Godzilla, Hydra, and Tribbles

Evolving a late 90's Java App to Cloud-based Microservices

Josh Graham
Atlassian
SaaS Architect
@delitescere


ONE DOES NOT SIMPLY
FORGET THE PRODUCT NAMES
imgflip.com

META

- Understand the code
- Why the code exists
- “You can't just walk into the desert and expect to find ancient pyramids.” — *Matt Quail*
- “Evolutionary biology might be a better metaphor than archaeology.” — *Charles Miller*

META

Don’t code?

Don’t architect.

Atlassian

@delitescere

大東宝が放つ空想映画の巨篇

水爆大怪獣映画

ゴジラ

東宝

原作・香山滋
監督・本多猪四郎

凄絶壮烈！死の放射能を
発する廿紀の怪獣ゴジラ
!!

宝田一男
河内桃子
平田昭彦
志村喬
明子彦喬

堺左千夫
村上冬樹
山本廉二
鈴木豊明
木田敬

脚本・村田武雄
撮影・玉井正夫

奥田社中林印刷所
奥田社大阪支店

21日1時ヨリ継続上映 大70 小40

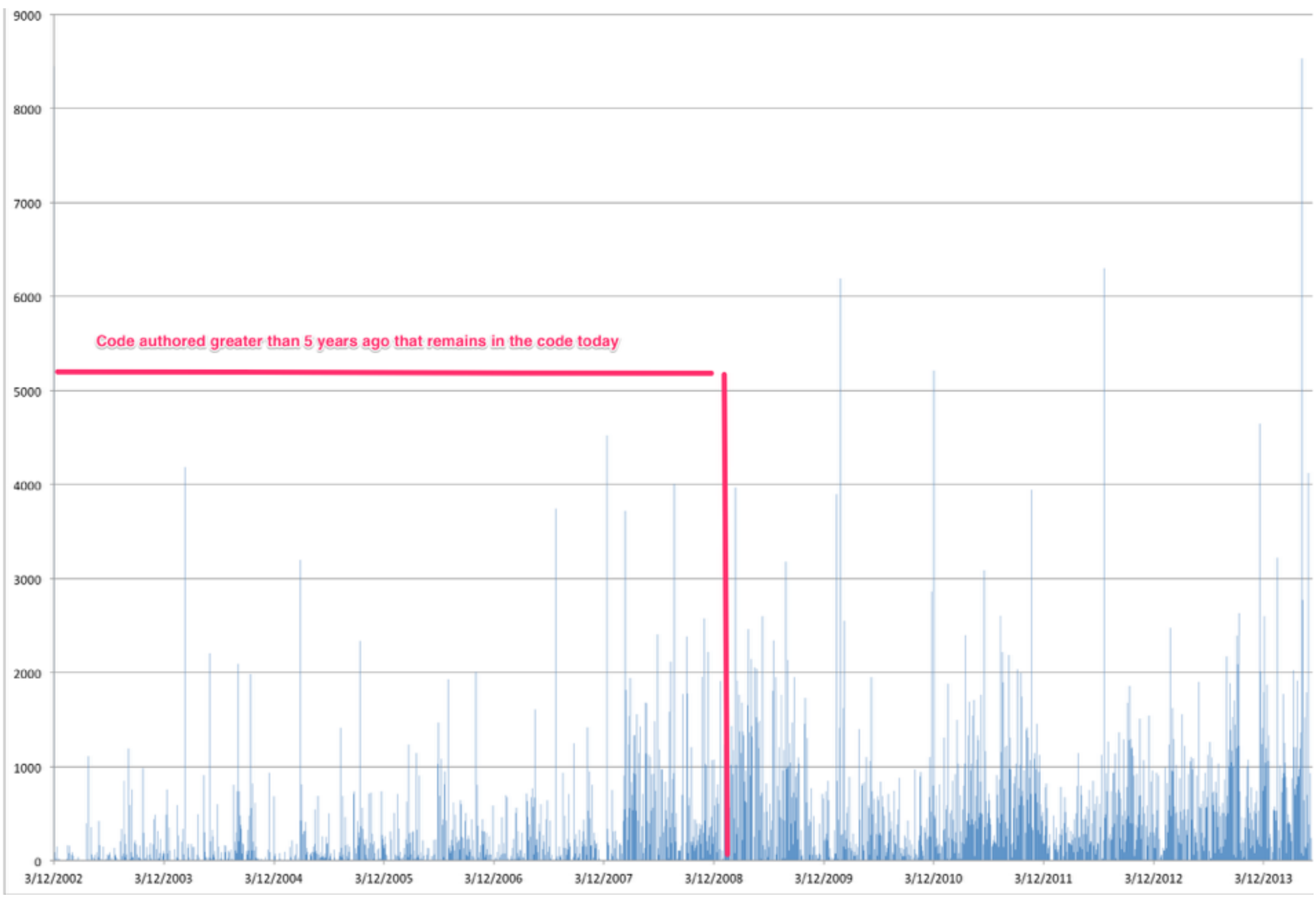
Ancient

- Since late 2001

```
commit 4b114e33ebf7bdecc09fc845f0b0ca3e3cc6f02e
Author: Mike Cannon-Brookes <mike@atlassian.com>
Date:   Mon Dec 17 03:09:37 2001 +0000

    Adding initial build files
```

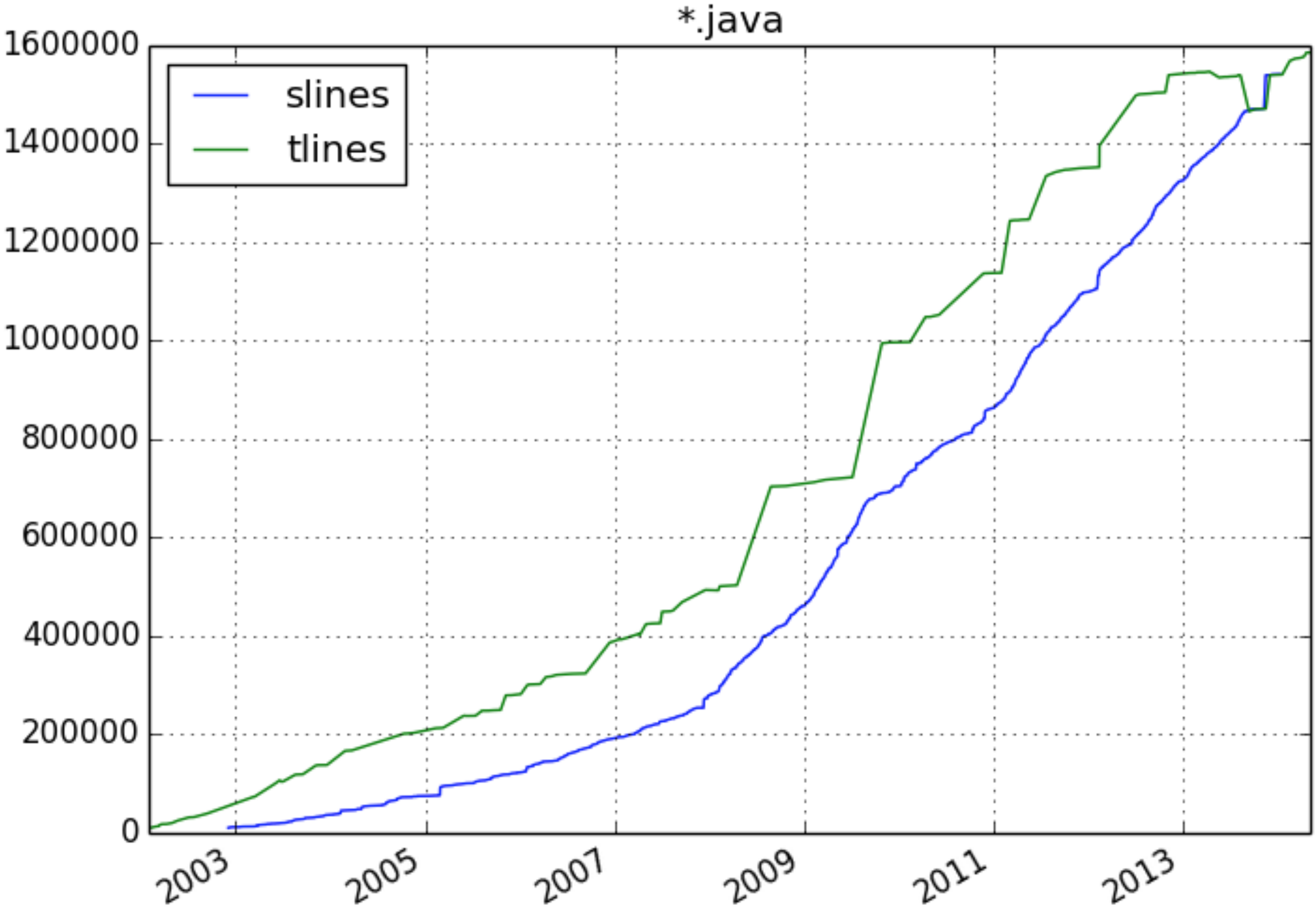
- “WORA”
- “Open For Business” Entity Engine, Hibernate



