adesso

business.
people.
technology.

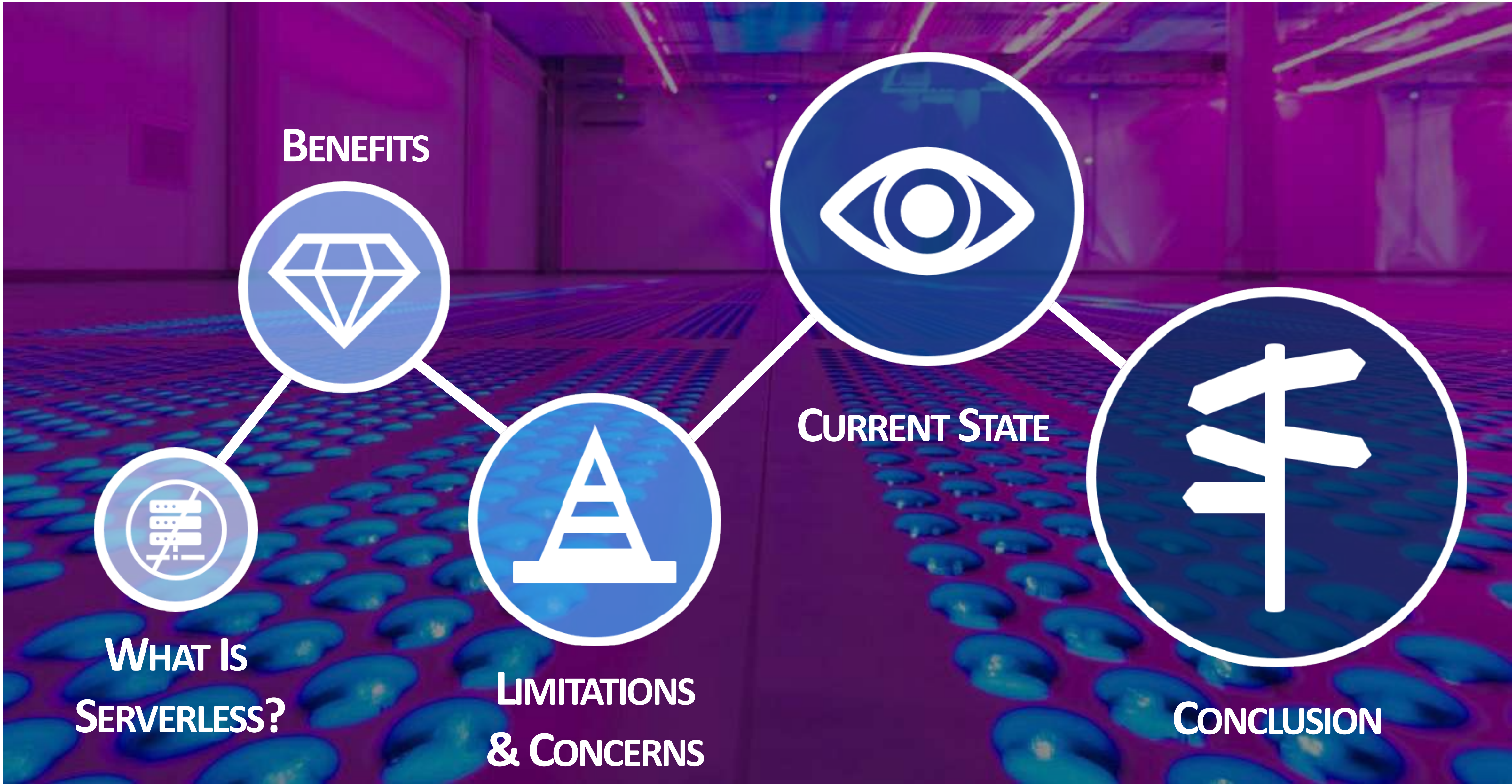
RedHat

The State of Serverless

Red Hat Forum 2019

Dr. Dominik Langer

Zürich, 10th of September 2019



What Is Serverless?

