Microservices Technology Stack

Eberhard Wolff
Fellow, innoQ
@ewolff

2. Auflage

Eberhard Wolff

Continuous Delivery

Der pragmatische Einstieg

dpunkt.verlag

Eberhard Wolff

Microservices

Grundlagen flexibler Softwarearchitekturen

dpunkt.verlag

http://microservices-buch.de/

Microservices

Flexible Software Architectures

Eberhard Wolff

http://microservices-book.com/

Eberhard Wolff

Microservices Primer

A Short Overview

FREE!!!!

innoQ

http://microservices-book.com/primer.html

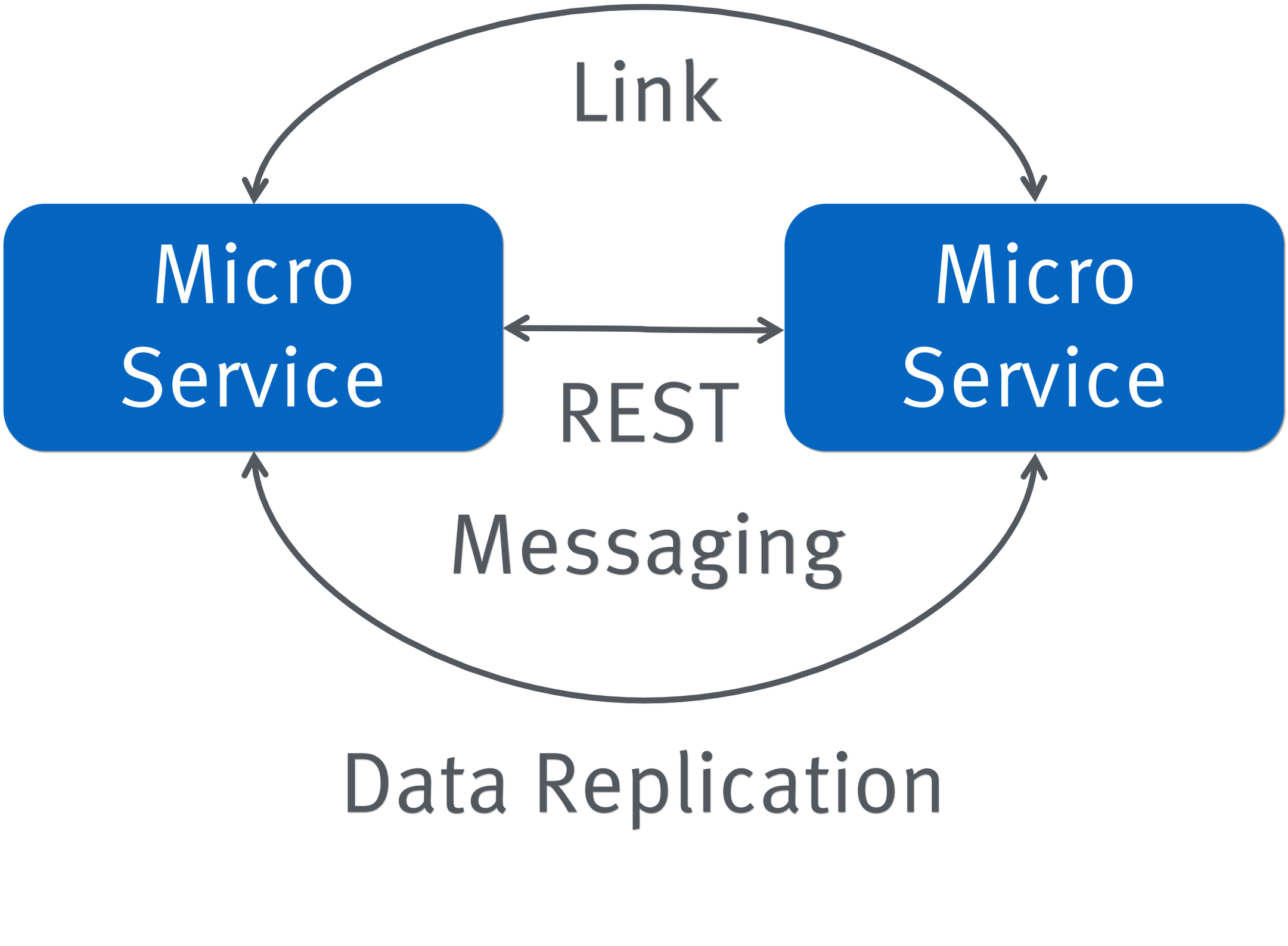
Microservices = Independent Deployment Units

Why can’t I just deploy Microservices on an Java EE server?

Microservices are just small REST services, right?

Integration

Components Collaborate



UI Integration

Browser

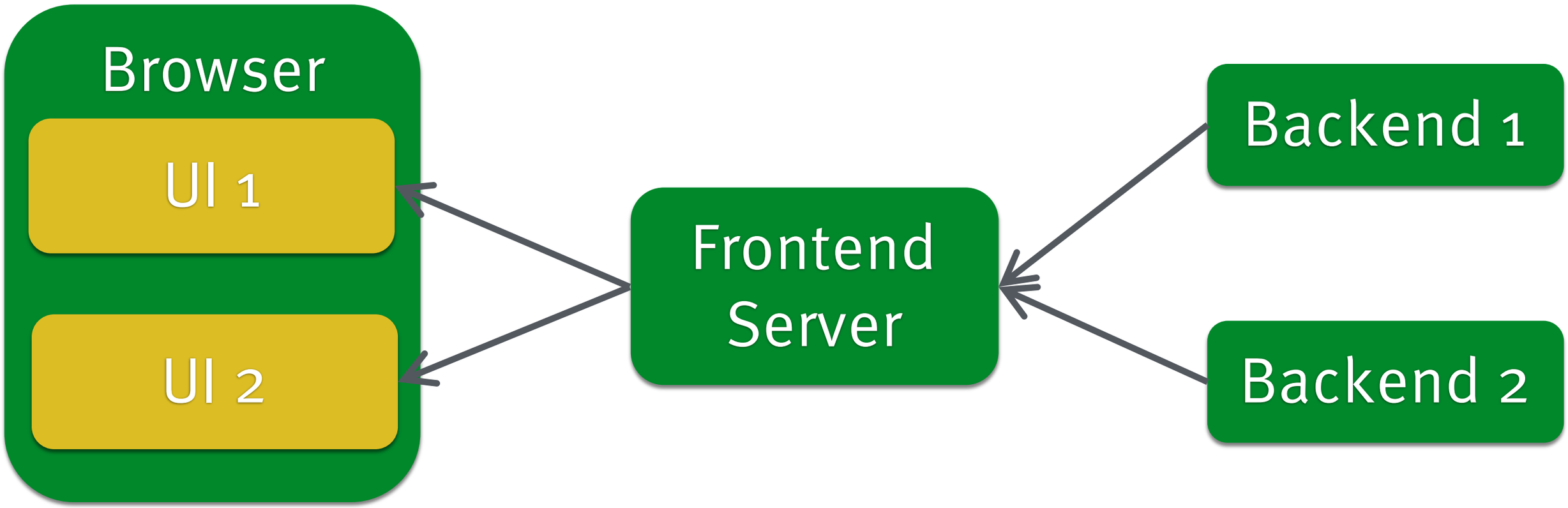
Order History

Recommendation

Order Microservice

Recommendation Microservice

Server-side integration



Server-side Integration: Technologies

- ESI (Edge Side Include)
- E.g. Varnish cache

- SSI (Server Side Include)
- E.g. Apache httpd, nginx

- Portal Server + Portlets

ESI (Edge Side Includes)

