Christoph Neijenhuis

Escaping the Lock-In

commercetools

@c_neijenhuis

aws

Google Cloud

Azure

iron.io

cloudevents

Serverless Workgroup

Cloud Native Computing Foundation

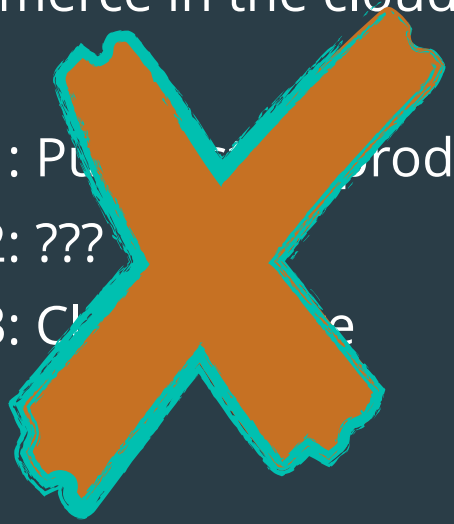
commercetools - Product Vision

- eCommerce in the cloud

- Step 1: Put existing product into VM
- Step 2: ???
- Step 3: Cloud-Native

commercetools - Product Vision

- eCommerce in the cloud
- Step 1: Pu[illegible]roduc[illegible]
- Step 2: ???
- Step 3: Cl[illegible]e



Doesn't scale (at least to zero)

Monolith

Ops heavy

NOT serverless

commercetools - Product Vision

- Cloud-Native eCommerce

- Multi-Tenant
- API-first, event-driven
- Auto-Scaling
- Uptime and Performance guarantees

- Fits perfectly into a serverless ecosystem

commercetools -

uarantees

ss ecosystem

State of the cloud in 2011

- Serverless didn’t exist
- AWS was already the king

- #2 public cloud was…

  Rackspace Cloud!

- Azure?
  - Still called “Windows Azure”
  - No support for Linux (until 2012)
- Google Cloud?
  - App Engine
  - Cloud Storage
  - Compute Engine came in 2012

Fast-forward to 2014...

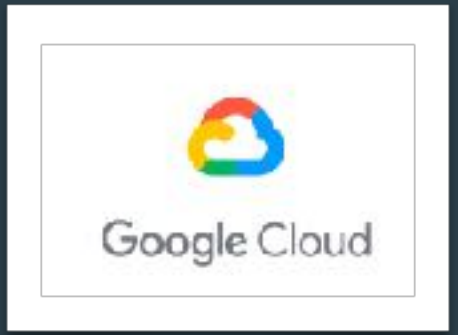
We need to get off Rackspace Cloud!

In 2015 

In 2016 

In 2017 

Customer Demand drives tech choices



- Limited technology choices
- Day-to-day multi-tasking hell
- Is the London region called…
  - eu-west-2
  - europe-west2
  - uk-south

Let’s define Lock-  **Out**

Let’s define Lock-Out

- Protect a file with a password
- 4 digit password

1234

- How to hack it?
  - Enumerate all possible passwords
  - Try each possible password

0000
0001
0002
0003
…
9998
9999

Let’s define Lock-Out

- Protect a file with a password
- 4 digit password
- 16 character password
- How to hack it?
  - Enumerate all possible passwords
  - Try each possible password