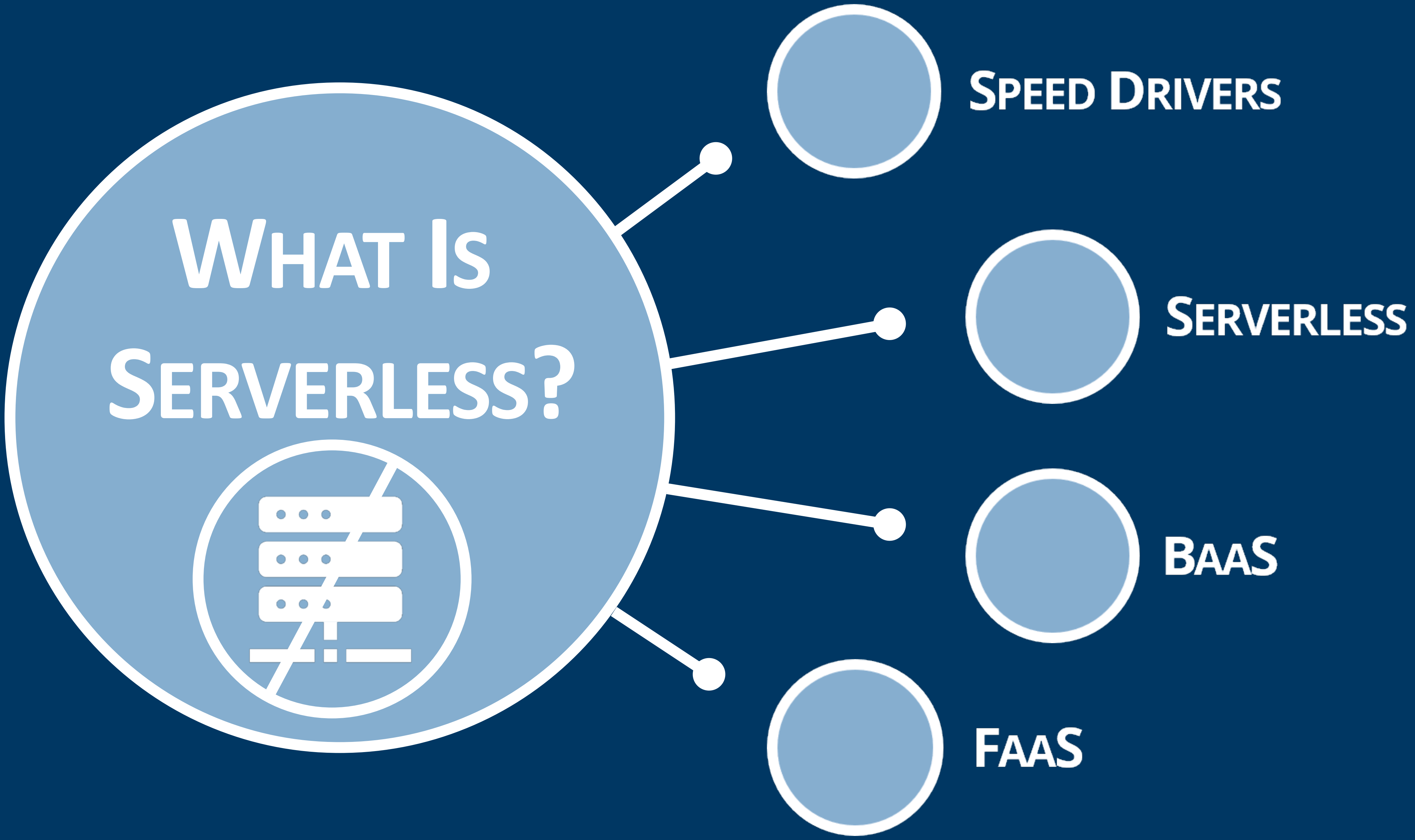
Benefits

Limitations & Concerns

Current State

Conclusion

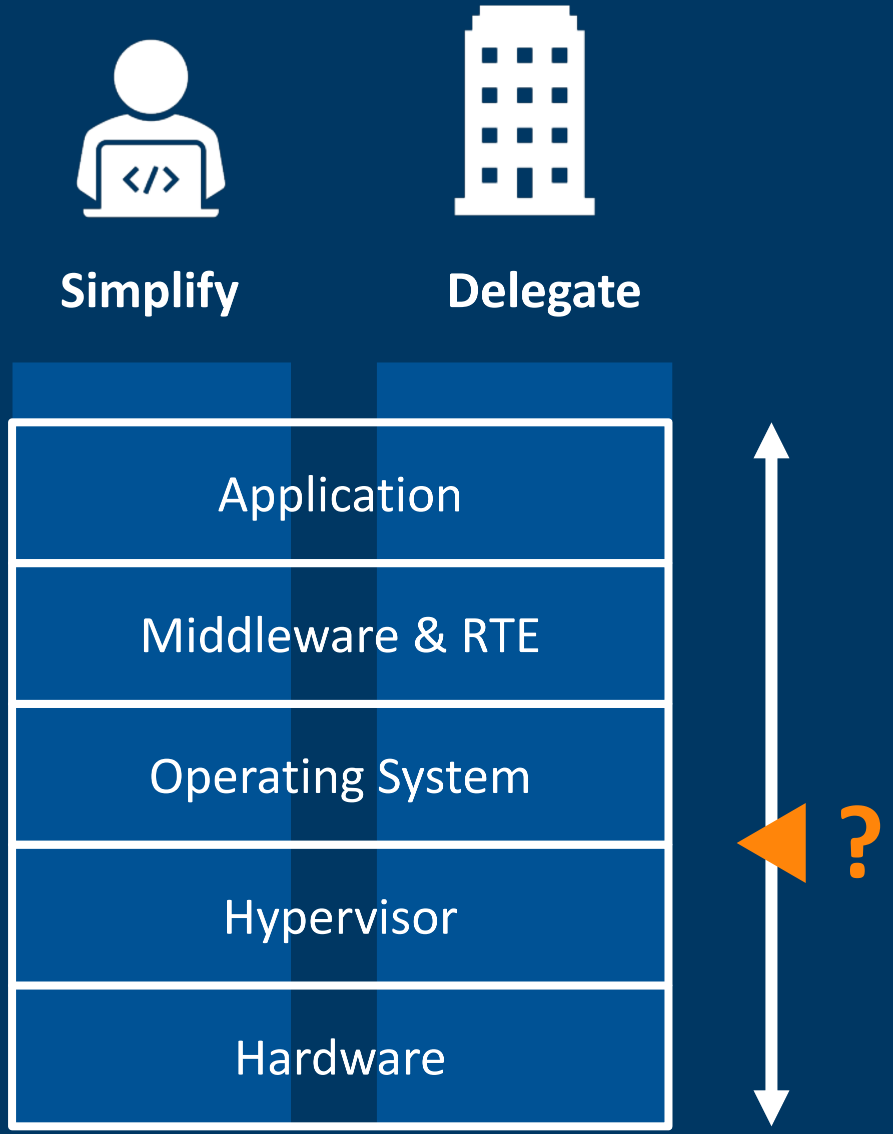
What Is Serverless?



- Speed Drivers
- Serverless
- BaaS
- FaaS

Speed Drivers

Higher levels of abstraction let you focus on differentiation.



SERVERLESS

Serverless abstracts away servers and server processes from developers.

