

Java Architectures

A New Hope

Eberhard Wolff

<http://ewolff.com>

What happens with a talk titled like this?

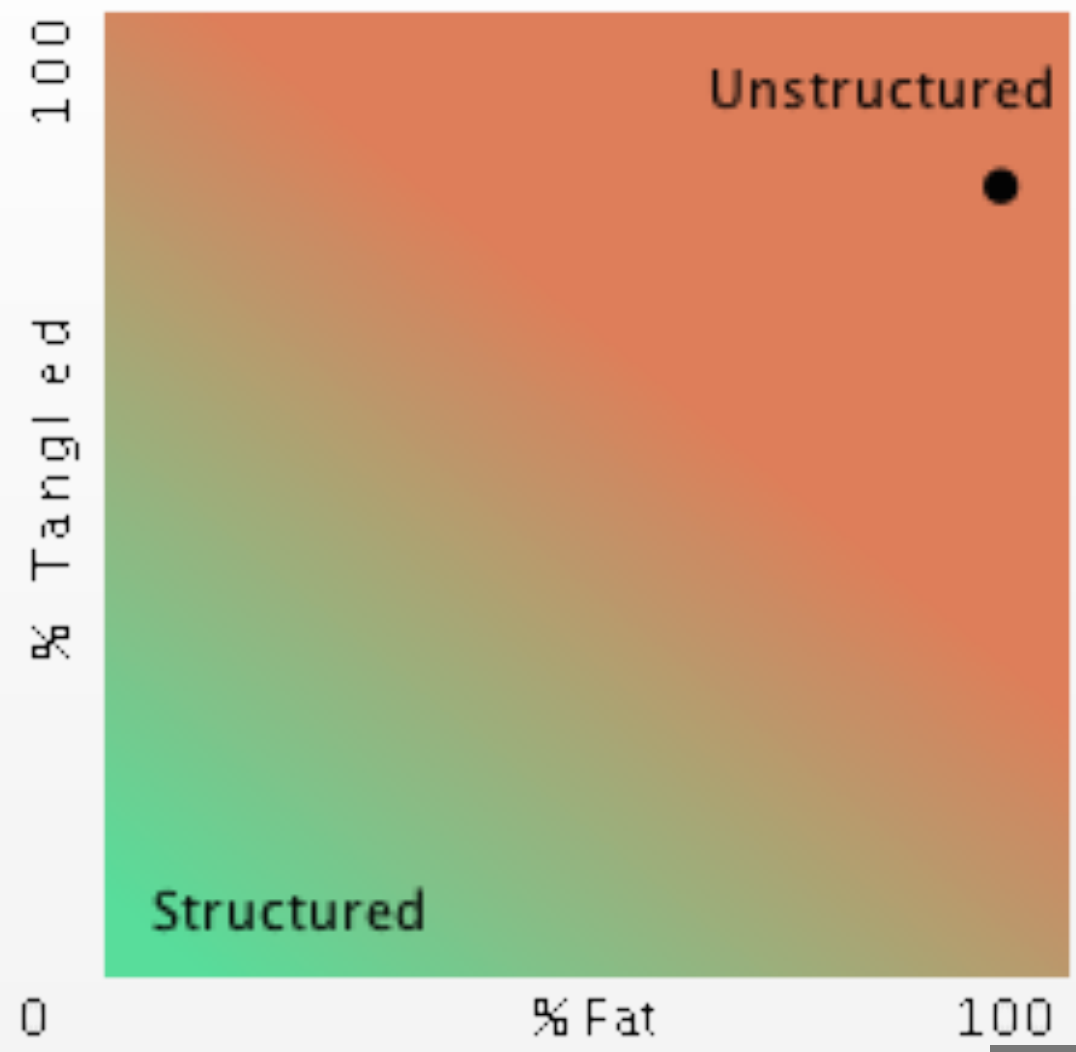


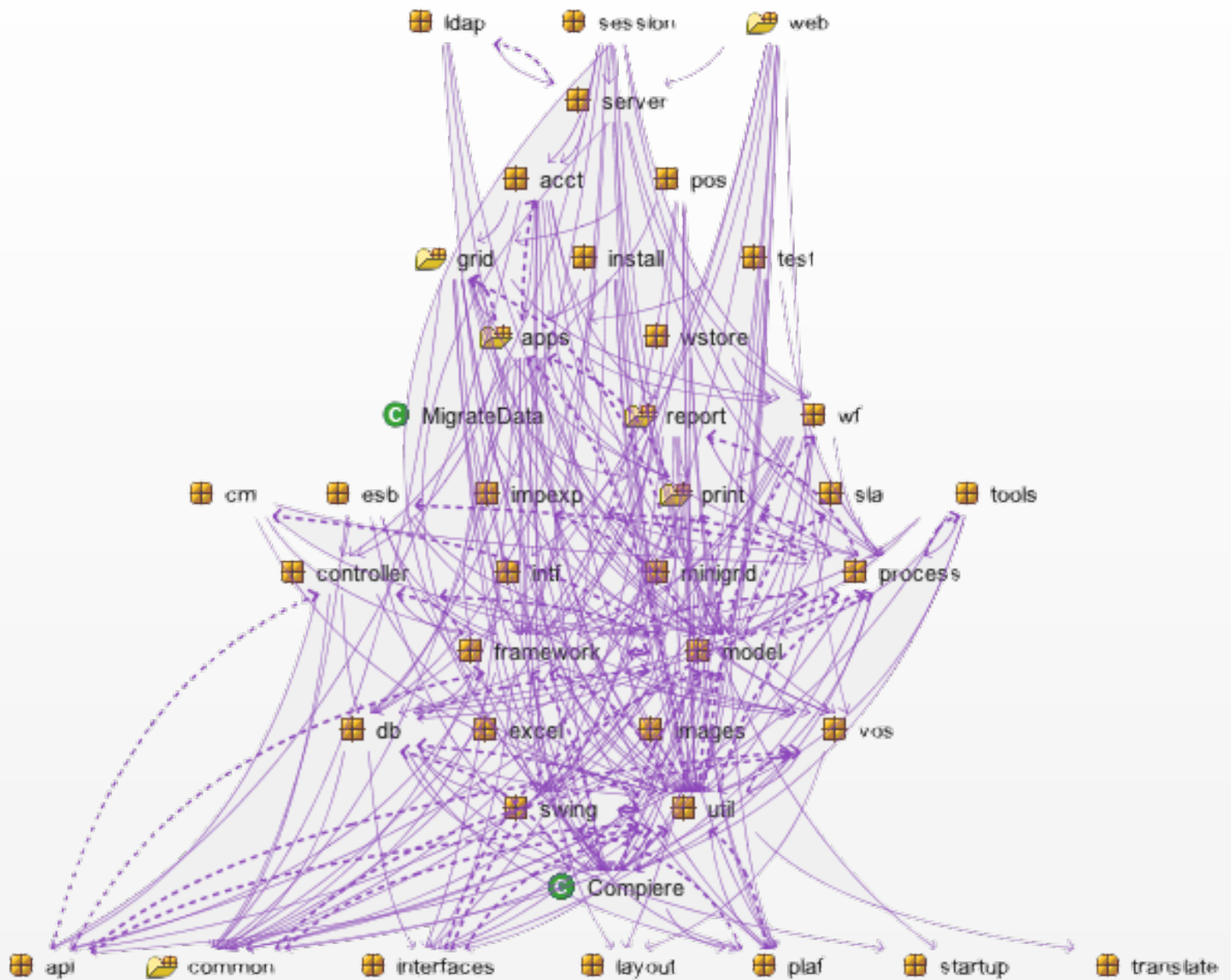
José, Darth Vader and **Darth Vader** followed you



