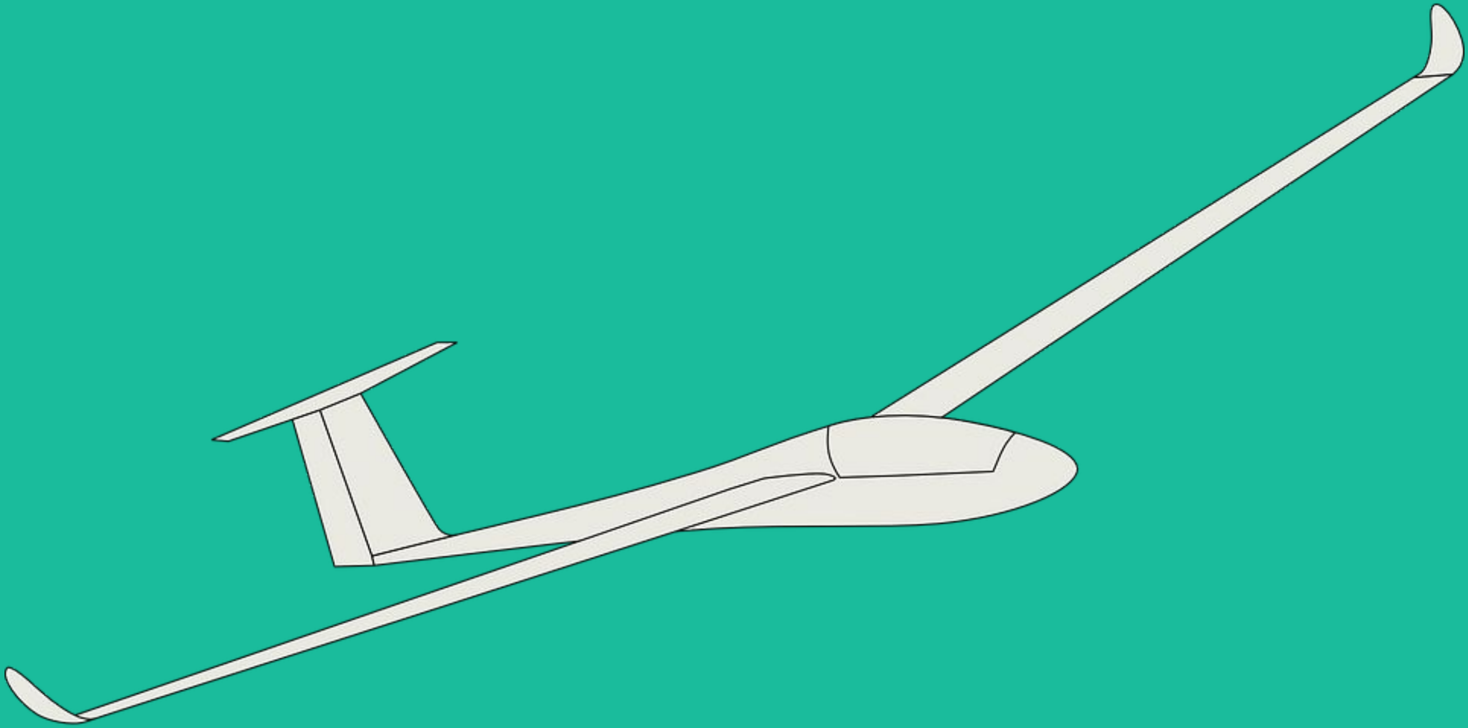
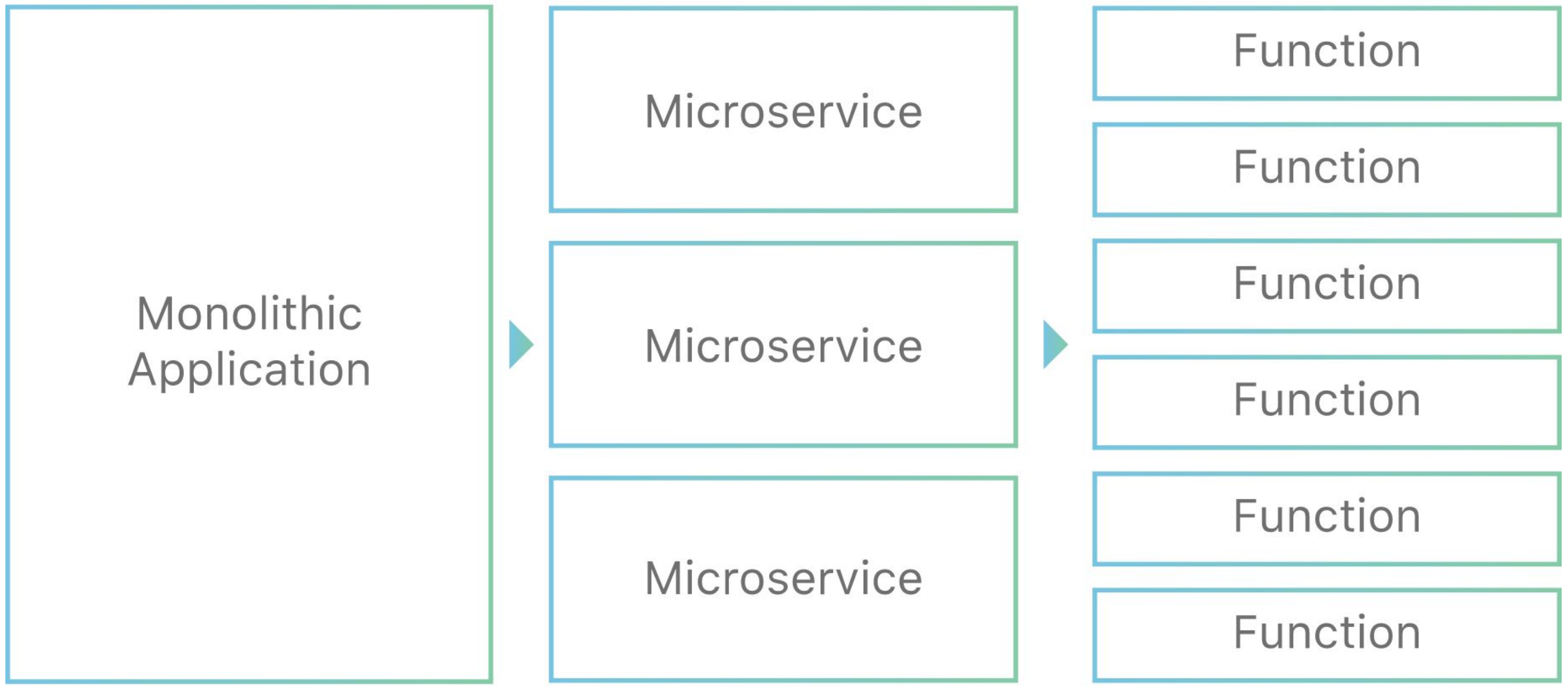


