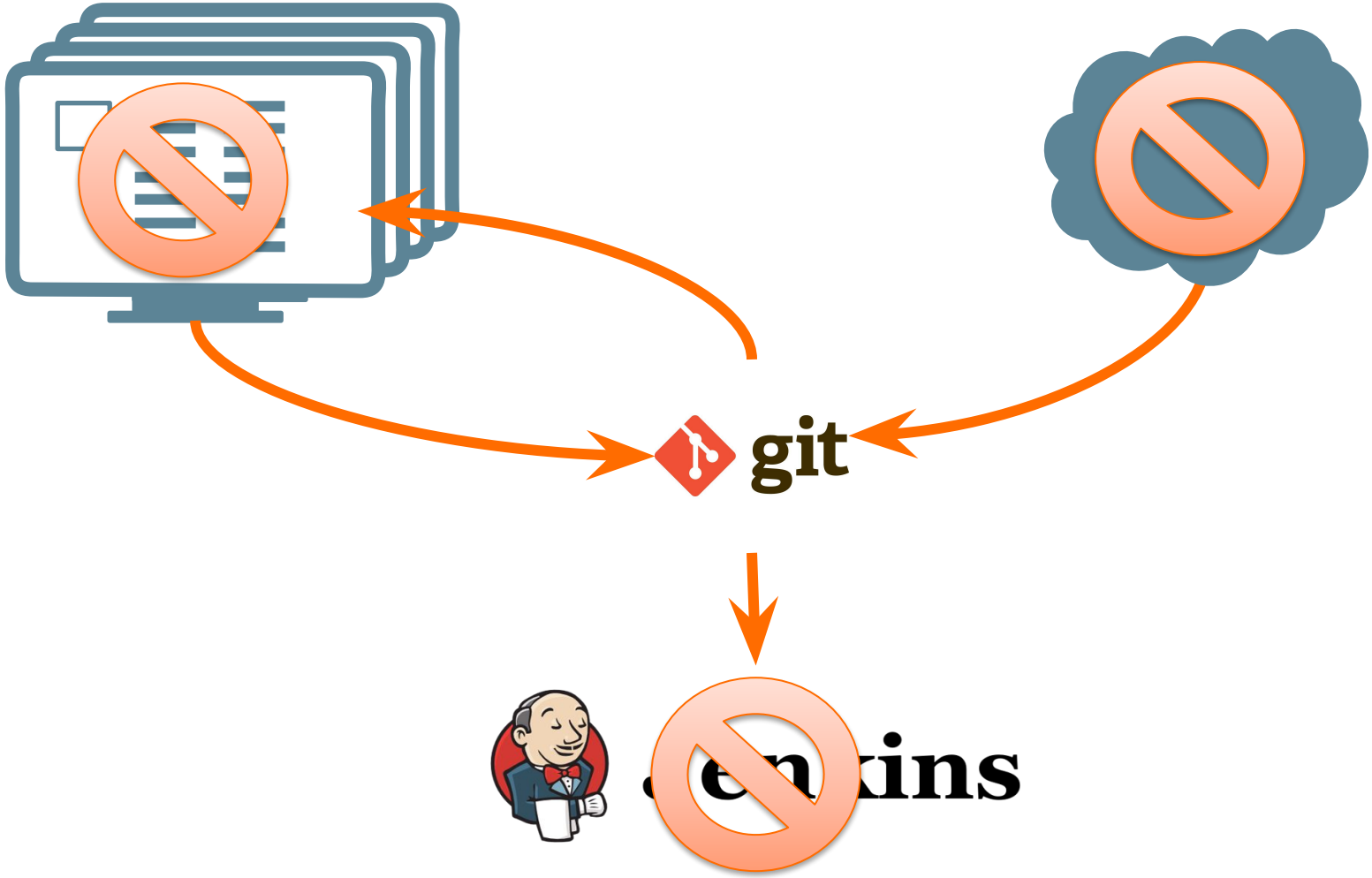
```
200 OK
{
    status: SENT
}
```




```
{
    path: /sms/send
    method: POST
    authentication: {...}
    serviceMapping: {...}
}
```

```
Response smsService.sendSms(
    destination,
    message,
    user
)
```