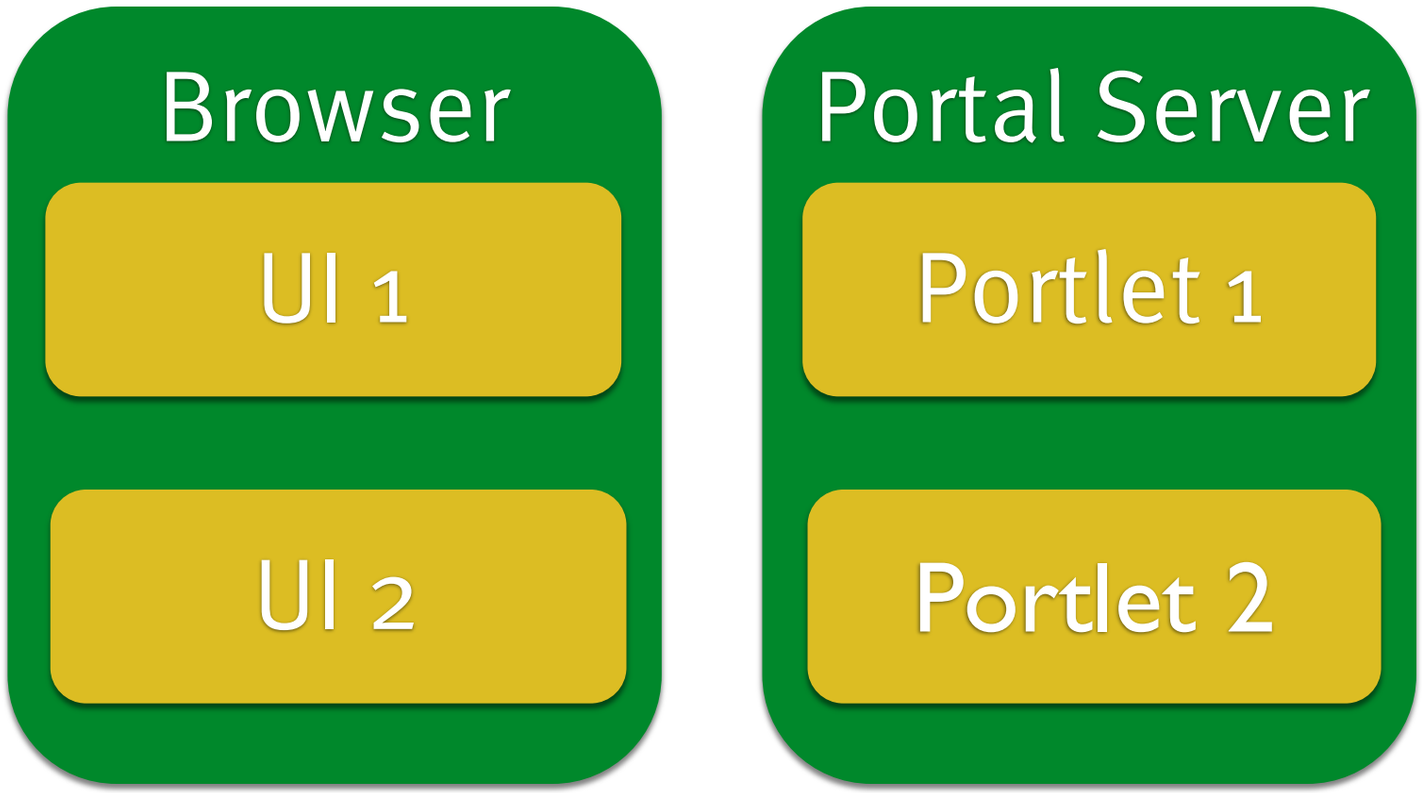
```
...
<header>
... Logged in as: Ada Lovelace  ...
</header>
...
 <div>
   ... a lot of static content and images ...
 </div>
 ...
<div>
  some dynamic content
</div>
```

ESI (Edge Side Includes)

```
...
<esi:include src="http://example.com/header" />
...
<div>
  ... a lot of static content and images ...
</div>
...
<esi:include src="http://example.com/dynamic" />
```

http://www.w3.org/TR/esi-lang

Portal Server



- Broad support and used in many projects
- Compose web application of Portlets
- Portlets might be deployed independently

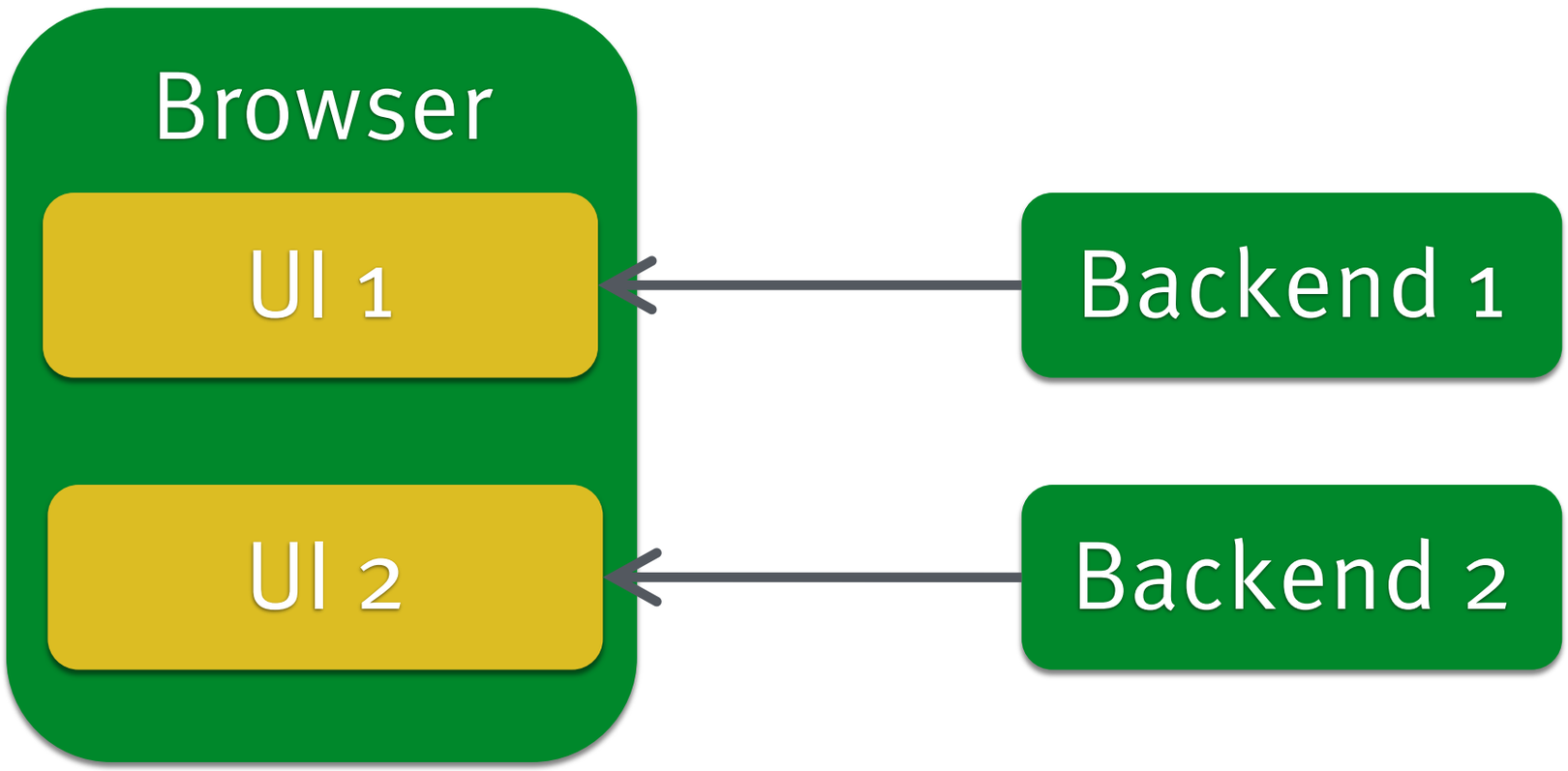
Portal Server: Challenges

- Do not really integrate diverse backends
- Web Services for Remote Portlets (WSRP) still require Portlets
- Portlets allow customization by user…
- …which make them very different from normal web applications

Portal Server

- No support for non-Java technologies
- Programming model very different from normal web applications
- Why would you use Portal Servers?

