Scaling Architecture @ Zalando

Felix Müller - @fmueller_bln

Zalando

Who knows Zalando?

2008

2 monoliths...

2015

How big are we now?

Zalando in Numbers

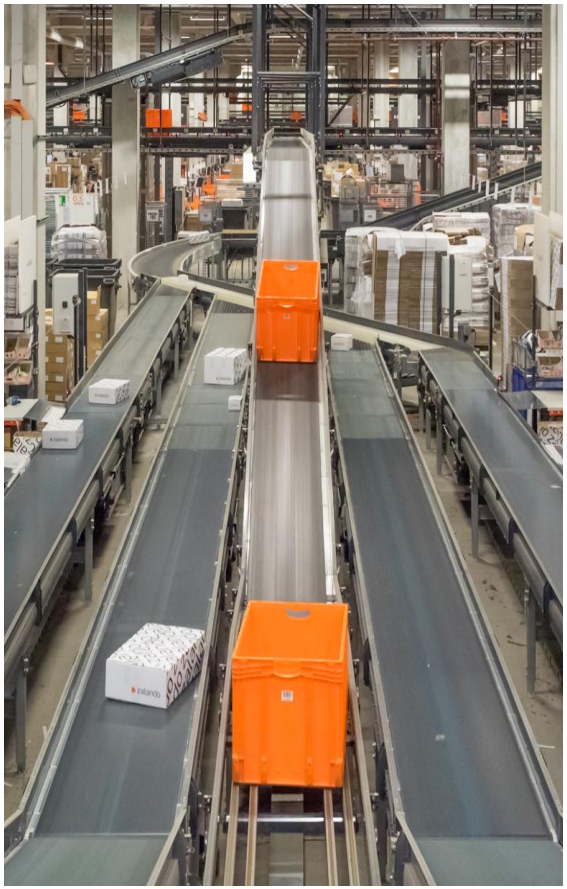
~ EUR **3 billion** revenue

> **160m** visits per month

> **11000** employees in Europe

~ **1600** tech employees

7 tech hubs: Berlin, Dublin, Helsinki...

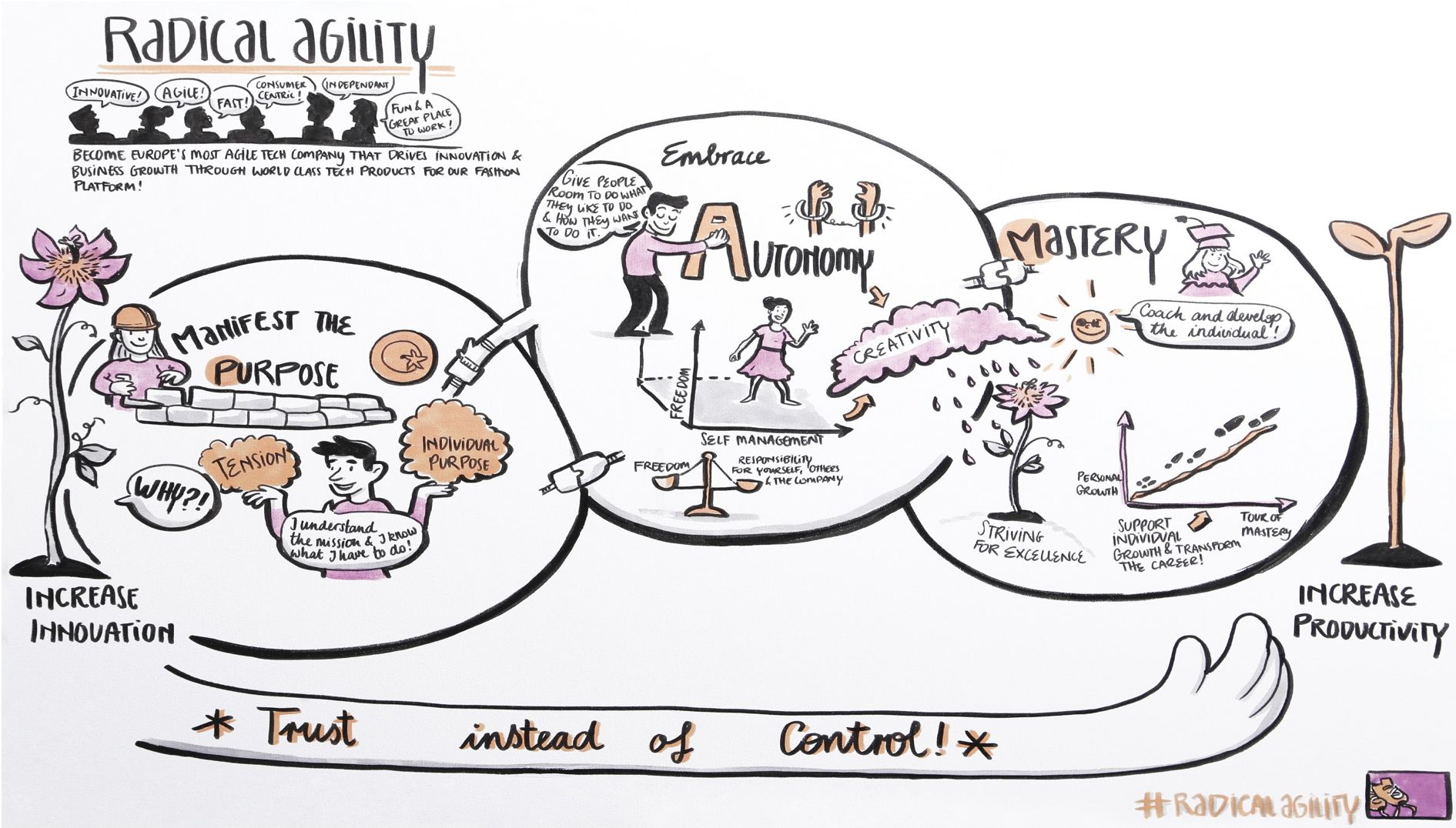


Still growing rapidly

2015 something more happened...

