



KubeCon



CloudNativeCon

Europe 2019

Oops!...I Did It Again...

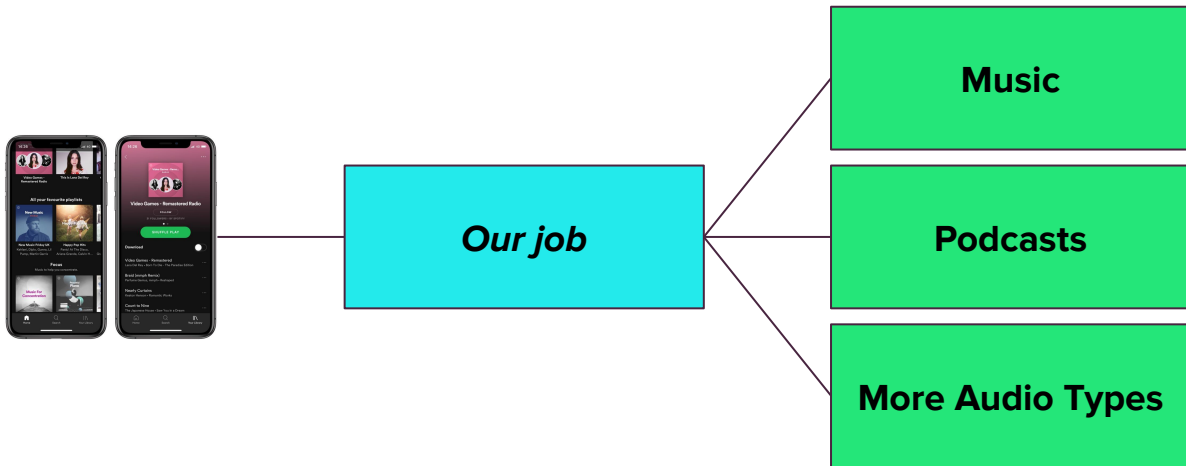
We Built gRPC

Austin Lamon
Dave Zolotusky



Hi there, I am Austin and this is Dave

We build the infrastructure that Spotify runs on!



But, let's just say, it's not quite that simple...

Spotify is everywhere!

Literally: spotify-everywhere.com, making it challenging to scale and maintain reliability

+150
Speakers

+40
Smart Devices

+15
TV Platforms

+70
Countries

Core Technologies

We leverage many CNCF technologies to bring Spotify to life for customers



However, it wasn't always this way..

Data Centers

First, we managed data centers



Data Center Era

- Self managed data centers
- Rapidly growing company
- Challenging engineering problems to handle scale



Cloud Hosted Era



Cloud Native Era

In the Cloud

Next, we moved to the cloud but carried technical debt with us



Data Center Era

Cloud Hosted Era

Cloud Native Era

- Forklifted to GCP
- Carried our in-house technologies built for the datacenter with us

Cloud Native

Finally, we shifted to cloud native technologies to improve speed and cost efficiency in the cloud



Data Center Era



Cloud Hosted Era



Cloud Native Era

Today!

Which jump was harder? Jump #1 or #2?



Yeah, Jump #2, but why?