Performance defined not via host size/count

HA

Implicit high availability

Self-autoscale & auto-provision based on load

Cost based on precise usage

Source: CNCF WG-Serverless Whitepaper v1.0 (2018) (2018); What Is Serverless? (Roberts & Chapin, 2017)

BaaS

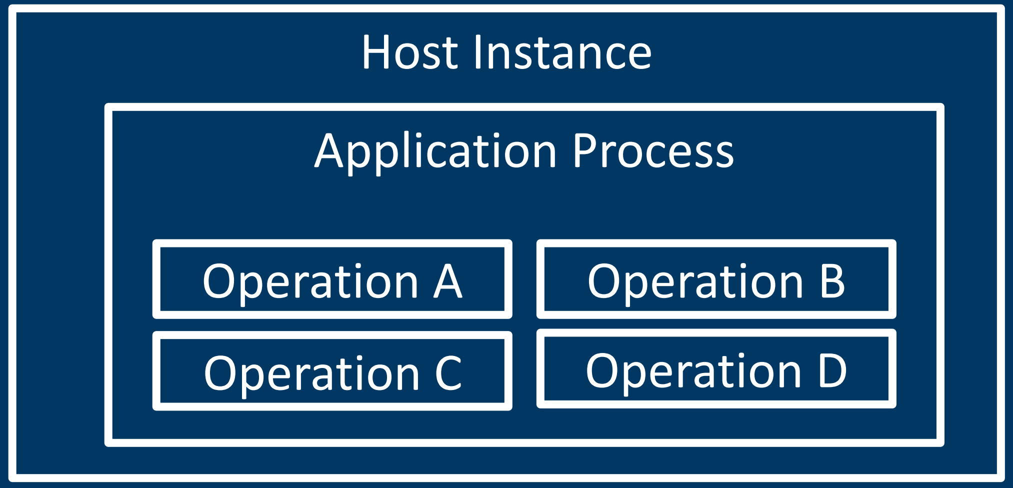
Back-end-as-a-Service are third-party API-based services that replace core subsets of functionality in an application.

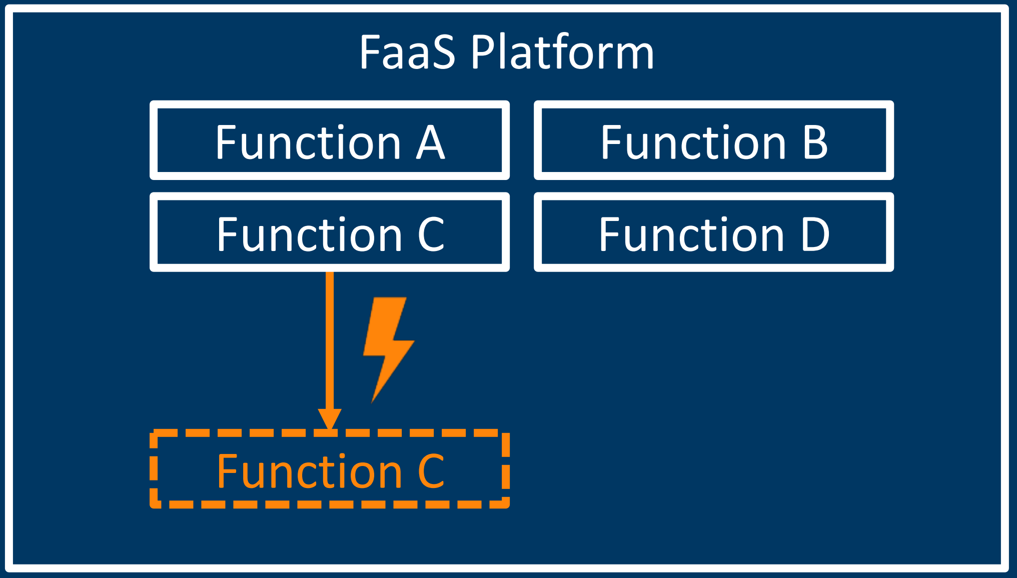
Examples

Source: CNCF WG-Serverless Whitepaper v1.0 (2018)