We want autonomous teams to deliver amazing products efficiently at scale.

Radical Agility


Radical agility
INNOVATIVE!
AGILE!
FAST!
CONSUMER CENTRIC!
INDEPENDANT
FUN & A GREAT PLACE TO WORK!
BECOME EUROPE'S MOST AGILE TECH COMPANY THAT DRIVES INNOVATION & BUSINESS GROWTH THROUGH WORLD CLASS TECH PRODUCTS FOR OUR FASHION PLATFORM!
Manifest the PURPOSE
WHY?!
TENSION
INDIVIDUAL PURPOSE
I understand the mission & I know what I have to do!
INCREASE INNOVATION
Embrace
GIVE PEOPLE ROOM TO DO WHAT THEY LIKE TO DO & HOW THEY WANT TO DO IT.
AUTONOMY
FREEDOM
SELF MANAGEMENT
FREEDOM
RESPONSIBILITY FOR YOURSELF, OTHERS & THE COMPANY
CREATIVITY
MASTERY
Coach and develop the individual!
STRIVING FOR EXCELLENCE
PERSONAL GROWTH
SUPPORT INDIVIDUAL GROWTH & TRANSFORM THE CAREER!
TOUR OF MASTERY
INCREASE PRODUCTIVITY
* Trust instead of Control! *
#RADICALAGILITY

Tech Constitution

#TECHCONSTITUTION

INDIVIDUAL

YOU HAVE THE RIGHT TO:

BE TRUSTED

MAKE INFORMED DECISIONS

ACHIEVE MASTERY IN SUPPORT OF PURPOSE

CHALLENGE WHEN PURPOSE IS UNCLEAR

BUILD A PERSONAL BRAND RELATED TO WORK

CHALLENGE ANYTHING AND ANYONE FOR THE

BETTERMENT

OF ZALANDO, ITS PEOPLE AND ITS VALUES

YOU HAVE THE RESPONSIBILITY TO:

DO THE RIGHT THING

WHEN YOUR BOSS IS WRONG

WORK TOWARDS PURPOSE

BE **EXCELLENT** TO EACH OTHER

STRIVE FOR MASTERY IN YOURSELF AND OTHERS

MAKE DECISIONS

INFORMED BY FACTS AND PEER REVIEW

BE ACCOUNTABLE FOR AND TO

LEARN FROM

THE CONSEQUENCES OF YOUR DECISIONS

BE OPEN TO COACHING AND PEER REVIEW

ZALANDO.
WE DRESS CODE.

#TECHCONSTITUTION

TEAM

YOUR TEAM HAS THE RIGHT TO:

ACHIEVE AUTONOMY, MASTERY AND PURPOSE FOR THE TEAM

DELIVER PRODUCTS THE TEAM IS PROUD OF

MAKE DECISIONS

ABOUT THE MEANS AND TOOLS OF DELIVERY AND OPERATION

TAKE REASONABLE RISKS TO

ACHIEVE GREATNESS

YOUR TEAM HAS THE RESPONSIBILITY TO:

OWN AND OPERATE

APPLICATIONS LONG TERM, EFFICIENTLY AND ACCORDING TO THEIR PURPOSE

DELIVER TO THE

HIGHEST STANDARD

APPROPRIATE TO THE PURPOSE

PLAY BY THE RULES, WHEN THE RULES EXIST

STRIVE FOR MASTERY AND PURPOSE WITHIN THE TEAM

CREATE A POSITIVE AND OPEN TEAM CULTURE

ENCOURAGE MOBILITY AND KNOWLEDGE SHARING

CHALLENGE AND IMPROVE WHAT DOESN'T WORK

FOSTER A CULTURE OF PEER REVIEW, INSIDE AND OUTSIDE THE TEAM

ZALANDO.
WE DRESS CODE.

#TECHCONSTITUTION

ZALANDO

WE HAVE THE RIGHT TO:

EXPECT EACH INDIVIDUAL TO ACT IN THE

BEST INTEREST OF ZALANDO

EXPECT EVERYONE TO ADHERE TO THIS CONSTITUTION

EXPECT EACH TEAM TO DELIVER GREAT PRODUCTS

CHANGE ZALANDO'S PURPOSE OVER TIME

EXPERIMENT WITH HOW TO ORGANIZE

IMPOSE RULES

WHICH TEAMS & INDIVIDUALS MUST FOLLOW

WE HAVE THE RESPONSIBILITY TO:

IMPOSE AS FEW RULES AS POSSIBLE

BE AS TRANSPARENT AS POSSIBLE, ABOUT ALL THINGS

BE THE CHAMPIONS OF AUTONOMY BY TRUSTING INDIVIDUALS AND TEAMS

PROVIDE THE **RESOURCES** FOR MASTERY

COMMUNICATE PURPOSE OPENLY AND CONSISTENTLY, AND TO DRIVE SHARED FOCUS

PROVIDE THE BEST

TOOLS AND ENVIRONMENT

FOR TEAMS TO ACHIEVE GREATNESS

ACKNOWLEDGE THE INDIVIDUAL'S PURPOSE

DEFEND THESE RIGHTS ABSOLUTELY

ZALANDO.
WE DRESS CODE.