Data Center Era



Cloud Hosted Era

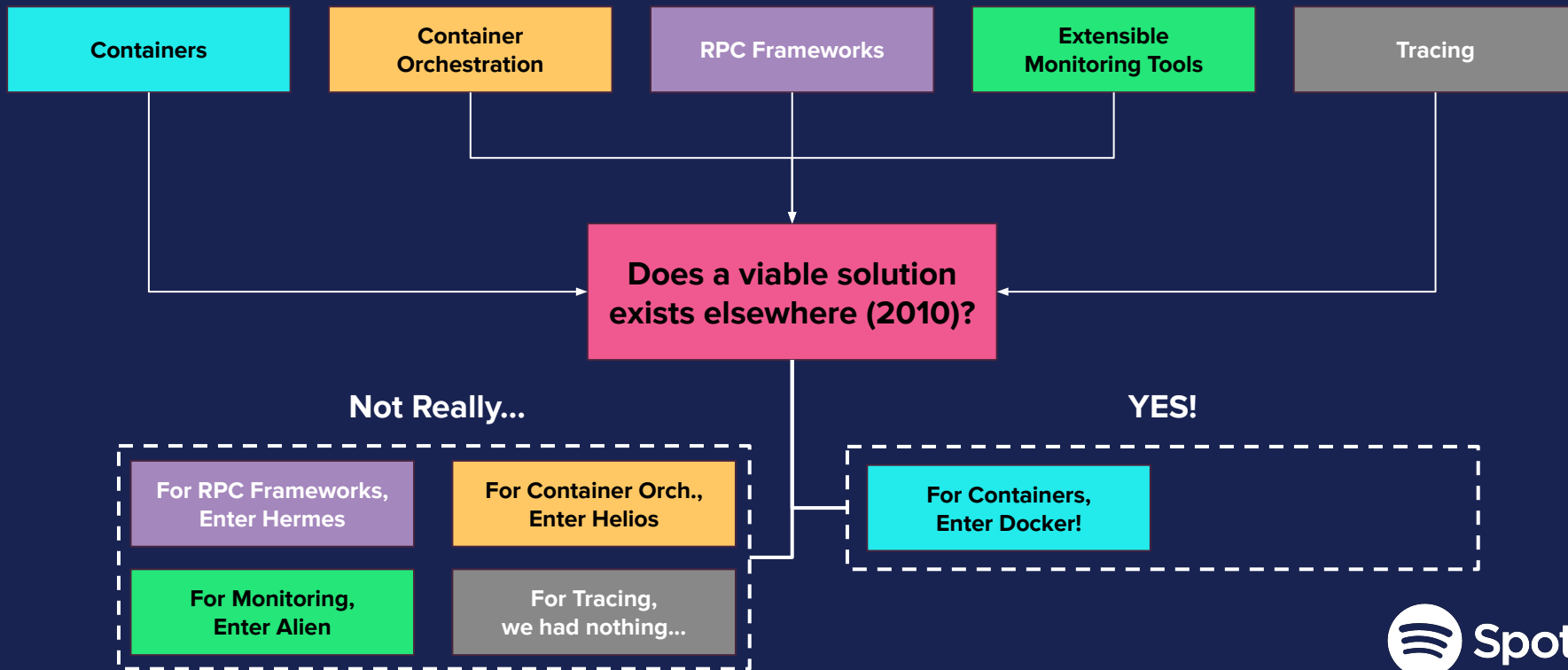
Jump #2



Cloud Native Era

Stepping Back to 2010

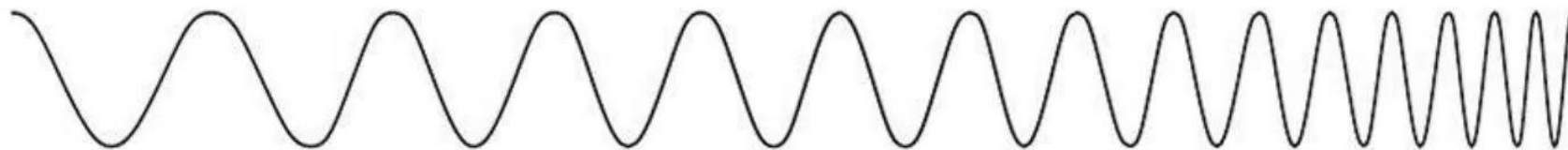
In the datacenter, we'd built technologies to solve key problems that lacked an industry standard



So, Open Source?

We open sourced many of these technologies, but didn't invest in building community around them

Increasing Investment Focus



Community Marketed

Feature Requests

Documented

Tested

Reliable & Helping
Spotify Grow!

