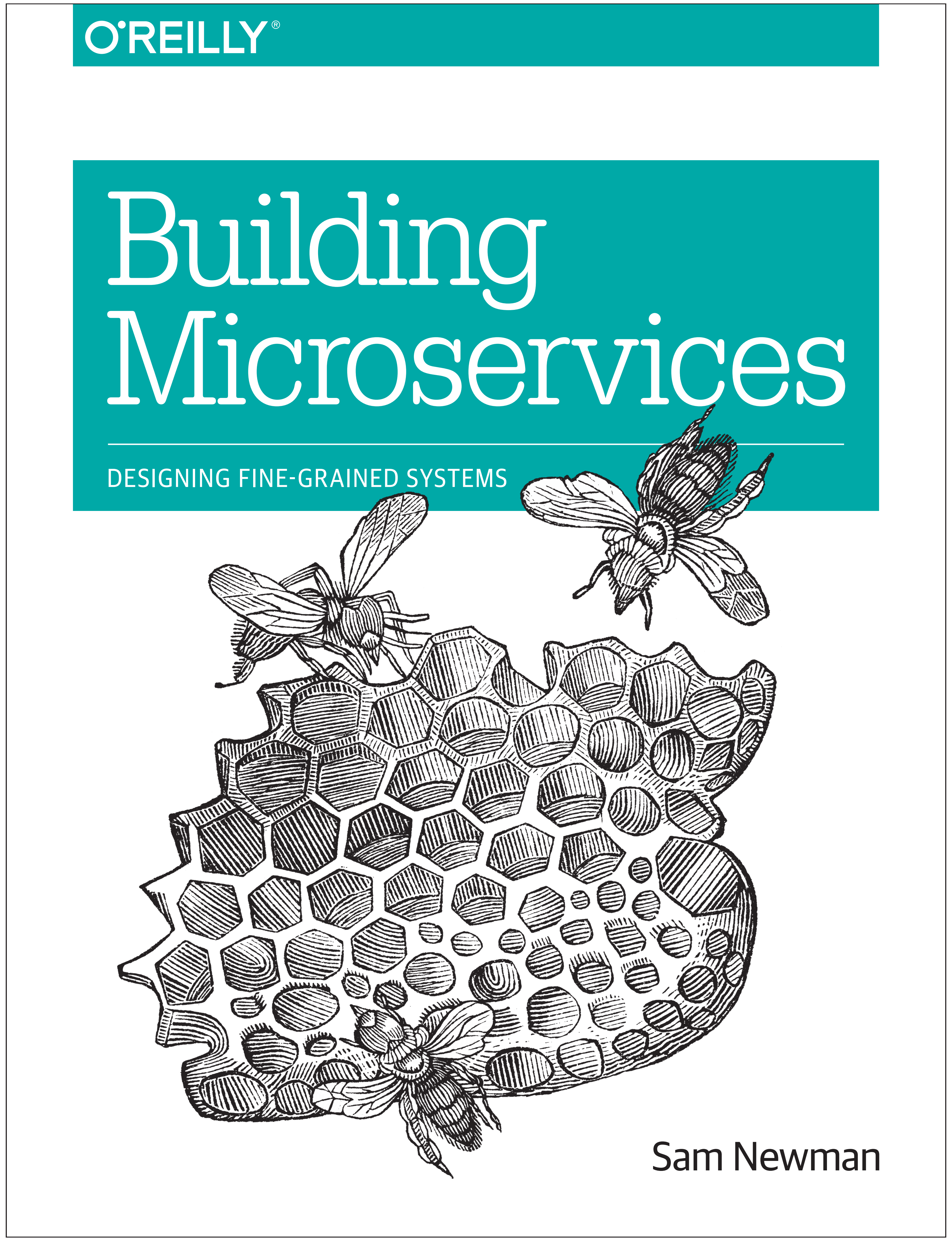
WHERE WE'RE GOING, WE DON'T NEED SERVERS!

SAM NEWMAN

https://www.flickr.com/photos/mindfrieze/4297260599/i

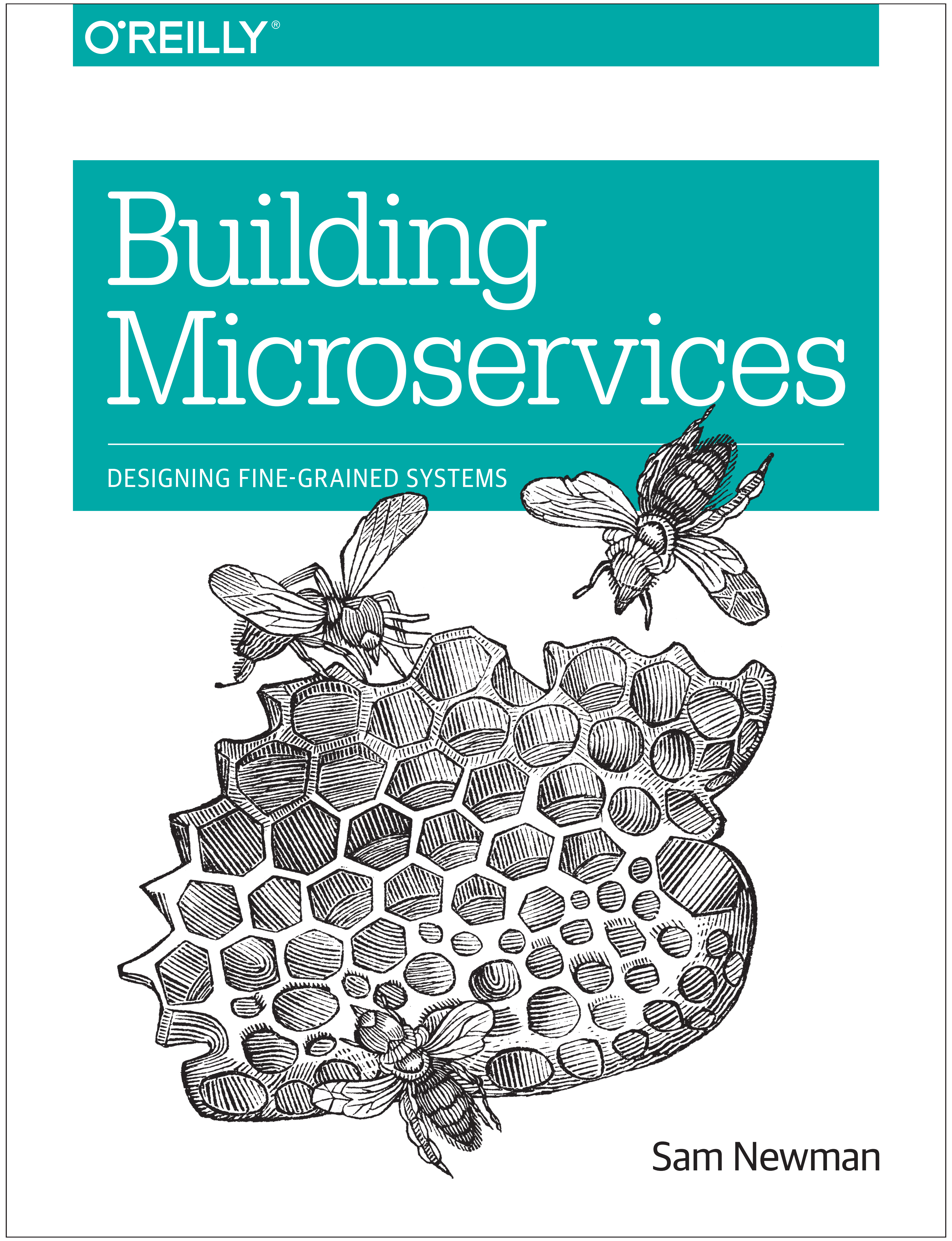


O'REILLY®

Building Microservices

DESIGNING FINE-GRAINED SYSTEMS

Sam Newman

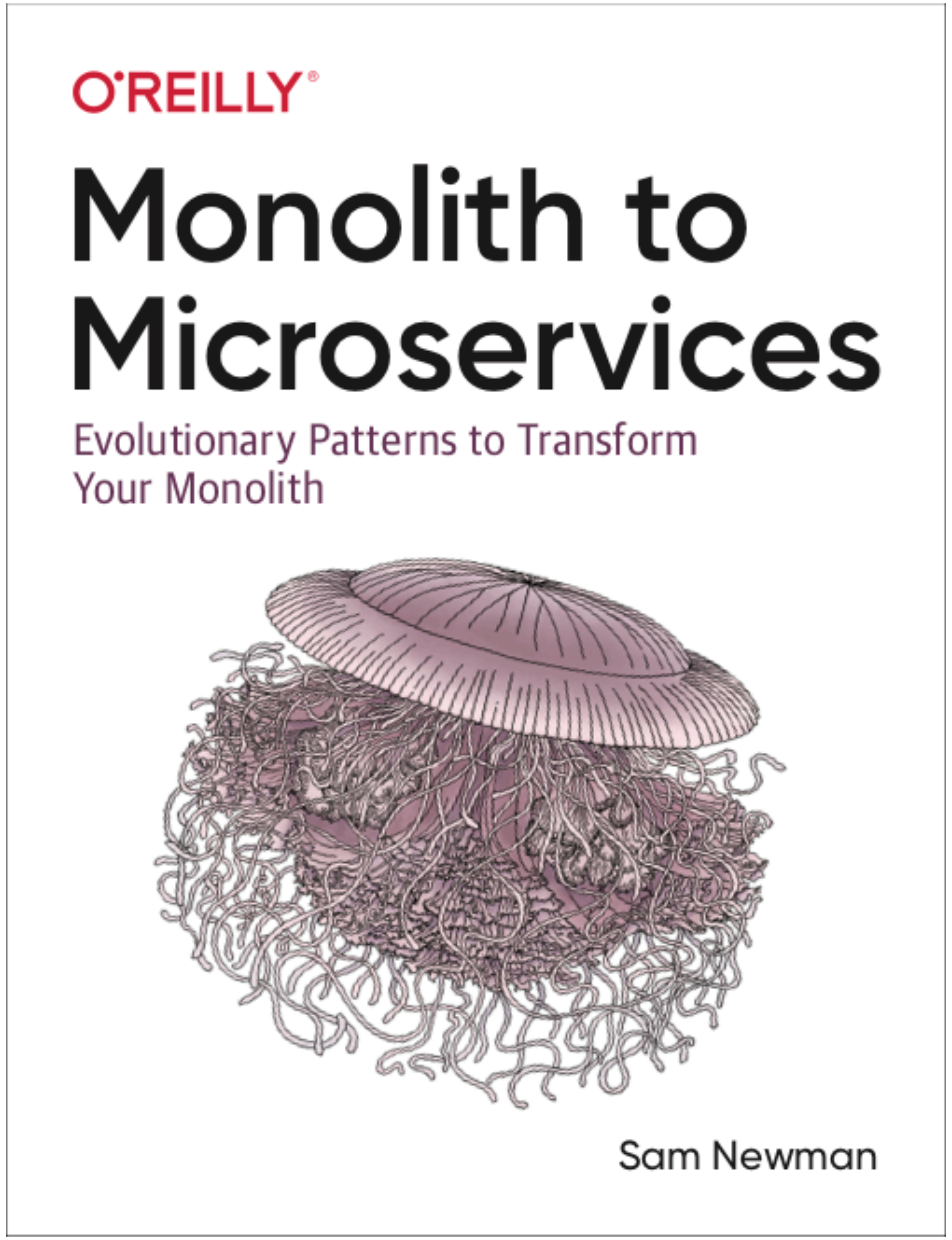


O'REILLY®

Building Microservices

DESIGNING FINE-GRAINED SYSTEMS

Sam Newman



O'REILLY®

Monolith to Microservices

Evolutionary Patterns to Transform Your Monolith

Sam Newman

Part 1: Cloud and Serverless

Part 1: Cloud and Serverless

Part 2: Microservices and Functions

Part 1: Cloud and Serverless

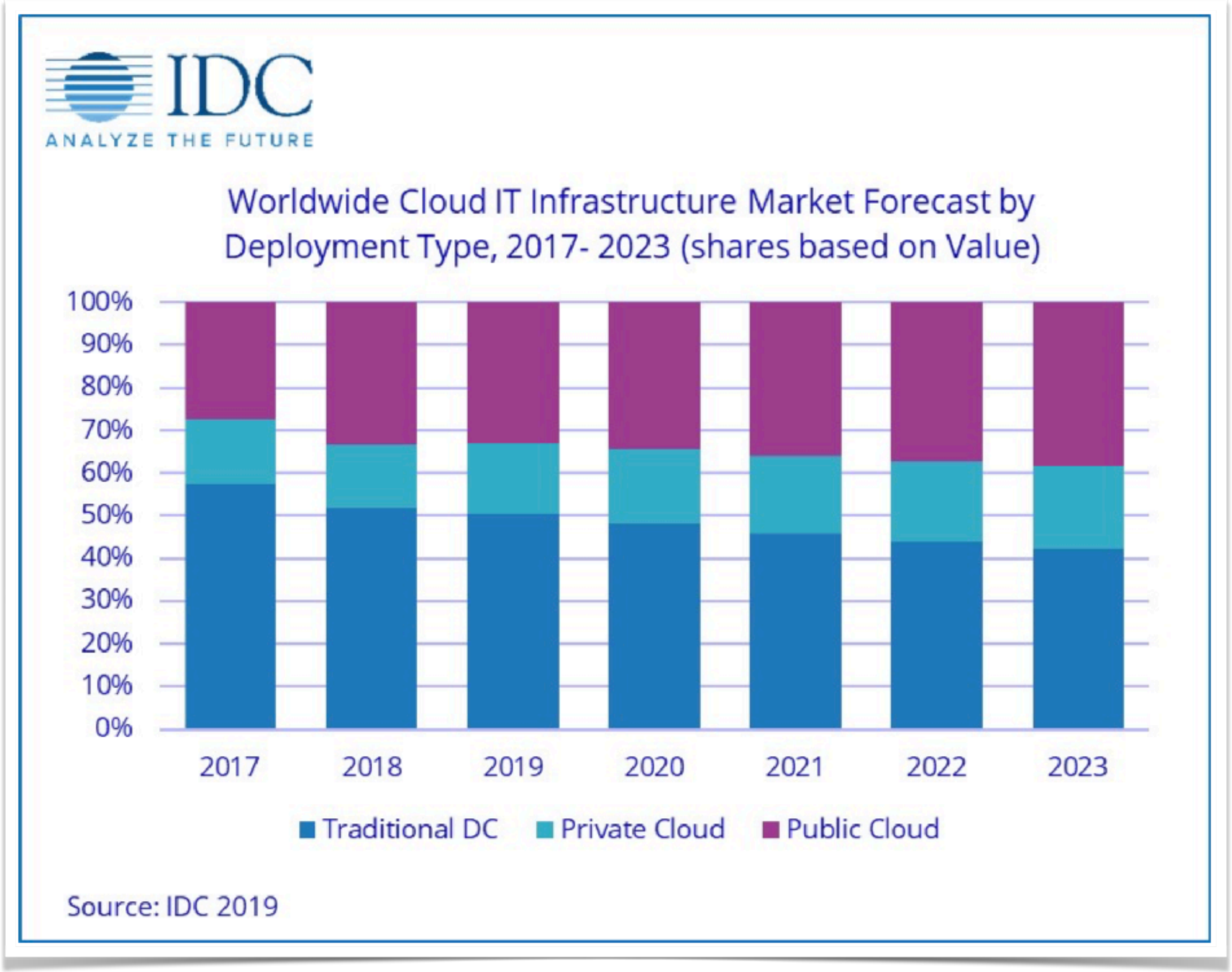
Part 2: Microservices and Functions

Part 3: What Should You Do About It?

Part 1: Cloud and Serverless

https://www.flickr.com/photos/99095055@N04/15240217372/

We are in love with our machines

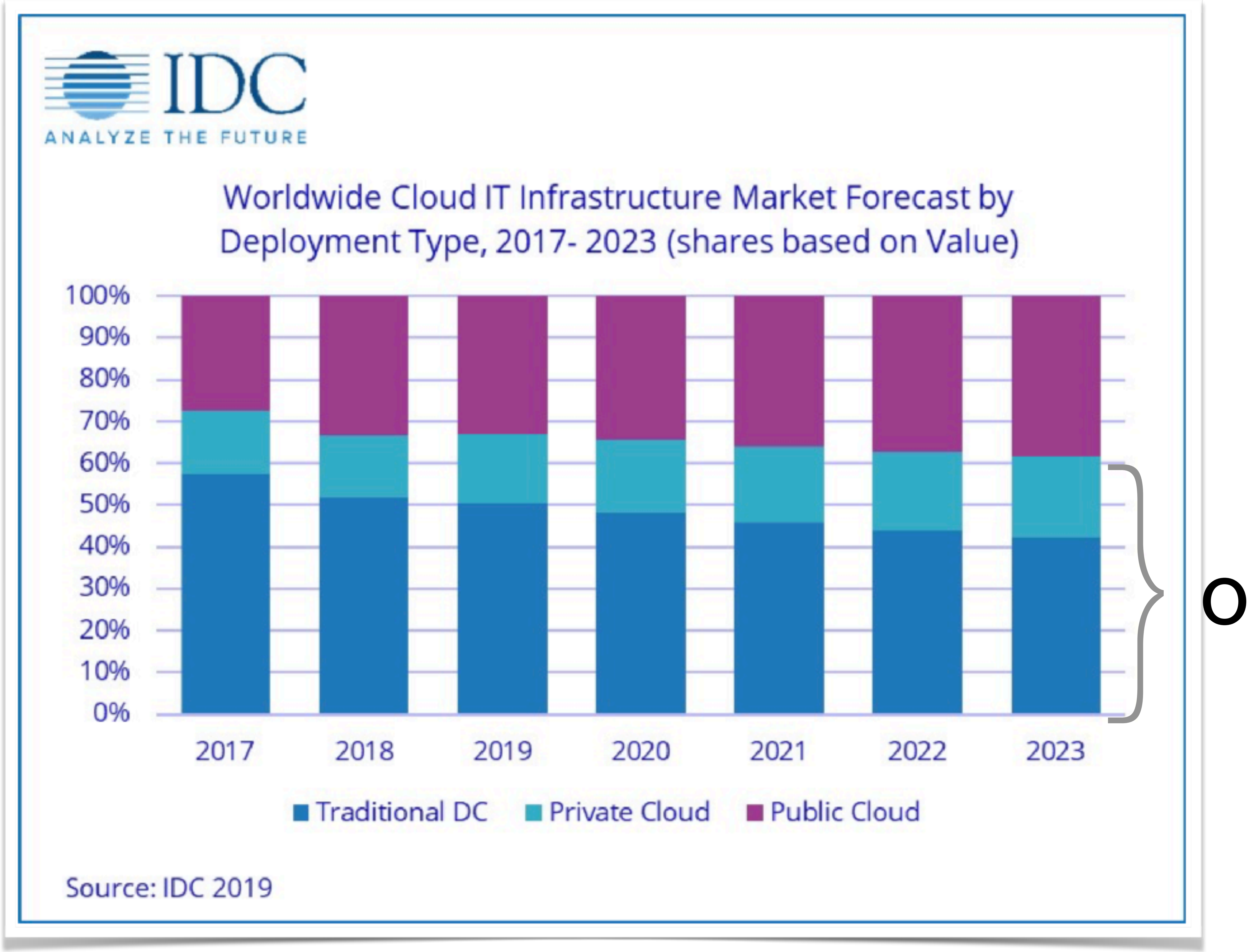


Worldwide Cloud IT Infrastructure Market Forecast by Deployment Type, 2017- 2023 (shares based on Value)

Source: IDC 2019

https://www.idc.com/getdoc.jsp?containerId=prUS45293719

@samnewman

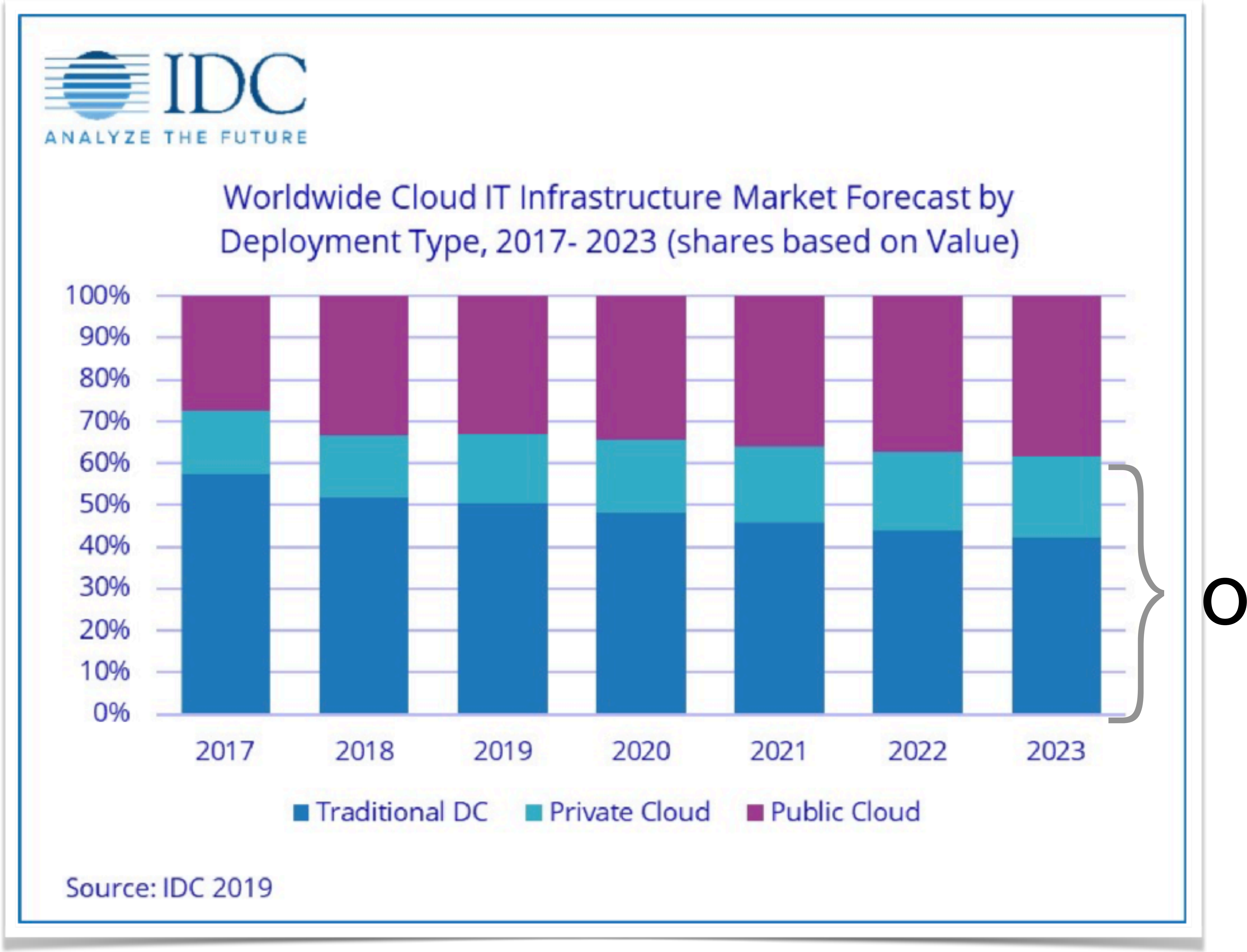


On-premise spend

https://www.idc.com/getdoc.jsp?containerId=prUS45293719

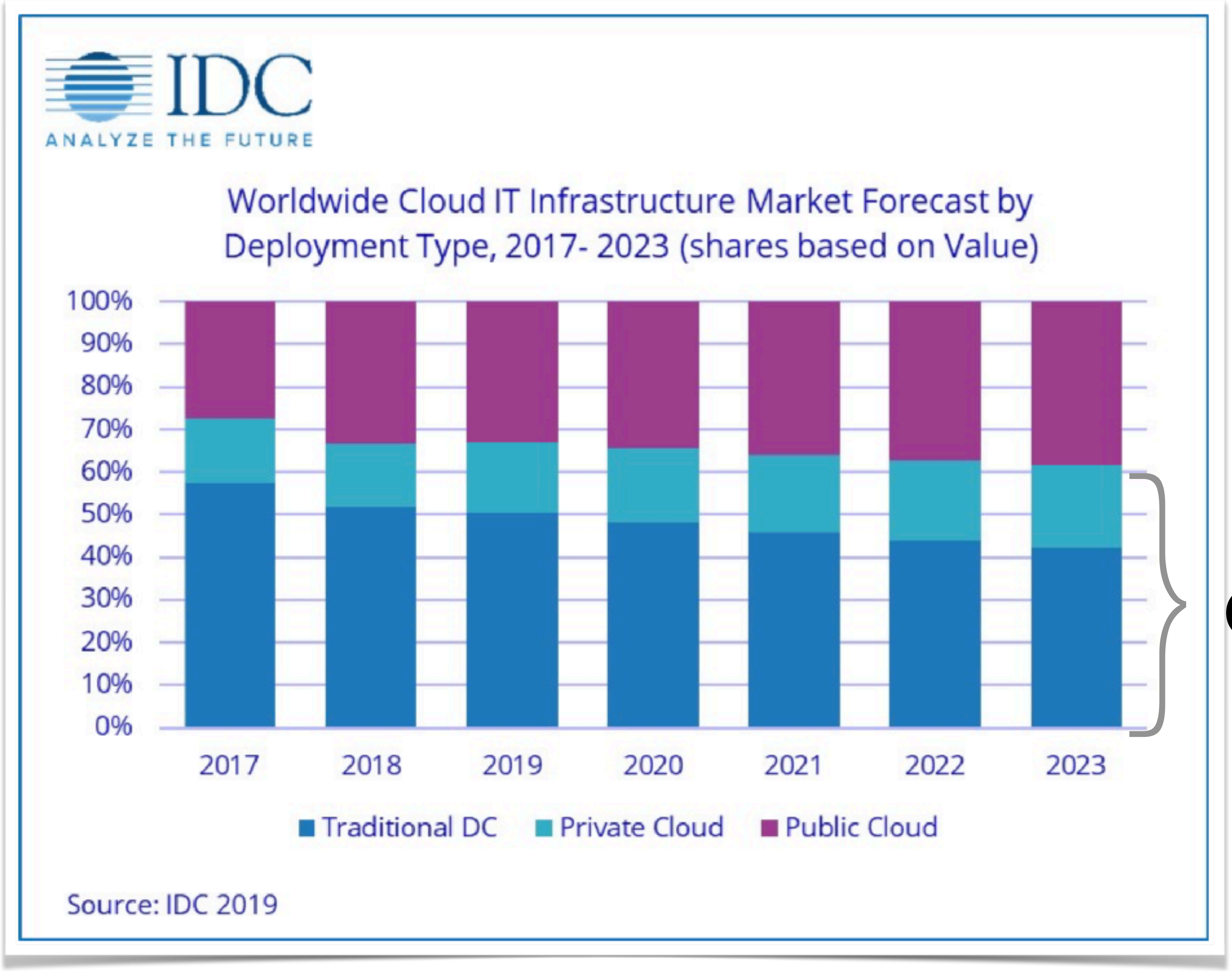
@samnewman

70% of all IT spend is on premise as of 2019



On-premise spend

https://www.idc.com/getdoc.jsp?containerId=prUS45293719



Worldwide Cloud IT Infrastructure Market Forecast by Deployment Type, 2017- 2023 (shares based on Value)

Source: IDC 2019

On-premise spend

70% of all IT spend is on premise as of 2019

Little change since 2018, despite forecasts

https://www.idc.com/getdoc.jsp?containerId=prUS45293719

https://www.youtube.com/watch?v=dKjCWfuvYxQ

A BRIEF (AND INCOMPLETE) HISTORY OF INFRA

Physical Infrastructure

A BRIEF (AND INCOMPLETE) HISTORY OF INFRA

Physical Infrastructure

Virtualised Infrastructure

Early 2000

A BRIEF (AND INCOMPLETE) HISTORY OF INFRA

Physical Infrastructure

Virtualised Infrastructure

Early 2000

OpenStack

2010s

A BRIEF (AND INCOMPLETE) HISTORY OF INFRA

Physical Infrastructure

Virtualised Infrastructure

Early 2000

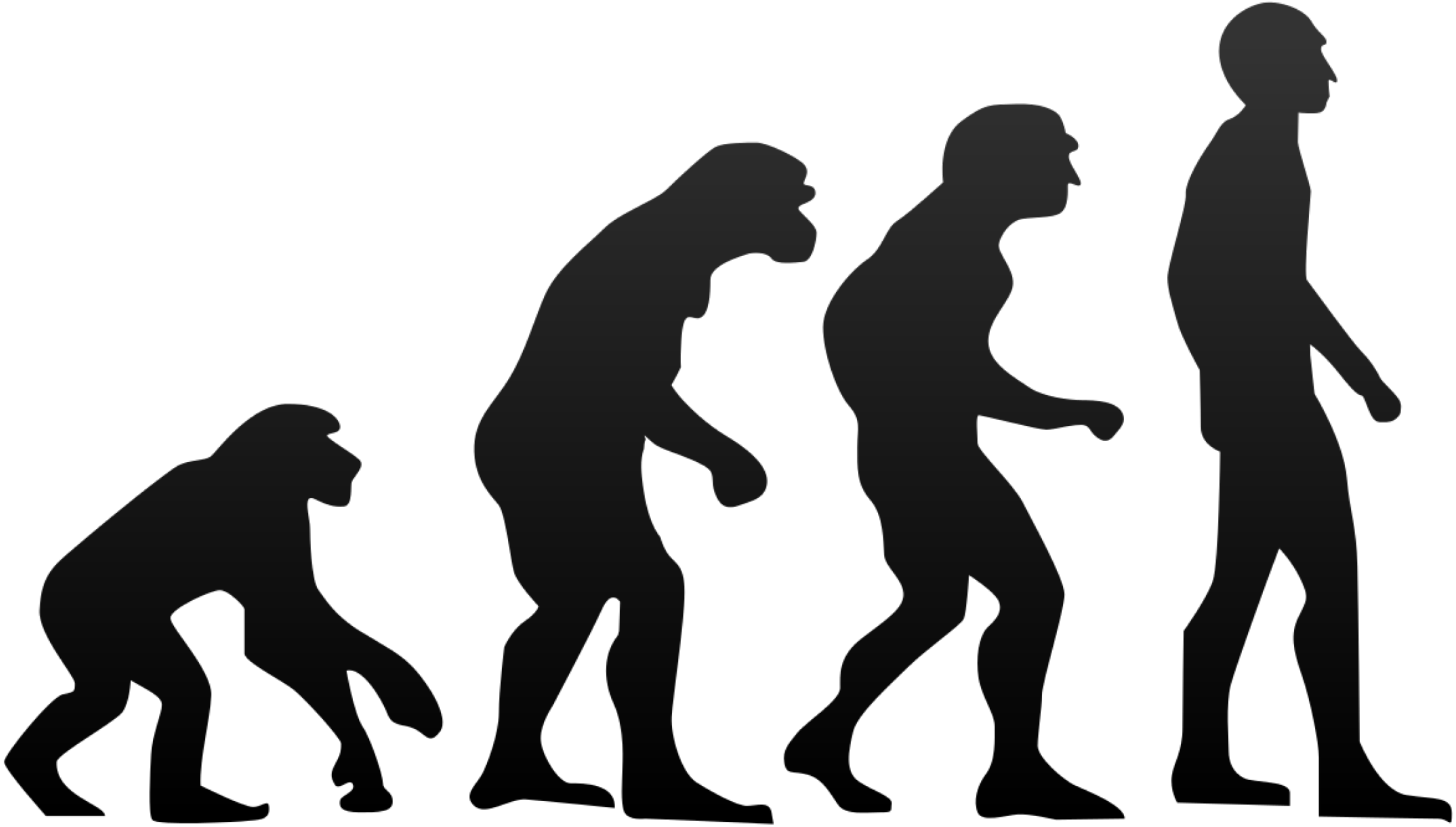
OpenStack

2010s

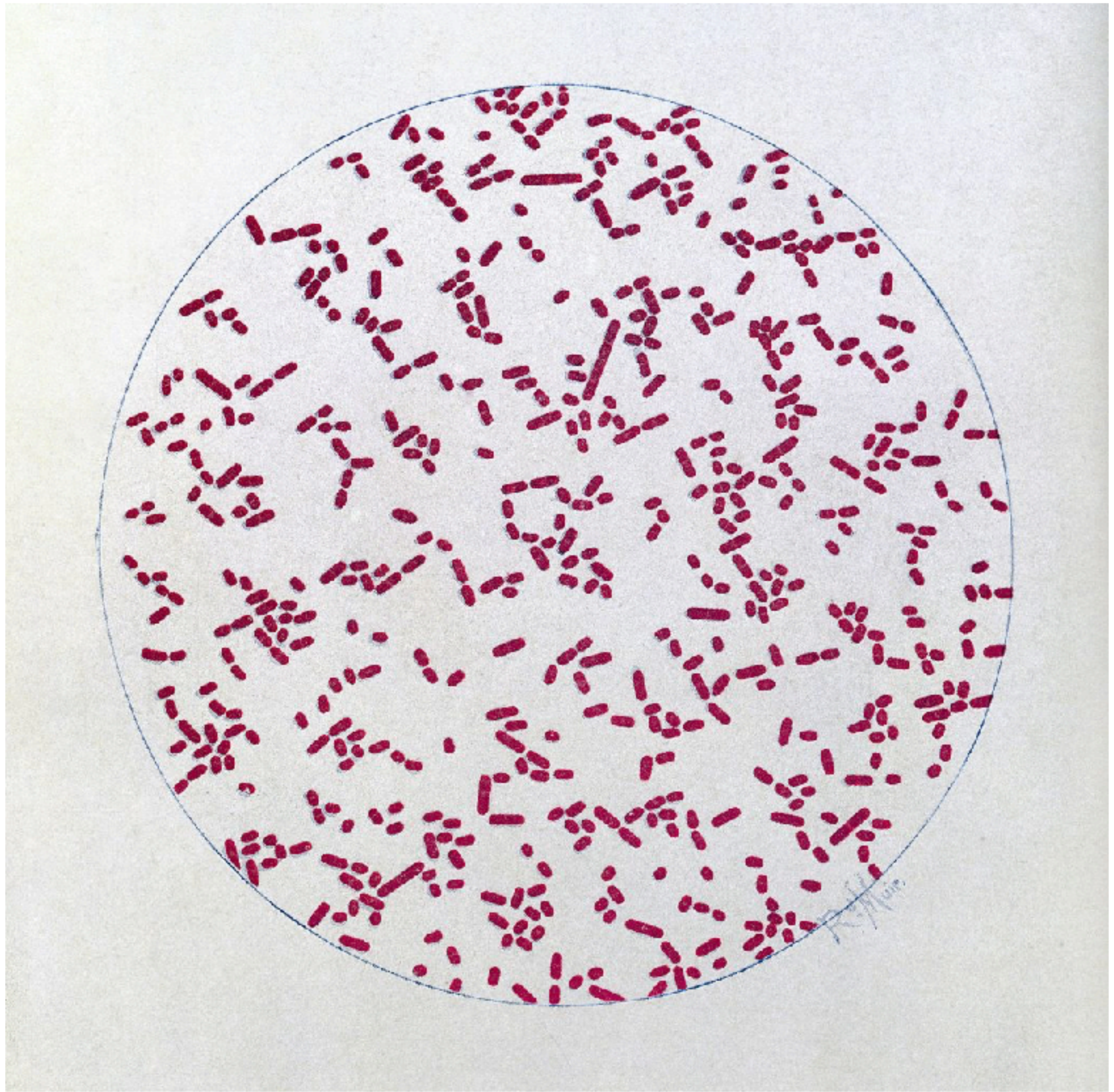
Kubernetes-based platforms

2018+

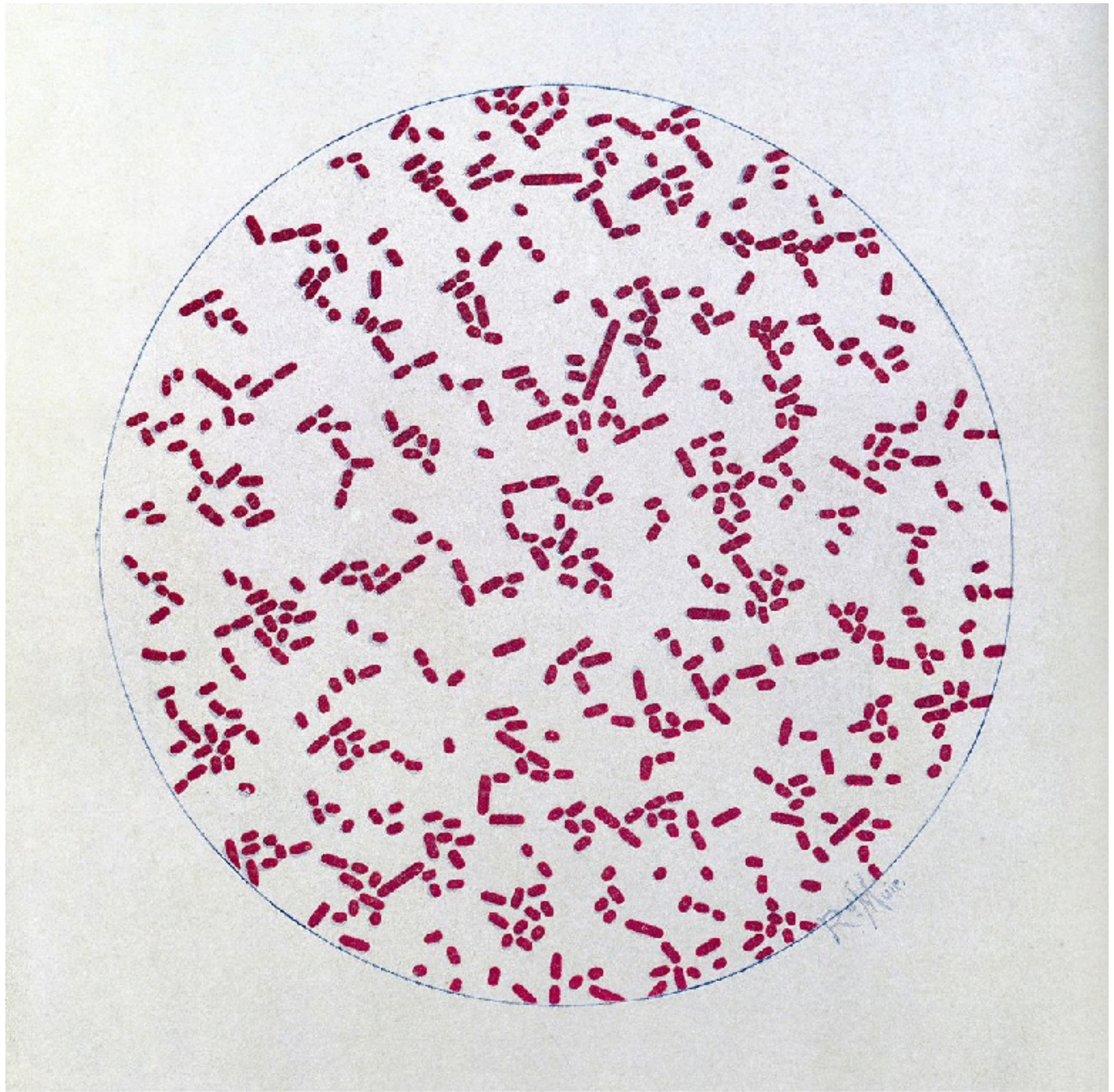
@samnewman



https://commons.wikimedia.org/wiki/File:Human_evolution.svg



https://commons.wikimedia.org/wiki/File:R._Muir,_Bacteriological_Atlas,_1927_Wellcome_L0030995.jpg https://commons.wikimedia.org/wiki/File:R._Muir,_Bacteriological_Atlas,_1927_Wellcome_L0030997.jpg