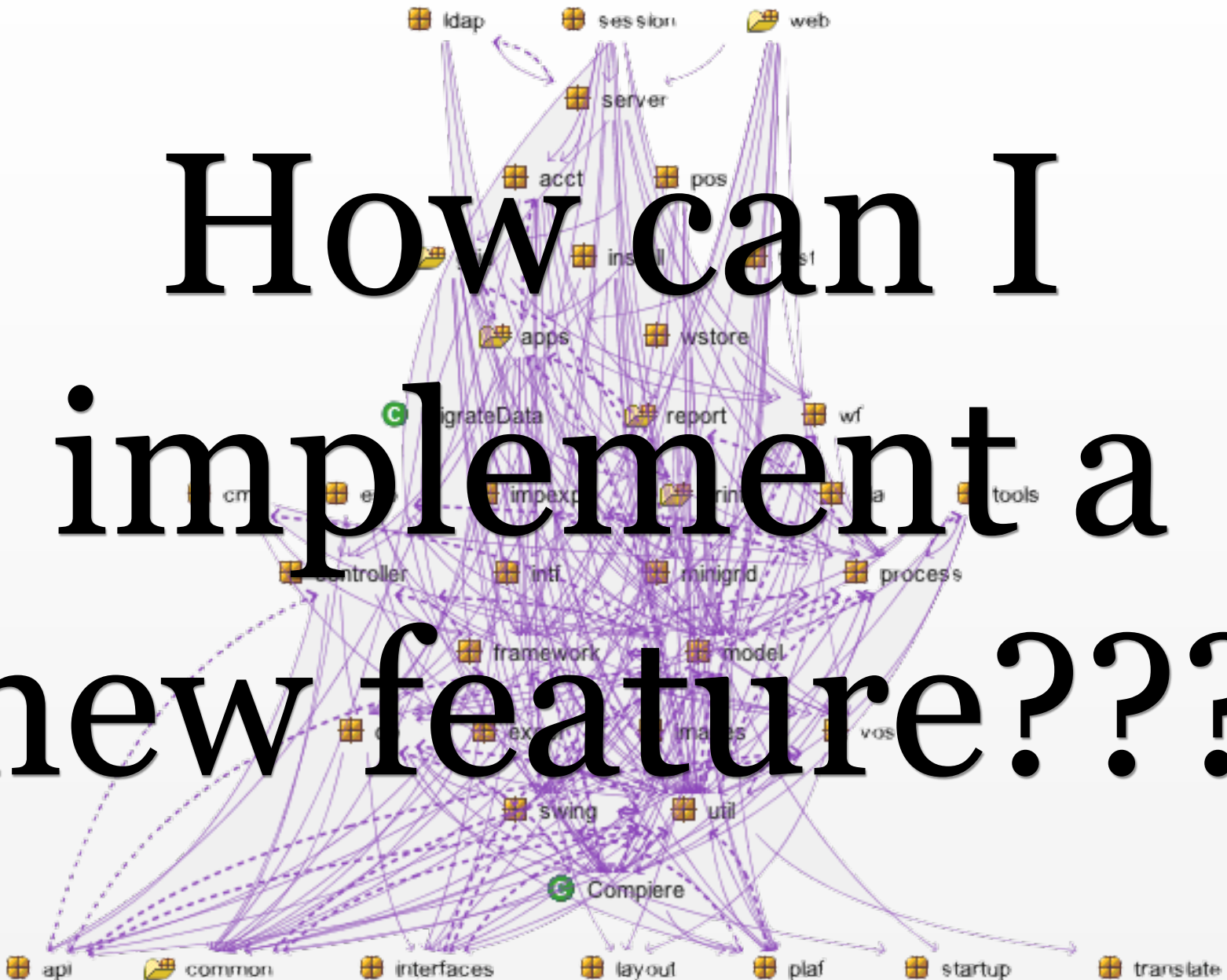
Architecture of Enterprise Java Apps

Top tangles and fat	Size	Problem
 org.compiere	955.512	Tangled
 org.compiere	1.080.0...	Fat



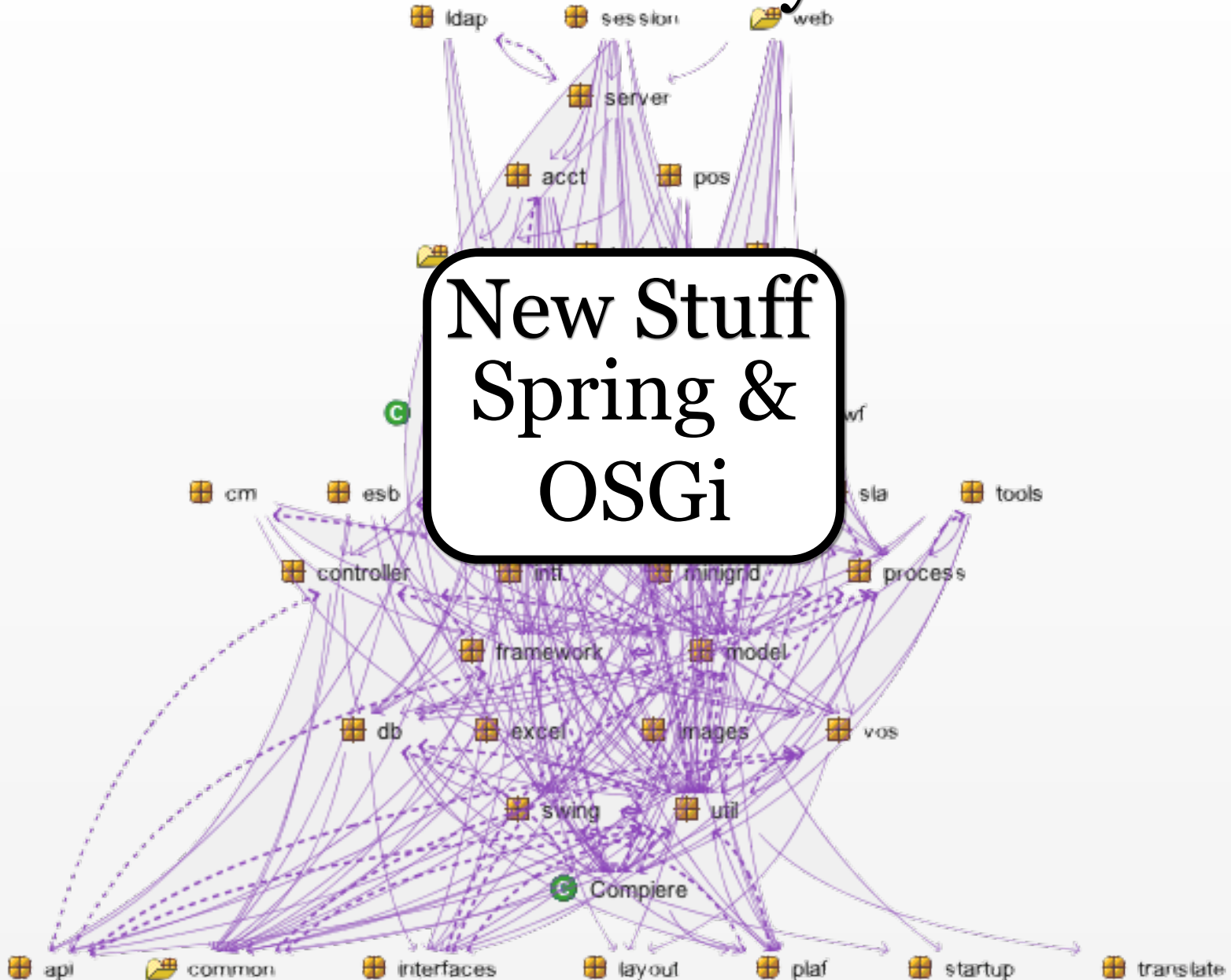


How can I
implement a
new feature???