Chapter X-2. Serverless

Bilkent University | CS443 | 2021, Spring | Dr. Orçun Dayıbaş

Serverless

- **Functions**
  - From monoliths to microservices, from VMs to containers
  - Is it all done? No…



Serverless

- **Definition**
  - Serverless is a cloud computing execution model in which the cloud provider runs the server, and dynamically manages the allocation of machine resources. Pricing is based on the actual amount of resources consumed by an application, rather than on pre-purchased units of capacity.
  - Generally, Serverless = FaaS + BaaS
- **Pros & Cons**
  - Scale to **zero/infinite is possible** (Google: “from prototype to production to planet-scale”) → Maximum elasticity
  - Cost-effective since there is **no fee for idle times**.
    - Flip side: infrequently-used code may suffer from greater latency (performance issues)
  - Lack of standards → vendor lock-in

Serverless

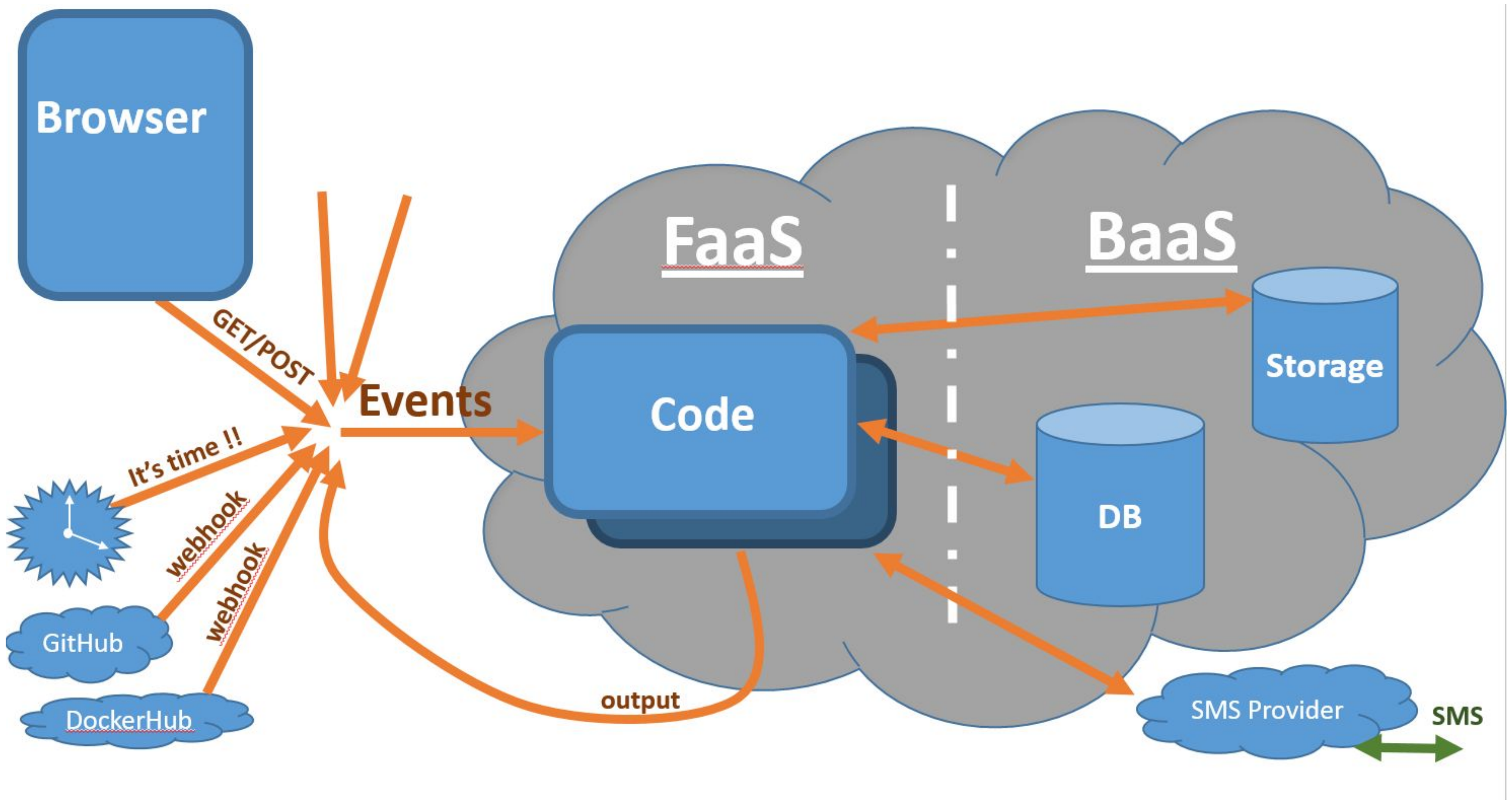
- **FaaS**
  - FaaS is a form of serverless computing, where you execute certain functions of your app. in a abstracted computing env.
  - Developers deploy an individual “function” (a piece of business logic). They are expected to start within milliseconds and process individual requests and then the process ends.
  - Major providers (see http://serverlesscalc.com/)
    - AWS Lambda (Amazon)
    - Azure Functions (Microsoft)
    - Cloud Functions (Google)
    - IBM Cloud Functions (IBM)
    - Pivotal Function Service (VMware)
  - PaaS vs. FaaS
    - PaaS simplifies dev./deployment process of applications and they run on server like a typical application once they deployed
    - FaaS provides the ability to deploy a single function (part of application) and **scale to zero** is possible since the rest is managed by the provider