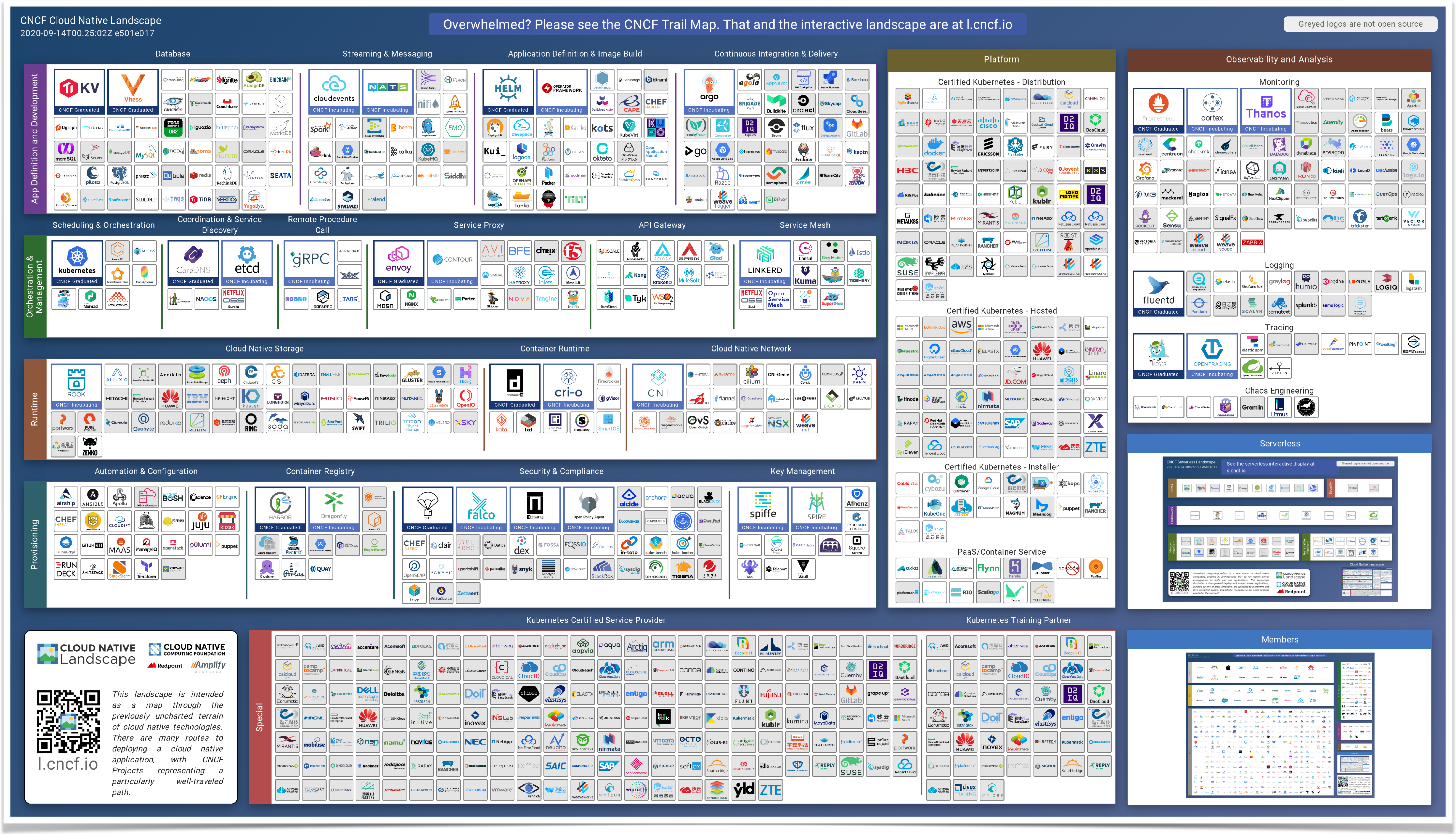
https://commons.wikimedia.org/wiki/File:R._Muir,_Bacteriological_Atlas,_1927_Wellcome_L0030995.jpg

kubernetes

https://www.flickr.com/photos/smithco/76749100/

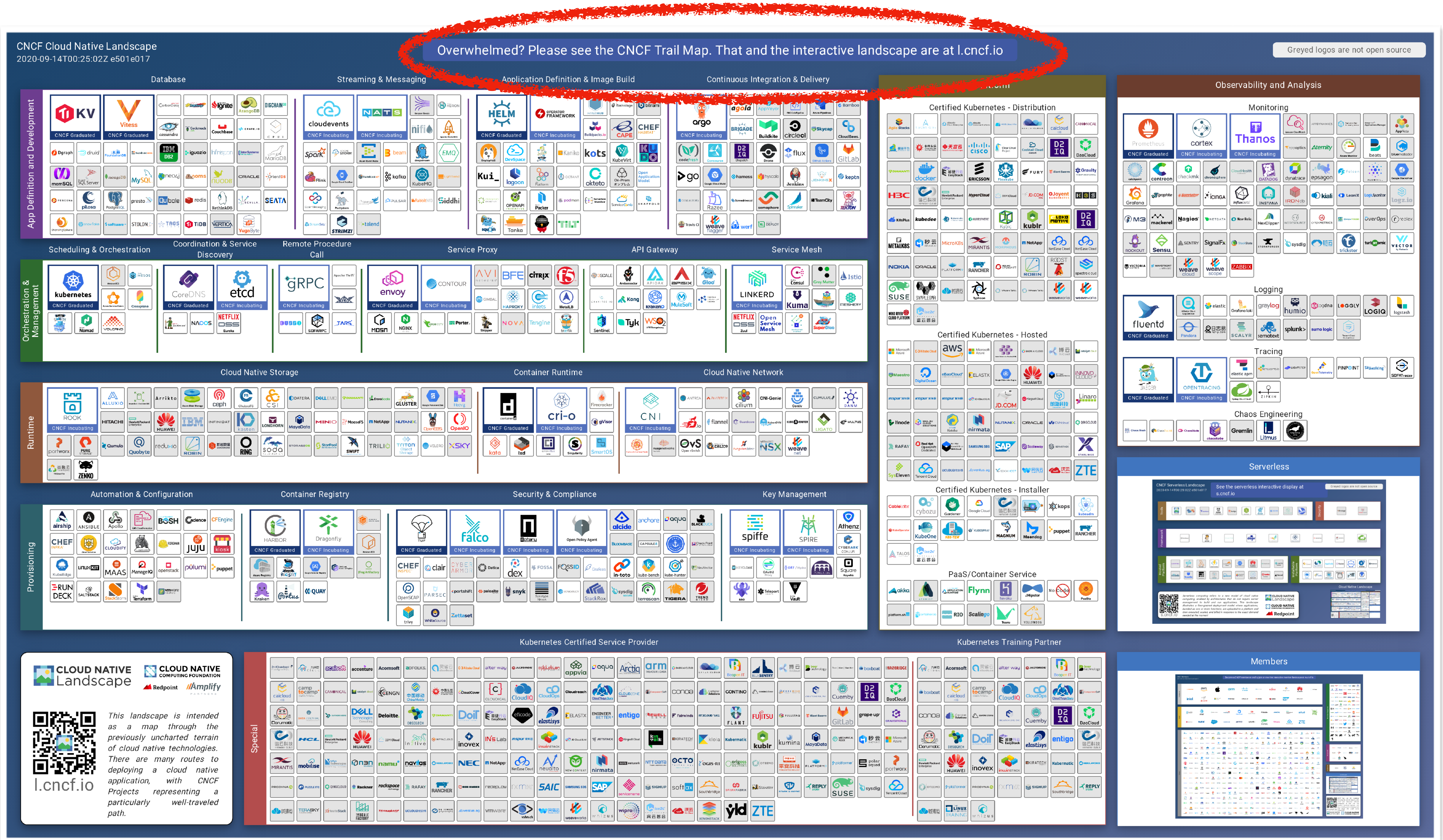
https://www.flickr.com/photos/felixdance/29013422242/

https://www.flickr.com/photos/jean_hort/9141538151/


CNCF Cloud Native Landscape
2020-09-14T00:25:02Z e501e017
Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io
Greyed logos are not open source
App Definition and Development
Database
Streaming & Messaging
Application Definition & Image Build
Continuous Integration & Delivery
Orchestration & Management
Scheduling & Orchestration
Coordination & Service Discovery
Remote Procedure Call
Service Proxy
API Gateway
Service Mesh
Runtime
Cloud Native Storage
Container Runtime
Cloud Native Network
Provisioning
Automation & Configuration
Container Registry
Security & Compliance
Key Management
Platform
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Certified Kubernetes - Installer
PaaS/Container Service
Observability and Analysis
Monitoring
Logging
Tracing
Chaos Engineering
Serverless
Kubernetes Certified Service Provider
Kubernetes Training Partner
Special
Members
CLOUD NATIVE Landscape
CLOUD NATIVE COMPUTING FOUNDATION
Redpoint
Amplify
This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.
l.cncf.io

https://landscape.cncf.io/images/landscape.png

@samnewman



CNCF Cloud Native Landscape
2020-09-14T00:25:02Z e501e017

Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io

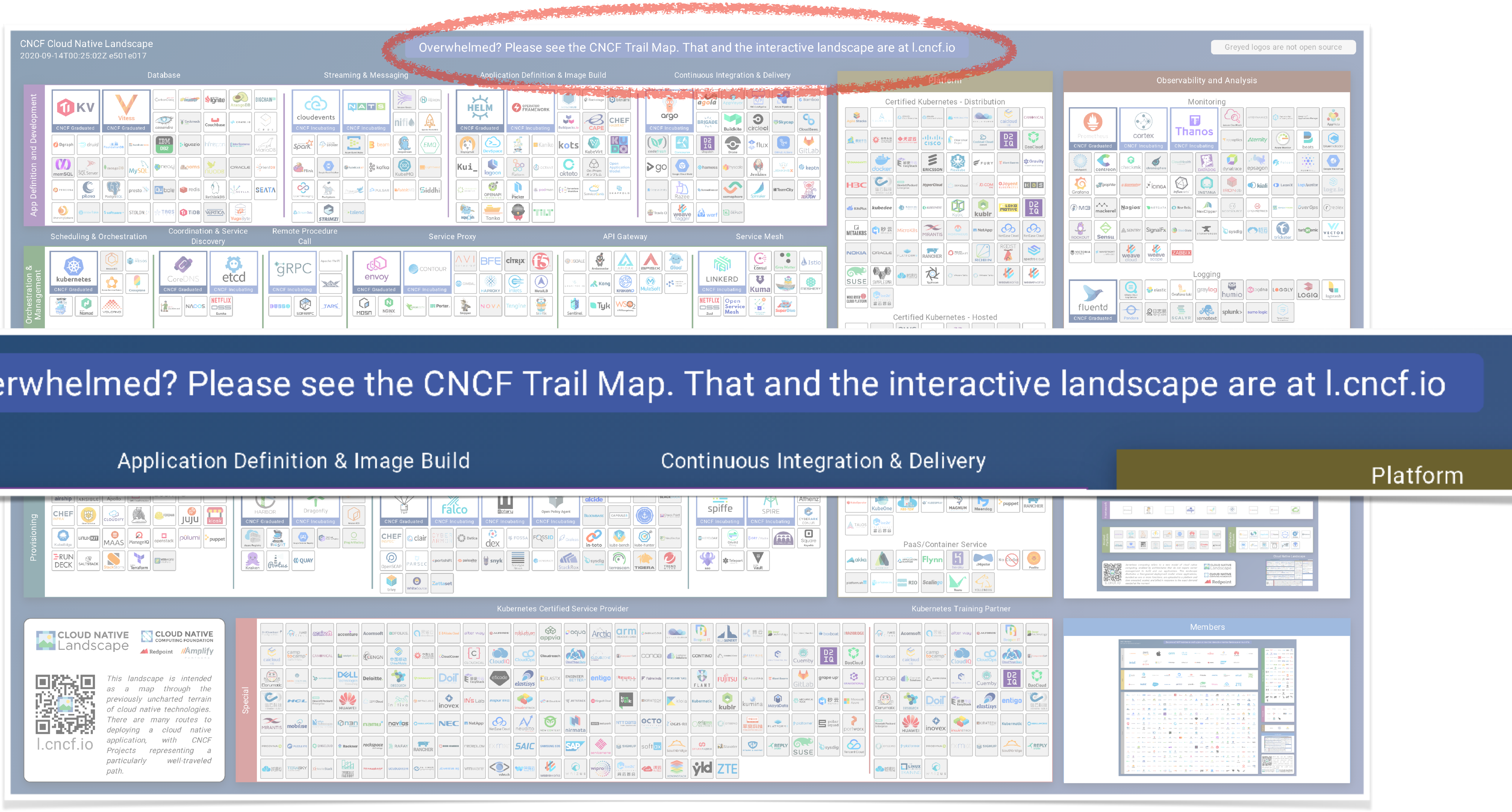
Greyed logos are not open source

App Definition and Development
Database
Streaming & Messaging
Application Definition & Image Build
Continuous Integration & Delivery
Orchestration & Management
Scheduling & Orchestration
Coordination & Service Discovery
Remote Procedure Call
Service Proxy
API Gateway
Service Mesh
Runtime
Cloud Native Storage
Container Runtime
Cloud Native Network
Provisioning
Automation & Configuration
Container Registry
Security & Compliance
Key Management
Platform
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Certified Kubernetes - Installer
PaaS/Container Service
Observability and Analysis
Monitoring
Logging
Tracing
Chaos Engineering
Serverless
Members
Special
Kubernetes Certified Service Provider
Kubernetes Training Partner
CNCF Graduated
CNCF Incubating
CLOUD NATIVE Landscape
CLOUD NATIVE COMPUTING FOUNDATION
Redpoint
Amplify
l.cncf.io

This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.

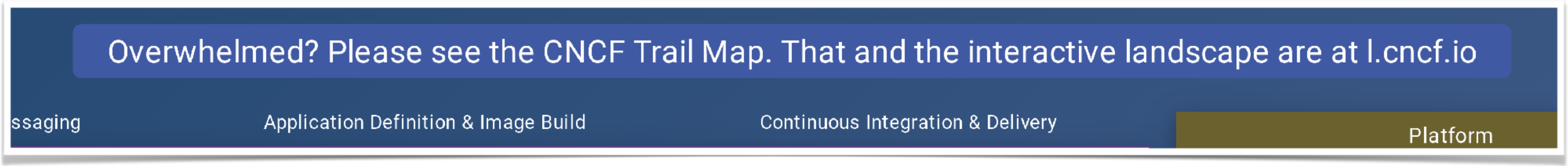
https://landscape.cncf.io/images/landscape.png

@samnewman


CNCF Cloud Native Landscape
2020-09-14T00:25:02Z e501e017
Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io
Greyed logos are not open source
App Definition and Development
Database
Streaming & Messaging
Application Definition & Image Build
Continuous Integration & Delivery
Platform
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Observability and Analysis
Monitoring
Logging
Orchestration & Management
Scheduling & Orchestration
Coordination & Service Discovery
Remote Procedure Call
Service Proxy
API Gateway
Service Mesh
Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io
aging
Application Definition & Image Build
Continuous Integration & Delivery
Platform
Provisioning
PaaS/Container Service
Kubernetes Certified Service Provider
Kubernetes Training Partner
Special
Members
CLOUD NATIVE Landscape
CLOUD NATIVE COMPUTING FOUNDATION
Redpoint
Amplify
This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.
l.cncf.io

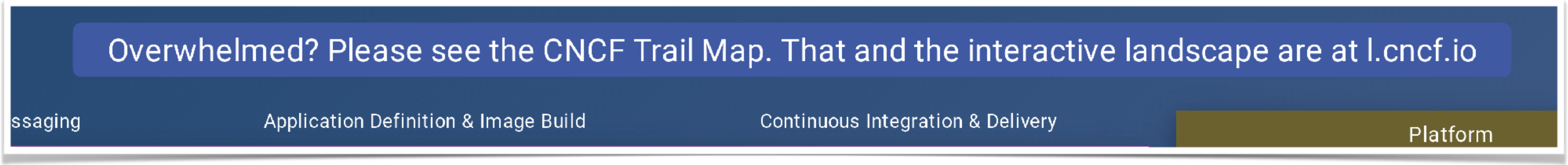
https://landscape.cncf.io/images/landscape.png

@samnewman



Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io

ssaging
Application Definition & Image Build
Continuous Integration & Delivery
Platform

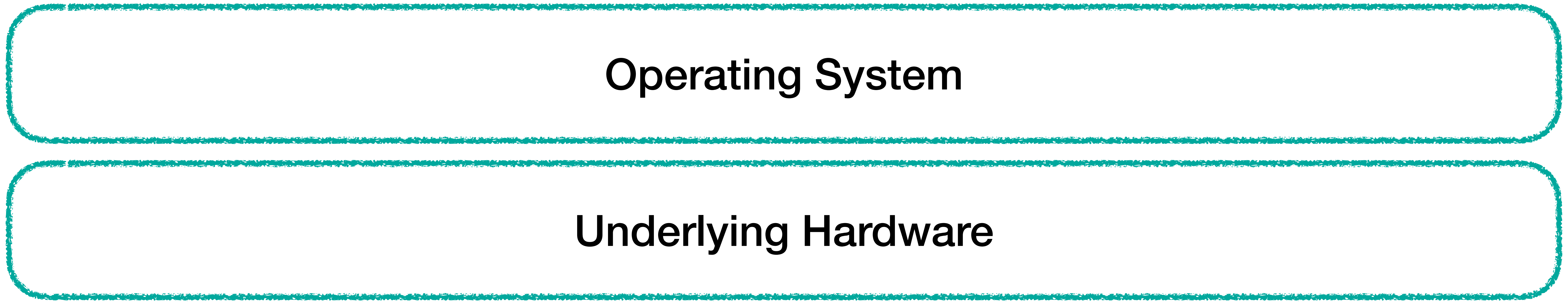


Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io

ssaging
Application Definition & Image Build
Continuous Integration & Delivery
Platform

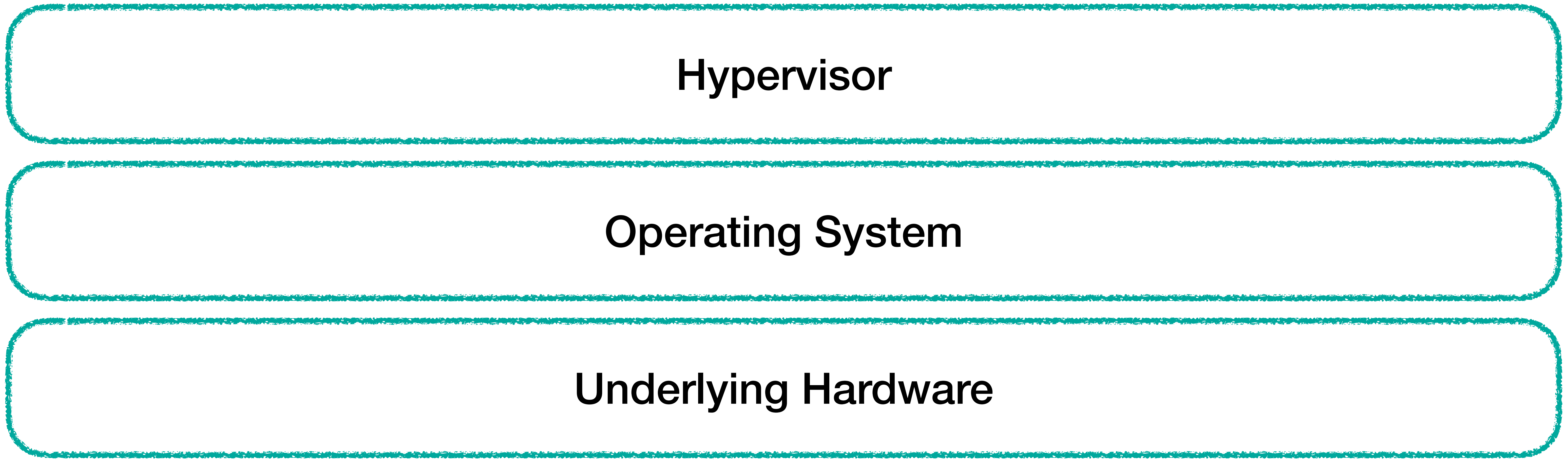
Overwhelmed? Please contact your closest expensive consultants. Or your therapist.

Underlying Hardware



Operating System

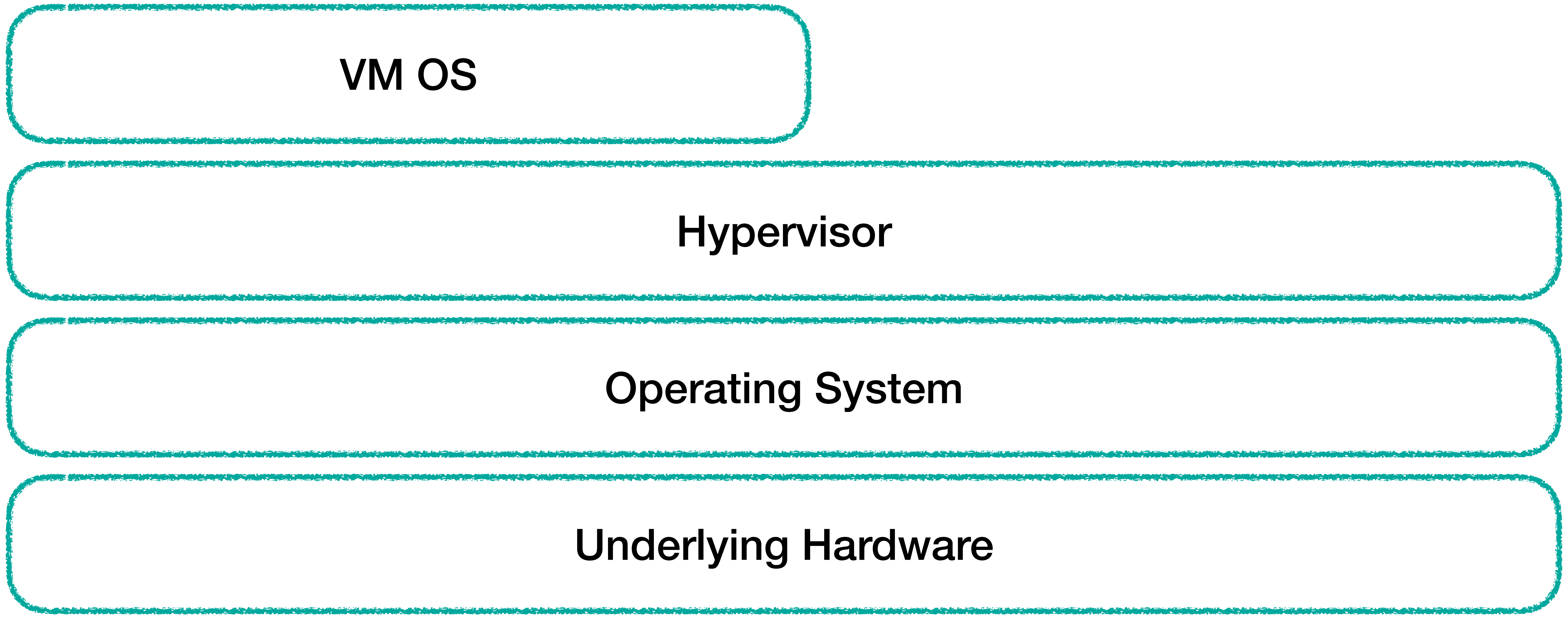
Underlying Hardware



Hypervisor

Operating System

Underlying Hardware

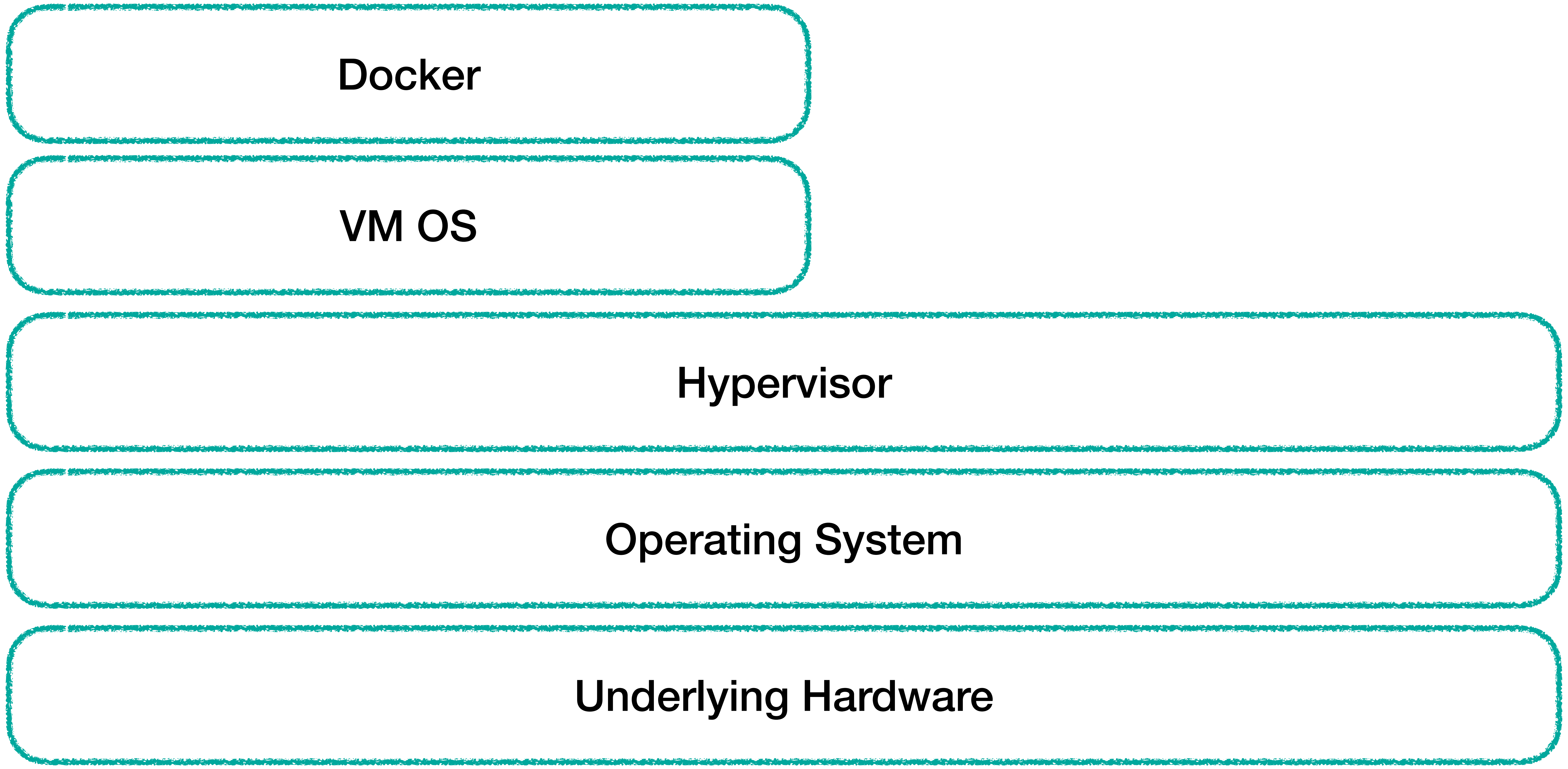


VM OS

Hypervisor

Operating System

Underlying Hardware



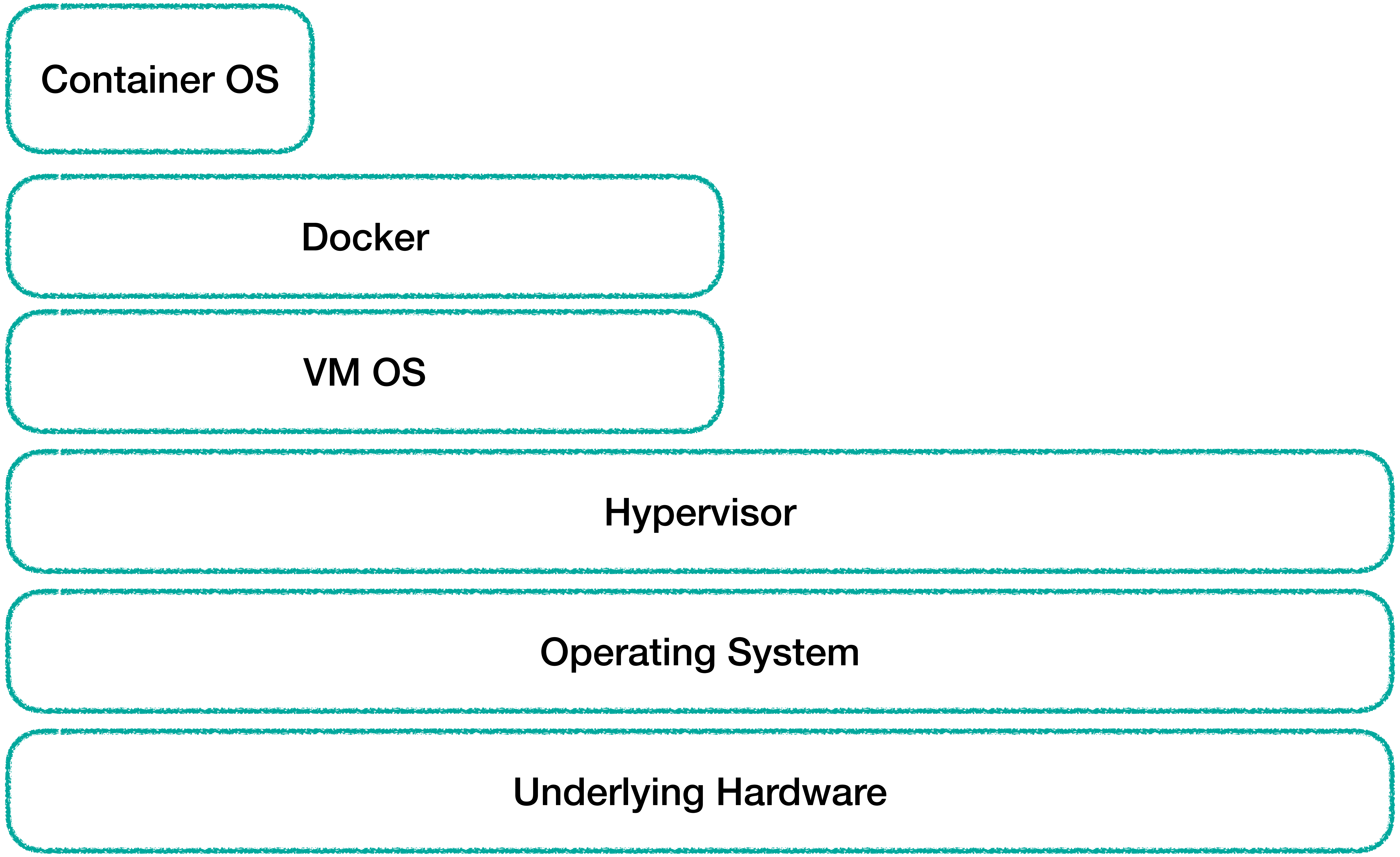
Docker

VM OS

Hypervisor

Operating System

Underlying Hardware



Container OS

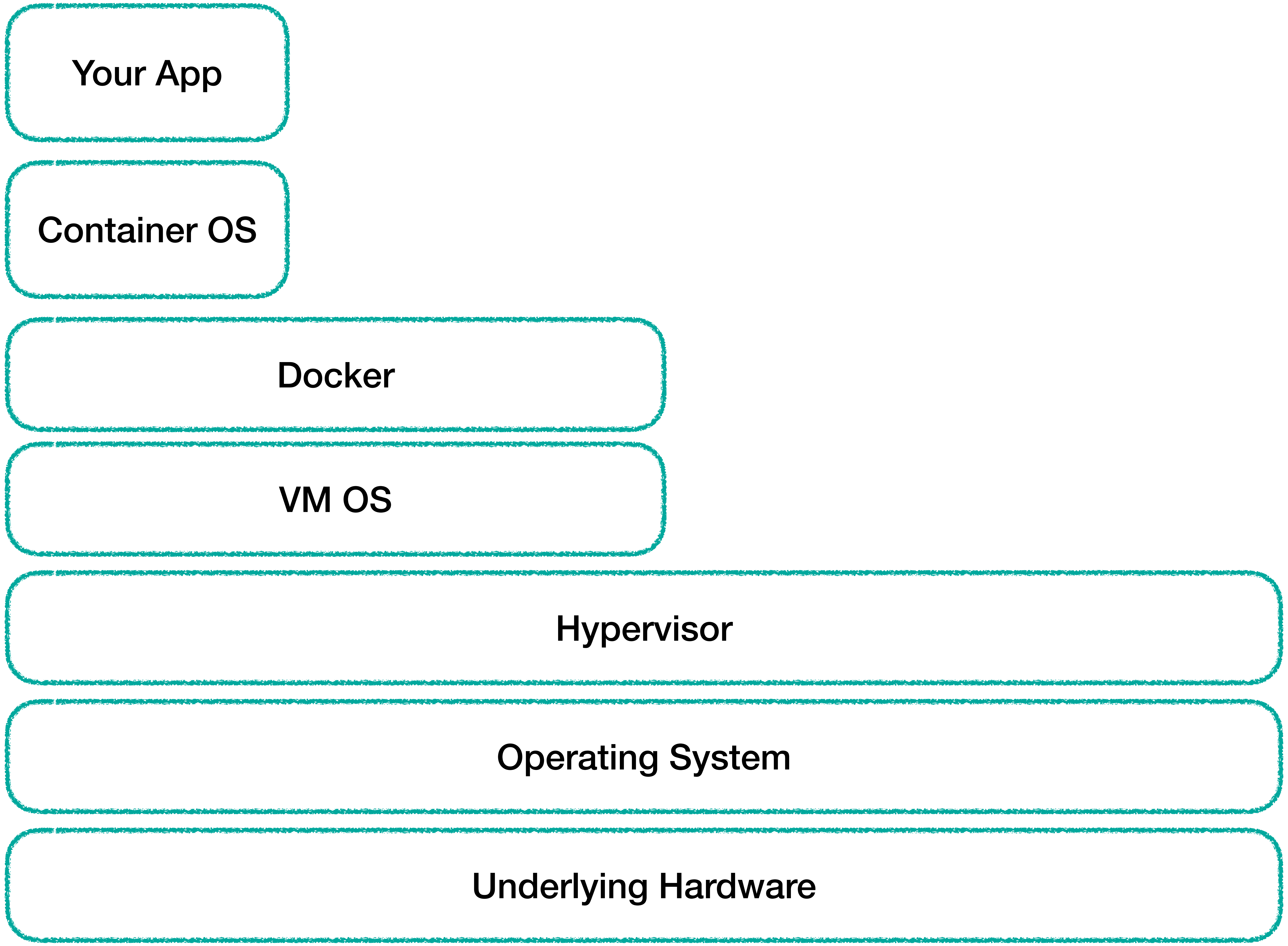
Docker

VM OS

Hypervisor

Operating System

Underlying Hardware



Your App

Container OS

Docker

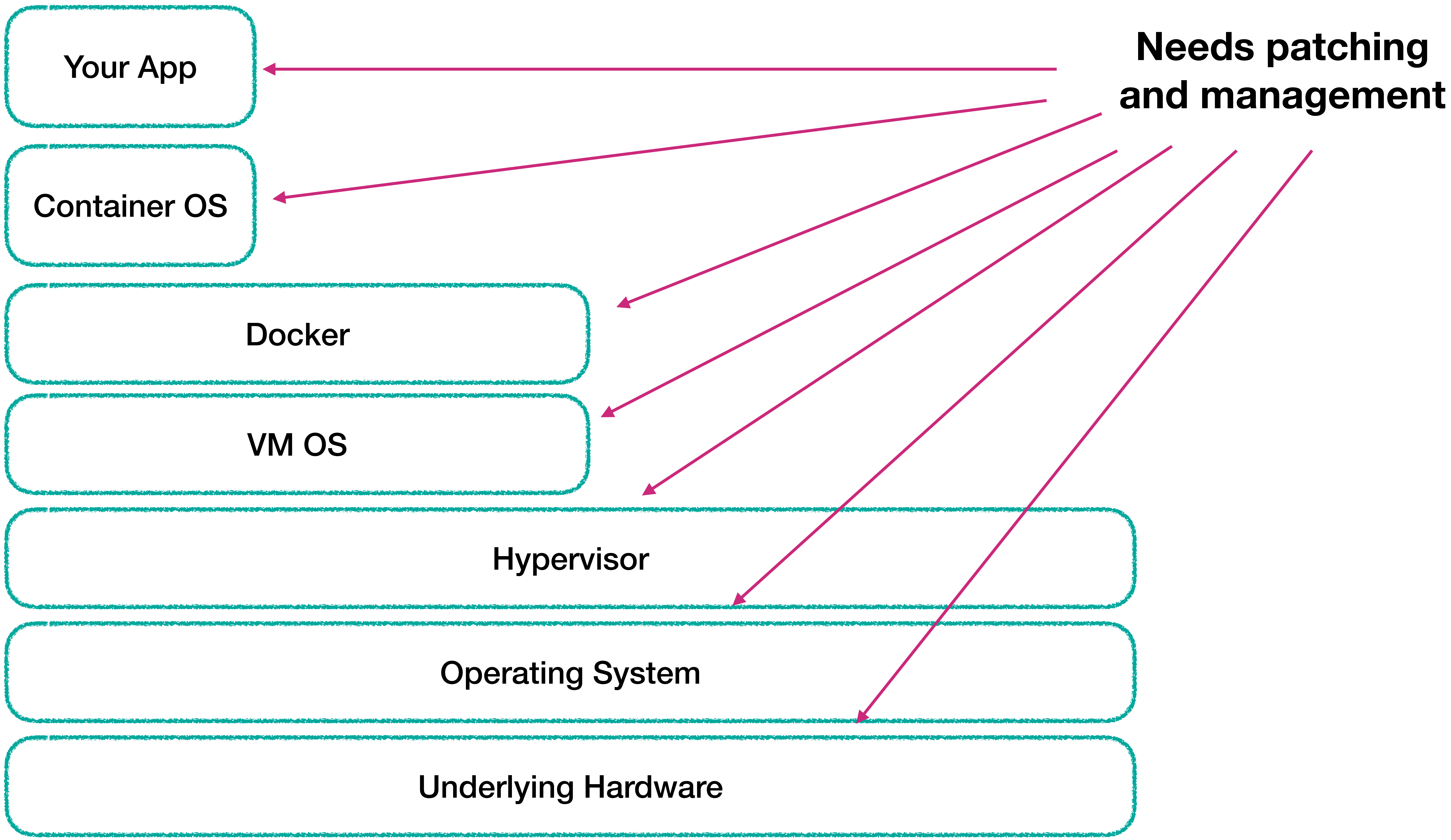
VM OS

Hypervisor

Operating System

Underlying Hardware

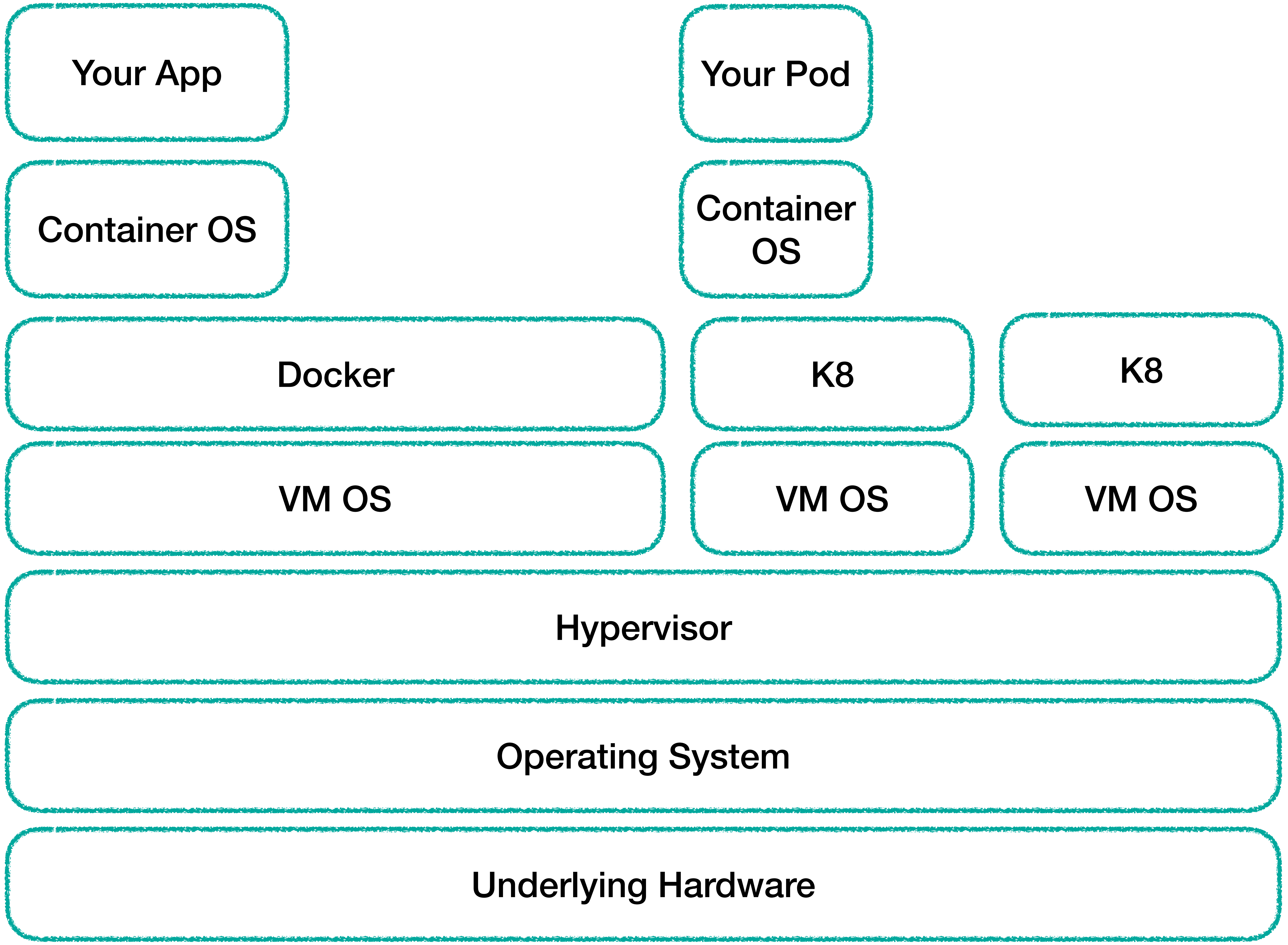
@samnewman



Needs patching and management

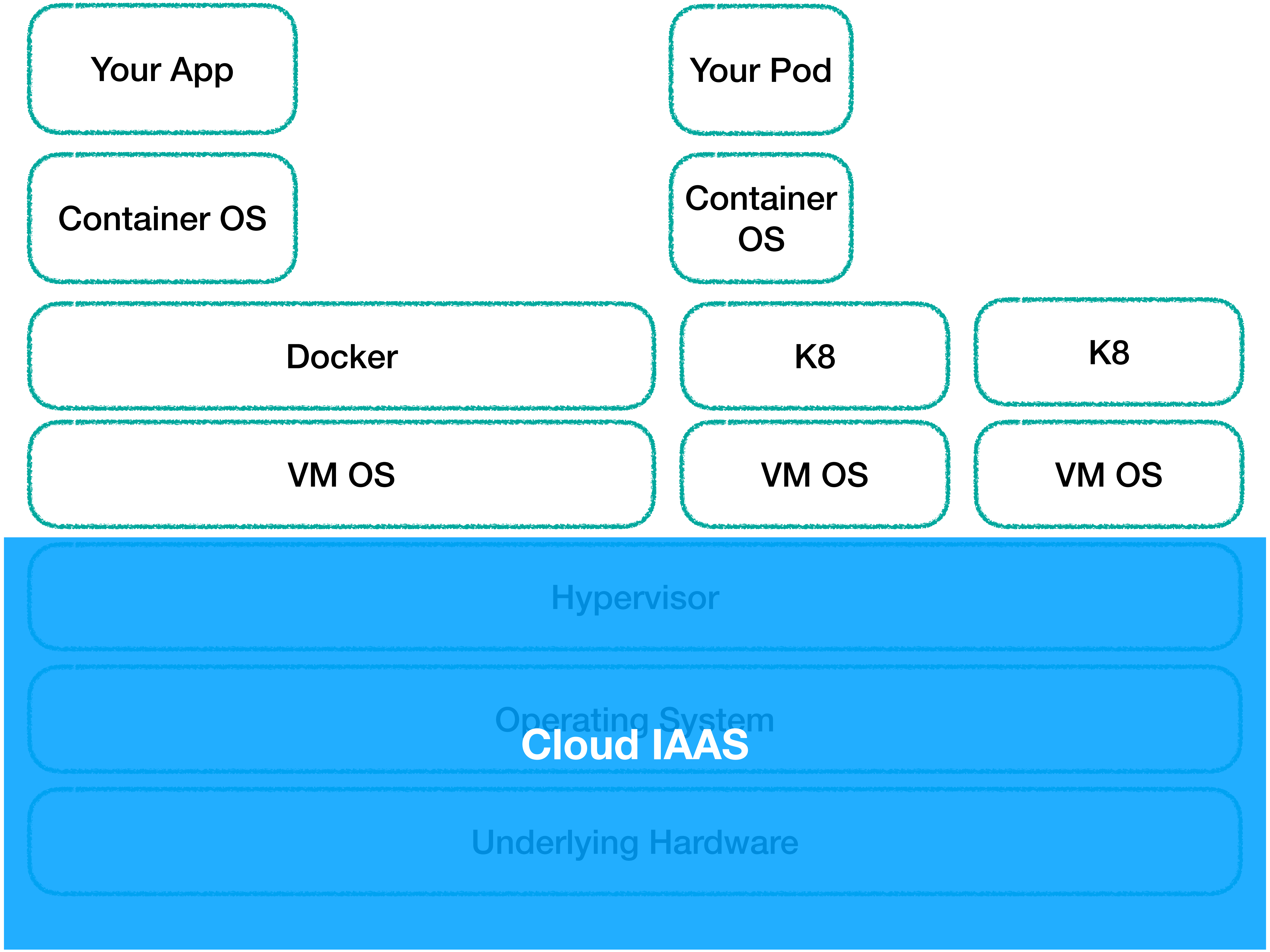
@samnewman

BETTER ON THE CLOUD?

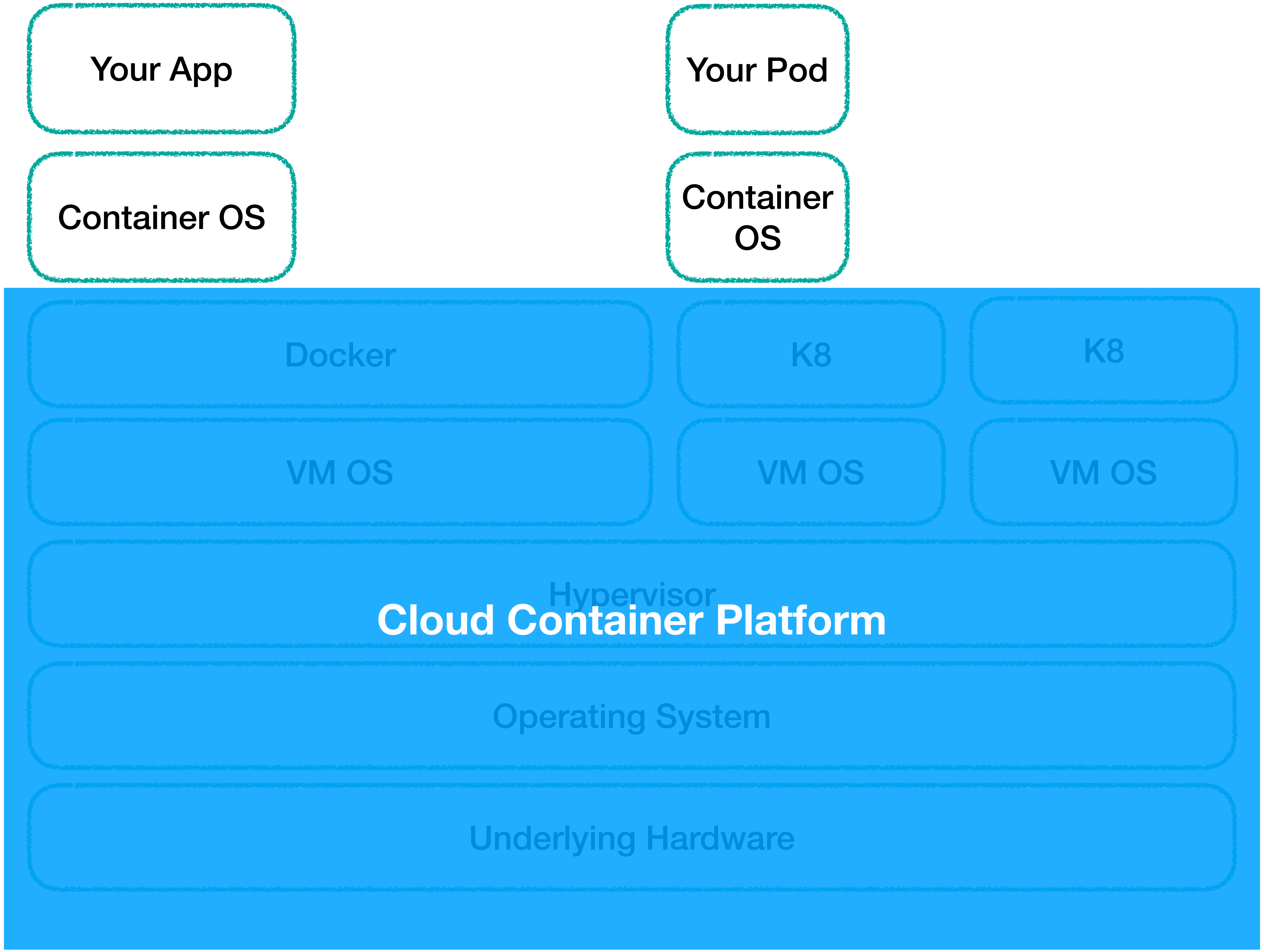


@samnewman

BETTER ON THE CLOUD?



