

D E V C O N F

DEATH

TO THE
SERVER

REUBEN KRUGER

Investec *Developer*

2024

WHAT IS SERVERLESS?

- Serverless Computing: Cloud model allocating resources on demand.

.....

WHAT IS SERVERLESS?

- Serverless Computing: Cloud model allocating resources on demand.
- Provider Responsibility: Manages servers, freeing customers from maintenance.

.....

WHAT IS SERVERLESS?

- Serverless Computing: Cloud model allocating resources on demand.
- Provider Responsibility: Manages servers, freeing customers from maintenance.
- Misnomer Clarified: "Serverless" still involves servers to execute developer code.

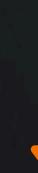
.....

PILLARS OF SERVERLESS

Investec *Developer*



ZERO
INFRASTRUCTURE
MANAGEMENT



Developers are not involved in infrastructure setup and management

PILLARS OF SERVERLESS

Investec *Developer*



ZERO
INFRASTRUCTURE
MANAGEMENT



EVENT DRIVEN
DESIGN

Developers are not involved in infrastructure setup and management

Serverless apps are event driven by nature

PILLARS OF SERVERLESS

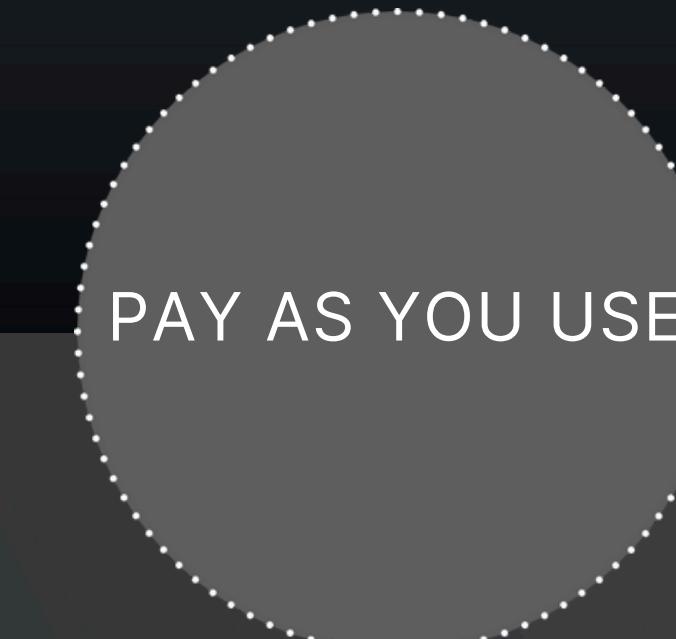
Investec *Developer*



ZERO
INFRASTRUCTURE
MANAGEMENT



EVENT DRIVEN
DESIGN



PAY AS YOU USE

Developers are not involved in infrastructure setup and management

Serverless apps are event driven by nature

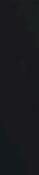
You only pay for what you use, consumption based pricing.

PILLARS OF SERVERLESS

Investec *Developer*



ZERO
INFRASTRUCTURE
MANAGEMENT



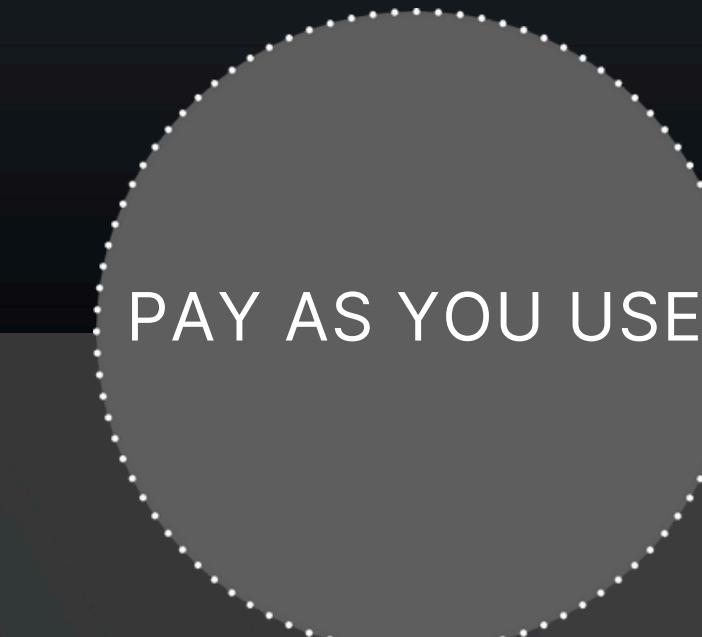
Developers are not involved in infrastructure setup and management



EVENT DRIVEN DESIGN



Serverless apps are event driven by nature



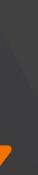
PAY AS YOU USE



You only pay for what you use, consumption based pricing.