Survivor

Atlassian

@delitescere



Survivor

Atlassian

@delitescere

Massive

- 73,000 commits
- 300 committers
- 580+ packages
- 6,500+ classes
- 100 dependency JARs
- 140MB download
- 100 bundled plugins
- Several GB heap 😒

Loner

Everything else

Atlassian

@delitescere

Loner

Nothing else

Loner

Nothing else

Issue Tracker

Wiki

Loner

- Source code customisations

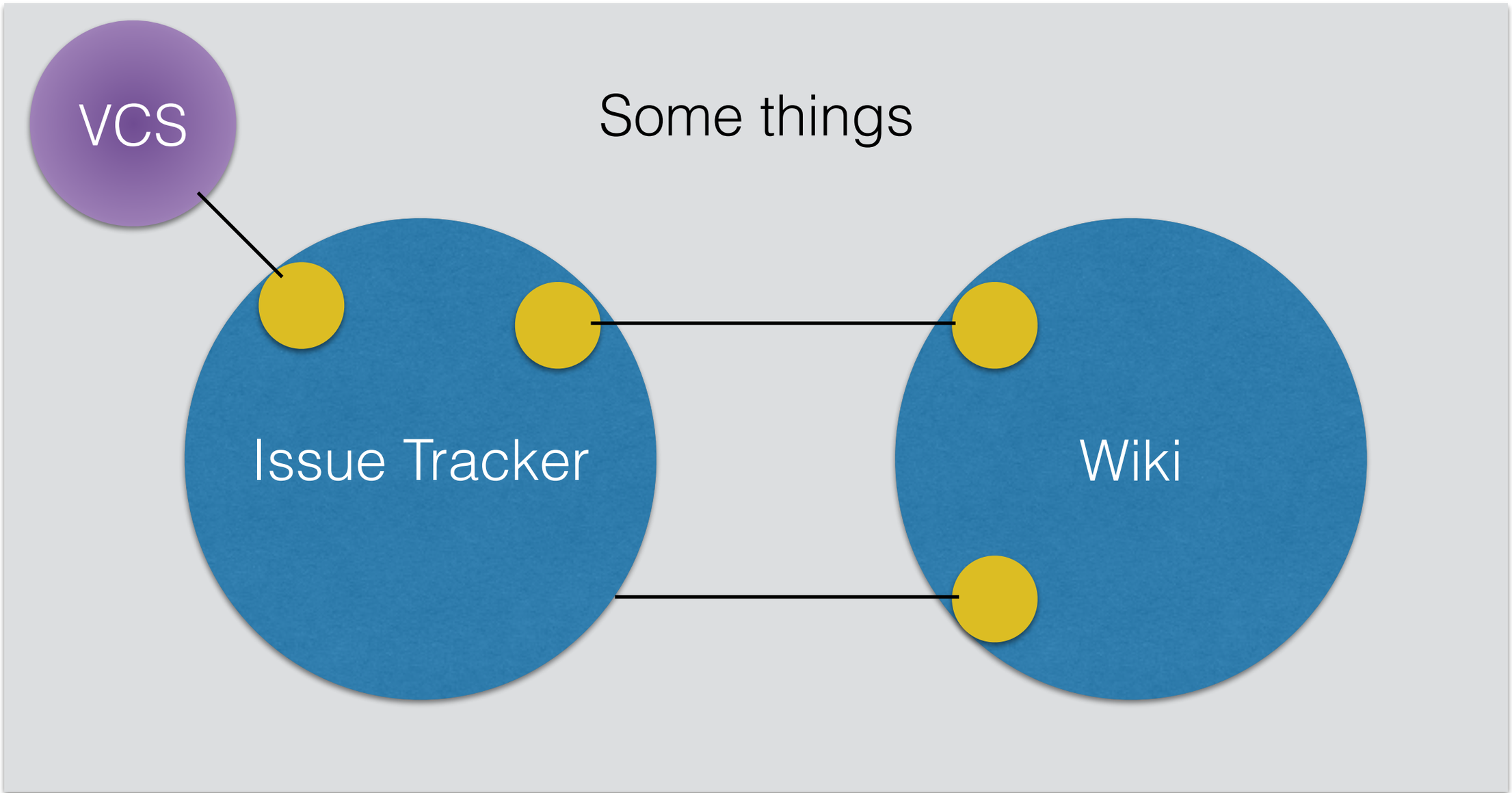
- A `.jar` in `WEB-INF/lib`

Loner (with extensions)