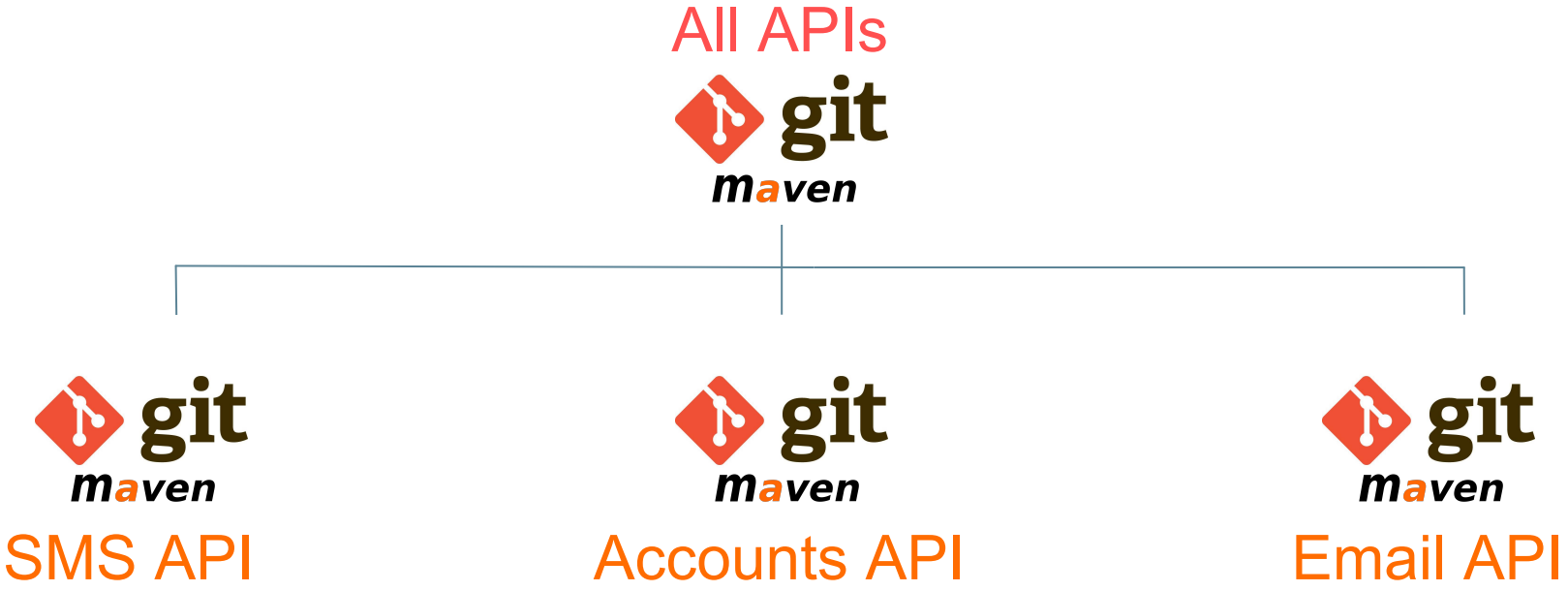
Ready for production!