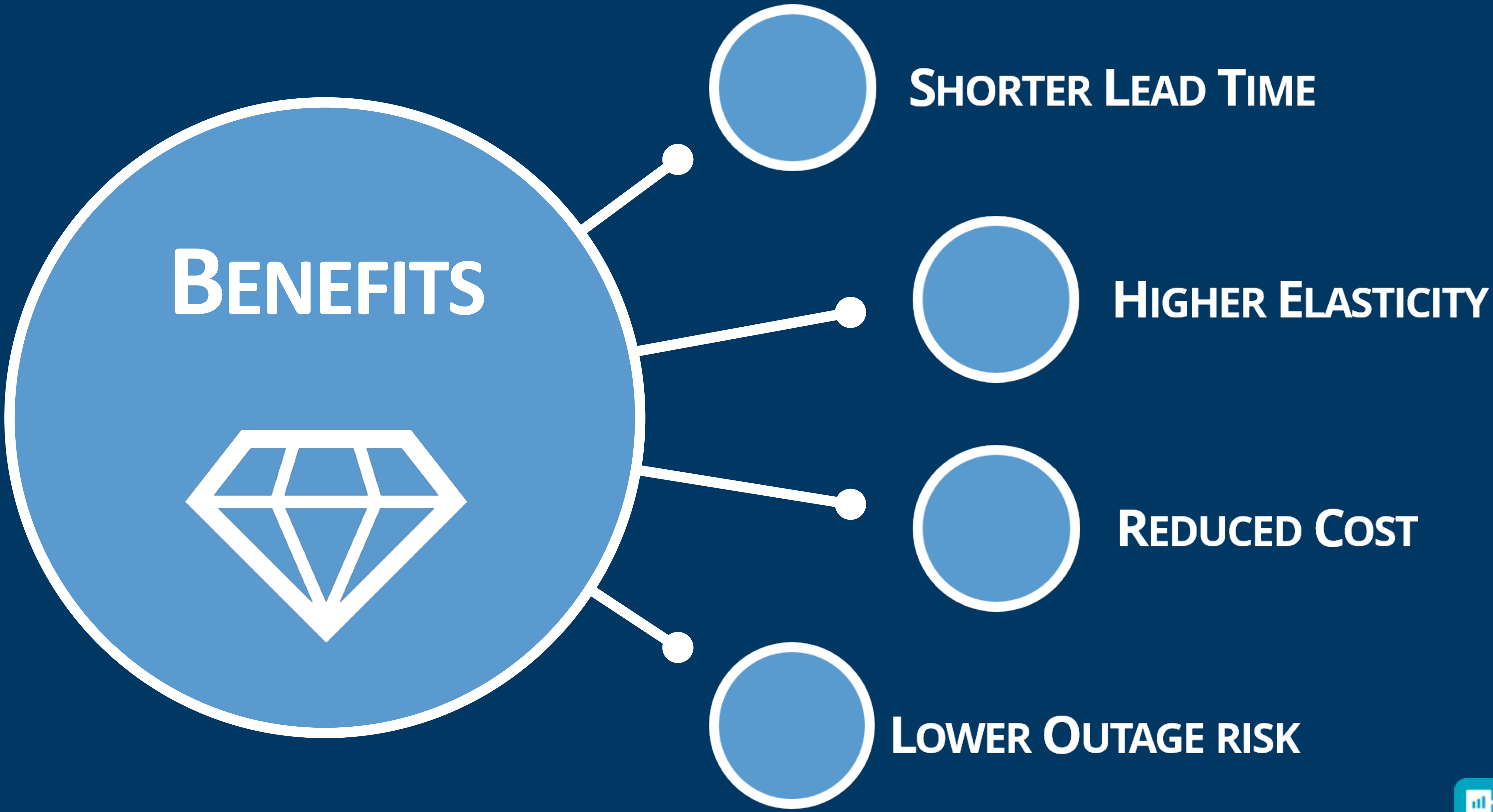
FaaS

Developers deploy small units of code, which are executed as needed as discrete actions in an event-based fashion.





Source: CNCF WG-Serverless Whitepaper v1.0 (2018) (2018); What Is Serverless? (Roberts & Chapin, 2017)

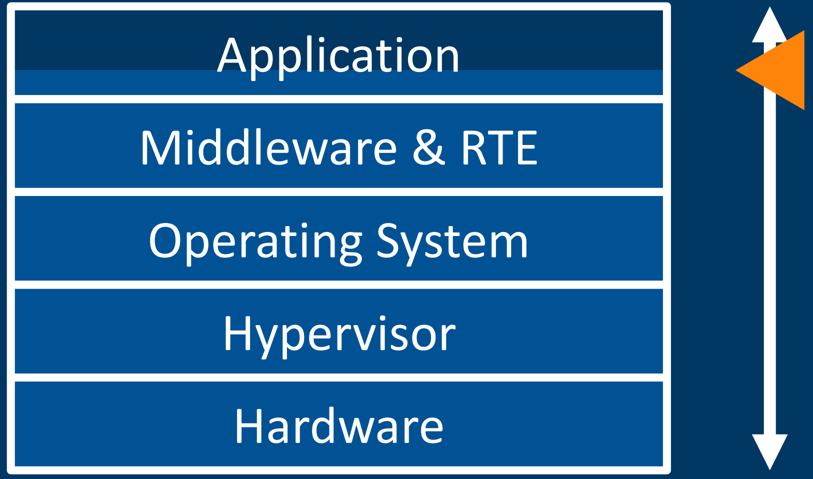


Benefits

- Shorter Lead Time
- Higher Elasticity
- Reduced Cost
- Lower Outage risk

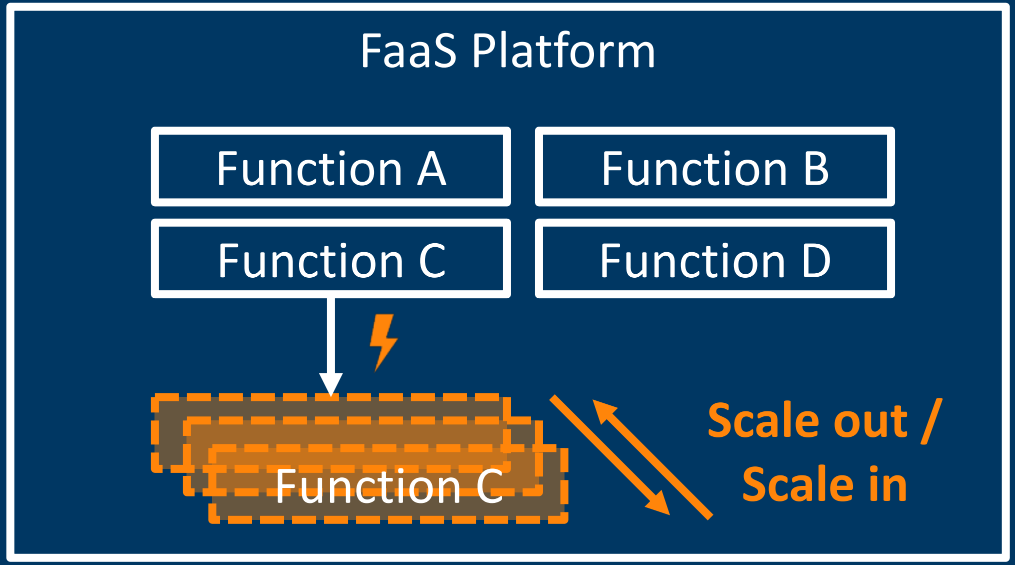
Shorter Lead Time

Developers can focus on creating business value and iterate faster. BaaS services can serve as bricks for rapid experimentation.