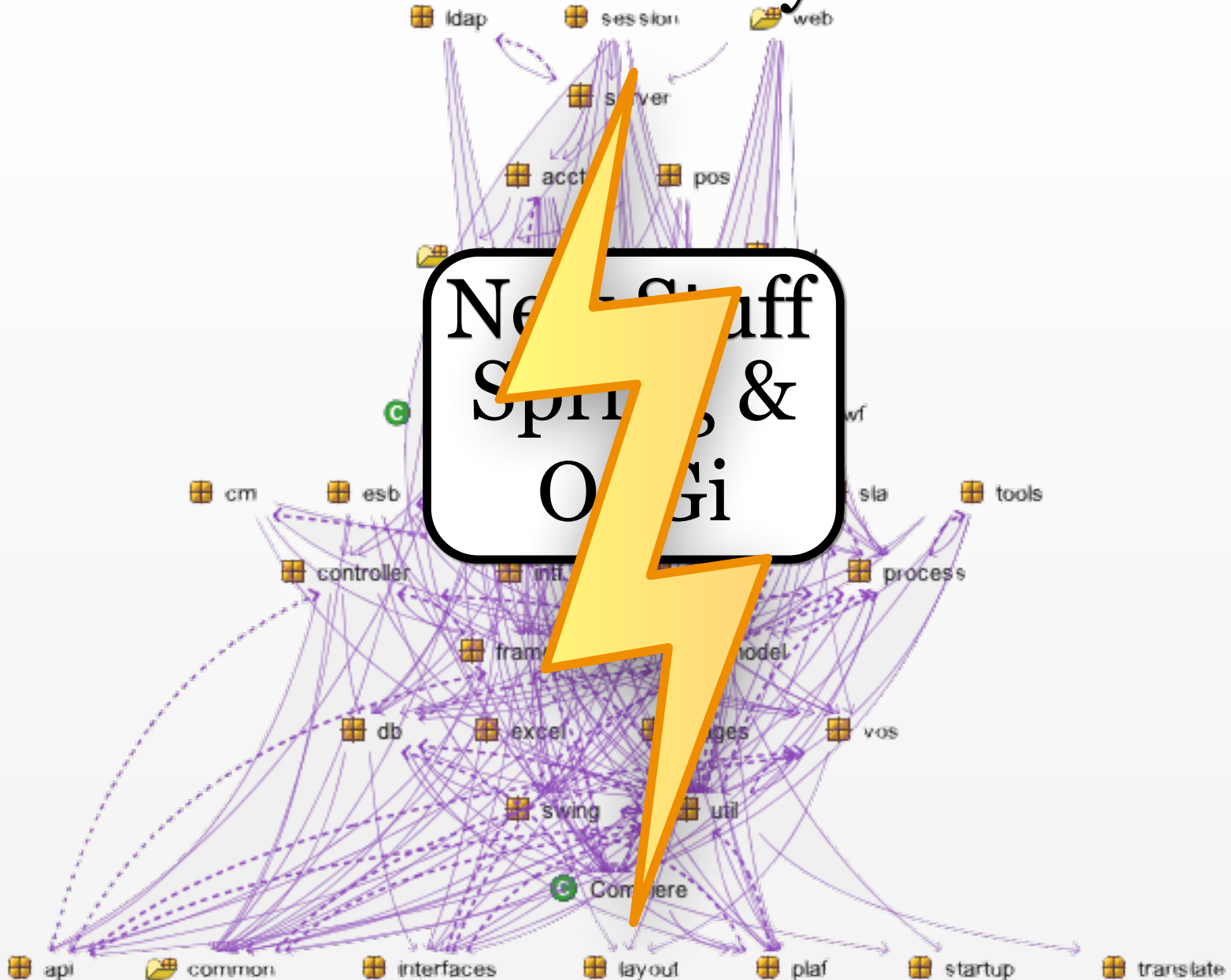


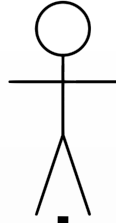


ECommerce System



ECommerce System

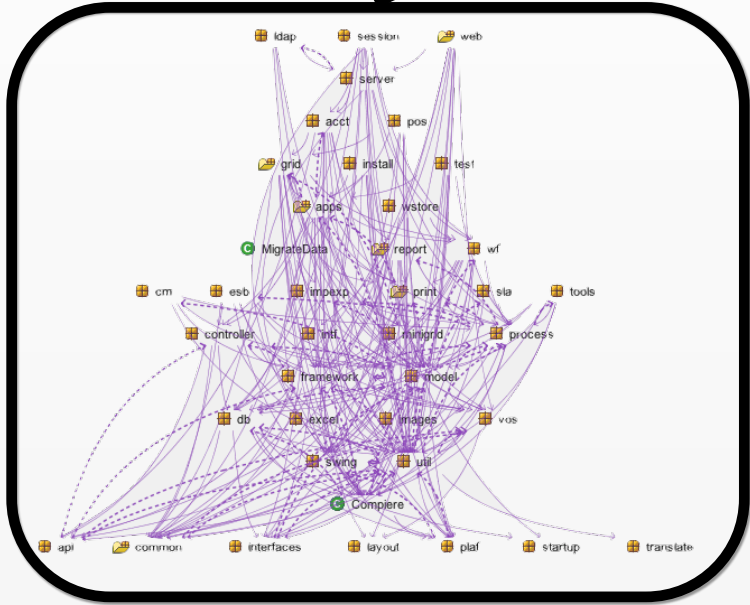




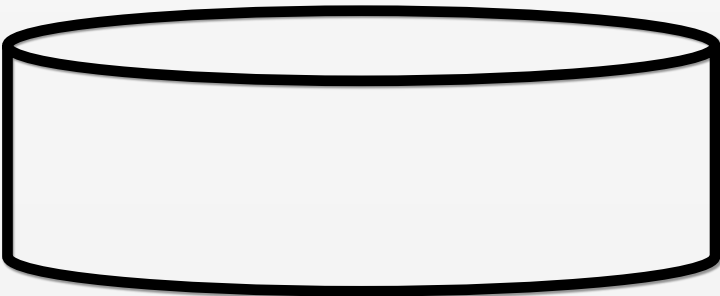
HTTP

Links

No legacy code
Any technology