Serverless

- **BaaS (or MBaaS)**
  - BaaS is a model for providing web/mobile app developers with a way to link their applications to backend cloud storage and APIs exposed by backend applications while also providing features such as user management, push notifications, and integration with some services (via SDK/API).
  - Major providers
    - AWS Amplify (Amazon)
    - Firebase (Google)
    - Azure Mobile Apps (Microsoft & Xamarin)
    - Mobile Cloud Service (Oracle)
    - Mobile Application Platform (Red Hat)
  - In basic terms, BaaS/MBaaS is the **use of 3rd party services/applications (in the cloud)** to handle the server-side logic and state.
  - https://aws.amazon.com/serverless/

Serverless

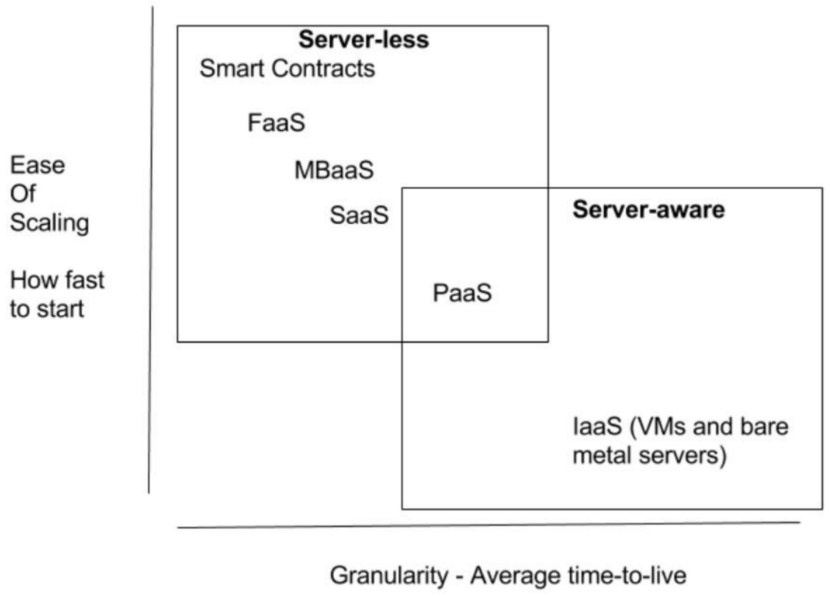
- **Event-driven Architecture**

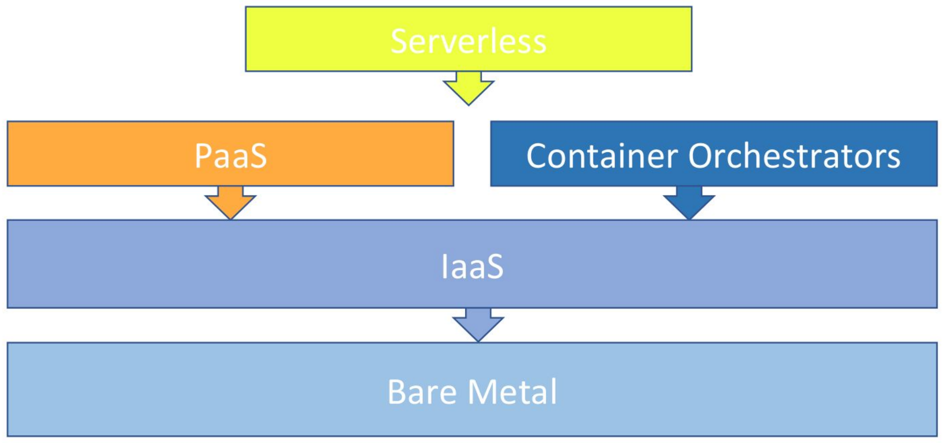


source: Cloud Computing resources abstraction in FaaS env. / Daniele Spiga / SOSC 2019

Serverless

- **Use cases**
  - Serverless platforms are for **short-running, stateless computation and event-driven applications** which scales up and down instantly and automatically.
  - General characteristics
    - latency tolerant, event-driven, short-lived, periodic
    - simply, where it doesn't make sense to pay for always-on services