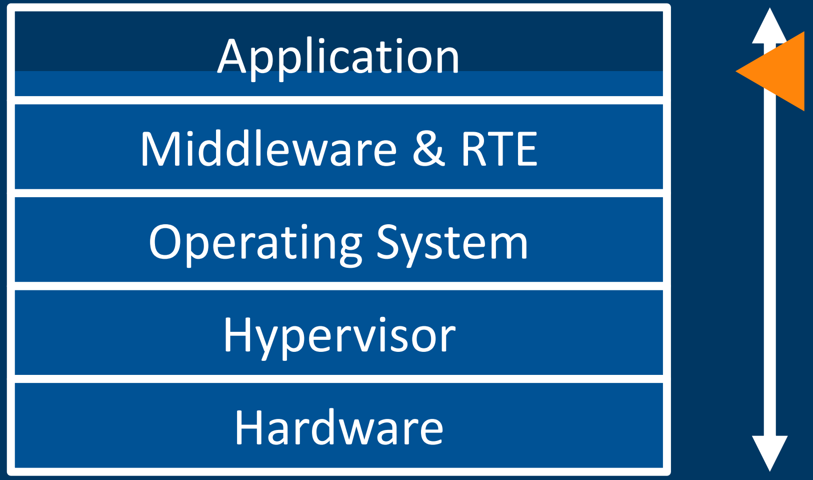
Higher Elasticity

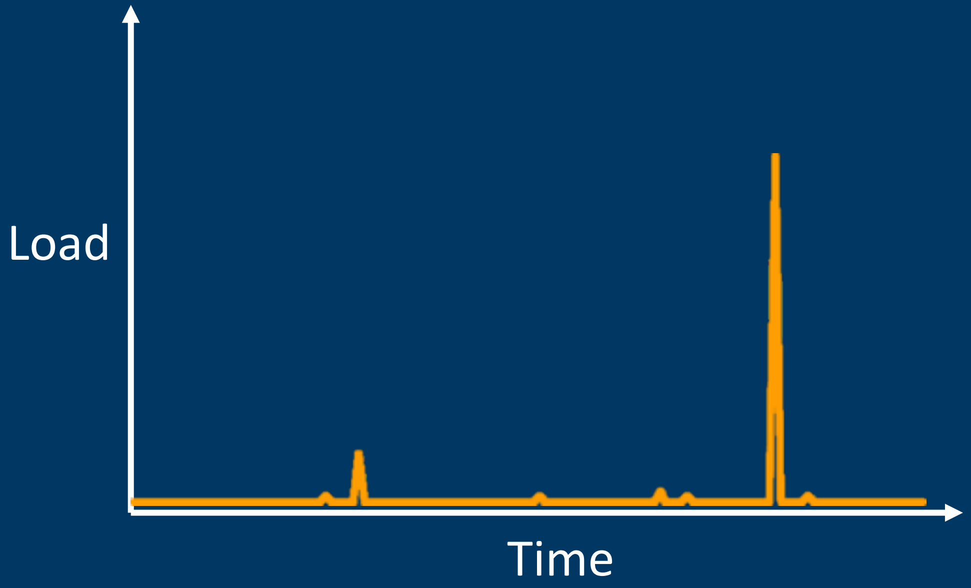
A serverless service automatically and precisely scales to your need.



Reduced Cost

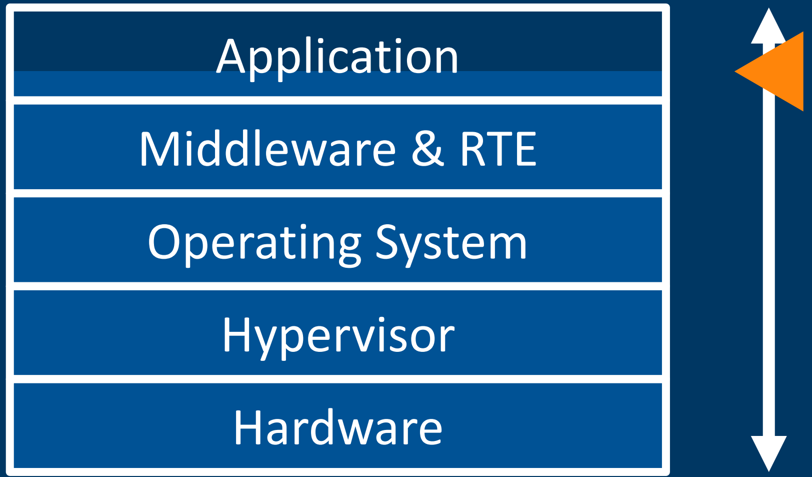
Serverless can reduce labor cost (due to less ops) as well as resource costs (due to less overprovisioning).