Rules of Play

https://github.com/zalando/zalando-rules-of-play

Which architecture style is fostered by Radical Agility?

Architecture Principles

Architecture Principles

https://github.com/zalando/zalando-rules-of-play#architecture

We prefer loosely coupled services.

Architecture Principles in detail

Microservice with RESTful APIs	Loosely coupled systems
Favor message-driven	Asynchronous communication
Everything fails	Resilient systems
Operational Excellence	Automate everything

How do we operate our services?

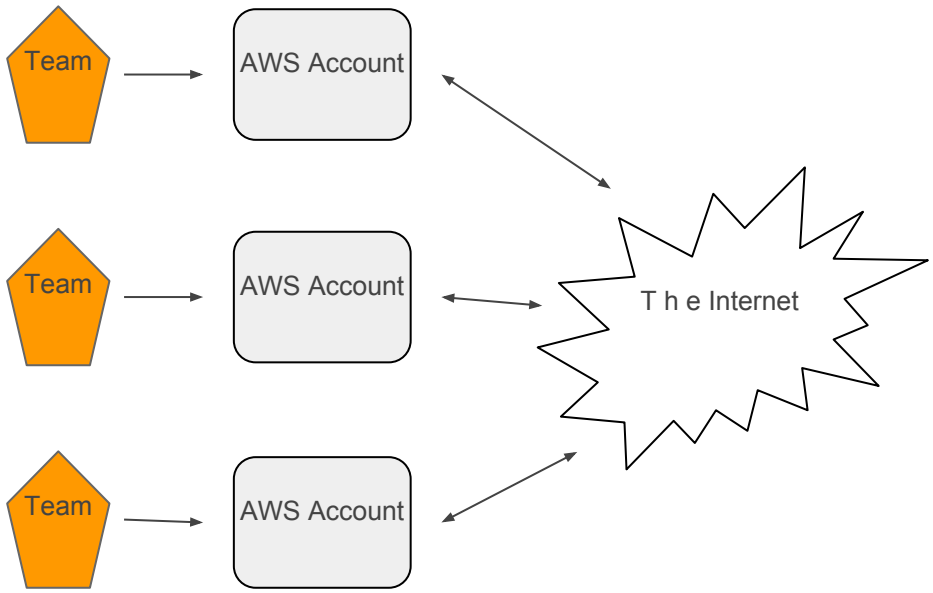
AWS Setup

AWS Setup

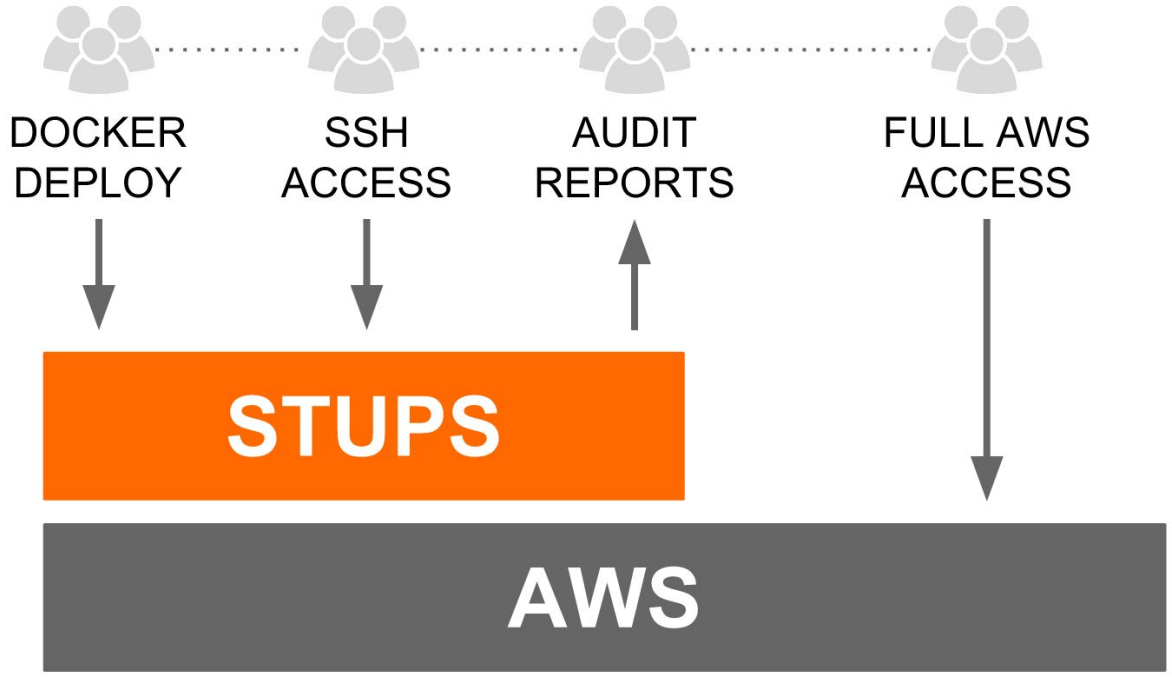