Client-side integration



- Client requirements (e.g. CORS, JS)
- Upcoming: HMTL Imports

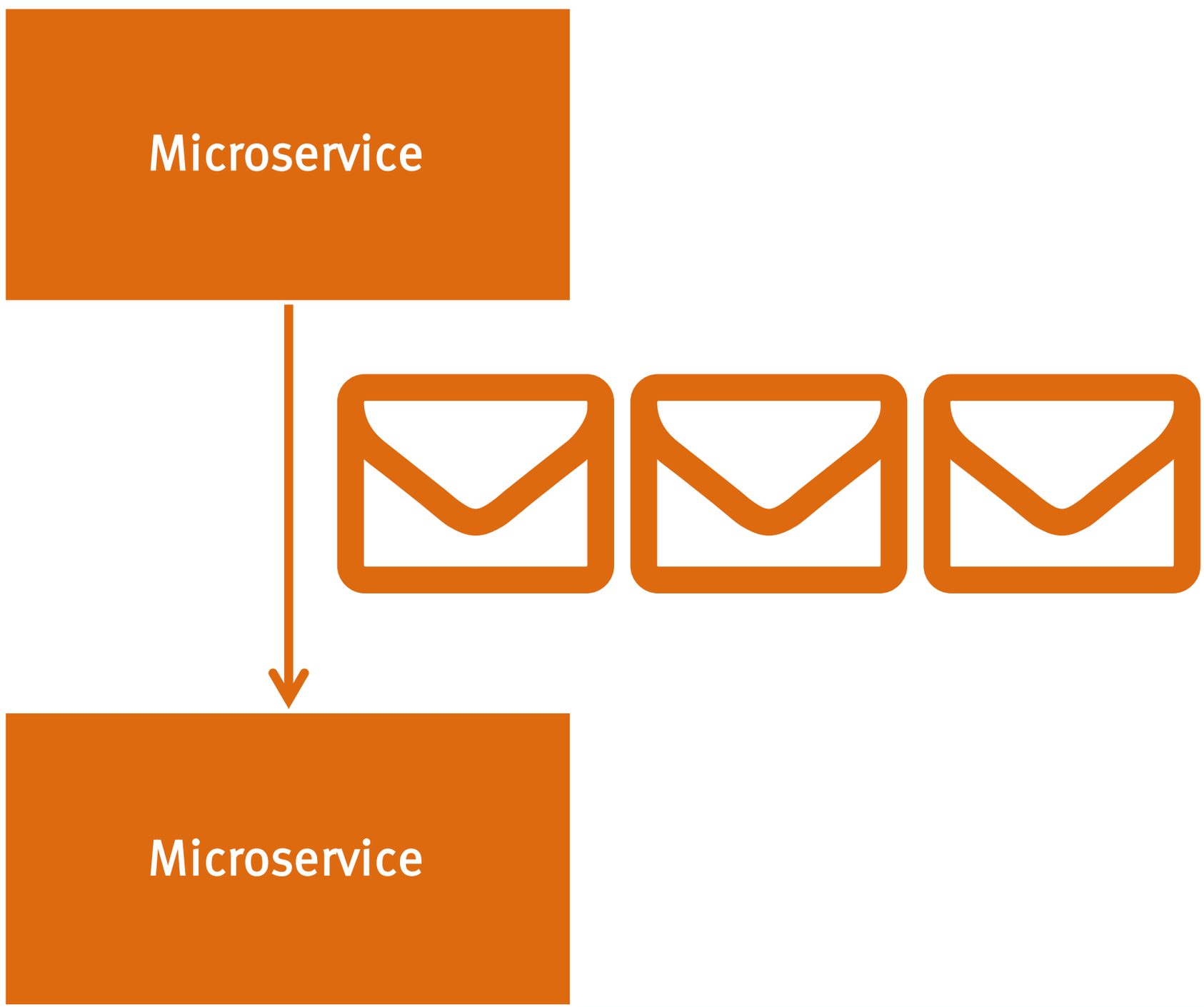
Client-side Integration: Code

- Replace ‹a href= " " class= "embeddable"› with content of document in a ‹div› (jQuery)

```
$("a.embeddable").each(function(i, link) {
  $("<div />").load(link.href, function(data, status, xhr) {
    $(link).replaceWith(this.children);
    });
});
```

Logic Integration

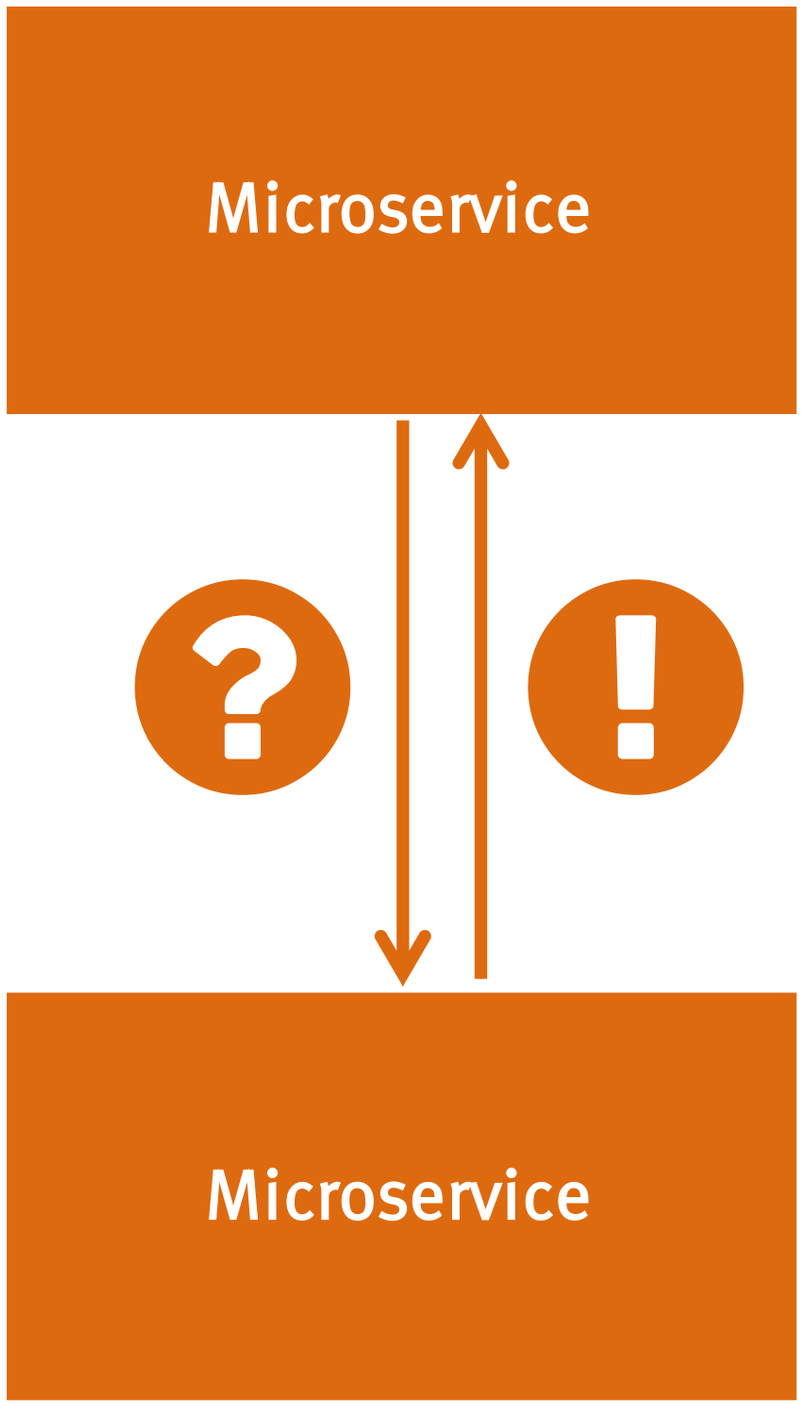
Messaging



Messaging

- Natural fit for asynchronous communications
  - E.g. event-driven
  - Decoupling: Sender and Receiver only know topic / queue
  - Re-submission for resilience
- Not natural for synchonous communication

REST



REST

- Natural fit for synchronous communications
- Evolvability with HATEOS
- Content negiotation

