Making AI **FaaS***t*

.

Dragos D Haut - Principal Engineer @Adobe

Akhilesh Kumar - Applied ML Engineer @Adobe

FaaS

Function as A Service

a.k.a Serverless

FaaS Value Props

FaaS Value Props

1. **FaaS**_ter_ to **PROTOTYPE** services

FaaS Value Props

1. **FaaS***ter* to **CREATE** services

FaaS Value Props

1. **FaaS***ter* to create services

2. Never pay for **Idle**

FaaS Value Props

1. **FaaS***ter* to create services

2. Never pay for **Idle**

3. **Low** maintenance overhead

FaaS: Build **more**, pay **less**

1. **FaaS***ter* to create services
2. Never pay for **Idle**
3. **Low** maintenance overhead

II

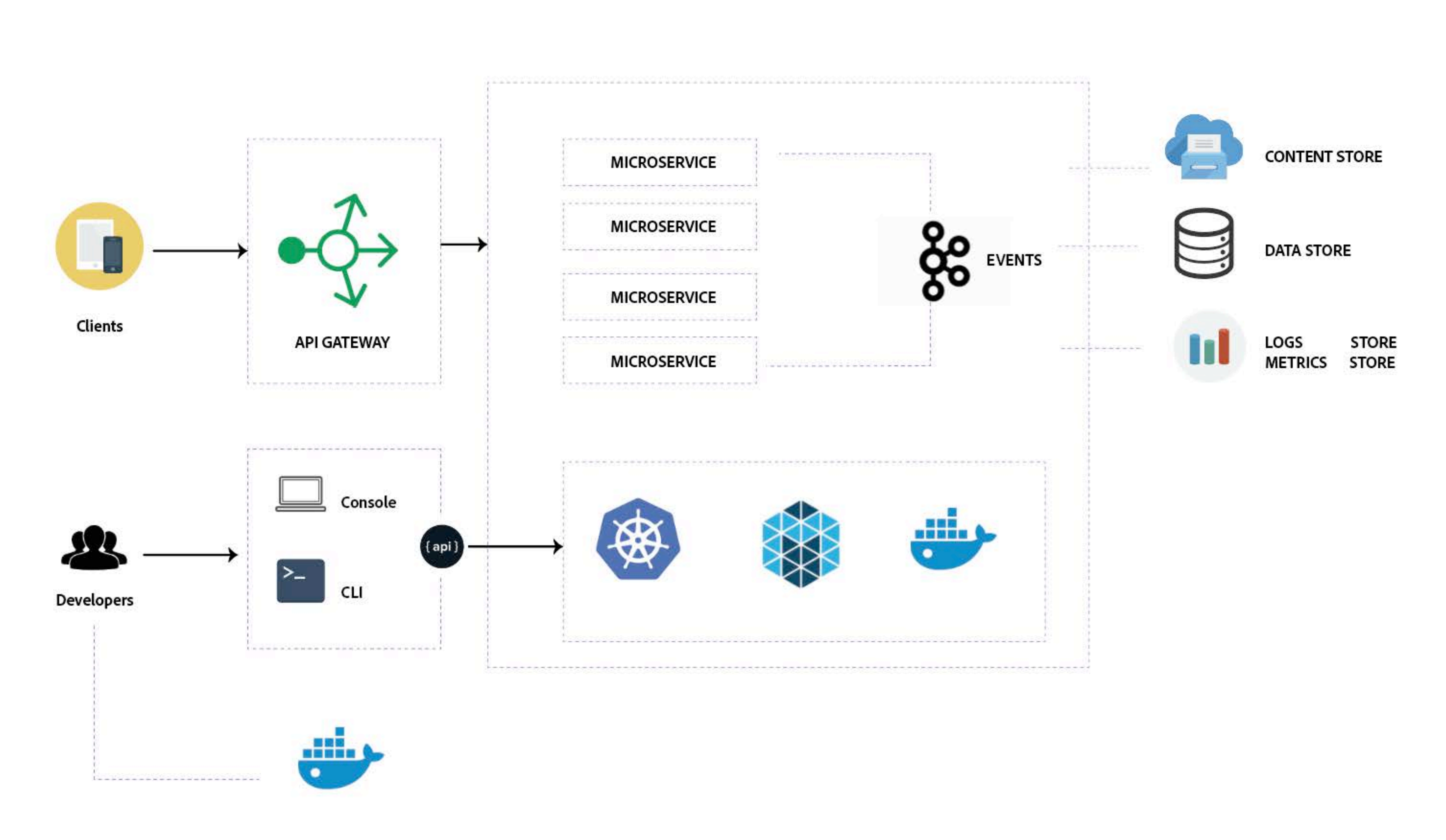
Evolution of Business Logic

Monolith

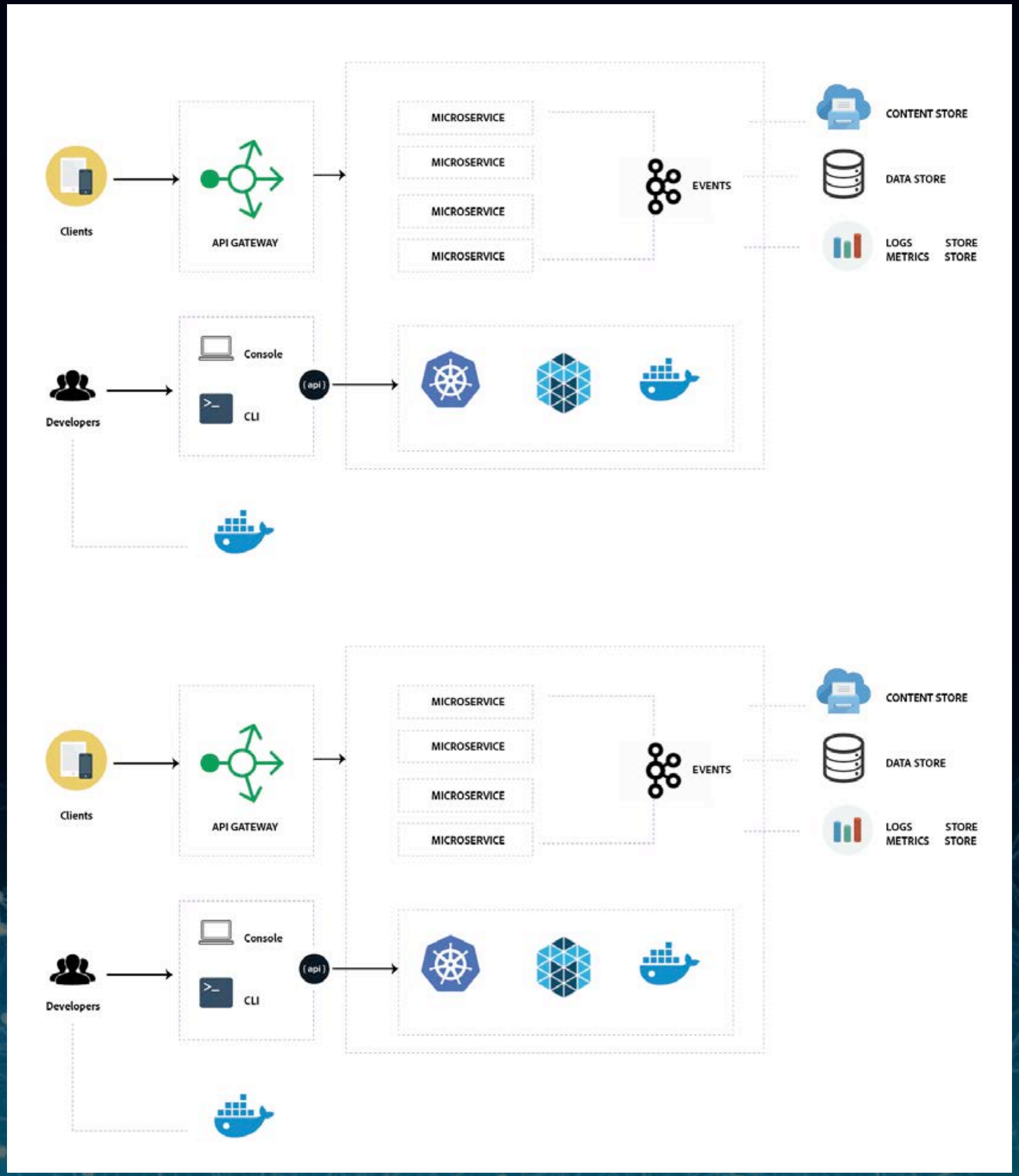
Microservices

Functions

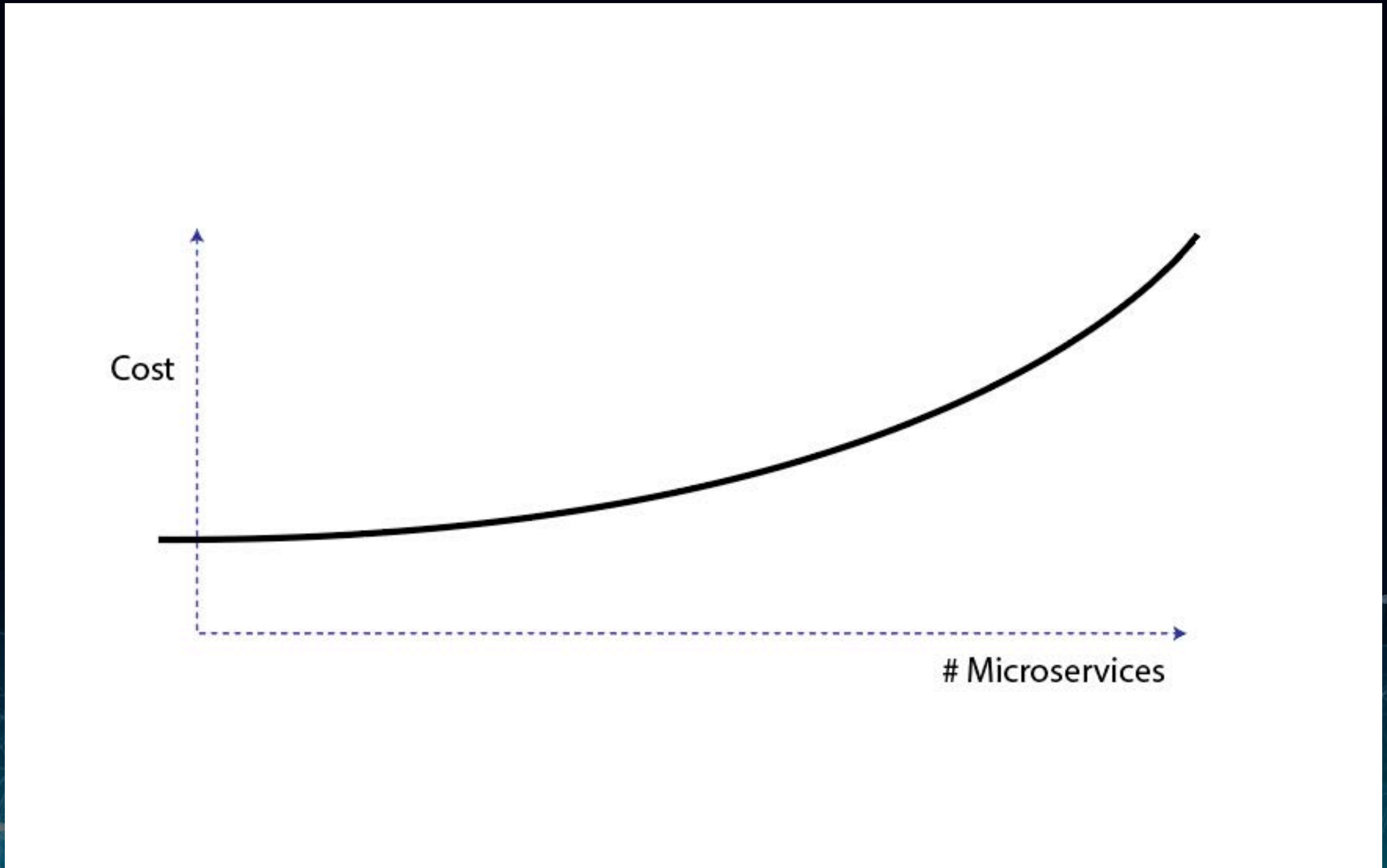
High-level Microservice architecture



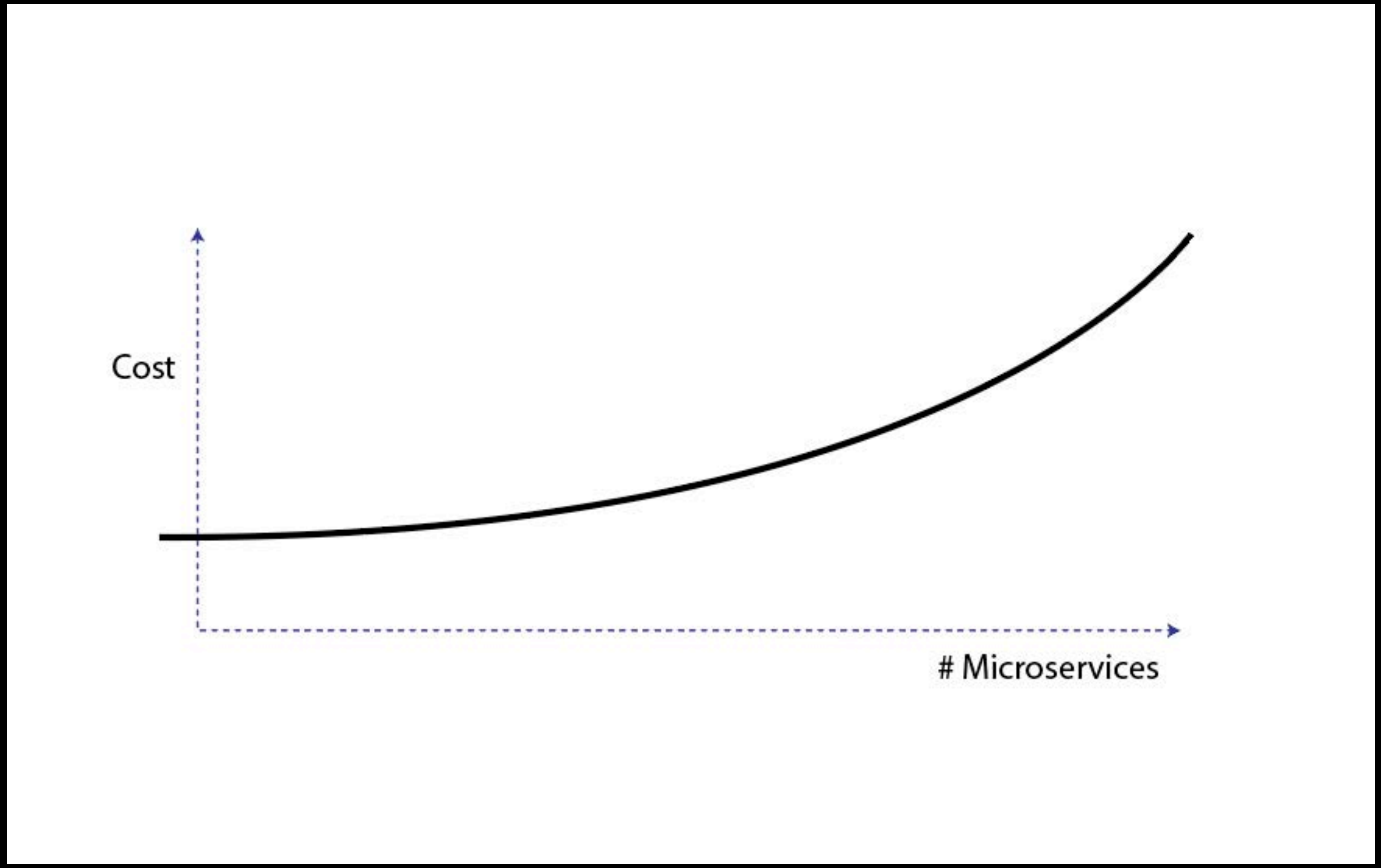
Multiple regions



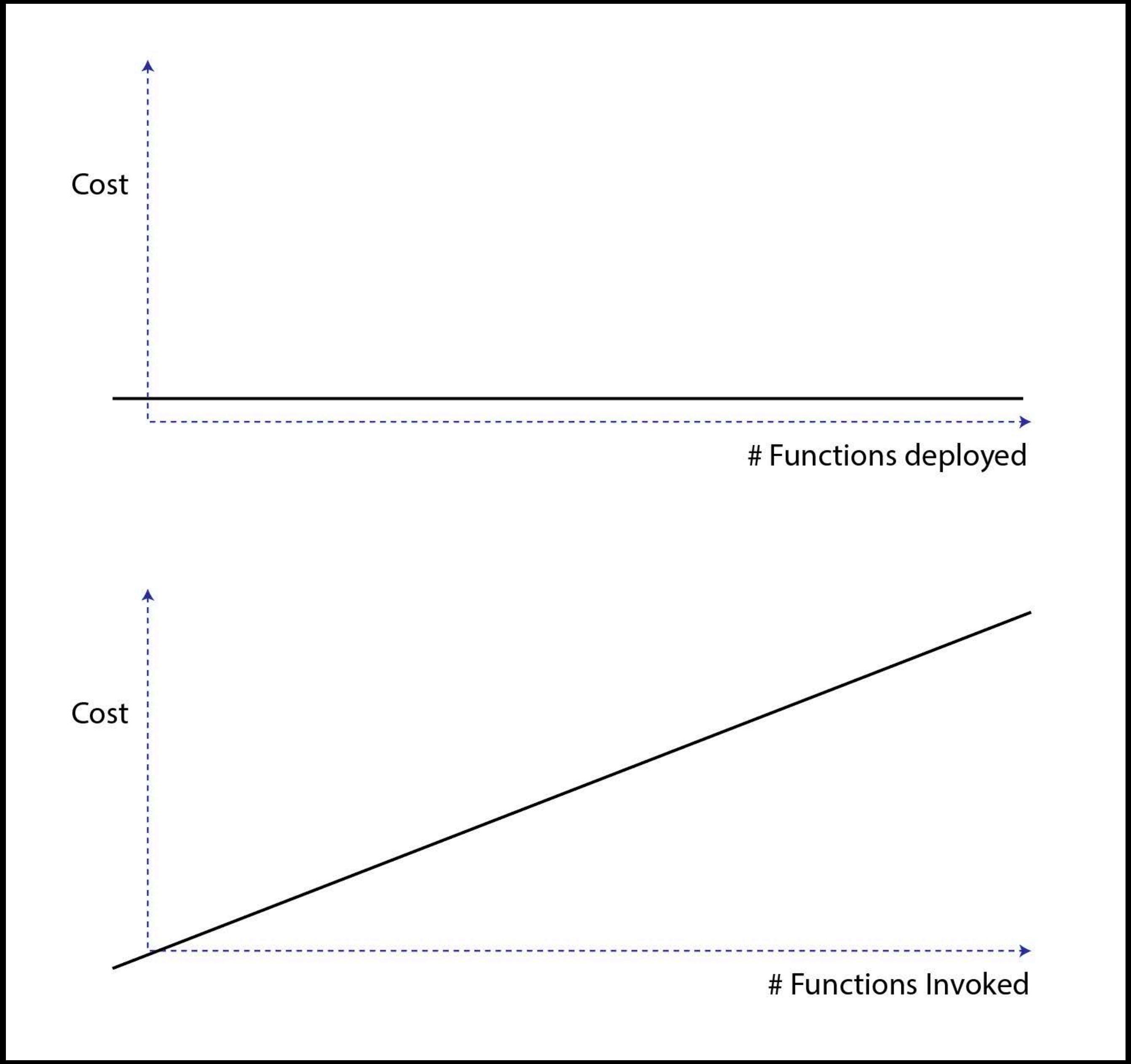
Microservice Cost & COGS



Microservice Cost vs **FaaS** Cost


Cost
# Microservices

VS


Cost
# Functions deployed
Cost
# Functions Invoked

How ?

FaaS has better premises