- `WEB-INF/lib/*.jar`
  - `macro-library.xml`
  - `atlassian-plugins.xml` (Plugins 1)
- MCB + Hani Suleiman = dynamic loading

Artist: Stjepan Sejic

More teeth



Atlassian

@delitescere

Many heads, many hands

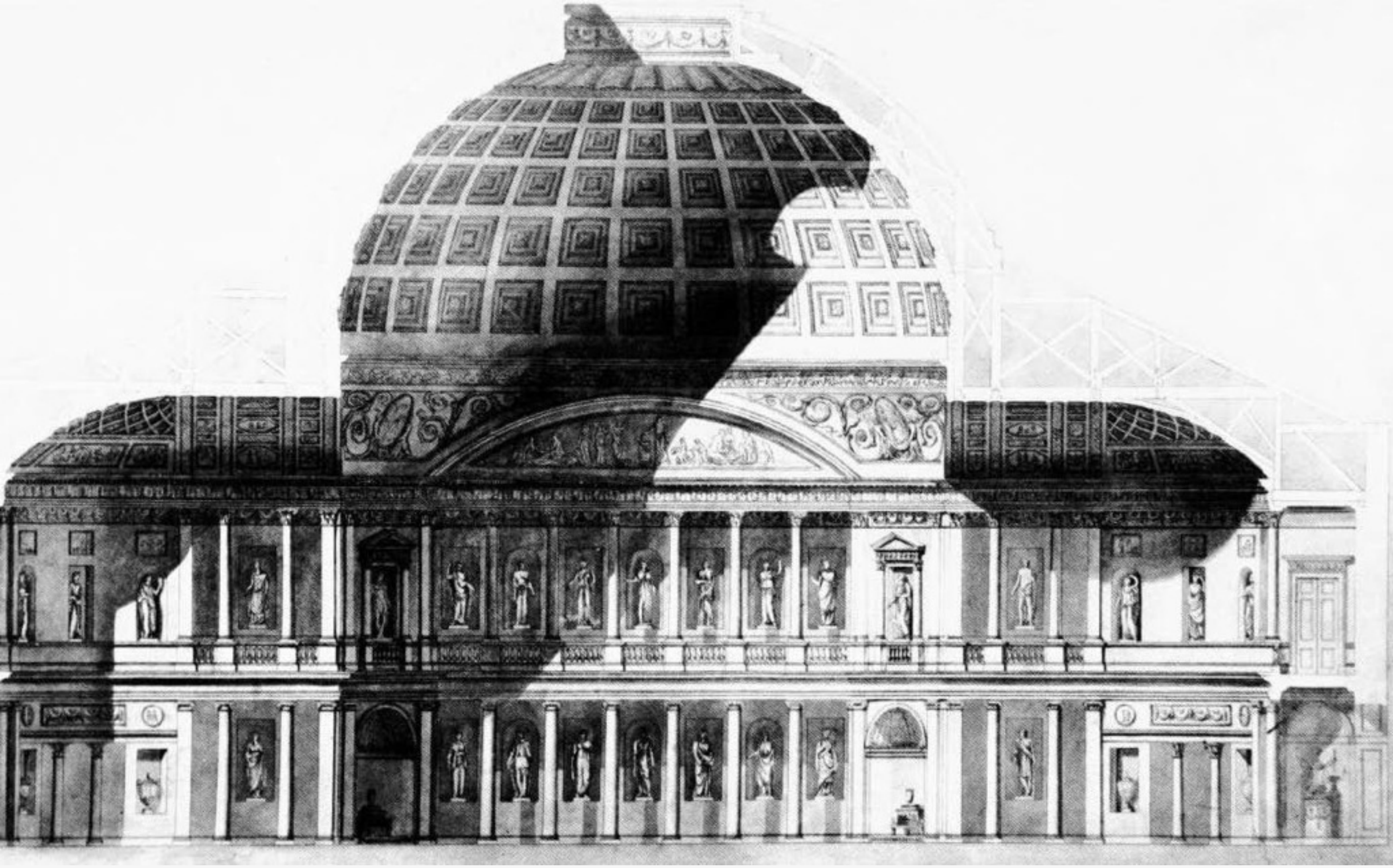
- Product as a platform
- Architecture created opportunity for ecosystem
- 15,000 add-ons, 7.6M downloads