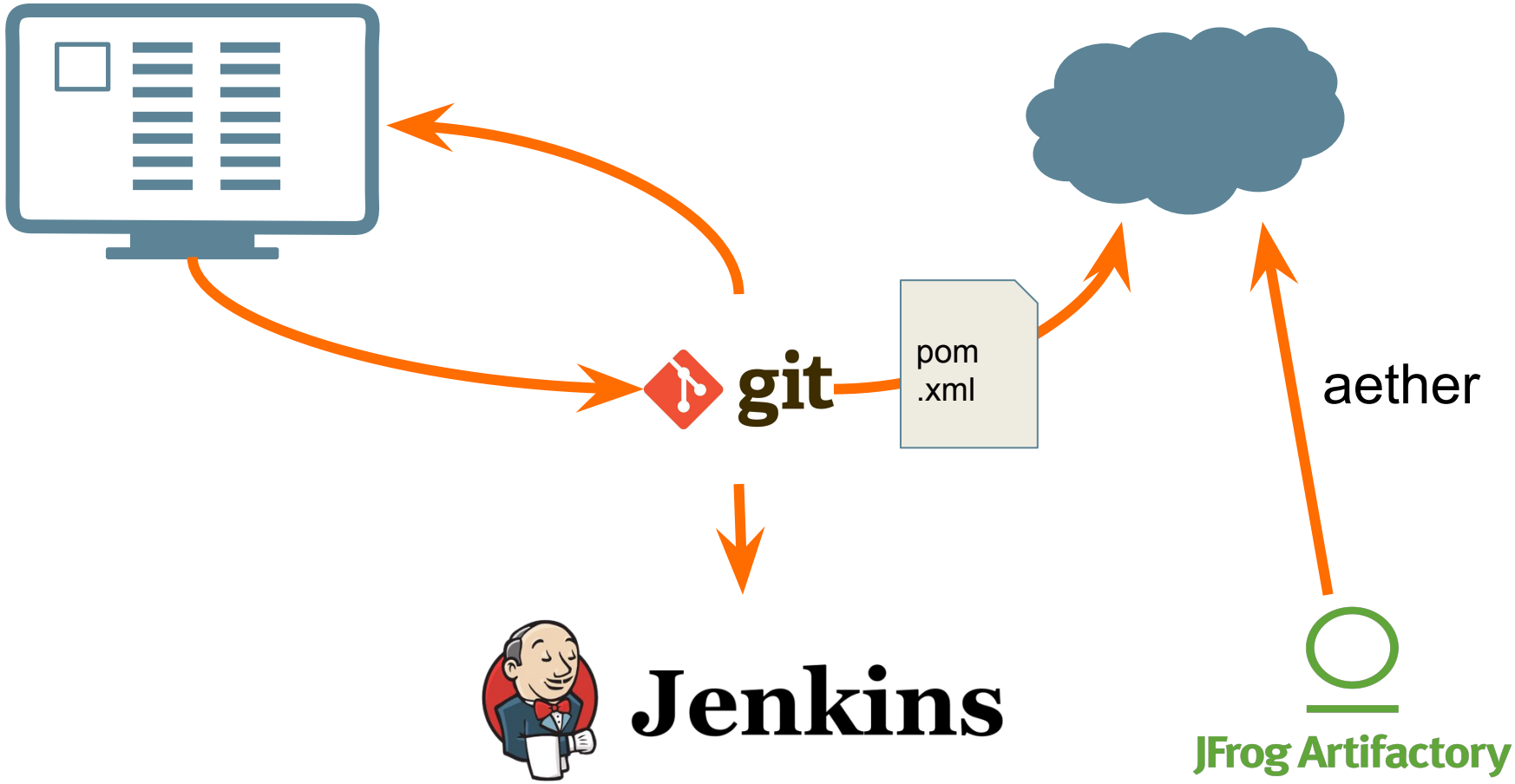

git
Jenkins

Remote Method Invocation

Everybody waiting for everybody

Time to evolve...

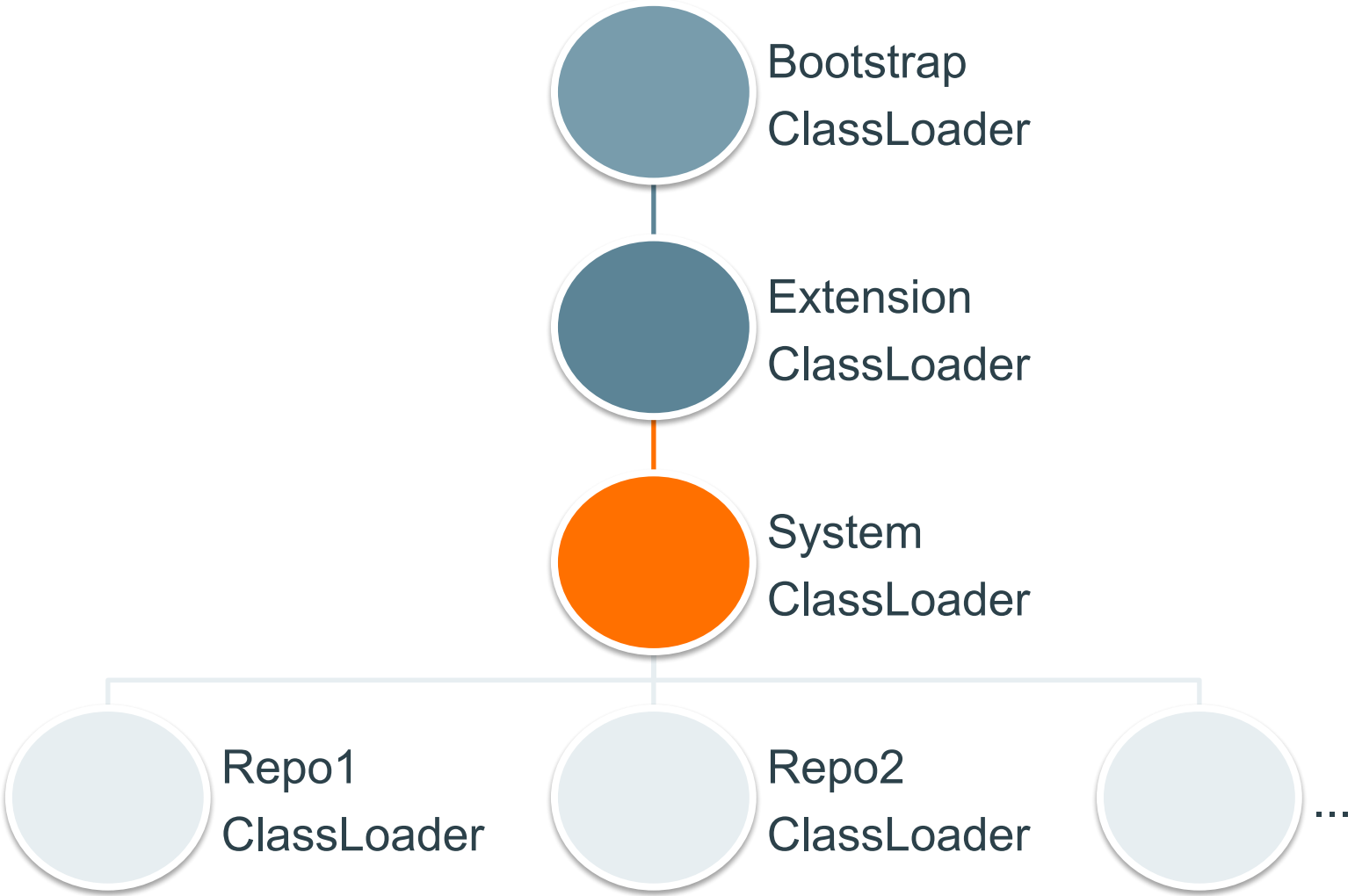

All APIs
git
maven
git
maven
SMS API
git
maven
Accounts API
git
maven
Email API


git
pom
.xml
aether
Jenkins
JFrog Artifactory

What could go wrong?