FaaS premises

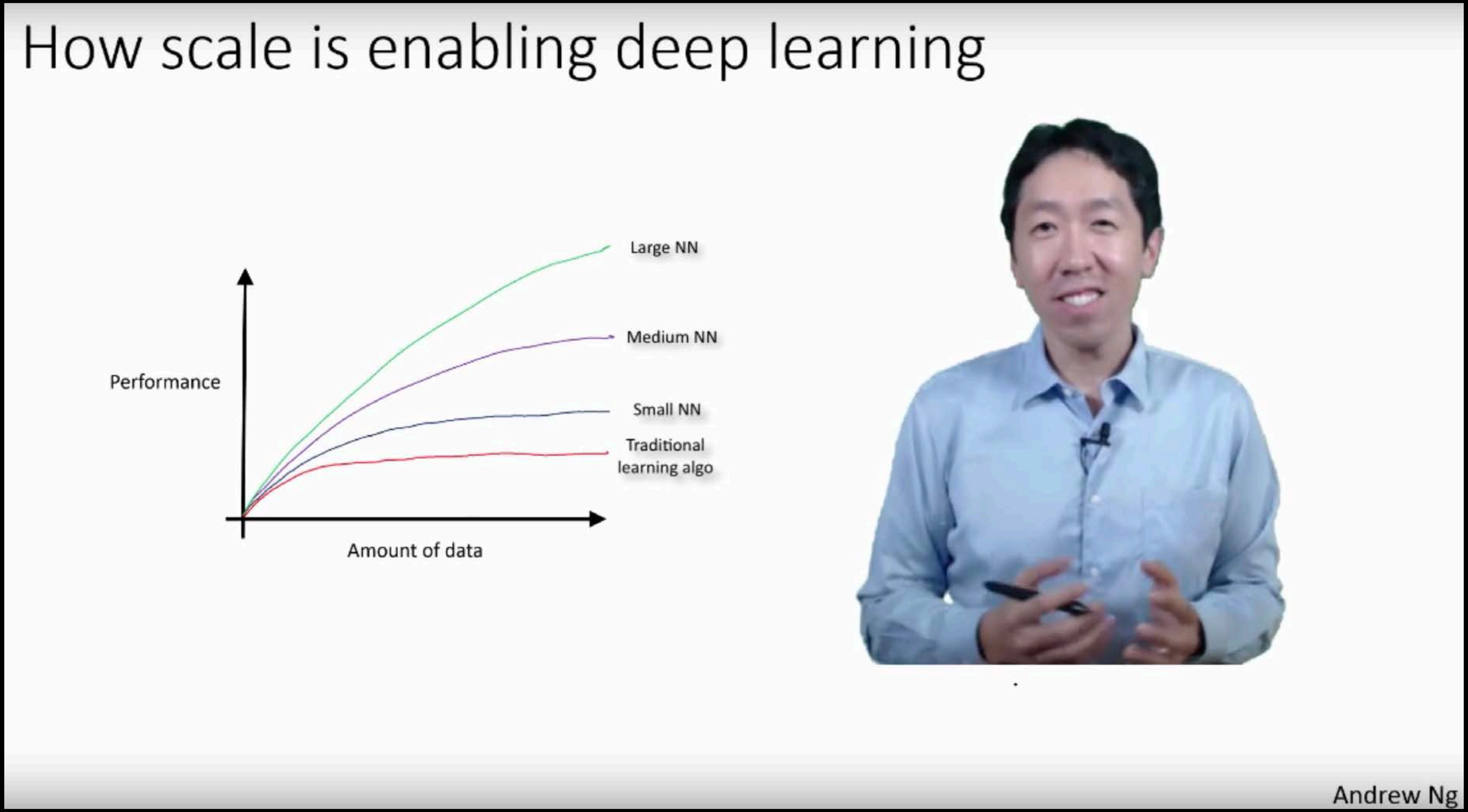
Code - a smaller unit to deploy and scale

FaaS premises

Code - a smaller unit to deploy and scale

Request based auto-scaling

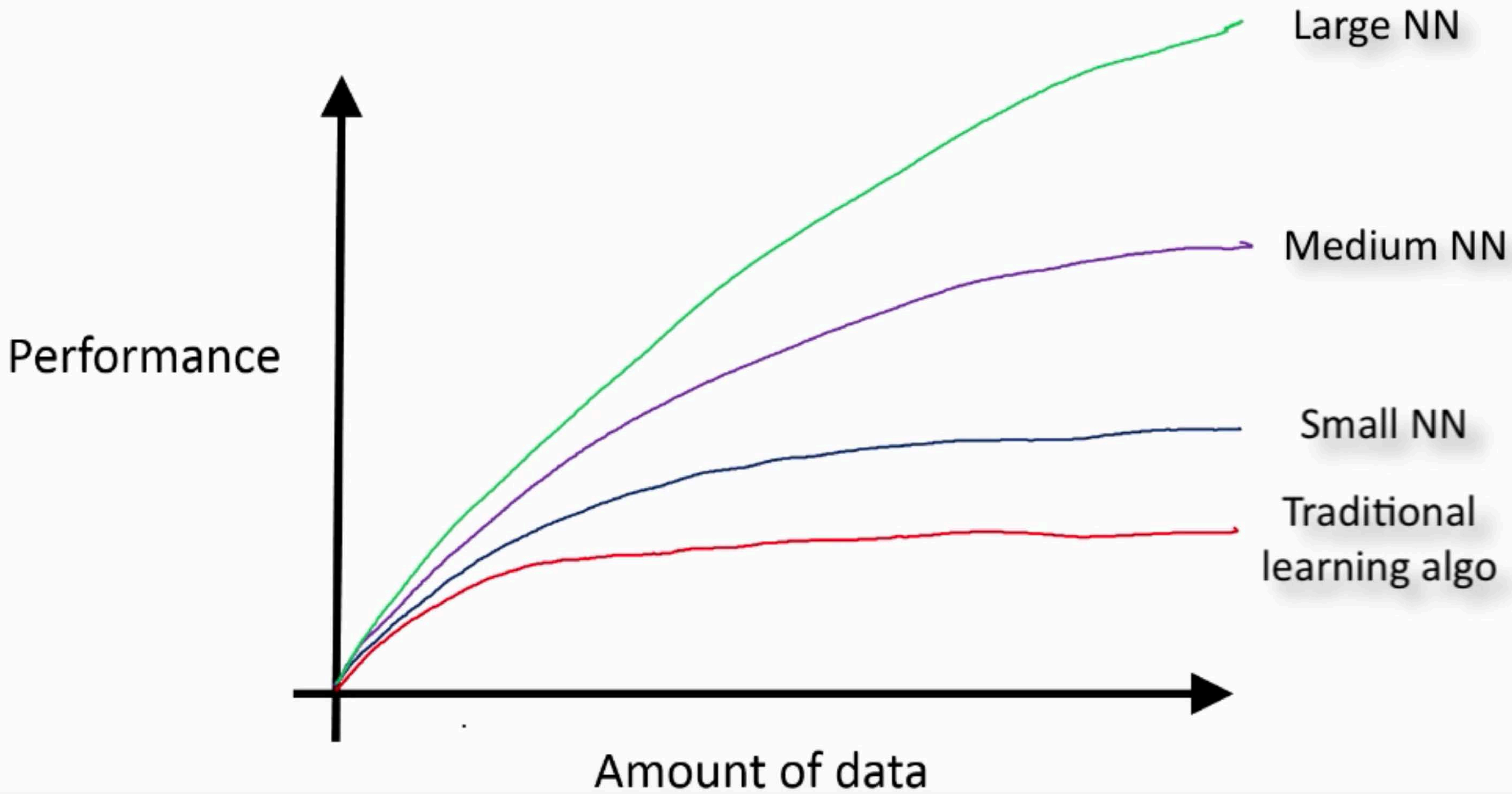
Making AI **FaaS**_t_



How scale is enabling deep learning

--- https://twitter.com/andrewyng/status/700086119333892096

One picture explaining the rise of Deep Learning



Andrew N

"With AI, we should look at the programmer more as a ***teacher***, rather than a ***micro-manager***. "

— Peter Norvig, Director of Research at Google.