- Not natural for asynchonous communication

https://jaxenter.de/microservices-rest-vs-messaging-29875

Service Discovery

- Services need to find each other
- Enables load balancing
- For REST only
- E.g. DNS

Netflix Eureka

- Cache on the client
- Quick
- Proven
- Support for non-Java systems with sidecar only
- No DNS

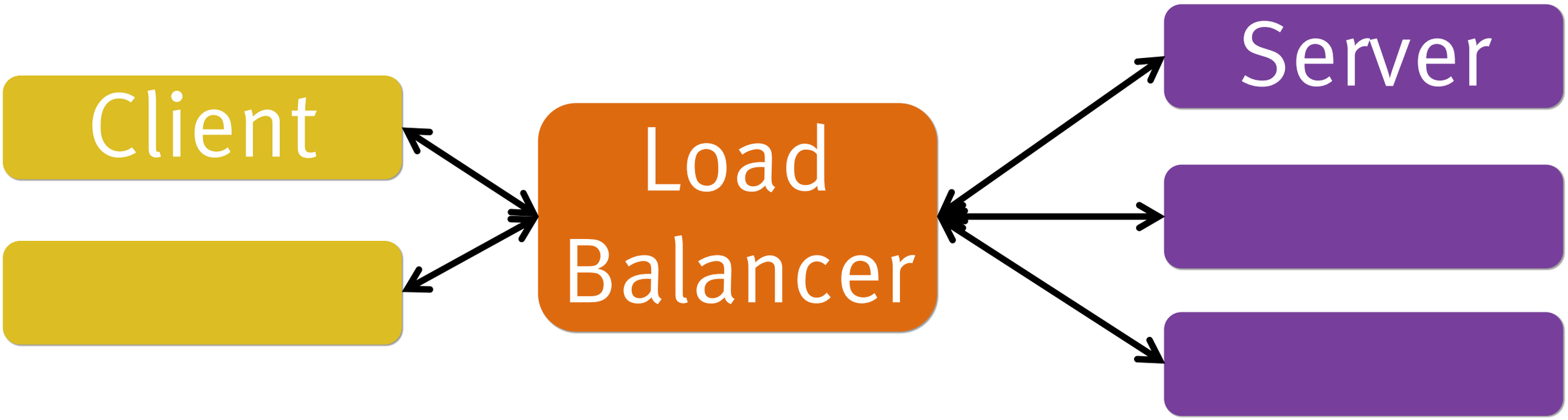
Hashicorp Consul

- DNS interface
- Supports storing configuration
- Consul templates to create configuration files
- Product

Resilience

- Service not available?
- Cicuit breaker
- Fail fast
- Time out
- Hystrix
- DO NOT IMPLEMENT RESILIENCE YOURSELF!

Proxy Load Balancing

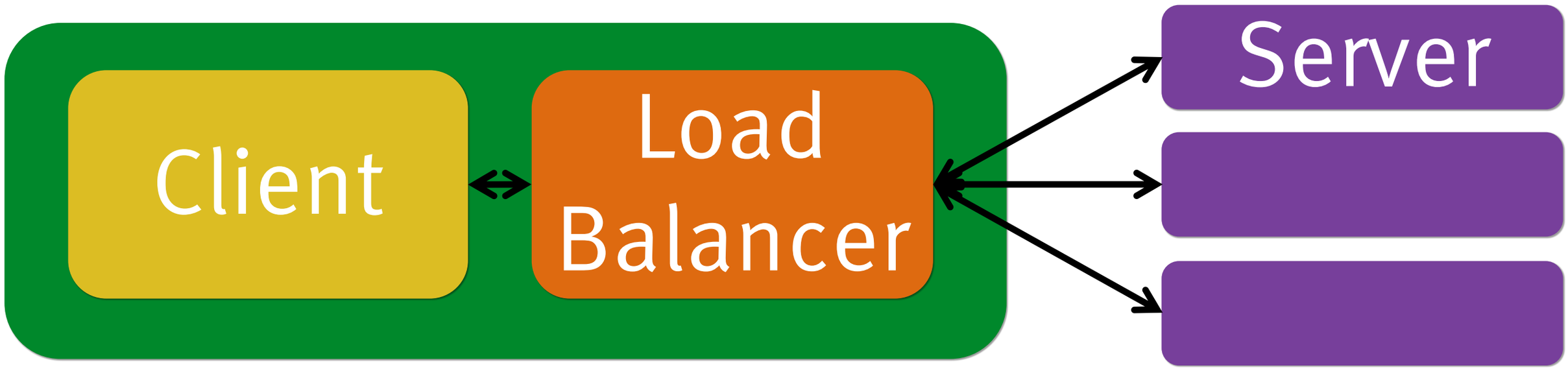


- Centralized Load Balancing
- Can become bottle neck
- Single point of failure

Ribbon: Client Side Load Balancing

- Decentralized Load Balancing
- No bottle neck
- Resilient
- Can consider response time
- Data might be inconsistent

Alternative: External Load Balancer



- Nginx, Apache httpd…
- Probably for each used Microservices
- Must each be configured

Load Balancing

Resilience

Service Discovery

REST

Messaging

Security

- Single Sign on for all Microservices
- Each Microservice decides about access policy

Client

User / password

Token with role

Security Server

Request with Token

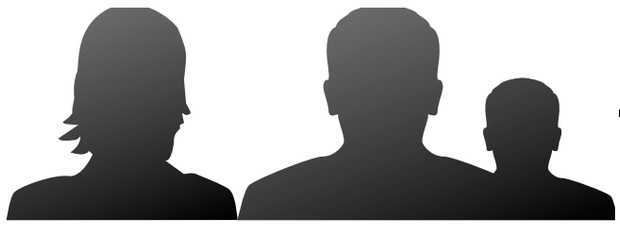
Access / no access

Micro-service

Security

- JSON Web Token (JWT)
- Define standard for tokens
- Includes additional data
- Can be stored in a cookie

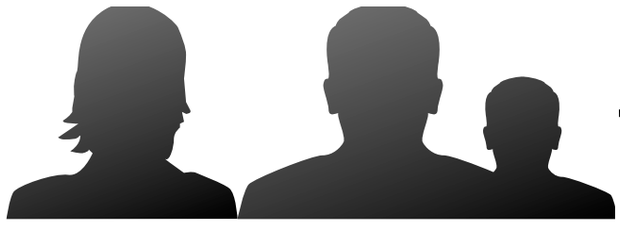
Testing



Search

Microservice

Continuous Delivery Pipeline



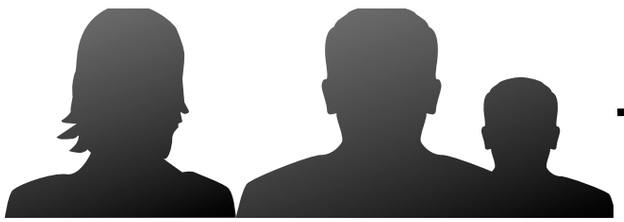
User

Microservice

Continuous Delivery Pipeline

Integration Tests

Production



Order Process

Microservice

Continuous Delivery Pipeline

...

Microservices = Independent Deployment Units

No independent testing = no independent deployment

Integration Test =

Bottleneck

Microservice

Request

Reply

Microservice

Microservice

Request

Reply

Stub

Stubs

- Does the calling Microservice work as expected?
- Use stubs to replace called Microservice

Microservice

Request

Reply

Microservice

Consumer-Driven Contract Test

Request

Reply

Microservice

Could just use some stub / test tool.