Lesson #9

Avoid dependency hell

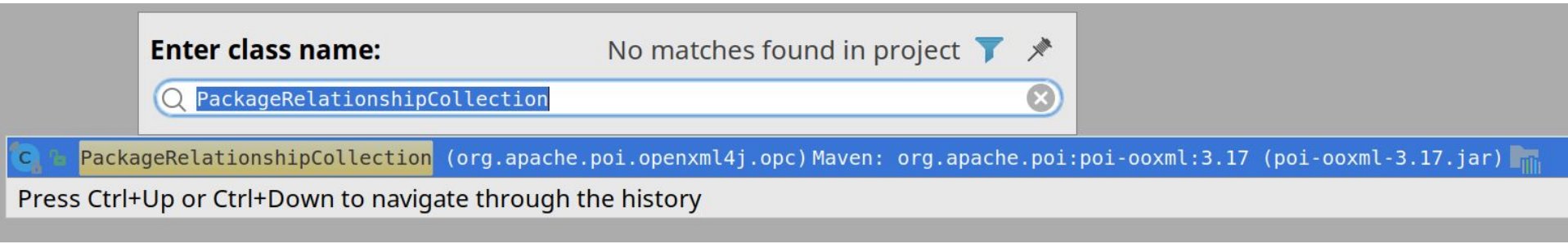

Bootstrap
ClassLoader
Extension
ClassLoader
System
ClassLoader
Repo1
ClassLoader
Repo2
ClassLoader
...

Resolved?

[INFO] java.lang.NoSuchMethodError:
org.apache.poi.util.POILogger.log(I[Ljava/lang/Object;)V

[INFO] at o.a.p.o.o.PackageRelationshipCollection.parseRelationshipsPart

(PackageRelationshipCollection.java:314)



org.apache.poi:poi-ooxml:3.17

?

```
304  @    public void parseRelationshipsPart(PackagePart relPart)
305           throws InvalidFormatException {
306       try {
307           logger.log(POILogger.DEBUG,  ...objs: "Parsing relationship: " + relPart.getPartName());
308           Document xmlRelationshipsDoc = DocumentHelper.readDocument(relPart.getInputStream());
309
310           // Browse default types
311           Element root = xmlRelationshipsDoc.getDocumentElement();
312
313           // Check OPC compliance M4.1 rule
314           boolean fCorePropertiesRelationship = false;
```

```
304  @    public void parseRelationshipsPart(PackagePart relPart)
305               throws InvalidFormatException {
306           try {
307               logger.log(POILogger.DEBUG,  ...objs: "Parsing relationship: " + relPart.getPartName());
308               Document xmlRelationshipsDoc = DocumentHelper.readDocument(relPart.getInputStream());
309
310               // Browse default types
311               Element root = xmlRelationshipsDoc.getDocumentElement();
312
313               // Check OPC compliance M4.1 rule
314               boolean fCorePropertiesRelationship = false;
```

```
304  @    public void parseRelationshipsPart(PackagePart relPart)
305               throws InvalidFormatException {
306           try {
307               logger.log(POILogger.DEBUG,  ...objs: "Parsing relationship: " + relPart.getPartName());
308               Document xmlRelationshipsDoc = DocumentHelper.readDocument(relPart.getInputStream());
309
310               // Browse default types
311               Element root = xmlRelationshipsDoc.getDocumentElement();
312
313               // Check OPC compliance M4.1 rule
314               boolean fCorePropertiesRelationship = false;
```

```
this.getClass()
.getProtectionDomain()
.getCodeSource()
.getLocation()
.getPath()
```

/home/dcutic/.m2/repository
/org/apache/poi/poi-ooxml
/3.14/poi-ooxml-3.14.jar

?

args.nodes = {NodeStack@25025} "[org.infobip.campaign:infobip-campaign-connector:jar:2.94.4 (compile), org.infobip.campaign:infobip-s

- nodes = {DependencyNode[96]@25394}
  - Not showing null elements
  - 0 = {DefaultDependencyNode@25023} "org.infobip.campaign:infobip-campaign-connector:jar:2.94.4 (compile)"
  - 1 = {DefaultDependencyNode@25056} "org.infobip.campaign:infobip-sending-analysis-connector:jar:1.4.15 (compile)"
  - 2 = {DefaultDependencyNode@25089} "org.infobip.campaign:infobip-people-query-model:jar:1.1.1 (compile)"
  - 3 = {DefaultDependencyNode@25119} "org.infobip.common:infobip-spring-bean-validation-boot-starter:jar:1.5.1 (compile)"
  - 4 = {DefaultDependencyNode@25150} "org.springframework.boot:spring-boot-starter-validation:jar:1.4.3.RELEASE (compile)"
  - 5 = {DefaultDependencyNode@25180} "org.springframework.boot:spring-boot-starter:jar:1.4.3.RELEASE (compile)"
  - 6 = {DefaultDependencyNode@25211} "org.springframework.boot:spring-boot:jar:1.4.3.RELEASE (compile)"
  - 7 = {DefaultDependencyNode@25242} "org.springframework:spring-test:jar:4.3.5.RELEASE (compile)"
  - 8 = {DefaultDependencyNode@25273} "org.springframework:spring-webmvc:jar:4.3.5.RELEASE (compile)"
  - 9 = {DefaultDependencyNode@25295} "net.sf.jasperreports:jasperreports:jar:6.3.1 (compile)"
  - 10 = {DefaultDependencyNode@25395} "org.apache.velocity:velocity:jar:1.7 (compile)"
  - 11 = {DefaultDependencyNode@25396} "commons-lang:commons-lang:jar:2.6 (compile)"
  - 12 = {DefaultDependencyNode@25397} "org.apache.commons:commons-lang3:jar:3.4 (compile)"
  - 13 = {DefaultDependencyNode@25398} "jaxen:jaxen:jar:1.1.6 (compile)"
  - 14 = {DefaultDependencyNode@25399} "xom:xom:jar:1.2.5 (compile)"
  - 15 = {DefaultDependencyNode@25400} "xalan:xalan:jar:2.7.0 (compile)"
  - size = 10