" We spent the last 40 years building up tools to build programs to deal with text (**code**) in a good way …"

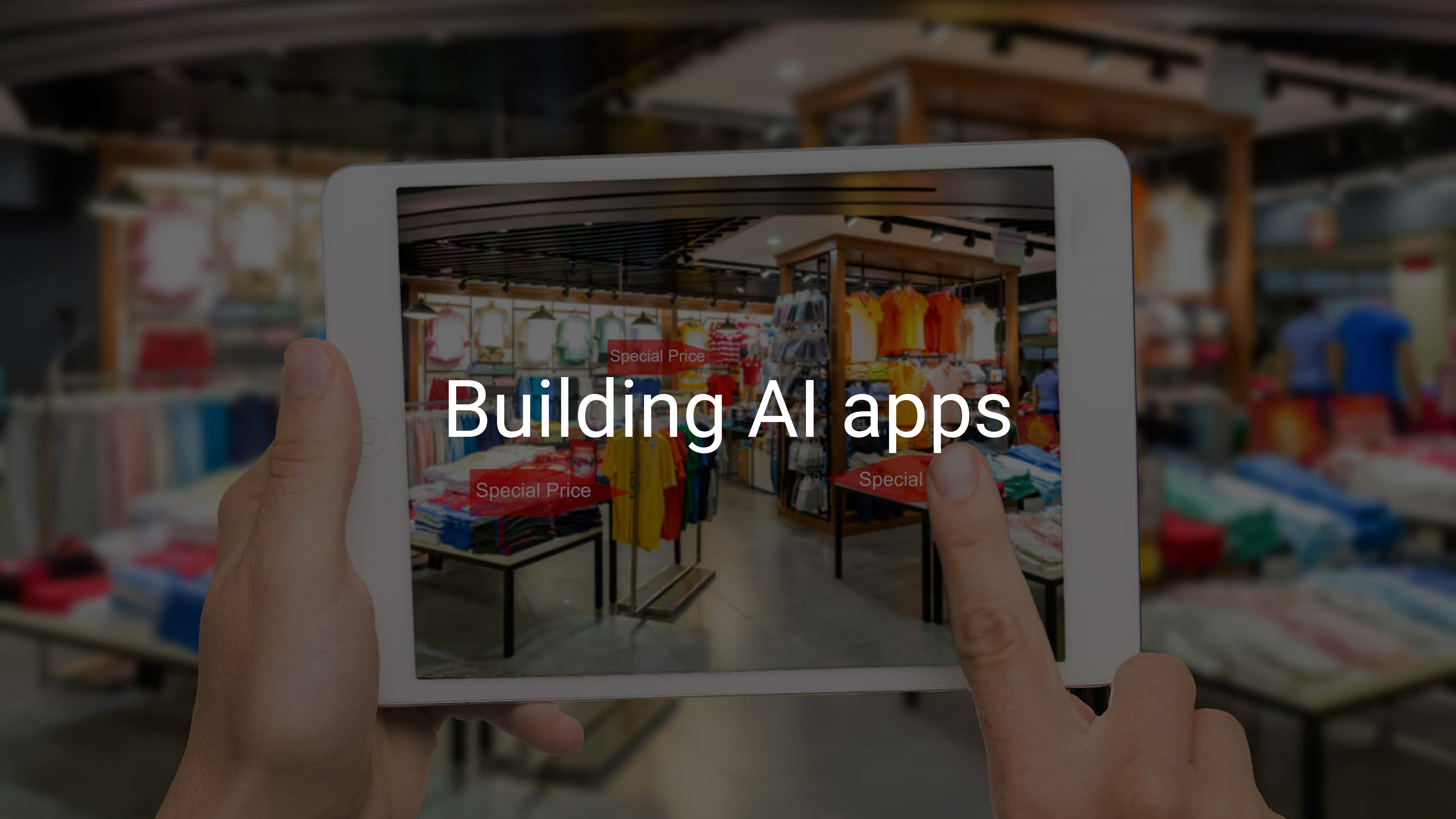
"… but right now we are creating **models instead of text**, and we just don’t have the tools to deal with that. We need to **retool the industry**."

— Peter Norvig, Director of Research at Google.

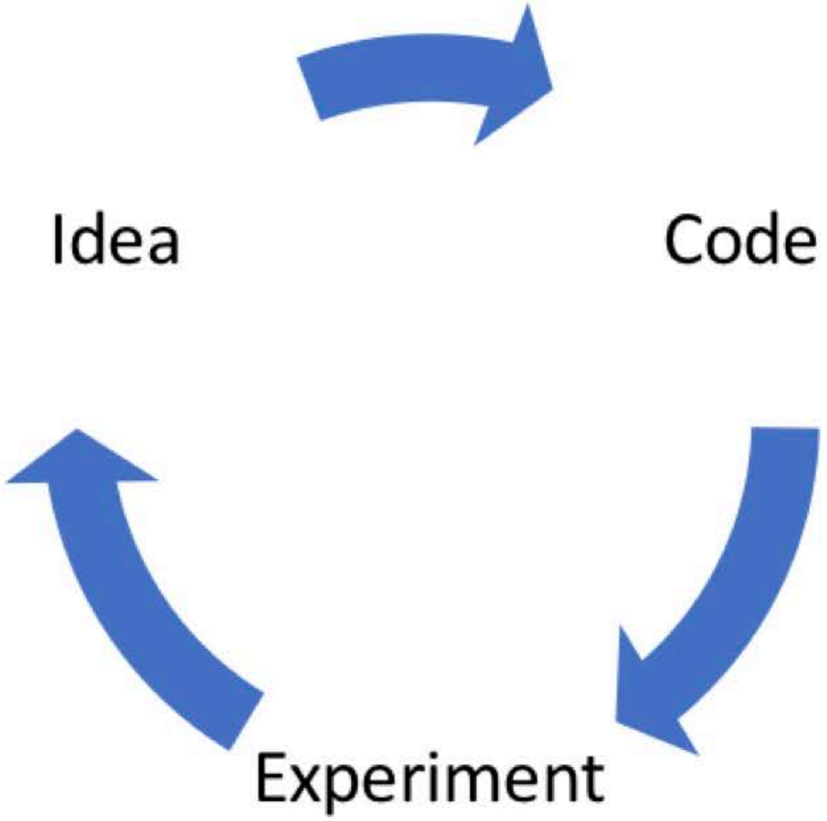
"**Neural networks** are not just another classifier, they represent the beginning of a fundamental shift in how we write software. They are **Software 2.0**."