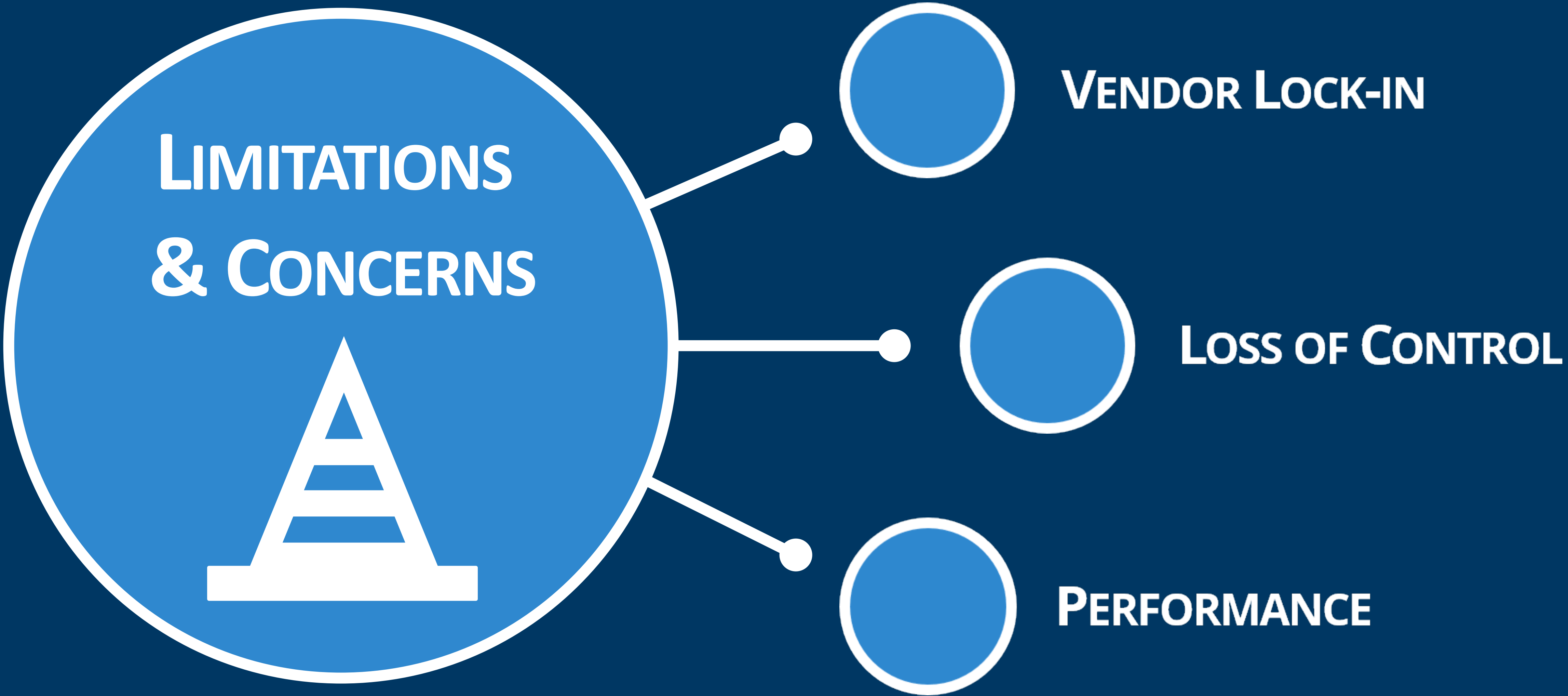




Lower Outage Risk

Outage risk is reduced since the expected downtime of components is reduced, and the time for them to be fixed is less volatile.





Limitations & Concerns

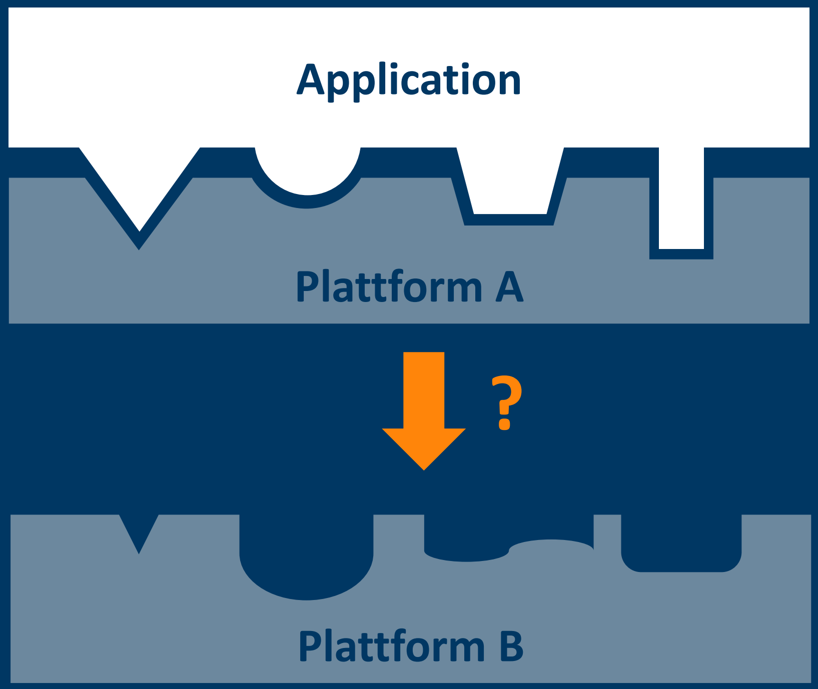
Vendor Lock-in

Loss of Control

Performance

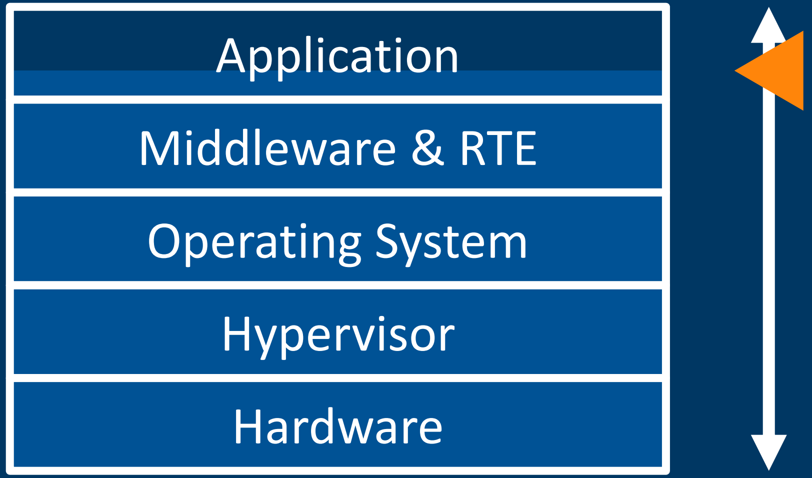
Vendor lock-in

The more you profit now, the higher the costs for a potential migration later on – and vice versa.