EFFORTLESS SCALE



Scaling is managed by the underlying provider

SELF MANAGED INSTANCES

Investec *Developer*



Owned hardware

Physical boxes maintained
by you

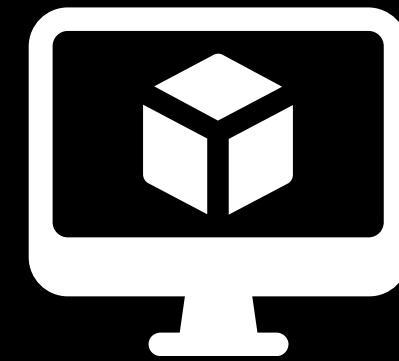
SELF MANAGED INSTANCES

Investec *Developer*



Owned hardware

Physical boxes maintained
by you



Virtual Machines

Complex to maintain

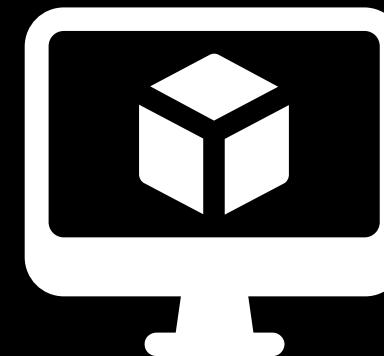
SELF MANAGED INSTANCES

Investec *Developer*



Owned hardware

Physical boxes maintained by you



Virtual Machines

Complex to maintain



Traditional webapps

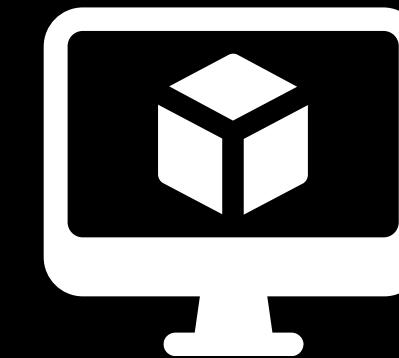
SELF MANAGED INSTANCES

Investec *Developer*



Owned hardware

Physical boxes maintained by you



Virtual Machines

Complex to maintain



Traditional webapps



Containers

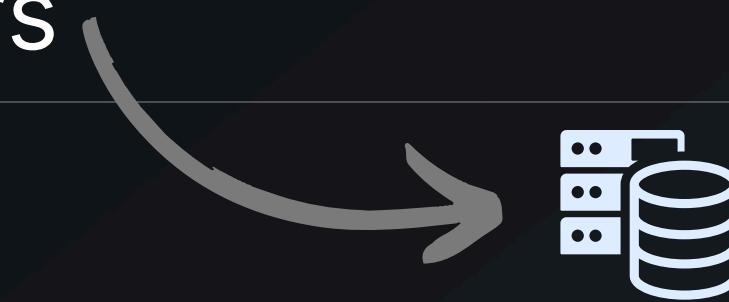
Can get expensive and complex depending on how its hosted

EXPENSIVE!!

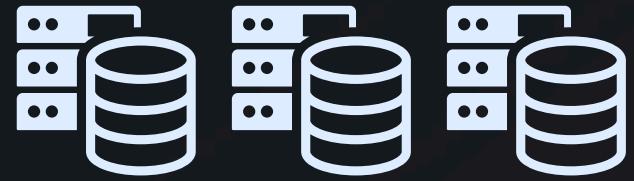
- Serverless
- Virtual machines
- Webapps
- Kubernetes clusters