— (Nov, 2017) - Andrej Karpathy, Director of AI at Tesla

Building AI apps

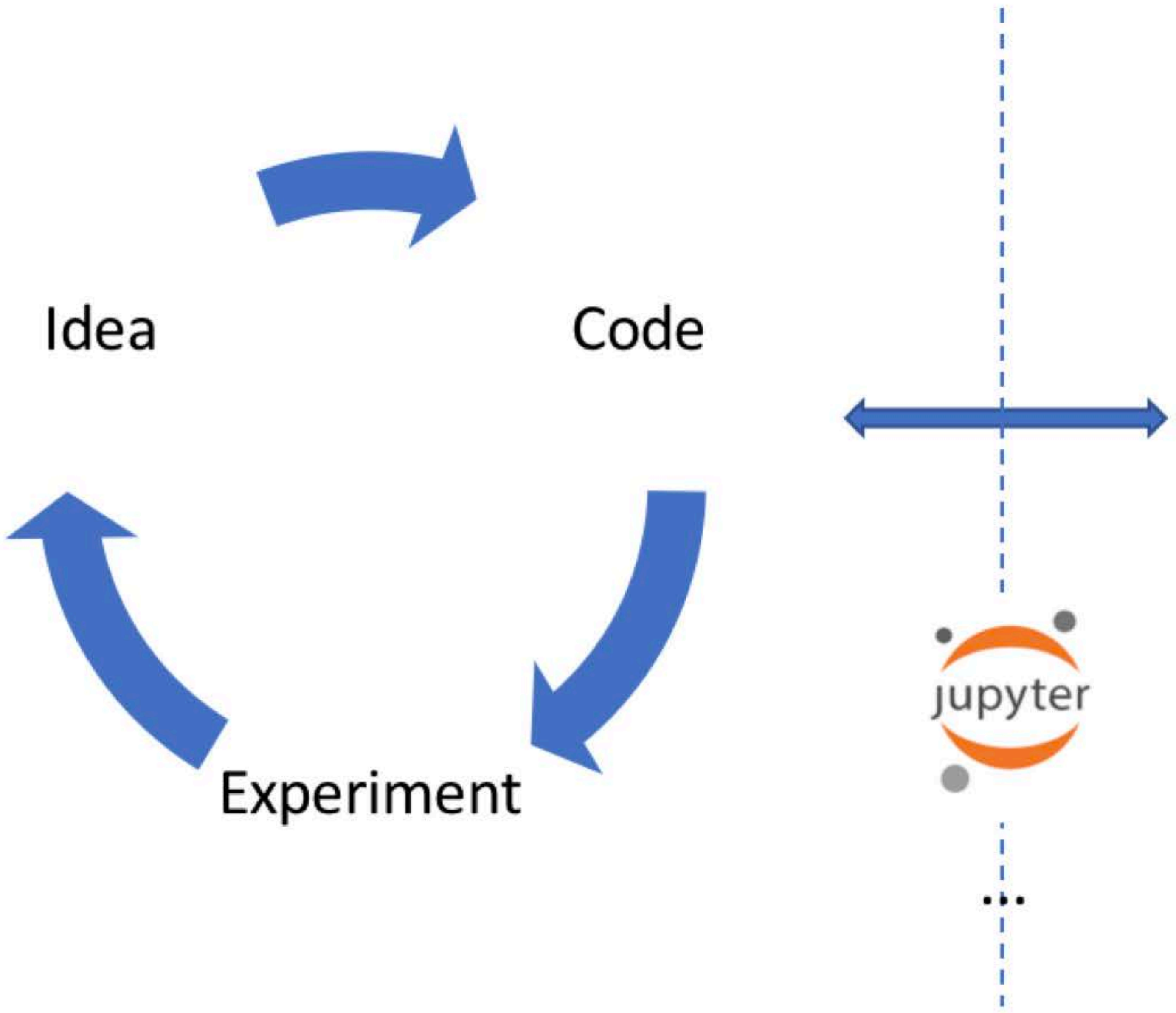


Process



Process

Tools



Idea

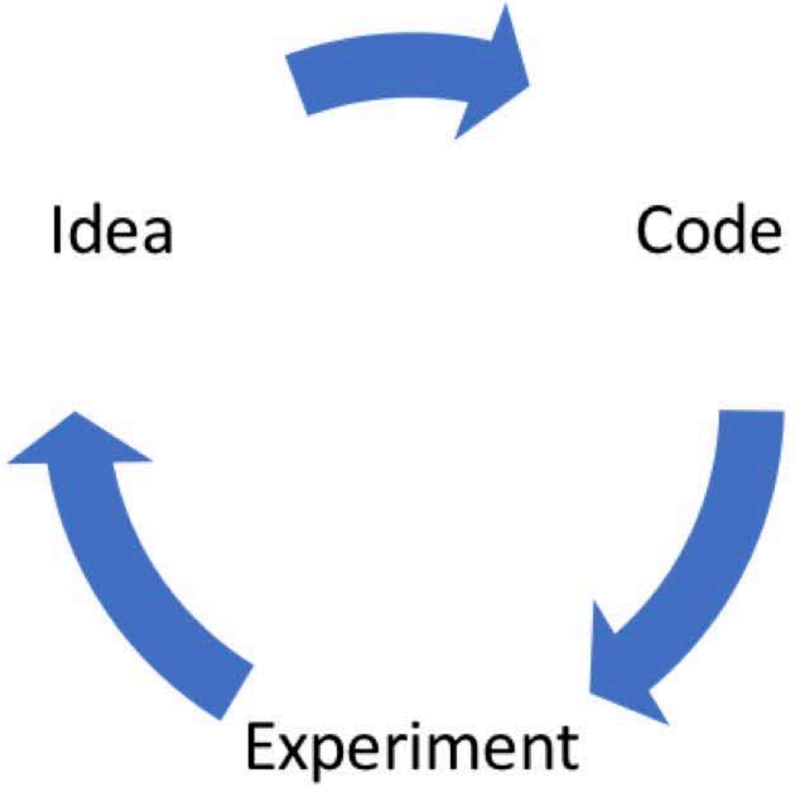
Code

Experiment

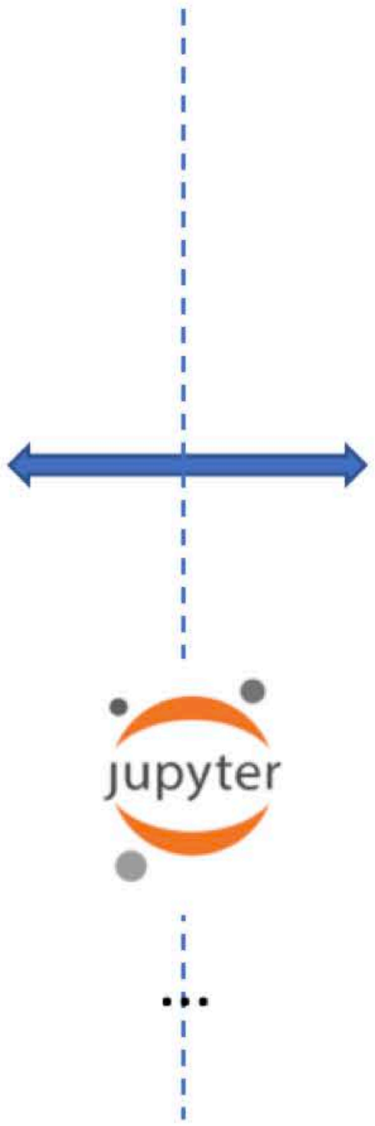
jupyter

...