Each team has its own account

You build it, you run it

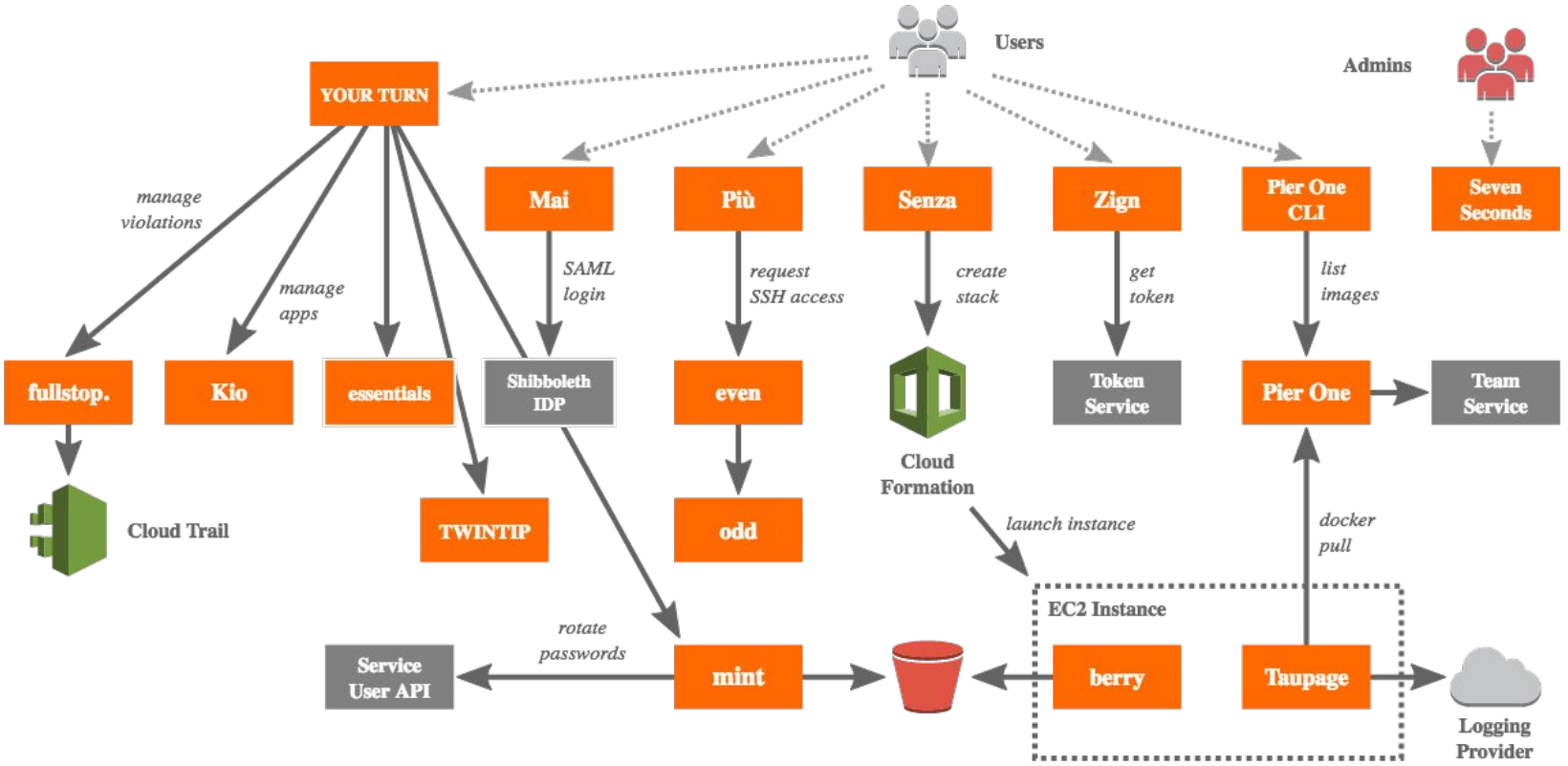
Deployment via our STUPS tooling



STUPS.io


DOCKER DEPLOY
SSH ACCESS
AUDIT REPORTS
FULL AWS ACCESS
STUPS
AWS

STUPS.io


Users
Admins
YOUR TURN
Mai
Più
Senza
Zign
Pier One CLI
Seven Seconds
manage violations
manage apps
SAML login
request SSH access
create stack
get token
list images
fullstop.
Kio
essentials
Shibboleth IDP
even
Token Service
Pier One
Team Service
Cloud Trail
TWINTIP
odd
Cloud Formation
launch instance
docker pull
EC2 Instance
rotate passwords
Service User API
mint
berry
Taupage
Logging Provider

How do we test our services?