Small code base



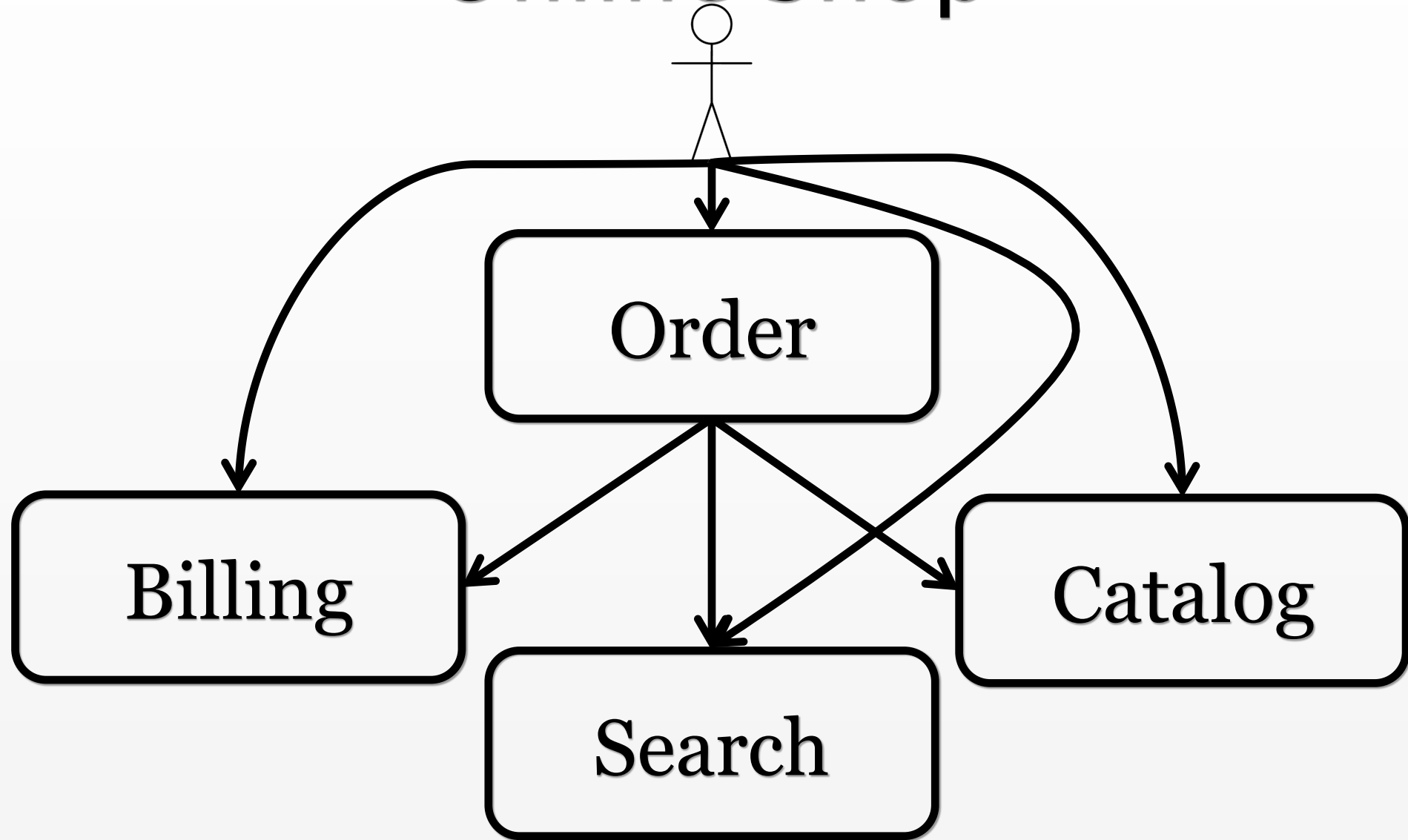
New Stuff



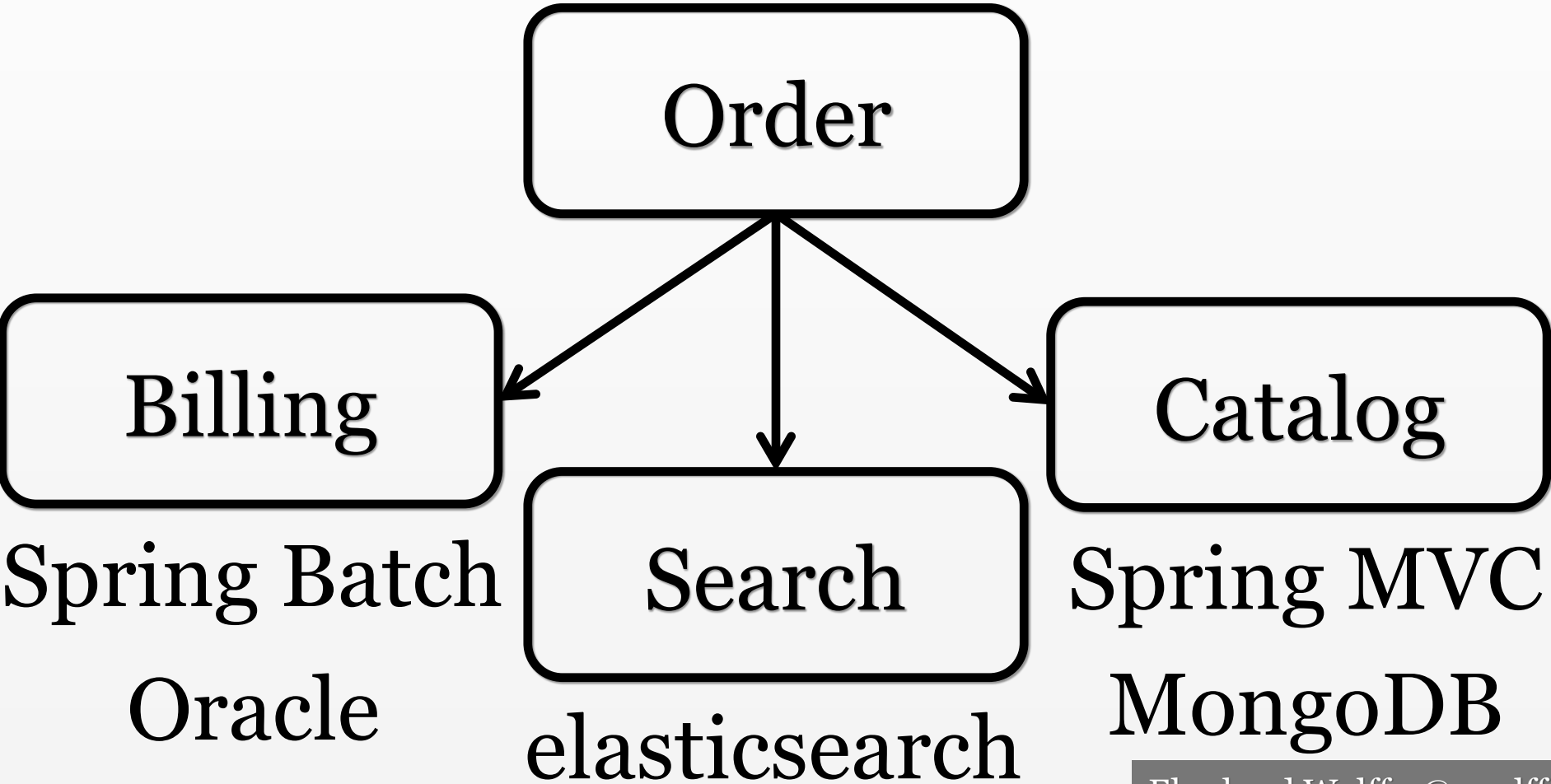
Micro Services

- Component Model
- Component...
- Separate process
- Individual deployment unit
- GUI+Logic
- Small services+GUI – unlike SOA