abcDEFghIJ1234#$

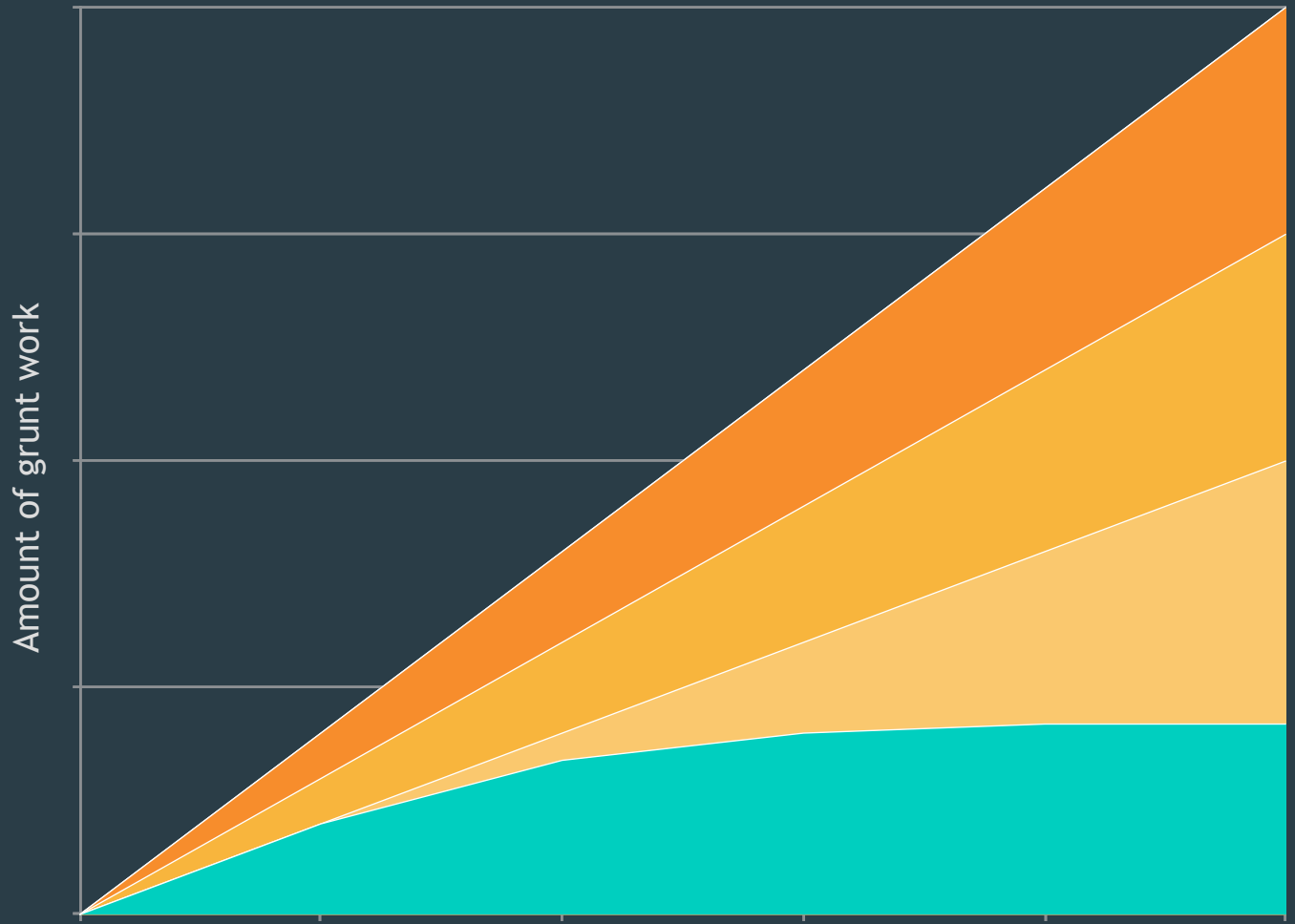
aaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaab
aaaaaaac
aaaaaaad
…
$$$$$$$$$$$$$$$#
$$$$$$$$$$$$$$$$