Testing

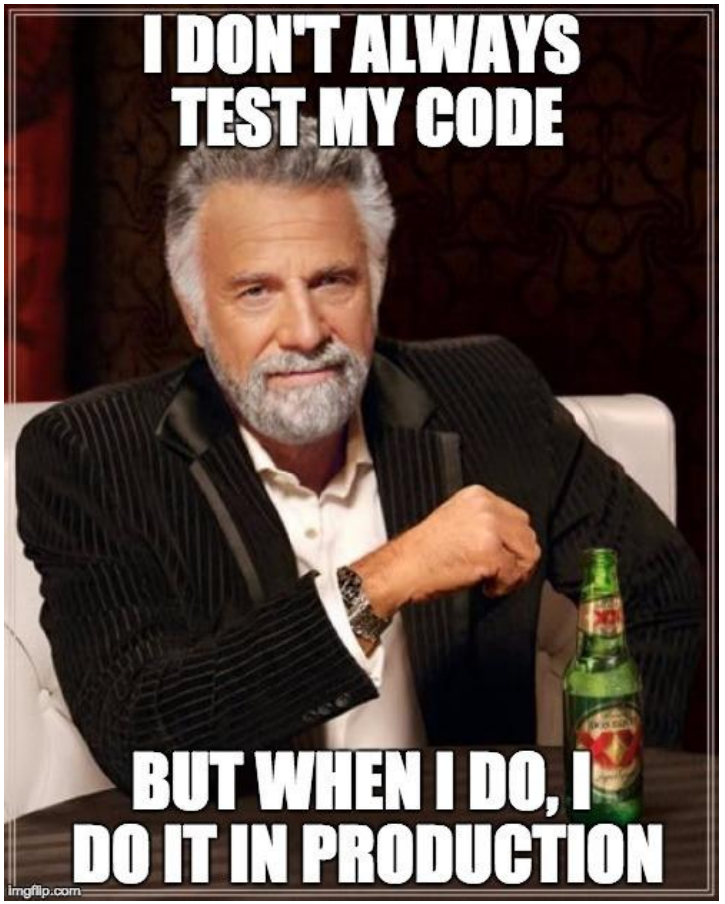
—

Testing

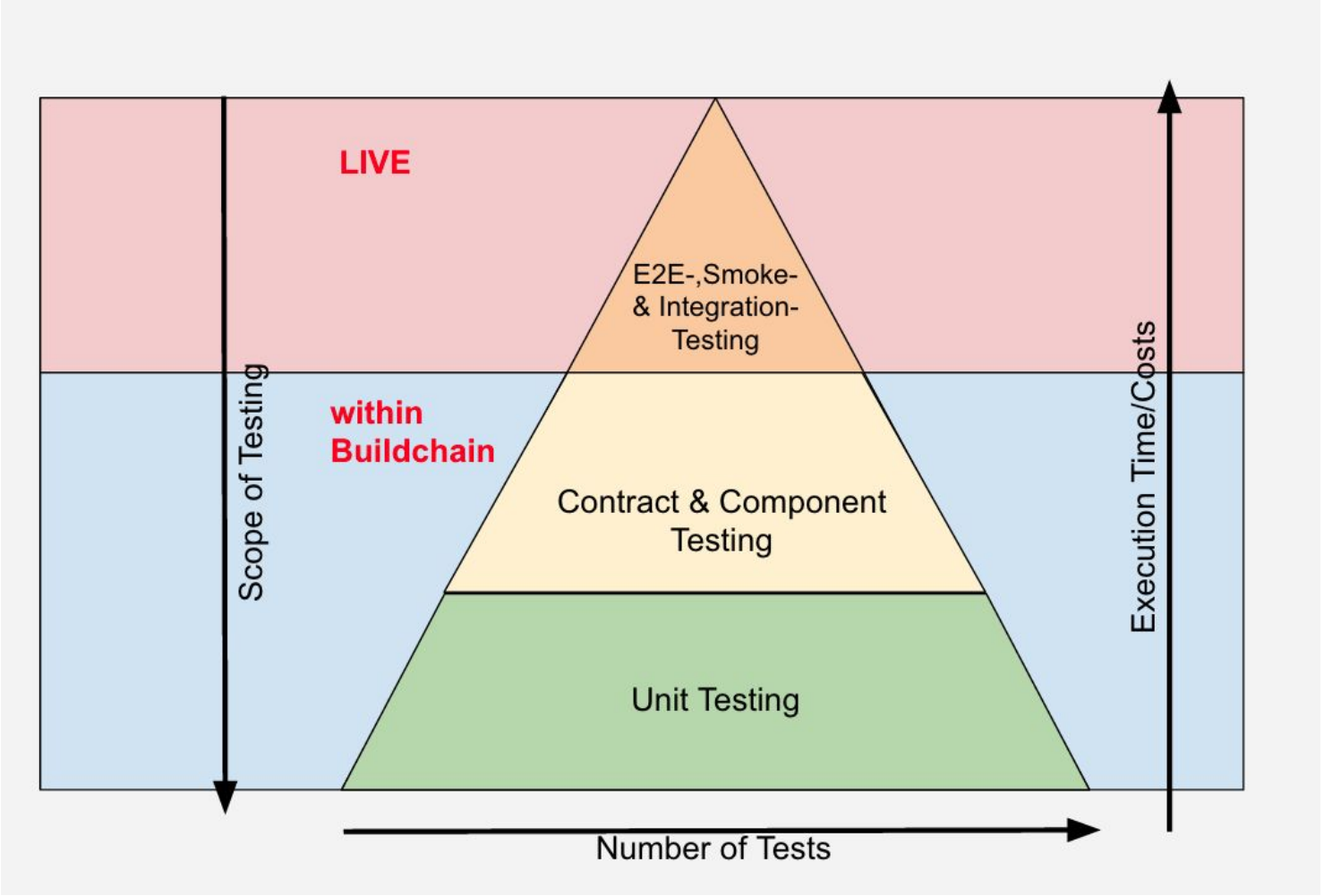
Of course, automated testing - a lot

Trade-off: Time to prod <> test coverage

Testing flows becomes harder with microservices



Our Testing Pyramid - where to run what

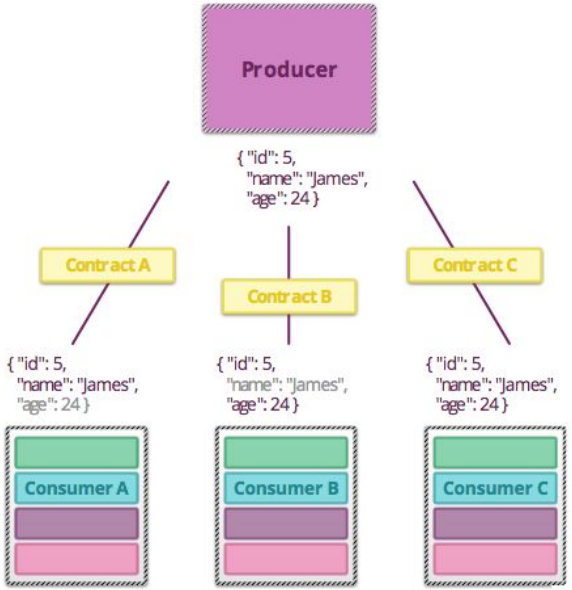


Consumer-driven Contracts

Better collaboration between teams

Emphasize importance of APIs

Lowers inter-team alignment efforts



How do teams collaborate?

Collaboration between teams

At best, not necessary to deliver

Through APIs

Through Code - shared code

Open Source Culture

Open Source Culture

https://github.com/zalando/zalando-howto-open-source

Open Source Evangelist, Lauri Apple

Open Source Guild

Management supports Open Source

Example for Open Source @ Zalando

https://github.com/integrations/zappr

Zappr
Approvals for pull requests

Started in our Innovation Labs

Now an official Github plugin

Open Source First

Shared code
between teams has
to be open source.

Inner Source

And how does architecture work takes place?

T h e Architect

~~The Architect~~

Service architecture is designed by delivery teams.

Global architecture is owned by the Architecture team.

Architecture Team

Architecture Team

Overarching team to support delivery teams

4 in Berlin, 1 in Dublin

Focus on new platform and APIs

Tech decisions with company-wide impact

We support teams in building most appropriate tech solutions.

Radical Agility vs Arc Team?

We respect self-autonomy of teams.

We support teams as technical consultants.

We can provide outside perspective + broader technical overview.

We let teams make their own architecture decisions.

There was team collaboration through APIs...

API Guild