Online Shop



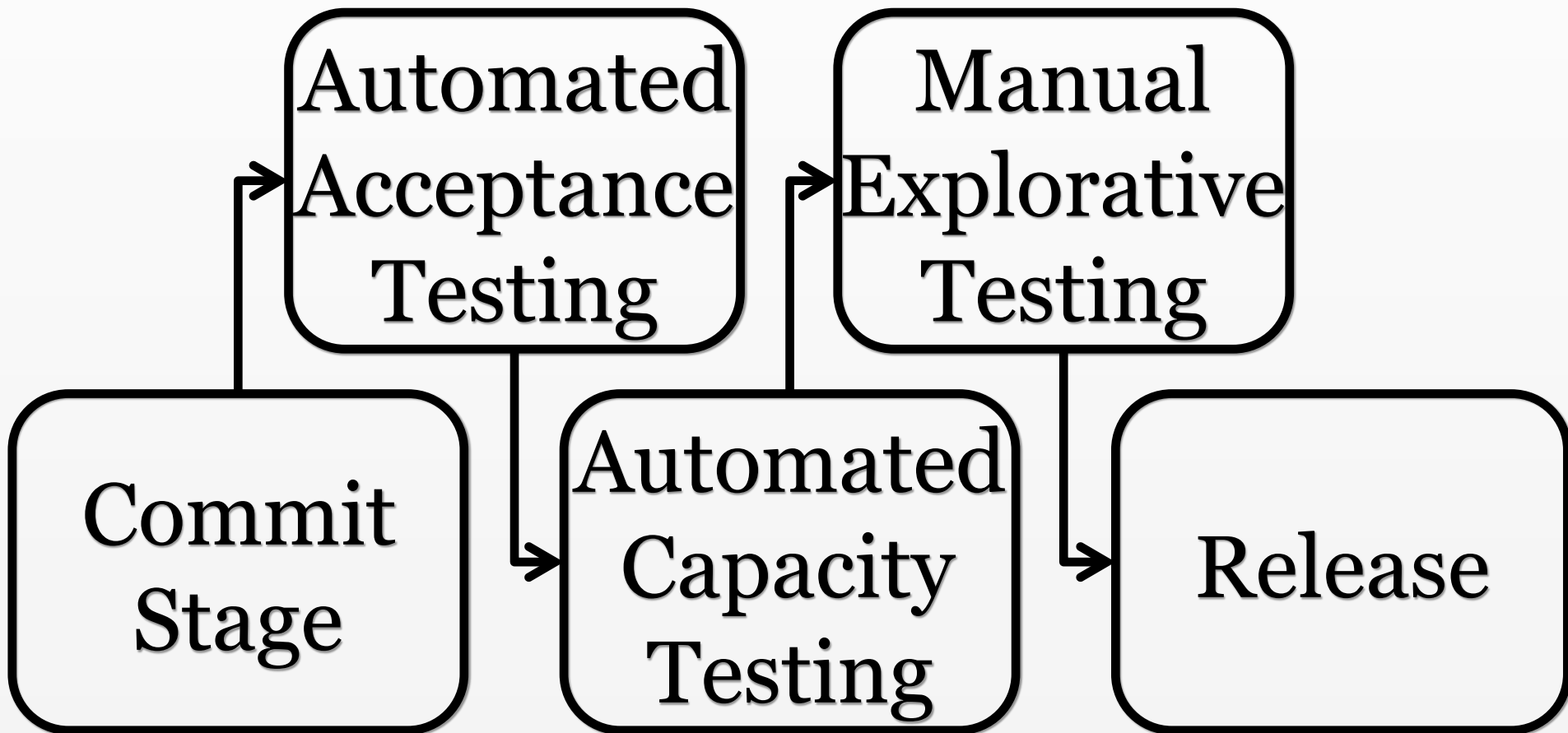
Online Shop



Service might
be even
smaller

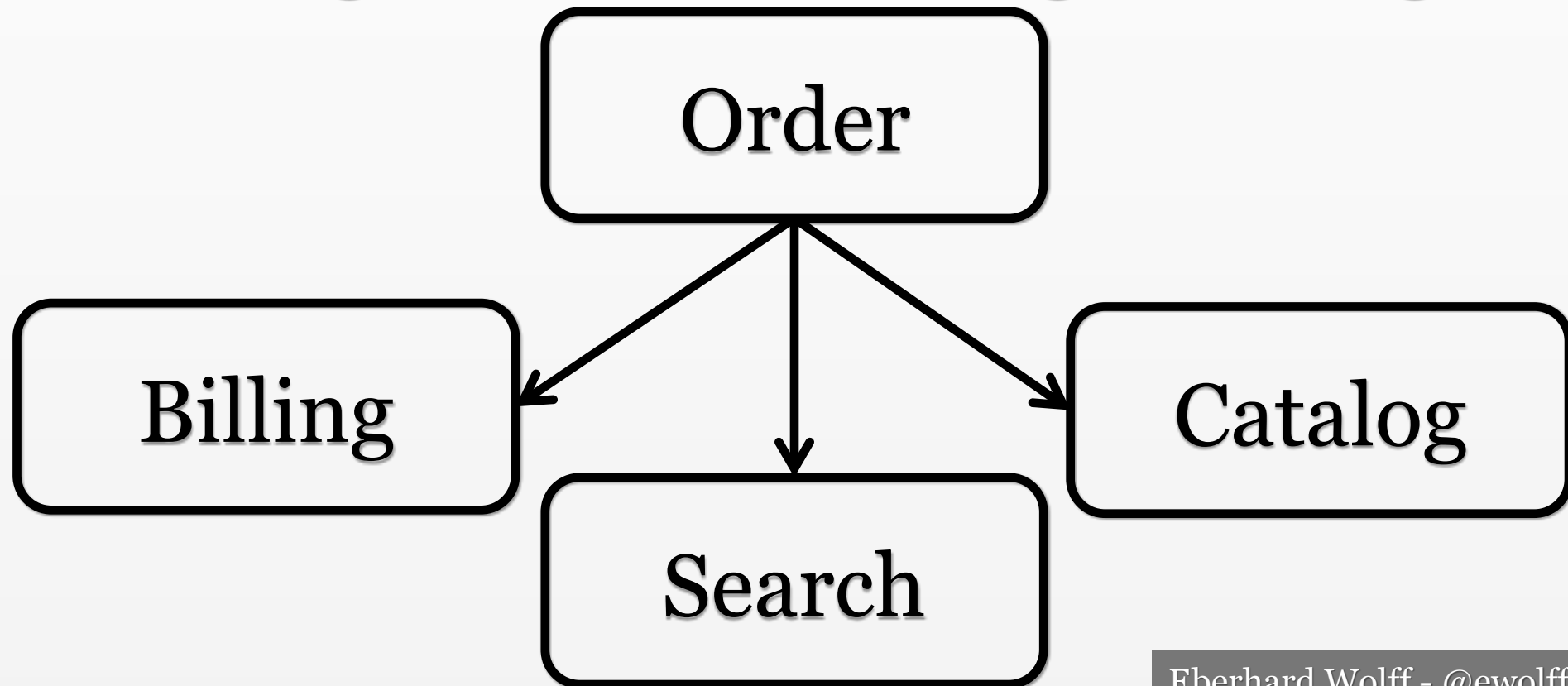
Continuous Delivery

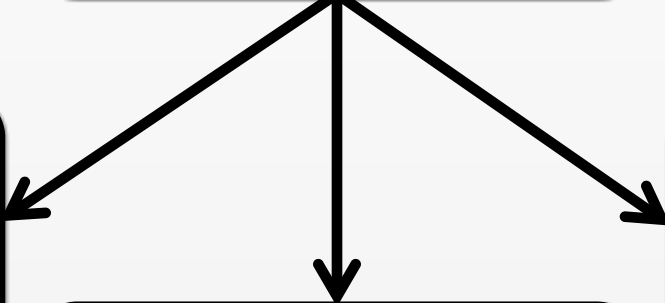
Continuous Delivery: Build Pipeline



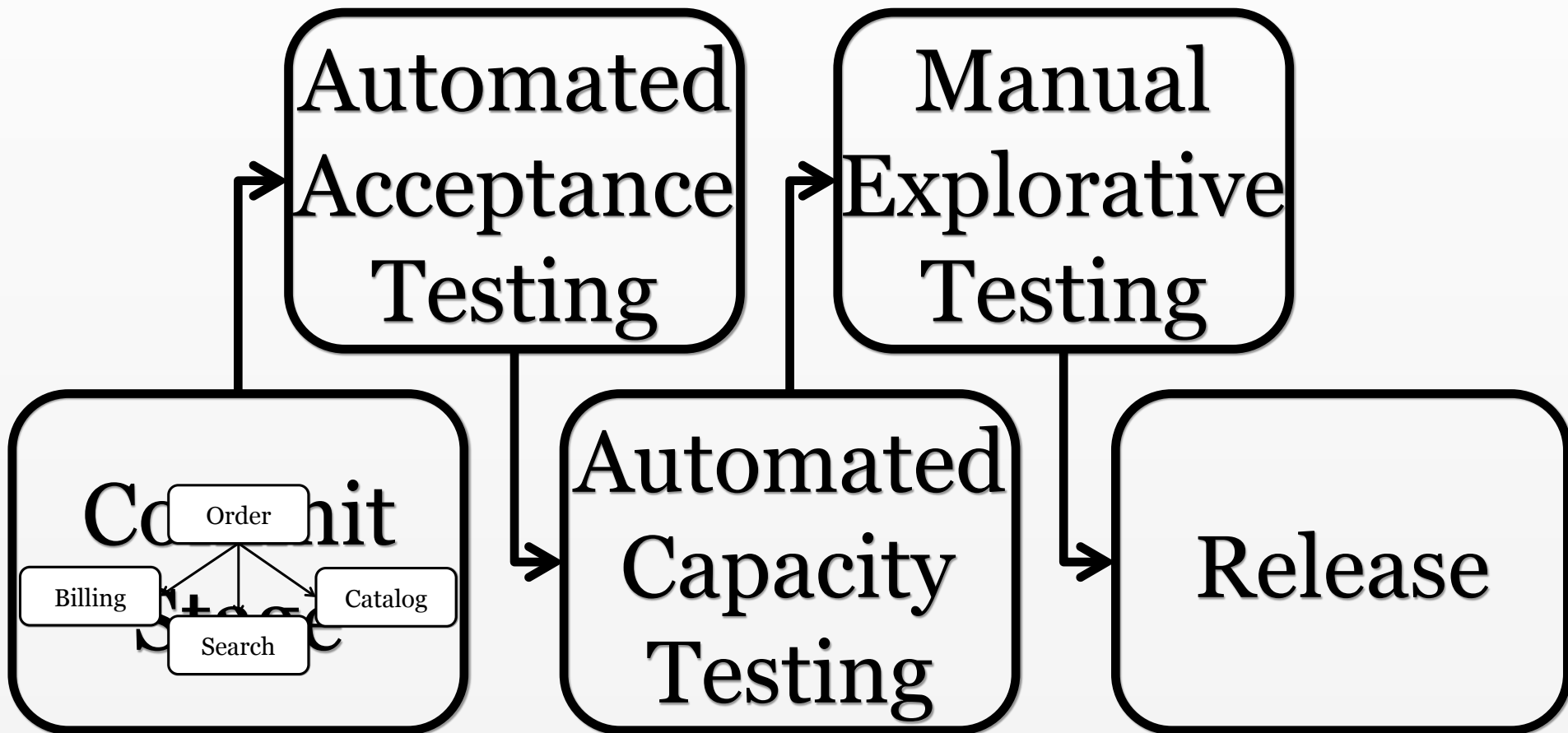
Without Micro Services

- Modify order process slightly
