

1.– 4. September 2014  
in Nürnberg



# Herbstcampus

Wissenstransfer  
par excellence

## Kleiner ist besser

Architekturen mit Micro Services

Eberhard Wolff

Freelance Trainer & Consultant / adesso AG

# Micro Services

# Kleiner ist besser

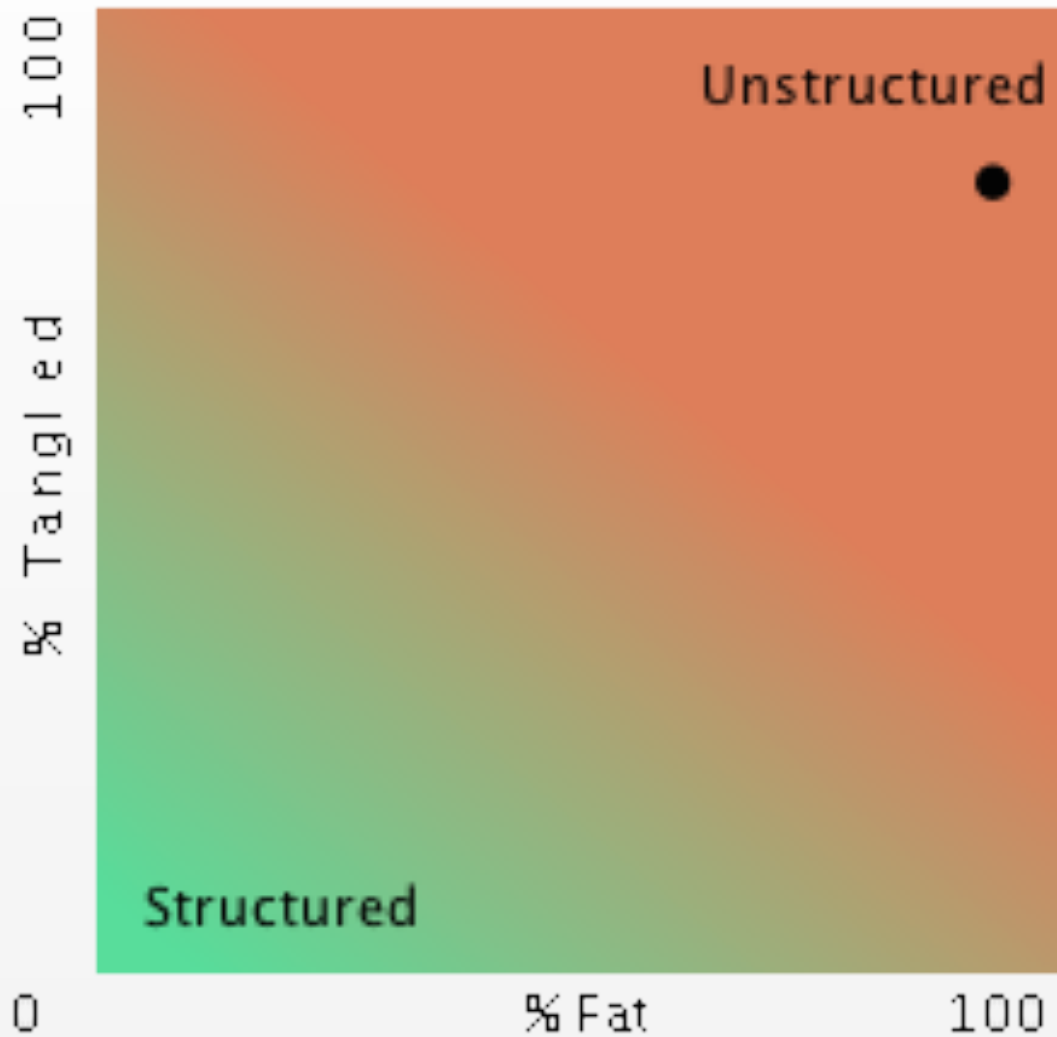
Eberhard Wolff

Freelancer / adesso AG