API Guild

Group of Architect and Engineers

Develops API Guidelines

Drives API Reviews

API First

API Reviews

Feedback Culture

Foster review culture

Get feedback as soon as possible

API Guild is public review partner

API Guidelines

API Guidelines

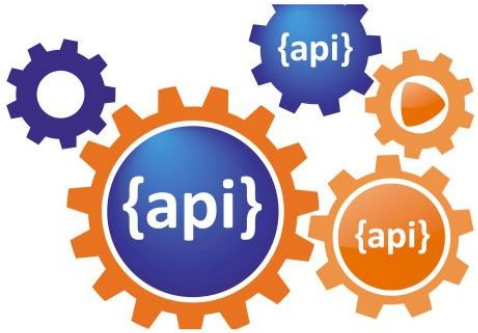
https://github.com/zalando/restful-api-guidelines

REST Maturity Level 2

Backward compatibility over versioning

Common Naming Rules and Data Objects

API Guidelines



Introduction

Zalando's software architecture centers around decoupled microservices that provide functionality via RESTful APIs with a JSON payload. Small engineering teams own, deploy and operate these microservices in their AWS (team) accounts. Our APIs most purely express what our systems do, and are therefore highly valuable business assets. Designing high-quality, long-lasting APIs has become even more critical for us since we started developing our new open platform strategy, which transforms Zalando from an online shop into an expansive fashion platform. Our strategy emphasizes developing lots of public APIs for our external business partners to use via third-party applications.

With this in mind, we've adopted "API First" as one of our key engineering principles. Microservices development begins with API definition outside the code and ideally involves ample peer-review feedback to achieve high-quality APIs. API First encompasses a set of quality-related standards and fosters a peer review culture including a lightweight review procedure. We encourage our teams to follow them to ensure that our APIs:

Our future system will consist of thousands of microservices.

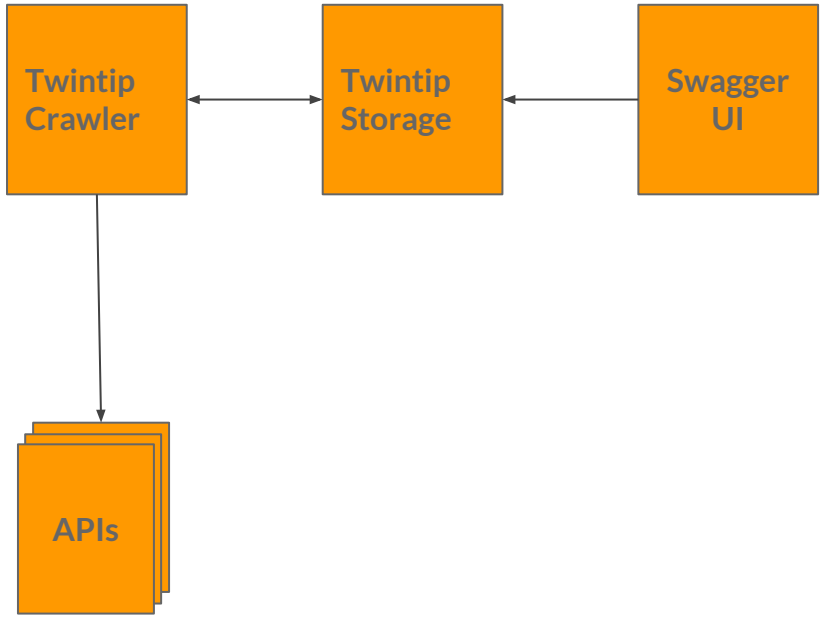
How do teams find available APIs?