- Billing, Search & Catalog unchanged





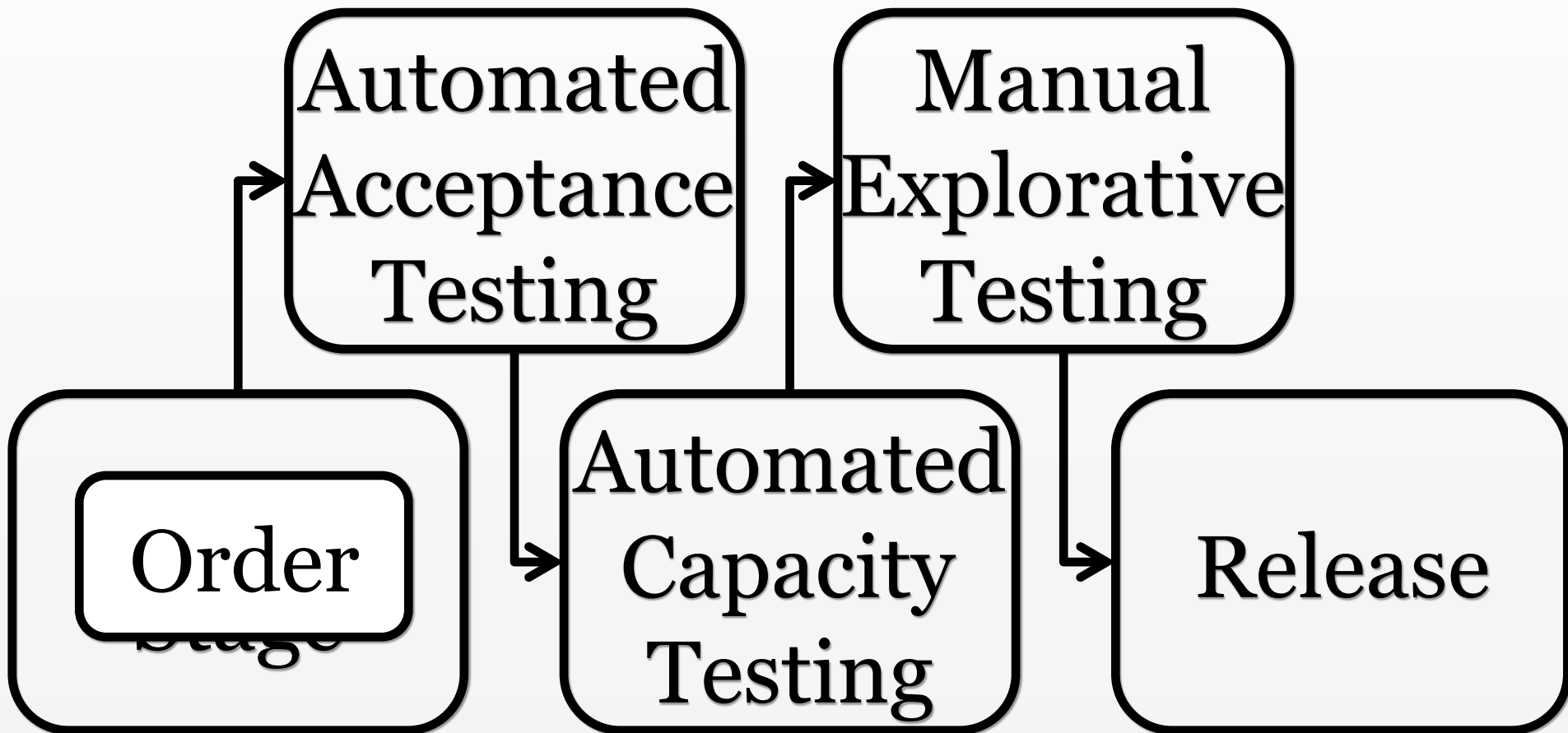
Continuous Delivery: Everything Tested etc



Build Pipeline

- Just one component changed!
- Lots of unneeded work
- Takes much too long

With Micro Services



Release

- Just a single component
- Faster feedback
- Easier deployment
- Rollback of single service also easier

How to scale
agile?