Consumer-driven Contract Test

- Does the called Microservice work as expected?
- Use consumer-driven contract test to replace calling Microservice
- Easy to test changes to interfaces

Microservice

Record

Request

Reply

Microservice

Consumer-driven contract test

Record

Replay request

Validate reply

Microservice

Microservice

Record

Request

Replay reply

Stub

- Pact
  https://github.com/realestate-com-au/pact

- Pacto
  https://github.com/thoughtworks/pacto

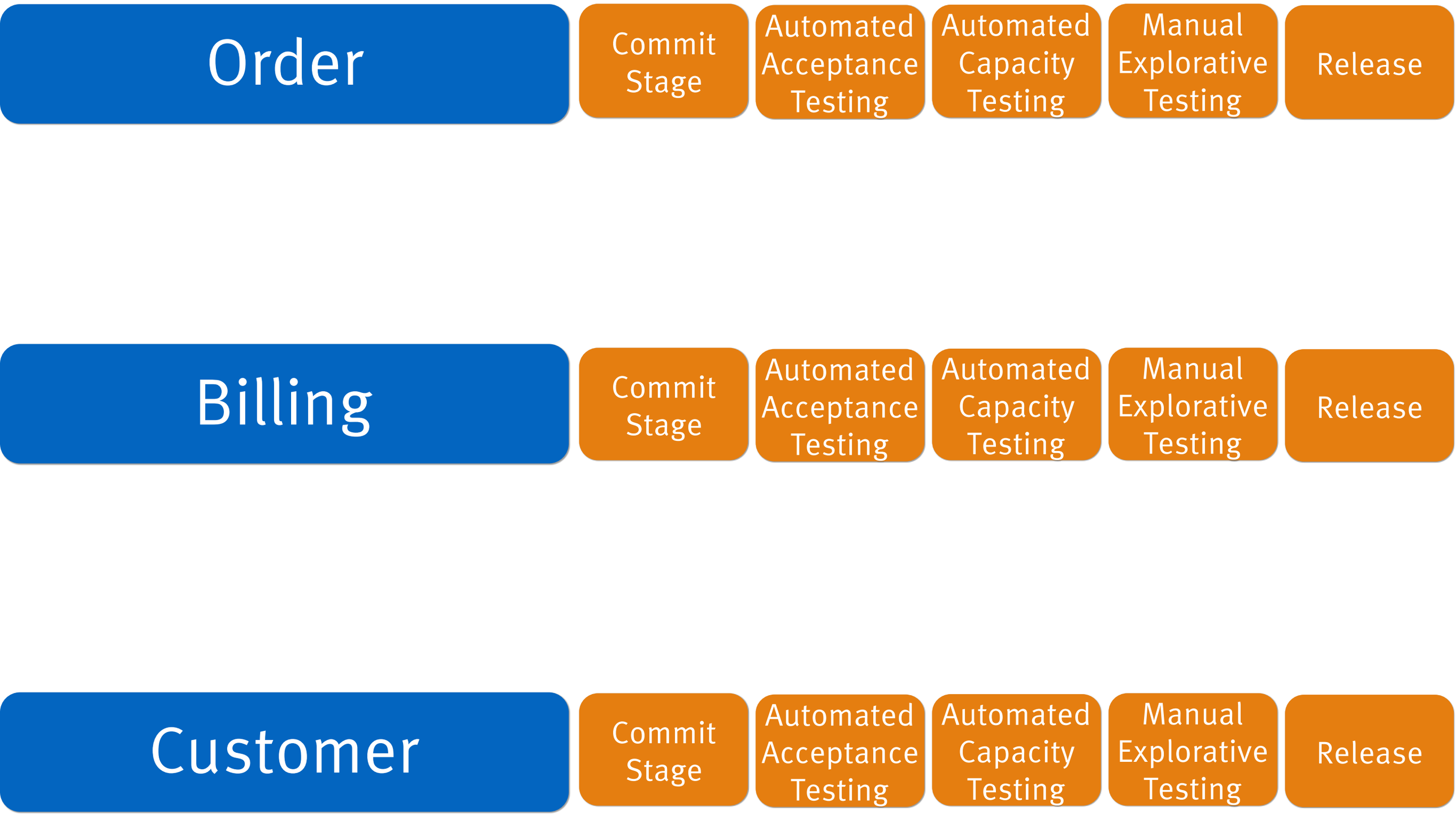
- Wiremock
  http://wiremock.org/

Operations

Hold my orange juice while I deploy a Microservice!

Deployment Not Easy

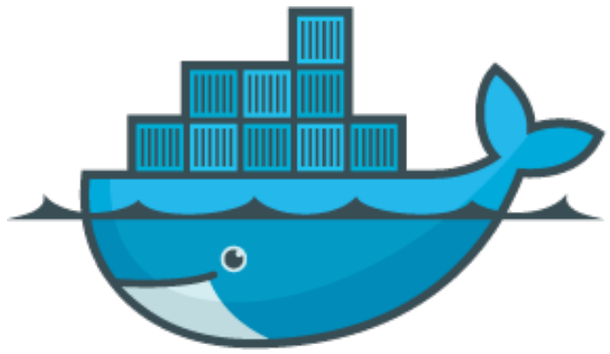
- Lots of microservices
- Lots of stages
- Deployment automation mandatory


Order
Commit Stage
Automated Acceptance Testing
Automated Capacity Testing
Manual Explorative Testing
Release
Billing
Commit Stage
Automated Acceptance Testing
Automated Capacity Testing
Manual Explorative Testing
Release
Customer
Commit Stage
Automated Acceptance Testing
Automated Capacity Testing
Manual Explorative Testing
Release

Deployment Automation

- Deployment automation mandatory
- Significant investment
- ...but worth it!!

Docker Images



- Easy to build
- Dockerfile = shell script
- Very efficient
- ...even for small changes

- Unfamiliar to Ops
- Replace VMware with Kubernetes?

CD Tools

- Puppet, Chef etc
- Idempotent installation
- E.g. easy update

- Complex
- Idempotent / updates not needed for Docker

Linux Packages

- Simple format
- Centralized installation easy

- Unfamiliar to devs

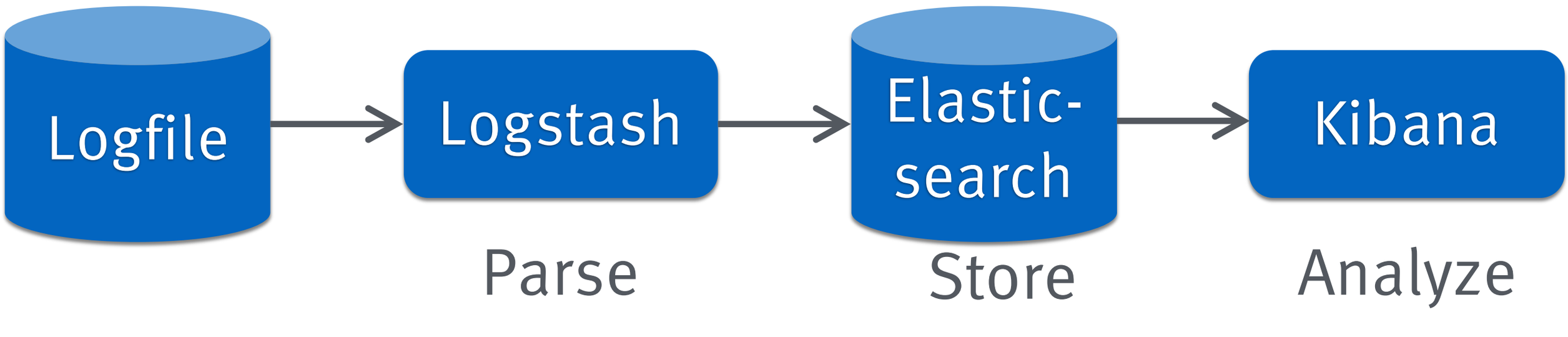
Hold my orange juice while I grep the logs for all Microservices!

Far too many files on far too many logs!

Centralized Logging

- Collect all logs on one server
- No need for local logs
- Read-only file system?