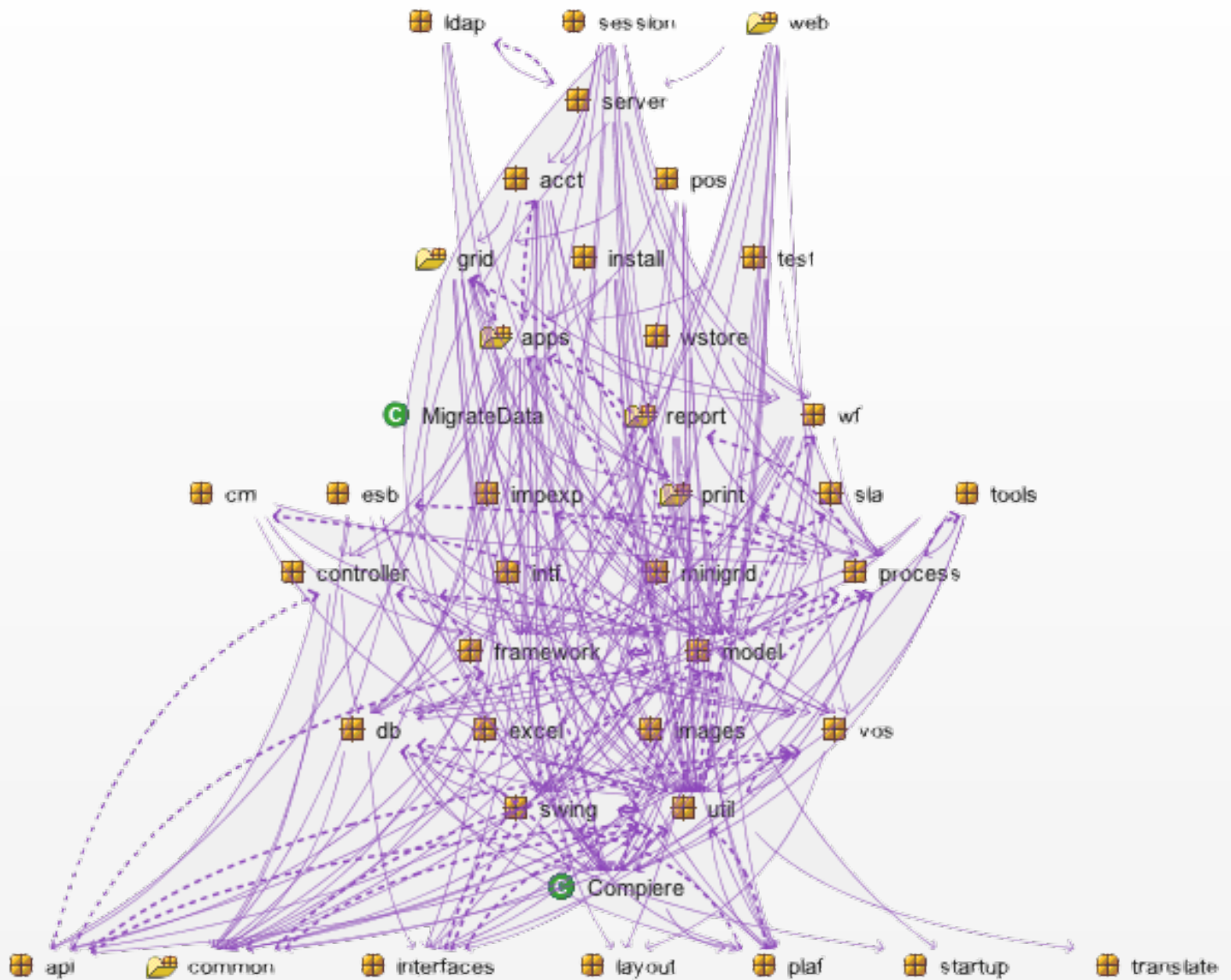
<http://ewolff.com>

# Architecture of Enterprise Java Apps

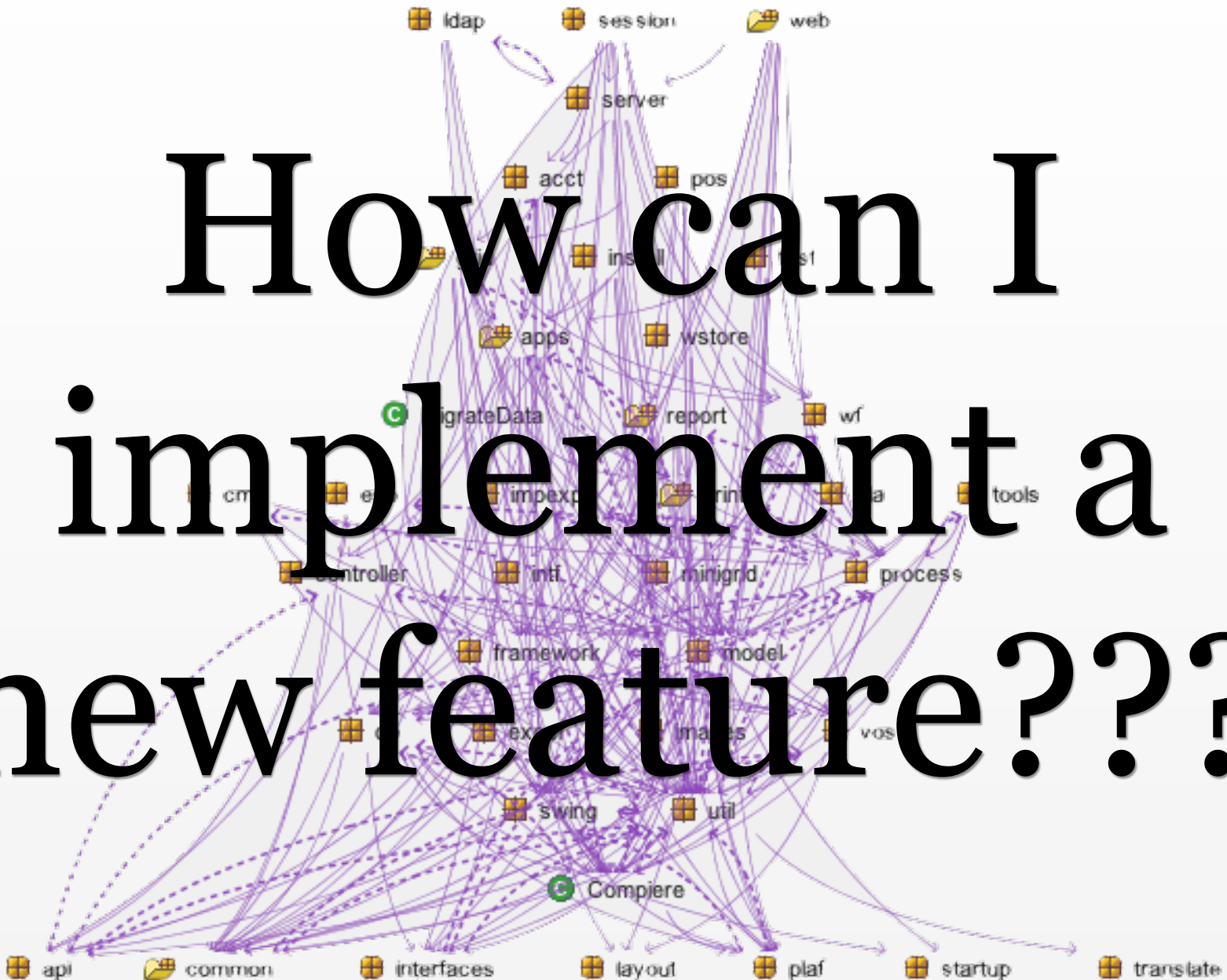
Top tangles and fat	Size	Problem
 org.compiere	955.512	Tangled
 org.compiere	1.080.0...	Fat





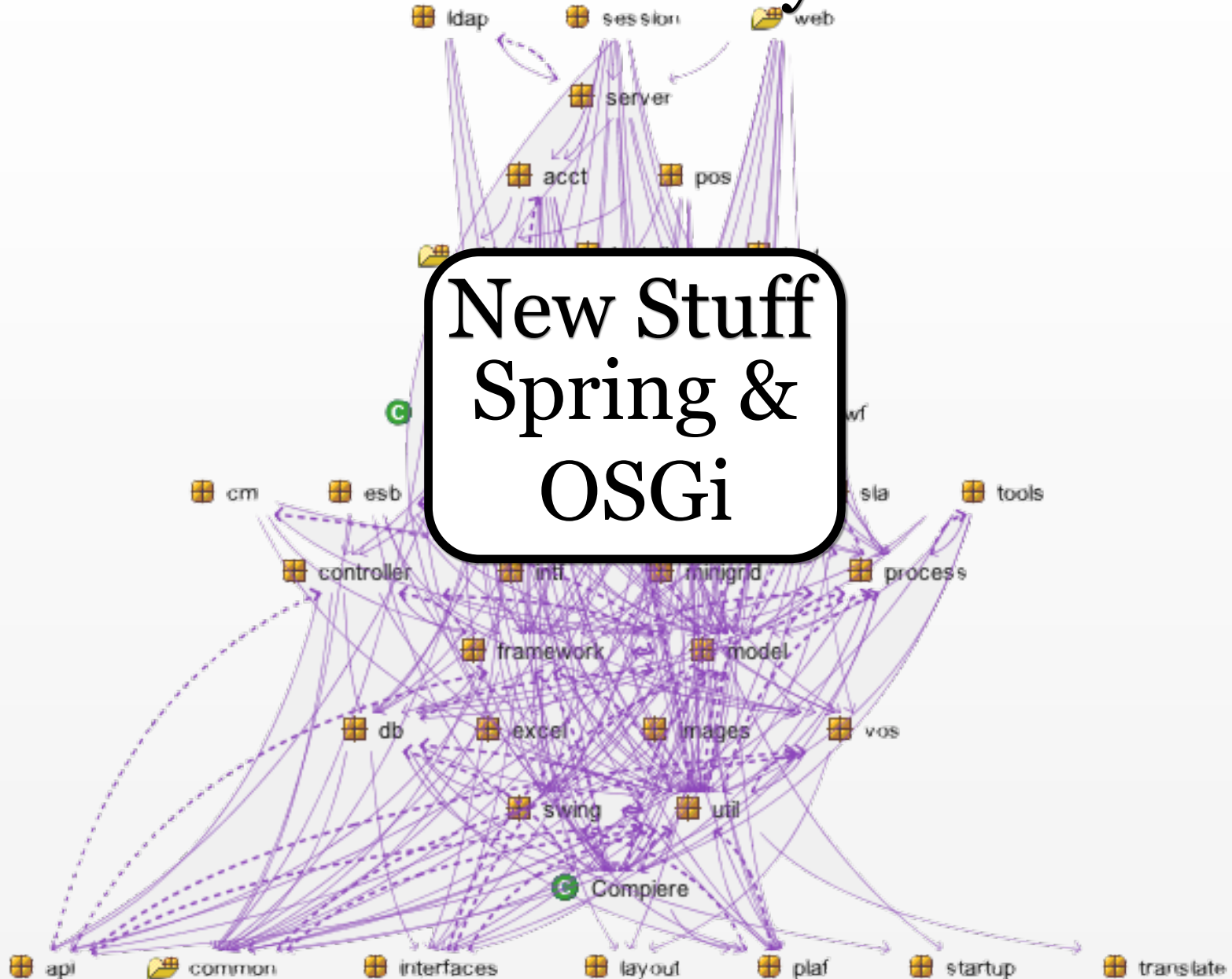


How can I  
implement a  
new feature???