Implement
more feature

Conways Law

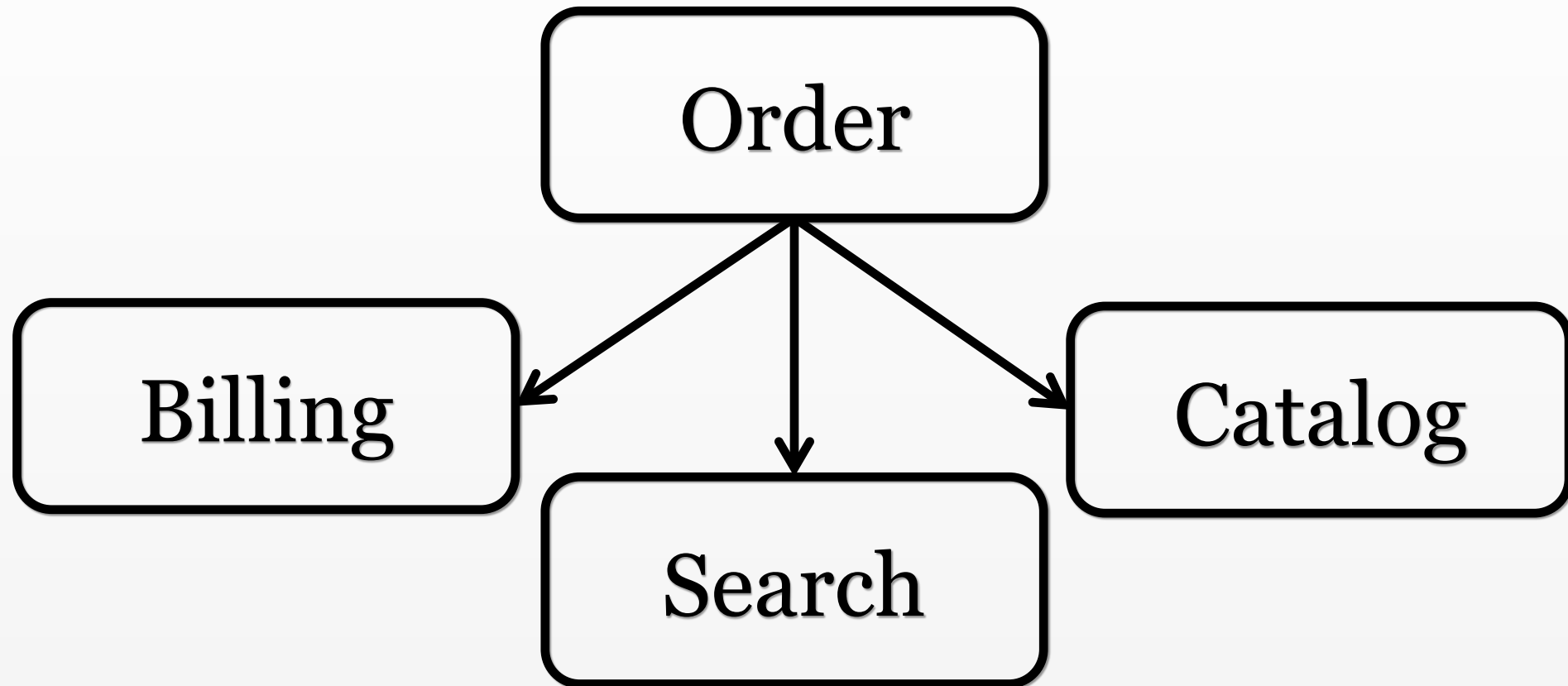
Architecture

copies

communication structures

of the organization

Online Shop



Component = Team

Micro Service

- Team independent from each other
- No code dependencies
- Can use individual technology stack
- Can even deploy independently
- Speed up development

Architecture Challenges

- Code Reuse?
- Handling interfaces?
- Managing dependencies between (>100) Services?
- Global architecture?
- Global refactorings?

Install and
configure App
Server for each
Microservice??

Simple Java
Application?

What about
Operations?

Spring Boot Demo

Why Spring Boot?

- Easier deployment
- i.e. just a JAR

- Built in operations support

Other Technologies

- Dropwizard by Yammer



- vert.x

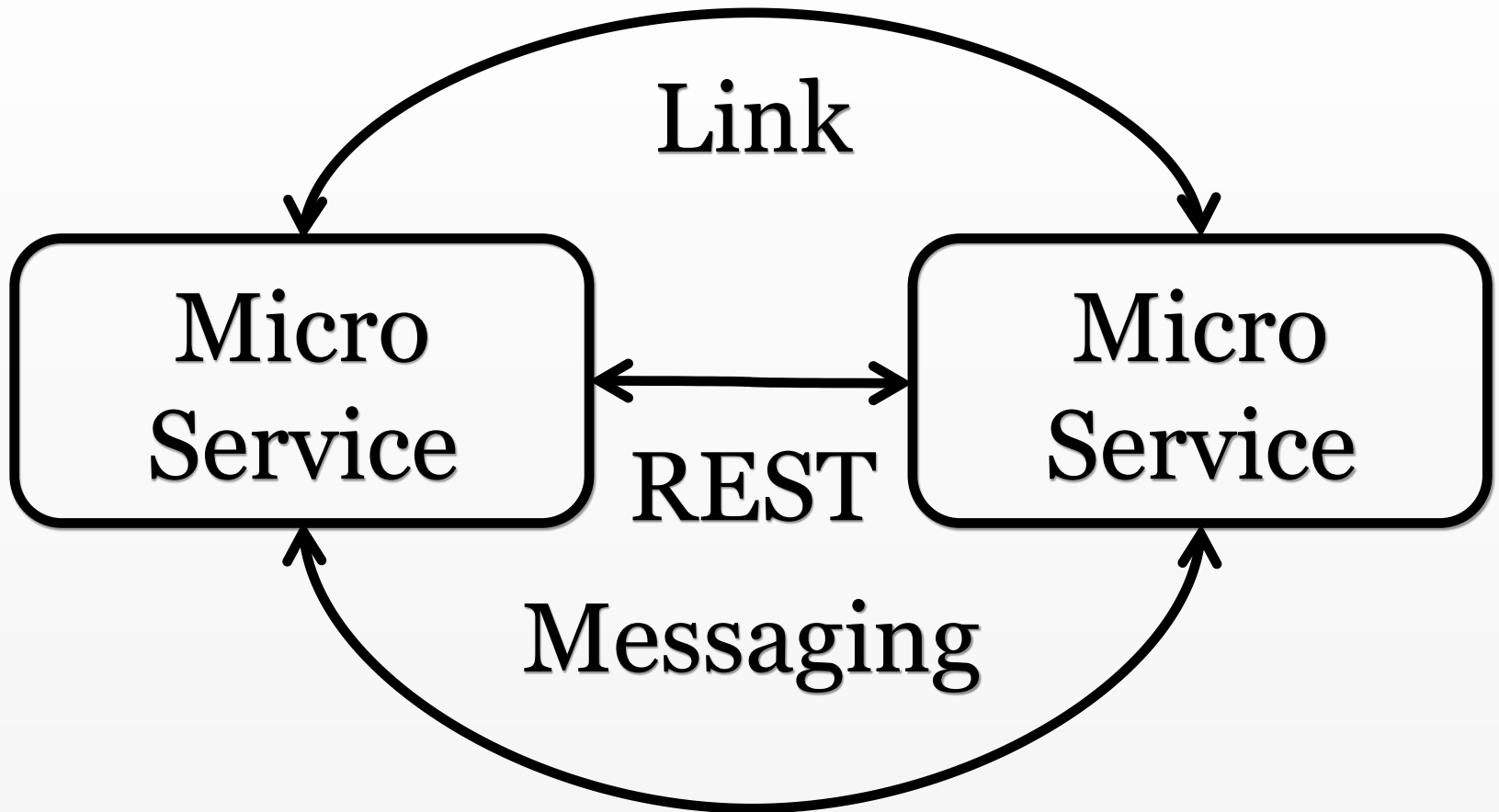
VERT.X

- Play Framework

play 

Micro Services as a Component System

Components Collaborate



Data Replication

Micro Service
can easily
integrate
with Legacy

Deploy &
Operate?

Component Model

- No restriction on language etc
 - Individual processes
 - + infrastructure (database etc)
 - JARs, WARs, EARs: No good fit
-
- Virtual machines as components?
 - Overhead??

Docker

- No true virtualization
- Linux Containers (lxc)
- i.e. shared kernel
- i.e. separate file systems