Definition Lock-C[illegible]: In

Grunt work >> work you're able to afford

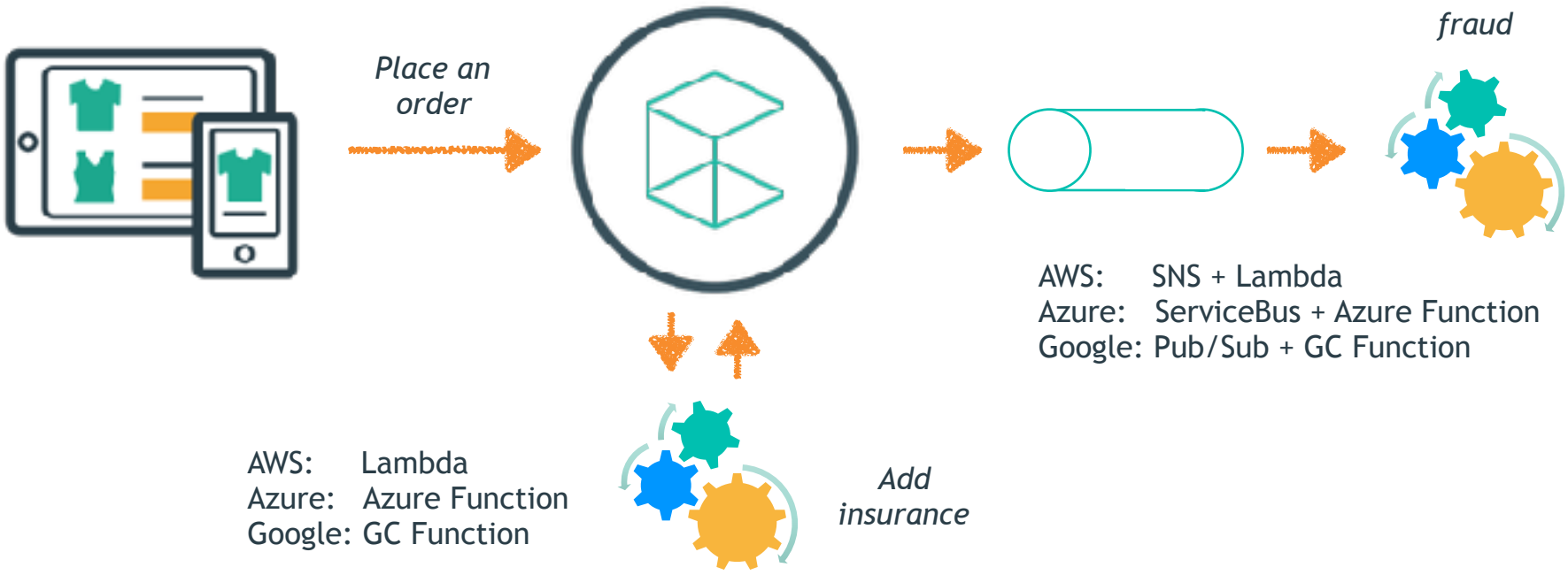
time

cost


Amount of grunt work
Size of code base

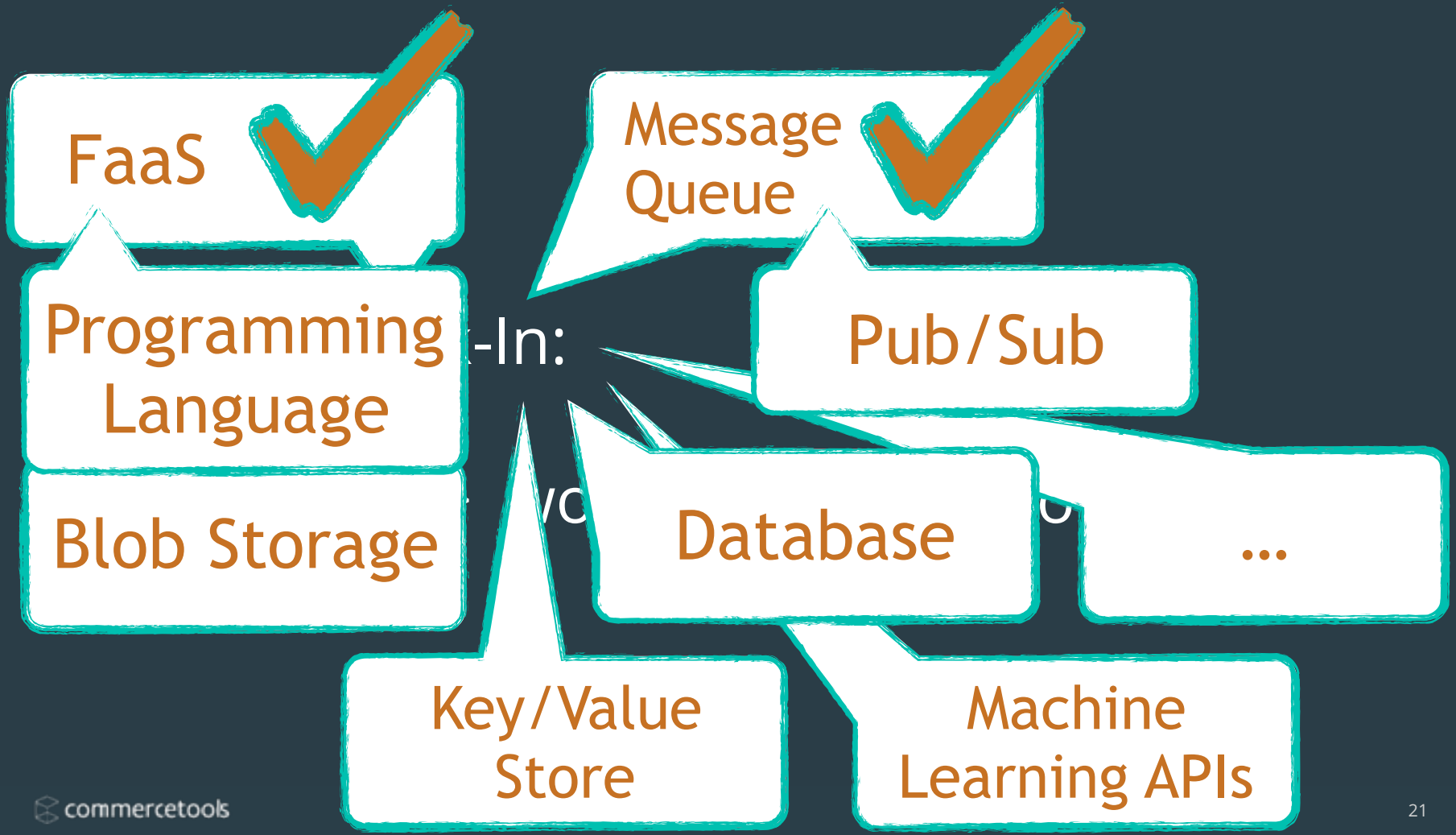
Keep use of proprietary APIs constant

Live Demo


Place an order
Detect fraud
AWS: SNS + Lambda
Azure: ServiceBus + Azure Function
Google: Pub/Sub + GC Function
AWS: Lambda
Azure: Azure Function
Google: GC Function
Add insurance

Summary Live Demo

- Functions from AWS, Azure and GCP share same fundamental concepts
  - Responds to synchronous events, such as HTTP requests
  - Responds to asynchronous events, from Message Queues
- The proprietary APIs are quite different
- A unified API can be established, and adapters be written for each FaaS
  - The unified API can be specific to your use case
  - You could also use a standardised API (e.g. express.js)


FaaS
Message Queue
Programming Language
-In:
Pub/Sub
Blob Storage
Database
...
Key/Value Store
Machine Learning APIs

Can an adapter be written?

- Same concept?
- Same feature set?

Messaging Infrastructure

1:1
AWS SQS, Azure ServiceBus Queues

1:n
AWS SNS, Azure EventGrid, Google Pub/Sub, …

Can an adapter be written?

- Same concept?
- Same feature set?

Messaging Filtering

Filters by message attributes:
AWS SNS, Azure EventGrid

Does not filter:
Google Pub/Sub, AWS SQS,
…

Can an adapter be written?

- Same concept?
- Same feature set?

Ordering

With ordering:
AWS SQS, Azure EventHub

Without ordering:
AWS SNS, Azure EventGrid, Google Pub/Sub, …

Can an adapter be written?

- Same concept?
- Same feature set?

Message Size

Google Pub/Sub:
16 MB

AWS:
256 KB

Azure EventGrid:
64 KB

Can an adapter be written for a given product?

- Conceptually simple
- Feature set is similar across different products, or you limit yourself to common feature set

- Examples:
  - Blob storage
  - Key/Value store

- Conceptually complex product with lots of features
- New, innovative product
  - No comparable product available