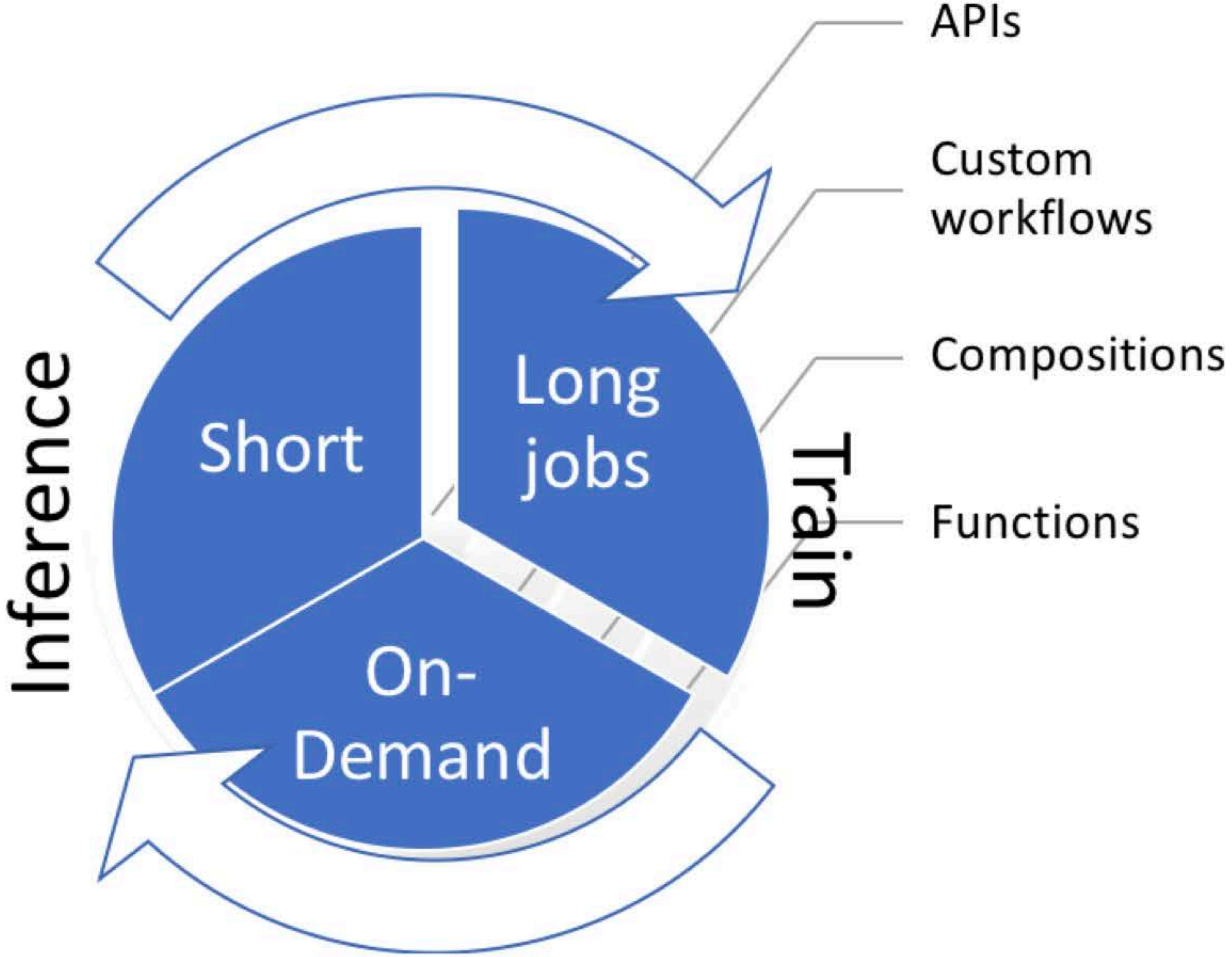
Process



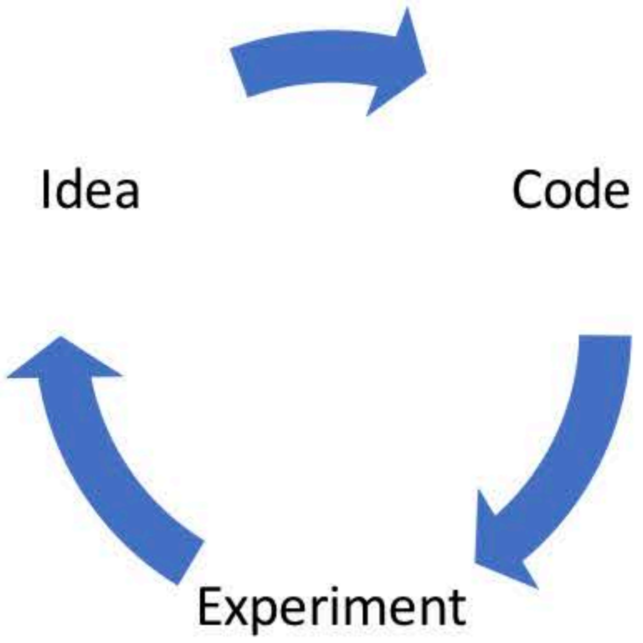
Tools



Compute

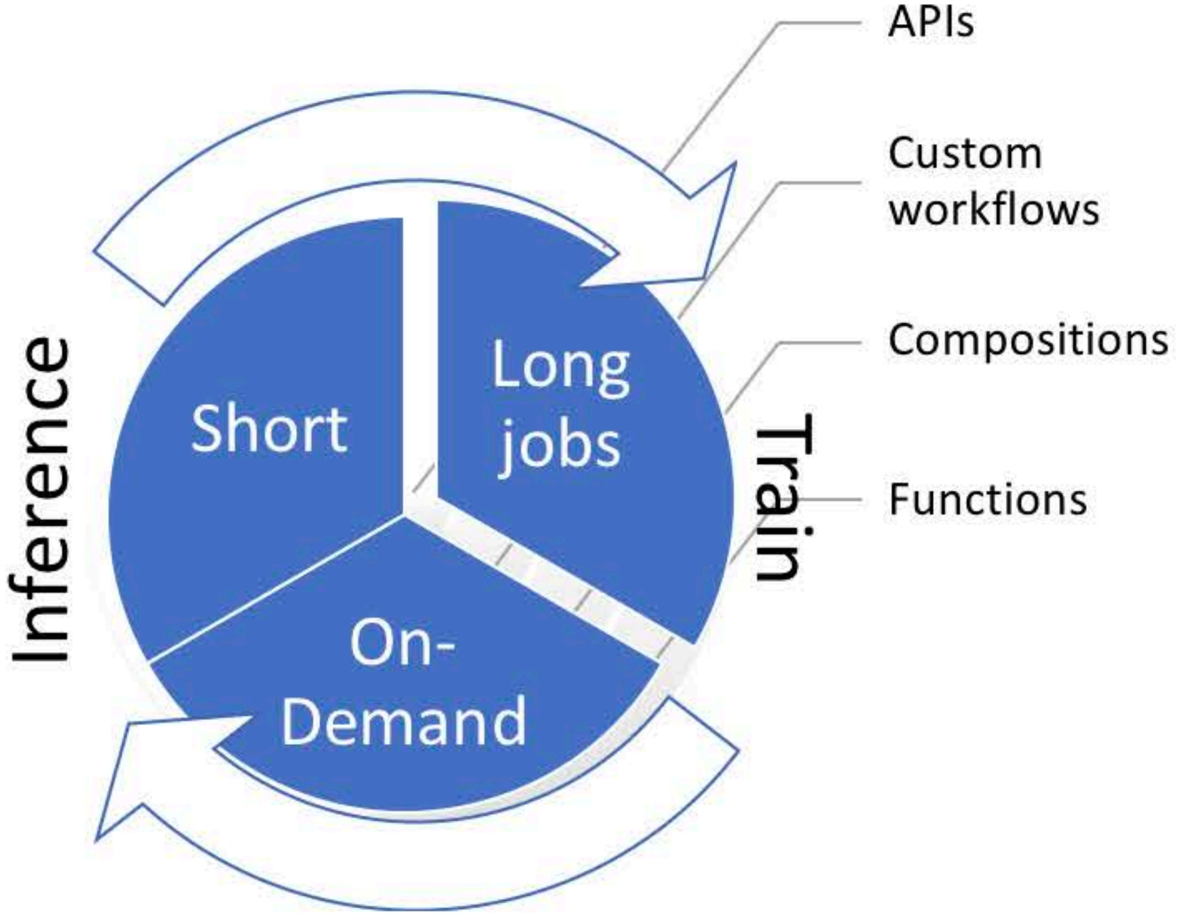


Process



Tools

Compute



Hardware

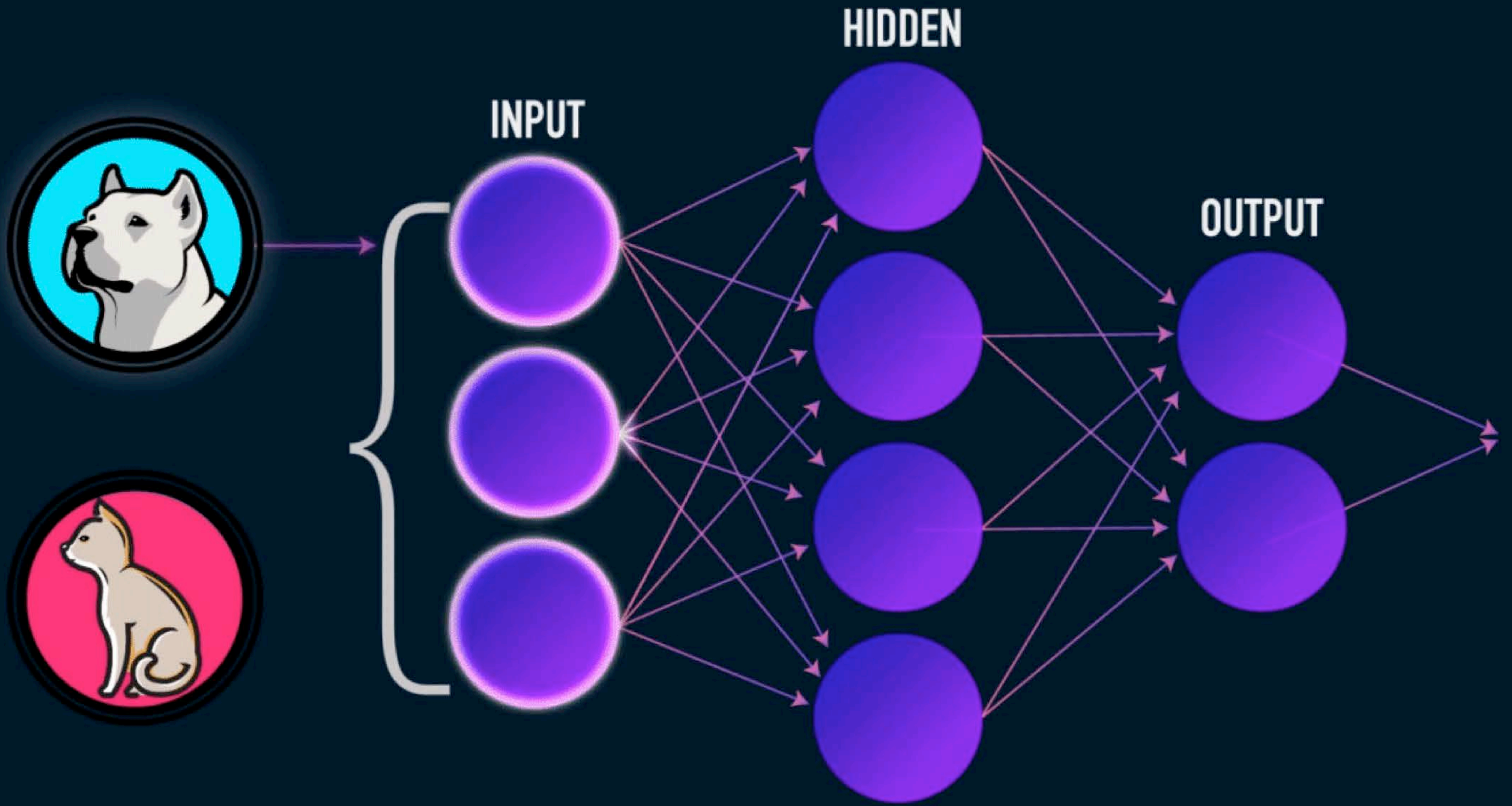
Training *vs* Inference

Training *vs* Inference

Learning *vs* Answering

Inference

Getting a new data sample to infer an **answer**


HIDDEN
INPUT
OUTPUT

Inference

Runs faster than Training

Models process one input at a time

Inference matches the **FaaS** model

Enough code for a function

Each function processes one request at a time

```
function (input) {
  //1. download and cache model
  //2. return inference(input)
}
```