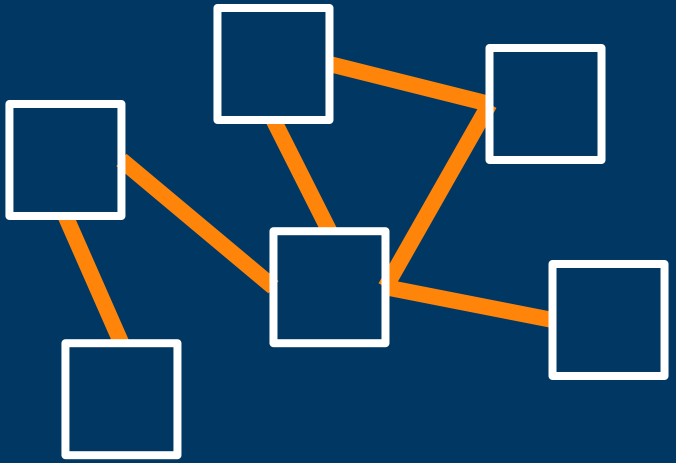
Loss of Control

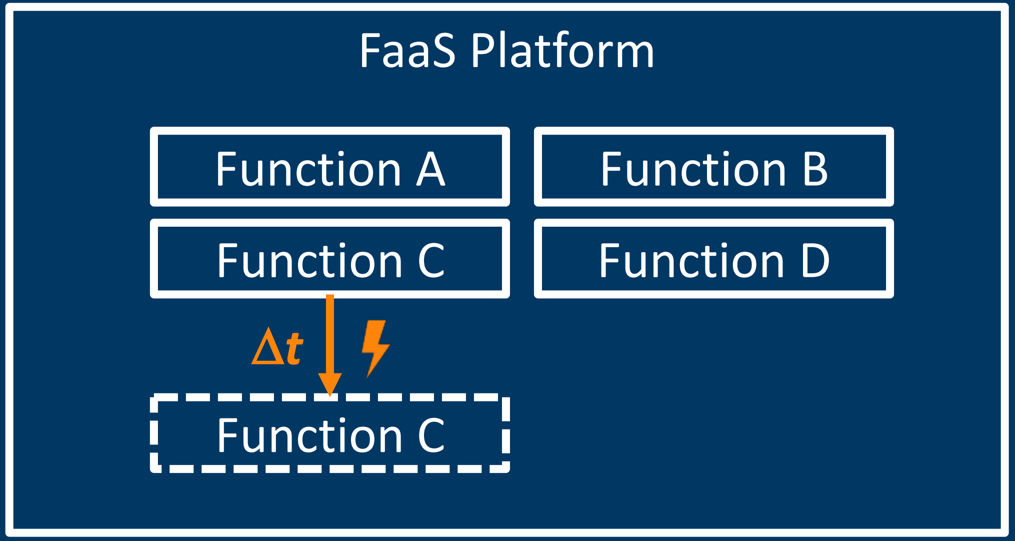
More abstraction means less control over details.



PERFORMANCE

Latency and performance variability can limit the applicability of serverless for certain types of applications.