BETTER ON THE CLOUD?



@samnewman

#serverless

A service offering that abstracts away the notion of underlying machines

KEY PROPERTIES OF SERVERLESS

KEY PROPERTIES OF SERVERLESS

No server management

KEY PROPERTIES OF SERVERLESS

No server management

Autoscale based on use

KEY PROPERTIES OF SERVERLESS

No server management

Autoscale based on use

Implicit high availability

KEY PROPERTIES OF SERVERLESS

No server management

Autoscale based on use

Implicit high availability

Pay as you go

EXAMPLE SERVERLESS PRODUCTS

Function As A Service (FAAS)

@samnewman

EXAMPLE SERVERLESS PRODUCTS

Function As A Service (FAAS)

Messaging Solutions

EXAMPLE SERVERLESS PRODUCTS

Function As A Service (FAAS)

Messaging Solutions

Databases

EXAMPLE SERVERLESS PRODUCTS

Function As A Service (FAAS)

Messaging Solutions

Databases

Storage

EXAMPLE SERVERLESS PRODUCTS

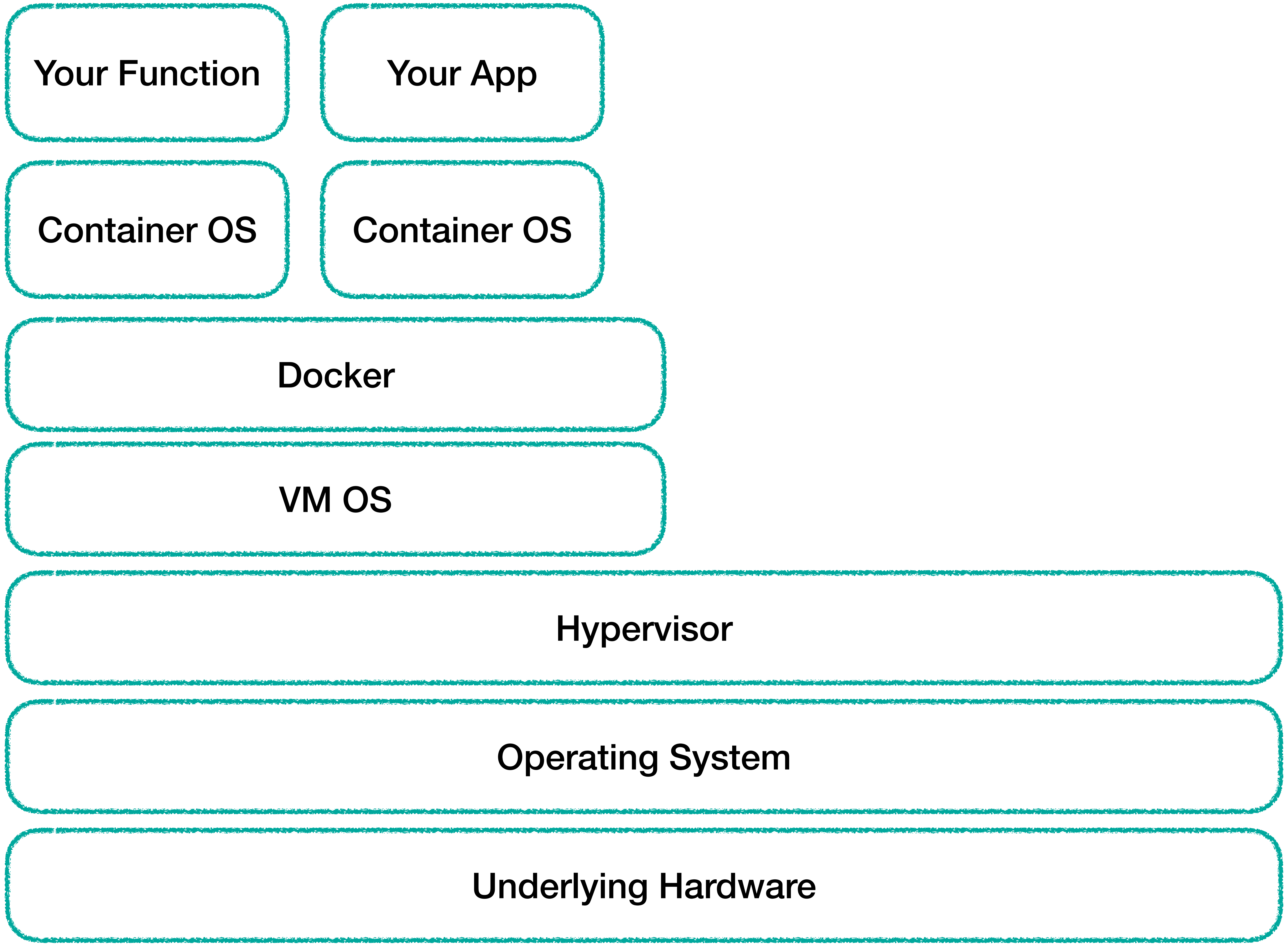
Function As A Service (FAAS)

Messaging Solutions

Databases

Storage

BETTER WITH FAAS?



Your Function

Your App

Container OS

Container OS

Docker

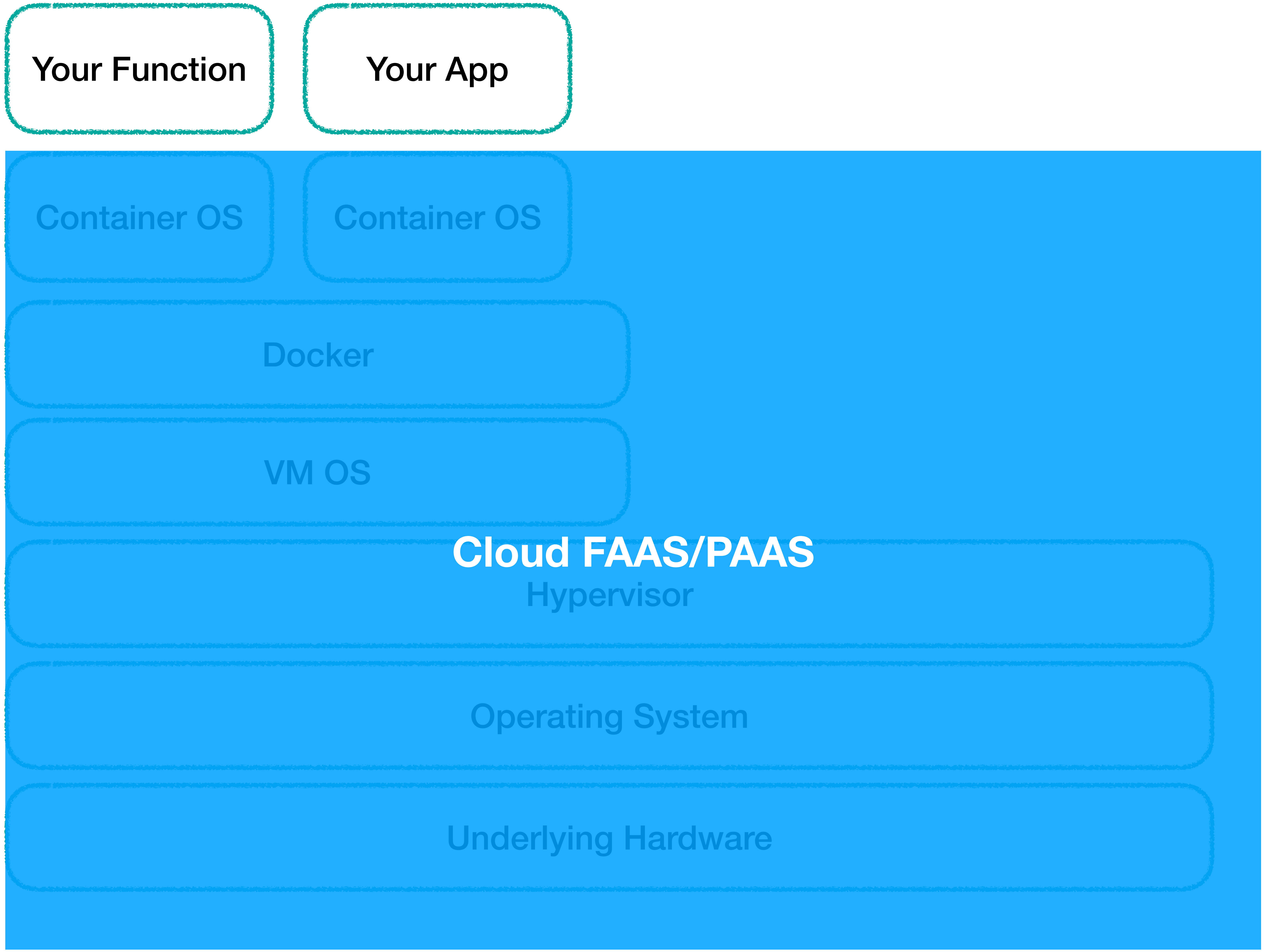
VM OS

Hypervisor

Operating System

Underlying Hardware

BETTER WITH FAAS?



Your Function

Your App

Container OS

Container OS

Docker

VM OS

Cloud FAAS/PAAS

Hypervisor

Operating System

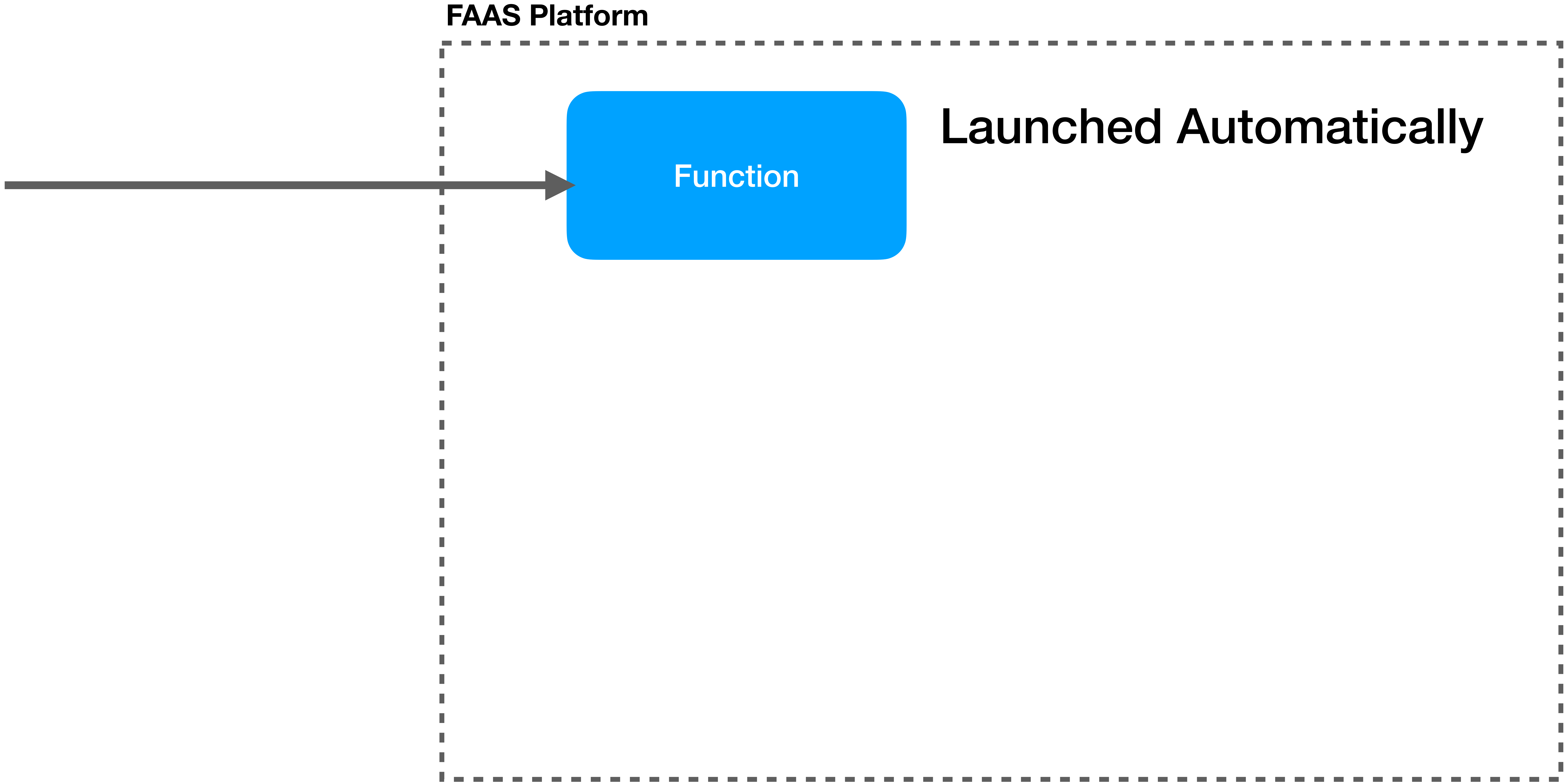
Underlying Hardware

FAAS In A Nutshell

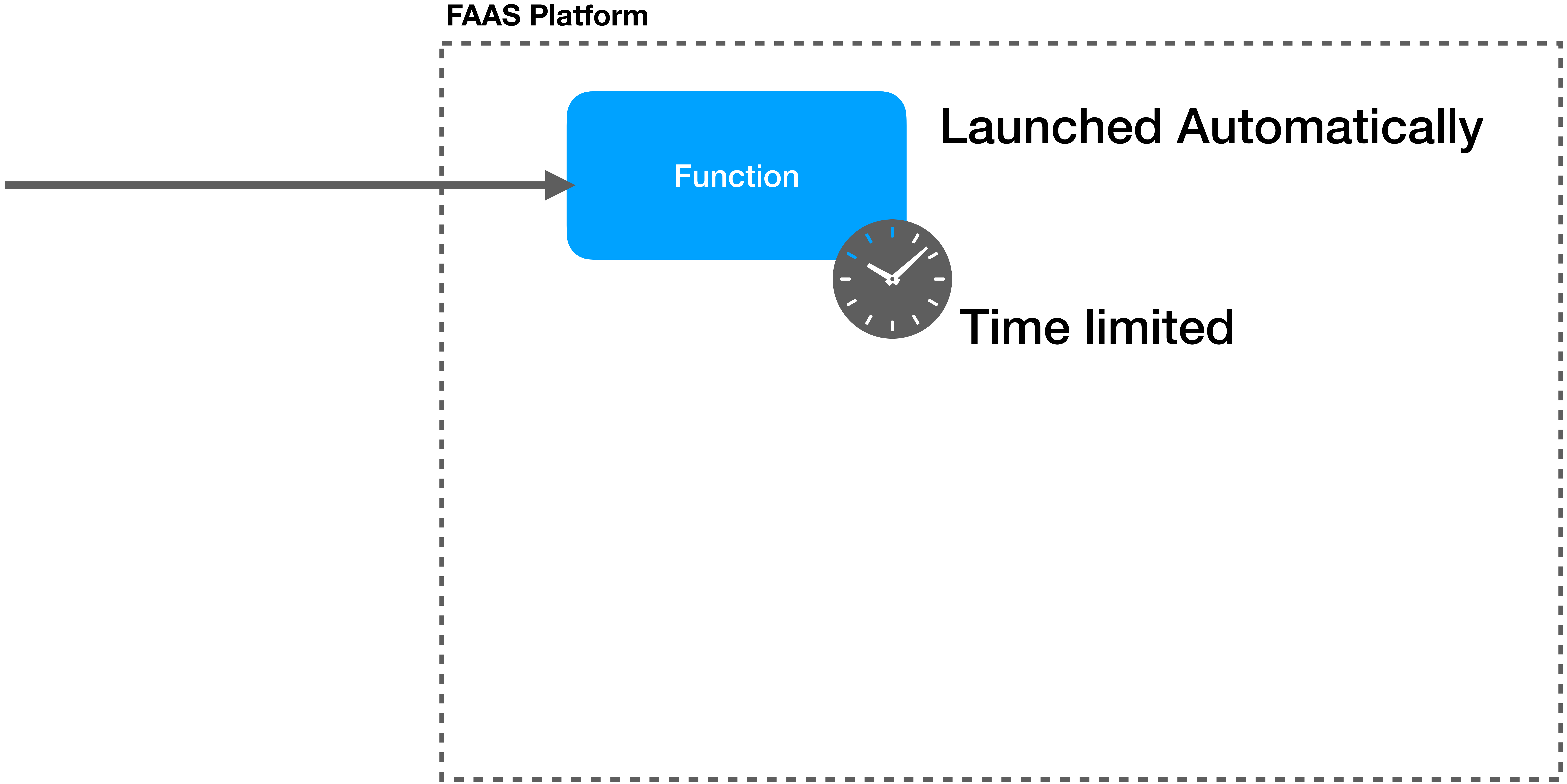
FAAS IN A NUTSHELL

FAAS Platform

FAAS IN A NUTSHELL



FAAS IN A NUTSHELL



FAAS Platform

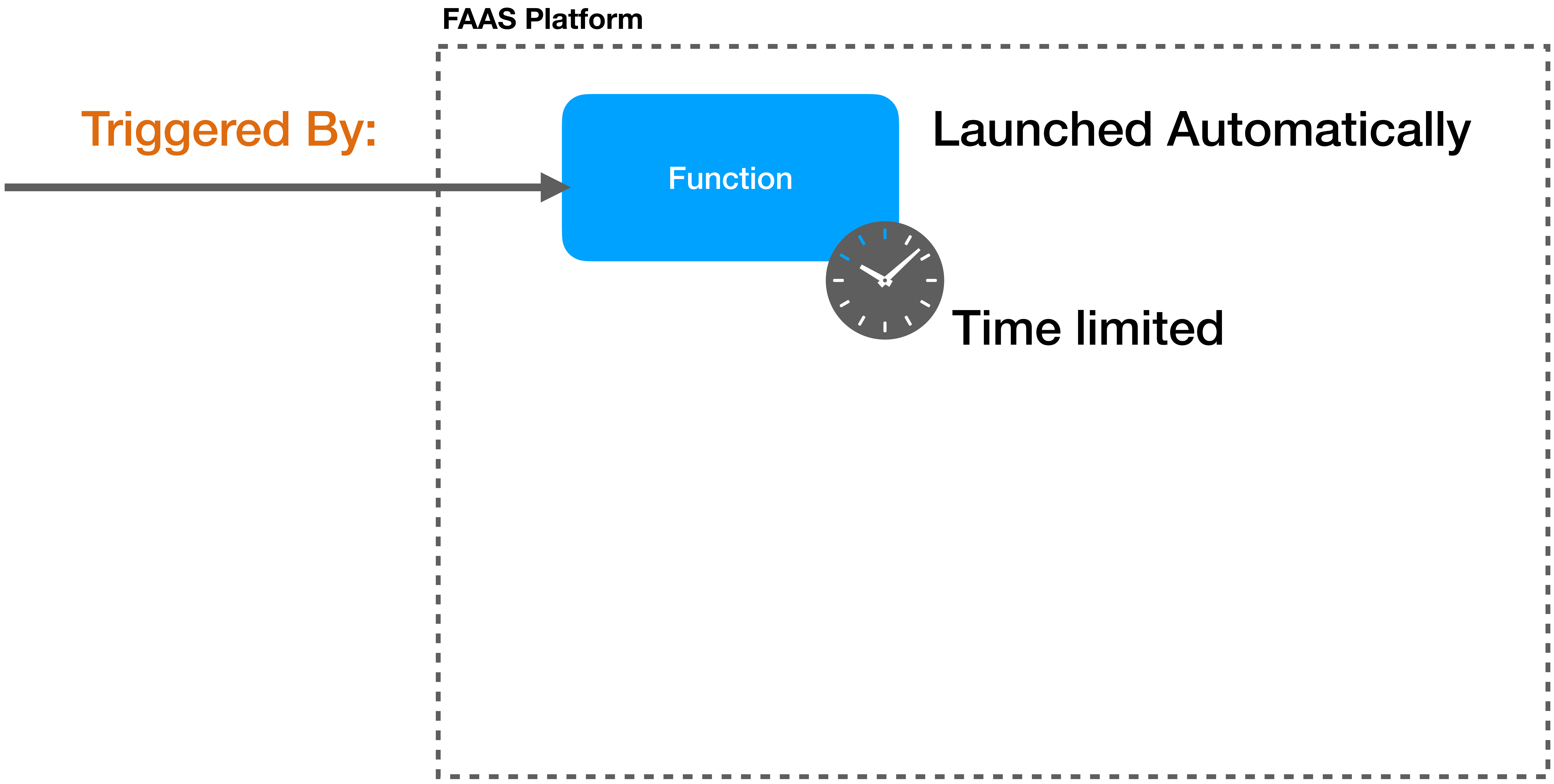
Function

Launched Automatically

Time limited

@samnewman

FAAS IN A NUTSHELL



FAAS Platform

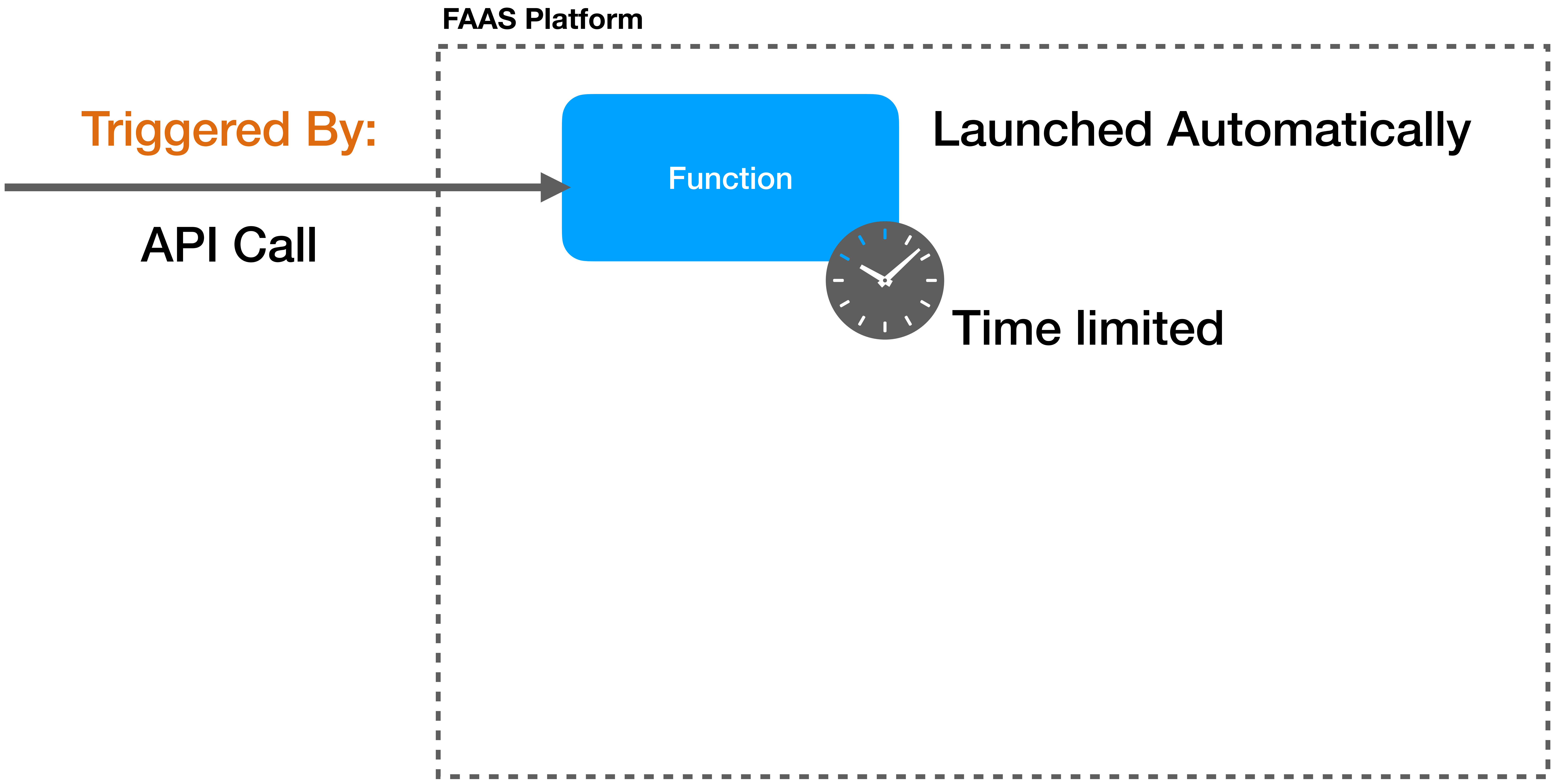
Triggered By:

Function

Launched Automatically

Time limited

FAAS IN A NUTSHELL



FAAS Platform

Triggered By:

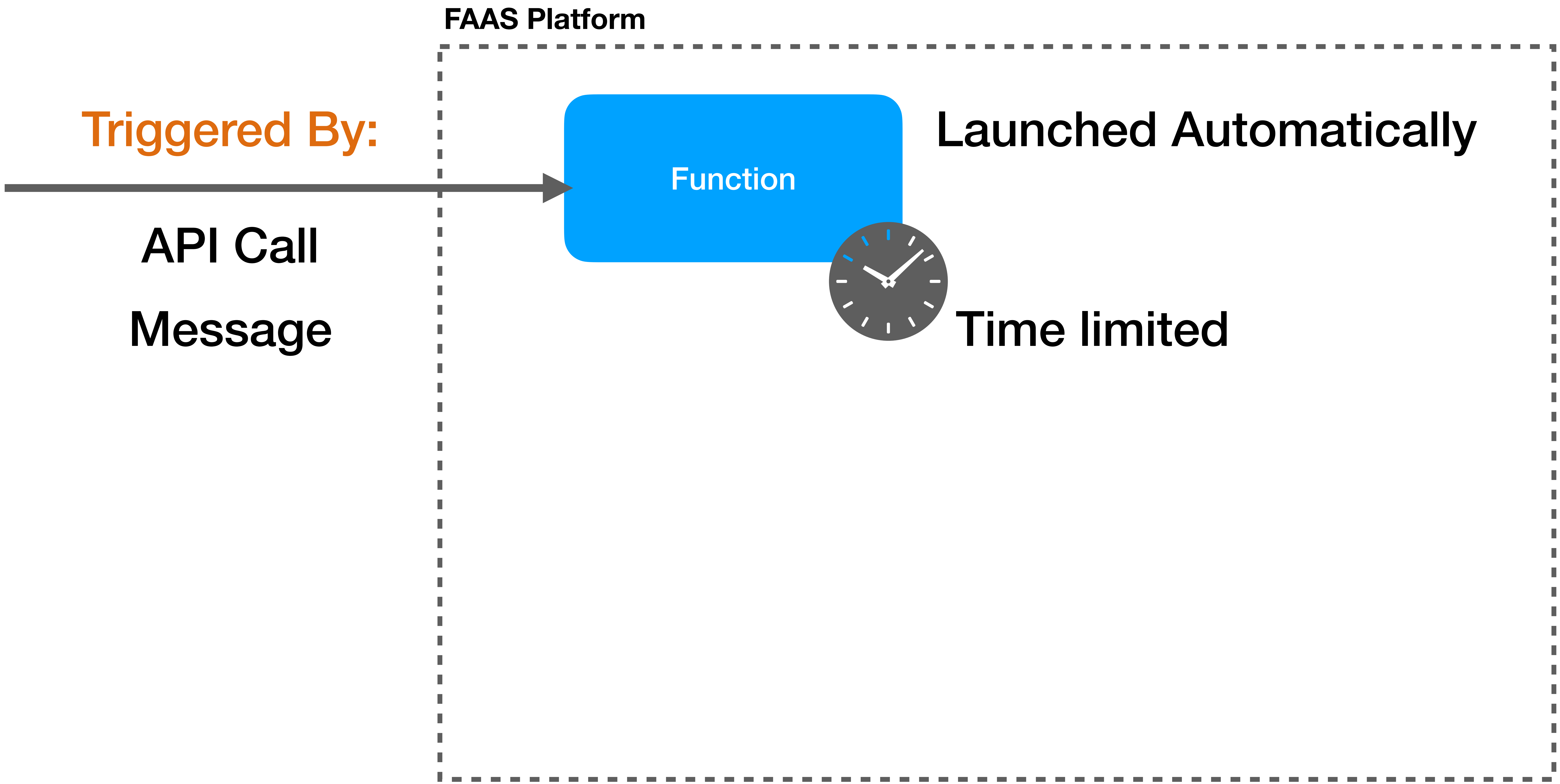
API Call

Function

Launched Automatically

Time limited

FAAS IN A NUTSHELL



FAAS Platform

Triggered By:

API Call

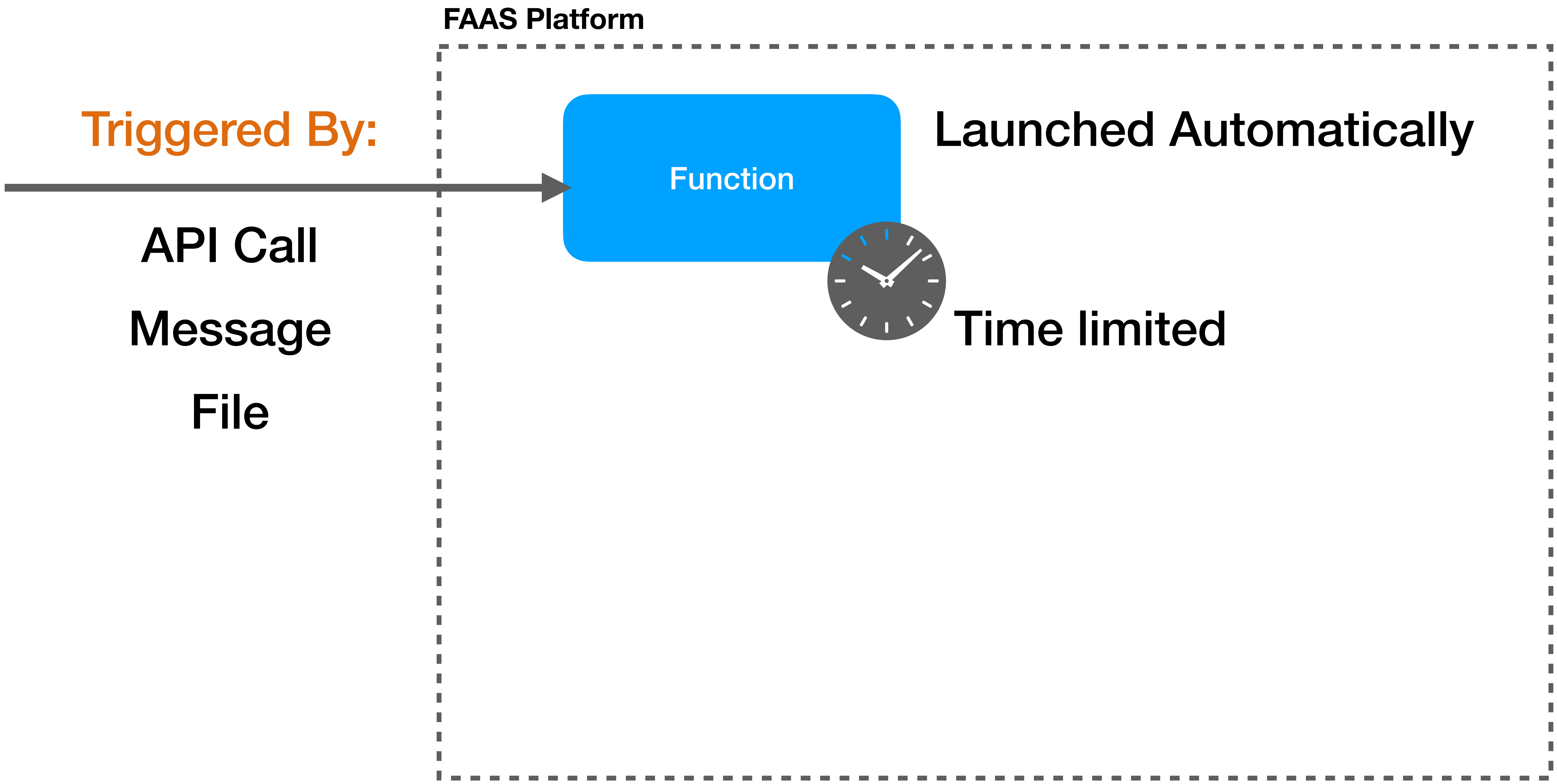
Message

Function

Launched Automatically

Time limited

FAAS IN A NUTSHELL



FAAS Platform

Triggered By:

API Call

Message

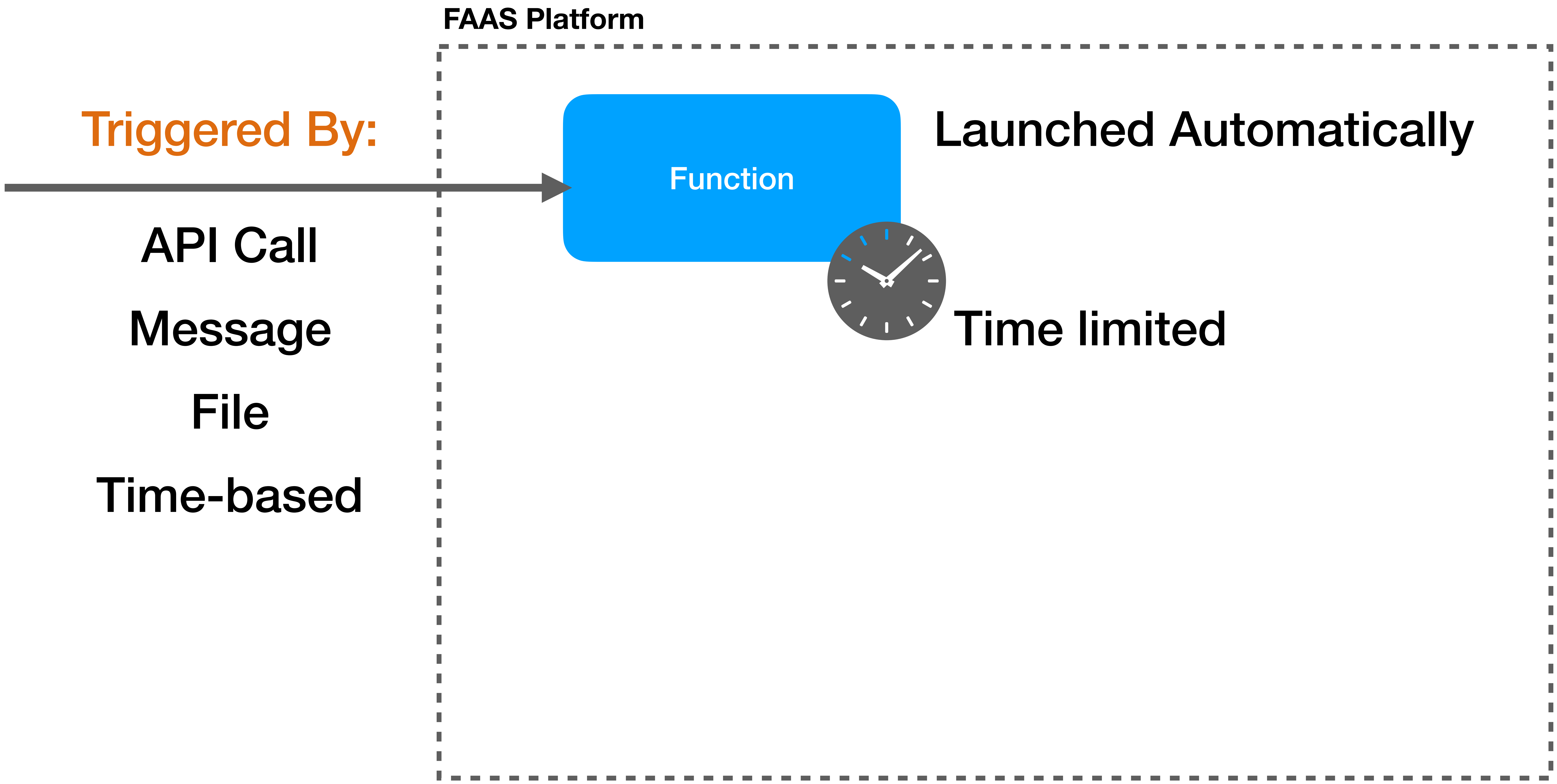
File

Function

Launched Automatically

Time limited

FAAS IN A NUTSHELL



Triggered By:

API Call

Message

File

Time-based

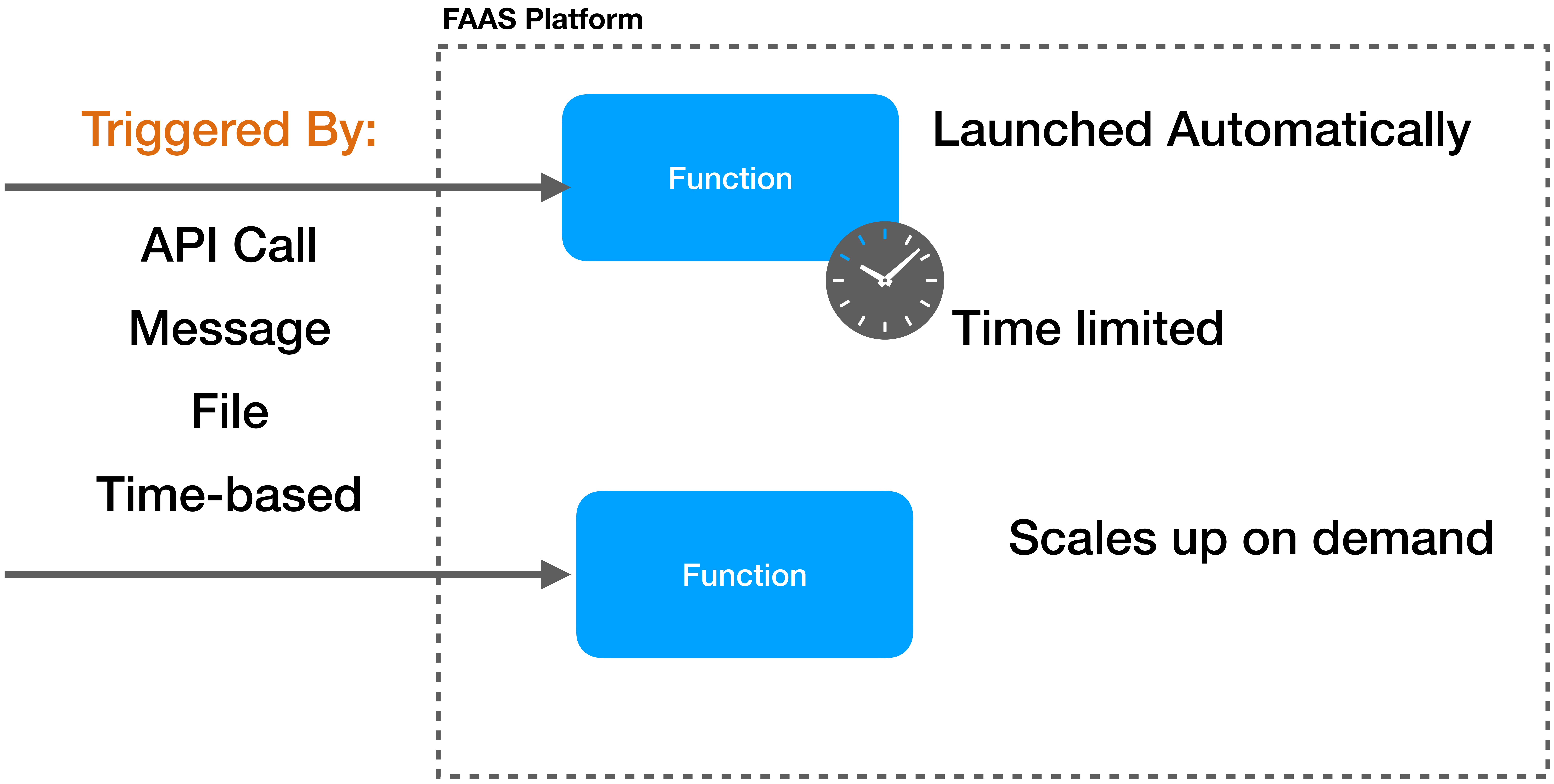
FAAS Platform

Function

Launched Automatically

Time limited

FAAS IN A NUTSHELL



FAAS Platform

Triggered By:

API Call

Message

File

Time-based

Function

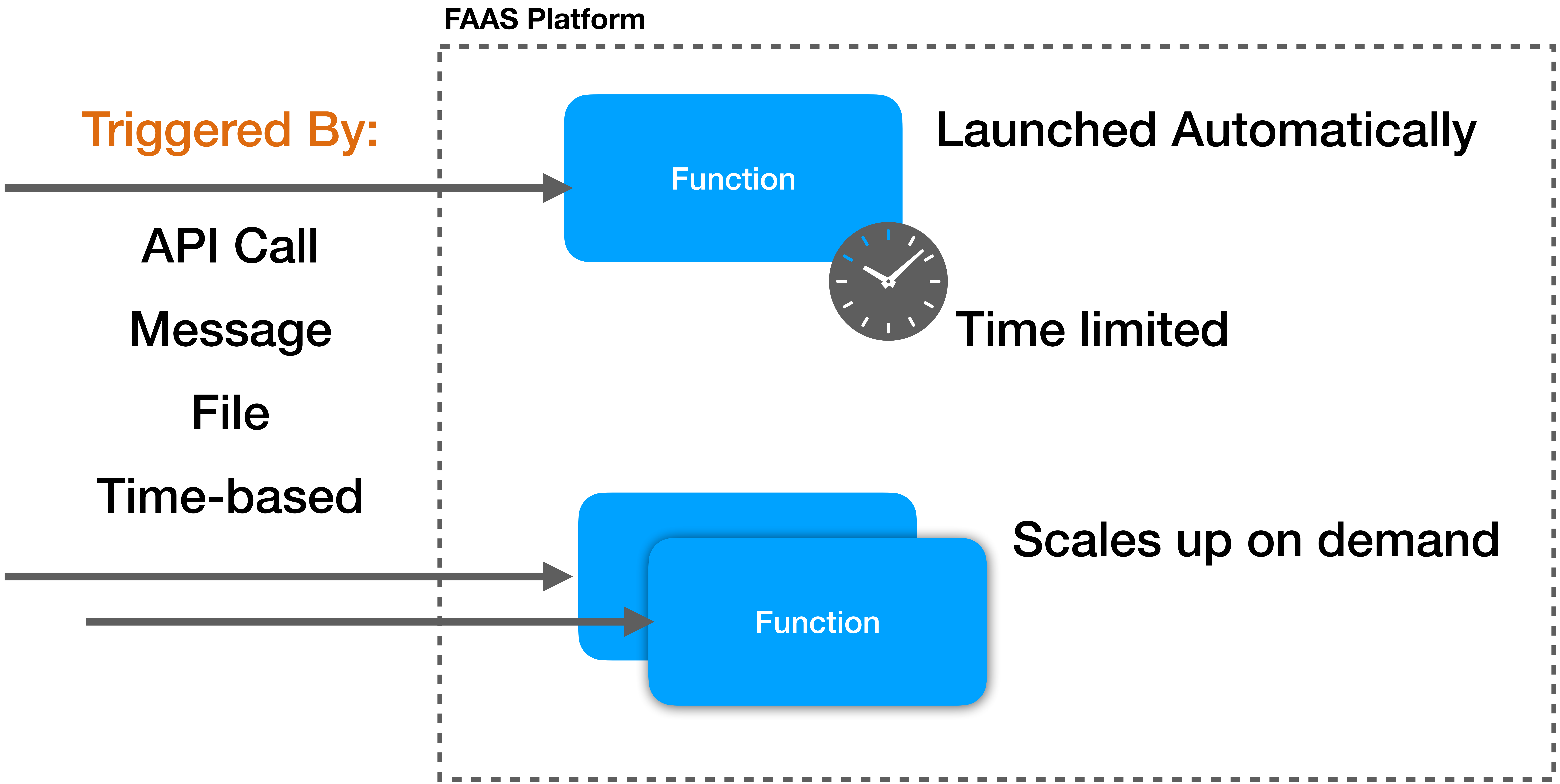
Launched Automatically

Time limited

Function

Scales up on demand

FAAS IN A NUTSHELL



FAAS Platform

Triggered By:

API Call

Message

File

Time-based

Function

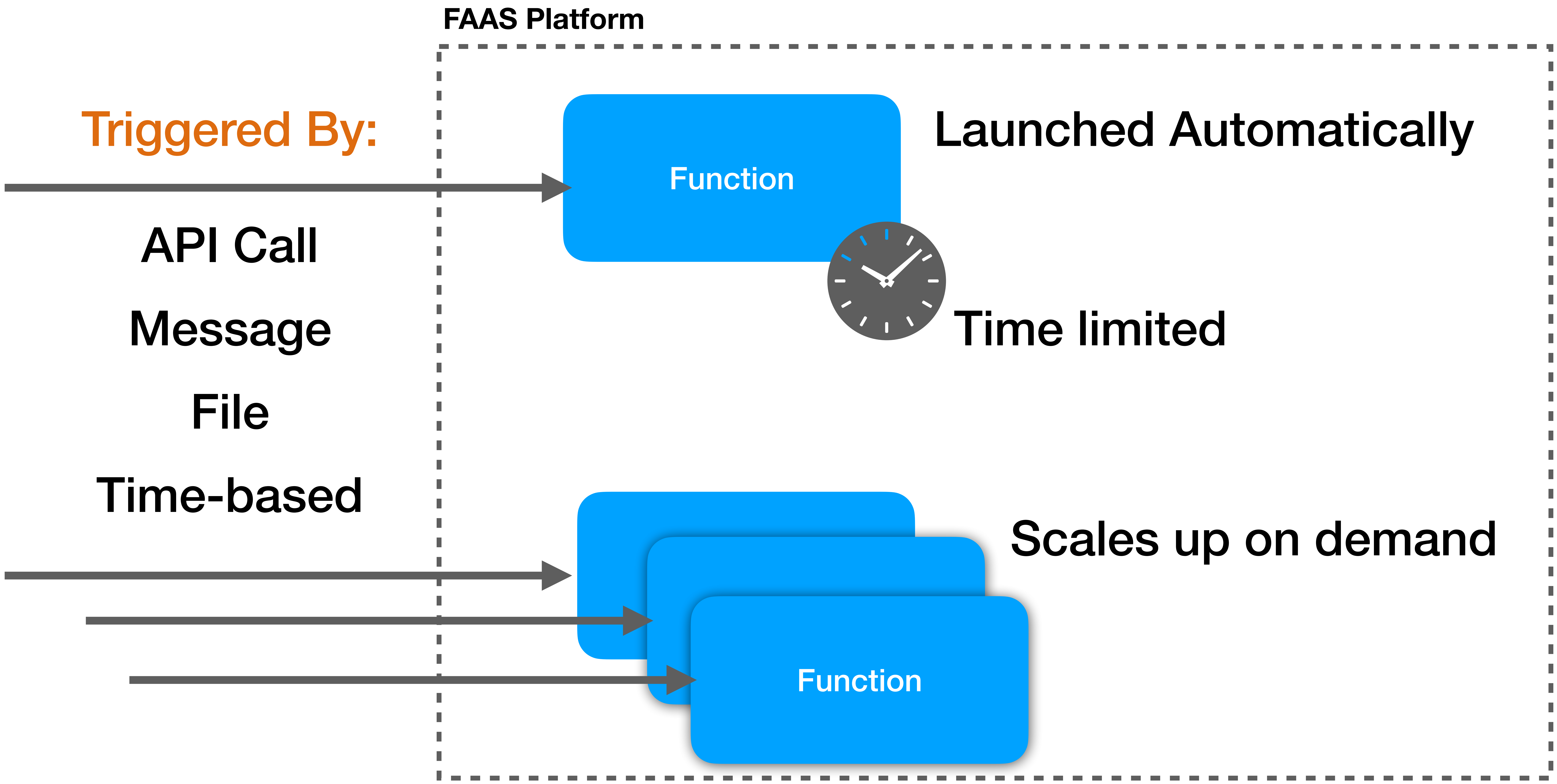
Launched Automatically

Time limited

Function

Scales up on demand

FAAS IN A NUTSHELL



FAAS Platform

Triggered By:

API Call

Message

File

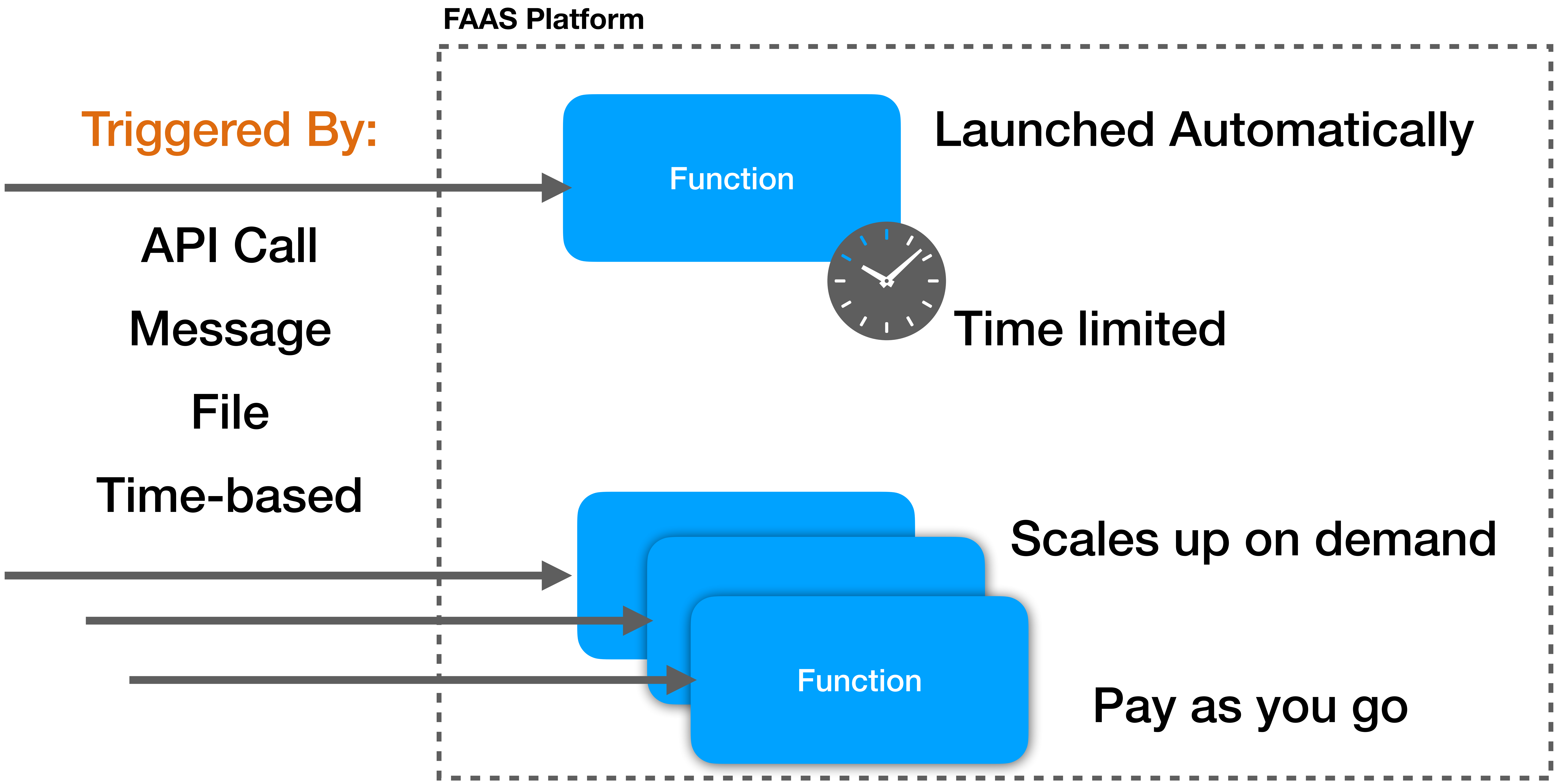
Time-based

Launched Automatically

Time limited

Scales up on demand

FAAS IN A NUTSHELL



FAAS Platform

Triggered By:

API Call

Message

File

Time-based

Launched Automatically

Time limited

Scales up on demand

Pay as you go

FAAS LIMITATIONS

FAAS LIMITATIONS

Time limited

FAAS LIMITATIONS

Time limited

Runtime restrictions

FAAS LIMITATIONS

Time limited

Runtime restrictions

Stateless

FAAS LIMITATIONS

Time limited

Runtime restrictions

Stateless

Limited ability to control scale-up

FAAS LIMITATIONS

Time limited

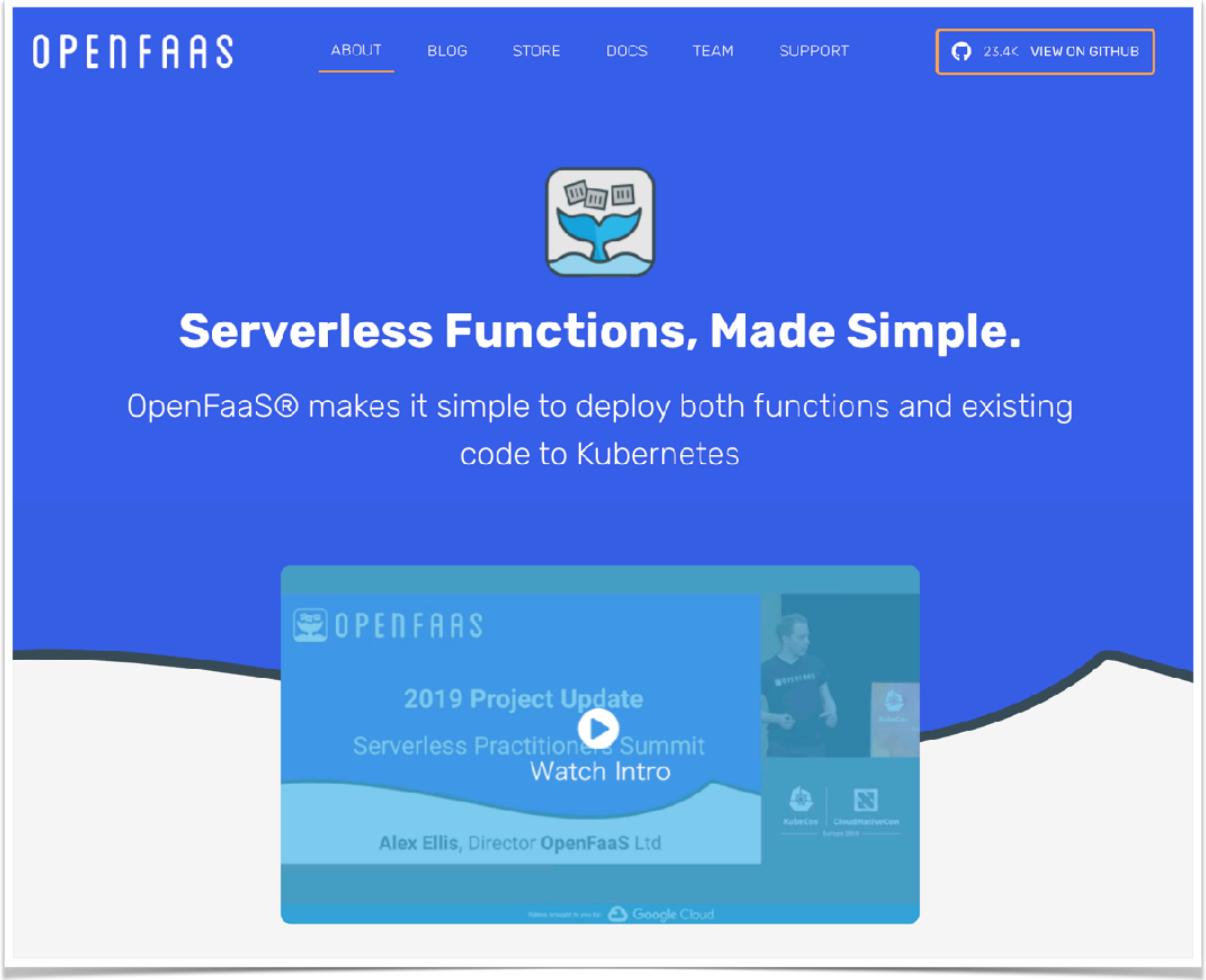
Runtime restrictions

Stateless

Limited ability to control scale-up

Often have to rely on platform observability

#serverless + Kubernetes?



OPENFAAS

ABOUT BLOG STORE DOCS TEAM SUPPORT

23.4k VIEW ON GITHUB

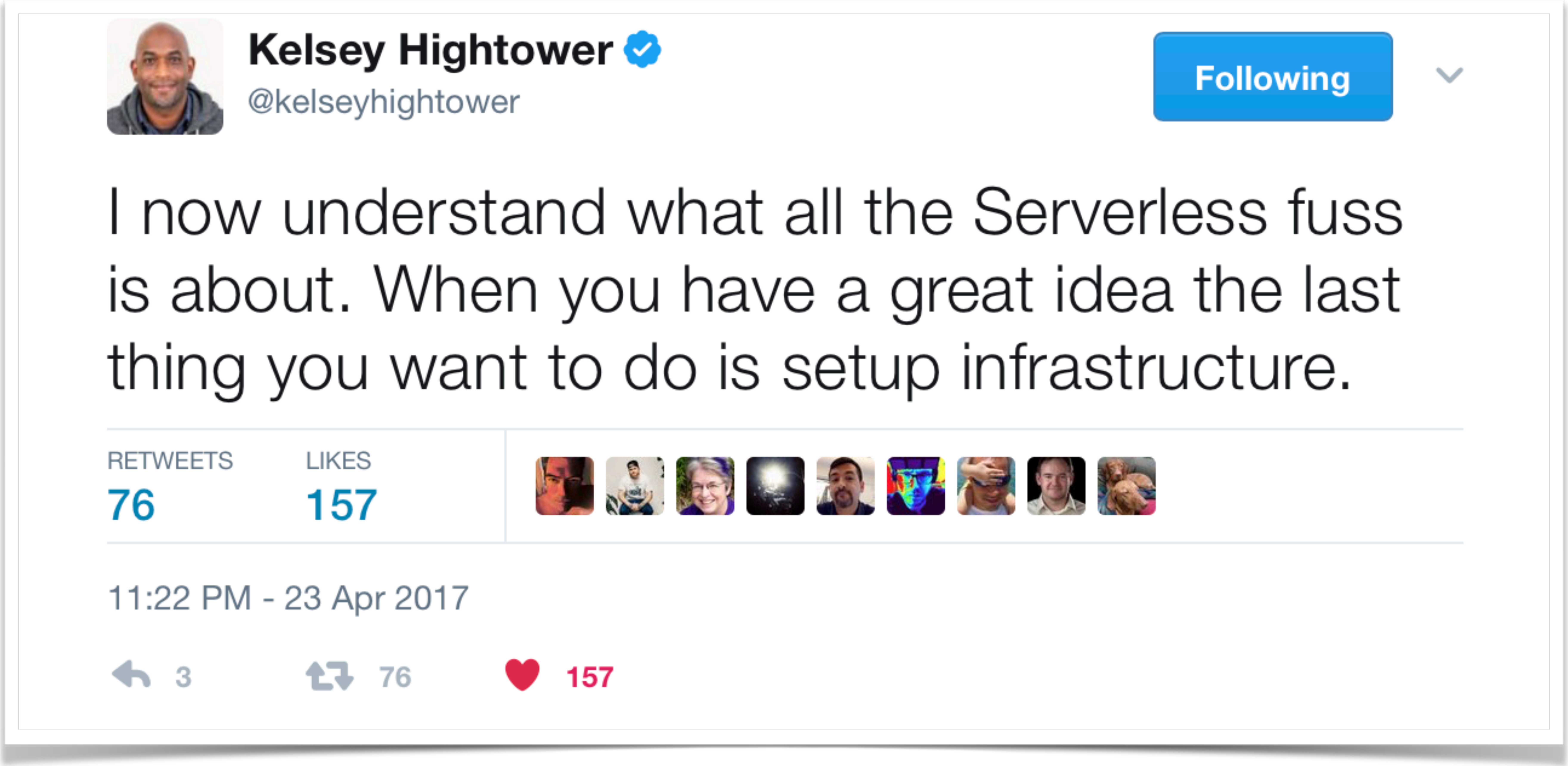
Serverless Functions, Made Simple.

OpenFaaS® makes it simple to deploy both functions and existing code to Kubernetes

OPENFAAS
2019 Project Update
Serverless Practitioners Summit
Watch Intro
Alex Ellis, Director OpenFaaS Ltd
Google Cloud

https://www.openfaas.com/

@samnewman



Kelsey Hightower
@kelseyhightower

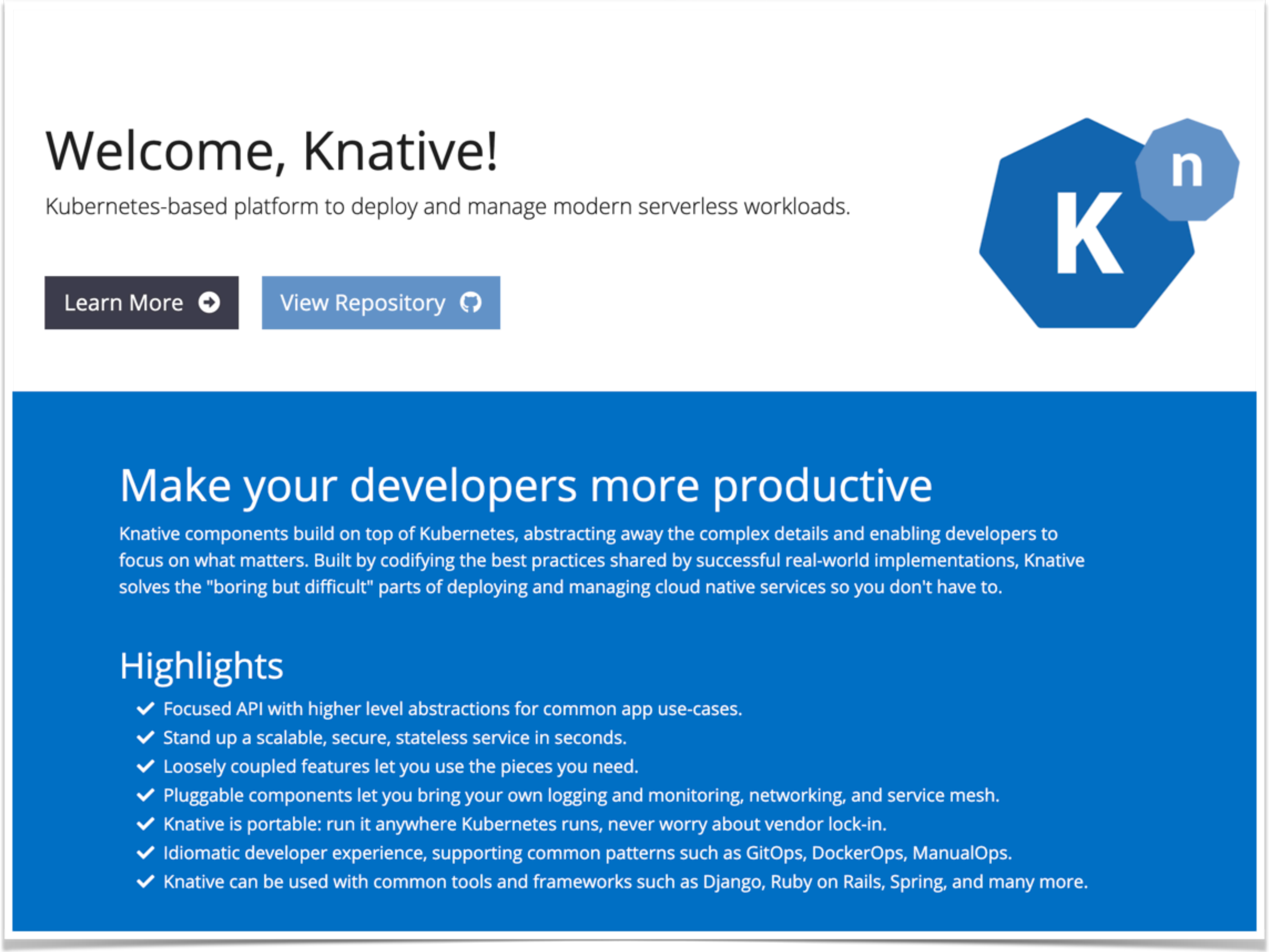
I now understand what all the Serverless fuss is about. When you have a great idea the last thing you want to do is setup infrastructure.

RETWEETS 76 LIKES 157

11:22 PM - 23 Apr 2017

https://twitter.com/kelseyhightower/status/856272003963039744

@samnewman



Welcome, Knative!

Kubernetes-based platform to deploy and manage modern serverless workloads.

Make your developers more productive

Knative components build on top of Kubernetes, abstracting away the complex details and enabling developers to focus on what matters. Built by codifying the best practices shared by successful real-world implementations, Knative solves the "boring but difficult" parts of deploying and managing cloud native services so you don't have to.

Highlights

- ✓ Focused API with higher level abstractions for common app use-cases.
- ✓ Stand up a scalable, secure, stateless service in seconds.
- ✓ Loosely coupled features let you use the pieces you need.
- ✓ Pluggable components let you bring your own logging and monitoring, networking, and service mesh.
- ✓ Knative is portable: run it anywhere Kubernetes runs, never worry about vendor lock-in.
- ✓ Idiomatic developer experience, supporting common patterns such as GitOps, DockerOps, ManualOps.
- ✓ Knative can be used with common tools and frameworks such as Django, Ruby on Rails, Spring, and many more.

https://knative.dev/

Less infrastructure management

Less infrastructure management

More focus on delivering value

FAAS is the best abstraction we’ve come up with for developer-friendly deployment of code since Heroku

FAAS is, hopefully, the future for most of us.

FAAS is, hopefully, the future for most of us.

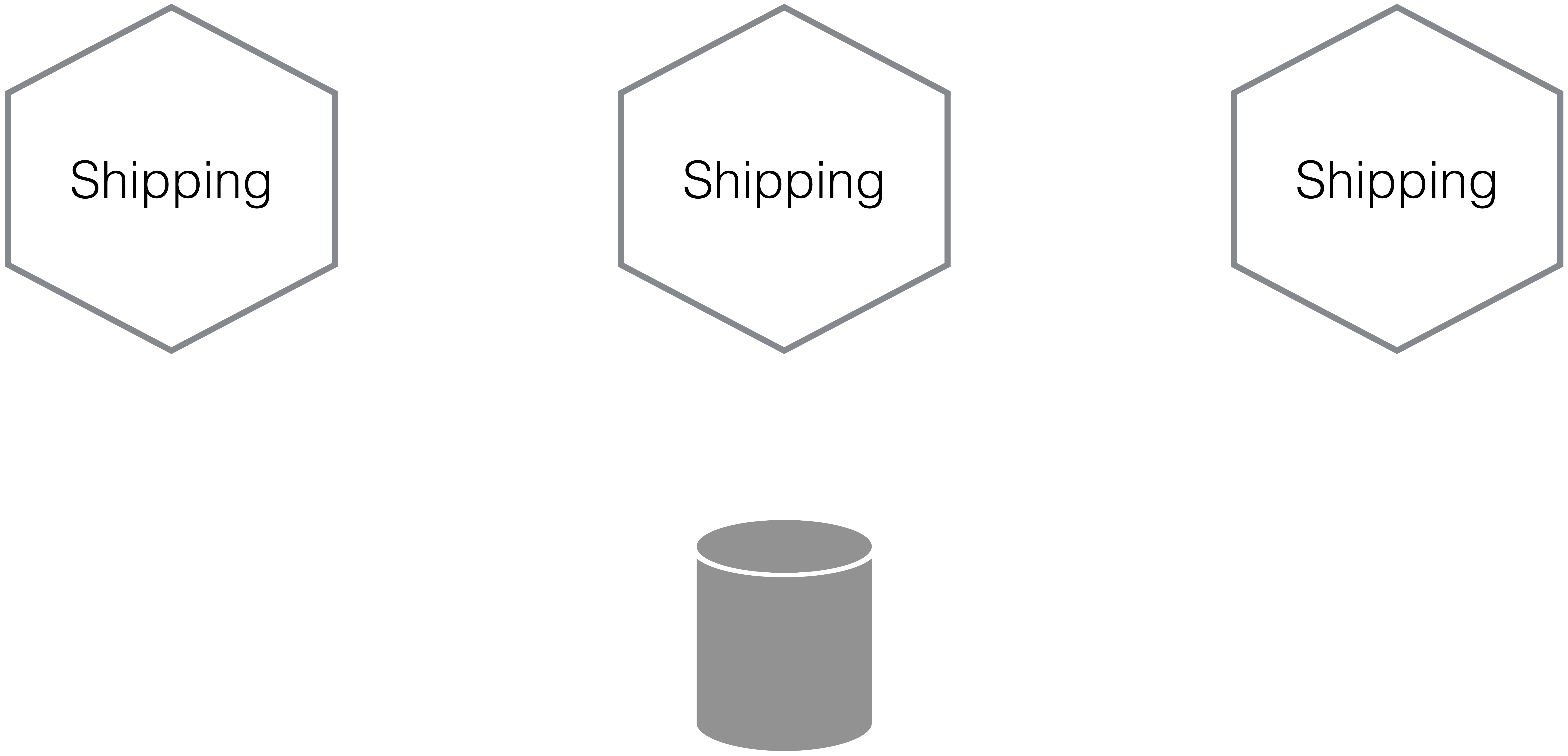
Even if some of the current implementations suck.

Part 2: Microservices and Functions

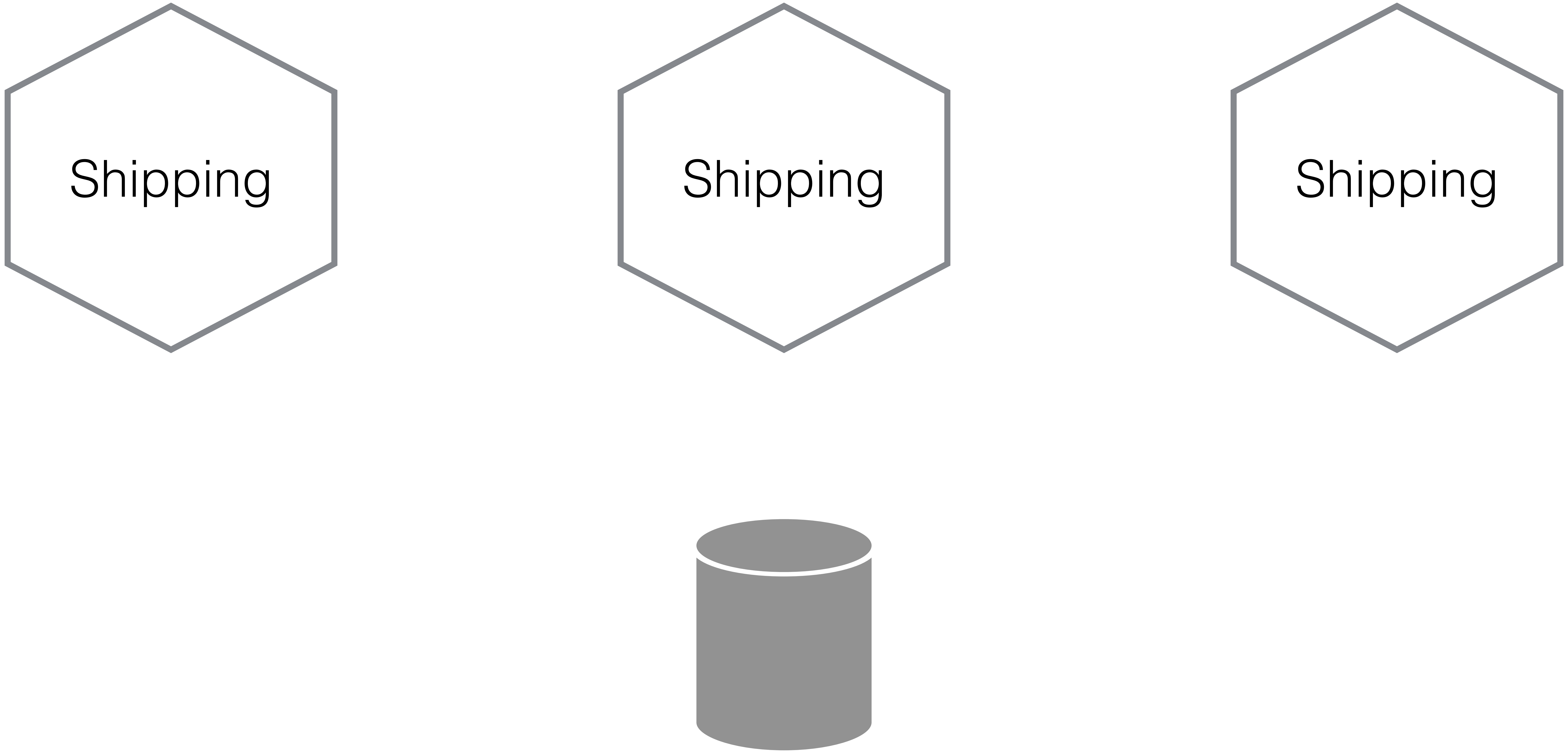
https://www.flickr.com/photos/mustangjoe/20437315996/

ANATOMY OF A MICROSERVICE

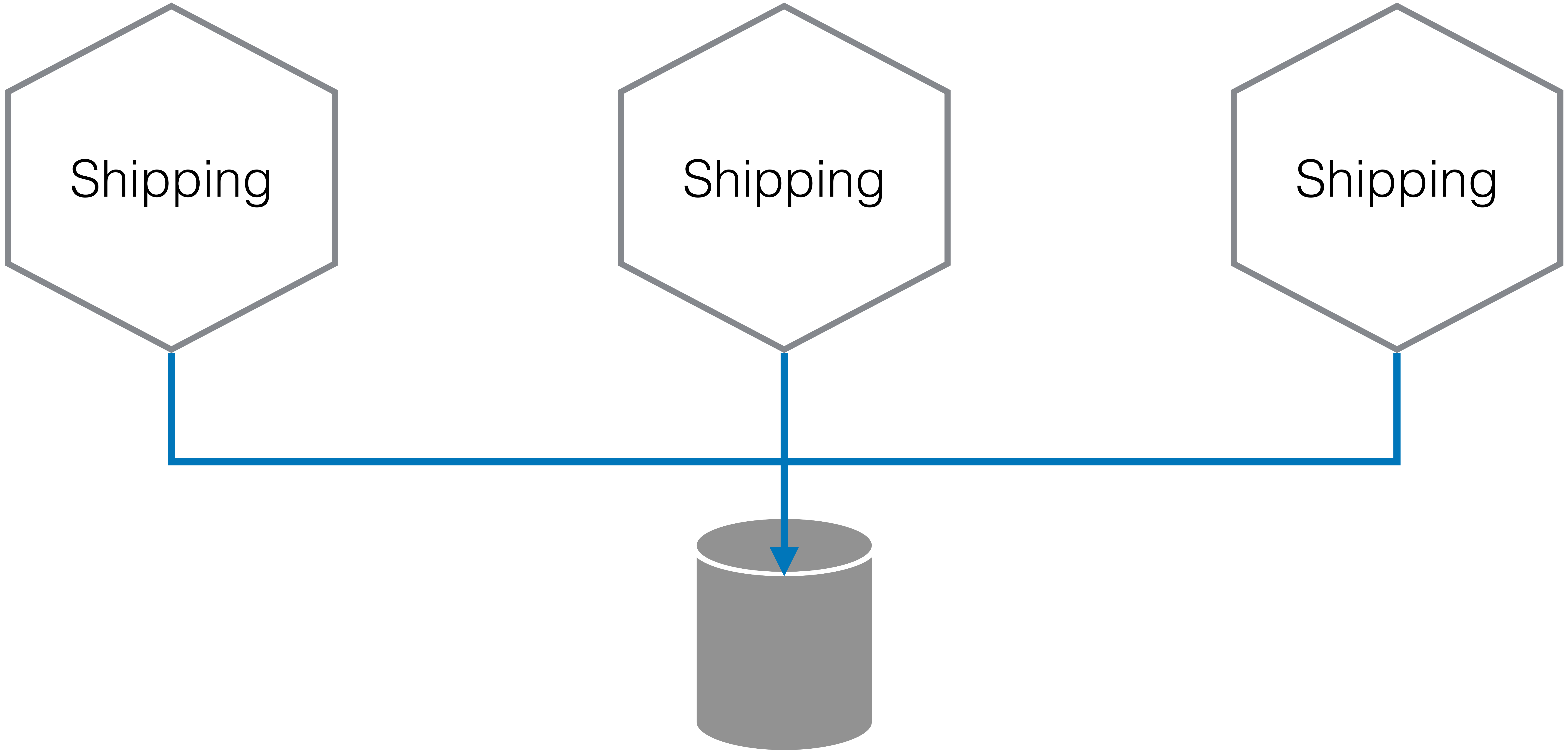
ANATOMY OF A MICROSERVICE



Multiple Instances

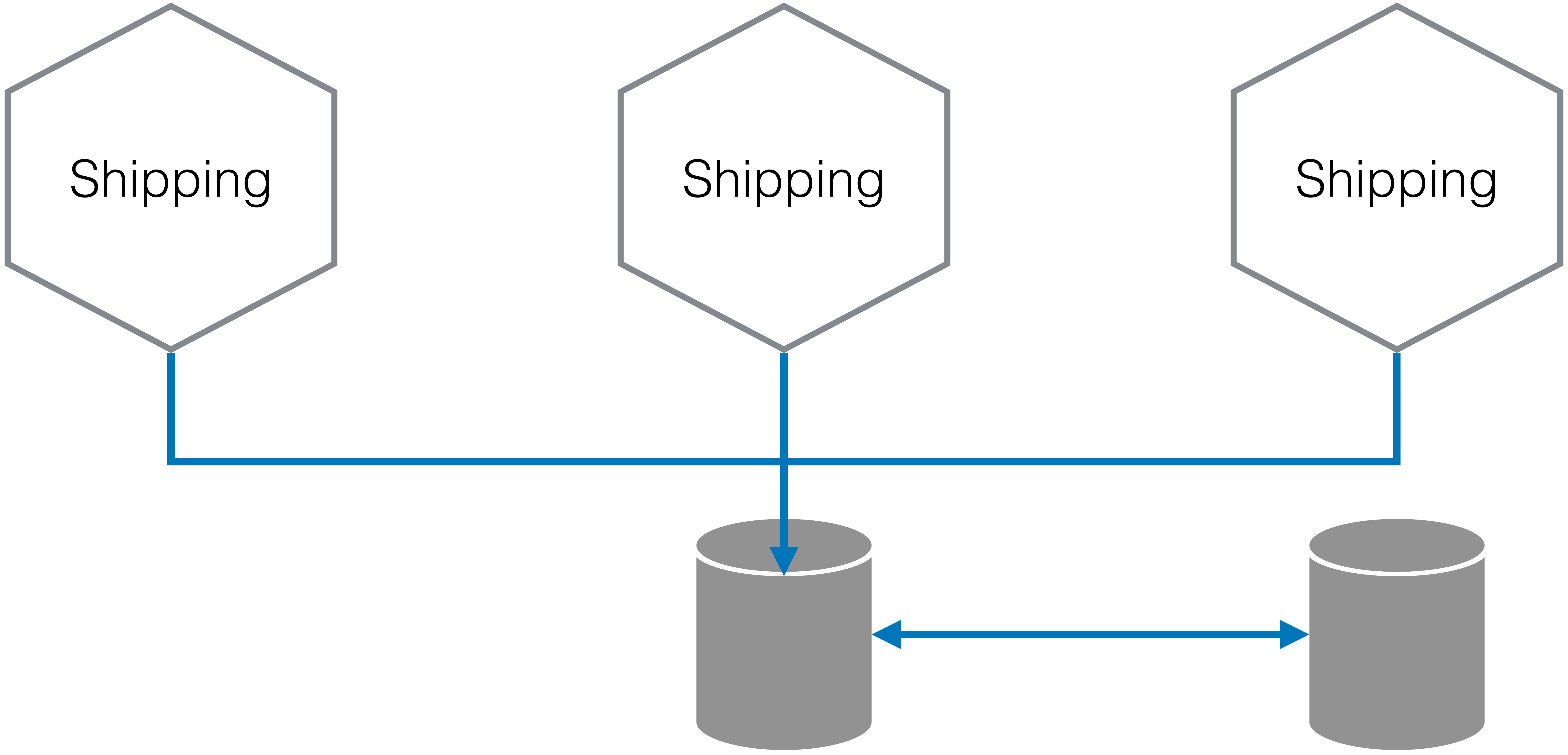


Multiple Instances



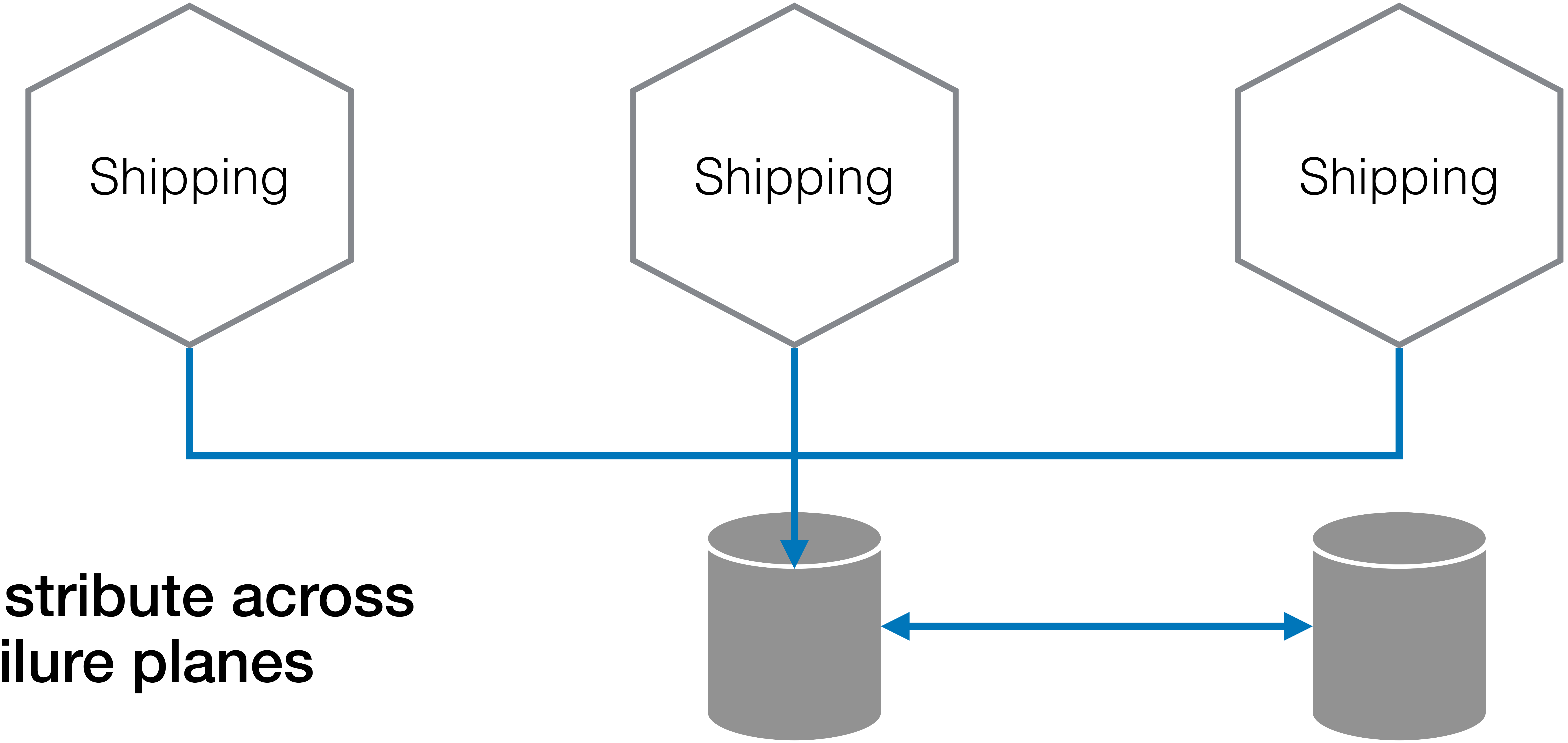
ANATOMY OF A MICROSERVICE

Multiple Instances



ANATOMY OF A MICROSERVICE

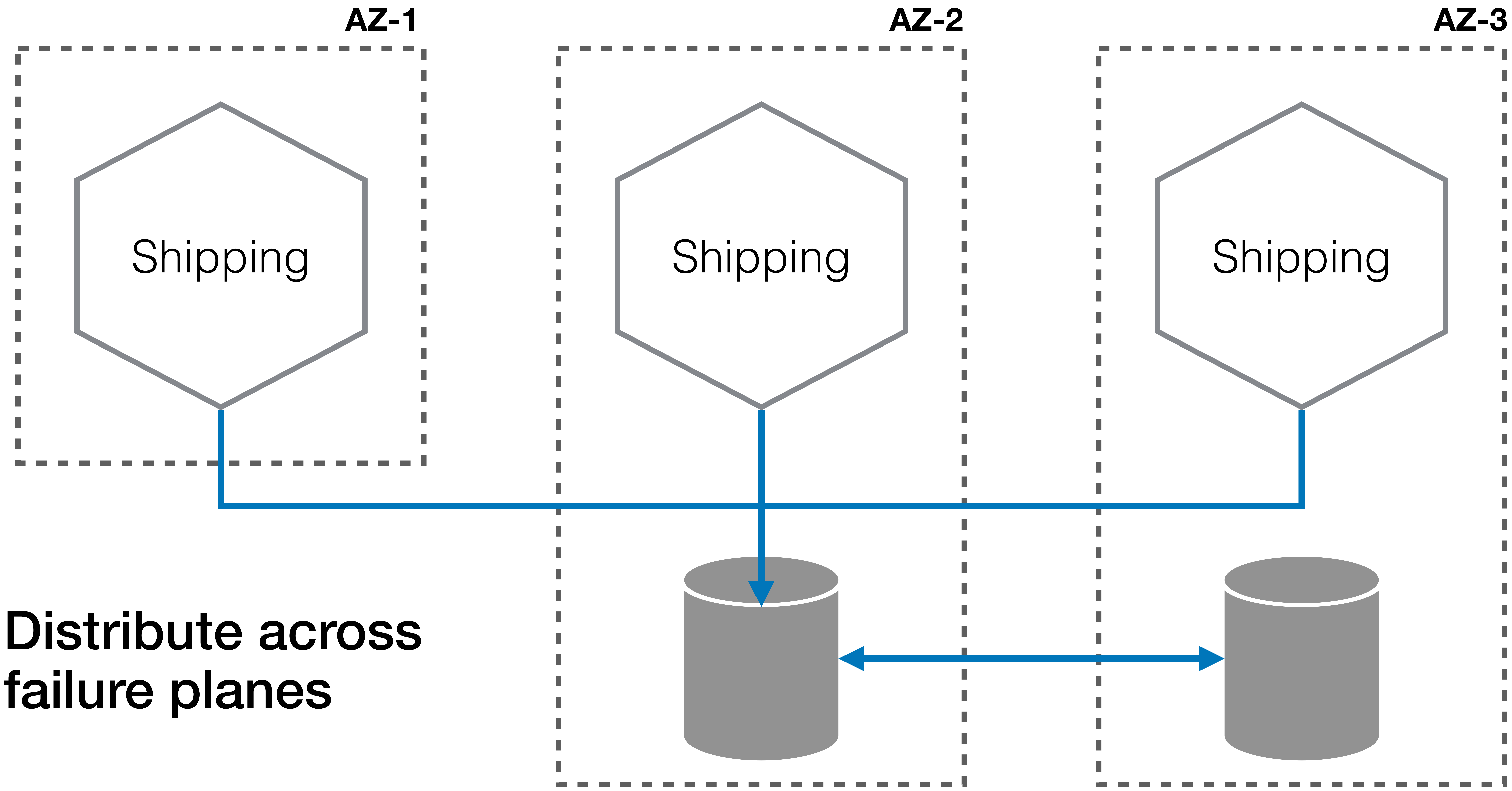
Multiple Instances



Distribute across failure planes

ANATOMY OF A MICROSERVICE

Multiple Instances



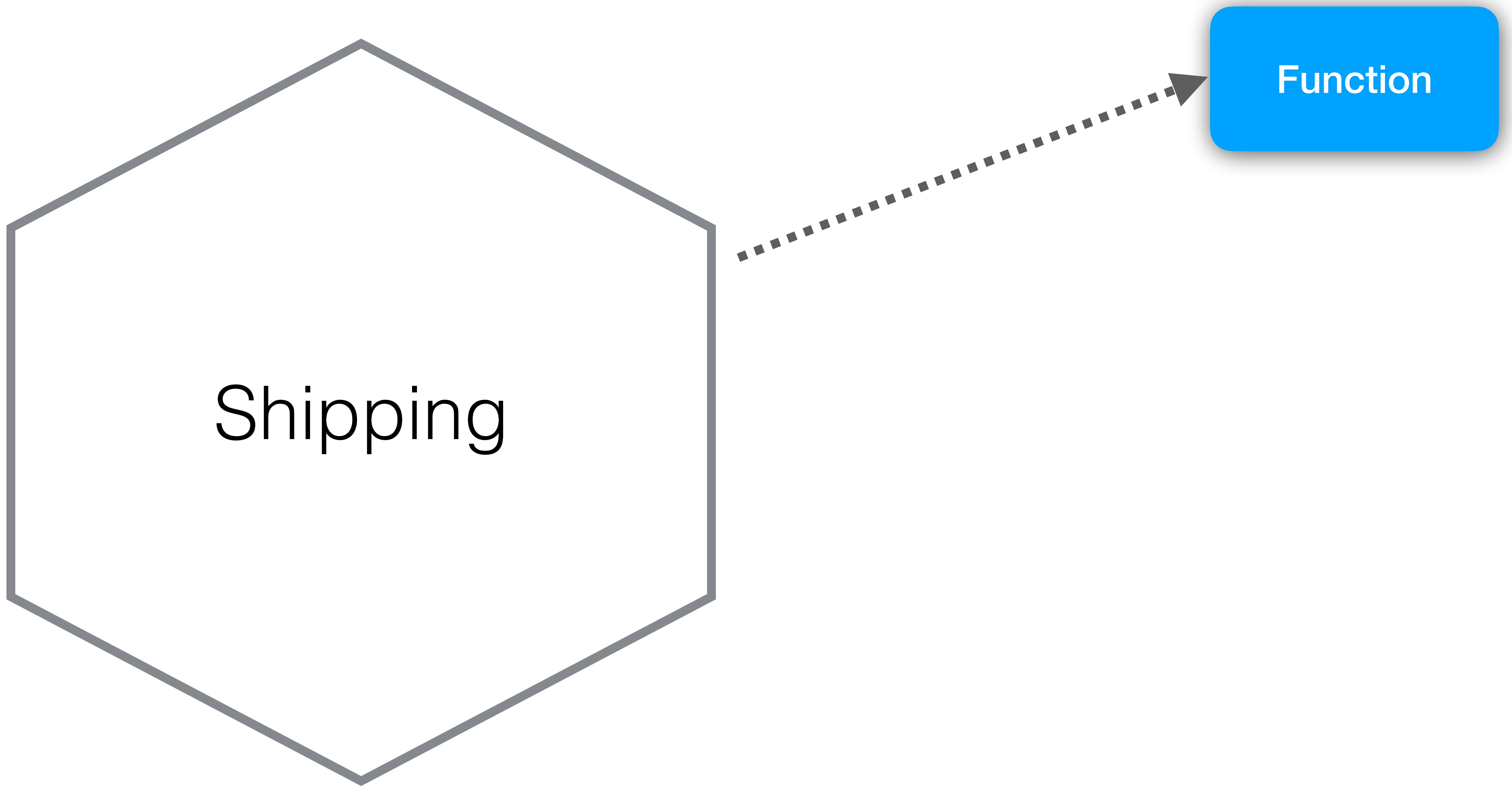
Distribute across failure planes

In general, a microservice instance is assumed to be permanently running waiting for something to happen