Additional **FaaS** benefits

It's **FaaS***ter* to deploy the code directly

Never pay for idle

Low maintenance overhead

Real apps integrate **multiple** algorithms into **workflows**

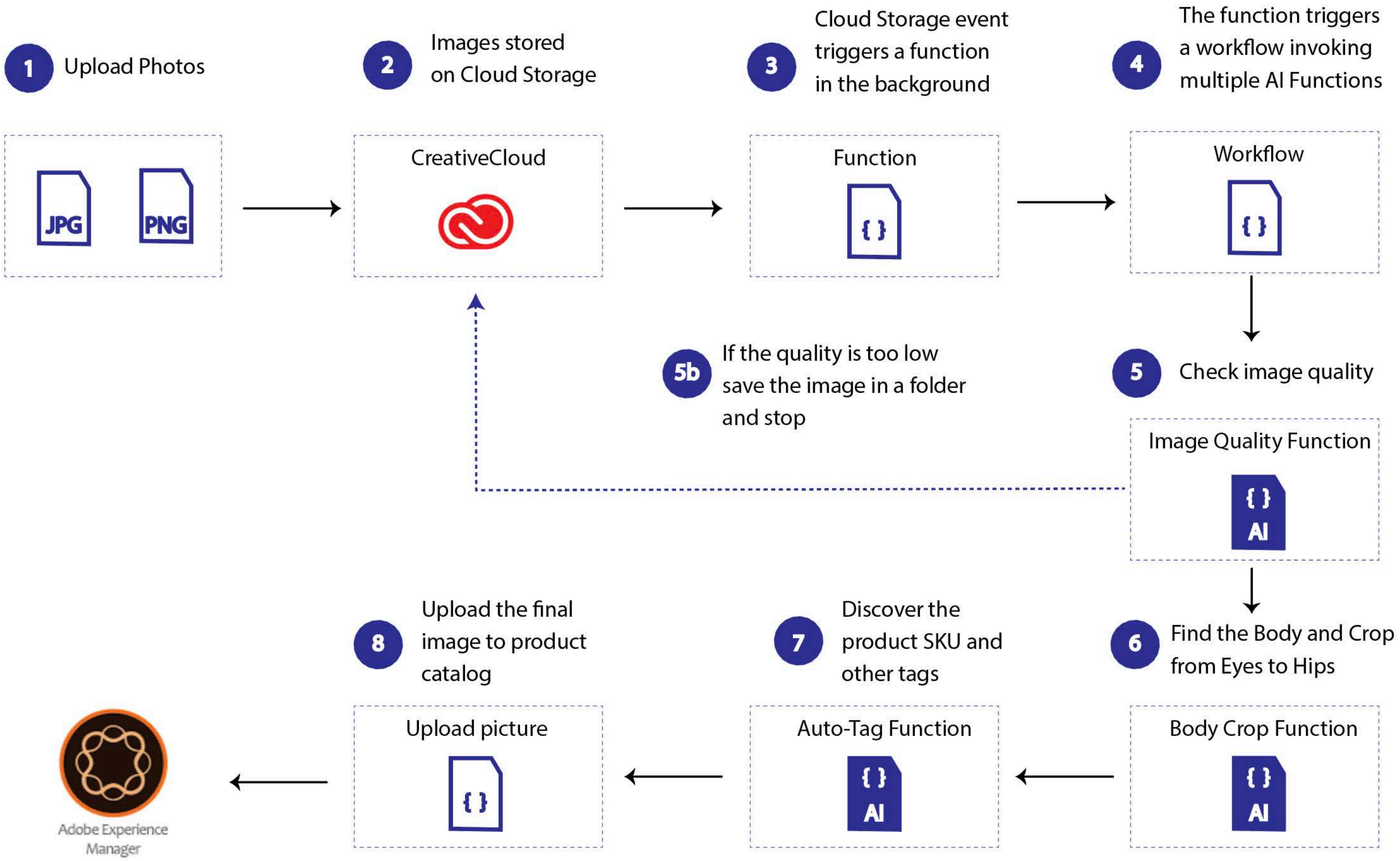
Real apps integrate **multiple** algorithms into **workflows**

reusing existing functions

Demo

- Jupyter Notebook
  https://github.com/akh64bit/qconsf

- AI Composition
  http://opensource.adobe.com/adobe-sensei-ai-functions/



1 Upload Photos

2 Images stored on Cloud Storage

CreativeCloud

3 Cloud Storage event triggers a function in the background

Function

4 The function triggers a workflow invoking multiple AI Functions

Workflow

5 Check image quality

Image Quality Function

5b If the quality is too low save the image in a folder and stop

6 Find the Body and Crop from Eyes to Hips

Body Crop Function

7 Discover the product SKU and other tags

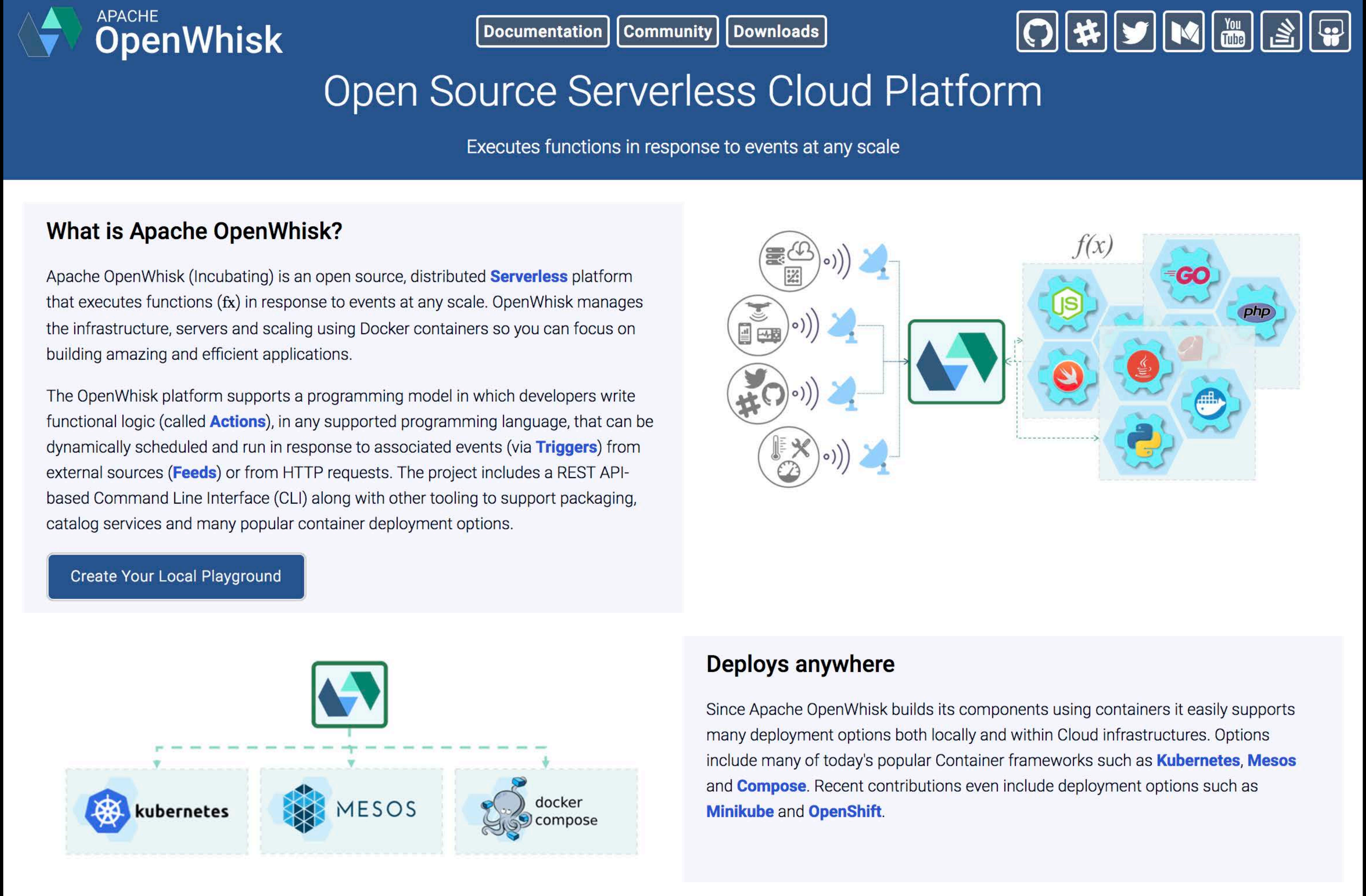
Auto-Tag Function

8 Upload the final image to product catalog

Upload picture

Cast

- **FaaS Platform** - Apache OpenWhisk
  `openwhisk.org`
- **Workflow** - Apache OpenWhisk Composer
  `github.com/ibm-functions/composer`
- **Editing AI Action** - JupyterLab Notebook
  `jupyter.org`
- **Deploying AI Action** - JupyterLab Notebook