- Examples:
  - API Gateways
  - Function Orchestration
  - AI/ML products

Pick products that are generally available

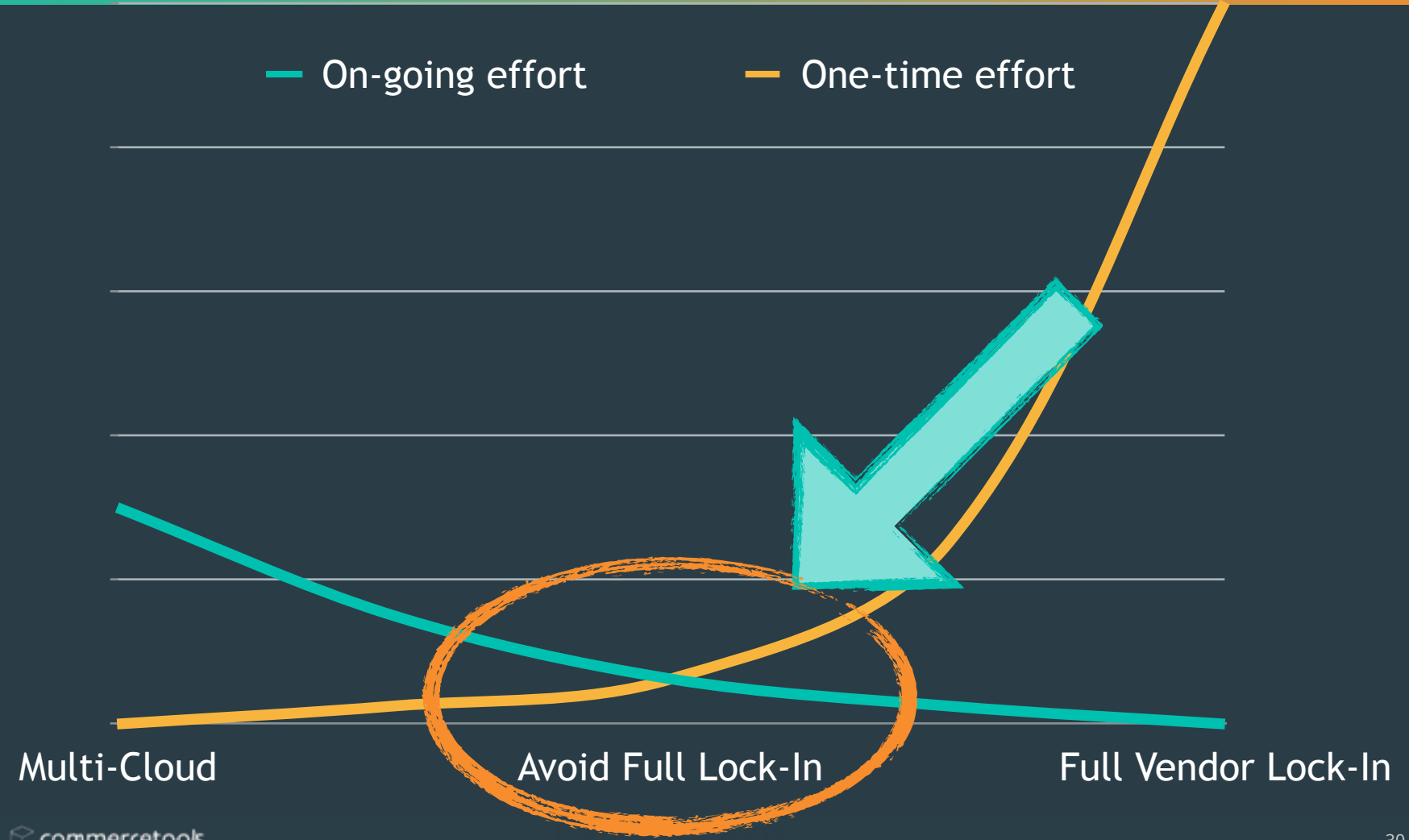
- Cloud vendors offer some popular open source products
  - MySQL, Postgres
- Most open source vendors offer their products managed, on multiple clouds
  - MongoDB - AWS, Azure, Google Cloud
  - Elastic search - AWS, Google Cloud
- API-first products not affiliated with a cloud vendor
  - Auth0, Twilio

Support standards

- Why was it rather easy to write an adapter for our functions?
  - In part, because HTTP is a standard!
- SQL is supported in many new cloud products
  - Google Cloud Spanner
  - Azure Cosmos DB

- CloudEvents: Standard for describing events

Conclusion


On-going effort
One-time effort
Multi-Cloud
Avoid Full Lock-In
Full Vendor Lock-In

- You’re truly locked-in if the work to migrate is unaffordable
  - Some work is to be expected
  - The goal is to keep the work not only relatively small, but constant
- FaaS share the same underlying concepts
  - If you stick to a common feature set, it’s possible to write functions that run on AWS, Azure and Google Cloud
- Products besides the FaaS are more likely to lock you in
  - Use Open-Source products, use standardised products

commercetools

We're hiring!

Thank you!

@c_neijenhuis