ELK Stack



- Logstash: JRuby
- Inputs, Parser, Outputs
- Elasticsearch: Storage, Java
- Kibana: Analysis, JavaScript

Discover - Kibana

Eberhard

localhost:5601/app/kibana#/discover?indexPattern=logstash-*&type=histogram&_a=(filters:!(),query:(query_s...

kibana

Discover | Visualize | Dashboard | Settings | Last 1 hour

*

logstash-*

Selected Fields

? _source

Available Fields

Popular

t level

Quick Count
(47 /47 records)

INFO 93.6%

ERROR 4.3%

WARN 2.1%

Visualize (1 warning ⚠)

t logmessage

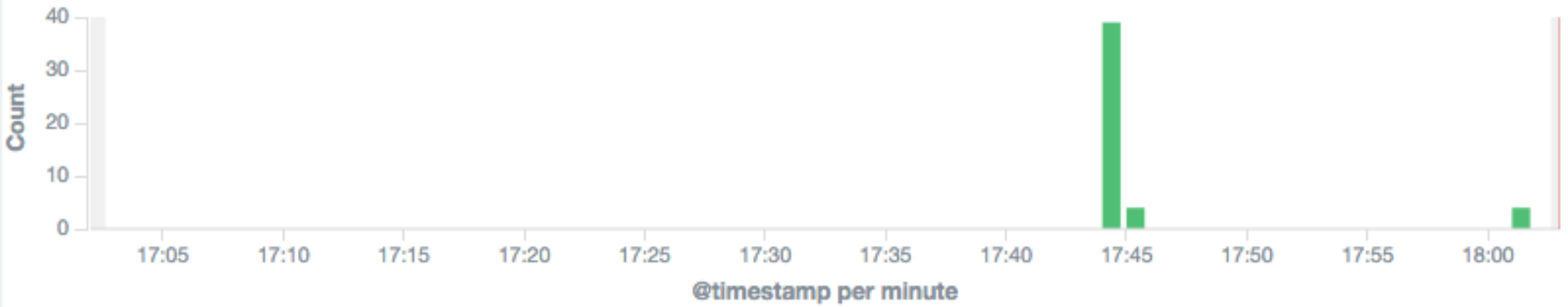
t "{[/],methods

t "{[/error],produces

t "{[/user],methods

47 hits

January 3rd 2016, 17:02:37.030 - January 3rd 2016, 18:02:37.030 — by minute



Time	_source
January 3rd 2016, 18:01:17.480	message: 2016-01-03 17:01:17.465 ERROR 8 --- [http-nio-8080-exec-1] o.a.c.c.C.[.[.[/].[dispatcherServlet] : Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed; nested exception is org.springframework.dao.DataIntegrityViolationException: PreparedStatementCallback; SQL [INSERT INTO T_USER(firstn
January 3rd 2016, 18:01:14.404	message: 2016-01-03 17:01:13.965 ERROR 8 --- [http-nio-8080-exec-10] o.a.c.c.C.[.[.[/].[dispatcherServlet] : Servlet.service() for servlet [di

Visualize - Kibana

Eberhard

localhost:5601/app/kibana#/visualize/create?type=histogram&indexPattern=logstash-*&_g=(refreshInterval:(di...

kibana

Discover | Visualize | Dashboard | Settings | Last 1 hour

*

logstash-*

Data | Options

metrics

Y-Axis Count

Select metrics type

Y-Axis

Cancel

buckets

X-Axis

Aggregation

Terms

Field

email.raw

Order By

metric: Count

Order: Descending

Size: 5

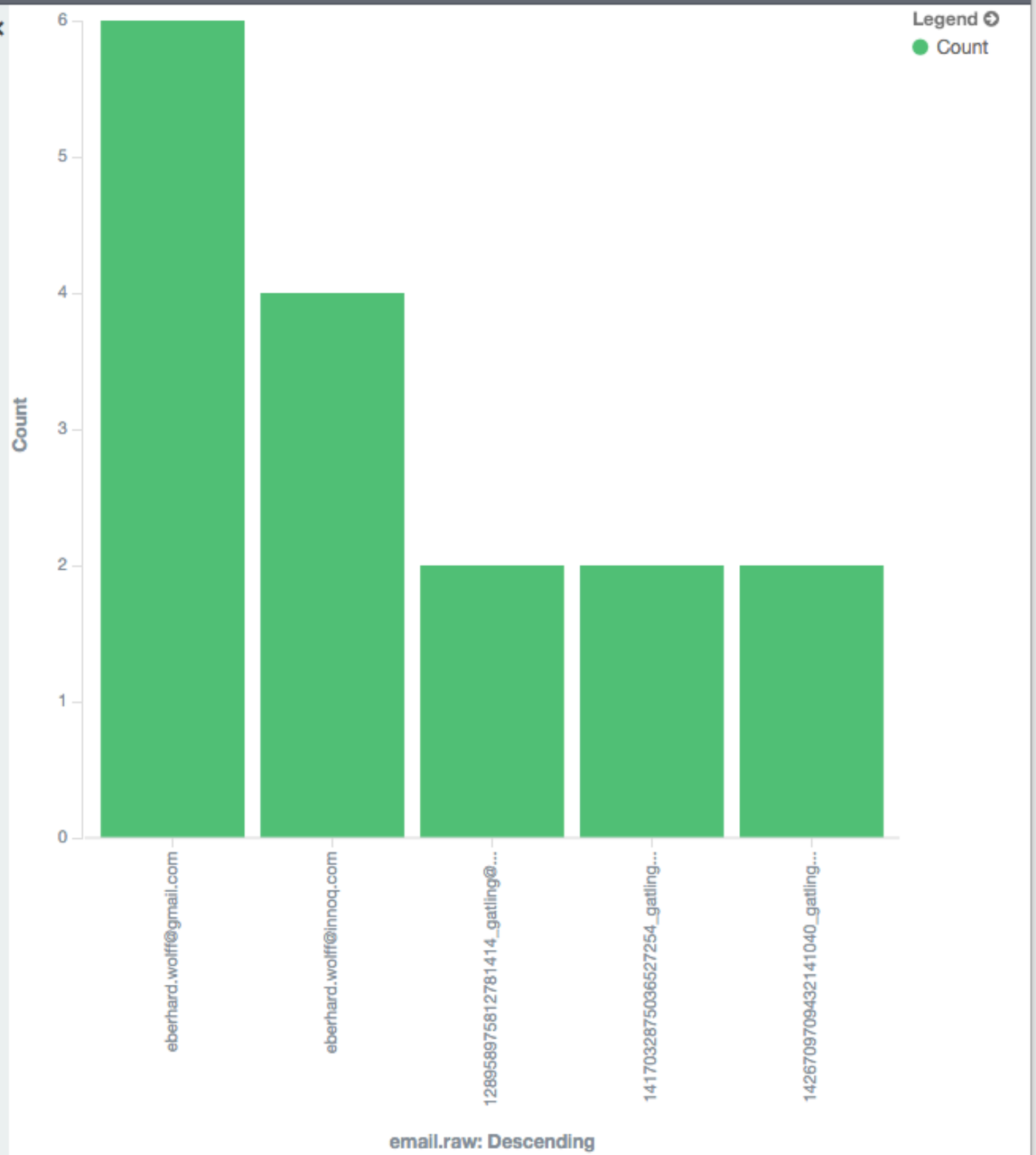
Advanced

Select buckets type

Split Bars

Split Chart

Cancel



Graylog

- Open Source
- Elasticsearch for storage
- MongoDB for Meta data
- GELF format for log messages