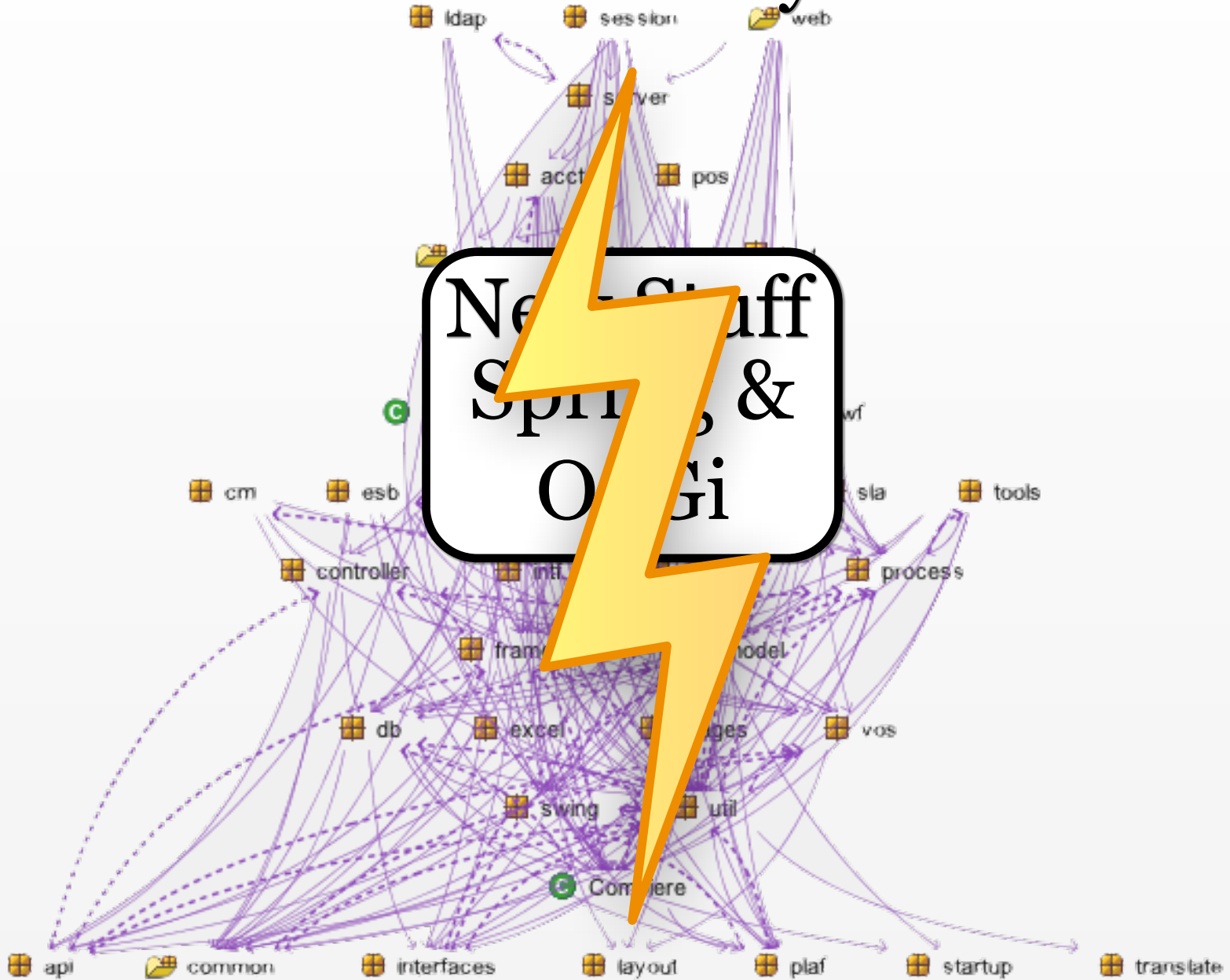




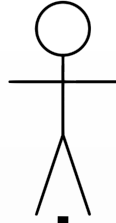
# ECommerce System



# ECommerce System



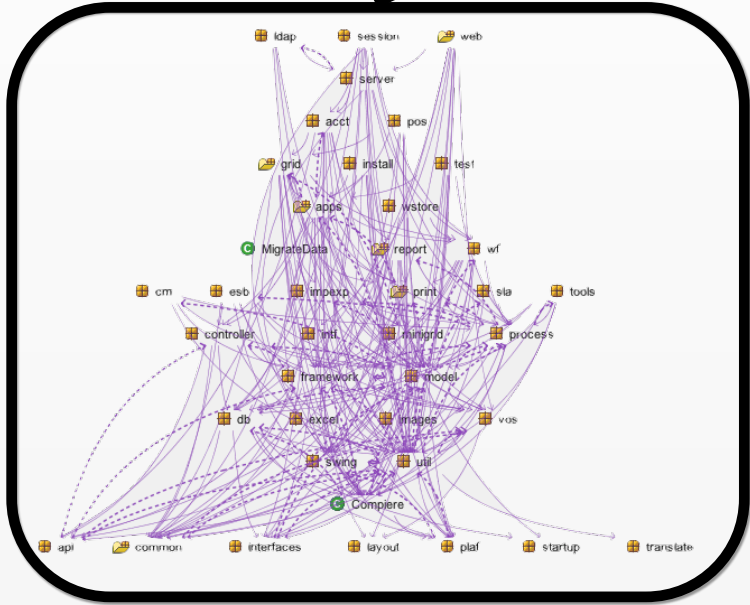




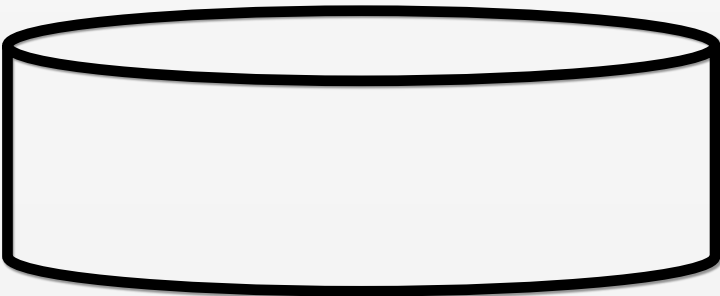
HTTP

Links

No legacy code  
Any technology  
Small code base



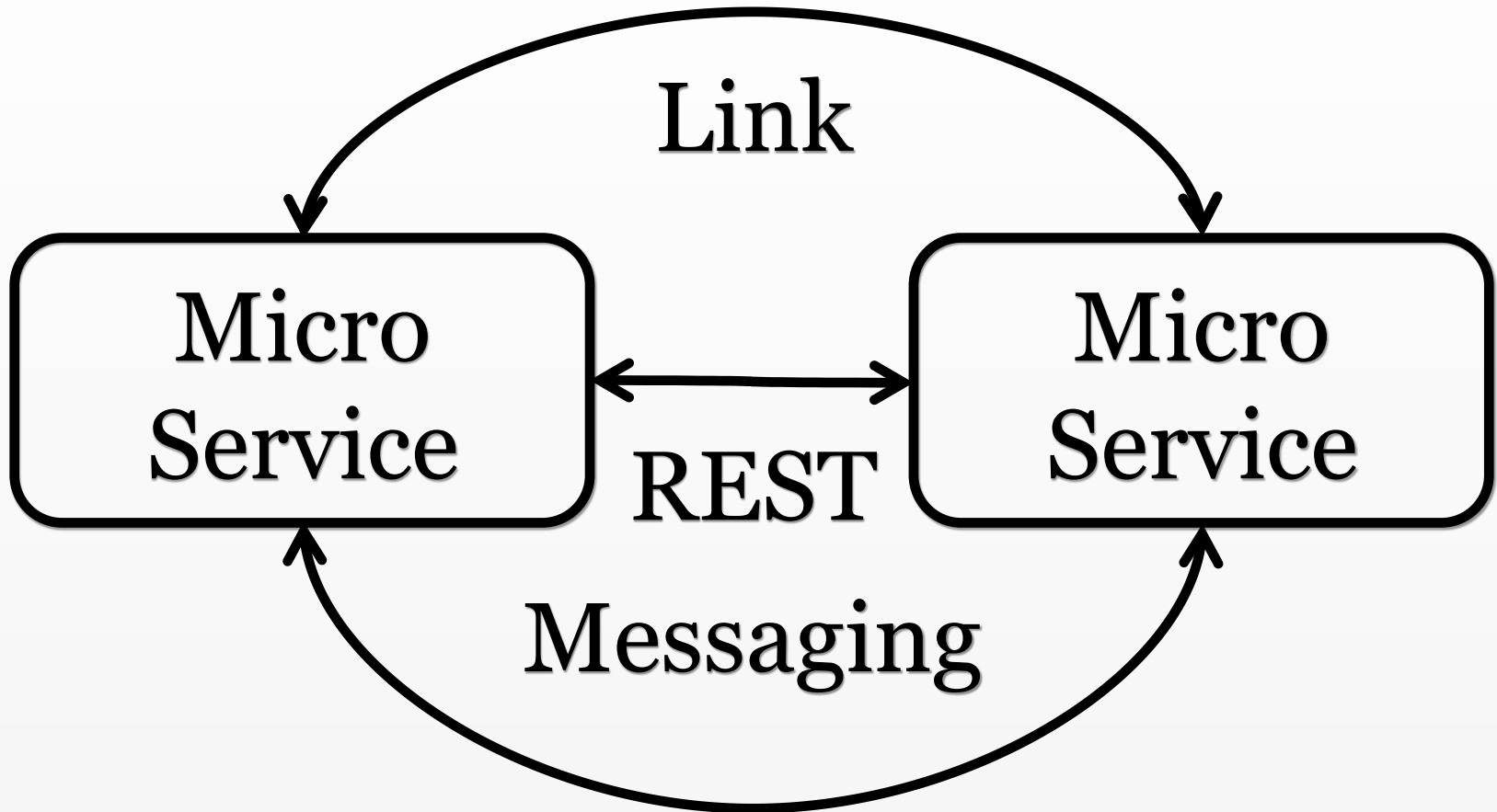
New Stuff



# Micro Services: A Component Model

- Separate process
- Individual deployment unit
- GUI (?)
- + Logic
- + database
- Independent technology stacks
- Small services + GUI – unlike SOA