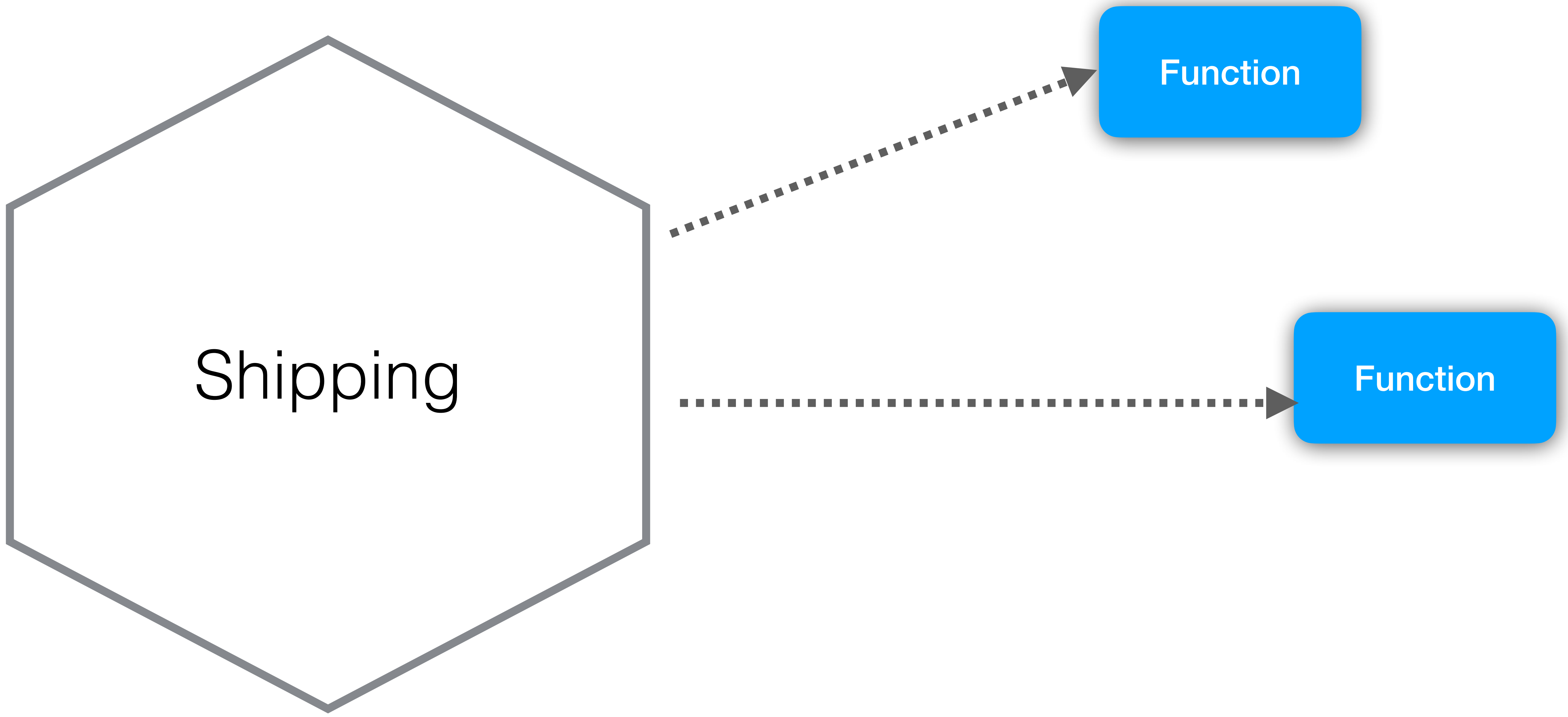
In general, a microservice instance is assumed to be permanently running waiting for something to happen

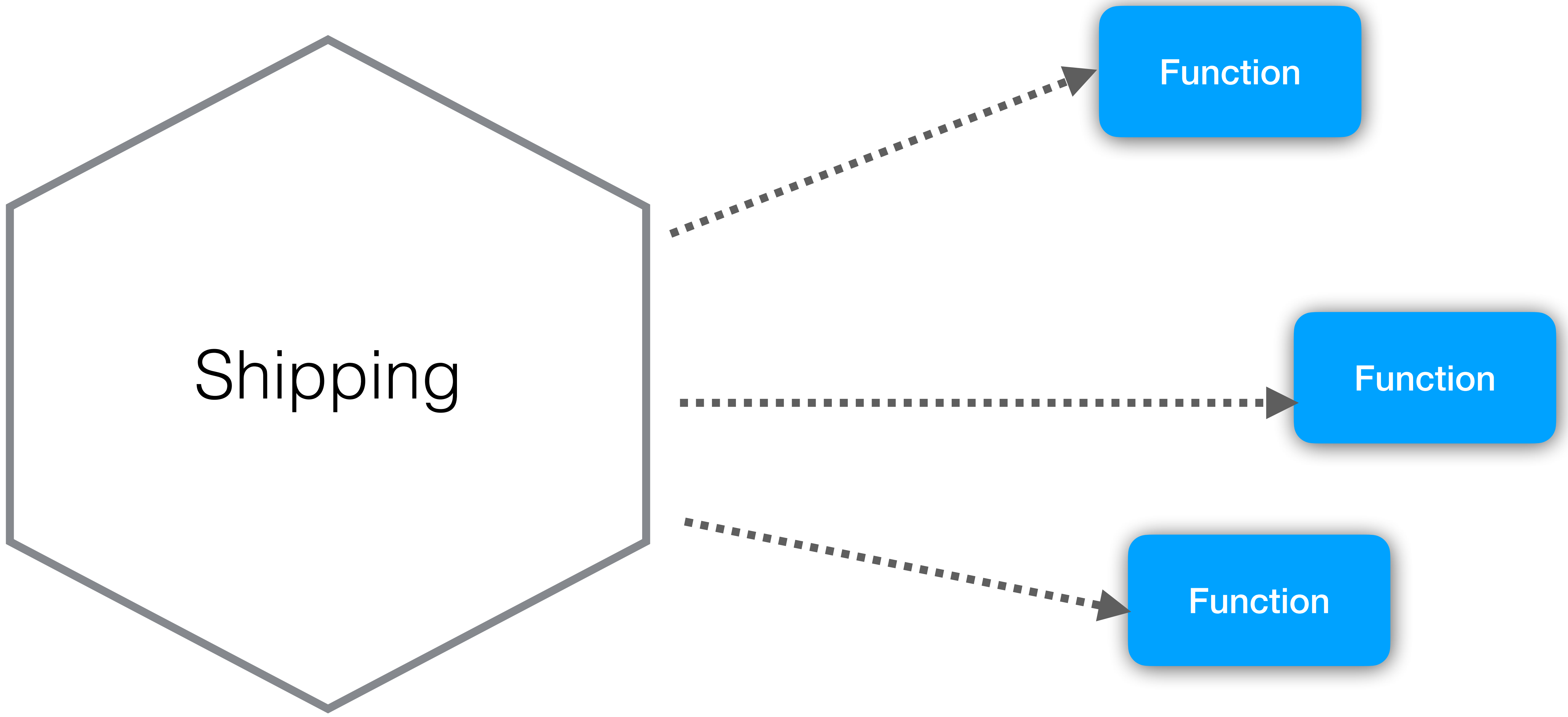
They can be deployed on to physical machines, VMs, or containers

Shipping

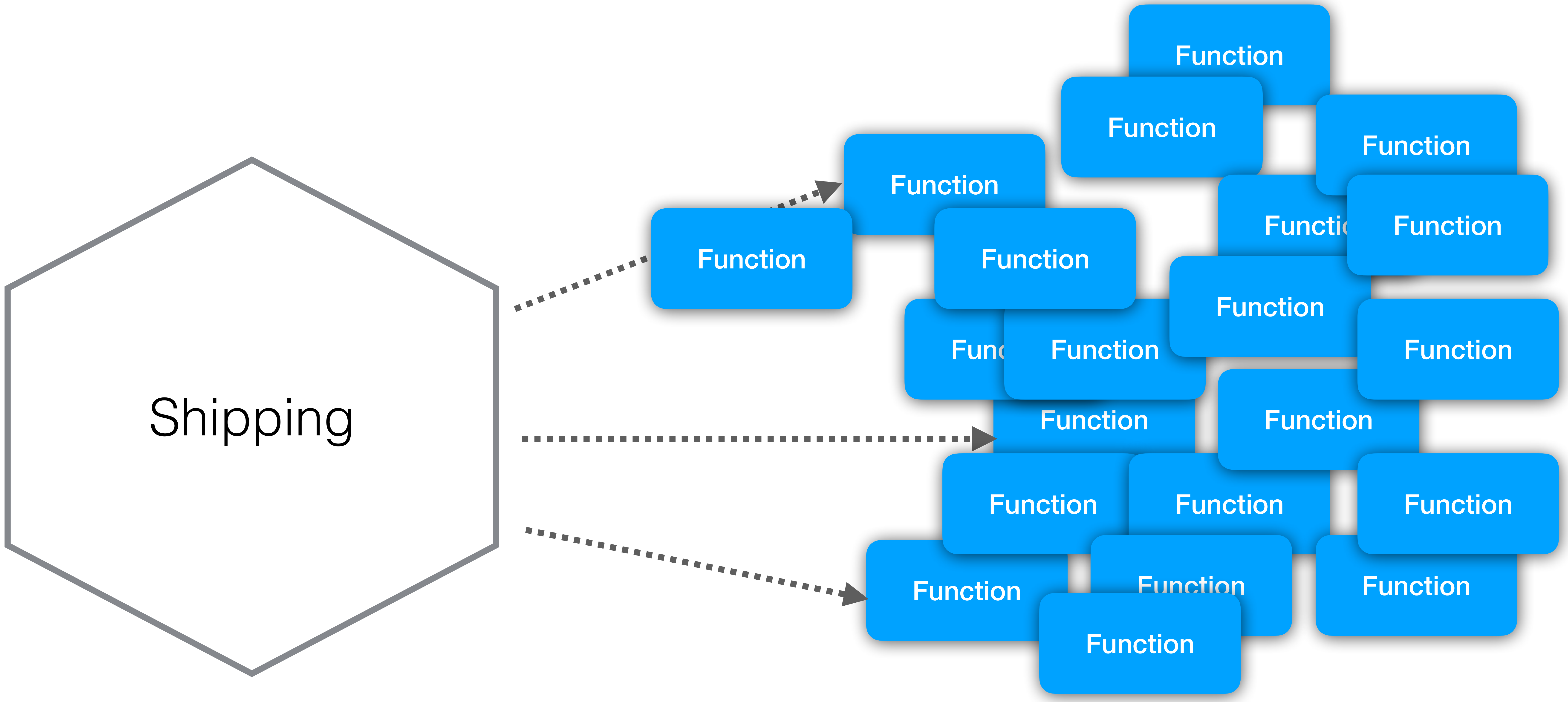


Shipping
Function


Shipping
Function
Function



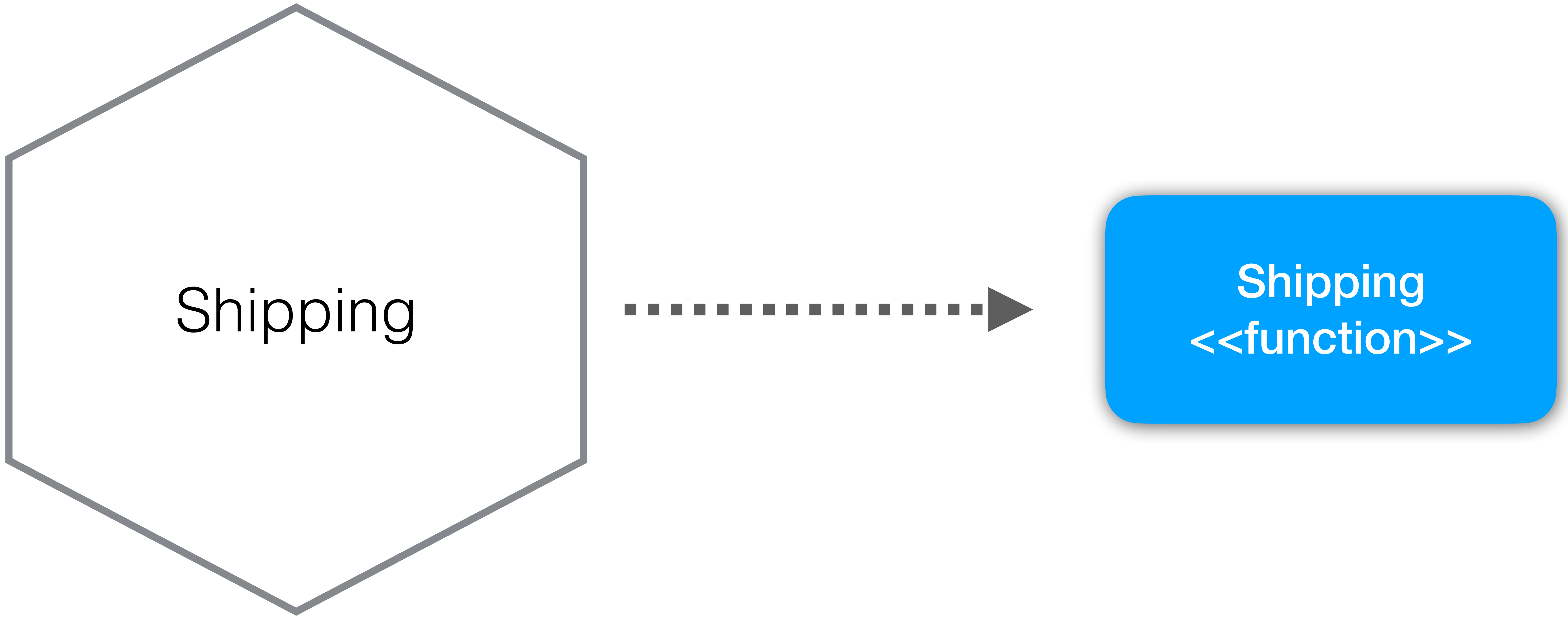
Shipping
Function
Function
Function


Shipping
Function
Function
Function
Function
Function
Function
Functio
Function
Function
Func
Function
Function
Function
Function
Function
Function
Function
Function
Function
Function
Function

@samnewman

A GOOD STARTING POINT

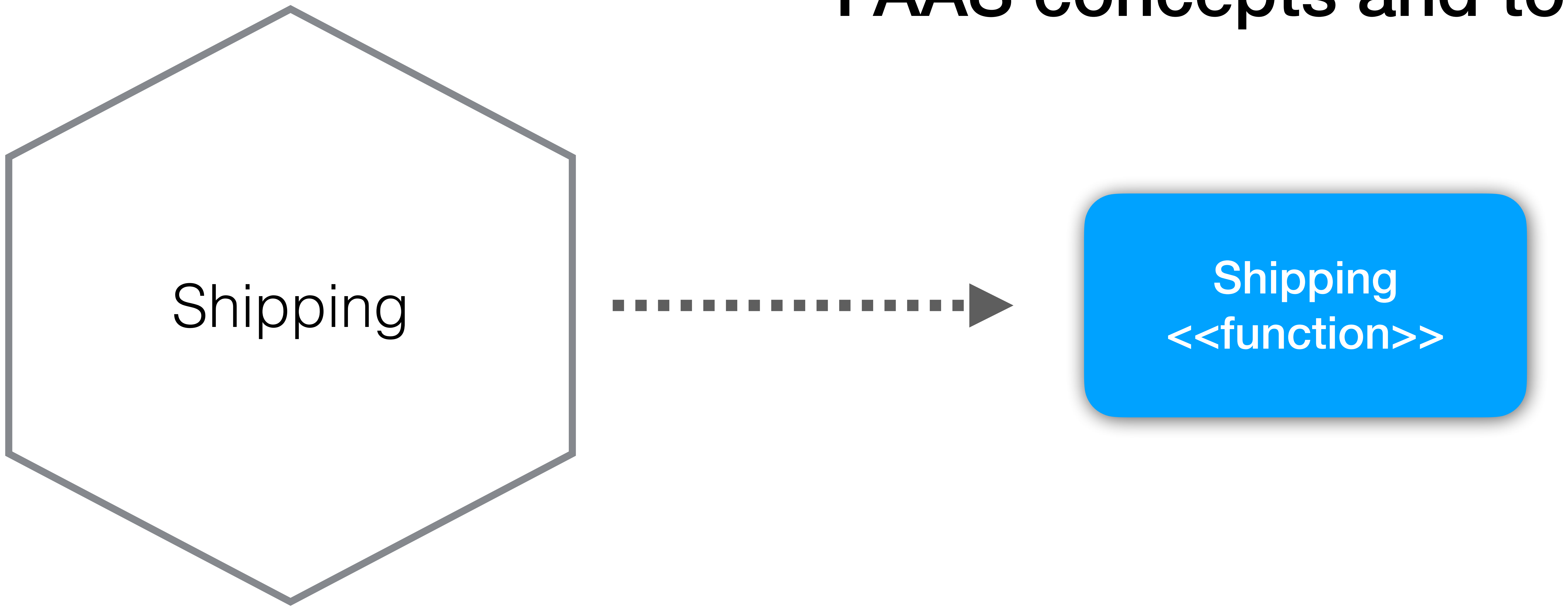
A GOOD STARTING POINT



@samnewman

A GOOD STARTING POINT

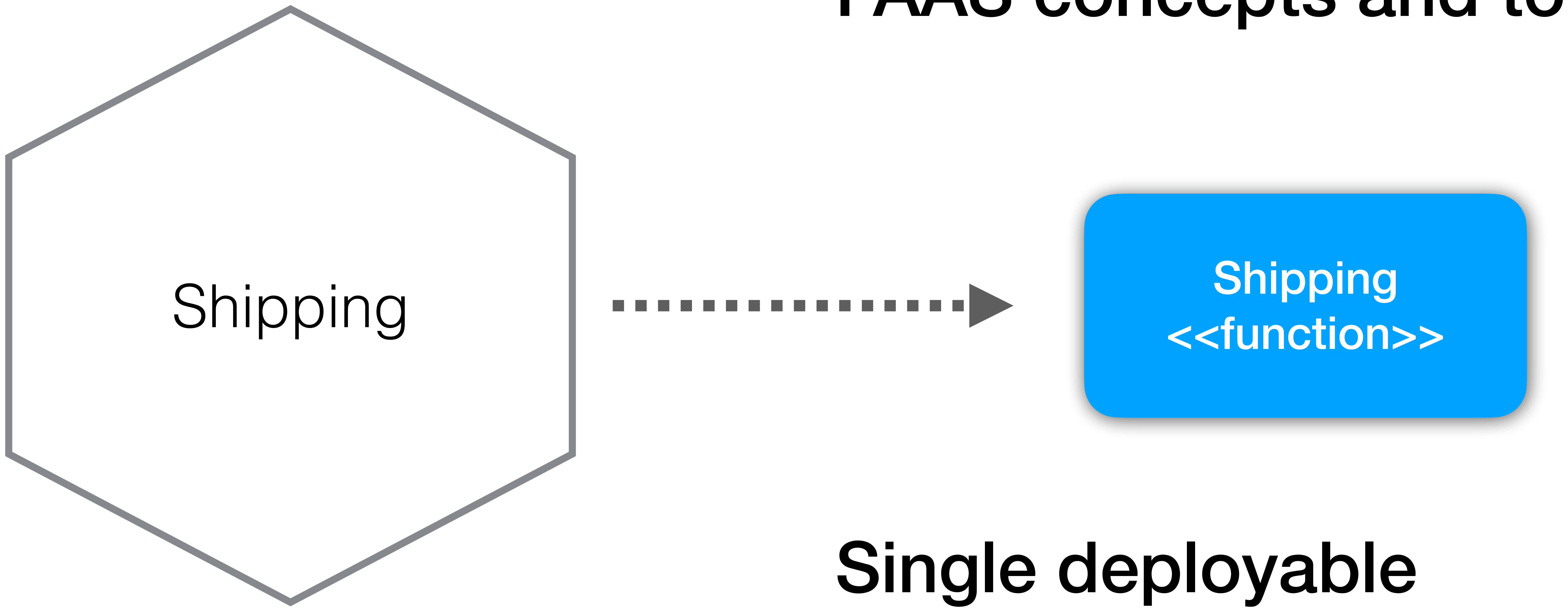
Helps gain familiarity with the FAAS concepts and tooling



@samnewman

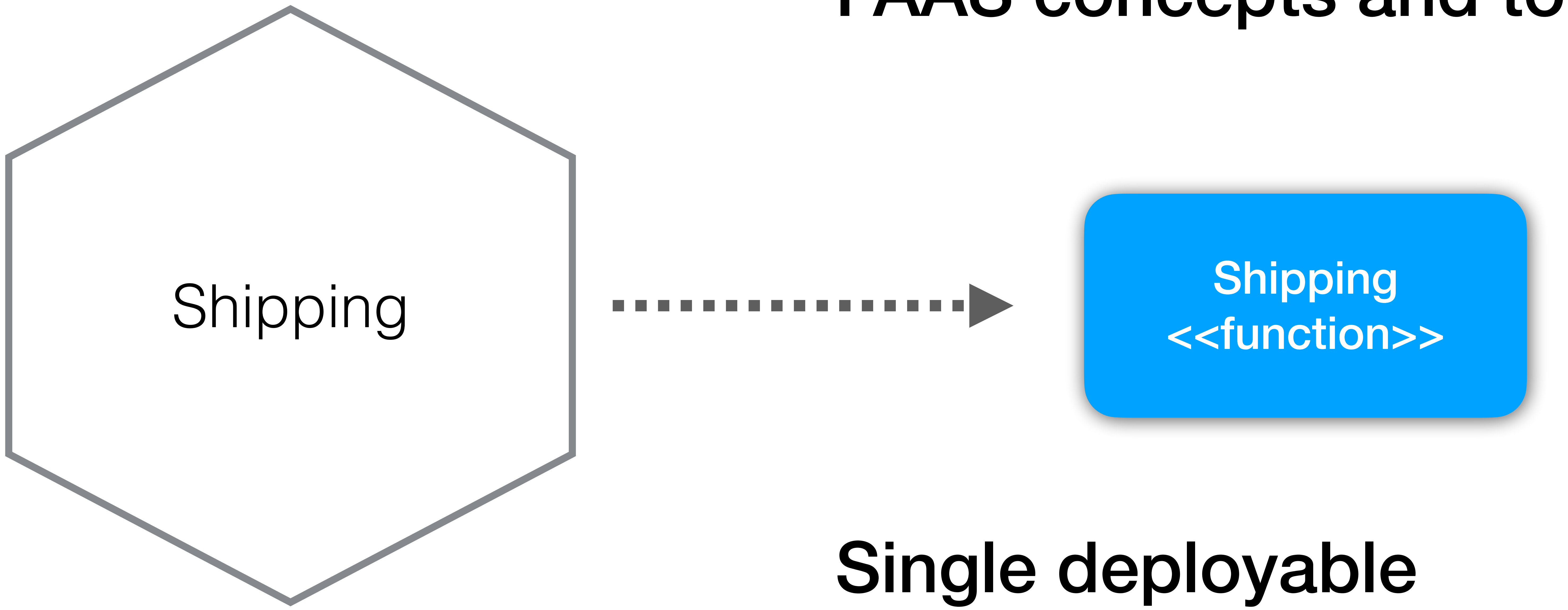
A GOOD STARTING POINT

Helps gain familiarity with the FAAS concepts and tooling



Single deployable

A GOOD STARTING POINT

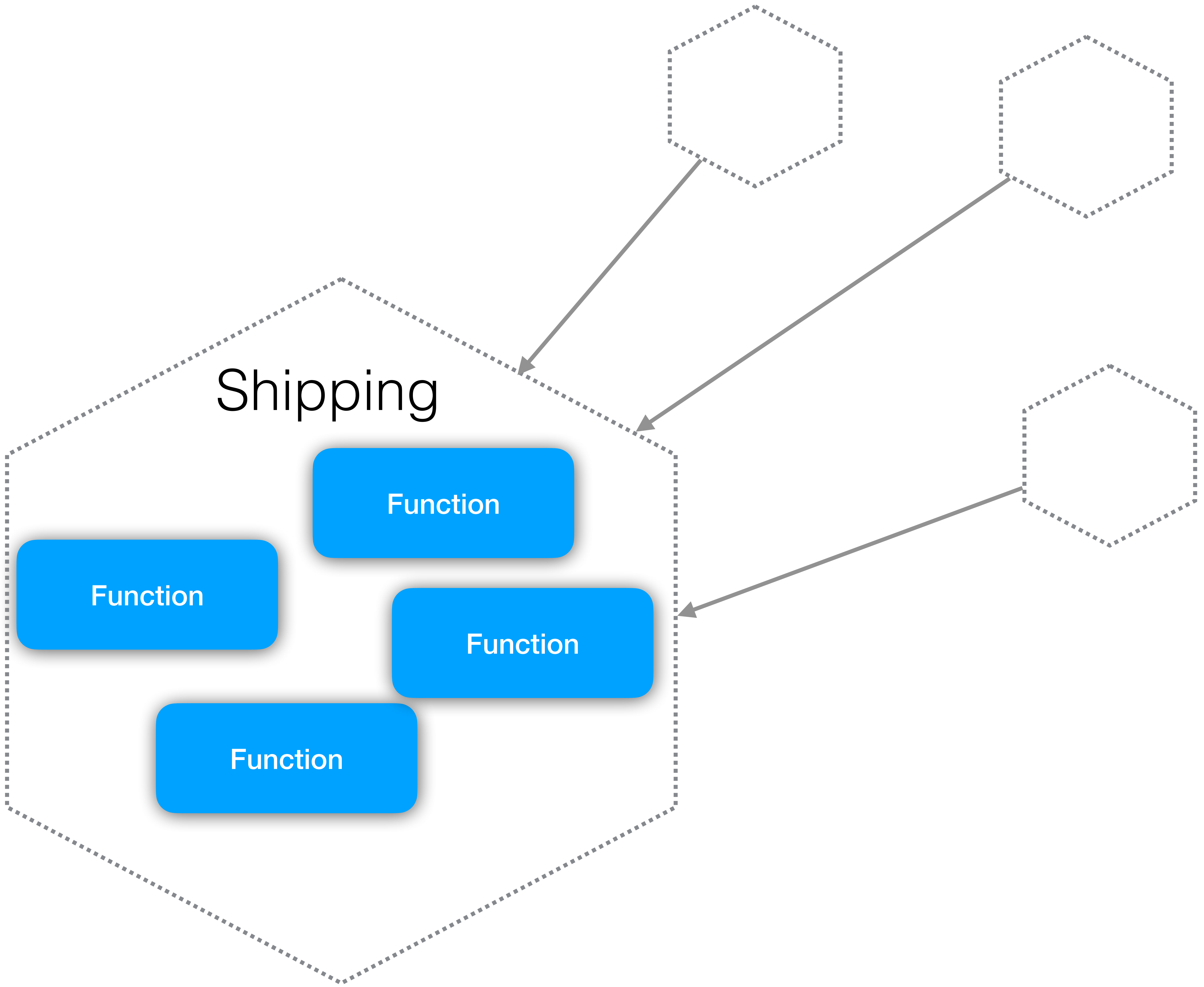


Helps gain familiarity with the FAAS concepts and tooling

Single deployable

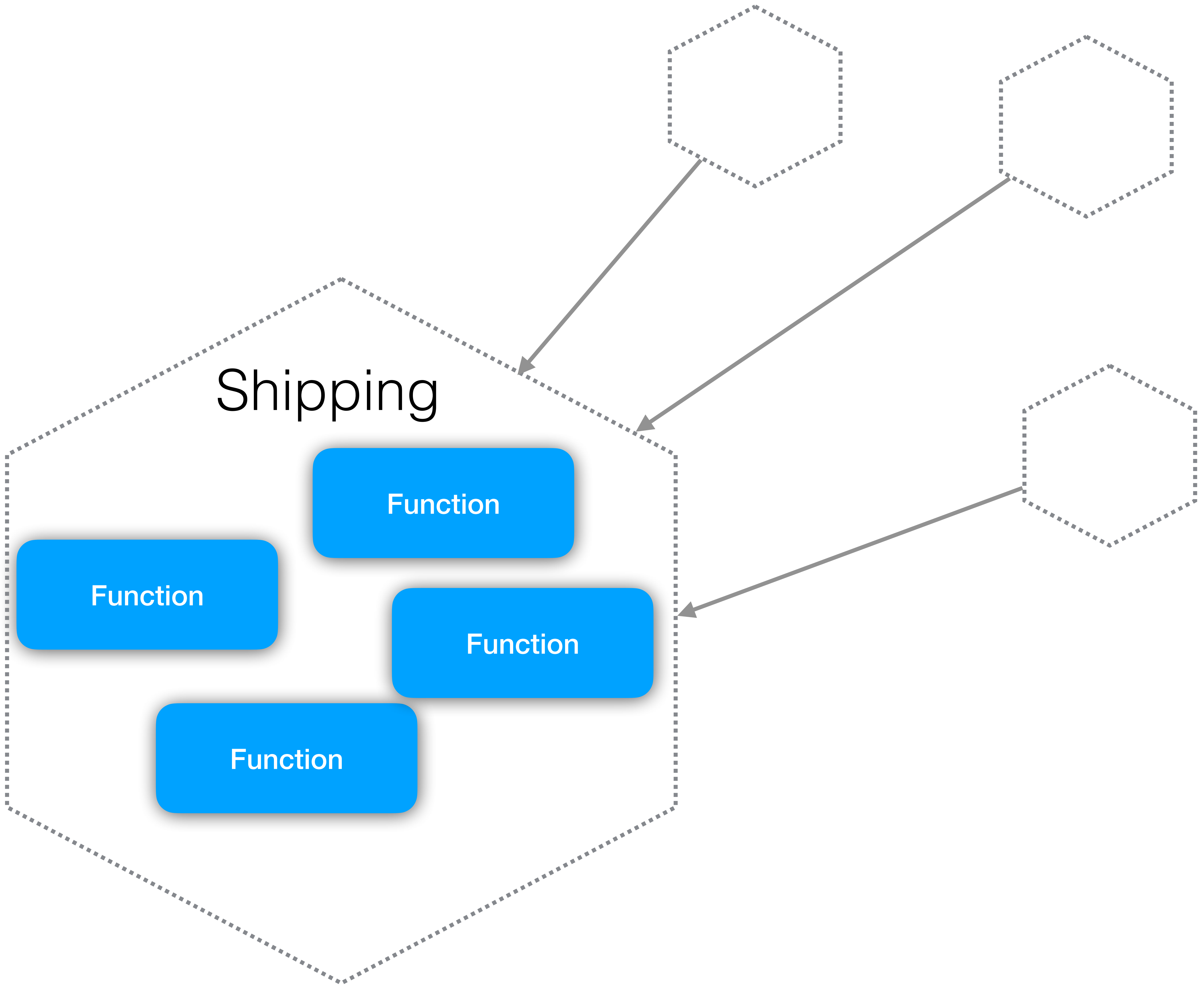
Drastically reduced infra work

GOING FINER GRAINED?



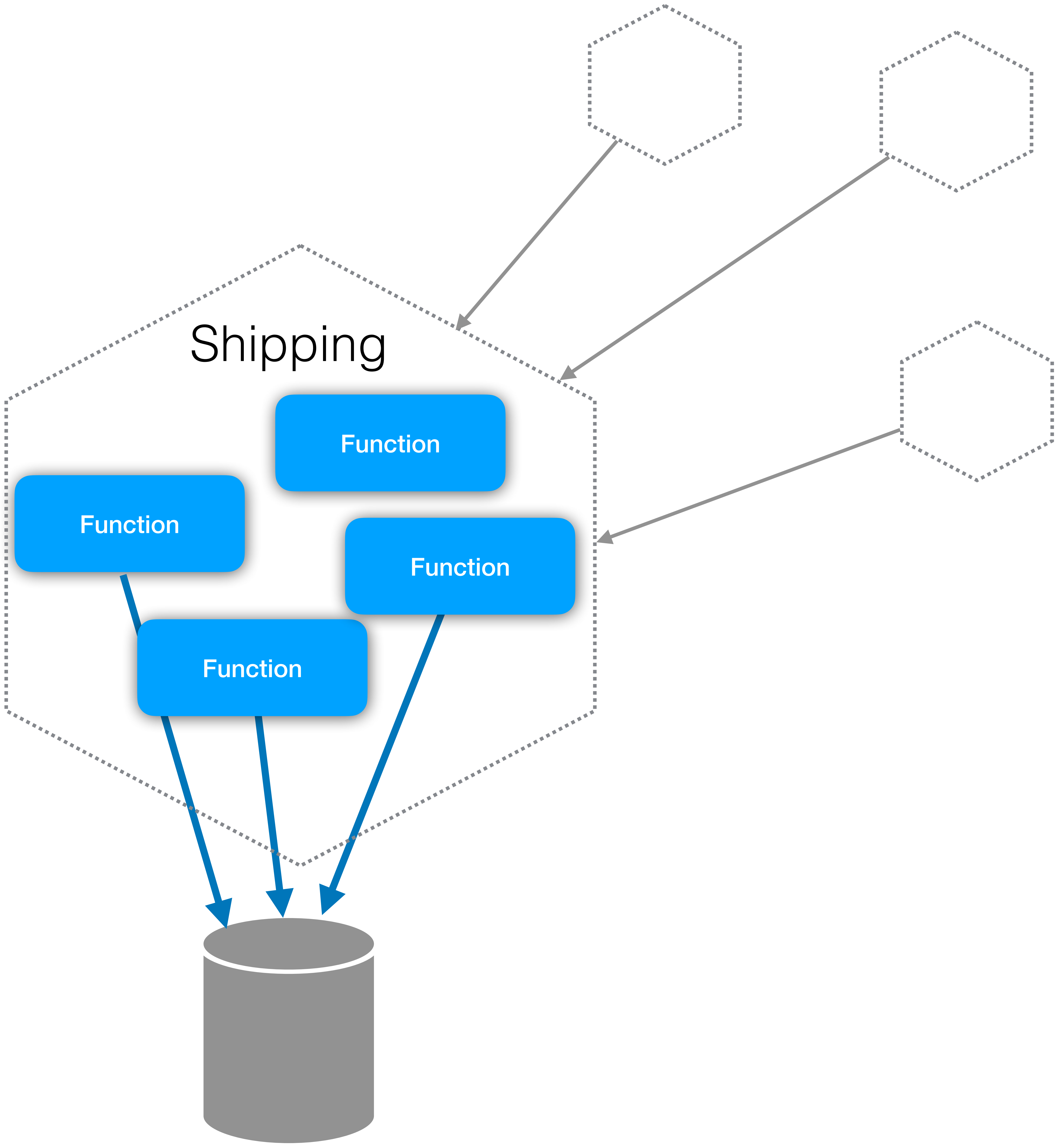
GOING FINER GRAINED?

Microservice becomes more of a “logical” unit, rather than a deployment unit



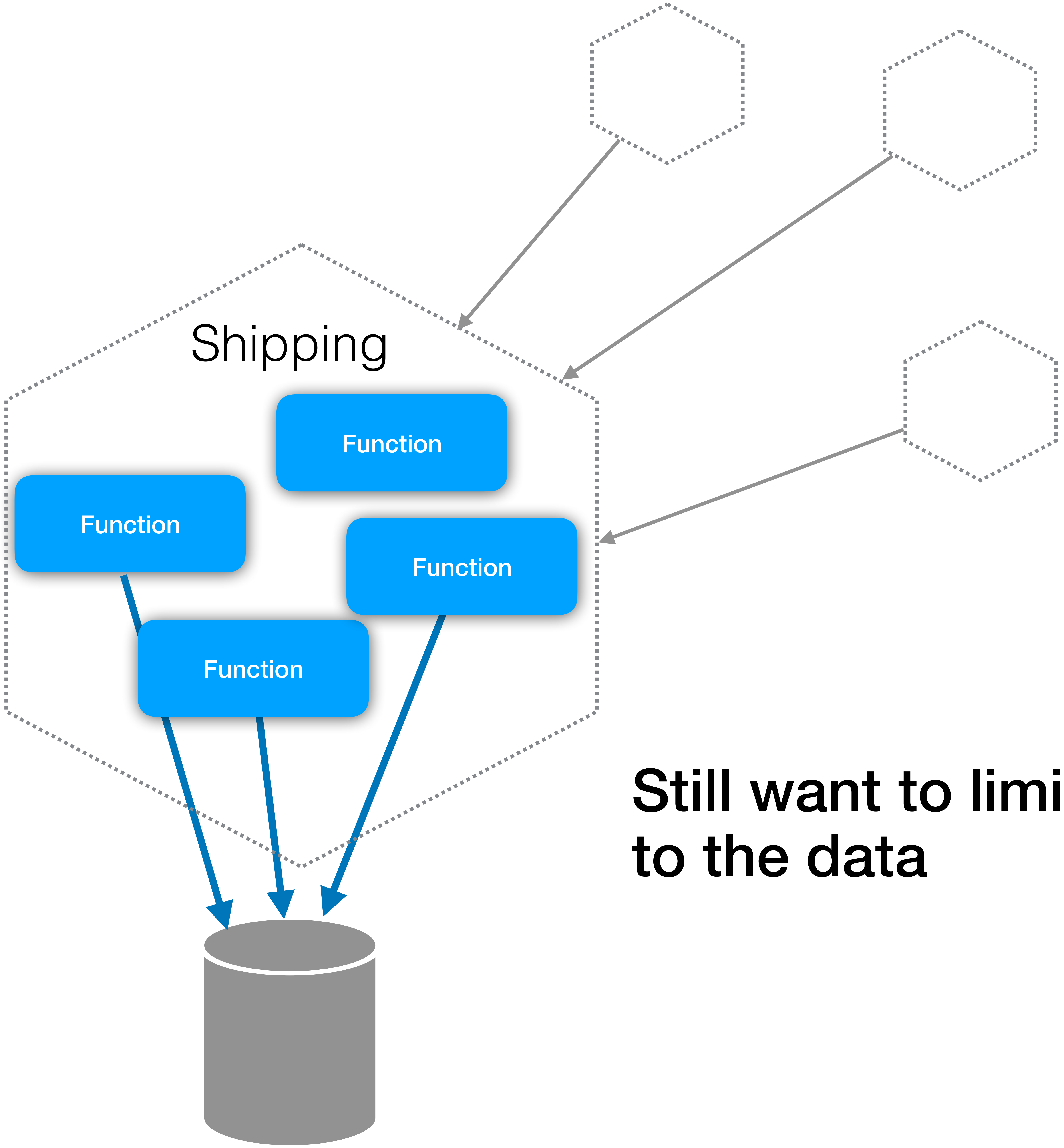
GOING FINER GRAINED?

Microservice becomes more of a "logical" unit, rather than a deployment unit



GOING FINER GRAINED?