Atlassian

SaaS

- Integrated experience
- Internal platform and APIs
- Memory consumption
- Availability
- Continuous Deployment

Atlassian



Indra

- Aggregate user information administration
- Login, Logout, SSO, Forgot Credentials, Sign up
- Google Apps OpenID, Crowd
- Scala, scalaz
- ≅1600 l.o.c.
- 2210 commits Jan 2012 to June 2014
- Top contributors:

```
   765  Josh Graham
   481  Stefan Saasen
   266  Hugh Giddens
   159  Brian McKenna
   142  Eero Kaukonen
```

Atlassian

```scala
trait Fallible[F[+_]] {
  def fail(message: String): F[Nothing]
  def onFailure[A](f: => F[A])(fn: Invalid => Unit): F[A]
}

object Fallible {
  def apply[F[+_]: Fallible] = implicitly[Fallible[F]]

  implicit def liftedFallible[FT[_[+_], +_], F[+_]](implicit FT: Hoist[FT], F: Fallible[F], FM: Monad[F]) =
               new Fallible[({ type λ[+ α] = FT[F, α] })#λ] {

    def fail(message: String) = FT.liftM(F.fail(message))

    def onFailure[A](f: => FT[F, A])(fn: Invalid => Unit) =
      FT.hoist(new (F ~> F) {
        def apply[B](v: F[B]) = F.onFailure(v)(fn)
      }).apply(f)
  }

  implicit object eitherFallible extends Fallible[({ type λ[+α] = Invalid \/ α })#λ] {

    def fail(message: String) = Invalid.Message(message).left

    def onFailure[A](f: => Invalid \/ A)(fn: Invalid => Unit) = f.leftMap(_ <| fn)
  }

  implicit def eitherTFallible[F[+_]: Applicative] =
               new Fallible[({ type λ[+ α] = EitherT[F, Invalid, α] })#λ] {

    def fail(message: String) = EitherT(Invalid.Message(message).left.point[F])

    def onFailure[A](f: => EitherT[F, Invalid, A])(fn: Invalid => Unit) = f.leftMap(_ <| fn)
  }
}
```

Architectural Principles

- No More Monoliths

Architectural Principles

- No More Monoliths
- Code for Failure

Architectural Principles

- No More Monoliths
- Code for Failure
- API is a Feature

Architectural Principles

- No More Monoliths
- Code for Failure
- API is a Feature
- Can You Replace It?

Atlassian

@delitescere

Architectural Principles

- No More Monoliths
- Code for Failure
- API is a Feature
- Can You Replace It?
- Don’t Rush Innovation

Atlassian

@delitescere

Remote Plugins (“Connect”)

- Allow “safe” extension of the SaaS offering
- API is first-class artifact
- Integration “on the glass” and backend
- Continuous Deployment
- 24x7 operations

Project Jeffersons

- Lower the SaaS price point
- World-class SaaS
- Decompose the monoliths

Atlassian

@delitescere

Project Prometheus

- It's a PaaS
- Microservices
- 1st-class REST
- Continuous Deployment
- Version-free Compatibility
- Branch-by-abstraction / Feature flags
- Immutable instances (code, config)
- Anti-fragile
- Engineering ownership in production
- Same on workstation, CI agent, production

Atlassian

@delitescere

On-premises + SaaS

- Simple code base
    - Single programming model, but
        - Multiple deployment topologies
- Server, Data Center, Cloud
- Deep, frictionless integration

Image credits

- Gojira http://commons.wikimedia.org/wiki/File:Gojira_1954_poster_3.jpg
  (public domain, except USA)

- Hydra http://www.deviantart.com/art/hydra-184405674
  (used with permission)

- Pantheon http://commons.wikimedia.org/wiki/File:Pantheon_Cross_section_edited.jpg
  (public domain)

- Rococo palace façade http://commons.wikimedia.org/wiki/File:Architecture_Minya.jpg
  (public domain)

- Tribbles http://en.wikipedia.org/wiki/Tribble#mediaviewer/File:ST_TroubleWithTribbles.jpg
  (Wikipedia fair use)

Godzilla, Hydra, and Tribbles

Evolving a late 90's Java App to Cloud-based Microservices

Josh Graham
Atlassian
SaaS Architect
@delitescere

speakerconf.com

vame.me

Your feedback

:)