# Components Collaborate

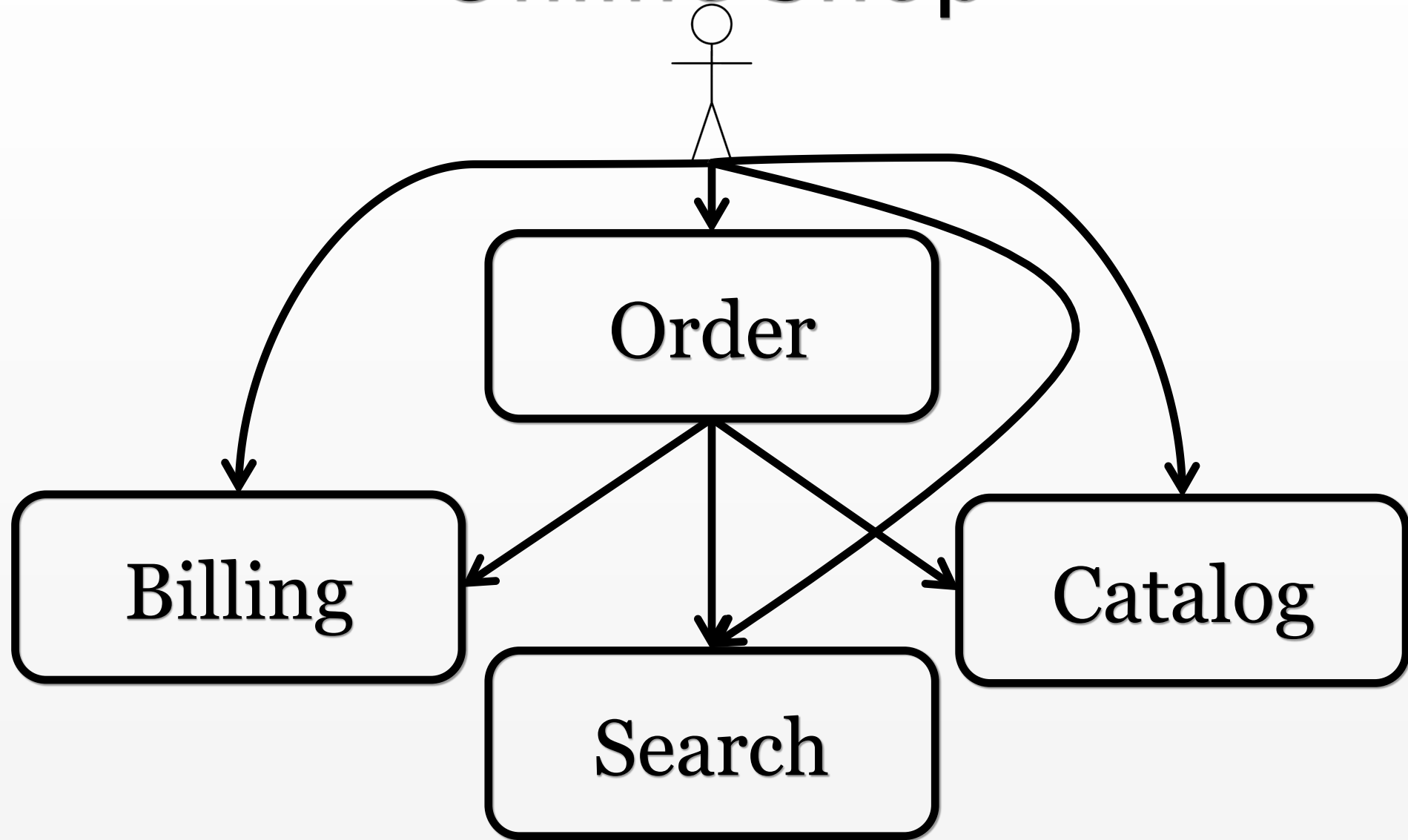


Data Replication

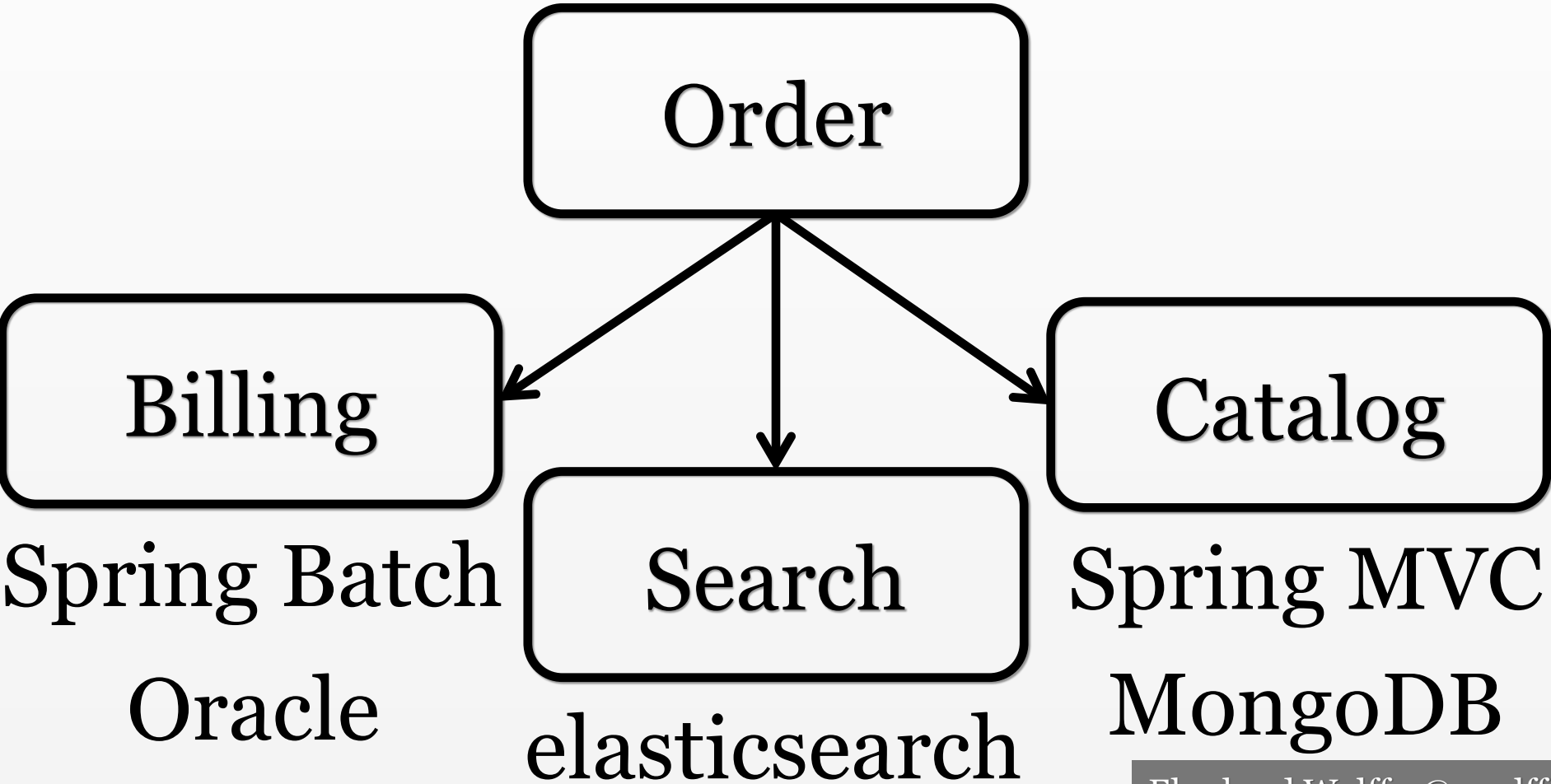


Micro Service  
can easily  
integrate  
with Legacy

# Online Shop



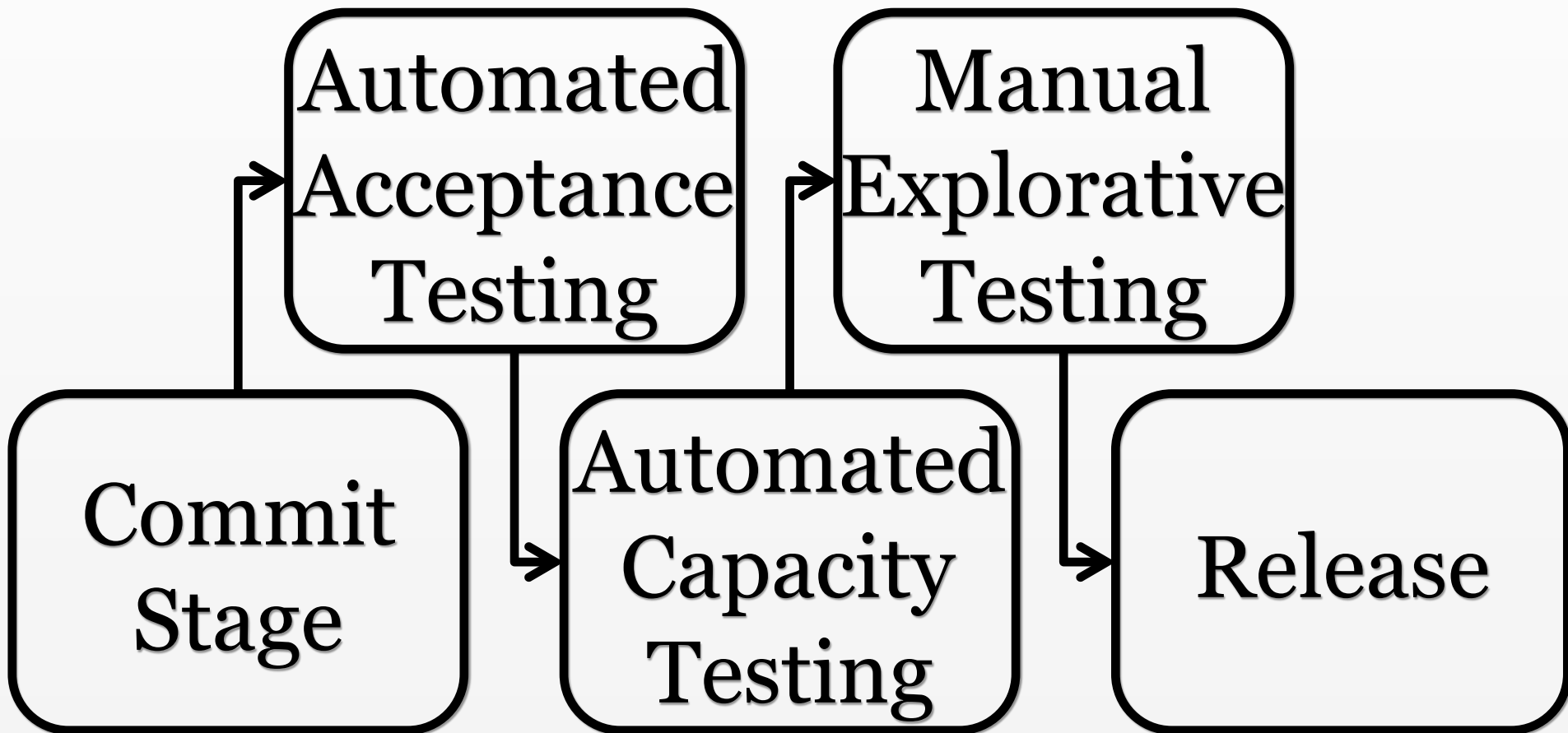
# Online Shop



Service might  
be even  
smaller

