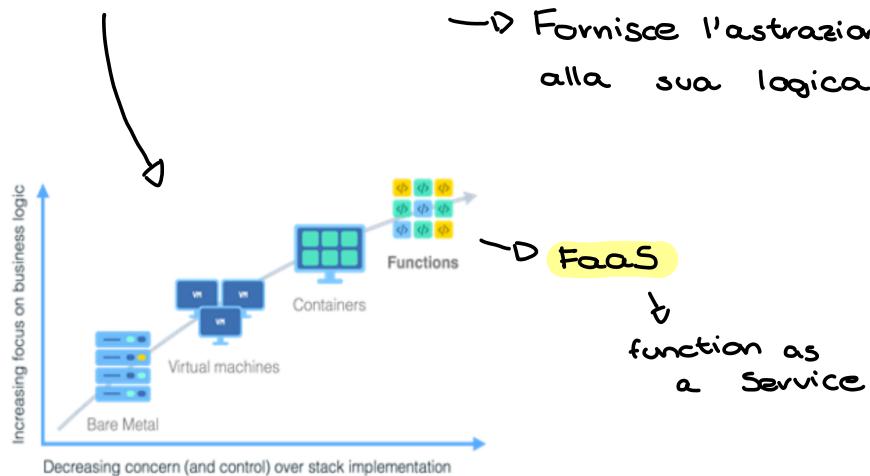


## → D Serverless Computing

→ Fornisce l'astrazione delle risorse di calcolo alla sua logica



### → D 1° Caso:

- Serverless è stato utilizzato per la prima volta per descrivere applicazioni che significativamente o completamente incorporano

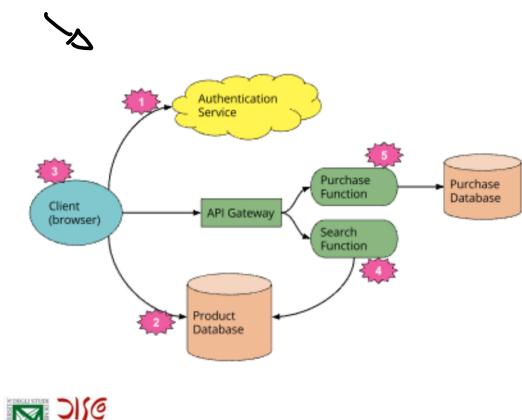
→ (BaaS) Backend as a service

### → D 2° Caso:

- Serverless può anche significare applicazioni in cui la logica del server è eseguita in container, attivati da eventi, effimeri, gestiti da terzi

→ FaaS Function as a Service

## → Architettura



- Authentication as a Service. (BaaS)
- Direct access to a Database (e.g. Firebase)
- Client is much more intelligent
- Purchase and Search functionality are splitted
- Instead of having an always-running server, as existed in the original architecture FaaS functions respond to HTTP requests

→ "AWS Lambda lets run code without provisioning or managing servers" ↴

FaaS, non si deve controllare il server e le proprie long-lived applicazioni

→ "You pay only for the compute time you consume"

→ "With Lambda, you can run code for virtually any type of application" ↴ Unix Process

→ "Just upload your code and Lambda takes care of everything required to run"

→ "and scale your code with high availability"

↳ automatic scalability

→ "you can set up your code to automatically trigger from other AWS services"

↳ eventi

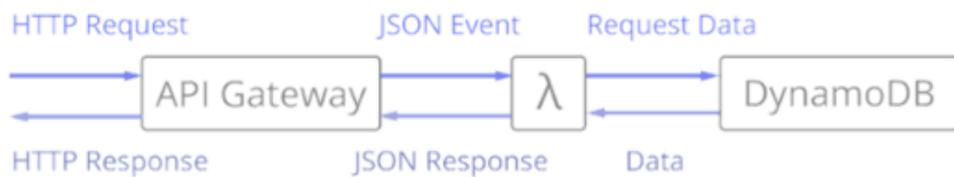
→ "or call it directly from any web or mobile app"

↳ http request

→ 5 minutes max!

→ Stateless, ma se si vuole la persistenza bisogna "externalized" fuori dalle istanze FaaS

## → Api Gateway



**Serverless is good for**  
*short-running  
stateless  
event-driven*



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

**Serverless is not good for**  
*long-running  
stateful  
number crunching*



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

Il "Server" c'è ma è spento, quando arriva una richiesta AWS Lambda o concorrenti, avviano una istanza dell'applicazione. Nel caso fosse già avviata attiva solo il processo.

→ local state → non garantisce che lo stato sia persistente "across multiple invocations"

→ externalized → per avere la persistenza

→ ↵ operations cost, development cost, execution cost,  
greener computing