Open Source Serverless Cloud Platform

Executes functions in response to events at any scale

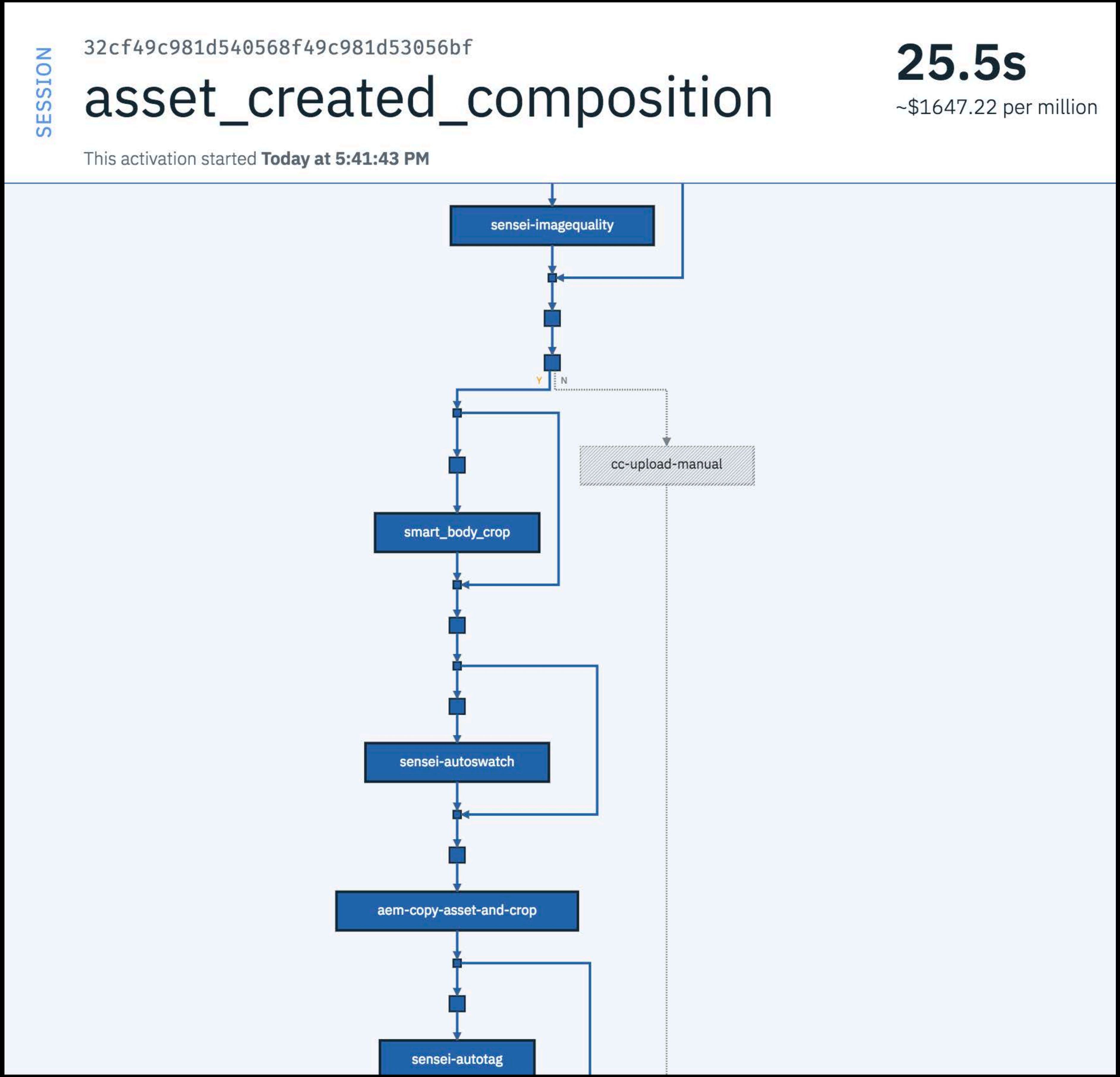
What is Apache OpenWhisk?

Apache OpenWhisk (Incubating) is an open source, distributed **Serverless** platform that executes functions (fx) in response to events at any scale. OpenWhisk manages the infrastructure, servers and scaling using Docker containers so you can focus on building amazing and efficient applications.

The OpenWhisk platform supports a programming model in which developers write functional logic (called **Actions**), in any supported programming language, that can be dynamically scheduled and run in response to associated events (via **Triggers**) from external sources (**Feeds**) or from HTTP requests. The project includes a REST API-based Command Line Interface (CLI) along with other tooling to support packaging, catalog services and many popular container deployment options.

Deploys anywhere

Since Apache OpenWhisk builds its components using containers it easily supports many deployment options both locally and within Cloud infrastructures. Options include many of today's popular Container frameworks such as **Kubernetes**, **Mesos** and **Compose**. Recent contributions even include deployment options such as **Minikube** and **OpenShift**.



SESSION

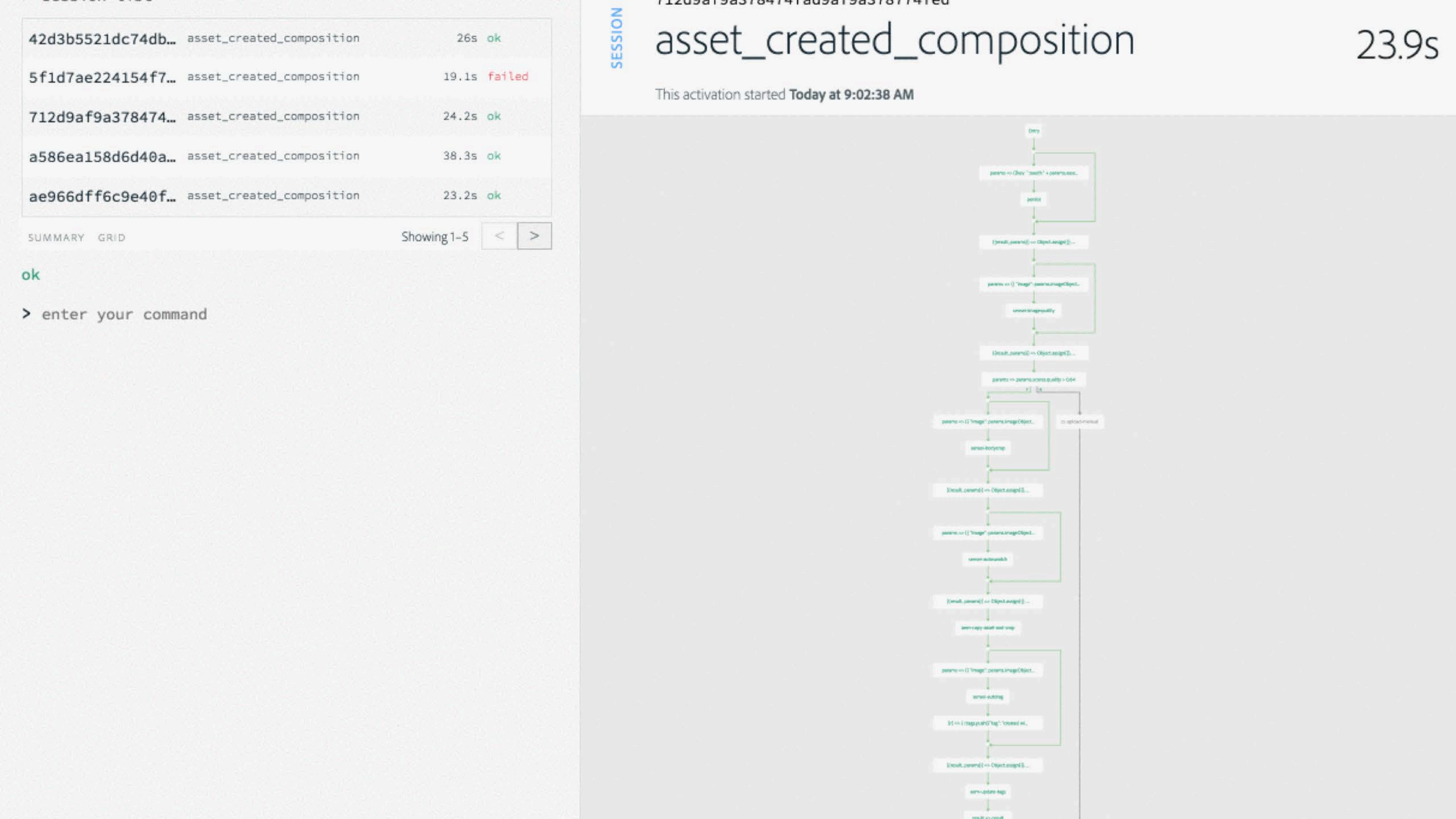
32cf49c981d540568f49c981d53056bf

asset_created_composition

25.5s

~$1647.22 per million

This activation started **Today at 5:41:43 PM**



```
42d3b5521dc74db…  asset_created_composition      26s  ok
5f1d7ae224154f7…  asset_created_composition    19.1s  failed
712d9af9a378474…  asset_created_composition    24.2s  ok
a586ea158d6d40a…  asset_created_composition    38.3s  ok
ae966dff6c9e40f…  asset_created_composition    23.2s  ok
```

```
ok

> enter your command
```

asset_created_composition 23.9s

This activation started **Today at 9:02:38 AM**

Software 2.0

`Model` + `Code`

Software 2.0

JupyterLab - assist in model development

Functions - assist in deploying the model

Software 2.0

ML Engineers collaborate with software developers

Software 2.0

With FaaS it's easy to deploy a new

AI Model-as-a-Function

Conclusions

FaaS platforms are still maturing

Conclusions

FaaS platforms are still maturing

It's **FaaS***ter* to deploy AI models

Conclusions

FaaS platforms are still maturing

It's **FaaS***ter* to deploy AI models

Build **more services**, pay **less**