Lesson #10

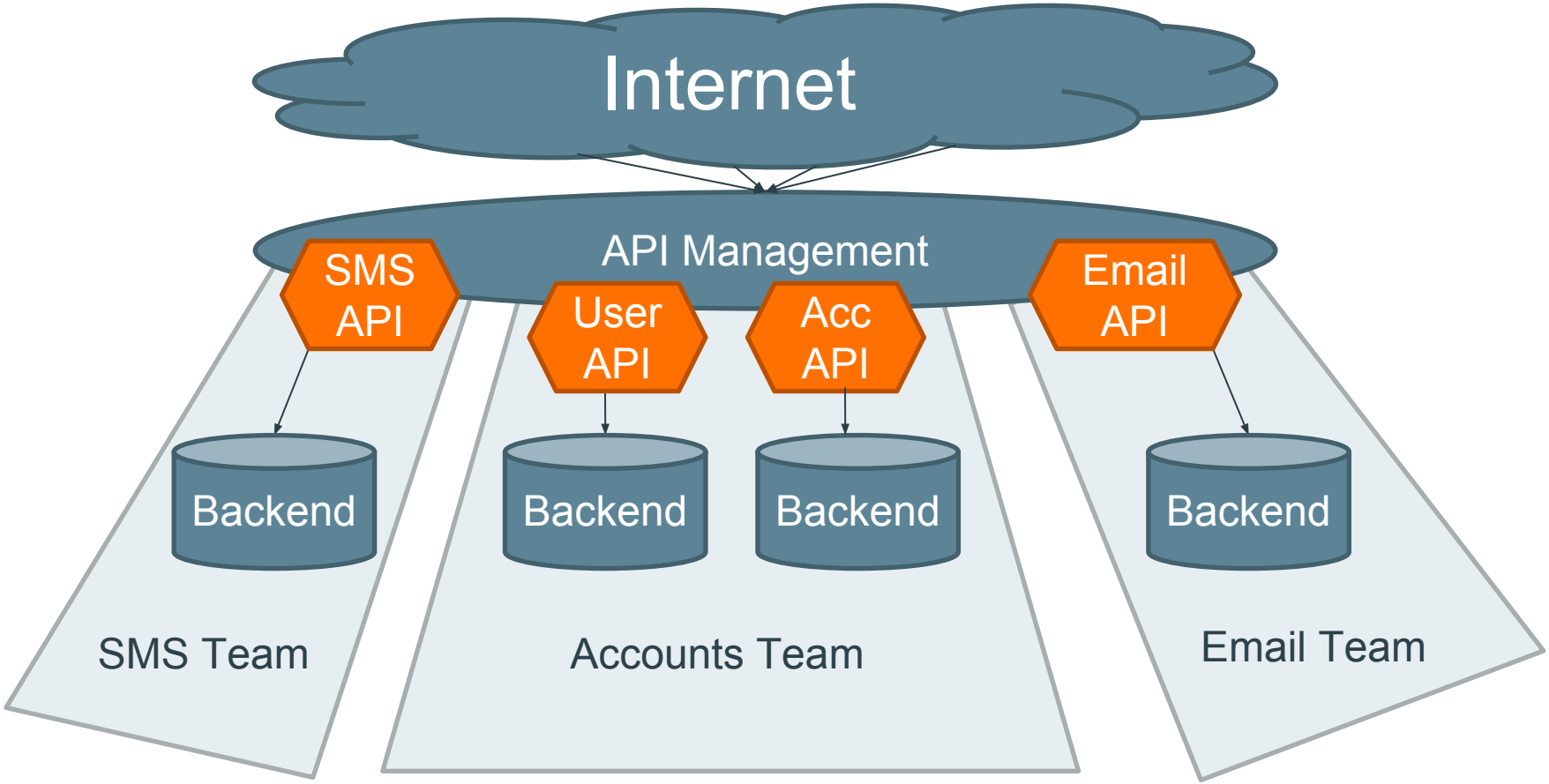
Watchout for Metaspace

References to classes

Duplicate classes

Bytebuddy

runtime.created.Class$ByteBuddy$uEI1wlrb


Internet
API Management
SMS API
User API
Acc API
Email API
Backend
Backend
Backend
Backend
SMS Team
Accounts Team
Email Team