More Alternatives

- Splunk: commercial
- +Cloud
- Cloud : Loggly
- Sumo Logic
- Papertrail

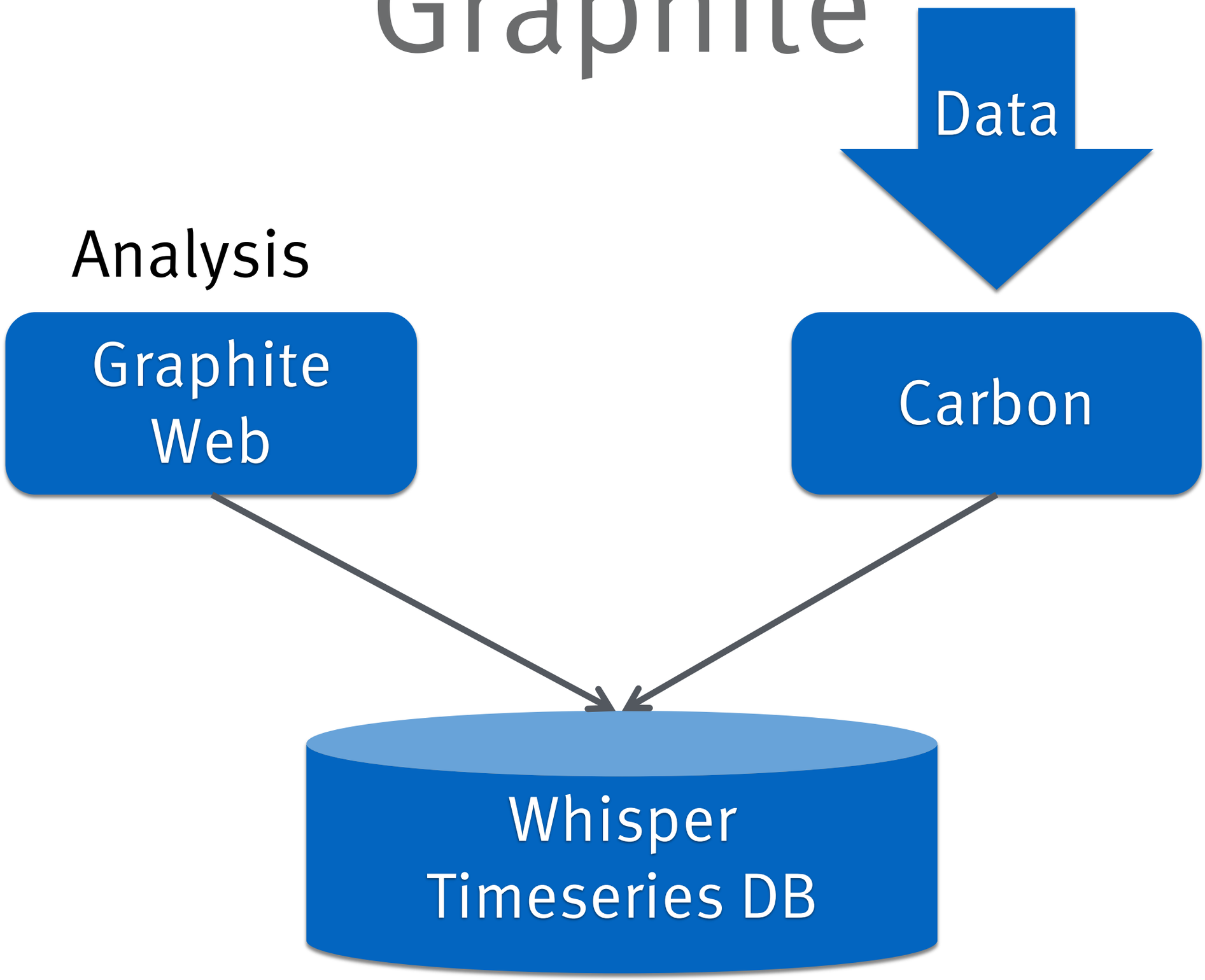
Hold my orange juice while I do top on my server!

Far too many servers!

Centralized Metrics

- Collect all metrics on one server
- Not just operating system, network
- ...but in particular application