Traditional
Servers



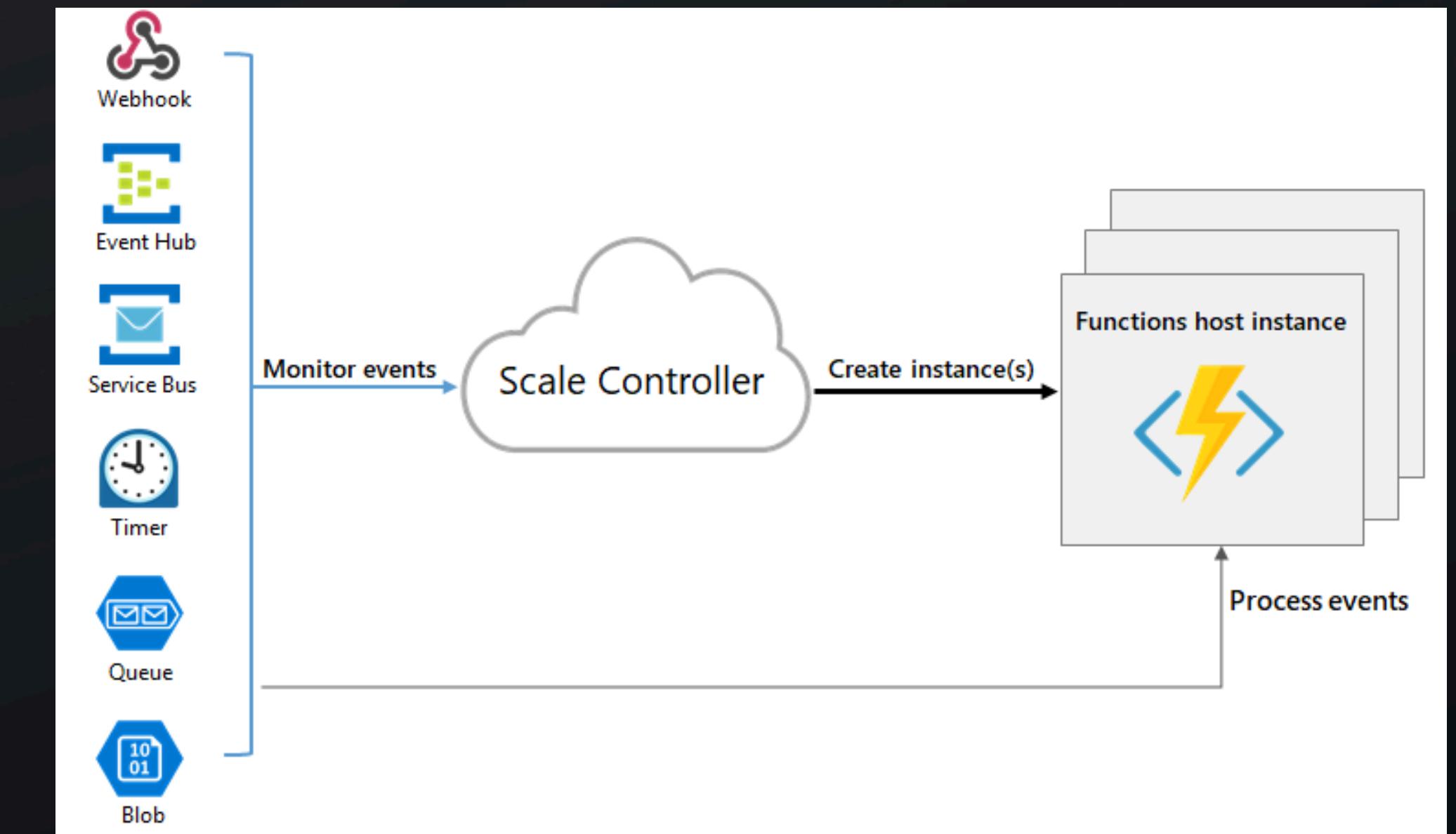
Investec *Developer*





SCALE

How easy is it to scale serverless?



WHERE TO START?



AZURE resources

Azure is owned by Microsoft corp



AWS resources

AWS is owned by Amazon

Azure Container apps/functions

Azure functions are free for life < 1 Mil req/month
Container apps Free for first 2 Mil req/Month