Complex issues

Independent APIs

Lesson #11

Development flow

Testing

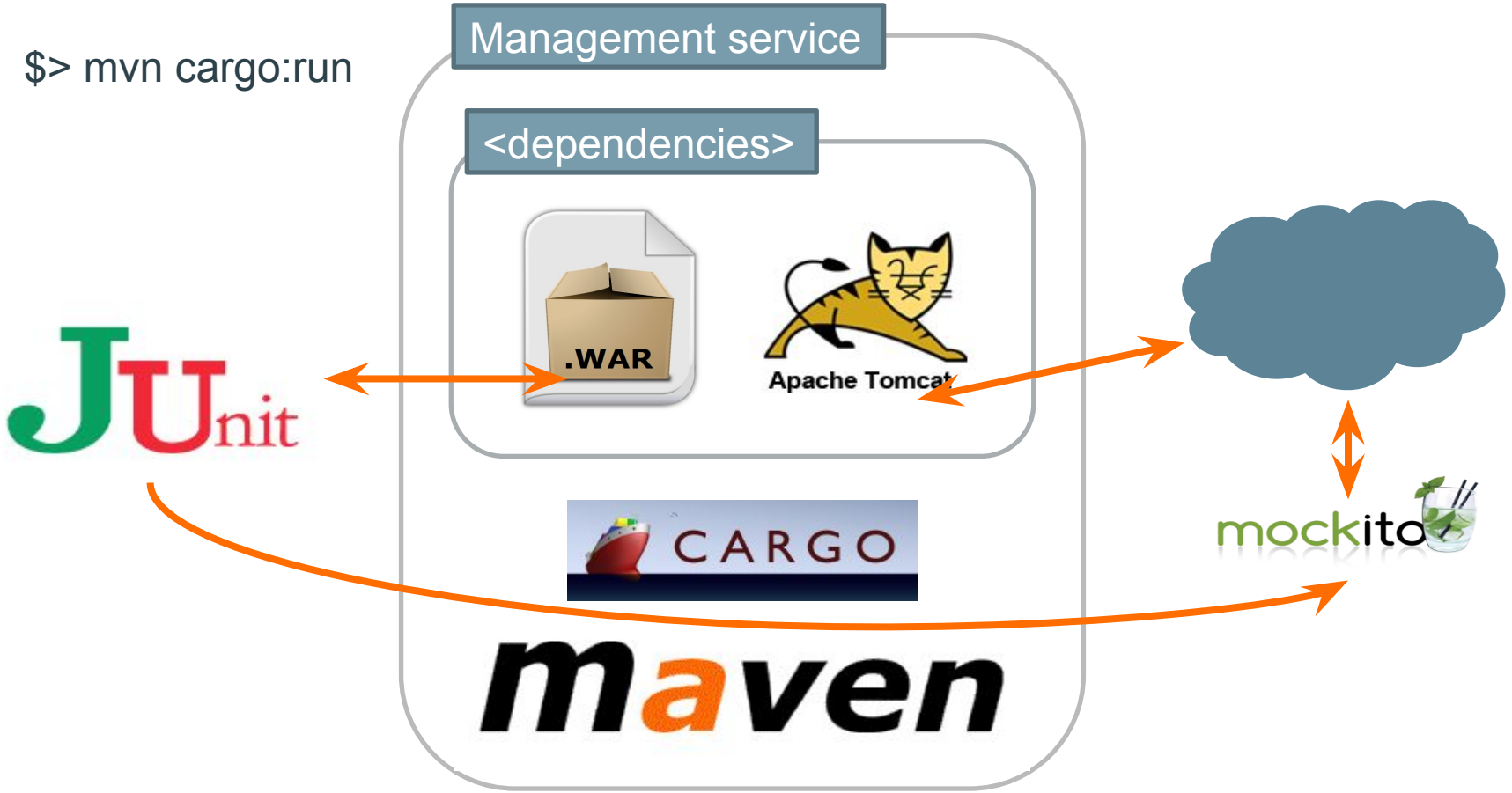
Running API
Testing framework
Dependency mocking

Management service

<dependencies>

maven

$> mvn cargo:run



Start RMI server

Serve mocks

Reference mocks from tests

1. Write the tests
2. Start the management system with the API
3. Work on the API endpoints
4. Run the tests
5. Repeat the steps 3. and 4. until all tests pass

Looking good :)

2016

20 APIs

10 teams

10M req/d

Lesson #12

Does it really scale? To what point?

Lesson #3

Always look ahead

http://static.tvtropes.org/pmwiki/pub/images/rake-take_the-simpsons_951.jpg

28 new APIs (48 total)

10 new teams (20 total)

10 times the traffic (100M req/d total)

Badly developed APIs
Fast development

Main issues

API Production issues

Main issues

Memory issues

Main issues

Heap leaks

Huge requests

Main issues

Lots of dependencies

Frequent reloading

Main issues

Memory gone native

Main issues

Resource exhaustion

Main issues

Startup time

Main issues

Balancer configuration

What to do?