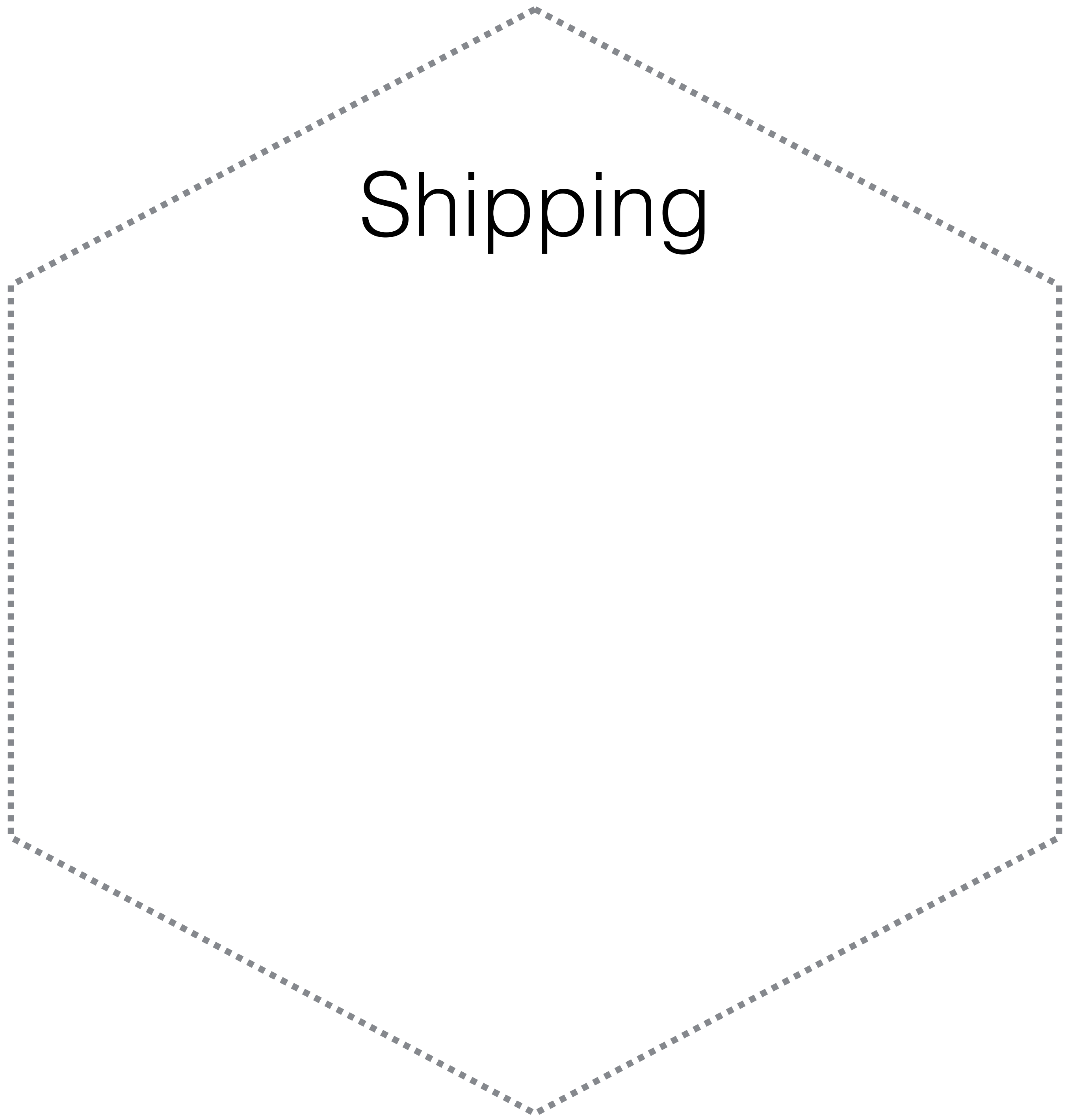
Microservice becomes more of a “logical” unit, rather than a deployment unit



Still want to limit access to the data

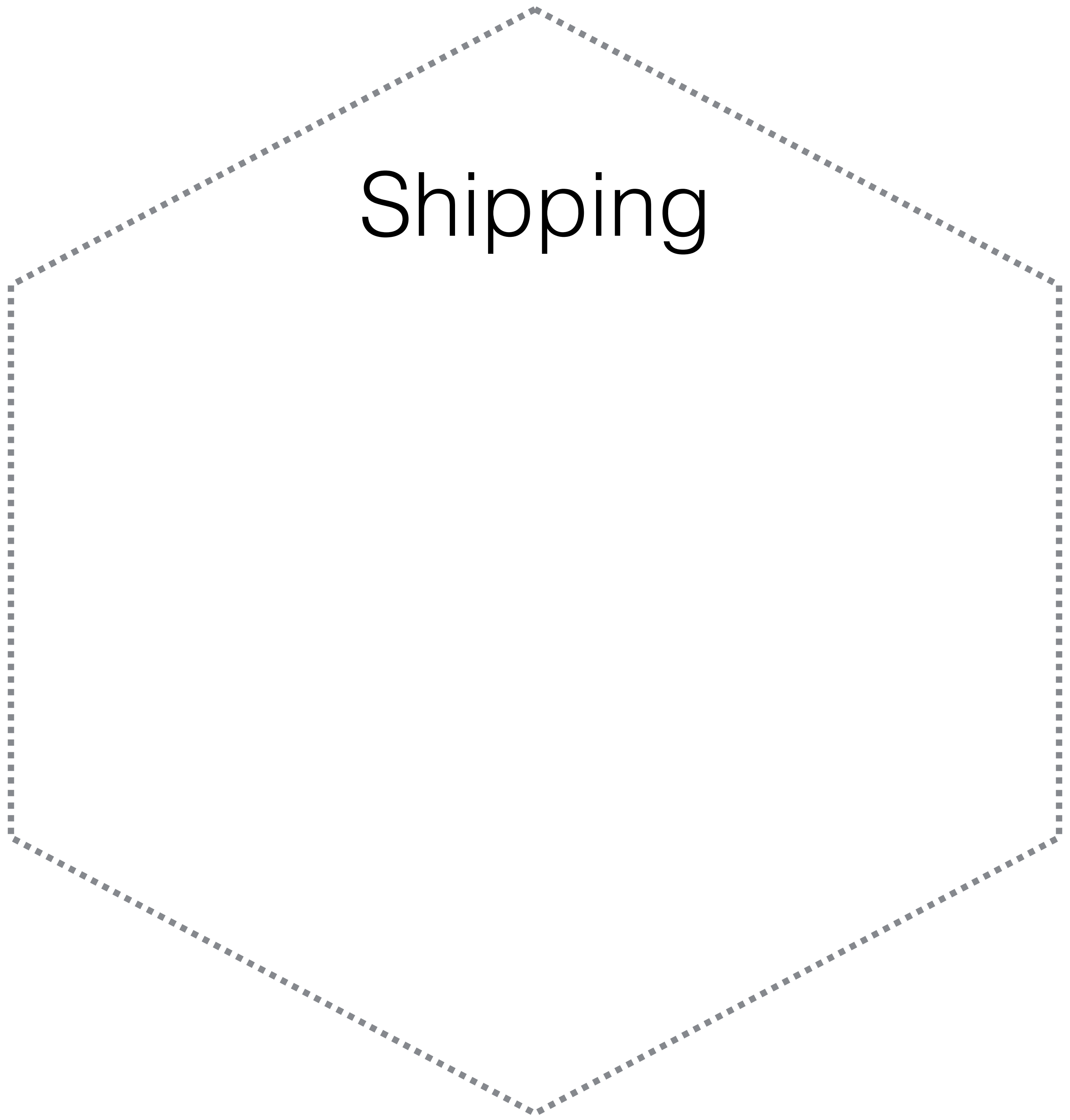
@samnewman

HOW TO FIND THESE FUNCTION BOUNDARIES?



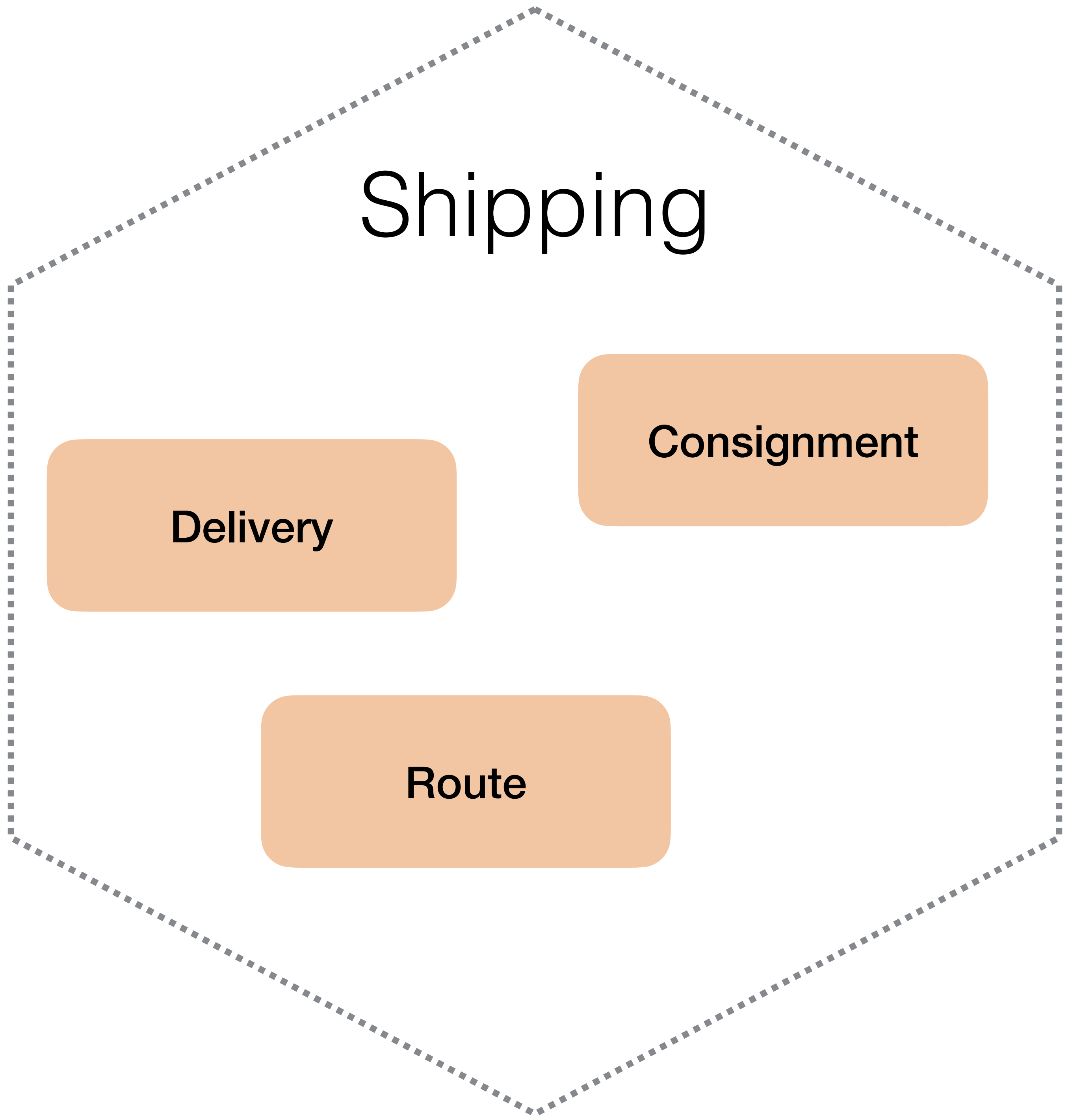
HOW TO FIND THESE FUNCTION BOUNDARIES?

Identify aggregates within your microservice



HOW TO FIND THESE FUNCTION BOUNDARIES?

Identify aggregates within your microservice



Shipping

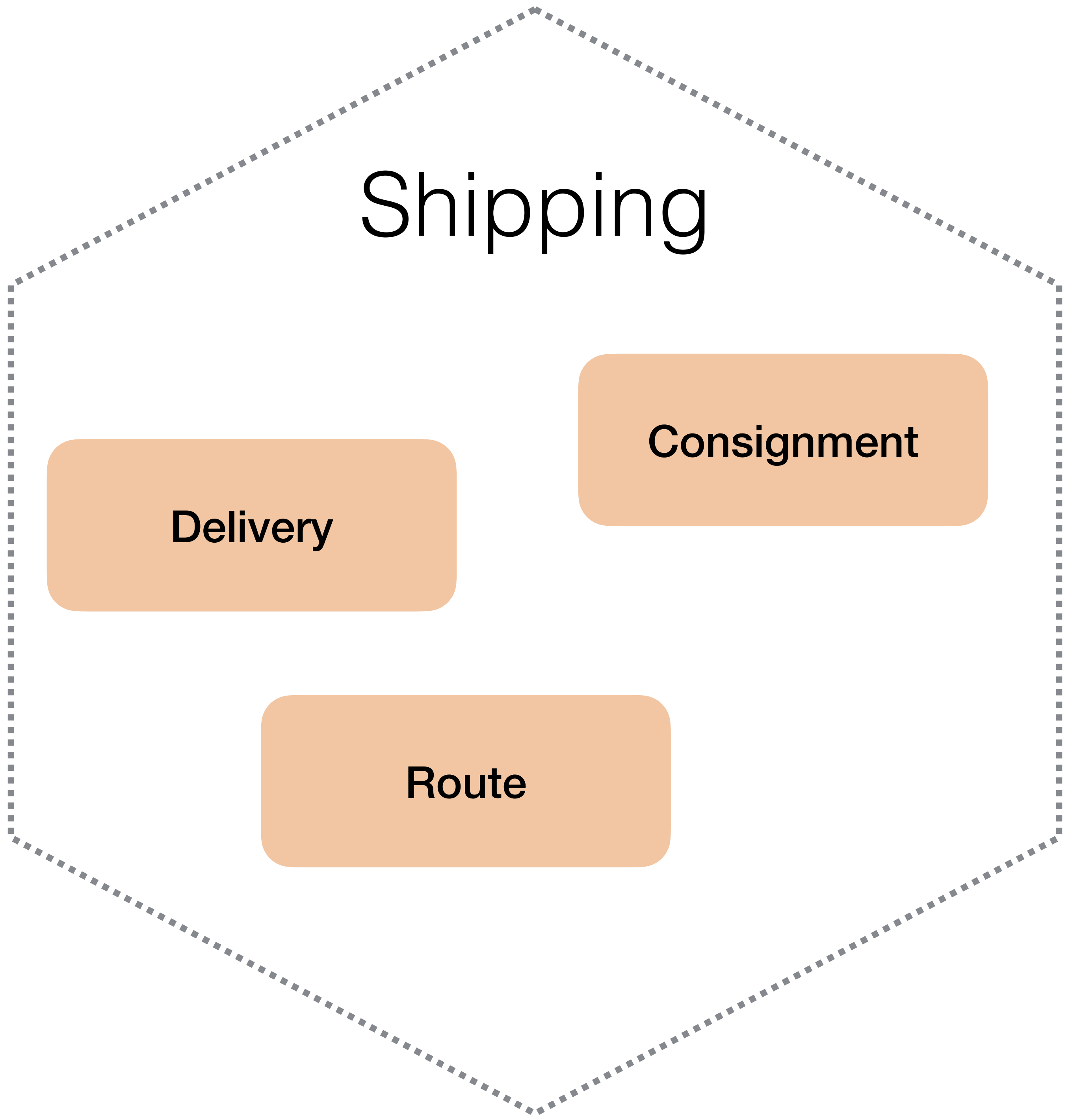
Delivery

Consignment

Route

HOW TO FIND THESE FUNCTION BOUNDARIES?

Identify aggregates within your microservice

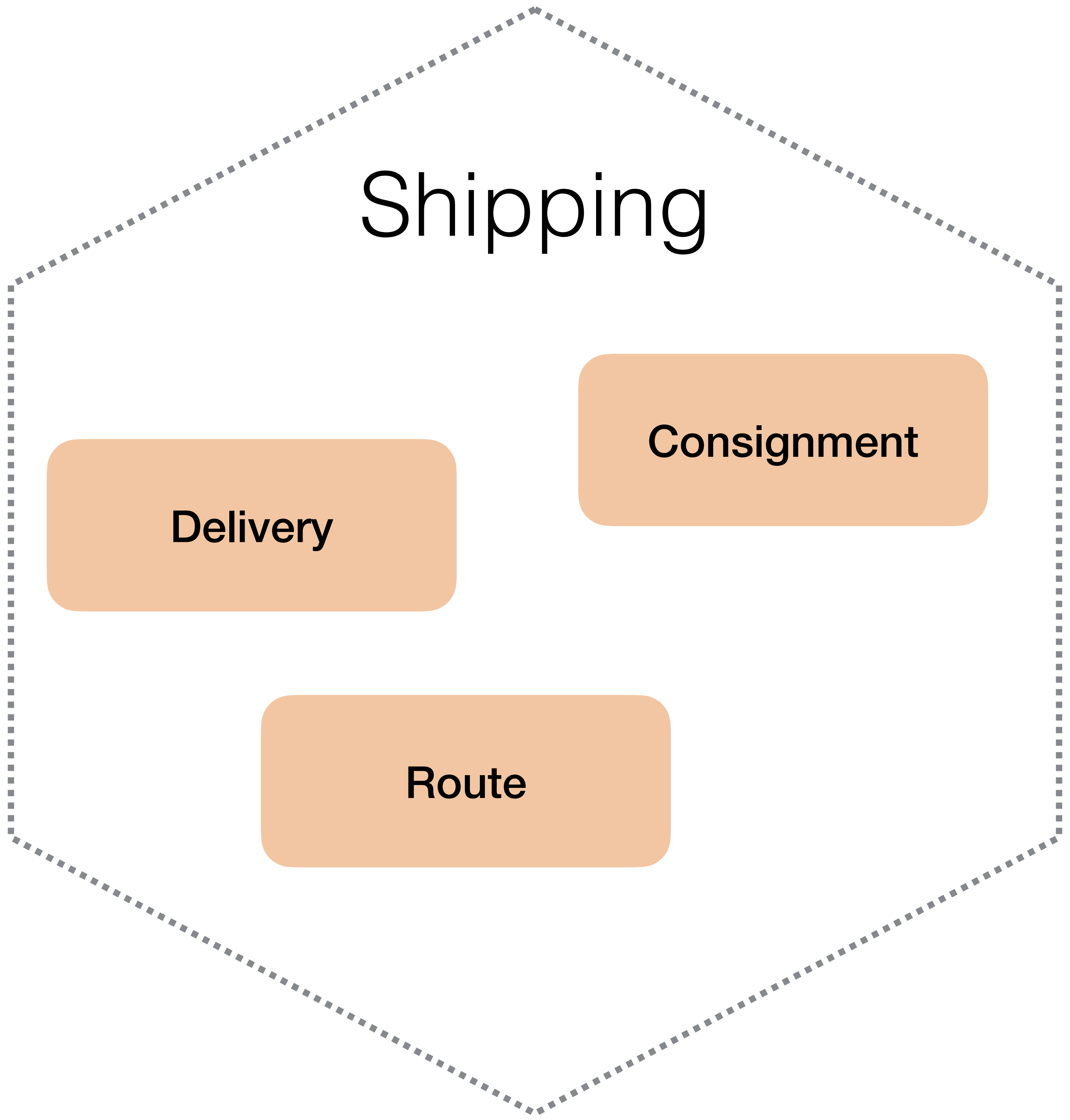


Map aggregates to functions

@samnewman

HOW TO FIND THESE FUNCTION BOUNDARIES?

Identify aggregates within your microservice



One function can manage all operations of the aggregate

Map aggregates to functions

https://www.flickr.com/photos/oatsy40/7617662386/

AGGREGATES

AGGREGATES

“A cluster of associated objects that are treated as a unit for the purpose of data changes”

- Eric Evans, Domain-Driven Design

AGGREGATES

“A cluster of associated objects that are treated as a unit for the purpose of data changes”

- Eric Evans, Domain-Driven Design

We want to manage an aggregate as a single “entity” in terms of state changes

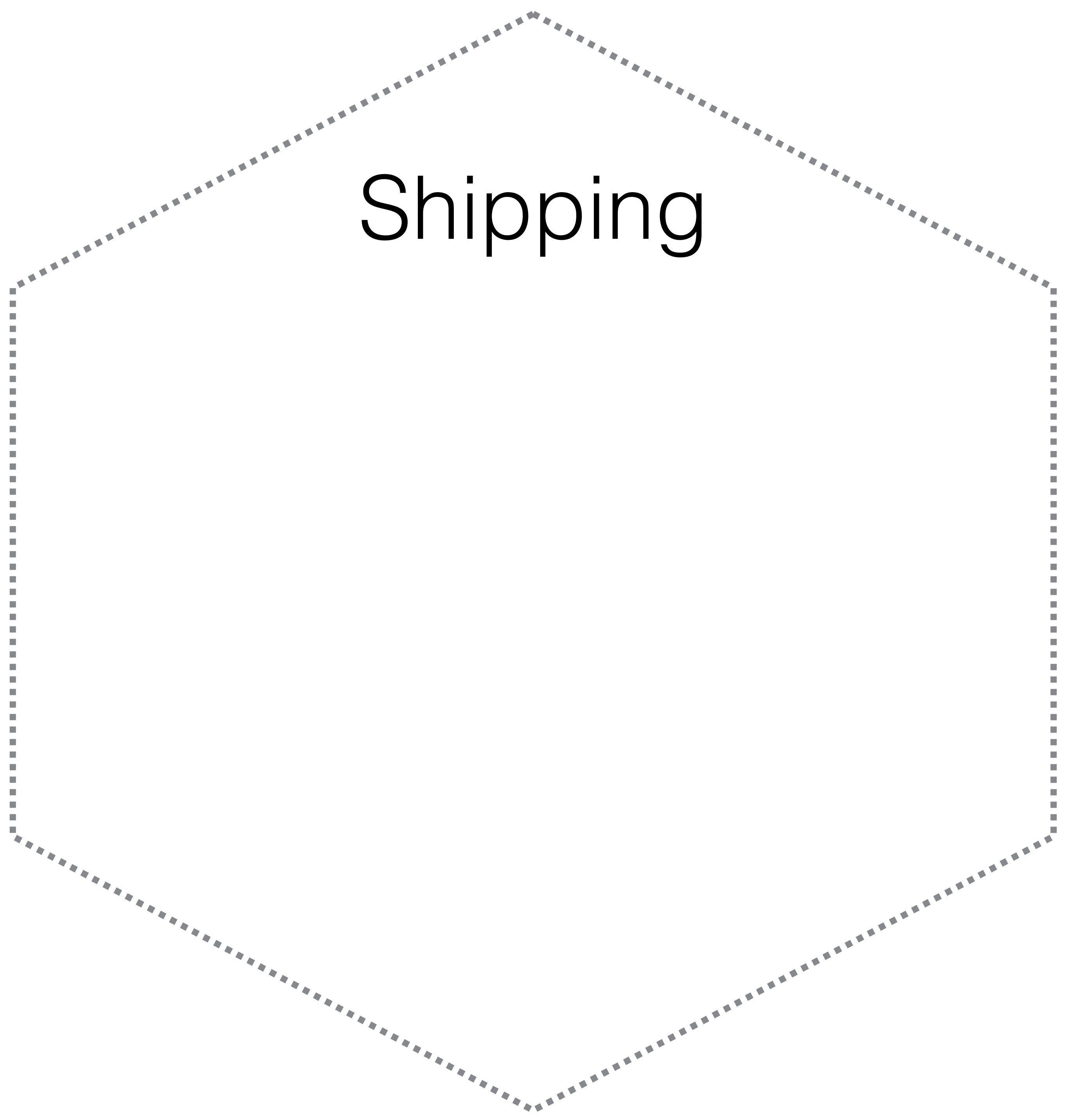
AGGREGATES

> “A cluster of associated objects that are treated as a unit for the purpose of data changes”
>
> *- Eric Evans, Domain-Driven Design*

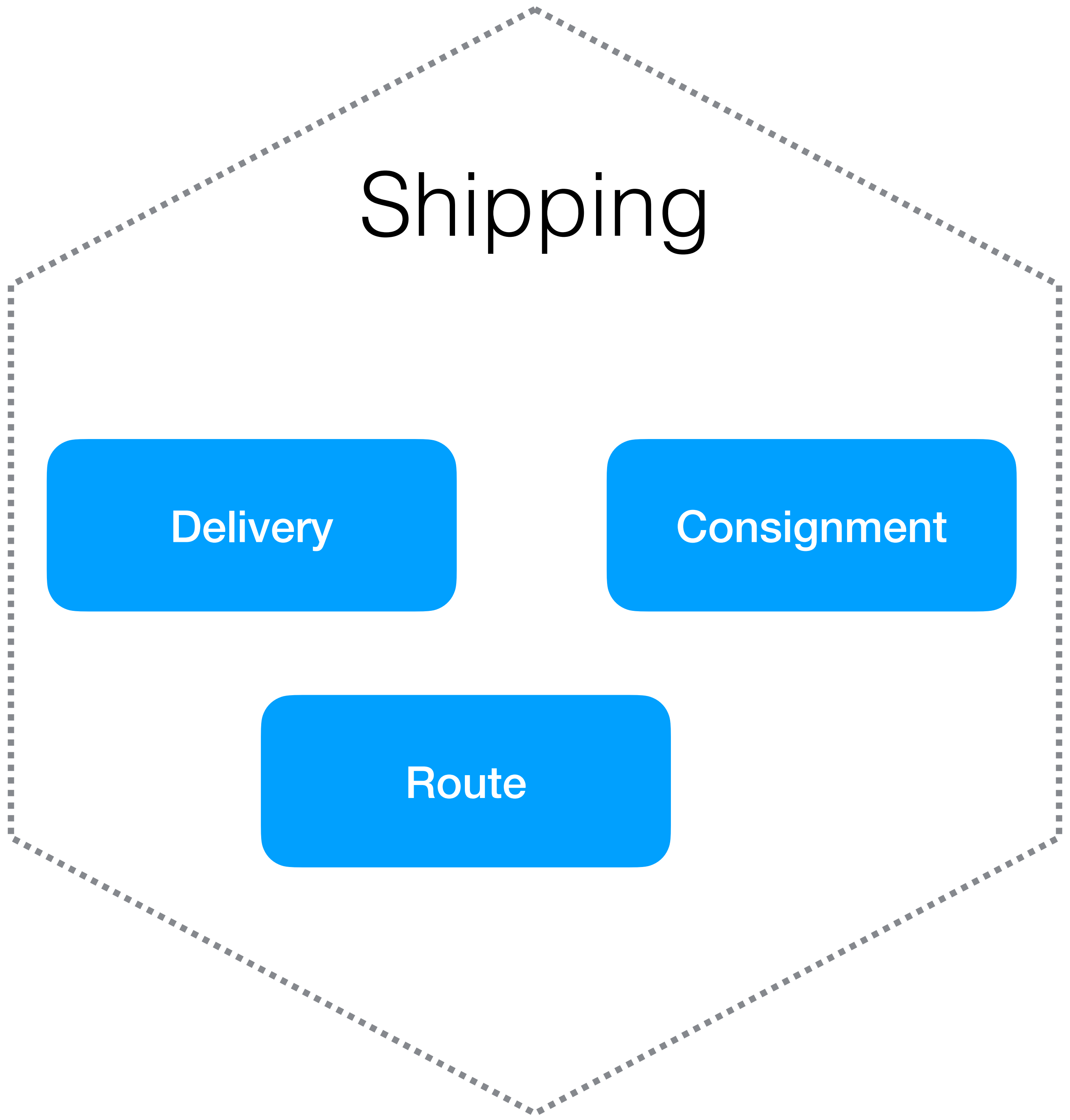
We want to manage an aggregate as a single “entity” in terms of state changes

We want all operations which manage the state to behave in a consistent fashion

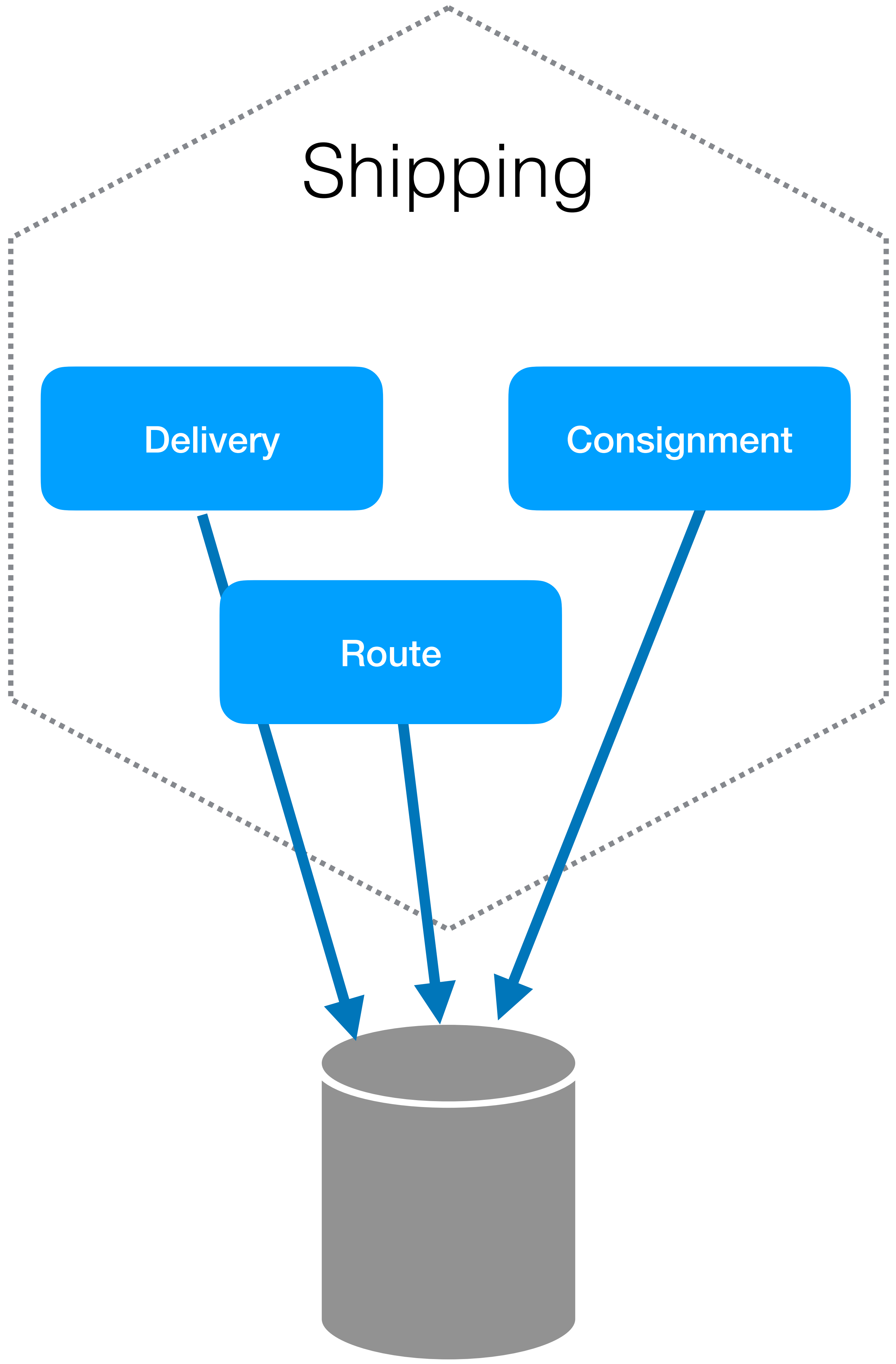
AGGREGATE MANAGEMENT AS A FUNCTION



AGGREGATE MANAGEMENT AS A FUNCTION



AGGREGATE MANAGEMENT AS A FUNCTION



@samnewman

WHAT ABOUT MULTIPLE FUNCTIONS FOR ONE AGGREGATE?

WHAT ABOUT MULTIPLE FUNCTIONS FOR ONE AGGREGATE?

In general, avoid!

WHAT ABOUT MULTIPLE FUNCTIONS FOR ONE AGGREGATE?

In general, avoid!

(I used to suggest this - I changed my mind)

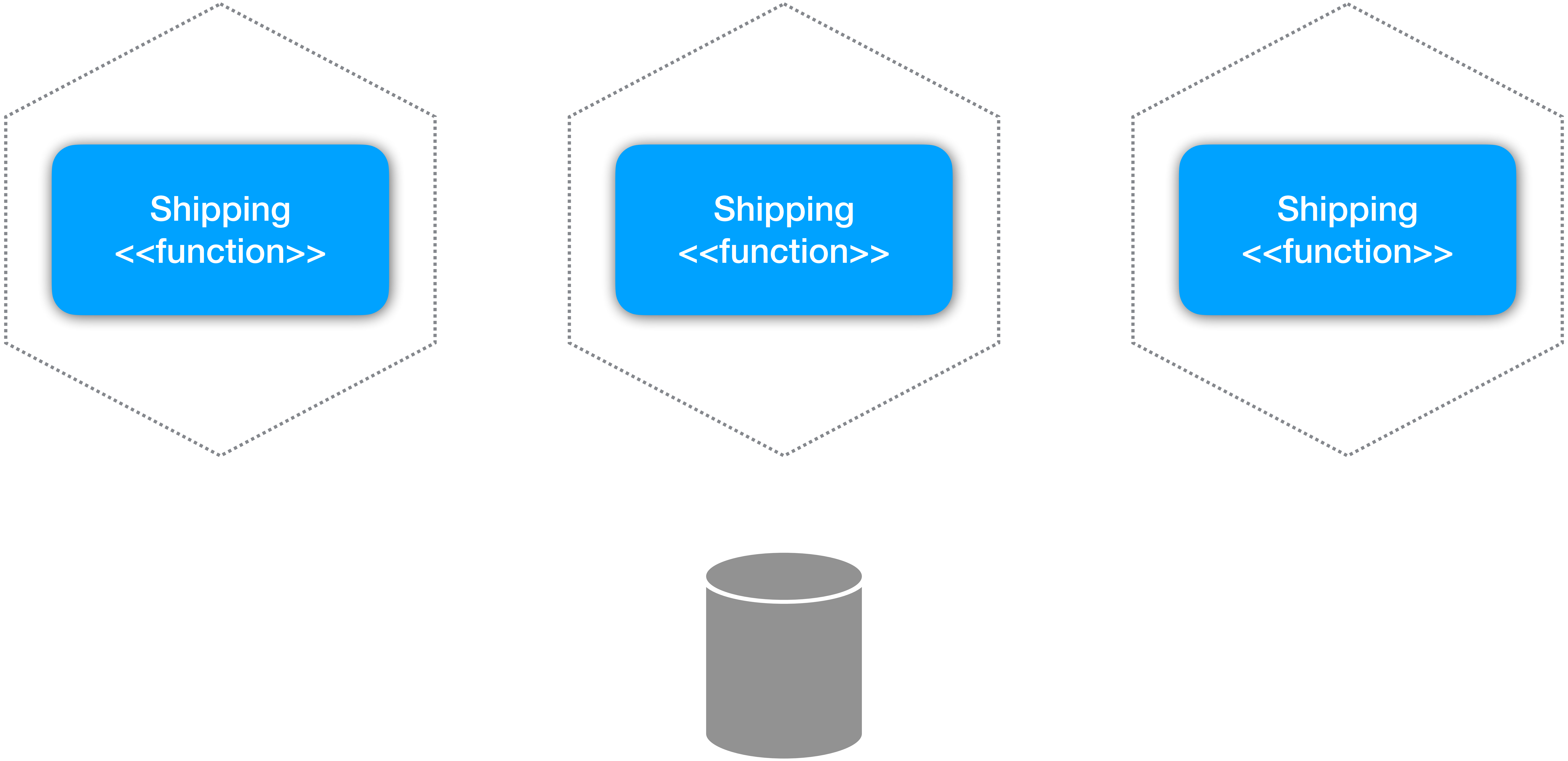
WHAT ABOUT MULTIPLE FUNCTIONS FOR ONE AGGREGATE?

In general, avoid!

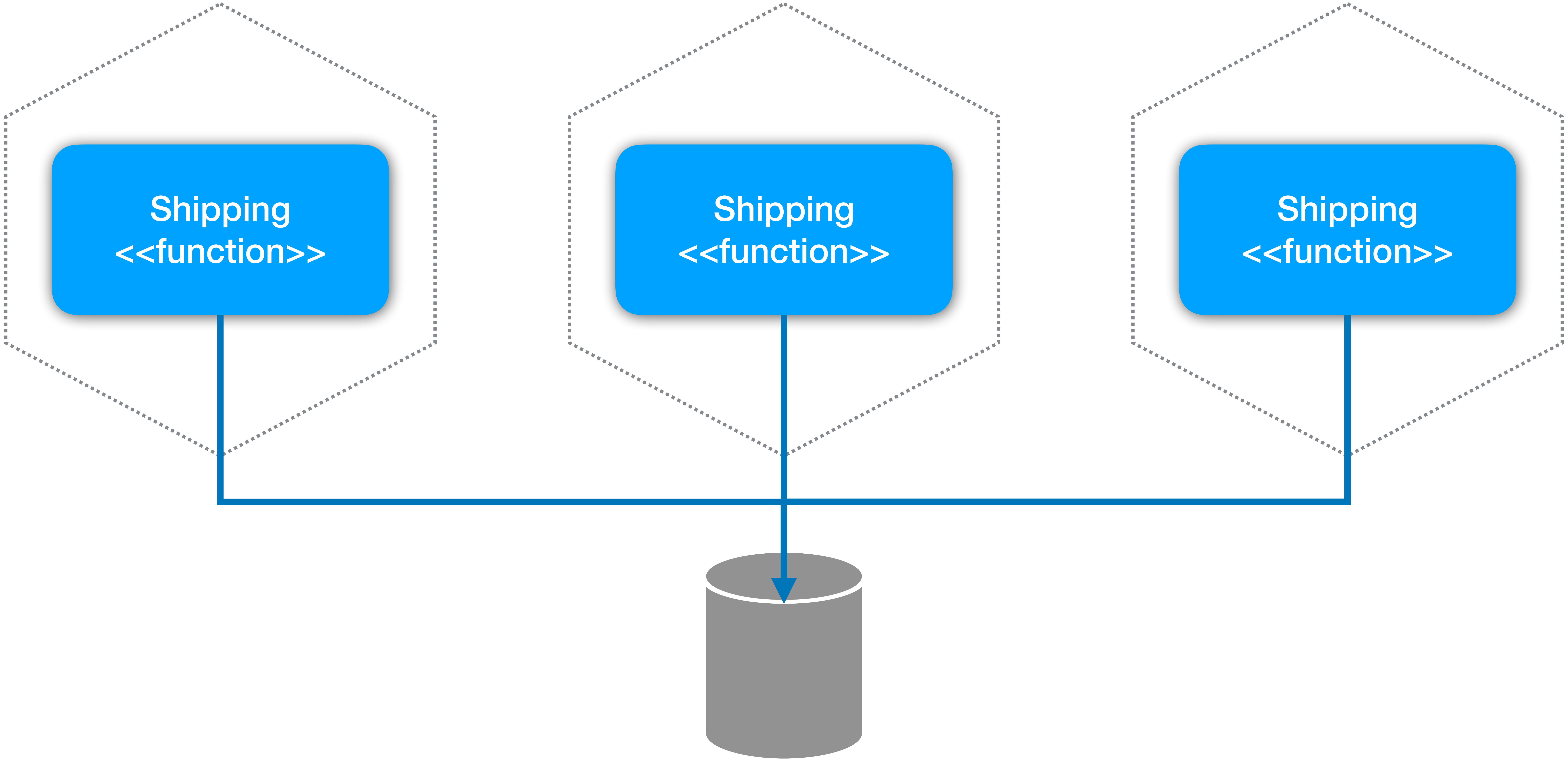
(I used to suggest this - I changed my mind)

If we want to manage the aggregate as a single unit, it’s easier to keep it as a single deployment

MORE SERVERLESS?

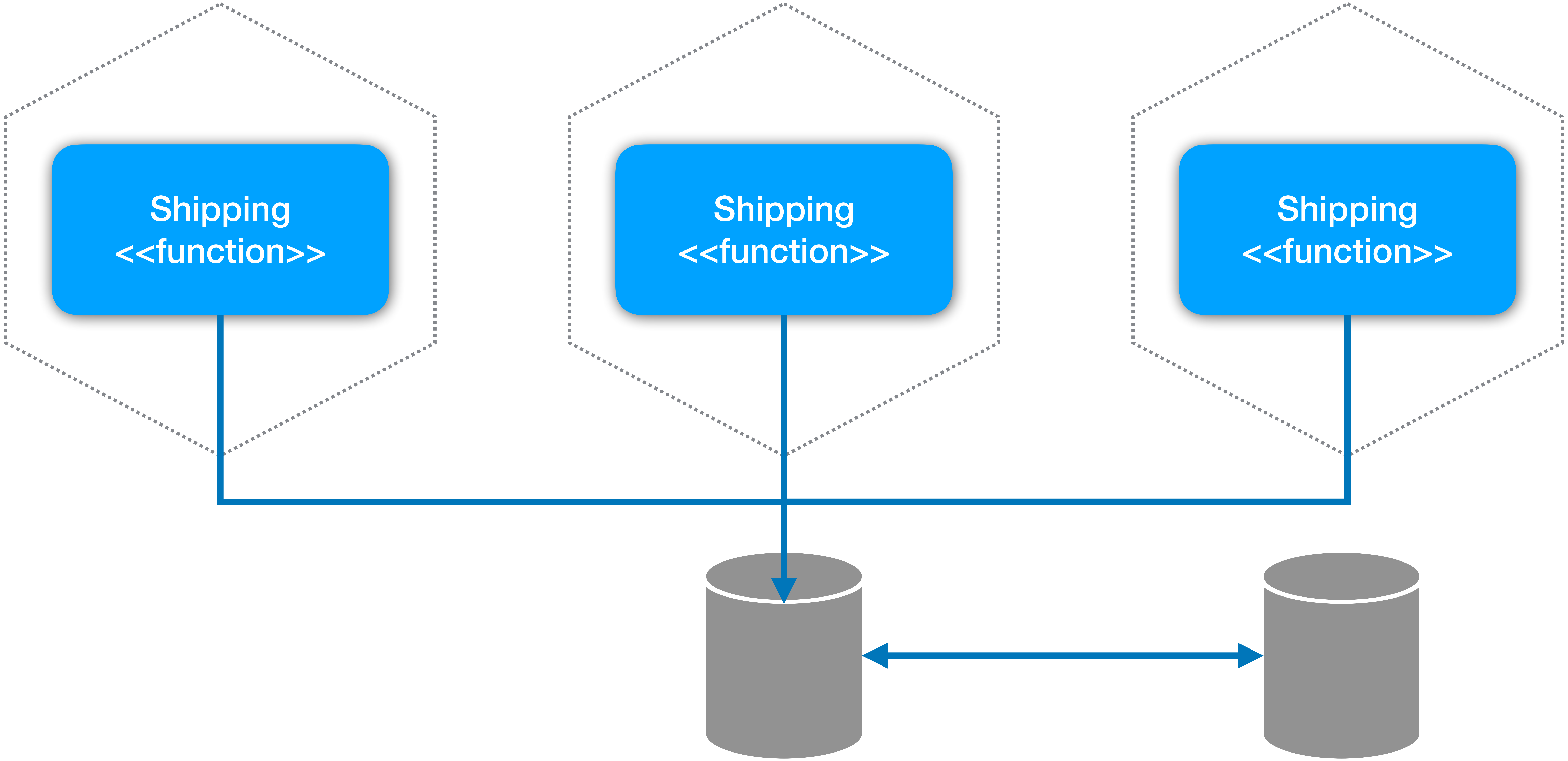


MORE SERVERLESS?



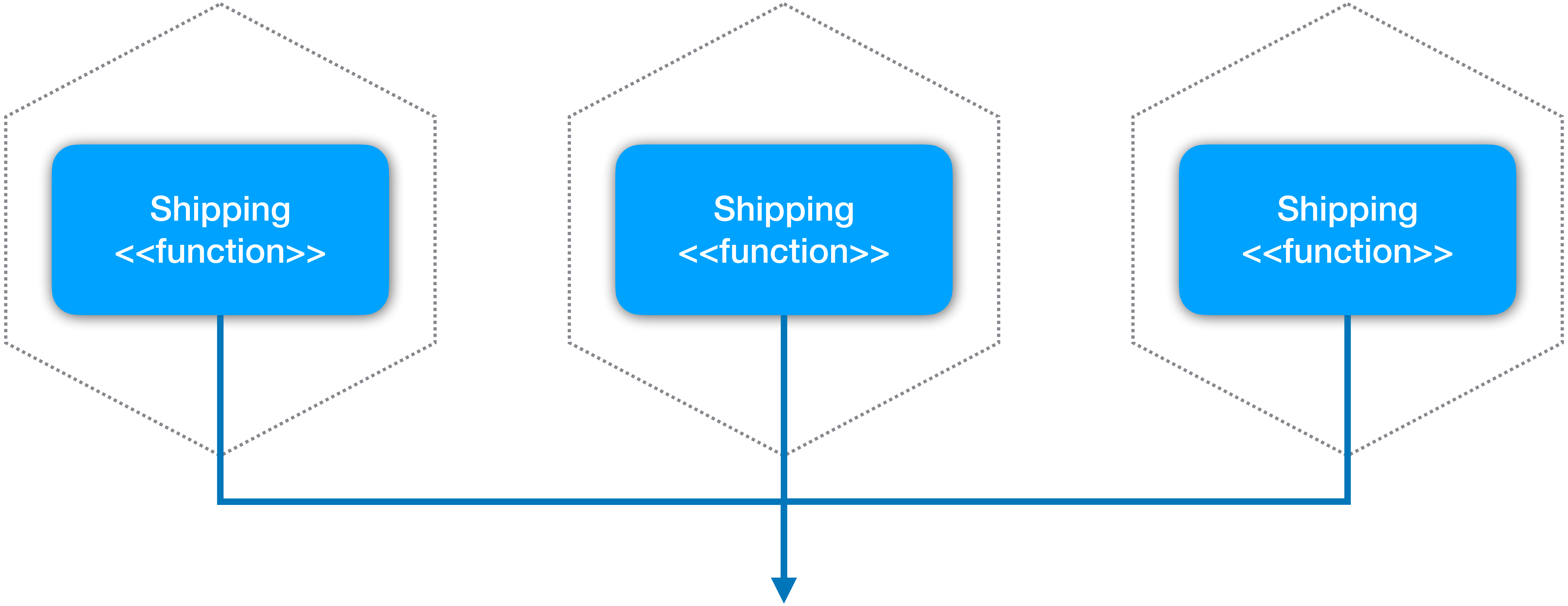
@samnewman

MORE SERVERLESS?