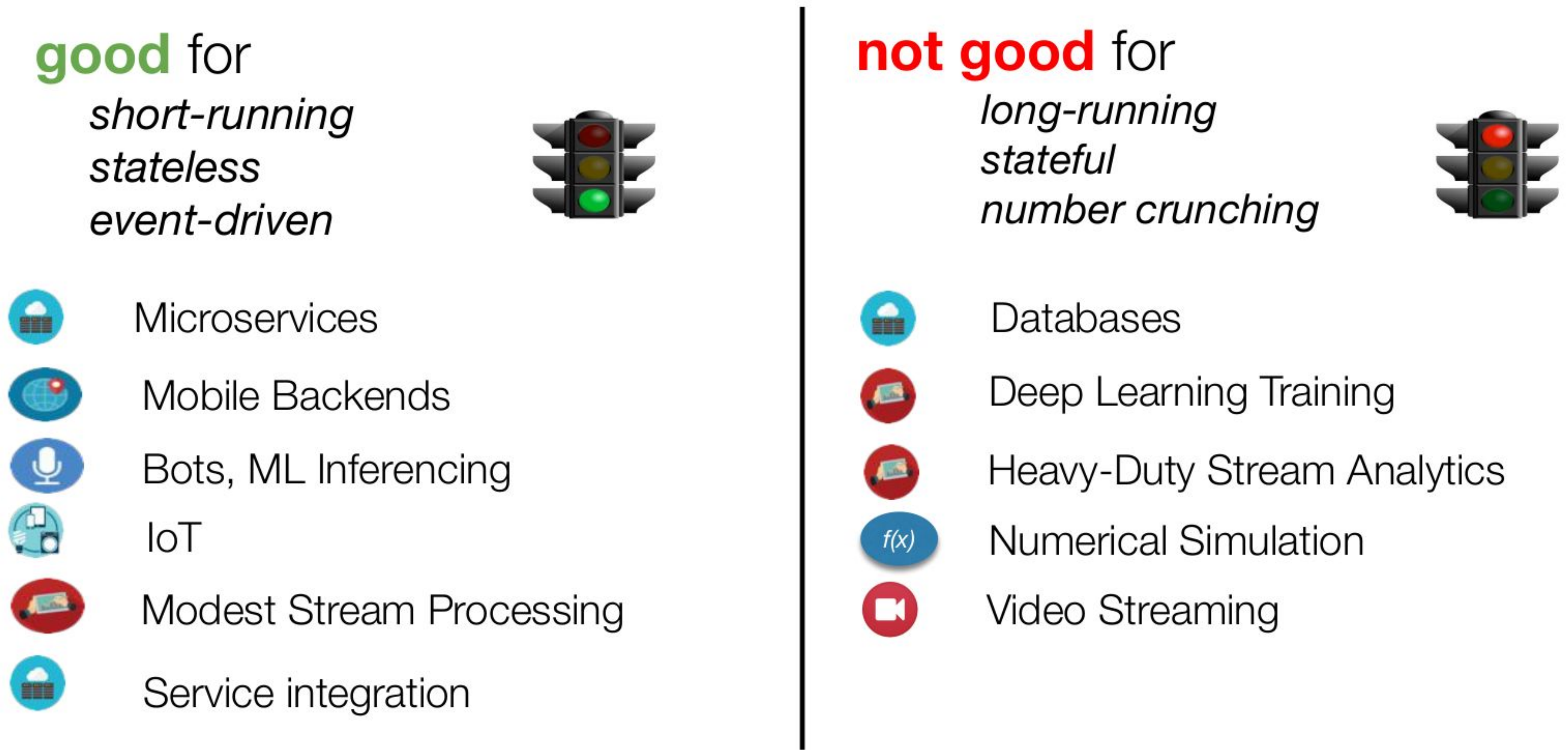




source: Cloud Computing resources abstraction in FaaS env. / Daniele Spiga / SOSC 2019

Serverless

- **Use cases**



good for

short-running
stateless
event-driven

- Microservices
- Mobile Backends
- Bots, ML Inferencing
- IoT
- Modest Stream Processing
- Service integration

not good for

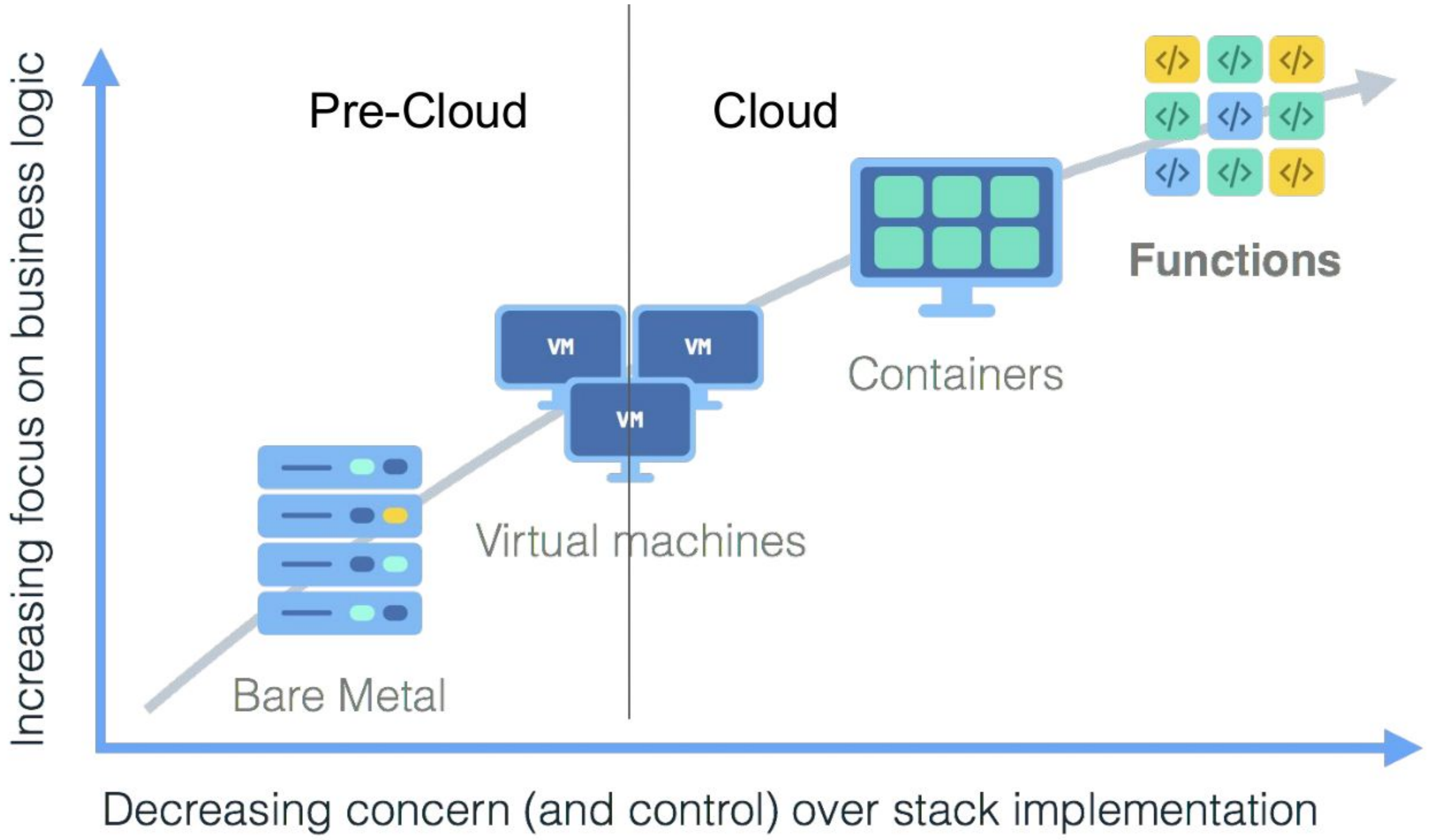
long-running
stateful
number crunching

- Databases
- Deep Learning Training
- Heavy-Duty Stream Analytics
- Numerical Simulation
- Video Streaming

source: Cloud Computing resources abstraction in FaaS env. / Daniele Spiga / SOSC 2019