API Discovery

API Discovery

A system to crawl and curate all deployed APIs.

https://github.com/zalando-incubator/api-discovery



API Discovery

How do teams choose tech stacks?

Technologists Guild

Technologists Guild

Knowledge sharing and discussion

Owns and curates Tech Radar

Owns internal Tech Compendium

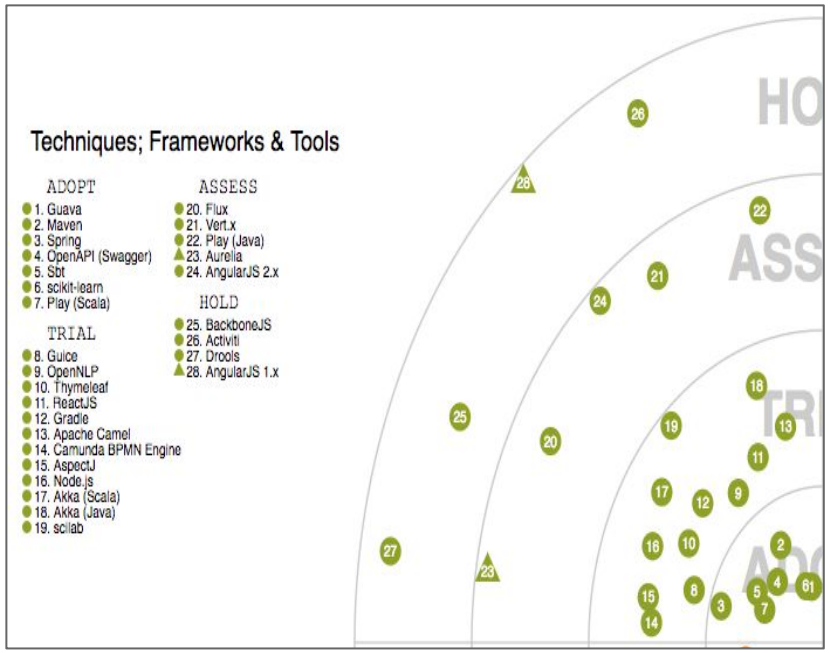
Tech Radar

Tech Radar

We try to give delivery teams guidance in choosing technologies.

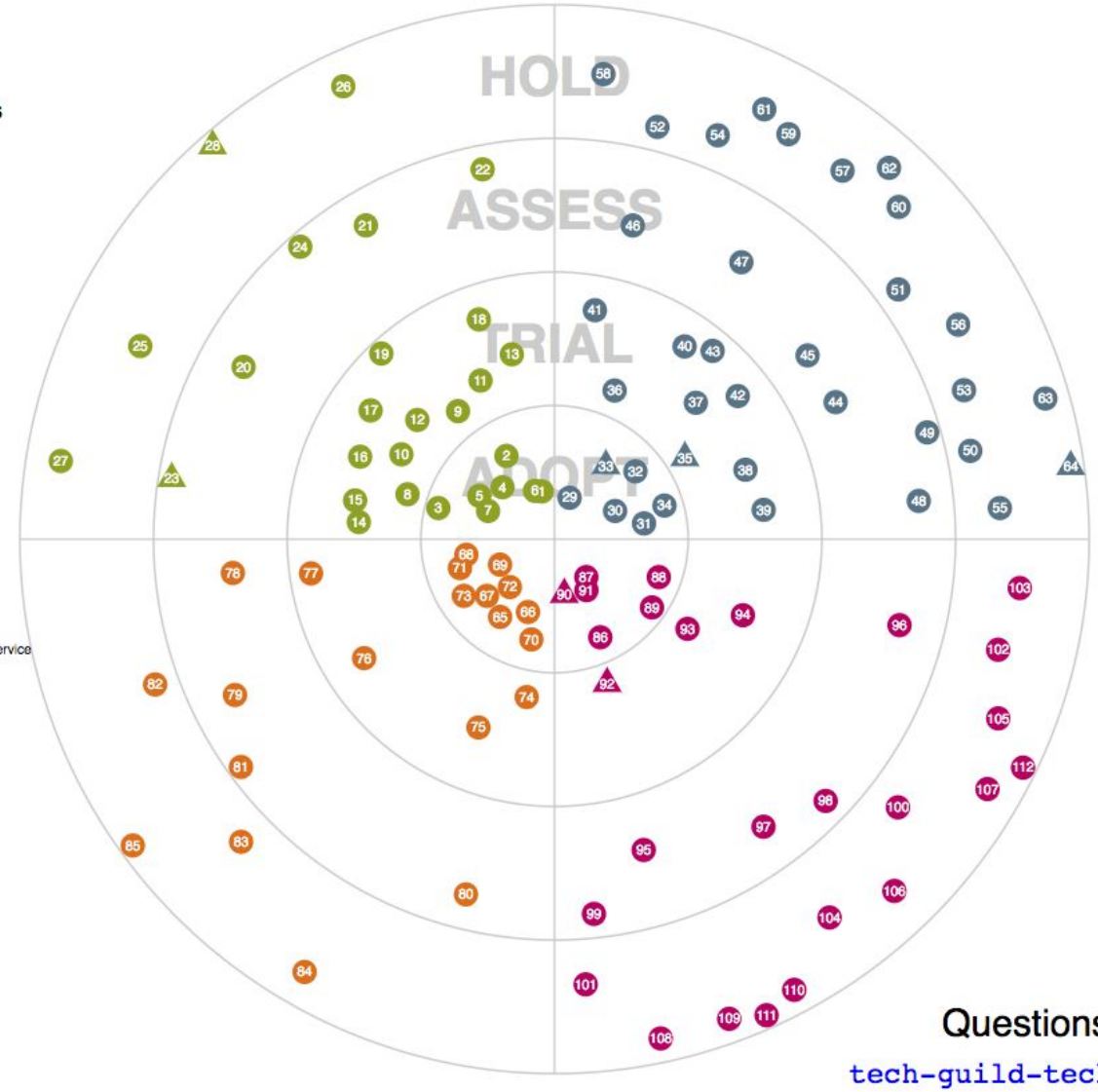
Tech Radar is created by involved engineers in the technologists guild.