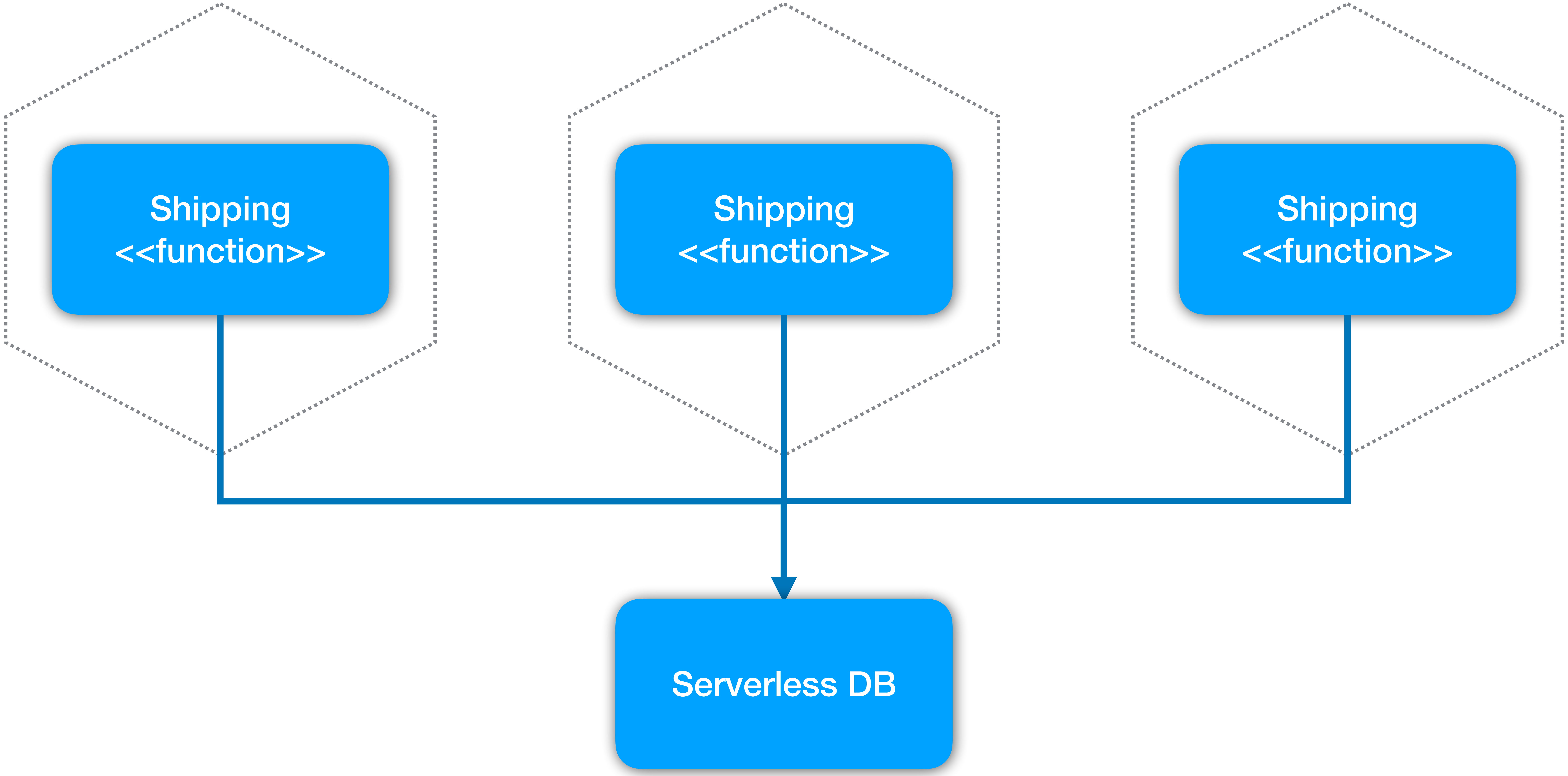
@samnewman

MORE SERVERLESS?

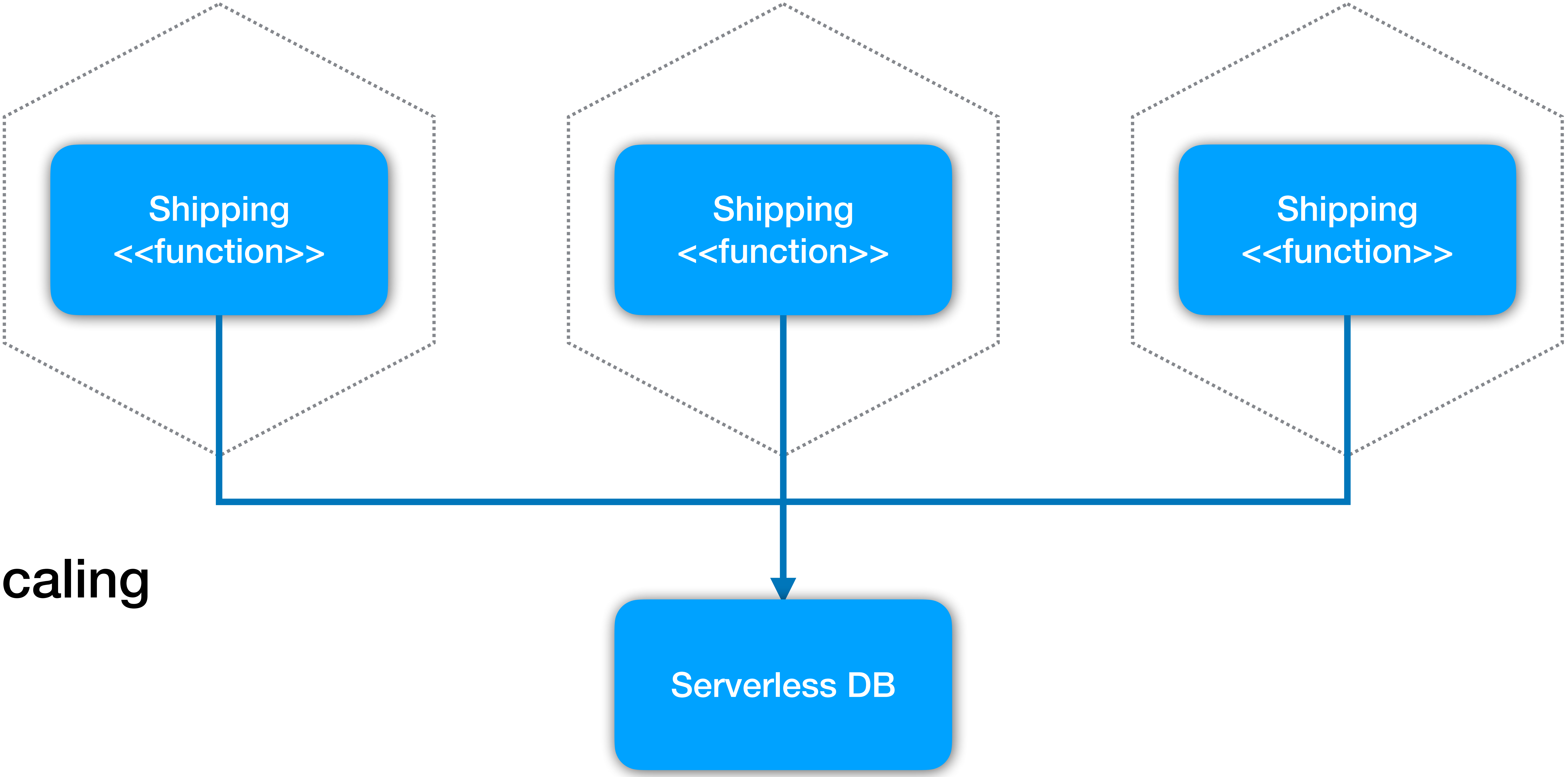


@samnewman

MORE SERVERLESS?

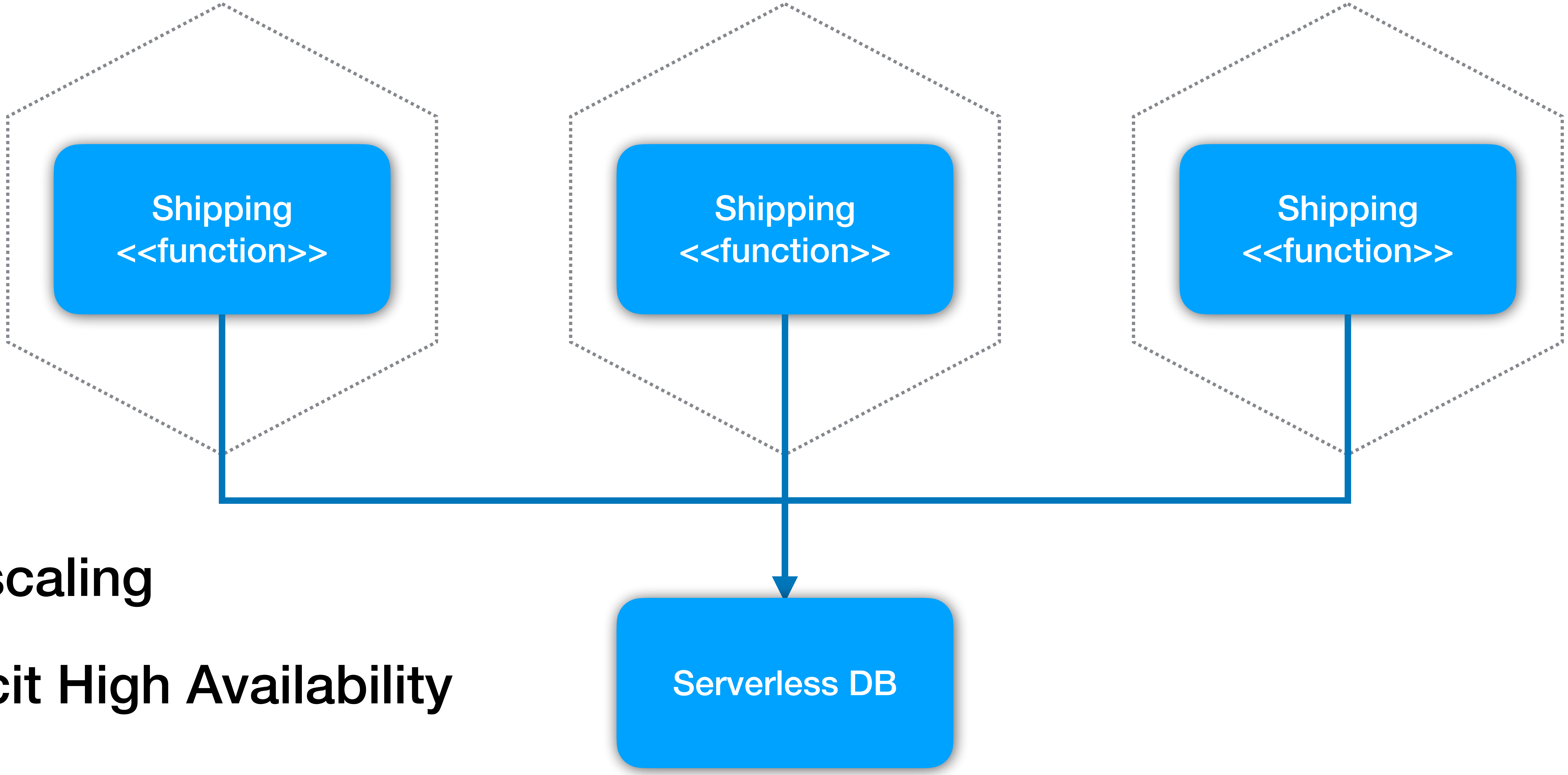


MORE SERVERLESS?



Autoscaling

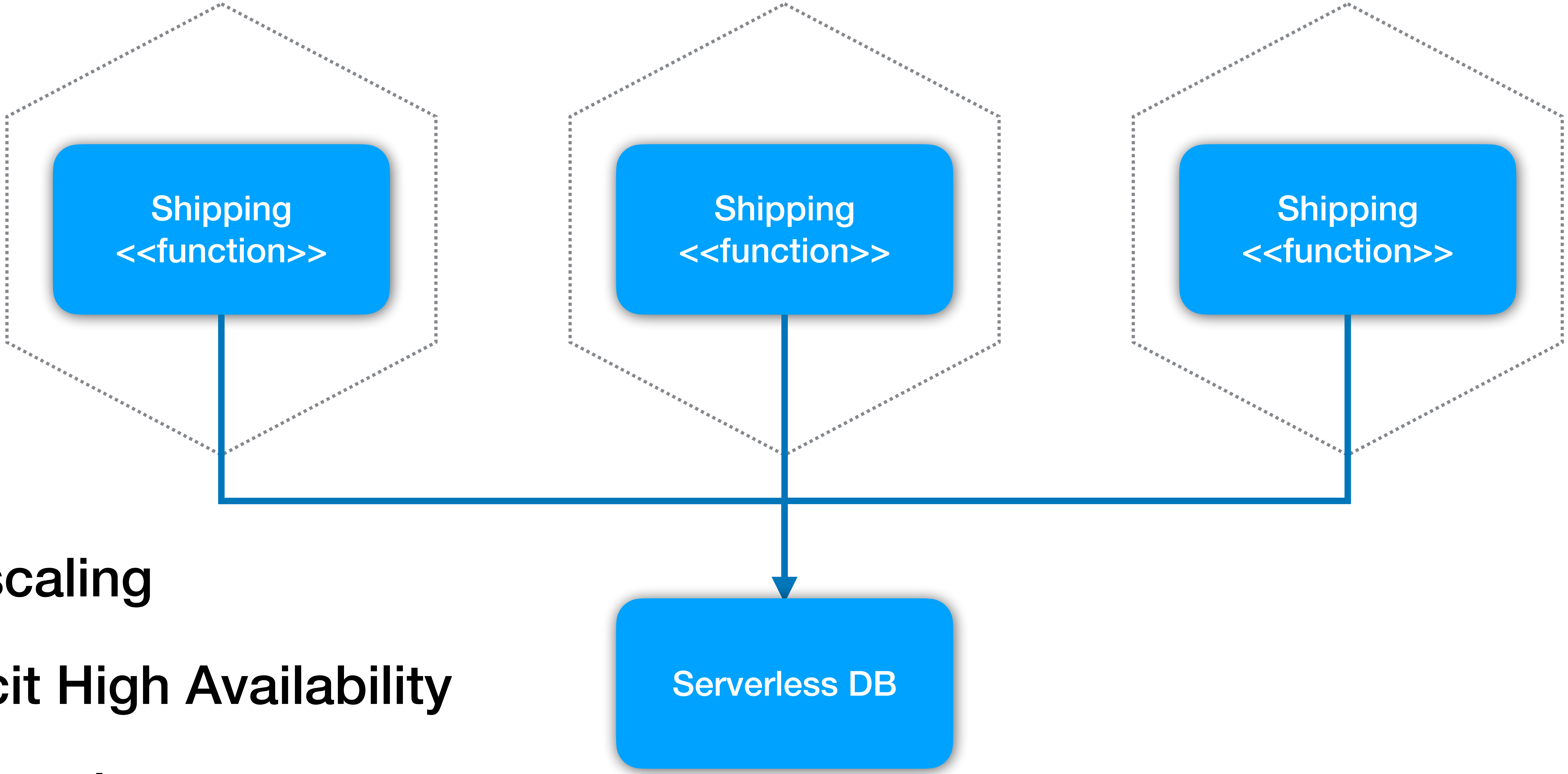
MORE SERVERLESS?



Autoscaling

Implicit High Availability

MORE SERVERLESS?



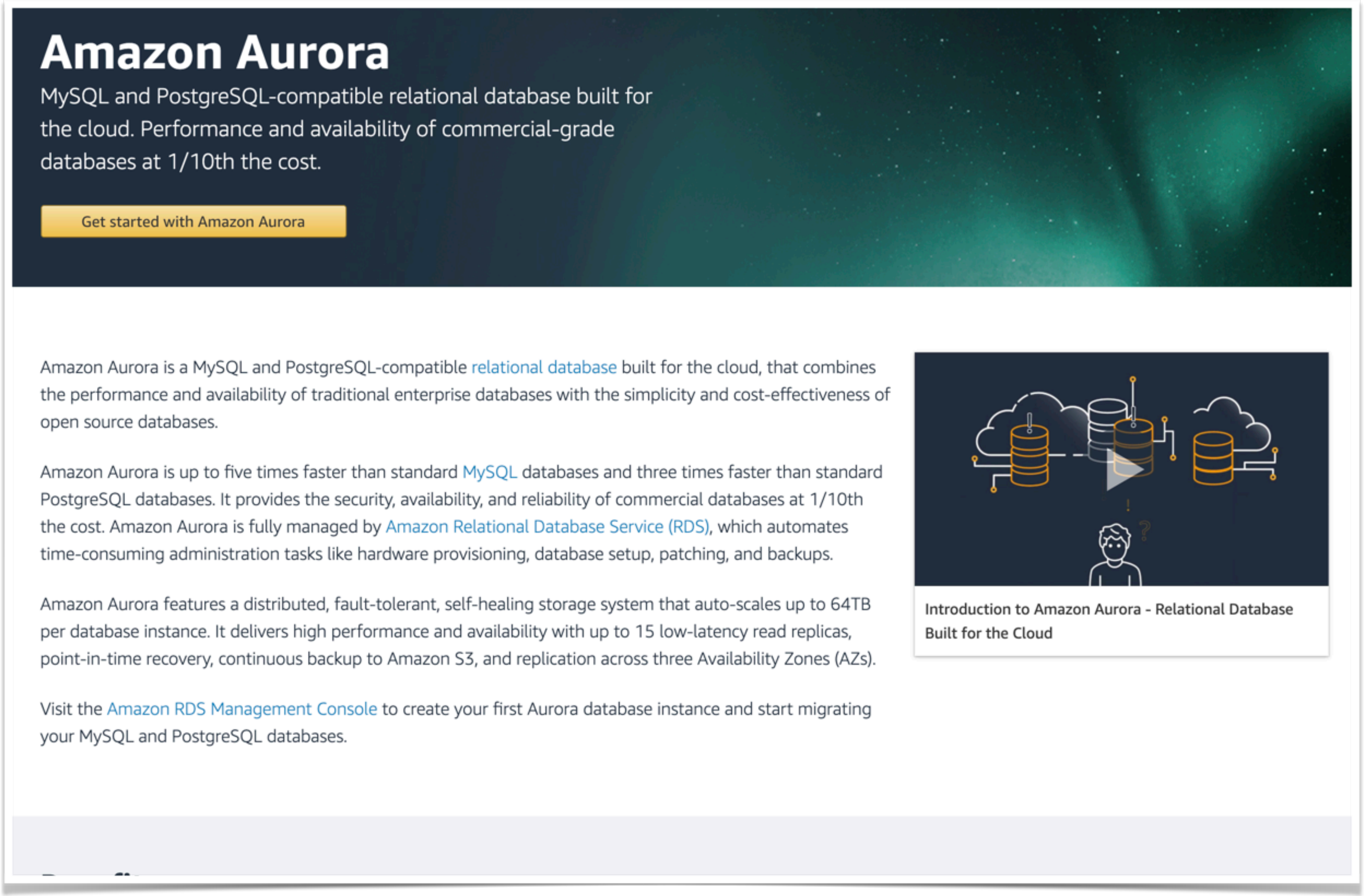
Autoscaling

Implicit High Availability

Pay for what you use

Extreme Caution!

Serverless databases can be different to what you are used to



Amazon Aurora

MySQL and PostgreSQL-compatible relational database built for the cloud. Performance and availability of commercial-grade databases at 1/10th the cost.

Amazon Aurora is a MySQL and PostgreSQL-compatible relational database built for the cloud, that combines the performance and availability of traditional enterprise databases with the simplicity and cost-effectiveness of open source databases.

Amazon Aurora is up to five times faster than standard MySQL databases and three times faster than standard PostgreSQL databases. It provides the security, availability, and reliability of commercial databases at 1/10th the cost. Amazon Aurora is fully managed by Amazon Relational Database Service (RDS), which automates time-consuming administration tasks like hardware provisioning, database setup, patching, and backups.

Amazon Aurora features a distributed, fault-tolerant, self-healing storage system that auto-scales up to 64TB per database instance. It delivers high performance and availability with up to 15 low-latency read replicas, point-in-time recovery, continuous backup to Amazon S3, and replication across three Availability Zones (AZs).

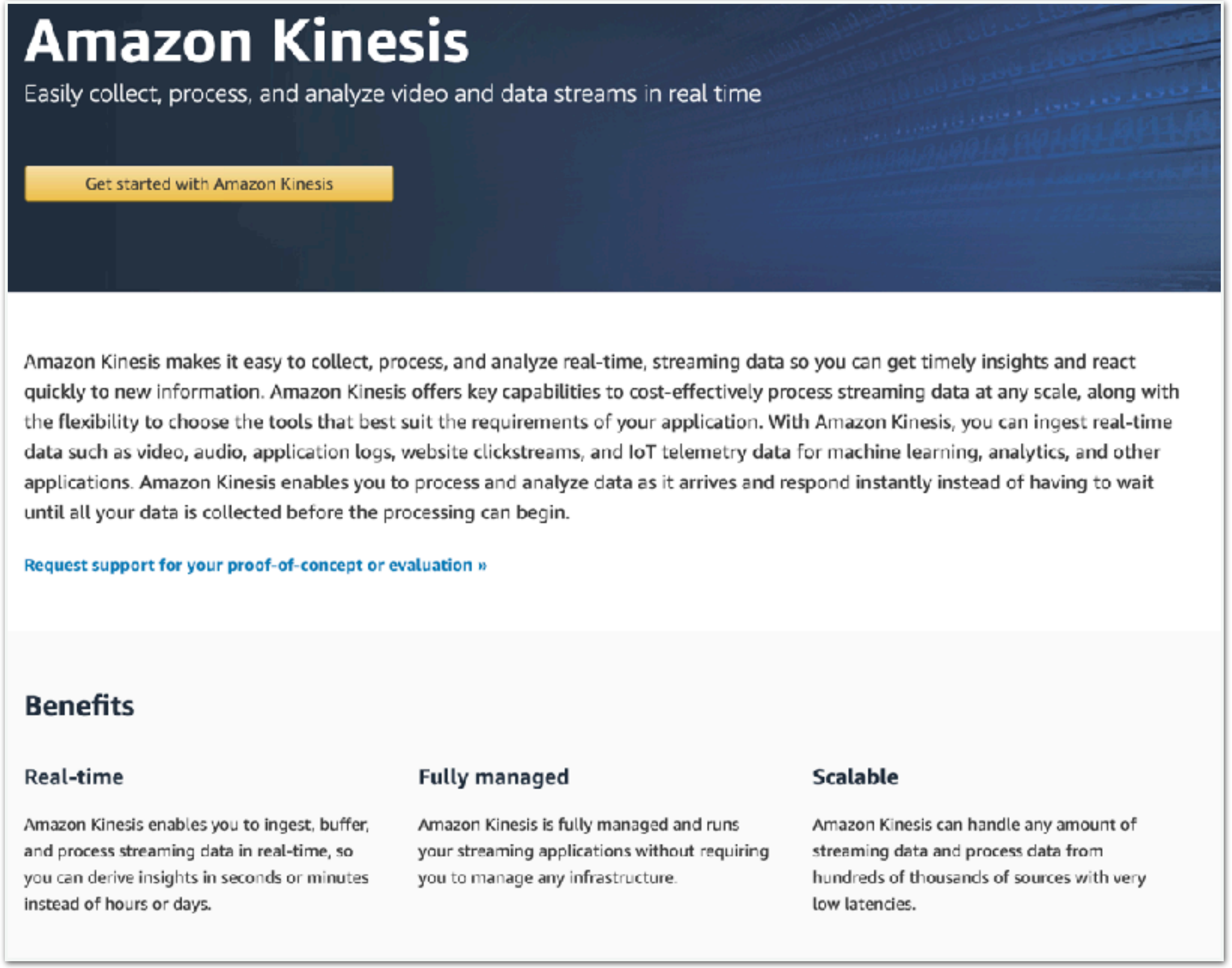
Visit the Amazon RDS Management Console to create your first Aurora database instance and start migrating your MySQL and PostgreSQL databases.

Introduction to Amazon Aurora - Relational Database Built for the Cloud

https://aws.amazon.com/rds/aurora/

MESSAGING

MESSAGING



Amazon Kinesis

Easily collect, process, and analyze video and data streams in real time

Amazon Kinesis makes it easy to collect, process, and analyze real-time, streaming data so you can get timely insights and react quickly to new information. Amazon Kinesis offers key capabilities to cost-effectively process streaming data at any scale, along with the flexibility to choose the tools that best suit the requirements of your application. With Amazon Kinesis, you can ingest real-time data such as video, audio, application logs, website clickstreams, and IoT telemetry data for machine learning, analytics, and other applications. Amazon Kinesis enables you to process and analyze data as it arrives and respond instantly instead of having to wait until all your data is collected before the processing can begin.

Request support for your proof-of-concept or evaluation »

Benefits

Real-time

Amazon Kinesis enables you to ingest, buffer, and process streaming data in real-time, so you can derive insights in seconds or minutes instead of hours or days.

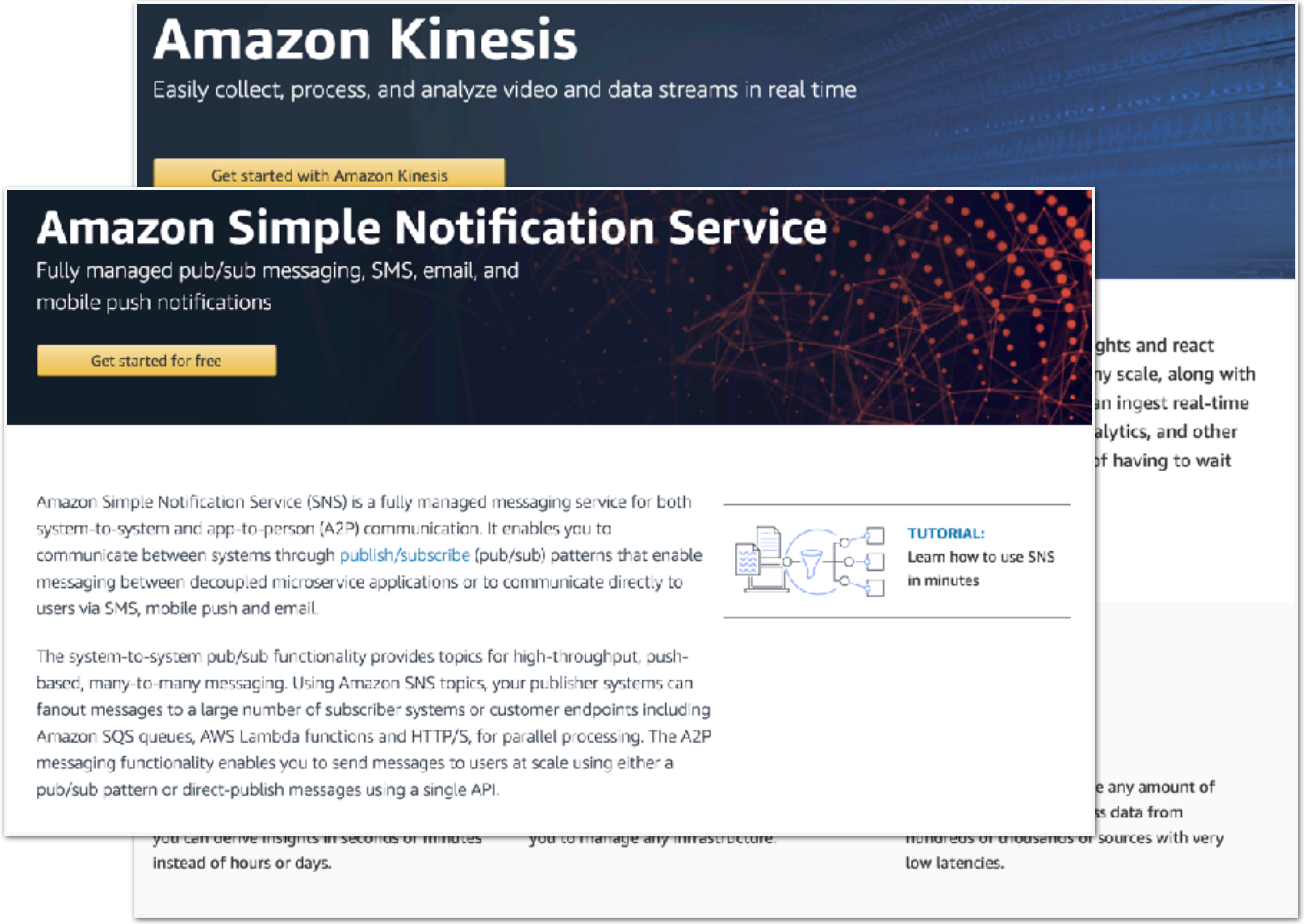
Fully managed

Amazon Kinesis is fully managed and runs your streaming applications without requiring you to manage any infrastructure.

Scalable

Amazon Kinesis can handle any amount of streaming data and process data from hundreds of thousands of sources with very low latencies.

MESSAGING



Amazon Kinesis

Easily collect, process, and analyze video and data streams in real time

ghts and react
ny scale, along with
an ingest real-time
alytics, and other
of having to wait

e any amount of
s data from
hundreds of thousands of sources with very
low latencies.

you can derive insights in seconds or minutes
instead of hours or days.

you to manage any infrastructure.

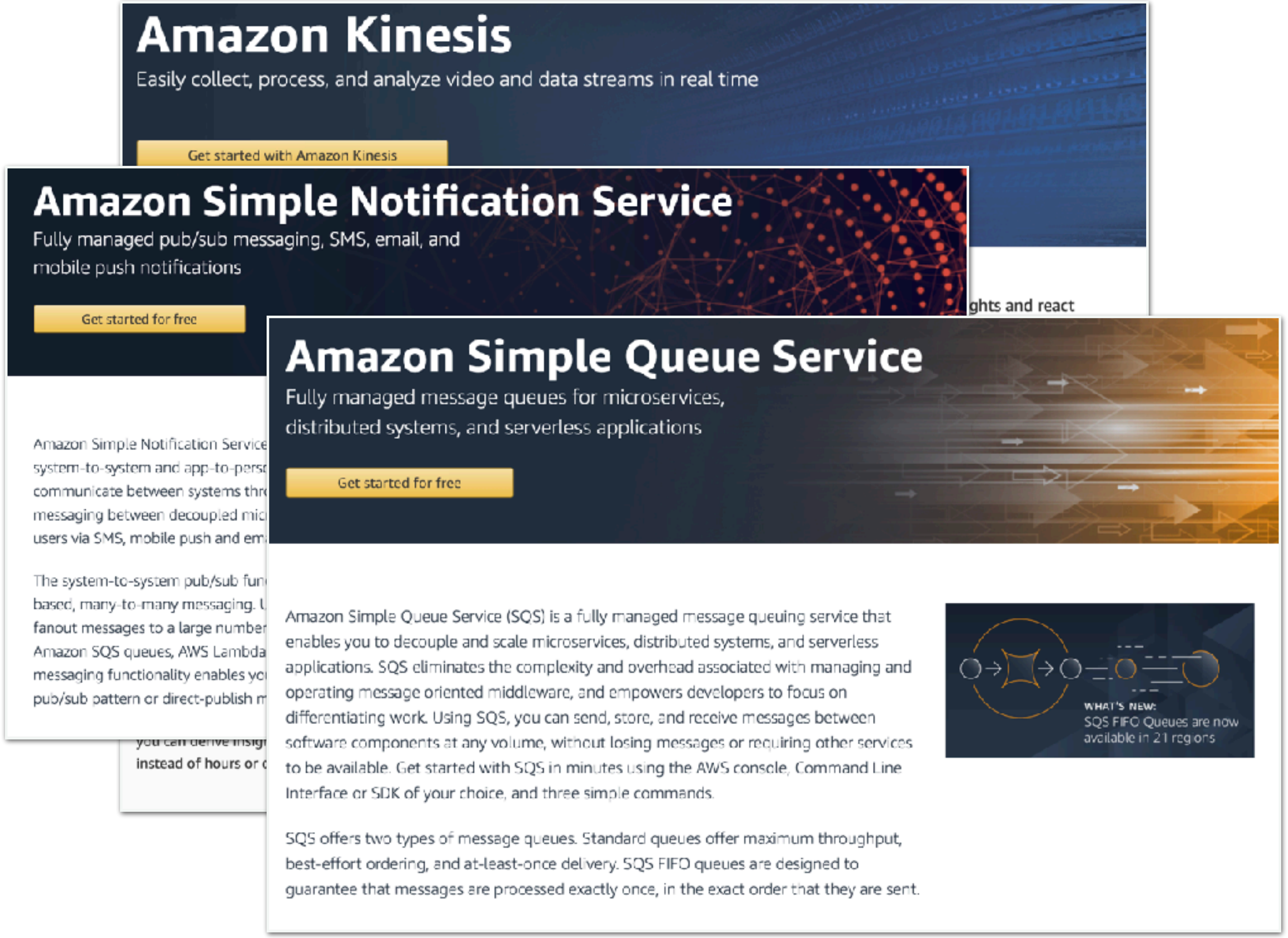
Amazon Simple Notification Service

Fully managed pub/sub messaging, SMS, email, and mobile push notifications

Amazon Simple Notification Service (SNS) is a fully managed messaging service for both system-to-system and app-to-person (A2P) communication. It enables you to communicate between systems through publish/subscribe (pub/sub) patterns that enable messaging between decoupled microservice applications or to communicate directly to users via SMS, mobile push and email.

The system-to-system pub/sub functionality provides topics for high-throughput, push-based, many-to-many messaging. Using Amazon SNS topics, your publisher systems can fanout messages to a large number of subscriber systems or customer endpoints including Amazon SQS queues, AWS Lambda functions and HTTP/S, for parallel processing. The A2P messaging functionality enables you to send messages to users at scale using either a pub/sub pattern or direct-publish messages using a single API.

MESSAGING



Amazon Kinesis

Easily collect, process, and analyze video and data streams in real time

ghts and react

you can derive insig

instead of hours or

Amazon Simple Notification Service

Fully managed pub/sub messaging, SMS, email, and mobile push notifications

Amazon Simple Notification Service
system-to-system and app-to-pers
communicate between systems thr
messaging between decoupled mic
users via SMS, mobile push and em

The system-to-system pub/sub fun
based, many-to-many messaging. U
fanout messages to a large number
Amazon SQS queues, AWS Lambda
messaging functionality enables yo
pub/sub pattern or direct-publish m

Amazon Simple Queue Service

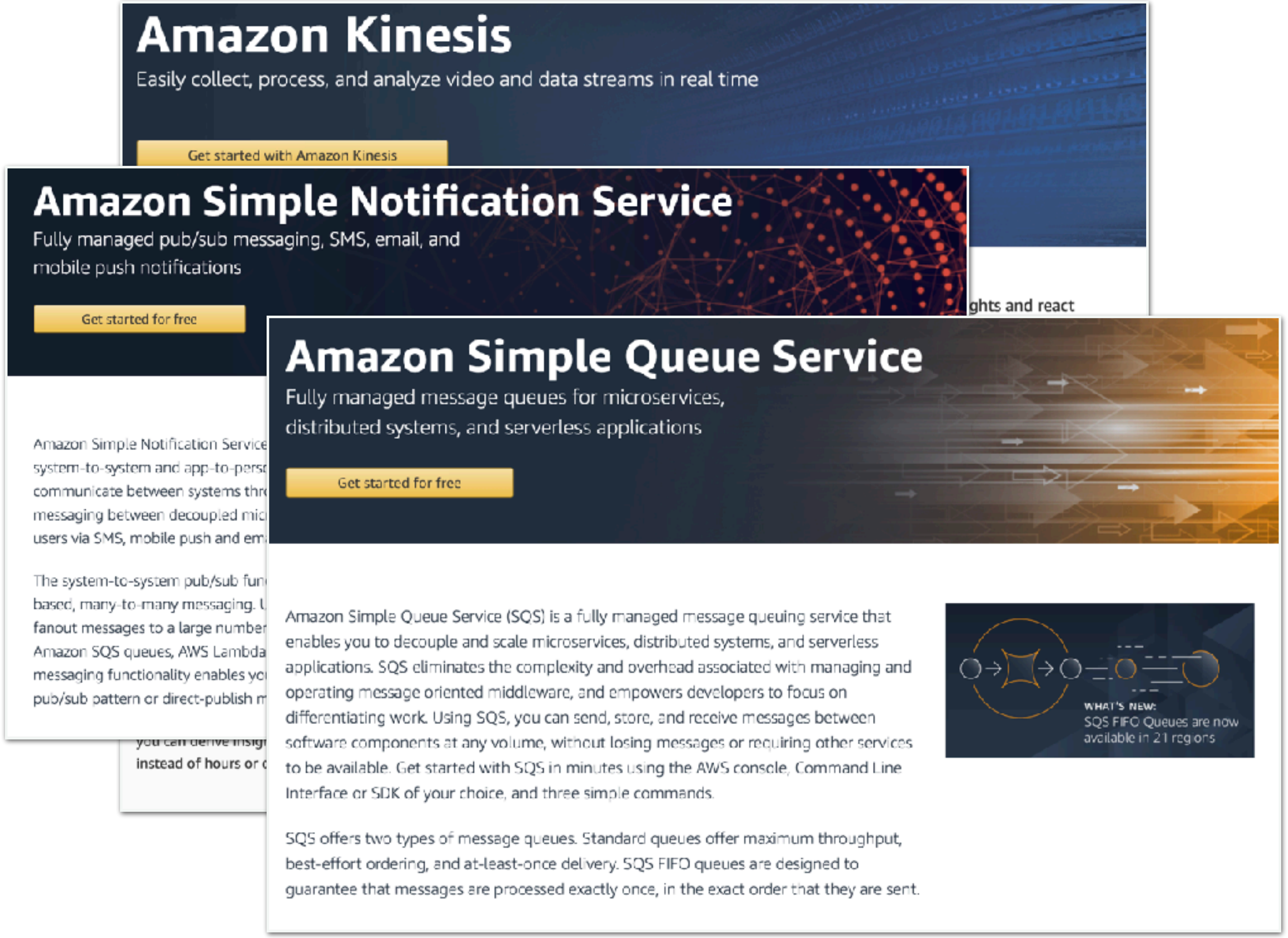
Fully managed message queues for microservices, distributed systems, and serverless applications

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS eliminates the complexity and overhead associated with managing and operating message oriented middleware, and empowers developers to focus on differentiating work. Using SQS, you can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available. Get started with SQS in minutes using the AWS console, Command Line Interface or SDK of your choice, and three simple commands.

SQS offers two types of message queues. Standard queues offer maximum throughput, best-effort ordering, and at-least-once delivery. SQS FIFO queues are designed to guarantee that messages are processed exactly once, in the exact order that they are sent.

WHAT'S NEW:
SQS FIFO Queues are now available in 21 regions

MESSAGING



Amazon Kinesis

Easily collect, process, and analyze video and data streams in real time

Get started with Amazon Kinesis

ghts and react

you can derive insig

instead of hours or d

Amazon Simple Notification Service

Fully managed pub/sub messaging, SMS, email, and mobile push notifications

Get started for free

Amazon Simple Notification Service
system-to-system and app-to-perso
communicate between systems thro
messaging between decoupled mic
users via SMS, mobile push and em

The system-to-system pub/sub fun
based, many-to-many messaging. U
fanout messages to a large number
Amazon SQS queues, AWS Lambda
messaging functionality enables yo
pub/sub pattern or direct-publish m

Amazon Simple Queue Service

Fully managed message queues for microservices, distributed systems, and serverless applications

Get started for free

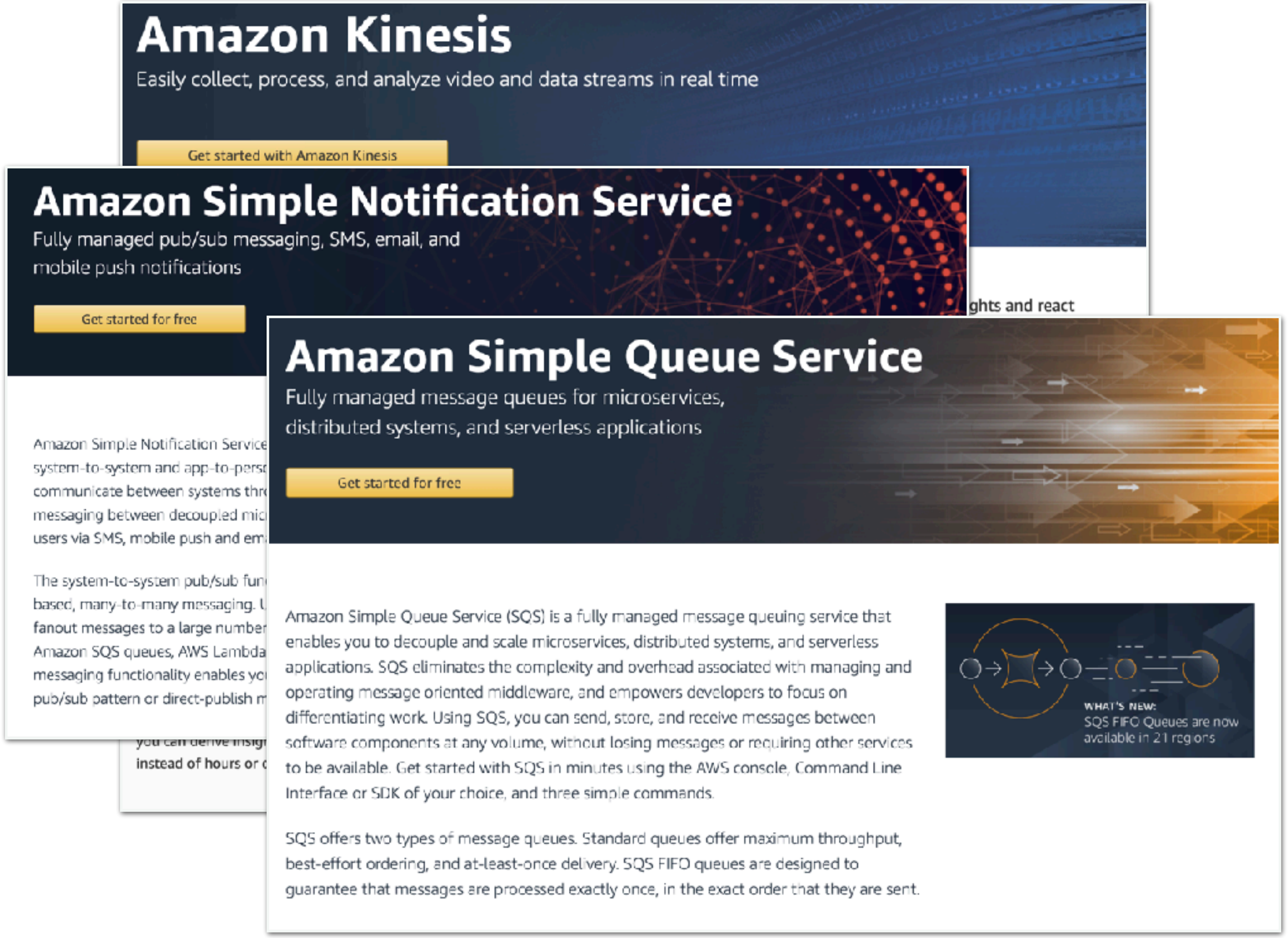
Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS eliminates the complexity and overhead associated with managing and operating message oriented middleware, and empowers developers to focus on differentiating work. Using SQS, you can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available. Get started with SQS in minutes using the AWS console, Command Line Interface or SDK of your choice, and three simple commands.

SQS offers two types of message queues. Standard queues offer maximum throughput, best-effort ordering, and at-least-once delivery. SQS FIFO queues are designed to guarantee that messages are processed exactly once, in the exact order that they are sent.