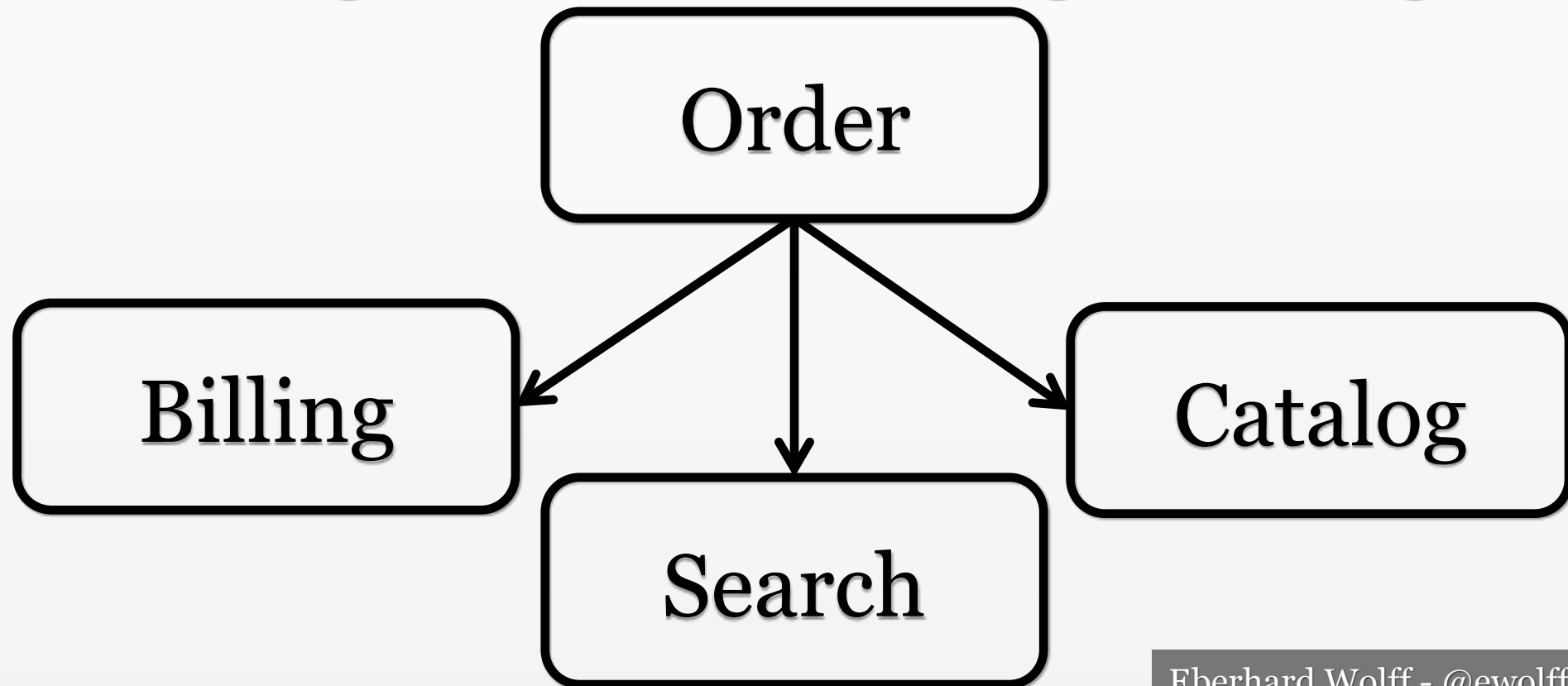
# Continuous Delivery

# Continuous Delivery: Build Pipeline



# Without Micro Services

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



Order



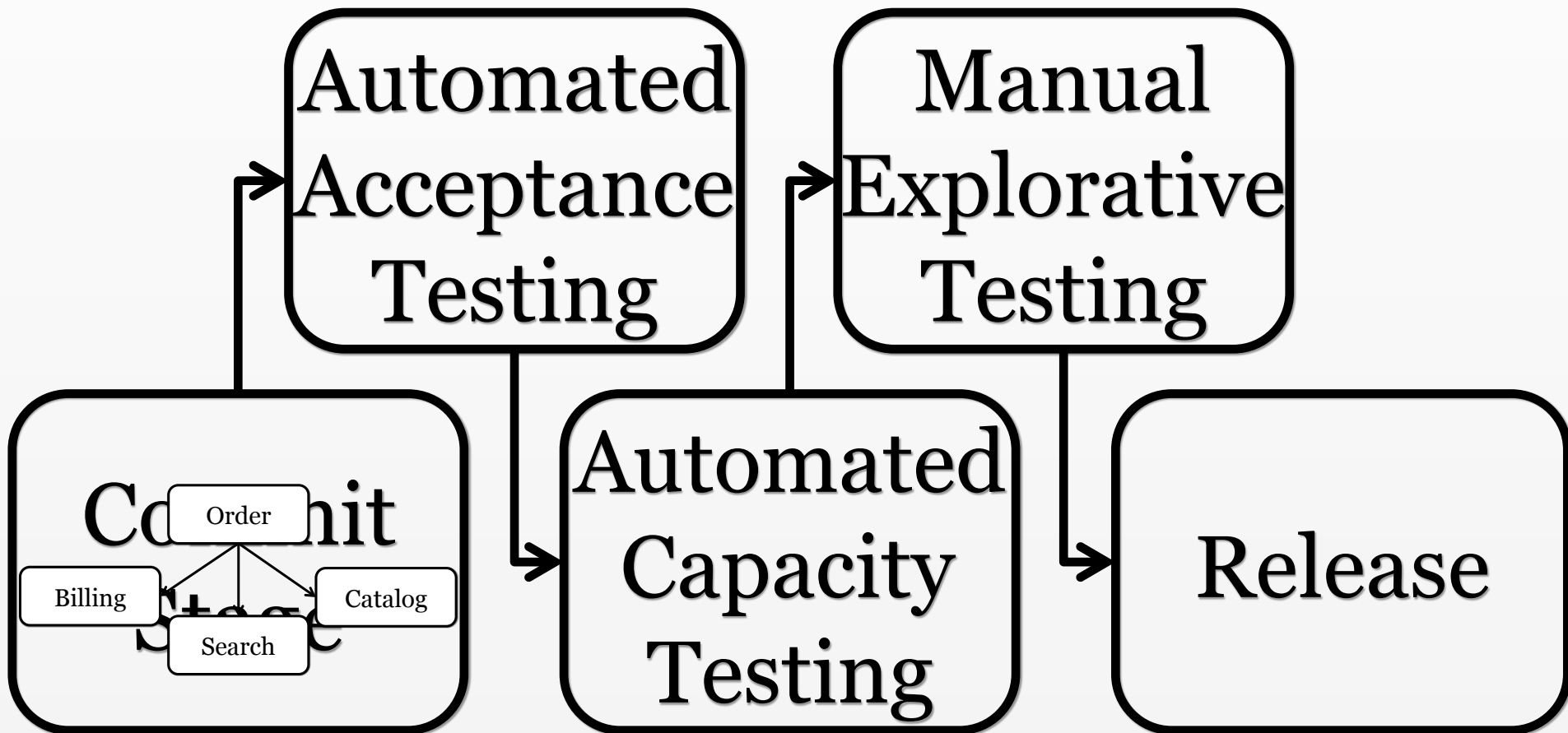
Billing

Search

Catalog



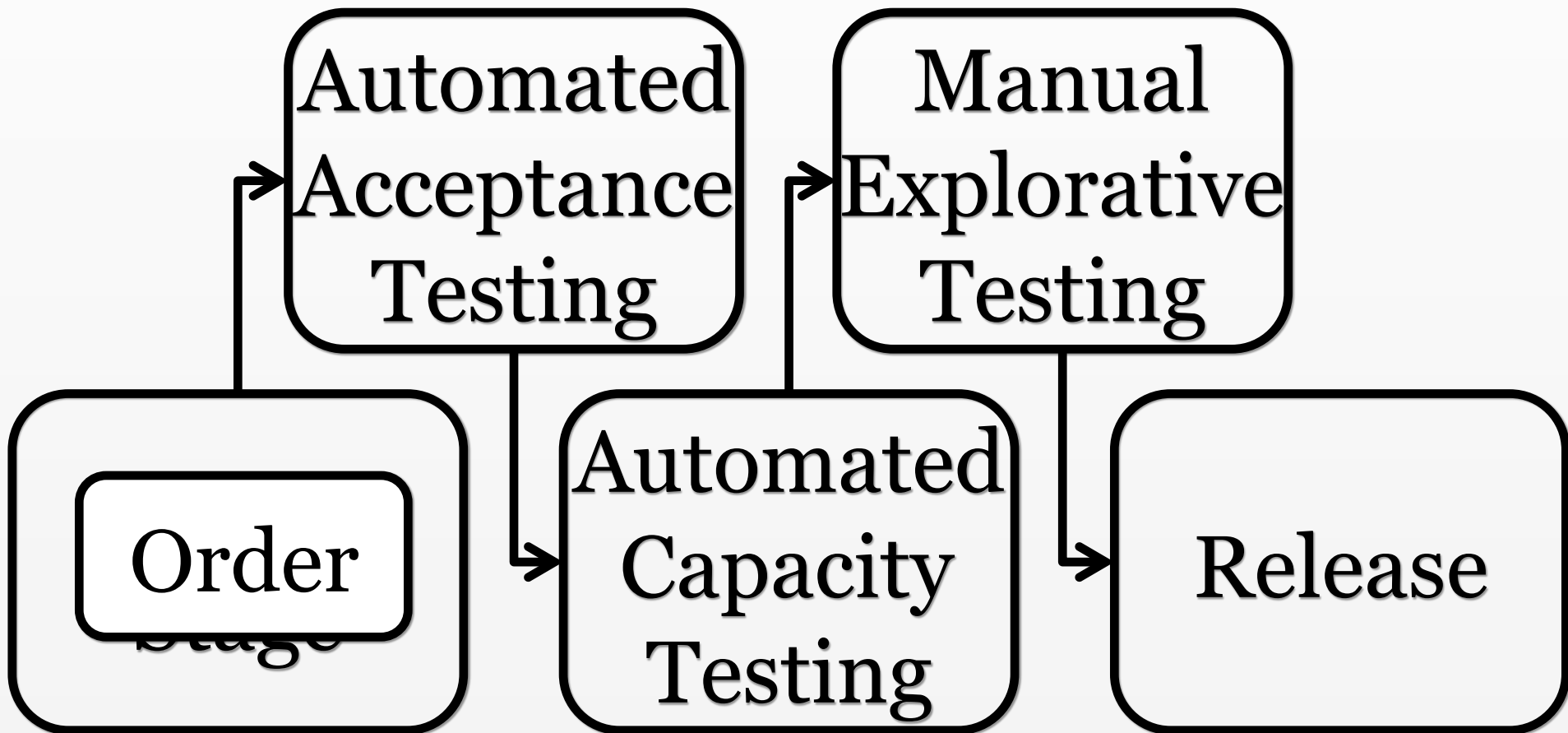