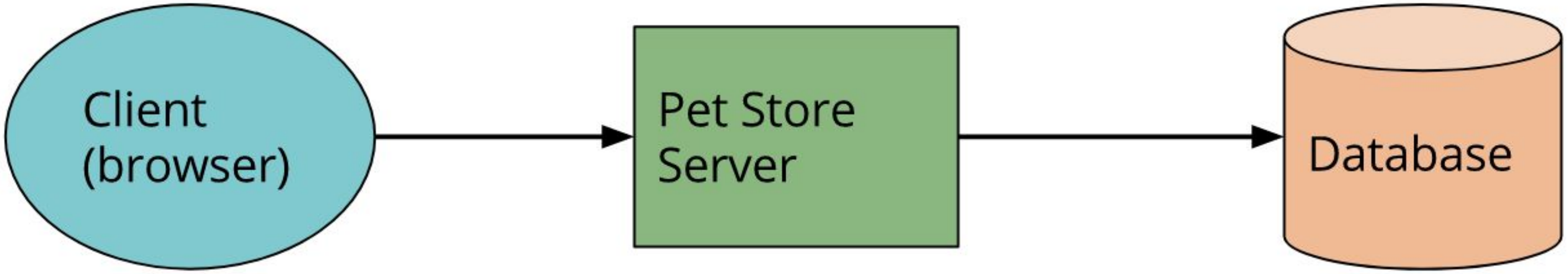
Serverless

- **Focus: Business logic vs. Tech stack**


Increasing focus on business logic
Pre-Cloud
Cloud
</>
Functions
Containers
VM
VM
VM
Virtual machines
Bare Metal
Decreasing concern (and control) over stack implementation

Serverless

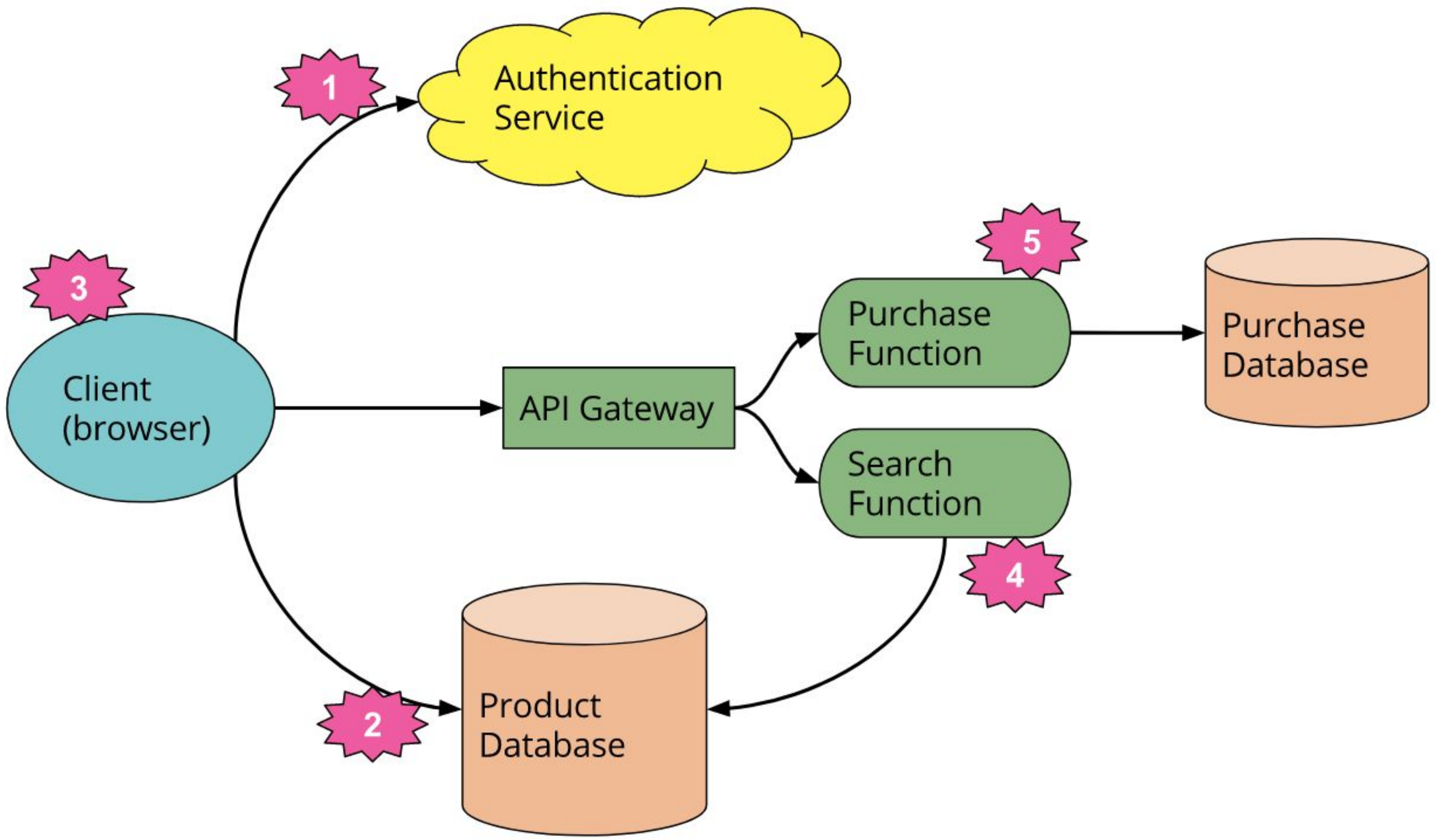
- **Example-1**
  - Pet Store (Naive)