Azure Servicebus/Event grid

Event grid first 100000 events free
Servicebus 750 Unit hours free in free tier

Azure CosmosDB

first 1000 RU/s and 25 GB of storage in the account for free

AWS Lambda

1 million free requests per month

AWS SQS

First 1 Mil req/Month free

AWS DynamoDB

First 25 GB of storage in the account for free

Architecting serverless software

Traditional



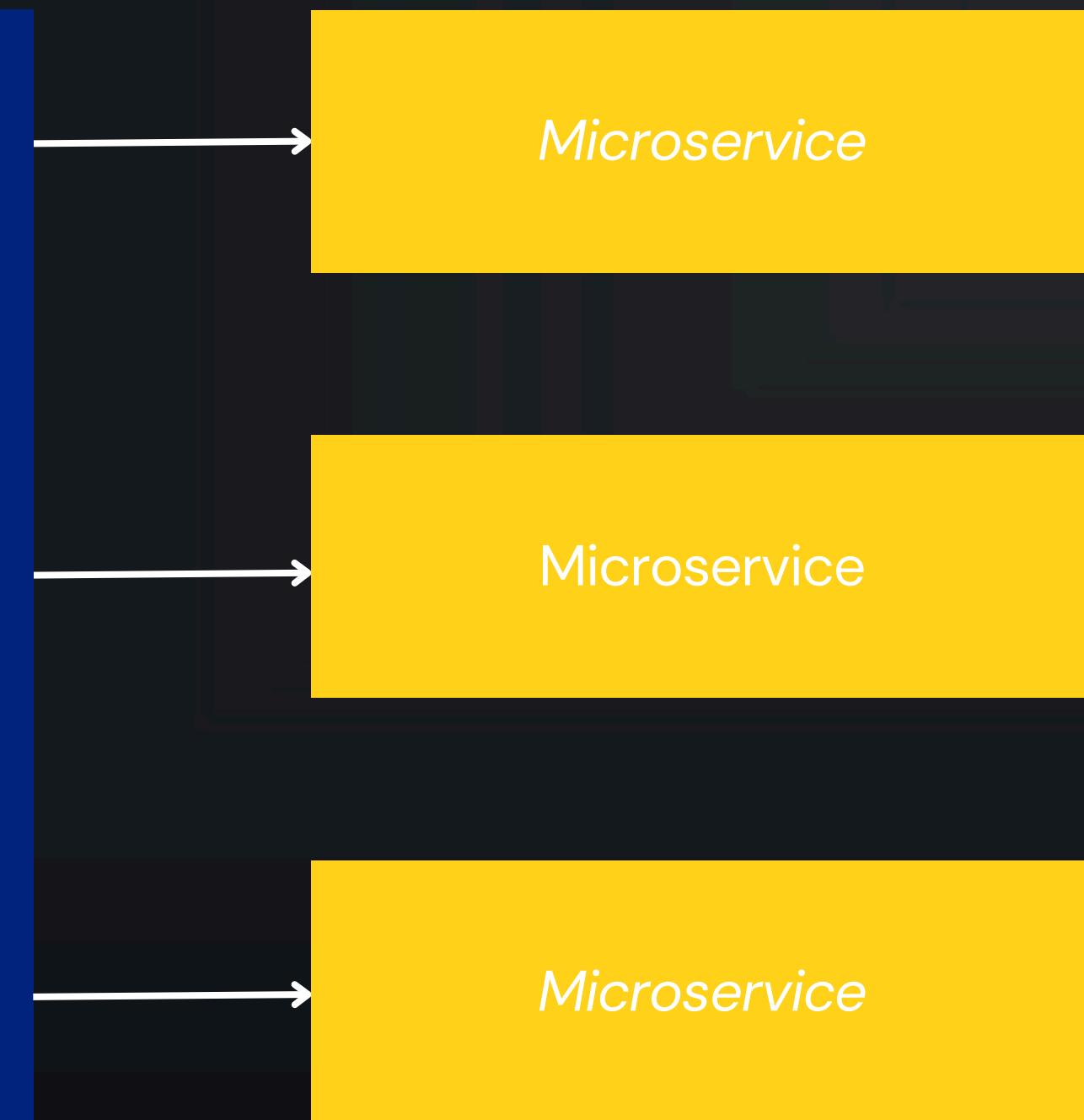
Monolith

Architecting serverless software

Traditional



Microservice



Monolith

Microservice

Microservice

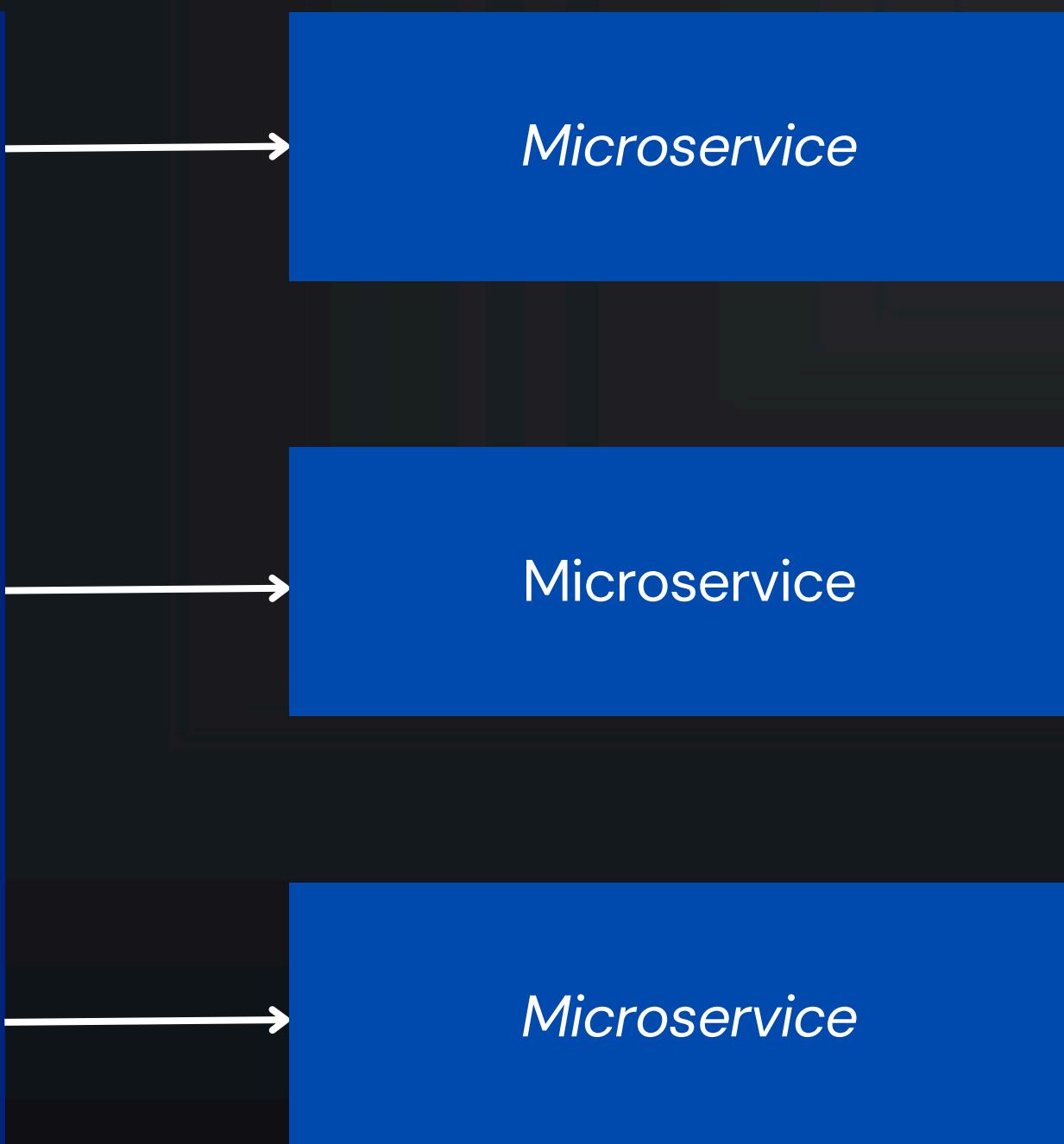
Microservice

Architecting serverless software

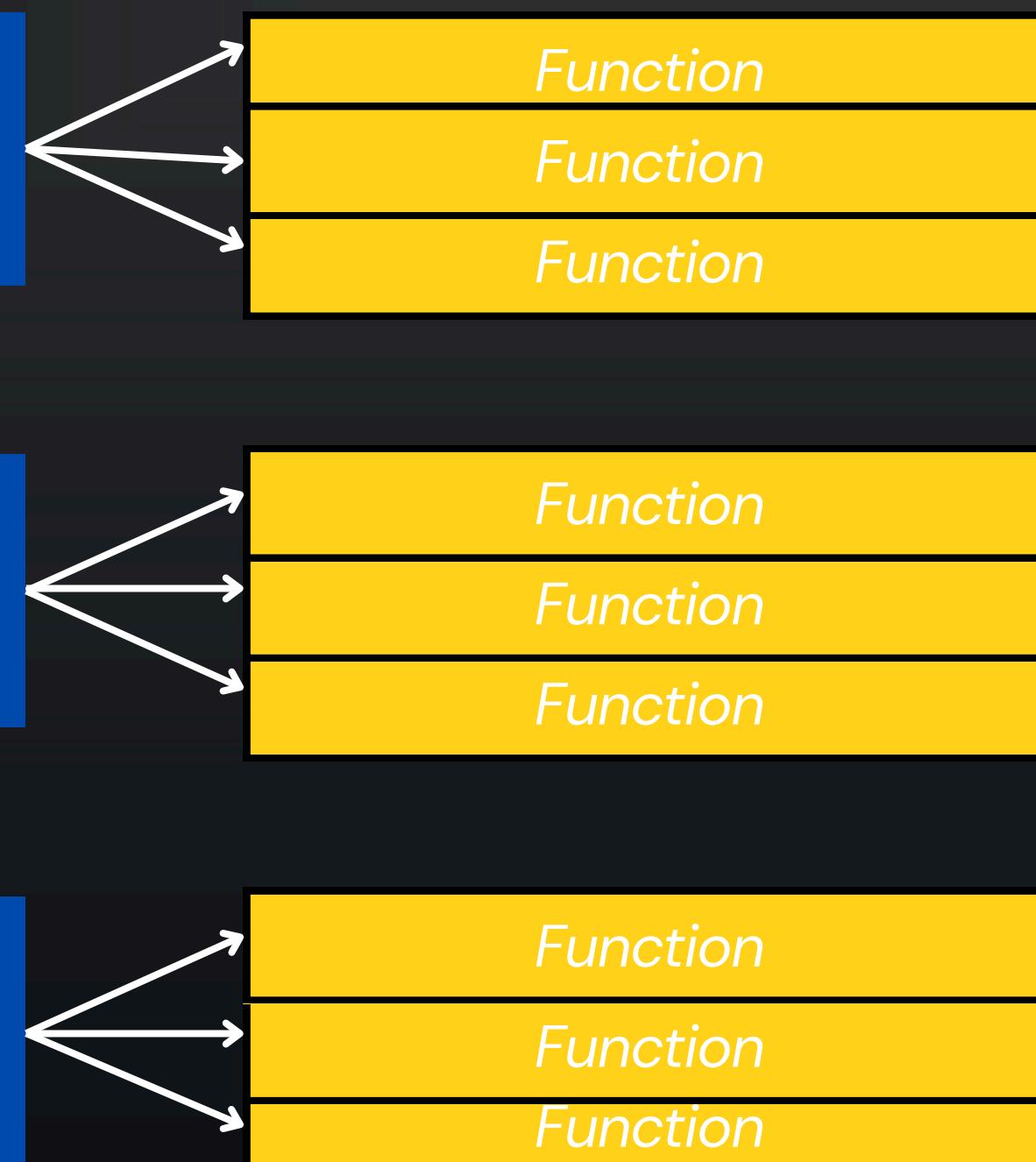
Traditional



Microservice

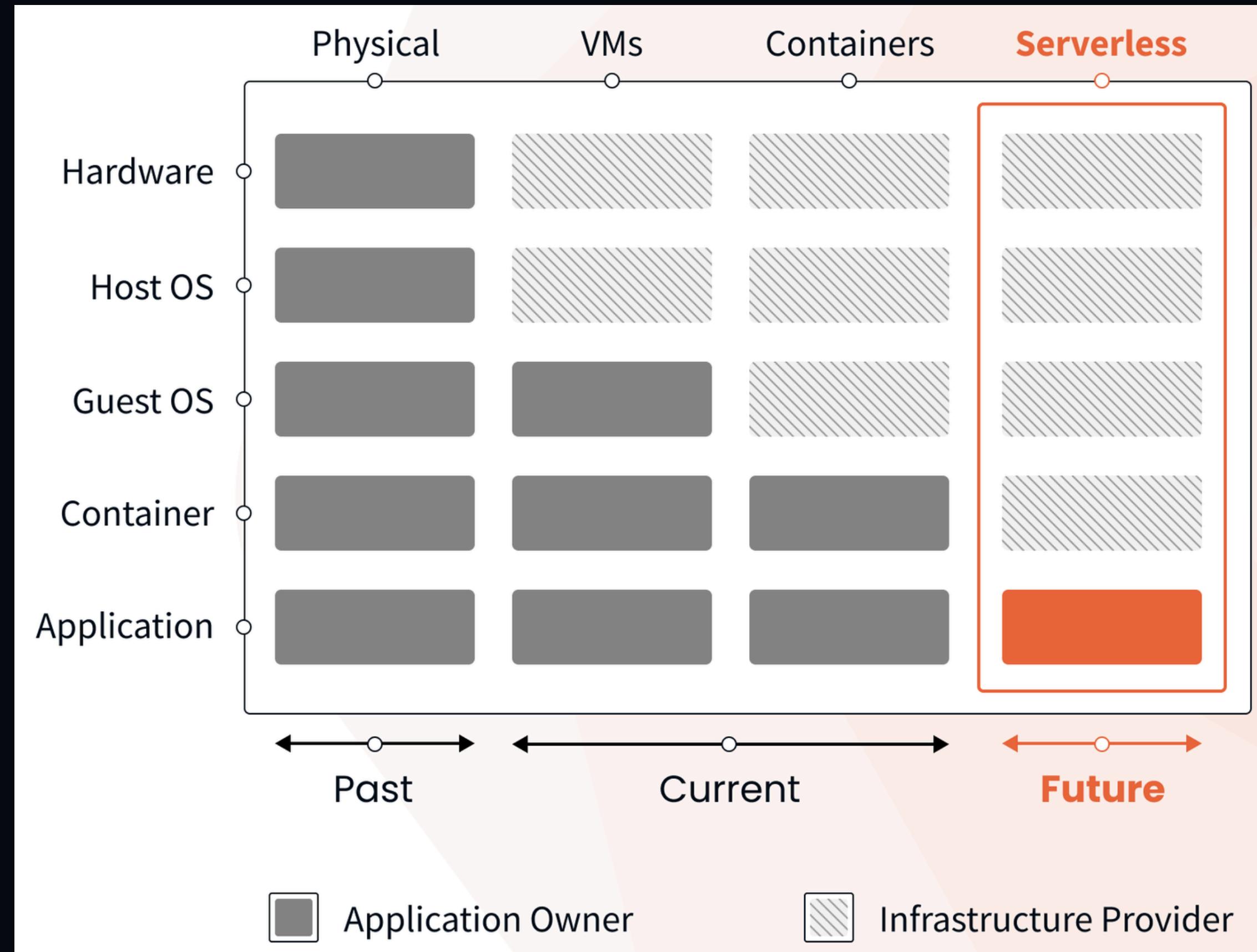