# Continuous Delivery: Everything Tested etc



# Build Pipeline

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

# With Micro Services



# Release

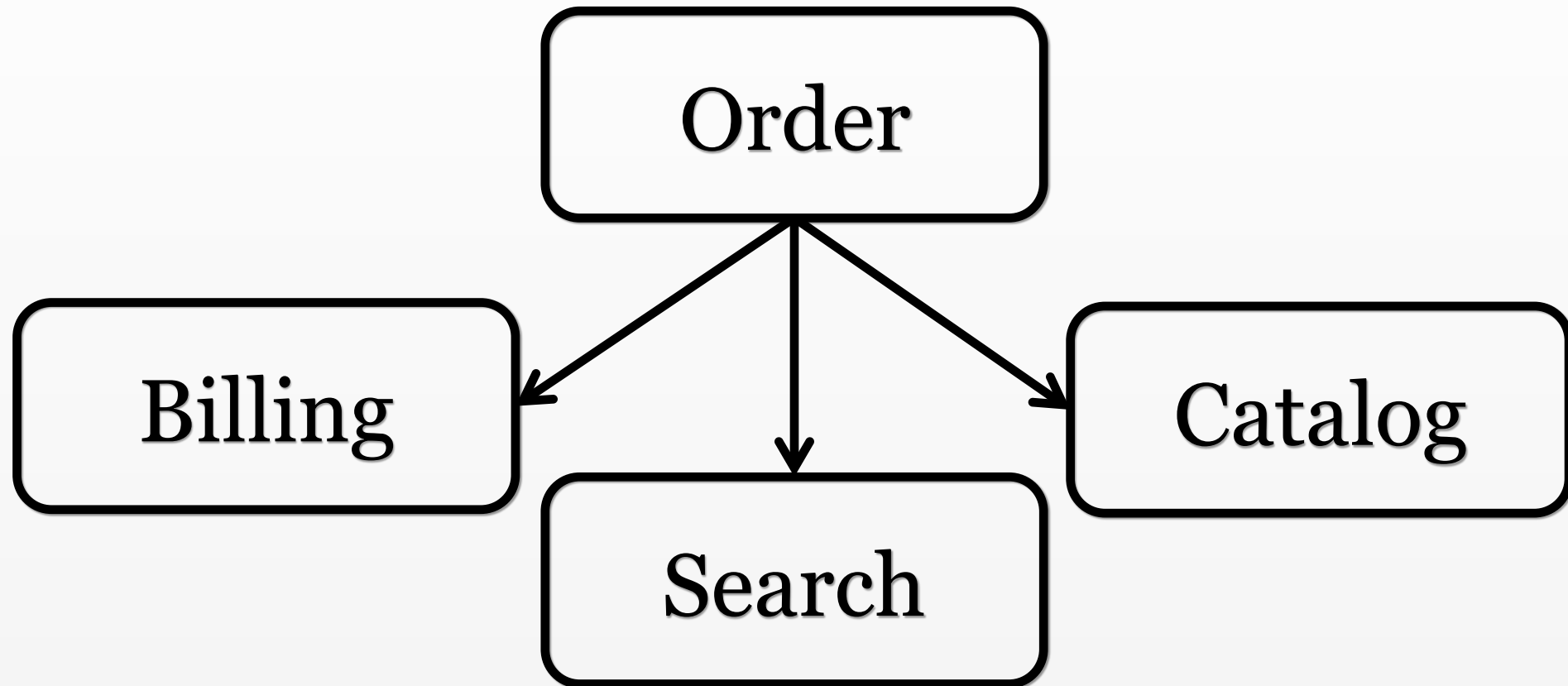
- Just a single component
- Faster feedback
- Easier deployment
- Rollback of single service also easier

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??

# Spring Boot Demo

# Why Spring Boot?

- Easier deployment
- i.e. just a JAR
  
- Built in operations support

# Technologies

- Dropwizard by Yammer



- Vert.x **VERT.X**

- Play Framework



Deploy &  
Operate?