source: https://martinfowler.com/articles/serverless.html

Serverless

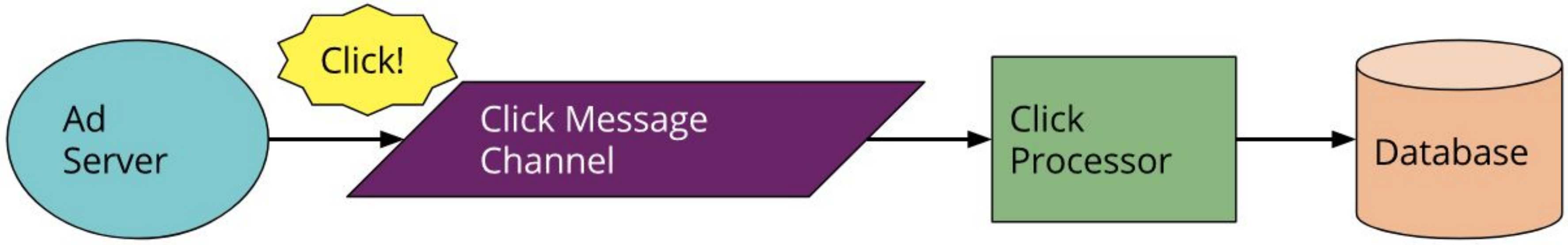
- **Example-1**
  - Pet Store (Serverless)



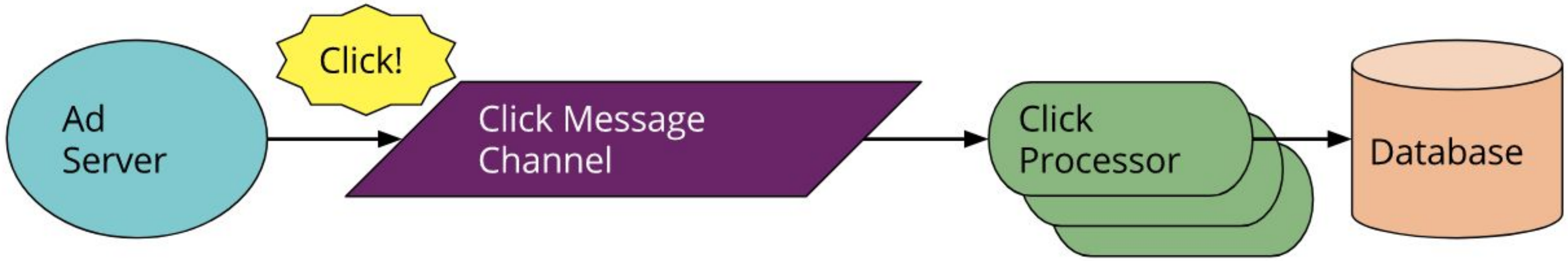
source: https://martinfowler.com/articles/serverless.html

Serverless

- **Example-2**
  - A message-driven application (naive)



  - A message-driven application (serverless)