Graphite

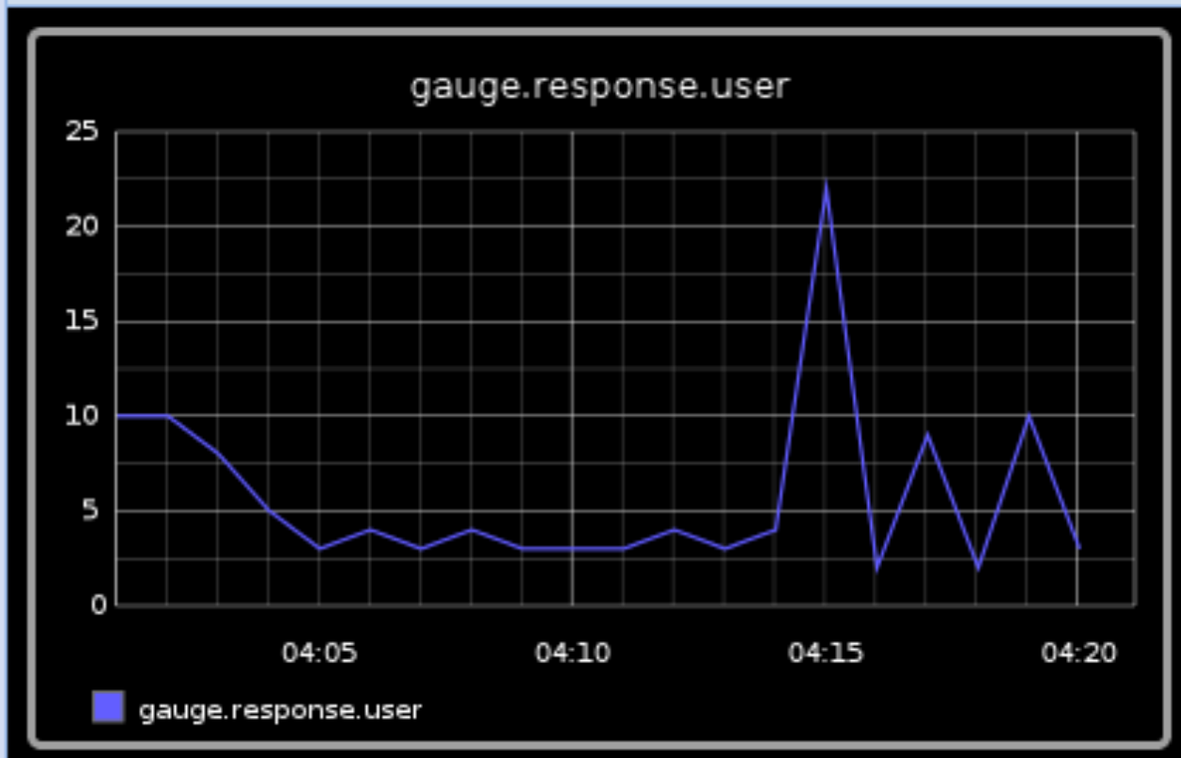


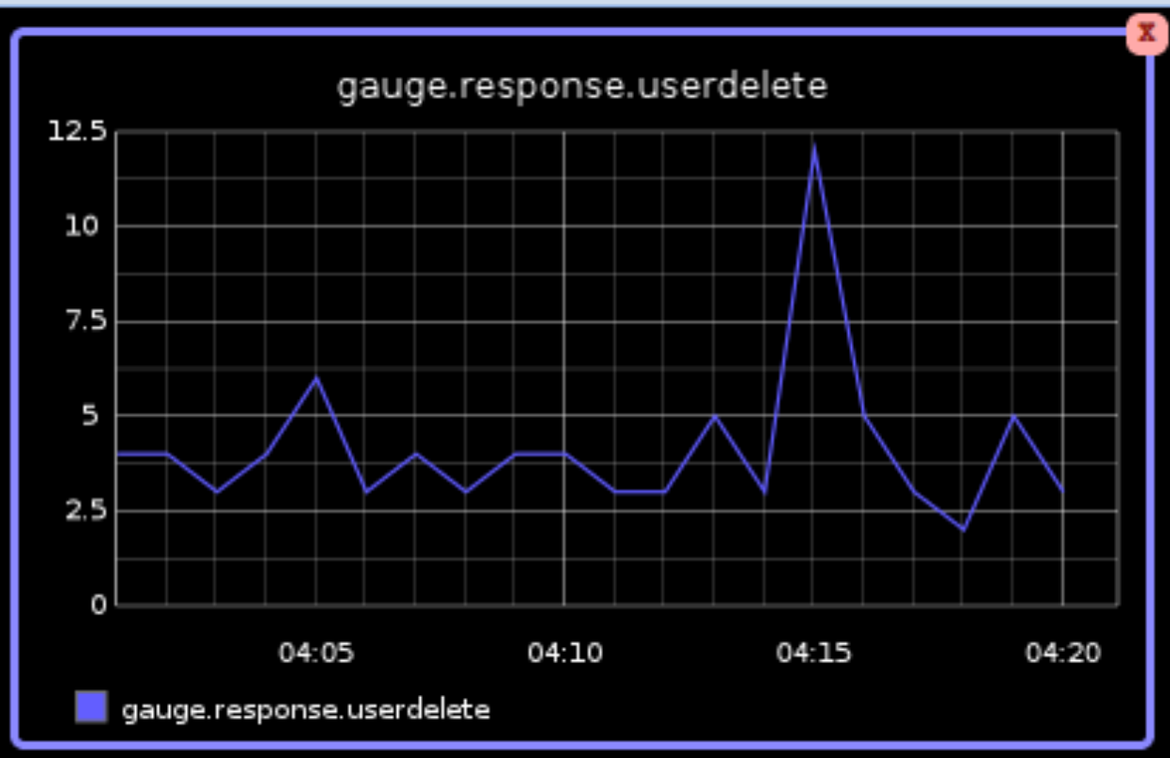
Library, writes / reads files

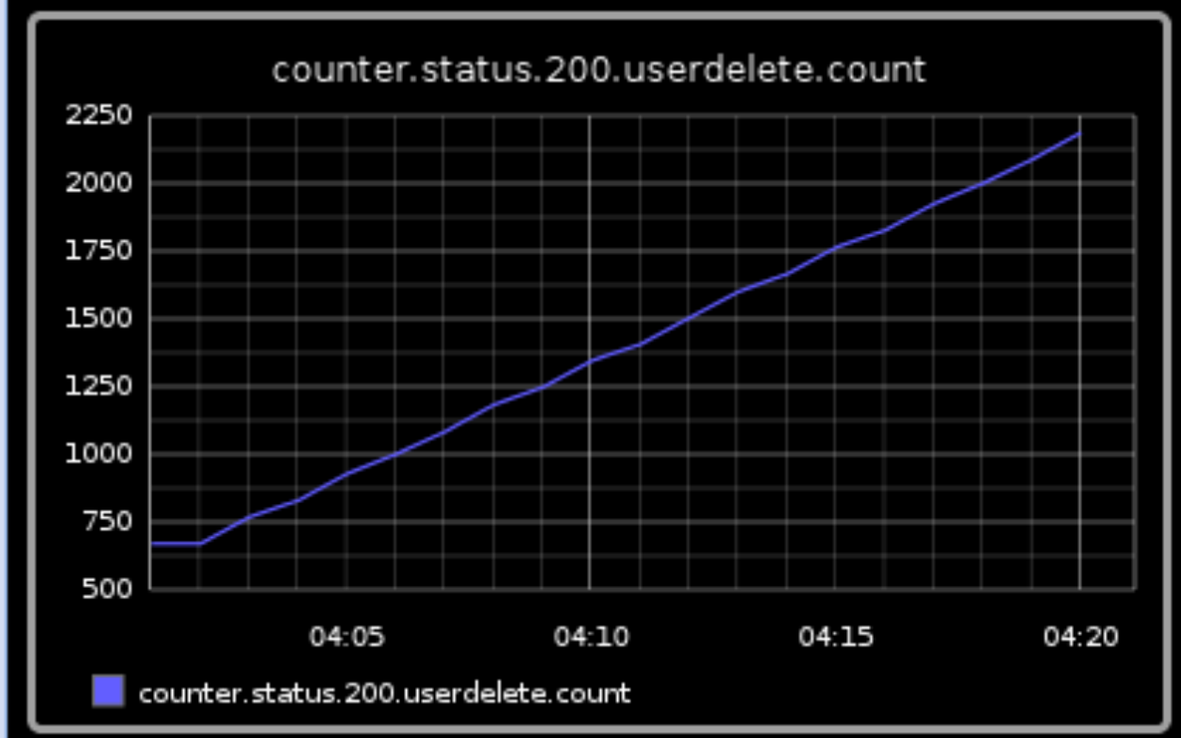
Graphite Dashboard

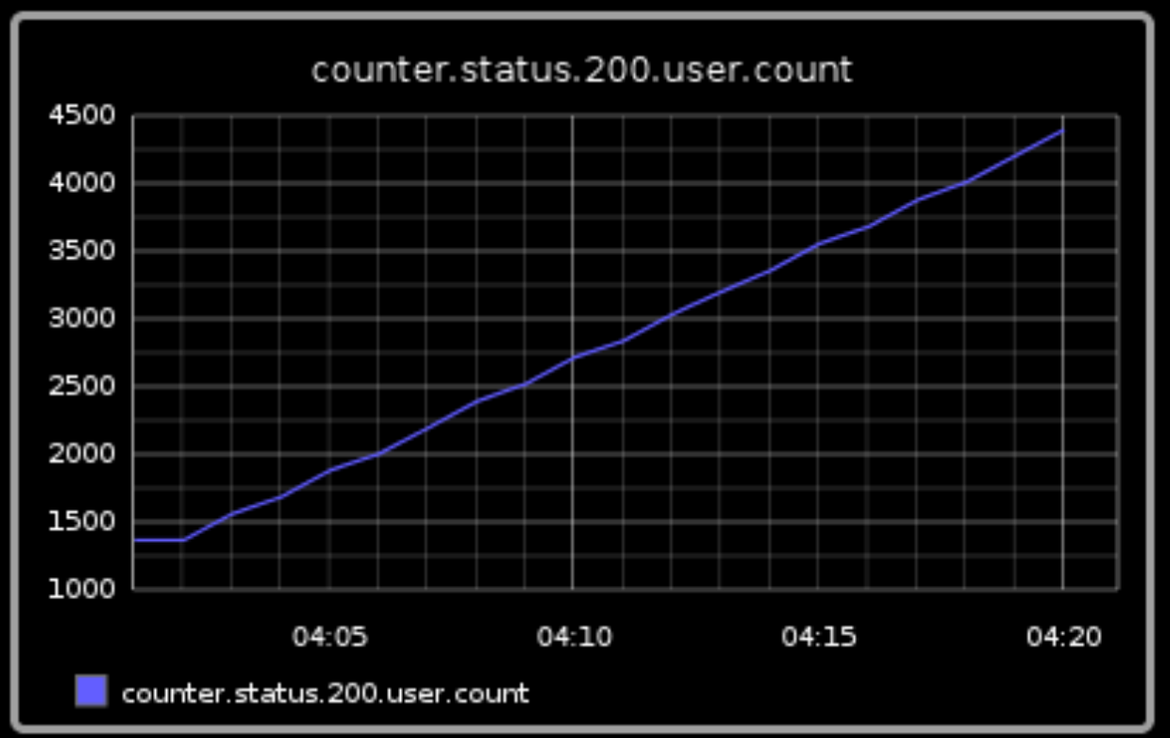
localhost:8082/dashboard&template=plain

Dashboard Graphs Share Relative Time Range Absolute Time Range Now showing the past 20 minutes Auto-Refresh every 60 seconds









Alternatives

- Nagios: Open Source monitoring
- Icinga: Originally Nagios fork
- Many commercial tools (HP, IBM, CA, BMC)
- Riemann: functional programming for event streams
- New Relic: Example for cloud

Alternatives

- TICK stack
- InfluxDB: Time-Series Database, also useful for Graphite
- Telegraf: Collects data
- Chronograf: Visulization
- Kapacitor: Alerts, anomaly detection

Alternatives

- Packetbeat: uses Elasticsearch for storage
- ...and Kibana for analysis

Hold my orange juice while I configure my server!

Configuration Files

- Easy to create
- Servers self-contained
- No centralized update possible
- Needed with canary releasing?

Configuration Server

- Easy to push out changes to all servers
- Can be the same server as for services discovery
- Additional technology

- Managing secrets
- Lease secrets
- Revoke secrets
- Key rolling
- Audit logs

Microservice Frameworks

- Lots and lots of them
- Each Microservice might use a different
- Not too much influence on the overall system

- Deployment Automation
- Centralized Logging & Monitoring

Operations

- Configuration
- Messaging or
- REST + Service Discovery + Reslience + Load Balancing

Communication

- Stubs & Consumer-driven Contract tests

Testing

So much stuff!

Technologies solve problems.

Use a technology only if it solves a problem.

- Deployment automation is great!
- Maybe Microservices finally enforce deployment automatiom?

Operations

- Operation & testing requires additional effort

Testing

- Communication: No DNS? No Messaging?

Communication

Microservices are a trade-off!

Microservices create additional technical complexity!