WHAT'S NEW:
SQS FIFO Queues are now available in 21 regions

Both AWS and Azure have lots of great options

MESSAGING



Amazon Kinesis

Easily collect, process, and analyze video and data streams in real time

ghts and react

you can derive insig

instead of hours or

Amazon Simple Notification Service

Fully managed pub/sub messaging, SMS, email, and mobile push notifications

Amazon Simple Notification Service
system-to-system and app-to-perso
communicate between systems thro
messaging between decoupled mic
users via SMS, mobile push and em

The system-to-system pub/sub fun
based, many-to-many messaging. U
fanout messages to a large number
Amazon SQS queues, AWS Lambda
messaging functionality enables yo
pub/sub pattern or direct-publish m

Amazon Simple Queue Service

Fully managed message queues for microservices, distributed systems, and serverless applications

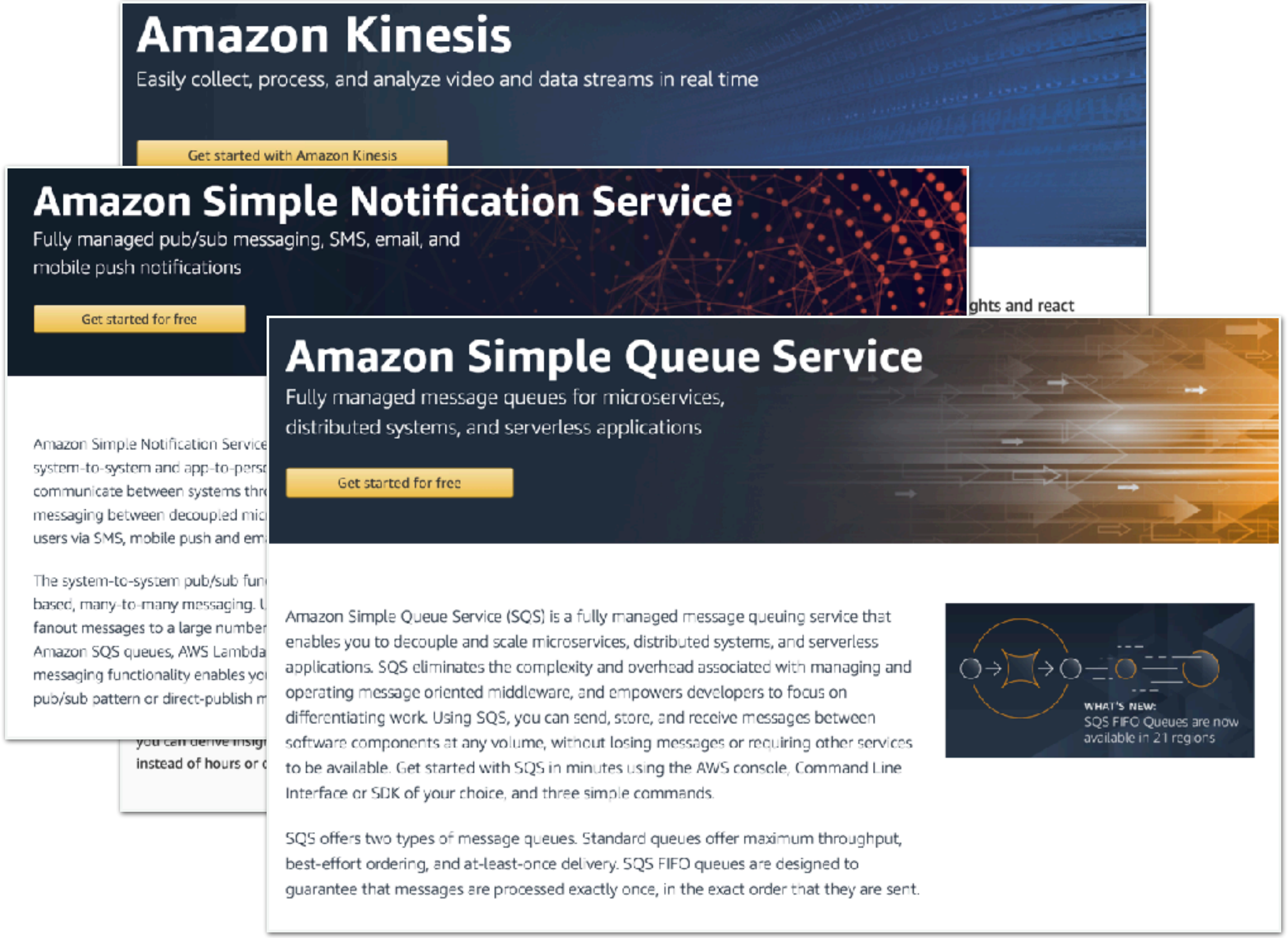
Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS eliminates the complexity and overhead associated with managing and operating message oriented middleware, and empowers developers to focus on differentiating work. Using SQS, you can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available. Get started with SQS in minutes using the AWS console, Command Line Interface or SDK of your choice, and three simple commands.

SQS offers two types of message queues. Standard queues offer maximum throughput, best-effort ordering, and at-least-once delivery. SQS FIFO queues are designed to guarantee that messages are processed exactly once, in the exact order that they are sent.

Both AWS and Azure have lots of great options

Again, behaviour can be different to what you're used to

MESSAGING



Amazon Kinesis

Easily collect, process, and analyze video and data streams in real time

Get started with Amazon Kinesis

ghts and react

you can derive insig

instead of hours or c

Amazon Simple Notification Service

Fully managed pub/sub messaging, SMS, email, and mobile push notifications

Get started for free

Amazon Simple Notification Service
system-to-system and app-to-perso
communicate between systems thro
messaging between decoupled mic
users via SMS, mobile push and em

The system-to-system pub/sub fun
based, many-to-many messaging. U
fanout messages to a large number
Amazon SQS queues, AWS Lambda
messaging functionality enables yo
pub/sub pattern or direct-publish m

Amazon Simple Queue Service

Fully managed message queues for microservices, distributed systems, and serverless applications

Get started for free

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS eliminates the complexity and overhead associated with managing and operating message oriented middleware, and empowers developers to focus on differentiating work. Using SQS, you can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available. Get started with SQS in minutes using the AWS console, Command Line Interface or SDK of your choice, and three simple commands.

SQS offers two types of message queues. Standard queues offer maximum throughput, best-effort ordering, and at-least-once delivery. SQS FIFO queues are designed to guarantee that messages are processed exactly once, in the exact order that they are sent.

WHAT'S NEW:
SQS FIFO Queues are now available in 21 regions

Both AWS and Azure have lots of great options

Again, behaviour can be different to what you're used to

For Kubernetes fans? Well…

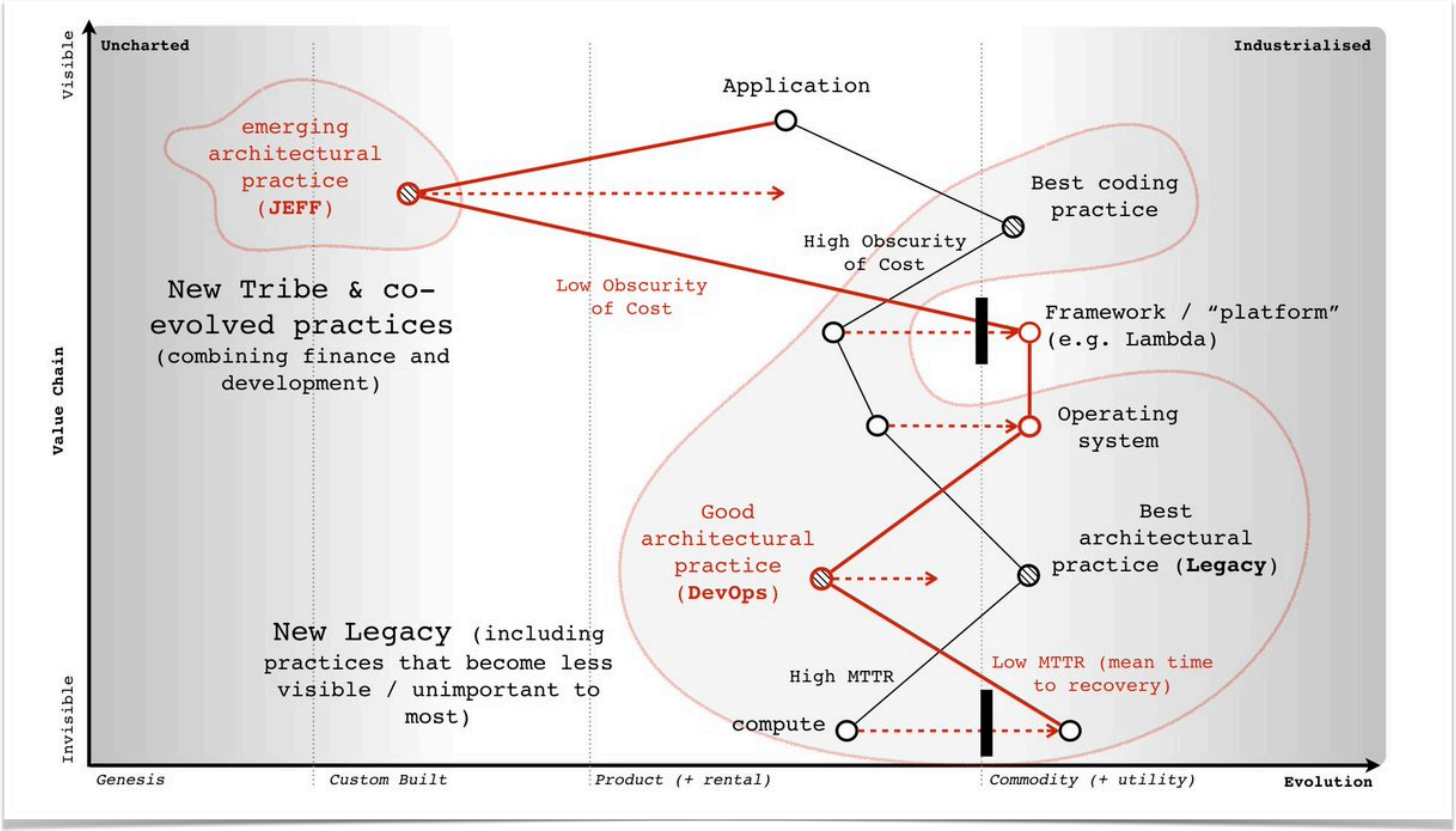
Part 3: What You Should Do About It

The secret to getting more things done, is to avoid doing things you don’t have to

Undifferentiated Heavy Lifting

Is the work you are doing making a difference?

Focus on what makes your product special,
outsource the rest



Uncharted
Industrialised
Visible
Value Chain
Invisible
Application
emerging architectural practice (JEFF)
Best coding practice
High Obscurity of Cost
Low Obscurity of Cost
New Tribe & co-evolved practices (combining finance and development)
Framework / "platform" (e.g. Lambda)
Operating system
Good architectural practice (DevOps)
Best architectural practice (Legacy)
New Legacy (including practices that become less visible / unimportant to most)
High MTTR
Low MTTR (mean time to recovery)
compute
Genesis
Custom Built
Product (+ rental)
Commodity (+ utility)
Evolution

https://twitter.com/swardley/status/914792448429297664

https://www.flickr.com/photos/t2thestreet/5892802537/

https://www.flickr.com/photos/jganderson/2933766374/

Ease into it

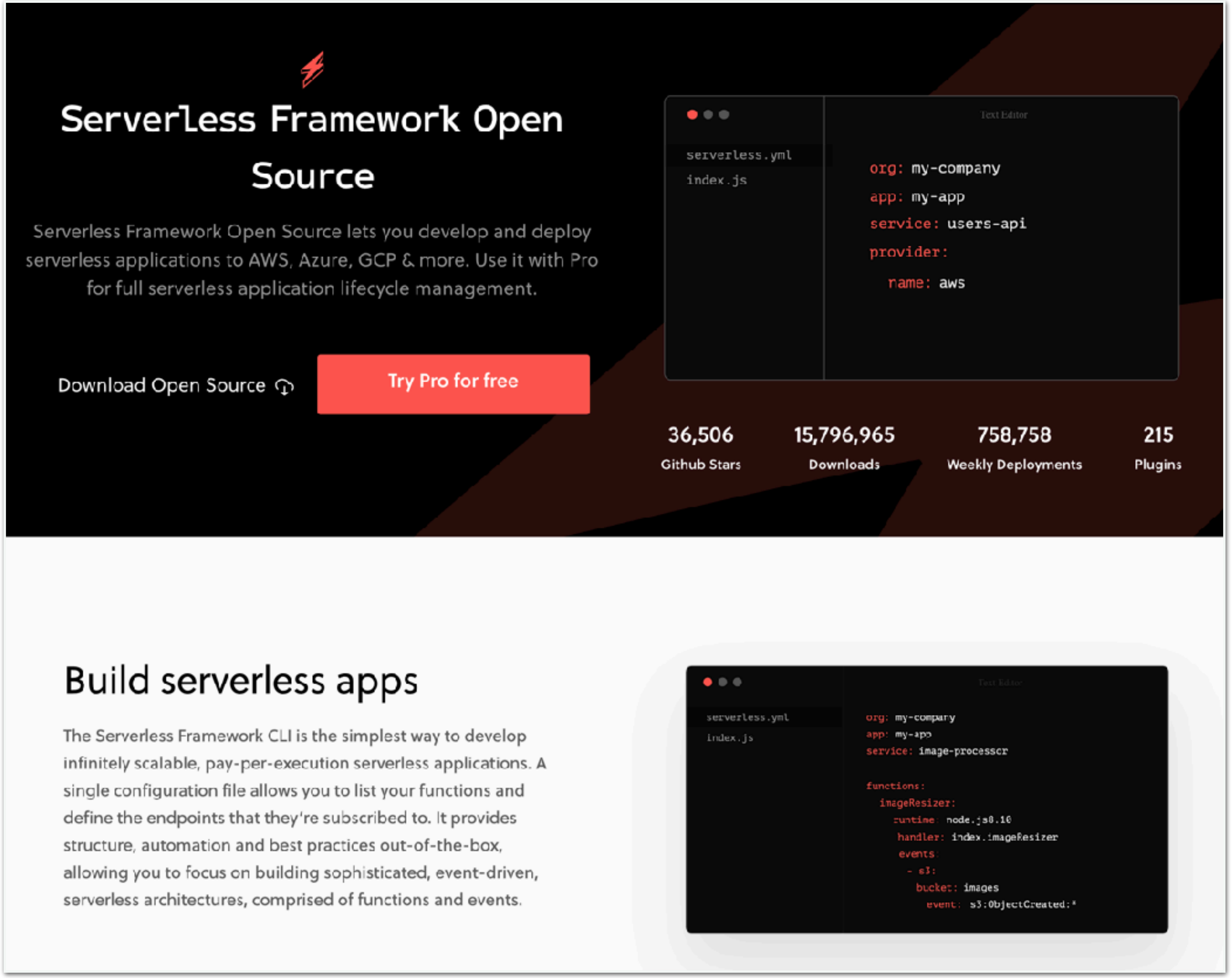
Map a whole microservice to a single function

Change happens one day at a time (hopefully)

You won’t know until you try

https://www.flickr.com/photos/dapuglet/33085627298/

THE SERVERLESS FRAMEWORK



Serverless Framework Open Source

Serverless Framework Open Source lets you develop and deploy serverless applications to AWS, Azure, GCP & more. Use it with Pro for full serverless application lifecycle management.

```
org: my-company
app: my-app
service: users-api
provider:
  name: aws
```

36,506	15,796,965	758,758	215
Github Stars	Downloads	Weekly Deployments	Plugins

Build serverless apps

The Serverless Framework CLI is the simplest way to develop infinitely scalable, pay-per-execution serverless applications. A single configuration file allows you to list your functions and define the endpoints that they're subscribed to. It provides structure, automation and best practices out-of-the-box, allowing you to focus on building sophisticated, event-driven, serverless architectures, comprised of functions and events.

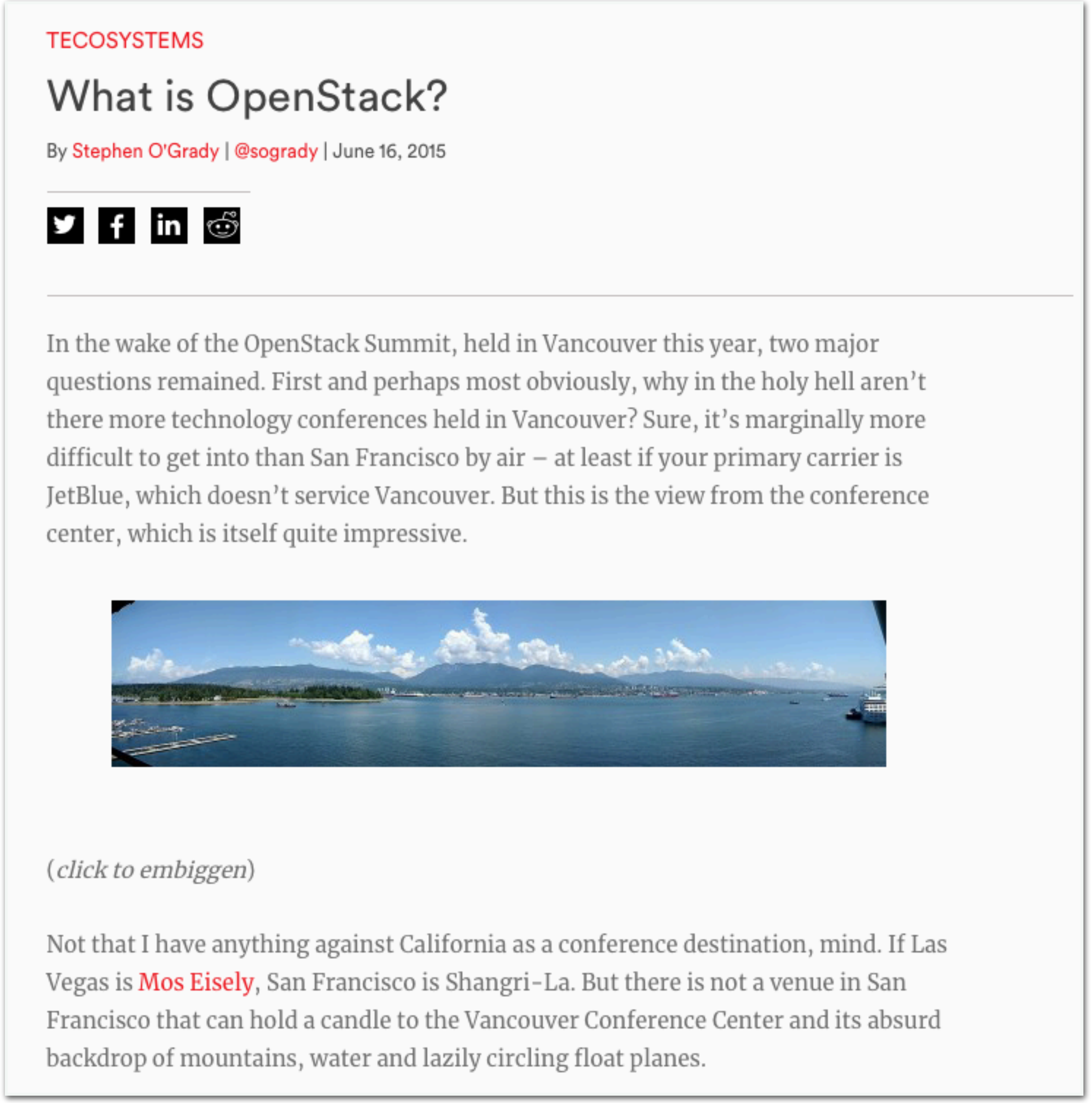
```
org: my-company
app: my-app
service: image-processor

functions:
  imageResizer:
    runtime: node.js8.10
    handler: index.imageResizer
    events:
      - s3:
          bucket: images
          event: s3:ObjectCreated:*
```

https://www.serverless.com/open-source/

@samnewman

FIGHT YOUR OWN NEED FOR INFRASTRUCTURE



TECOSYSTEMS

What is OpenStack?

By Stephen O'Grady | @sogrady | June 16, 2015

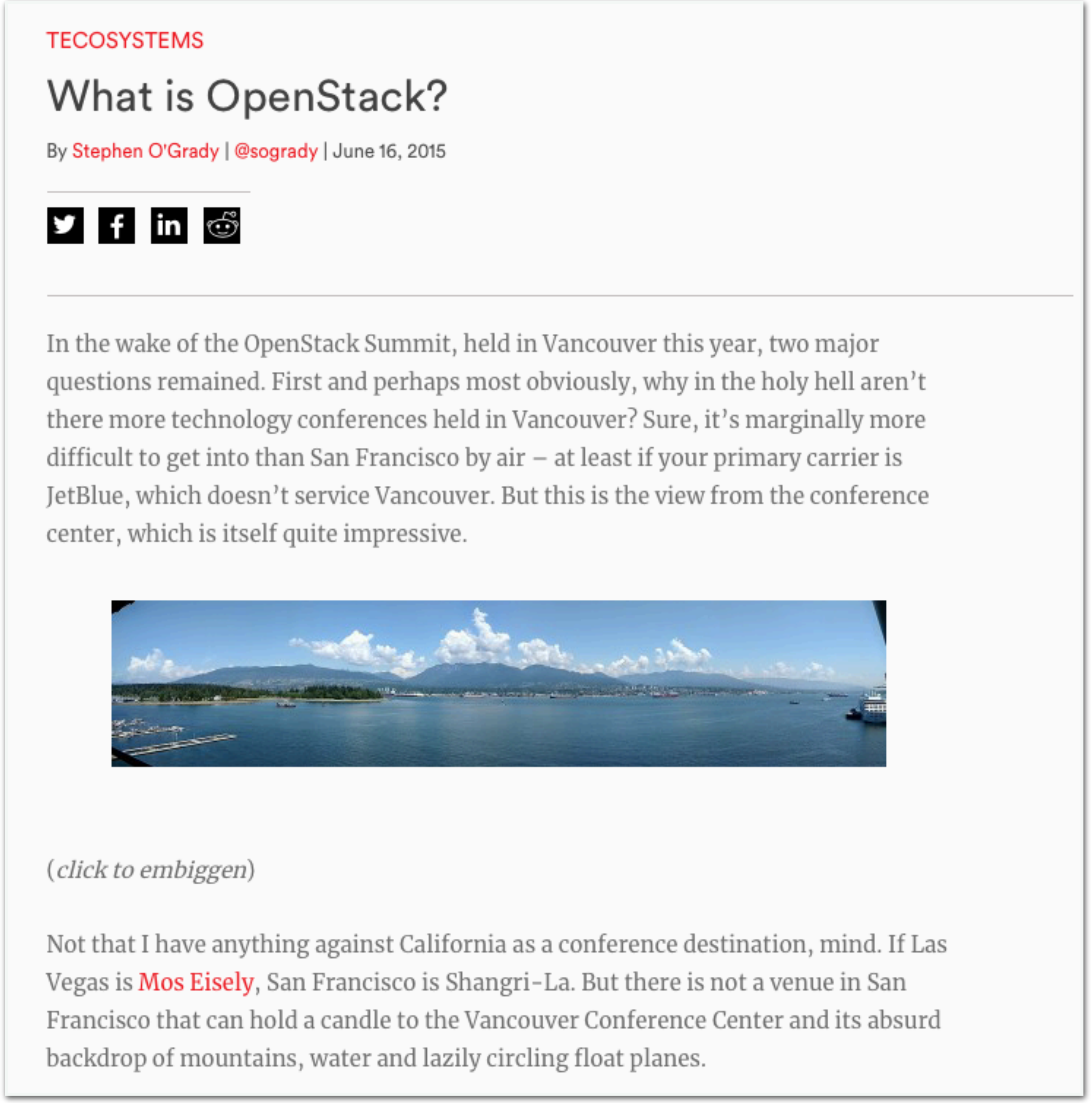
In the wake of the OpenStack Summit, held in Vancouver this year, two major questions remained. First and perhaps most obviously, why in the holy hell aren't there more technology conferences held in Vancouver? Sure, it's marginally more difficult to get into than San Francisco by air – at least if your primary carrier is JetBlue, which doesn't service Vancouver. But this is the view from the conference center, which is itself quite impressive.

(*click to embiggen*)

Not that I have anything against California as a conference destination, mind. If Las Vegas is Mos Eisely, San Francisco is Shangri-La. But there is not a venue in San Francisco that can hold a candle to the Vancouver Conference Center and its absurd backdrop of mountains, water and lazily circling float planes.

https://redmonk.com/sogrady/2015/06/16/what-is-openstack/

FIGHT YOUR OWN NEED FOR INFRASTRUCTURE



TECOSYSTEMS

What is OpenStack?

By Stephen O'Grady | @sogrady | June 16, 2015

In the wake of the OpenStack Summit, held in Vancouver this year, two major questions remained. First and perhaps most obviously, why in the holy hell aren't there more technology conferences held in Vancouver? Sure, it's marginally more difficult to get into than San Francisco by air – at least if your primary carrier is JetBlue, which doesn't service Vancouver. But this is the view from the conference center, which is itself quite impressive.

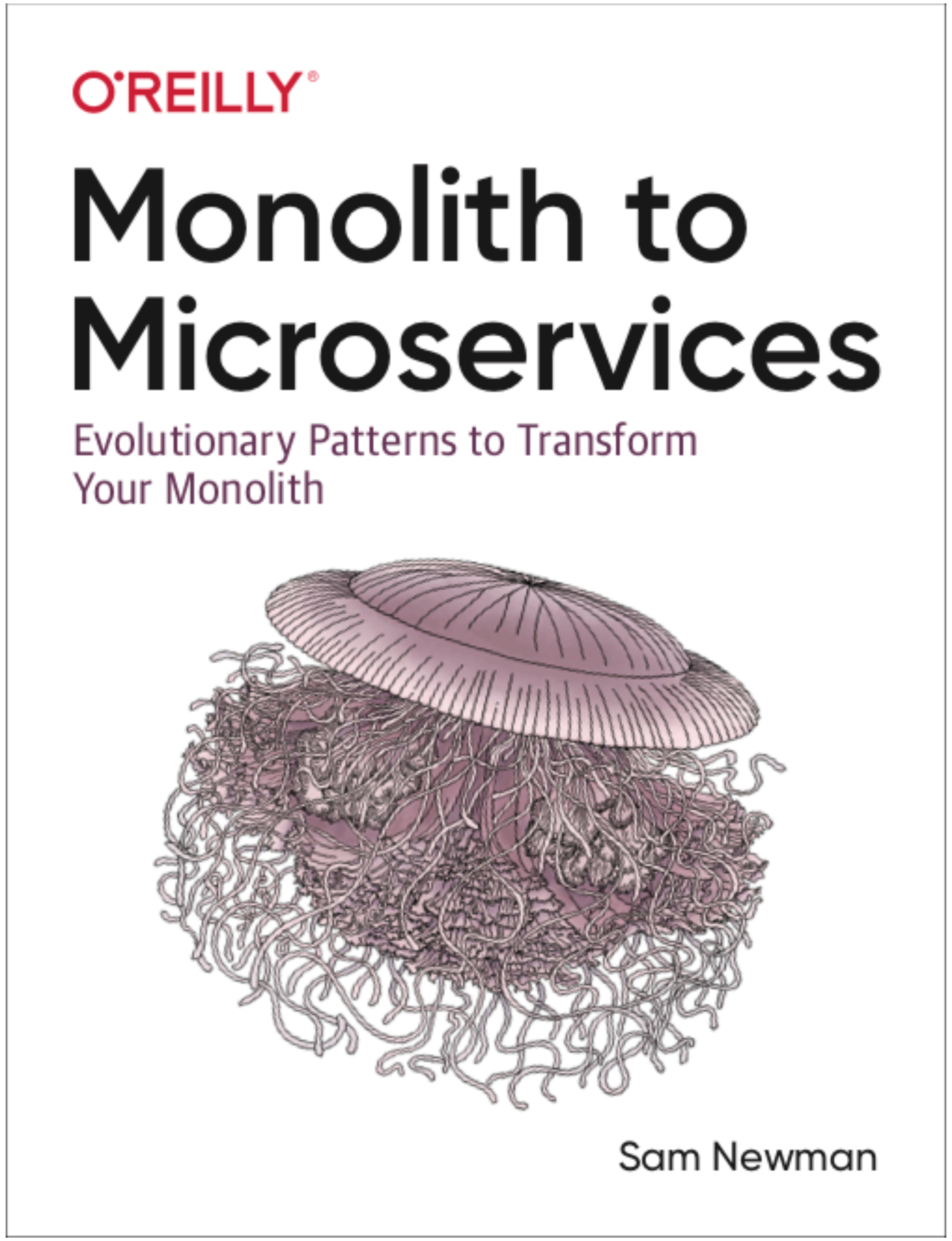
(*click to embiggen*)

Not that I have anything against California as a conference destination, mind. If Las Vegas is Mos Eisely, San Francisco is Shangri-La. But there is not a venue in San Francisco that can hold a candle to the Vancouver Conference Center and its absurd backdrop of mountains, water and lazily circling float planes.

"…there are legions of IT staffers that will be protecting what they believe is their livelihood – the private infrastructure – at all costs. Unless technical leadership is willing to wage total war on its own infrastructure, then, private infrastructure will continue to be a thing."

- Stephen O'Grady, Redmonk

https://redmonk.com/sogrady/2015/06/16/what-is-openstack/



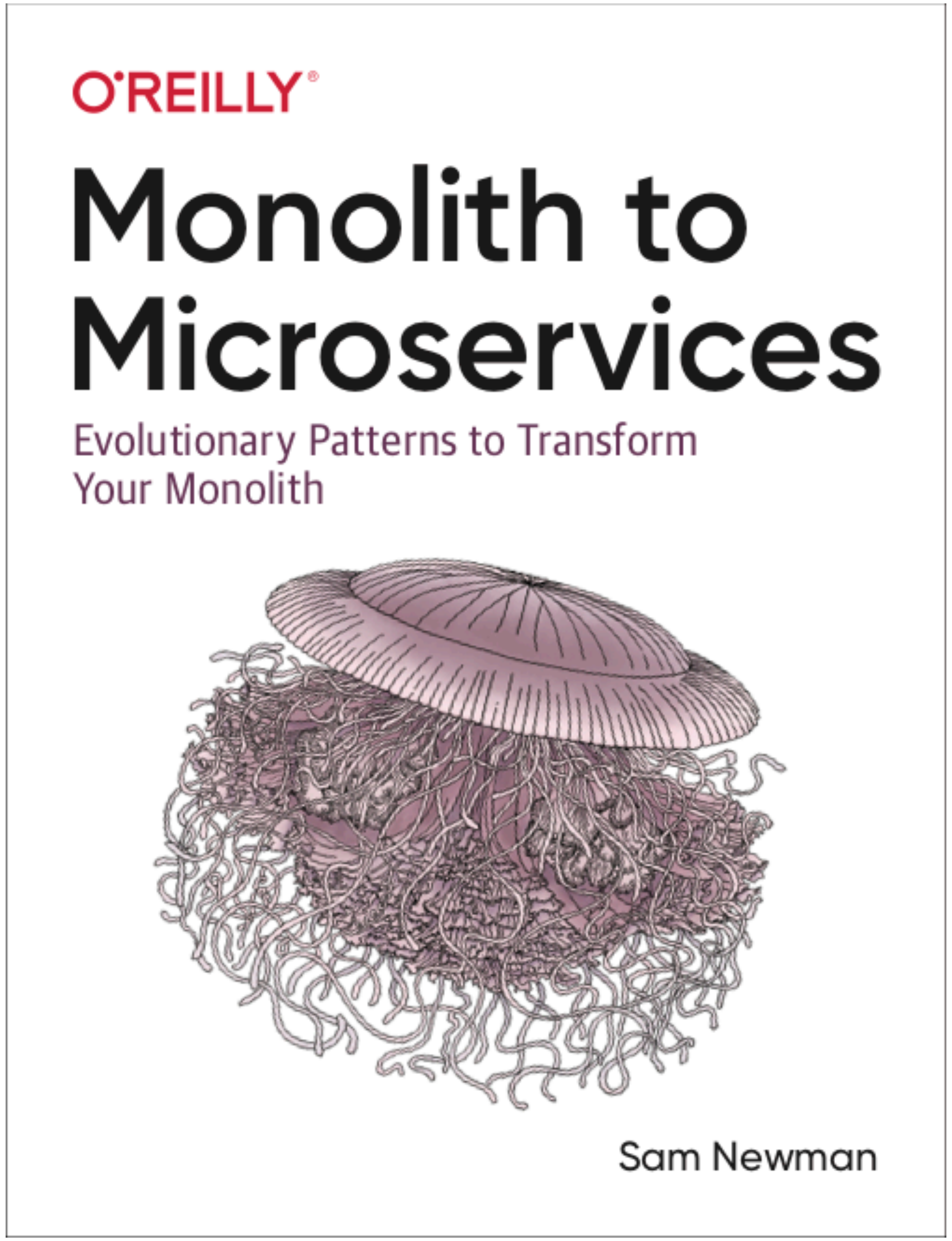
O'REILLY

Monolith to Microservices

Evolutionary Patterns to Transform Your Monolith

Sam Newman

https://samnewman.io/books/monolith-to-microservices/

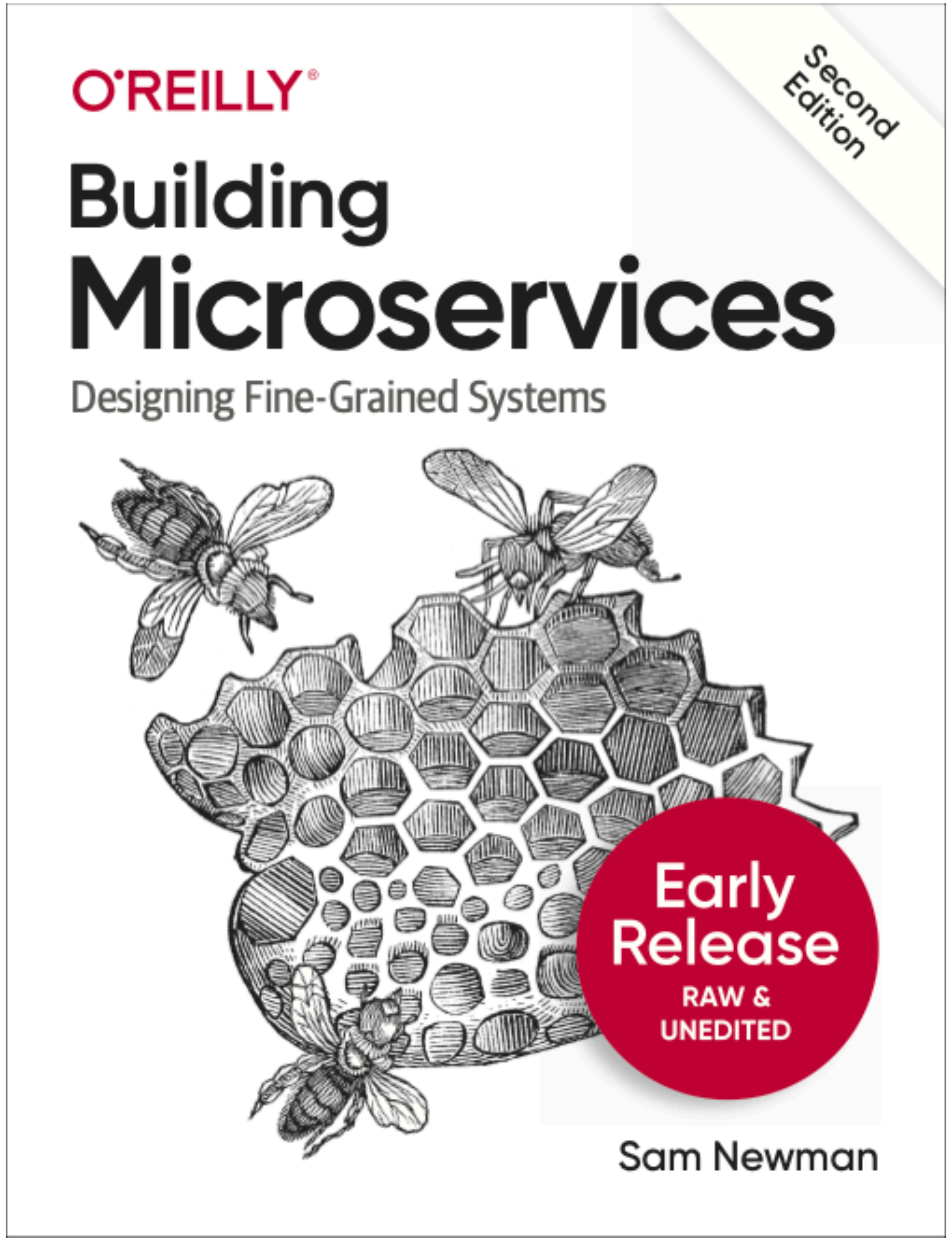


O'REILLY®

Monolith to Microservices

Evolutionary Patterns to Transform Your Monolith

Sam Newman



O'REILLY®

Second Edition

Building Microservices

Designing Fine-Grained Systems

Early Release

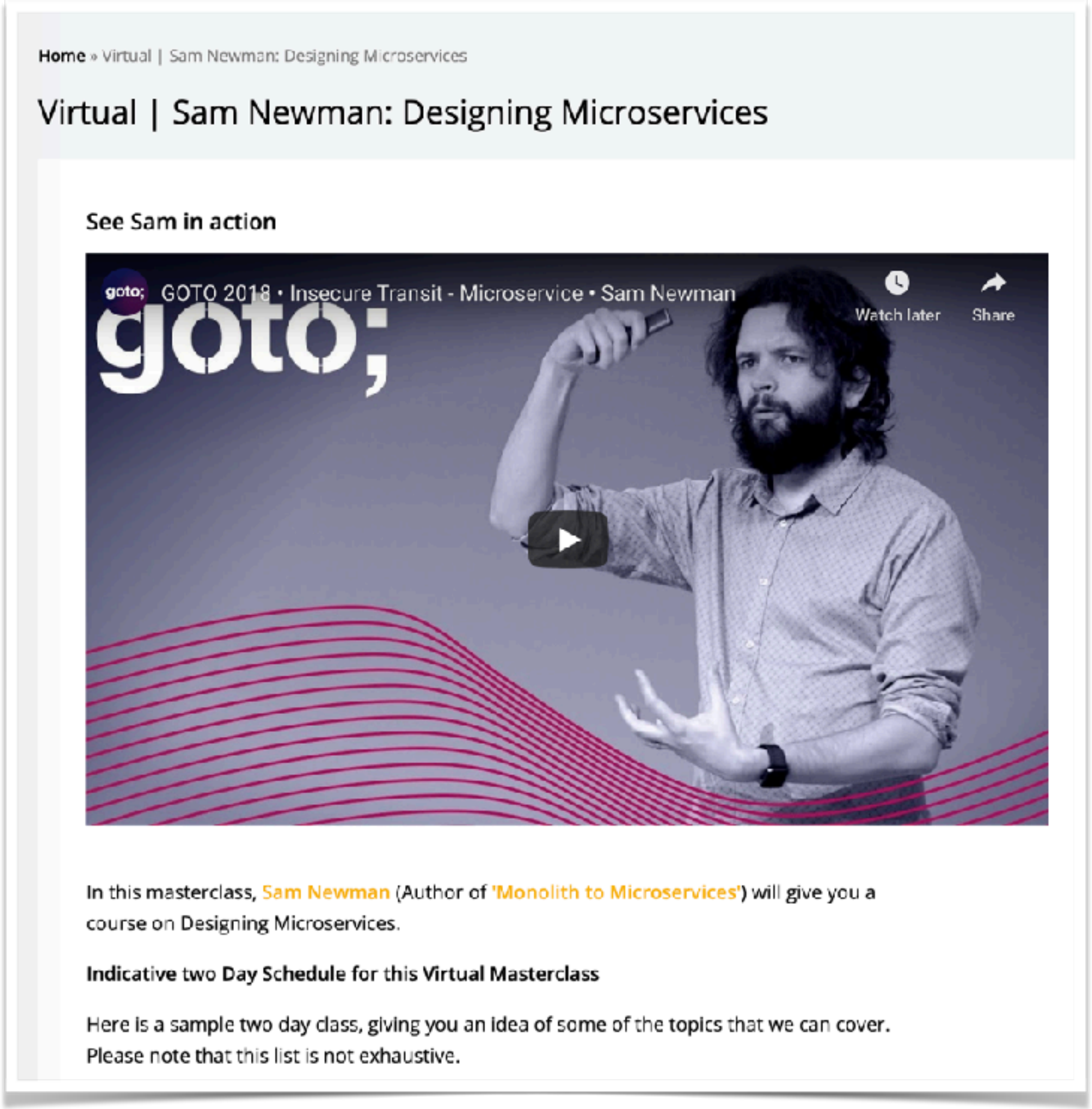
RAW & UNEDITED

Sam Newman

https://samnewman.io/books/monolith-to-microservices/

https://samnewman.io/

THANKS!



Home » Virtual | Sam Newman: Designing Microservices

Virtual | Sam Newman: Designing Microservices

See Sam in action

In this masterclass, Sam Newman (Author of 'Monolith to Microservices') will give you a course on Designing Microservices.

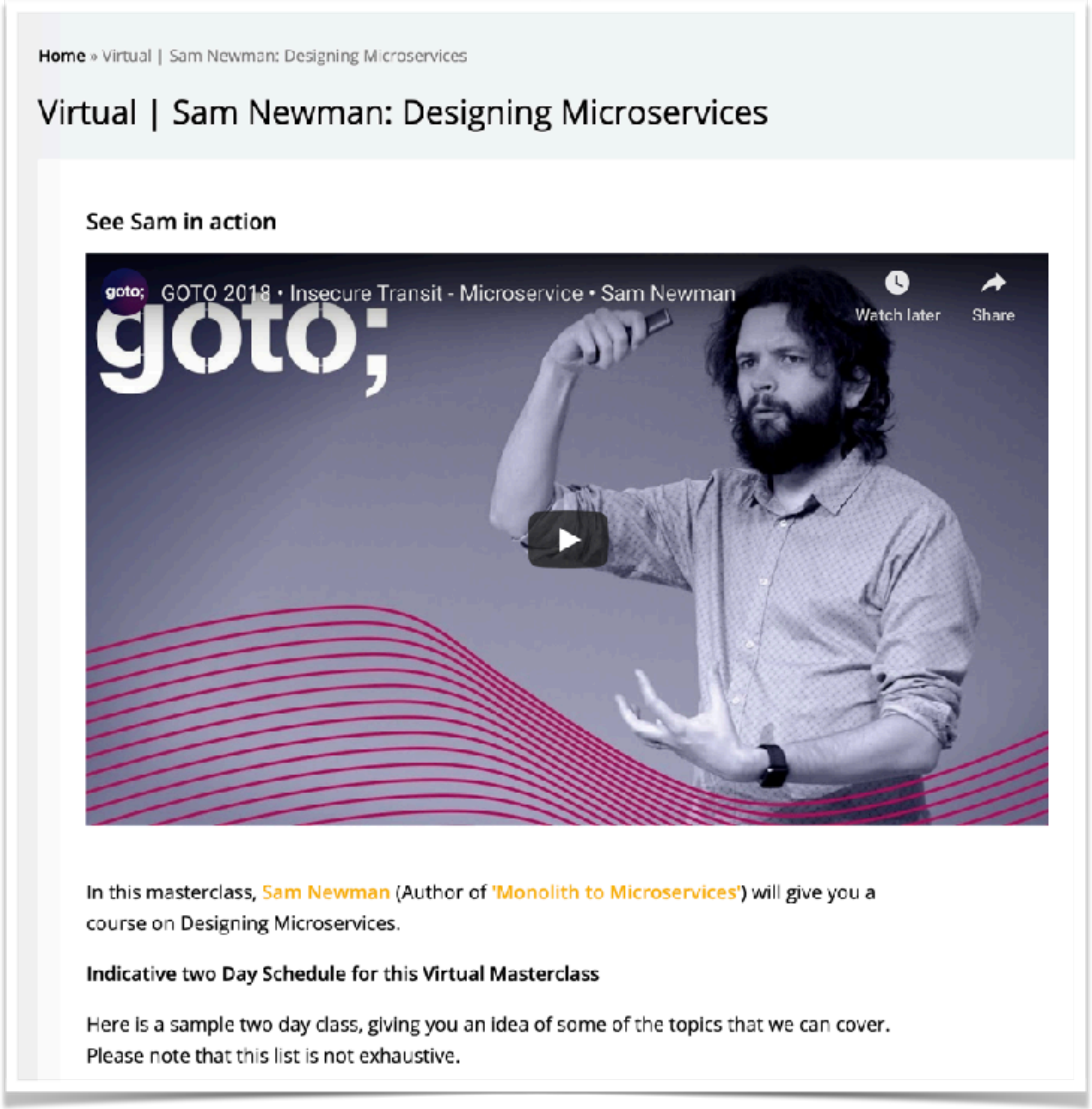
Indicative two Day Schedule for this Virtual Masterclass

Here is a sample two day class, giving you an idea of some of the topics that we can cover. Please note that this list is not exhaustive.

https://www.gotoacademy.nl/collections/virtual-sam-newman-designing-microservices

@samnewman

THANKS!



Home » Virtual | Sam Newman: Designing Microservices

Virtual | Sam Newman: Designing Microservices

See Sam in action

GOTO 2018 • Insecure Transit - Microservice • Sam Newman

In this masterclass, Sam Newman (Author of 'Monolith to Microservices') will give you a course on Designing Microservices.

Indicative two Day Schedule for this Virtual Masterclass

Here is a sample two day class, giving you an idea of some of the topics that we can cover. Please note that this list is not exhaustive.

Sam Newman & Associates

https://www.gotoacademy.nl/collections/virtual-sam-newman-designing-microservices