source: https://martinfowler.com/articles/serverless.html

Serverless

- **Implementations**

APACHE OpenWhisk

AWS Lambda

OpenLambda

Iron.io

Azure Functions

fission

Kubernetes

IBM Cloud Functions

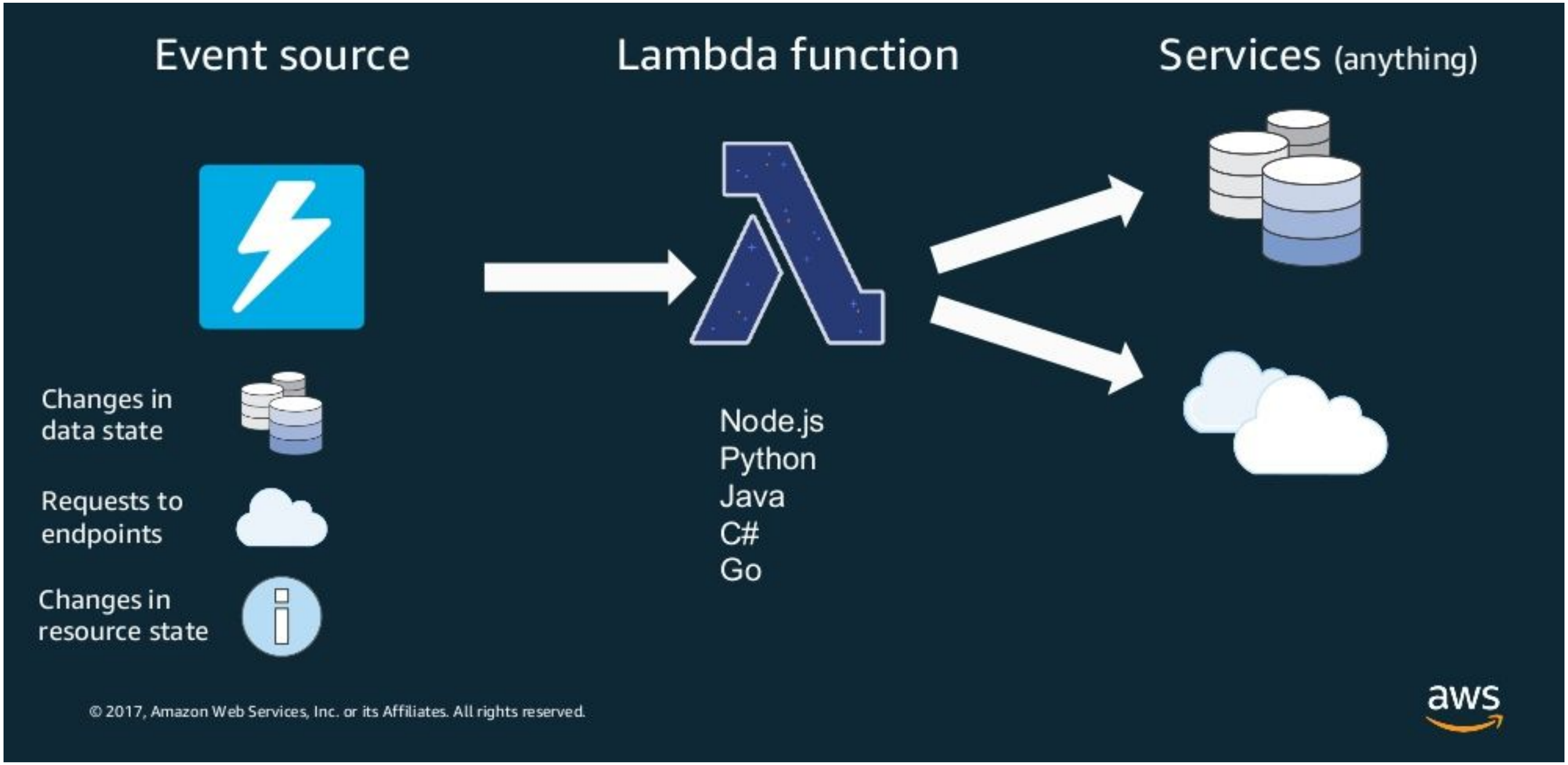
Red-Hat

Google Functions

fn

Serverless

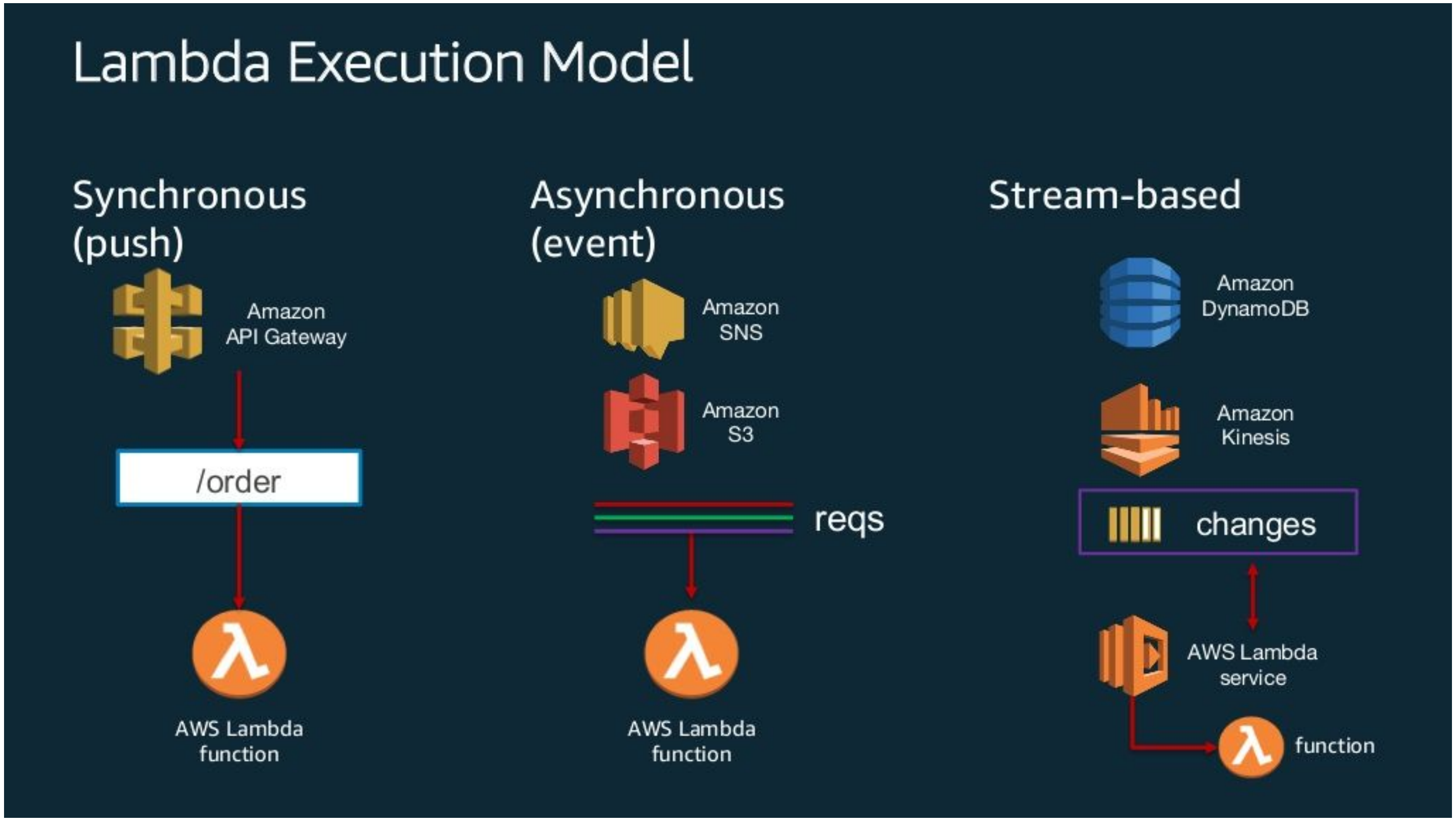
- **AWS Lambda**



source: https://www.slideshare.net/AmazonWebServices/introduction-to-serverless-106198220

Serverless

- **AWS Lambda**
  - Serverless App. Model: https://aws.amazon.com/serverless/sam/



source: https://www.slideshare.net/AmazonWebServices/introduction-to-serverless-106198220