Docker File Systems

- Read only base images
- +read/write image
- Can be stacked

Read / Write

Application

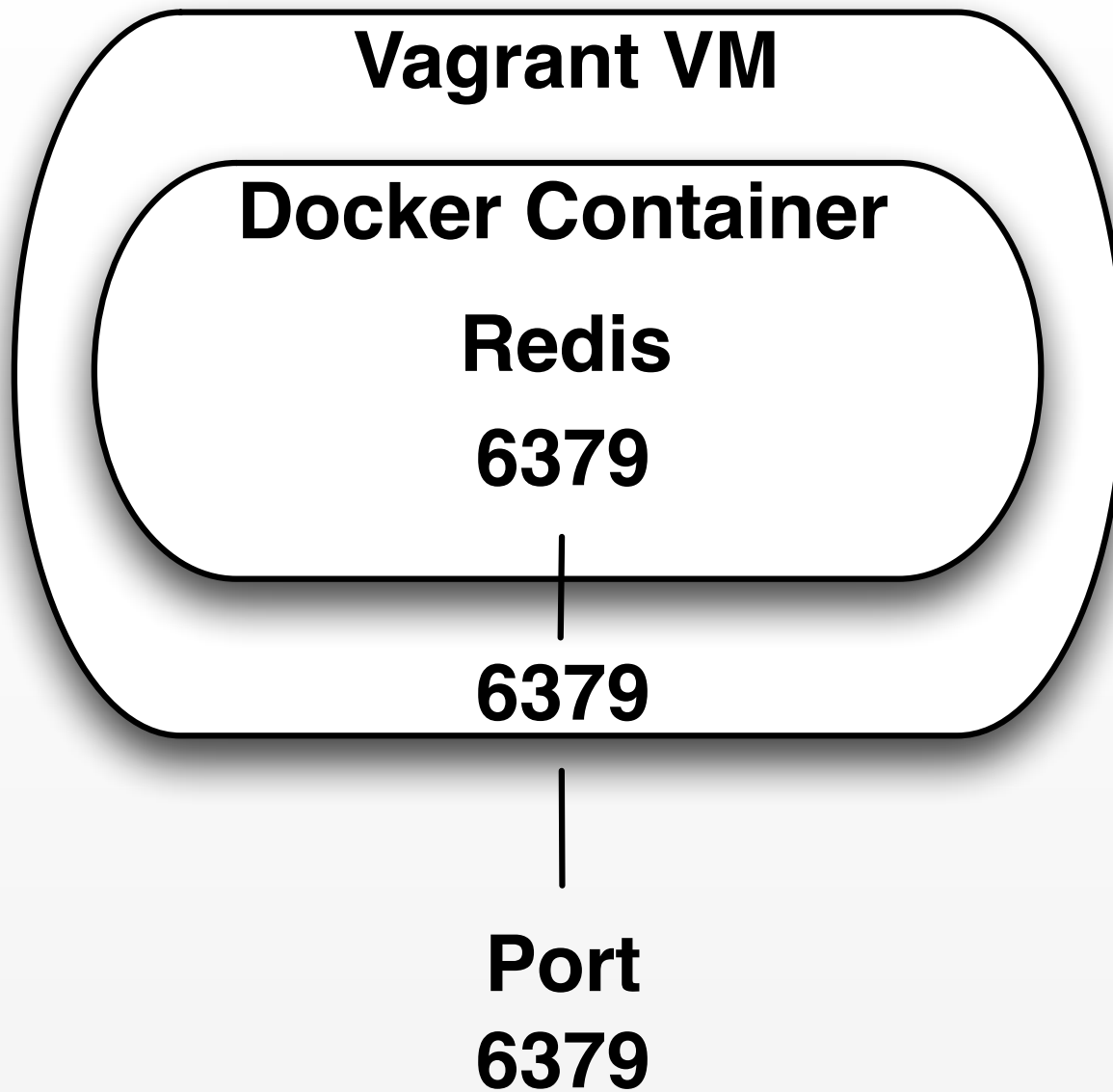
Java

Ubuntu

Docker

- Linux only
- So: Run it in VM
- Controlled by Vagrant

Docker Demo



Communication Between Docker Container

- Via ports
- Via data volumes

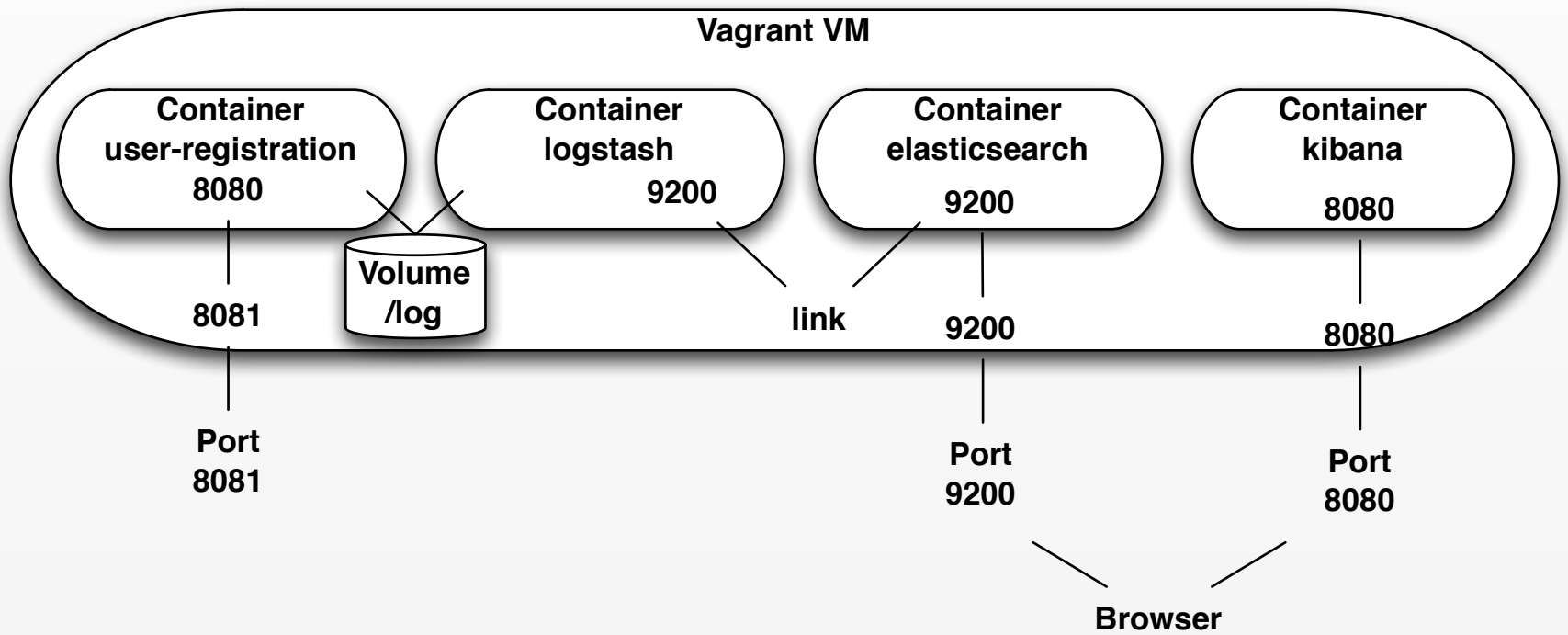
Docker is a
Component
System

Example

- Log file analysis
- ELK Stack
- Elasticsearch: Store
- Logstash: Parse
- Kibana: Web based GUI
- Install three servers
- Automate installation ... ☹️

Dockerized ELK Stack

- 1 Vagrantfile (25 lines)
 - 6 Dockerfiles (total 28 lines)
 - 1 HTML file (11 lines)
 - 1 Config (Logstash) (29 lines)
-
- Total: 93 lines, 9 files
 - Including one RegExp ☹️



Demo ELK Stack

Links

- <https://github.com/ewolff/user-registration>
- <http://www.heise.de/developer/artikel/Episode-44-Modularisierte-Architektur-fuer-grosse-Systeme-2195818.html>
- <http://projects.spring.io/spring-boot/>
- <http://www.docker.com/>
- <http://www.docker.com/tryit>

Android wird andersARTig: Technische Evolution der Plattform ▶ 108

Deutschland €9,80

Dänemark € 10,90 Schweiz sfr 16,50 Luxemburg € 11,10

8.2014

JAVA Mag

Java magazin

Java • Architekturen • Web • Agile

www.javamagazin.de

BDD mit JBehave

Verhaltensregeln für Anwendungen ▶ 101

JSF und Bootstrap

Frechzellenkur für Web-Apps ▶ 86

wjax14

Programminfos
ab Seite 671

MICRO SERVICES

Löst die Silver Bullet unsere Probleme? ▶ 29

Nie wieder Monolithen! Ein Praxisbericht ▶ 36

Hystrix: Mehr Stabilität in verteilten Anwendungen ▶ 44



**JavaFX-Tutorial:
Von der Klasse
zur EXE ▶ 54**

**Java 8 Streams:
Erzeugung und
Operationen ▶ 16**

**Mit OpenStack zur
eigenen Cloud:
Privatsache ▶ 95**

Micro Services

- A new hope against Java Monoliths
- Also great for legacy
- Spring Boot for implementation
- Docker as component model & runtime

Thank You!!

eberhard.wolff@gmail.com
@ewolff