Microservices, geerdet
René Lengwinat
inovex GmbH
https://keybase.io/rlengwinat

Michael Bruns
inovex GmbH
https://keybase.io/michaelbruns
Why Microservices?

› Abstract problem split into small solutions

› Monolith would simply be too hard to handle

› Large portions of the application don’t need to know about the rest of the application
Why Microservices?

› Different people with different backgrounds and skills
› Load peaks on small parts of the application are expected
› Most important: We, i.e. the developers, wanted to do it!
When to be careful...

› Real-life conversation:
  › Manager: “We need Microservices!”
  › Dev: “Ok. Why?”
  › Manager: “Err...”
  › Dev: “Did you at least talk to Dev and Ops?”
  › Manager: “Err...”
When to be careful...

› Real-life conversation:
   › Dev: “Can we operate the VMs with our team?”
   › Manager: “No, the Ops team does that.”
   › Dev: “Can we shift people from the Ops team to our team?”
   › Manager: “No.”
   › Dev: “Ok... Then I need to talk to the Ops team.”
   › Manager: “You can’t, they’re too busy with their roadmap.”
When to be careful...

› Real-life conversation:

› Dev: “We need two small VMs, one per data center.”

› Ops: “We don’t do it that way. You’ll get three per data-center, 64 GB RAM and 6 CPUs each.”

› Dev: “What?! Well, alright... Is it possible to get them by this afternoon?”

› Ops: “Nah, average delivery time is eight weeks.”
Who am I? - Session Management

$n_{RequestsOnSessionService} = n_{Requests} \times n_{Services}$

Old school

Old school in new env

New Way

SPOF

SPOF + BOTTLENECK

Manageable Risk
Secret Sauce - The JWT - JSON Web Token

```json
{
   "alg": "HS256",
   "typ": "JWT"
}
```

```json
{
   "sub": "1234567890",
   "iss": "serviceA",
   "exp": "1460667024",
   "name": "John Doe",
   "rights": ["READ", "WRITE"]
}
```

```plaintext
base64(header) + "." + base64(claims) + "." +
base64(hmacSha256(secret, base64(header) + "." + base64(claims)))
```

Signature
Secret Sauce - The JWT - JSON Web Token

- Applications
  - Sessions
  - Account confirmation
  - Password reset
  - Short-lived tokens (app to web handshake)
FRONT ENDS!

SO MANY FRONT ENDS!

Source: https://i.imgflip.com/125yfz.jpg
What about the UI?

Monolithic Service

- Header
- Content I
- Content II
- Content III
What about the UI?
What about the UI?

Service I

Service II

Service III

Front end Service

Header

Content I

Content II

Content III
What about the UI?
What about the UI?

- S3 Bucket as CDN
- Mustache templates
- Lightweight, no SPAs
- Planned: Web Components
EAT ALL THE CAKE
My code, your code - Shared vs Template

› What’s wrong with shared code?
  › Reuse
  › Coupling
  › Opinionated

IT DEPENDS -> organisational structure, maturity
My code, your code - Shared vs Template

› Alternative -> Template Project
  › Yeoman, leiningen, git repo
  › Fast setup
  › Doesn’t fit? Change / Fork
  › Standards embedded
  › Updates
BOILERPLATE INFRASTRUCTURE?

THAT'S A PADDLIN'!
Don’t build a platform, use one - in our case: AWS

Use what’s already there: EC2, RDS, S3, Elasticsearch, Kinesis, Route53, ...

Add shared stuff (e.g. JVM, nginx) to base image

Leave the rest to the services
Foster DevOps, i.e. tear down all political and technological barriers

Choose the right tool for the right job:
Terraform, Go CD, Prometheus, flood IO, ...

Make your infrastructure reproducible
IaaS - Yet Another Platform?

› Don’t build a gateway service if it’s not necessary

› Start small, expect growth

› Geo-redundancy

› Monitor your stuff, send alarms
IaaS - Yet Another Platform?

› Short insight:

› 40+ EC2 instances, 20+ Auto-Scaling Groups, 50+ Security Groups, 500+ GB in S3 Buckets

› ~20m documents stored in ~3k shards in Elasticsearch

› 40+ Go CD pipelines, >100 Git Repositories

› ~15 people working in 5+ locations
TOO MUCH TECH DEBT?

TOLD YOU SO!
Many services = Technical debt?

› Local debt vs global debt
› Don’t be afraid of tracking publicly
› Track across service boundaries
› Continuously review tech debt
› Continuously reduce tech debt
Many services = Technical debt?

› Continuous Tech Review
  › Weekly review
  › Track and attack tech debt
  › Discuss possible improvements
  › Discuss risks
  › Time boxed!
Many services = Technical debt?

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service A</td>
<td>Service B</td>
<td>Service C</td>
<td></td>
</tr>
<tr>
<td>Error Page</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Json error logging</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSL Certs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Follow new Makefile convention</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set licence headers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Legende</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>doesn't apply</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>done</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>todo</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Service Commons Backlog
MICROSERVICES?

WAY TO GO!

Source: https://imgflip.com/i/12lr94
Vielen Dank

René Lengwinat &
Michael Bruns

inovex GmbH
Ludwig-Erhard-Allee 6
76131 Karlsruhe

rene.lengwinat@inovex.de
michael.bruns@inovex.de