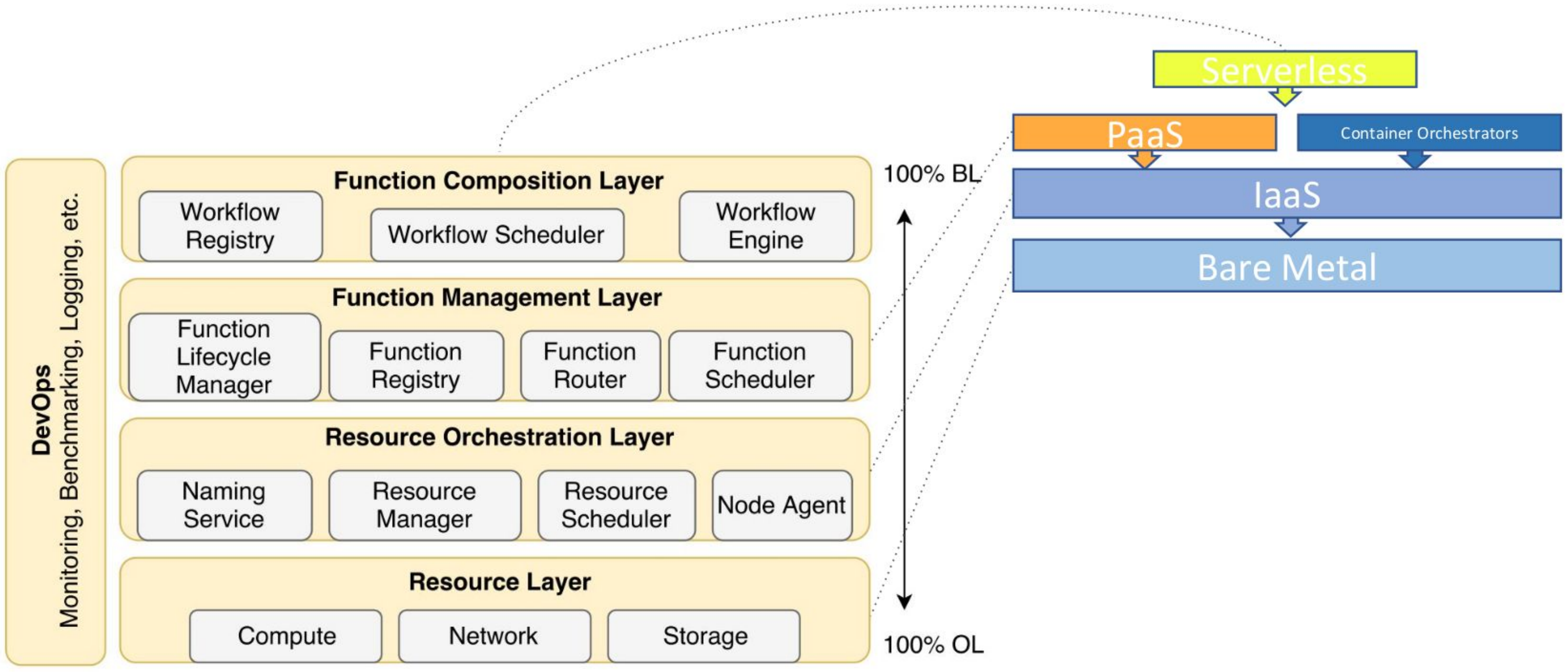
Serverless

- **Benefits**
  - Reduced operational cost
  - Easier operational management
  - Time to market and continuous experimentation
  - Greener computing

- **Drawbacks**
  - Execution duration limits
  - Startup latency & cold starts
  - Security concerns
  - Repetition of logic across client platforms
  - Vendor lock-in

Serverless

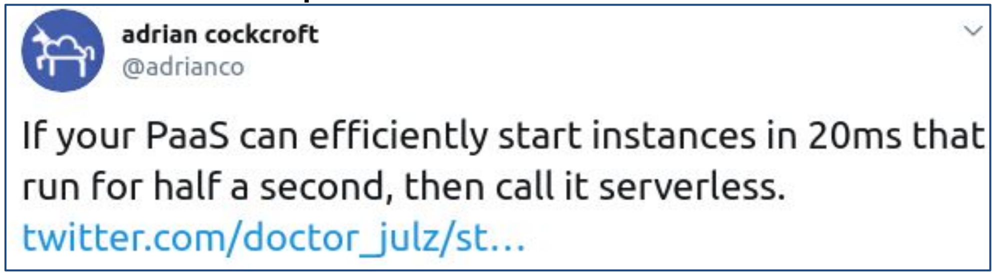
- **Under the hood**



source: Cloud Computing resources abstraction in FaaS env. / Daniele Spiga / SOSC 2019

Serverless

- **Recap**
  - Beware hype & marketing, it's not a magic (like anything else)
    - In the end, the container is created and destroyed by algorithms used in FaaS platforms and the operational team have no control over that.

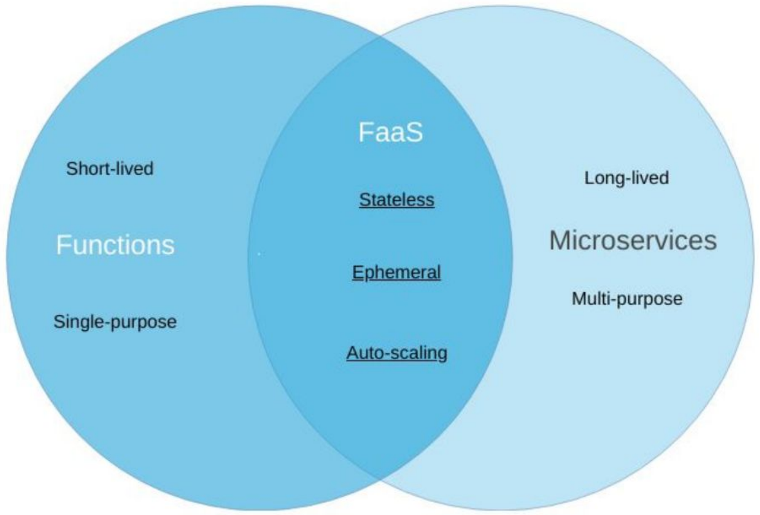


adrian cockcroft
@adrianco

If your PaaS can efficiently start instances in 20ms that run for half a second, then call it serverless.
twitter.com/doctor_julz/st...

(link)

  - It is not a general purpose solution
    - FaaS & Microservices will co-exist
    - Not for everyone (see use cases)
  - Standardization is important
    - CNCF Serverless Work Group
      - https://github.com/cncf/wg-serverless/
      - Meeting minutes



Q/A