Serverless

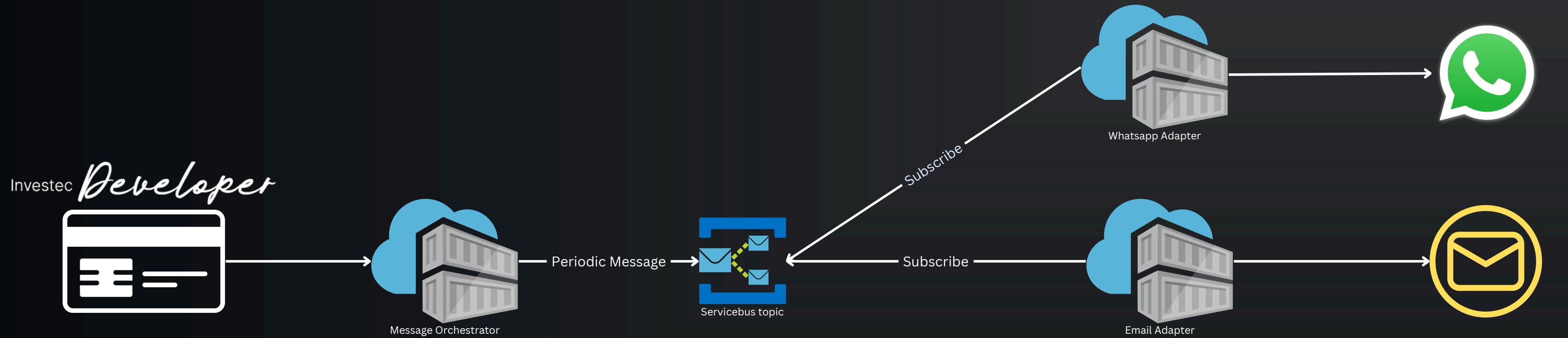


Monolith

Our involvement in the tech

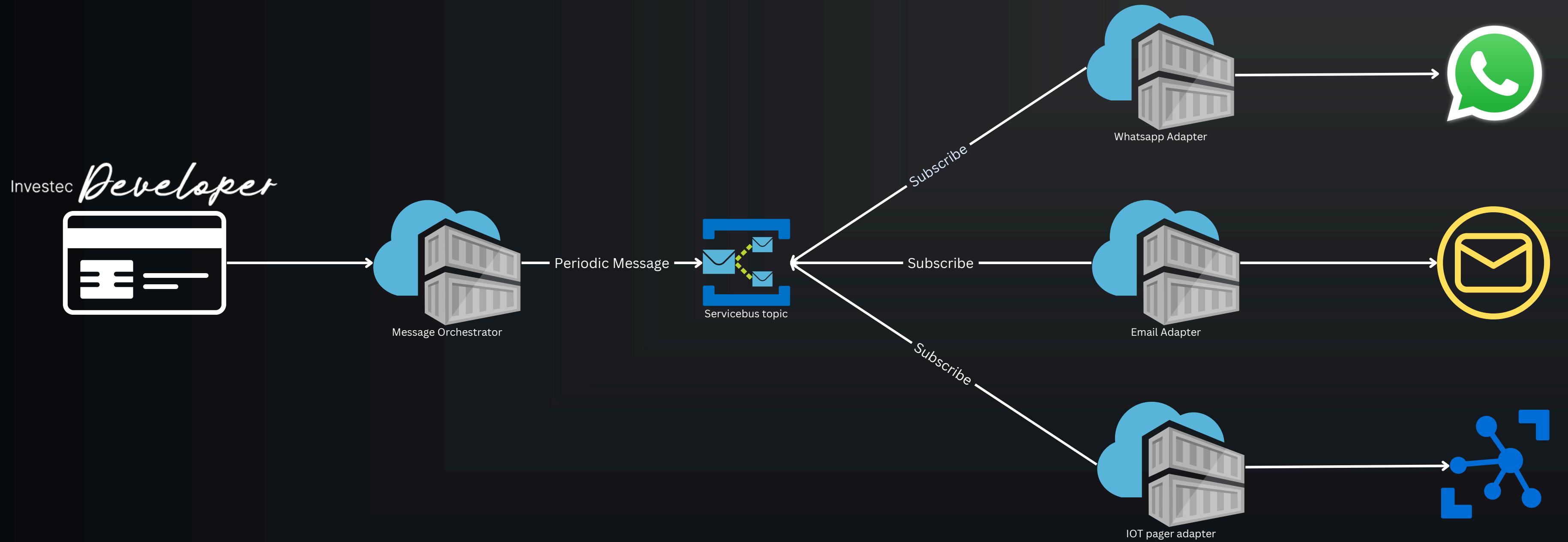


Demo Architecture



Investec *Developer*

Demo Architecture

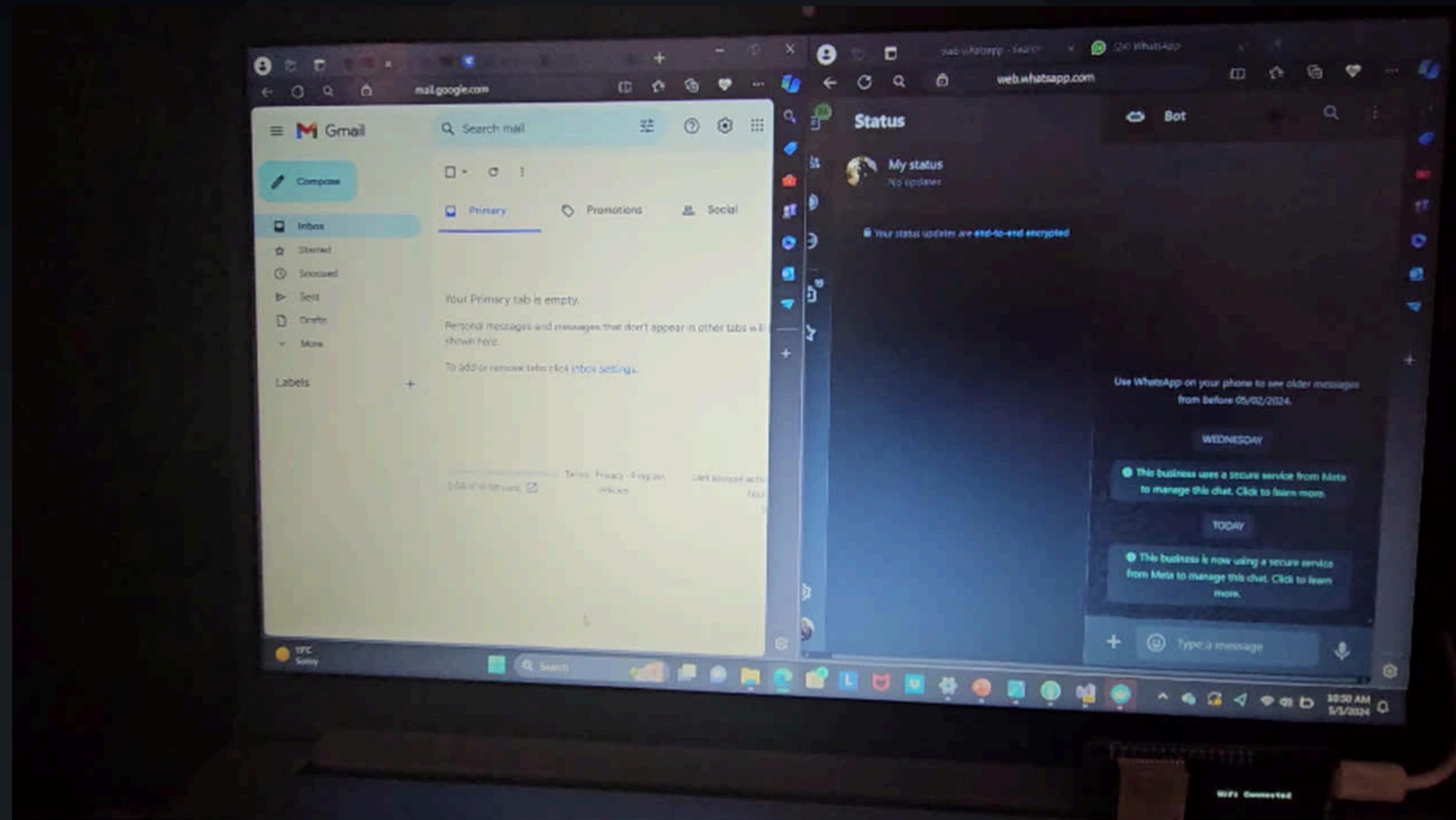


- Develop new adapter
- Create SB Subscriber
- Deploy

Investec *Developer*

DEMO





Programmable card call

```
// It has a limited execution time, so keep any code short-running.  
const beforeTransaction = async (authorization) => {  
    console.log(authorization);  
    return true;  
};  
  
// This function runs after an approved transaction.  
const afterTransaction = async (transaction) => {  
    const response = await fetch('https://serverlessdemo.azure-api.net/submittransact  
        method: 'POST',  
        headers: {  
            'Content-Type': 'application/json'  
        },  
        body: JSON.stringify(transaction)  
    );  
  
    console.log(response);  
};  
  
// This function runs after a declined transaction  
const afterDecline = async (transaction) => {  
    console.log(transaction);  
};
```

QUESTIONS?