Lesson #13

Careful with that axe, Eugene

Changes are slow and painful

Clustering

Improving deployment

Lesson #14

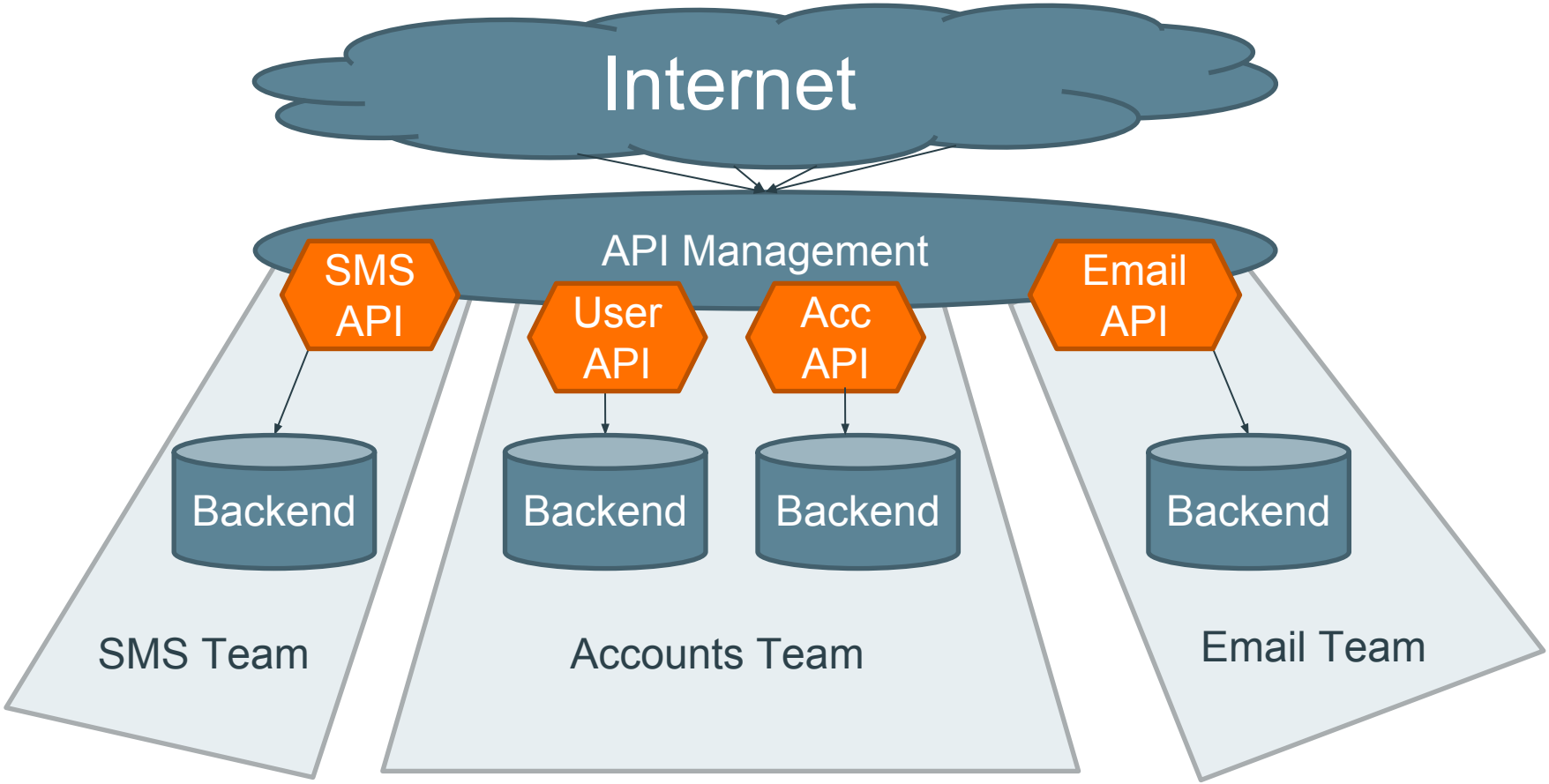
Prevent impending doom

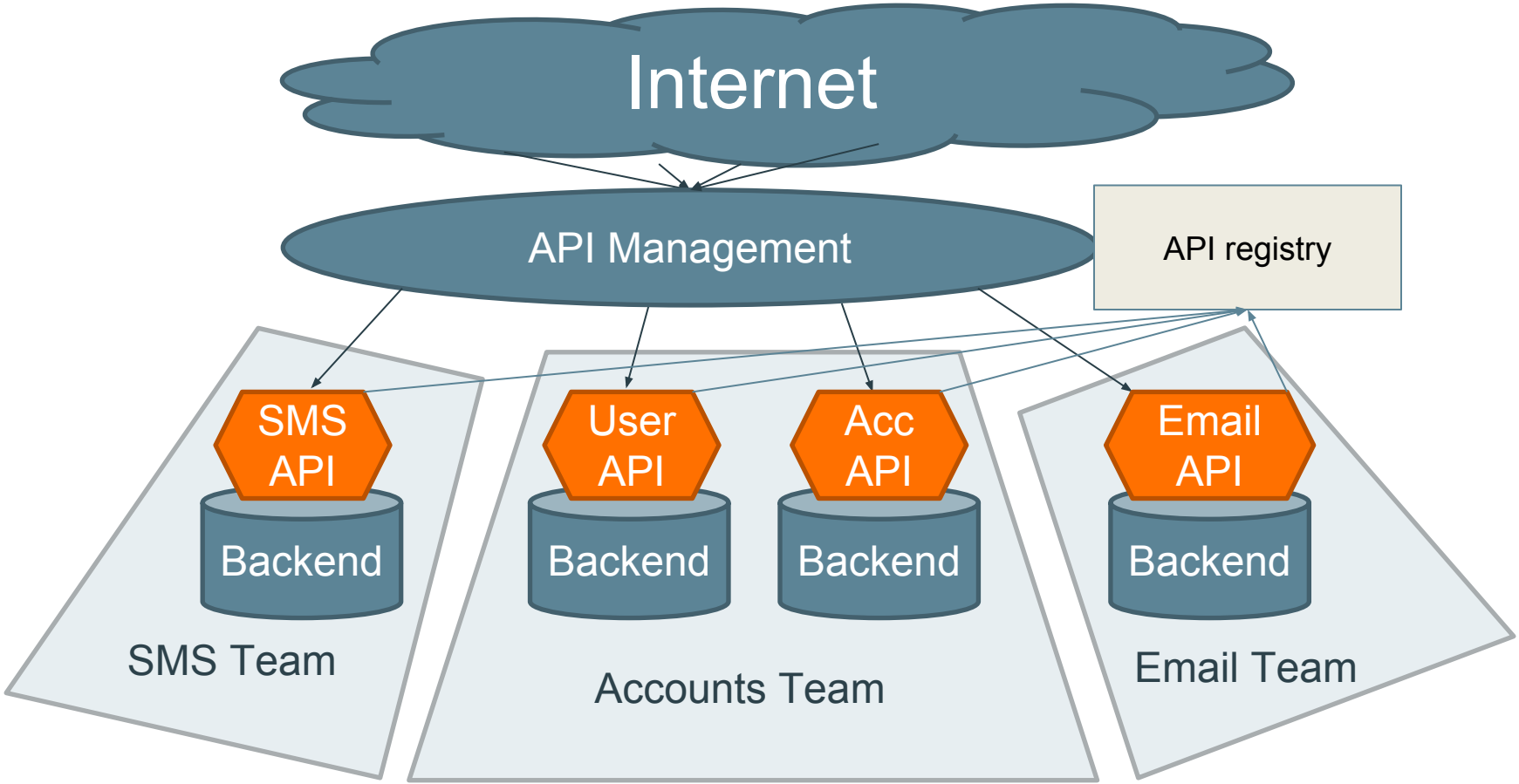
Lesson #15

Avoid tight coupling

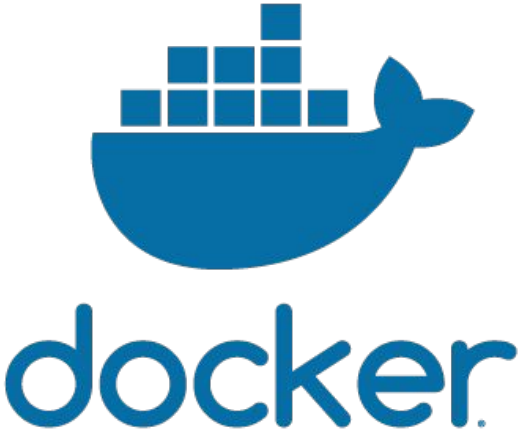
APIs
API management system

API registry


Internet
API Management
SMS API
User API
Acc API
Email API
Backend
Backend
Backend
Backend
SMS Team
Accounts Team
Email Team


Internet
API Management
API registry
SMS API
User API
Acc API
Email API
Backend
Backend
Backend
Backend
SMS Team
Accounts Team
Email Team

Make existing APIs standalone



End of story...

Lesson #1

Don’t ever do this

```
if (this) {
    do something
} else if (that) {
    do something similar
} else if (friend) {
    be nice
} else if (foo) {
    be rude
} else if (boss) {
    be extra nice
}
```

Lesson #2

Legacy public APIs (almost) never die

Lesson #3

Always look ahead

http://static.tvtropes.org/pmwiki/pub/images/rake-take_the-simpsons_951.jpg

Lesson #4

Monitor everything, alert few

Lesson #5

Can I copy your homework?

Lesson #6

Make it async

Lesson #7

Async all the way

Lesson #8

Even more async

Lesson #9

Avoid dependency hell

Lesson #10

Watchout for Metaspace

Lesson #11

Development flow

Lesson #12

Does it really scale? To what point?

Lesson #13

Careful with that axe, Eugene

Lesson #14

Prevent impending doom

Lesson #15

Avoid tight coupling

CONCLUSION