In short, we tried and were varying degrees of successful, but, we'd essentially built some rock solid internal tools

Positive Side Effects

These tools didn't come without side effects

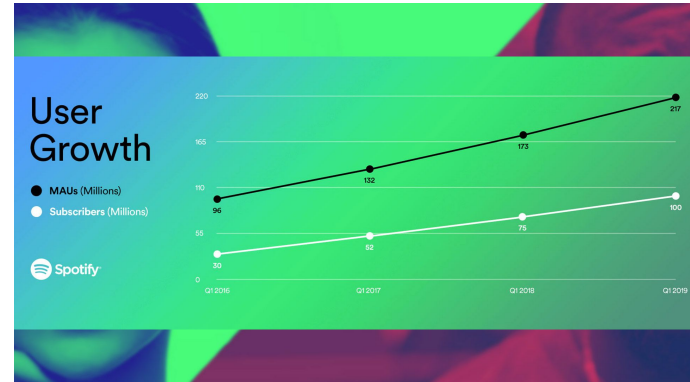
①

We're still here!



②

Our infrastructure has scaled!



③

Most importantly, we're confident in our ability to solve hard problems

Less Positive Side Effects

These tools didn't come without side effects

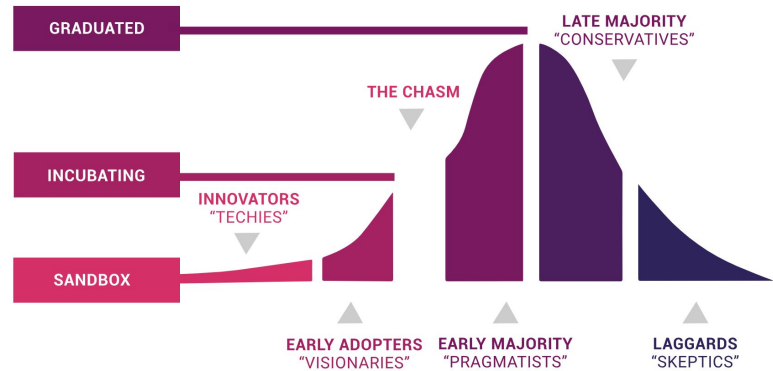
①

Occasional divergence from the industry



②

Bias toward building versus adopting





We hear we're not alone!

(and want to help you learn from us)



Tackling the Less Positive Side Effects

These tools didn't come without side effects

①

Occasional divergence from the industry

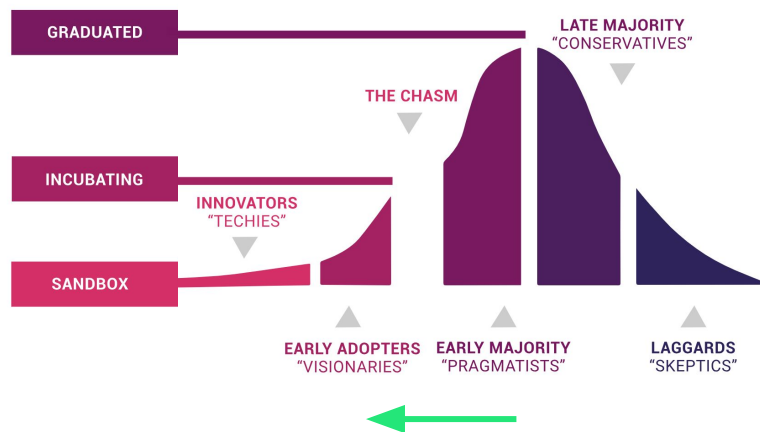


Easy, we're migrating:

- gRPC
- Kubernetes
- Grafana
- OpenTracing

②

Bias toward building versus adopting



Very hard, we're shifting our mindset and investment tactics to the left

Tackling the Less Positive Side Effects

These tools didn't come without side effects

①

Occasional divergence from the industry



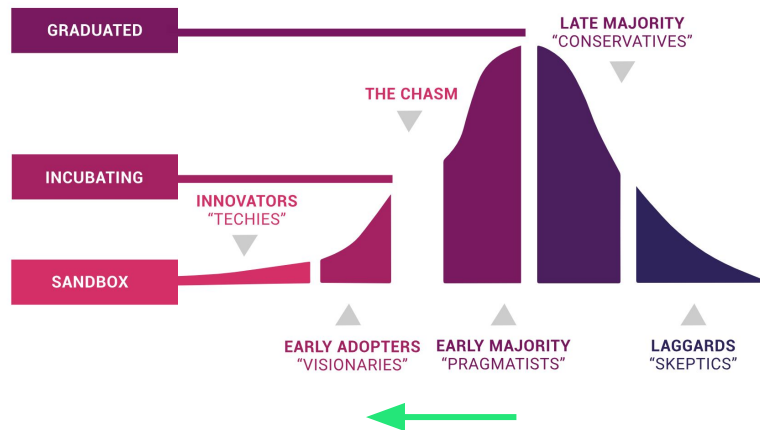
Easy, we're migrating:

- gRPC
- Kubernetes
- Grafana
- OpenTracing

Let's look at gRPC as a case study for this transition!

②

Bias toward building versus adopting



Very hard, we're shifting our mindset and investment tactics to the left

gRPC: Shifting Mindset

Spotify leveraged an internal RPC framework throughout the stack

HERMÈS

- Establish: 2012-ish
- Sort of an RPC framework internal to Spotify
- Well supports existing Spotify use cases
- Maintained, documented and supported by Spotify for Spotify
- Used throughout the Spotify stack

gRPC

- Established: 2015-ish
- Open Source RPC Framework
- Maintained, documented and supported by everyone for everyone

So why transition?

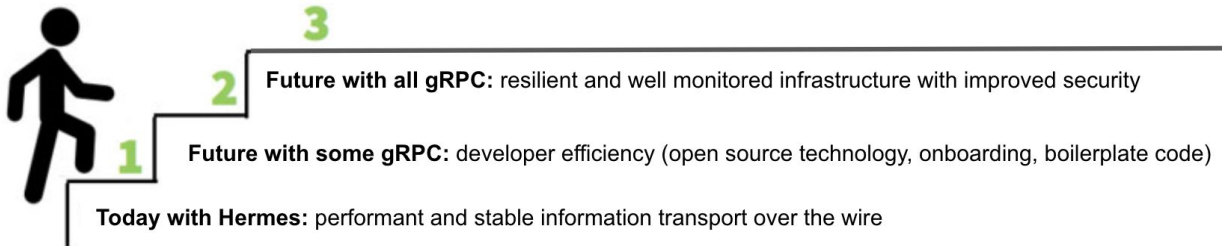
It works...



WINS!

How we tried to sell people on gRPC @ Spotify...

- ▶ **Less boilerplate**
- ▶ **Quicker onboarding**
- ▶ **Easier integration with 3rd party tooling**
- ▶ **Resilience patterns**



WINS!

How we tried



Developer happiness

- What developers care about
 - Quickly shipping bug-free code...
 - ... so that they can solve customers' problems
- What developers **don't** care about
 - Setting timeout values
 - New RPC technologies



WINS!

How we actually sold people on gRPC @ Spotify

- ▶ ~~Less boilerplate~~
- ▶ ~~Quicker onboarding~~
- ▶ ~~Easier integration with 3rd party tooling~~
- ▶ ~~Resilience patterns~~

These are infra
team problems!

- ▶ **Message Integrity, Encryption, Authentication**
- ▶ **Consistency, Global Monitoring**
- ▶ **Strong contracts for all services**
- ▶ **Ease of integration for new tools and languages**

Strategies we've learned in moving to gRPC

1. Talk to people in terms of their problems, not why you think gRPC is good or Hermes sucks
2. Actively call-out 'not invented here' and explain the upside of focusing on Spotify differentiating problems versus building a gRPC competitor
3. Be more active in the community, you learn a lot and move a lot faster (also, cool people!)

“Does a viable solution exists elsewhere” isn’t good enough...



Needs persist longer than Solutions

COMMUNITY!

Let's touch on “Does a viable solution exists elsewhere”
again

COMMUNITY!

- ▶ gRPC on phones
- ▶ gRPC-Web
- ▶ Internal to Spotify [#grpc-users](#)

*We hope our learnings
can help you!*