Each team is encouraged to contribute experience reports.



https://zalando.github.io/tech-radar

Zalando Tech Radar — 2016.10



Techniques; Frameworks & Tools

ADOPT
- 1. Guava
- 2. Maven
- 3. Spring
- 4. OpenAPI (Swagger)
- 5. Sbt
- 6. scikit-learn
- 7. Play (Scala)

TRIAL
- 8. Guice
- 9. OpenNLP
- 10. Thymeleaf
- 11. ReactJS
- 12. Gradle
- 13. Apache Camel
- 14. Camunda BPMN Engine
- 15. AspectJ
- 16. Node.js
- 17. Akka (Scala)
- 18. Akka (Java)
- 19. scilab

ASSESS
- 20. Flux
- 21. Vert.x
- 22. Play (Java)
- ▲ 23. Aurelia
- 24. AngularJS 2.x

HOLD
- 25. BackboneJS
- 26. Activiti
- 27. Drools
- ▲ 28. AngularJS 1.x

Data Management

ADOPT
- 29. Cassandra
- 30. PostgreSQL
- 31. Redis
- 32. Solr
- ▲ 33. ElasticSearch
- 34. Kafka

TRIAL
- ▲ 35. Nakadi
- 36. Kairos
- 37. Spark
- 38. AWS SNS
- 39. AWS SQS
- 40. AWS EMR
- 41. Google BigQuery
- 42. Hadoop/HDFS
- 43. Flink

ASSESS
- 44. AWS DynamoDB
- 45. AWS Kinesis
- 46. Hadoop/YARN
- 47. Storm
- 48. Google BigTable
- 49. Hadoop/MR

HOLD
- 50. InfluxDB
- 51. RabbitMQ
- 52. Neo4j
- 53. Riak
- 54. Hadoop/HBase
- 55. CouchBase
- 56. HornetQueue
- 57. Aerospike
- 58. Apache Artemis
- 59. MySQL
- 60. Esper CEP
- 61. Graphite
- 62. MongoDB
- 63. ActiveMQ
- ▲ 64. MemCached

Platforms & Infrastructure

ADOPT
- 65. Docker
- 66. Jetty
- 67. ZMON
- 68. log4j2
- 69. STUPS
- 70. Nginx
- 71. Hystrix
- 72. Tomcat
- 73. HAProxy

TRIAL
- 74. etcd
- 75. logback
- 76. Undertow
- 77. Zookeeper

ASSESS
- 78. Mesos
- 79. AWS Simple Workflow Service
- 80. AWS Lambda
- 81. Rocket (rkt)

HOLD
- 82. Grizzly
- 83. log4j
- 84. JBoss
- 85. Puppet

Languages

ADOPT
- 86. Java
- 87. Javascript
- 88. Python
- 89. Swift
- ▲ 90. Go
- 91. Scala

TRIAL
- ▲ 92. Typescript
- 93. Clojure
- 94. ClojureScript

ASSESS
- 95. R
- 96. Elm
- 97. ScalaJS
- 98. Rust
- 99. Kotlin

HOLD
- 100. Elixir
- 101. Haskell
- 102. Erlang
- 103. C/C++
- 104. Coffeescript
- 105. Groovy
- 106. Jython
- 107. PHP
- 108. JRuby
- 109. Perl
- 110. Ruby
- 111. .NET
- 112. Mono

Questions? Comments? Ideas?

tech-guild-technologists@zalando.de
#guild-technologists

● unchanged ▲ changed since last edition (2016.04)

Scaling Architecture @ Zalando

Zalando in general	Architecture specific
Autonomous delivery teams	API Guild + Guidelines
Overarching architecture team	API Reviews
Rules of Play	API Discovery
Architecture Principles	Technologists Guild
Open Source Culture	Tech Radar

Thanks.
Questions?

Twitter: @fmueller_bln