# Component Model

- No restriction on language & environment
- 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

**Written data**

**Application**

**Java**

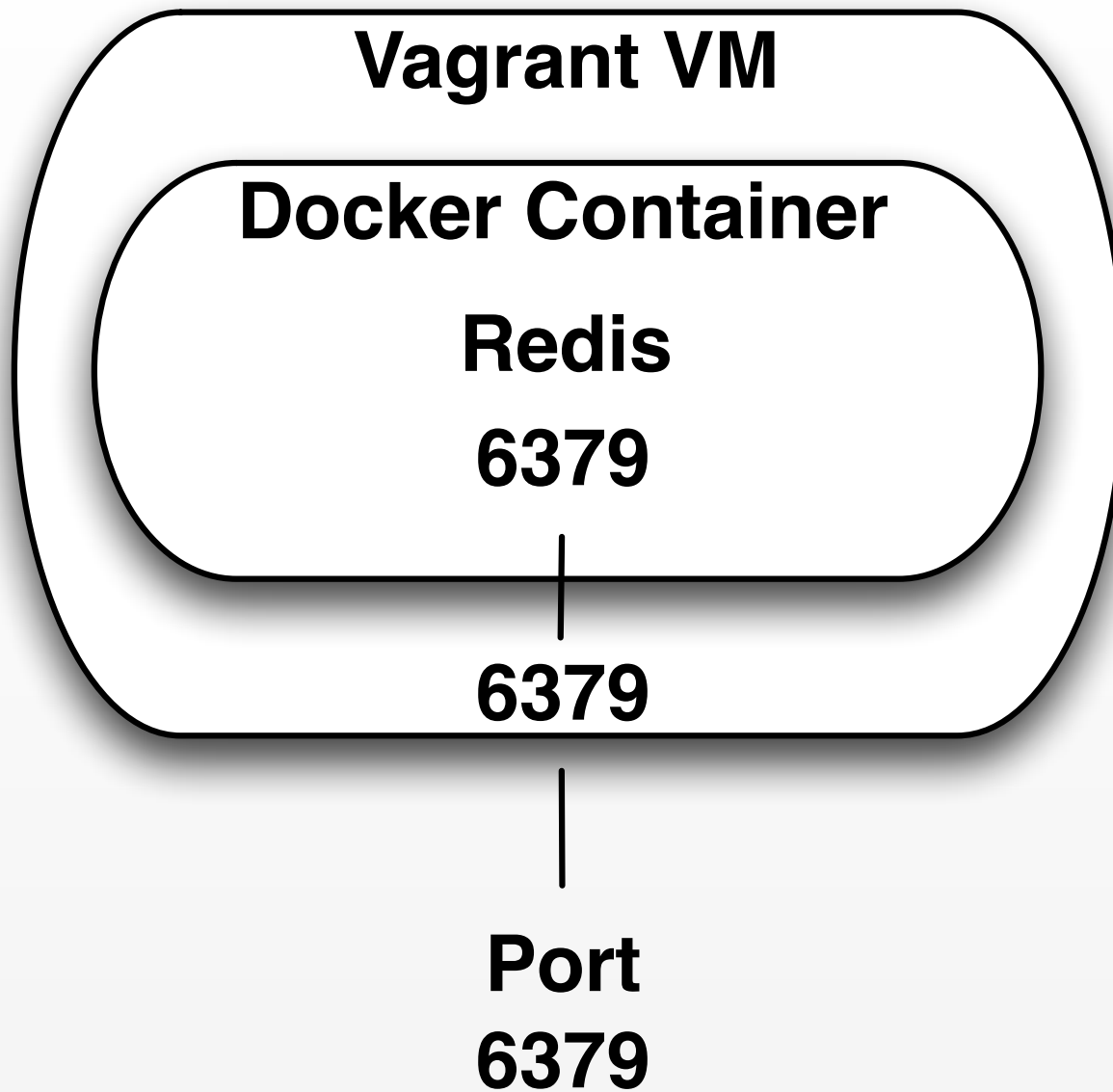
**Ubuntu**

# Docker

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

# Redis on Docker

- Small NoSQL database
- In memory
- Quite a few cool features



# Docker Demo

# Communication Between Docker Container

- Via ports
- Via data volumes

Docker is a  
Component  
System

# Example: Log File Analysis



# No Docker, no Micro Services

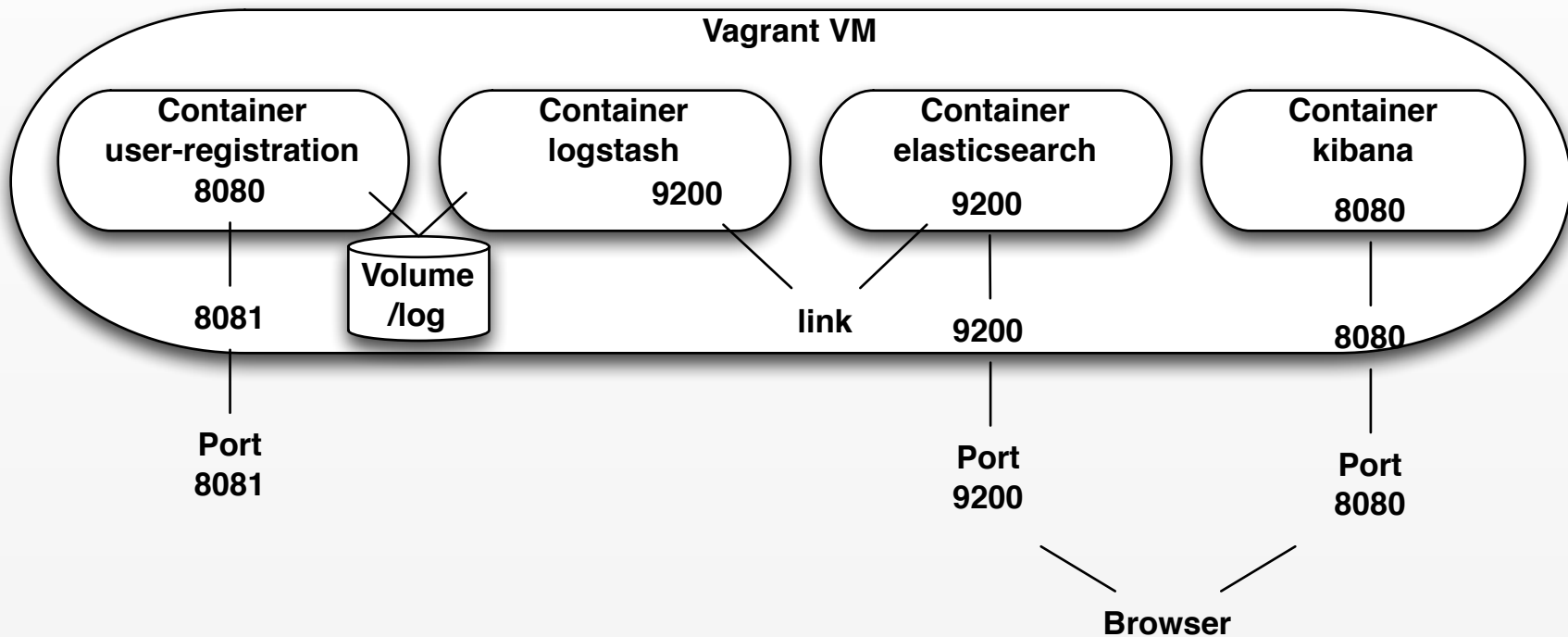
- Log file analysis
- One Application
- Store (database)
- Parse
- GUI
- One artifact on one server
- Logs provided e.g. by a port

# Docker, Small Services

- 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 22 lines)
  - 1 HTML file (11 lines)
  - 1 Config (Logstash) (29 lines)
- 
- Total: 87 lines, 9 files
  - Including one RegExp ☹️



# Demo ELK Stack

# Individual Scaling per Service

# 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 11,90 Schweiz sfr 16,50 Luxemburg 6,11 €

8.2014

JAVA Mag

# Java magazin

Java • Architekturen • Web • Agile

www.javamagazin.de

**BDD mit JBehave**

Verhaltensregeln für Anwendungen ▶ 101

**JSF und Bootstrap**

Fragezellenkur für Web-Apps ▶ 86

WJAX 14

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!!

@ewolff