FaaS Platform

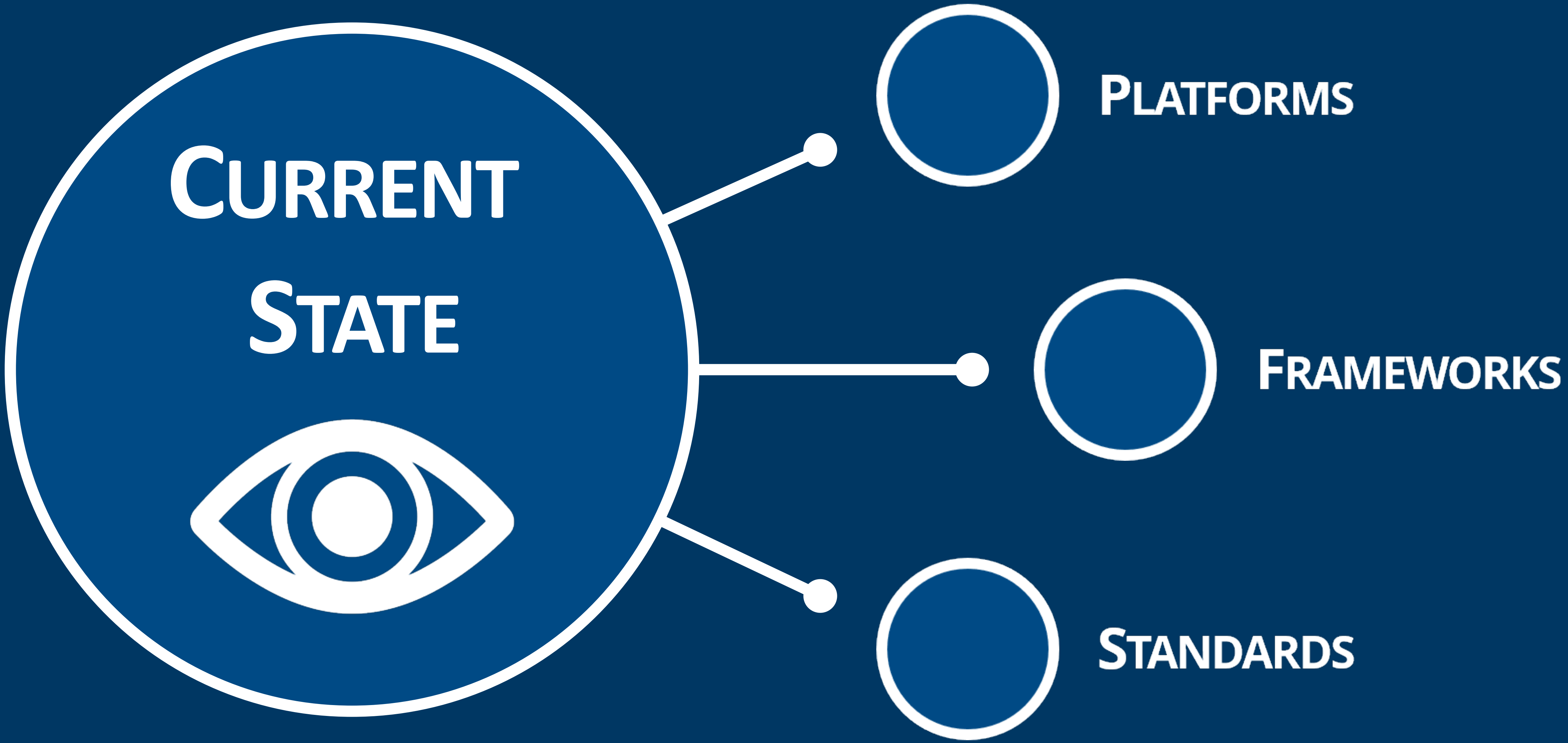
Function A

Function B

Function C

Function D

Function C



Current State

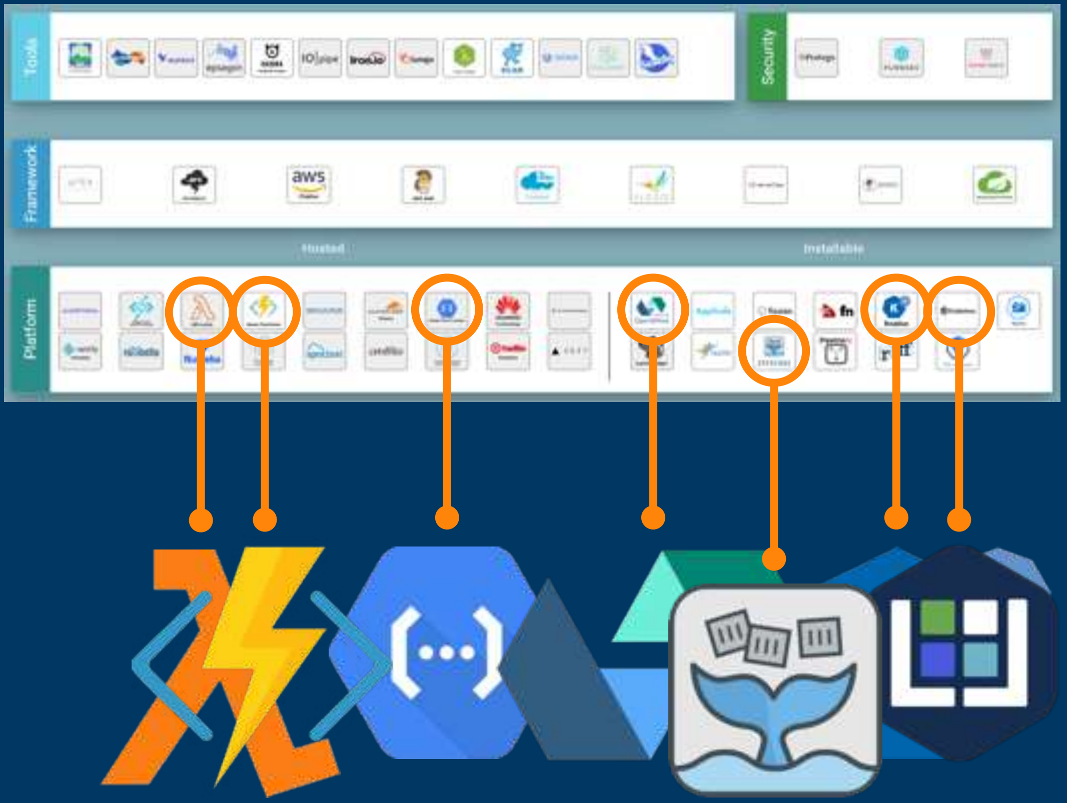
Platforms

Frameworks

Standards

Platforms

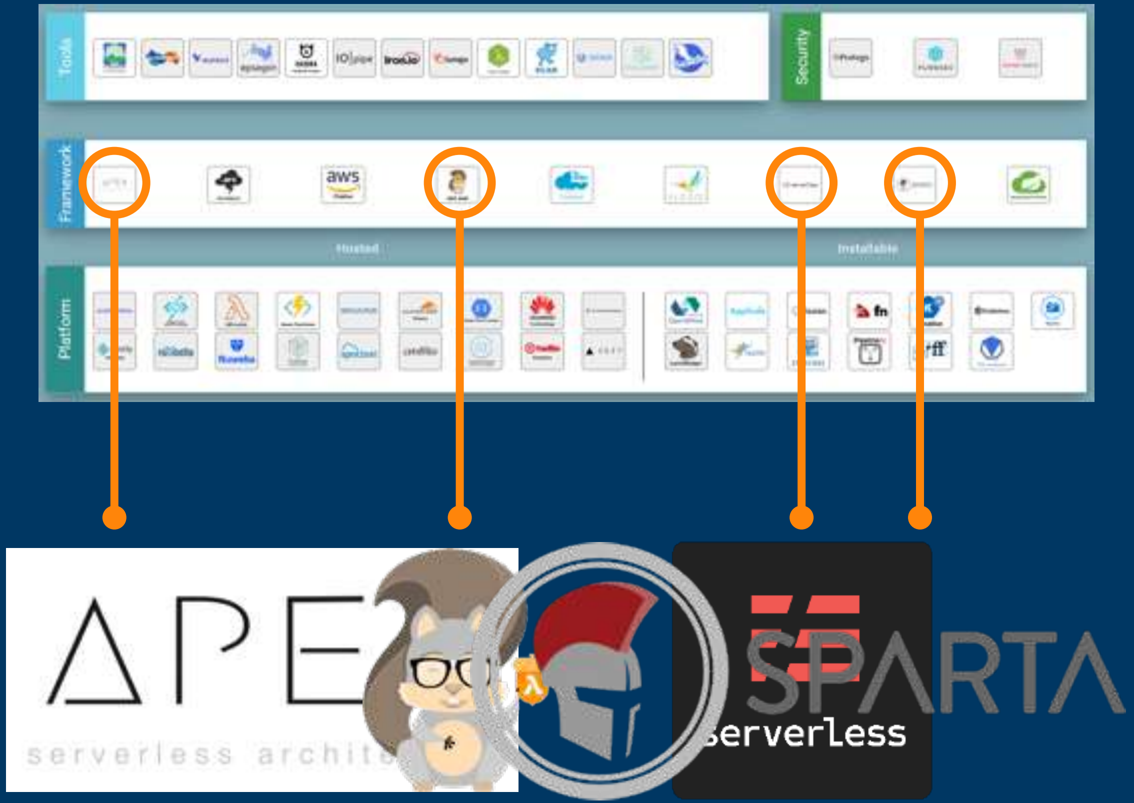
Some purists see true serverless only in in the public cloud.



Source: Cloud Native Computing Foundation; AWS; Azure; Google; openwhisk.apache.org; knative.dev; kubeless.io;

FRAMEWORKS

Frameworks increase productivity further by providing an additional layer of abstraction. Some of them also claims to promote portability.



Source: Cloud Native Computing Foundation; apex.run; AWS; gosparta.io

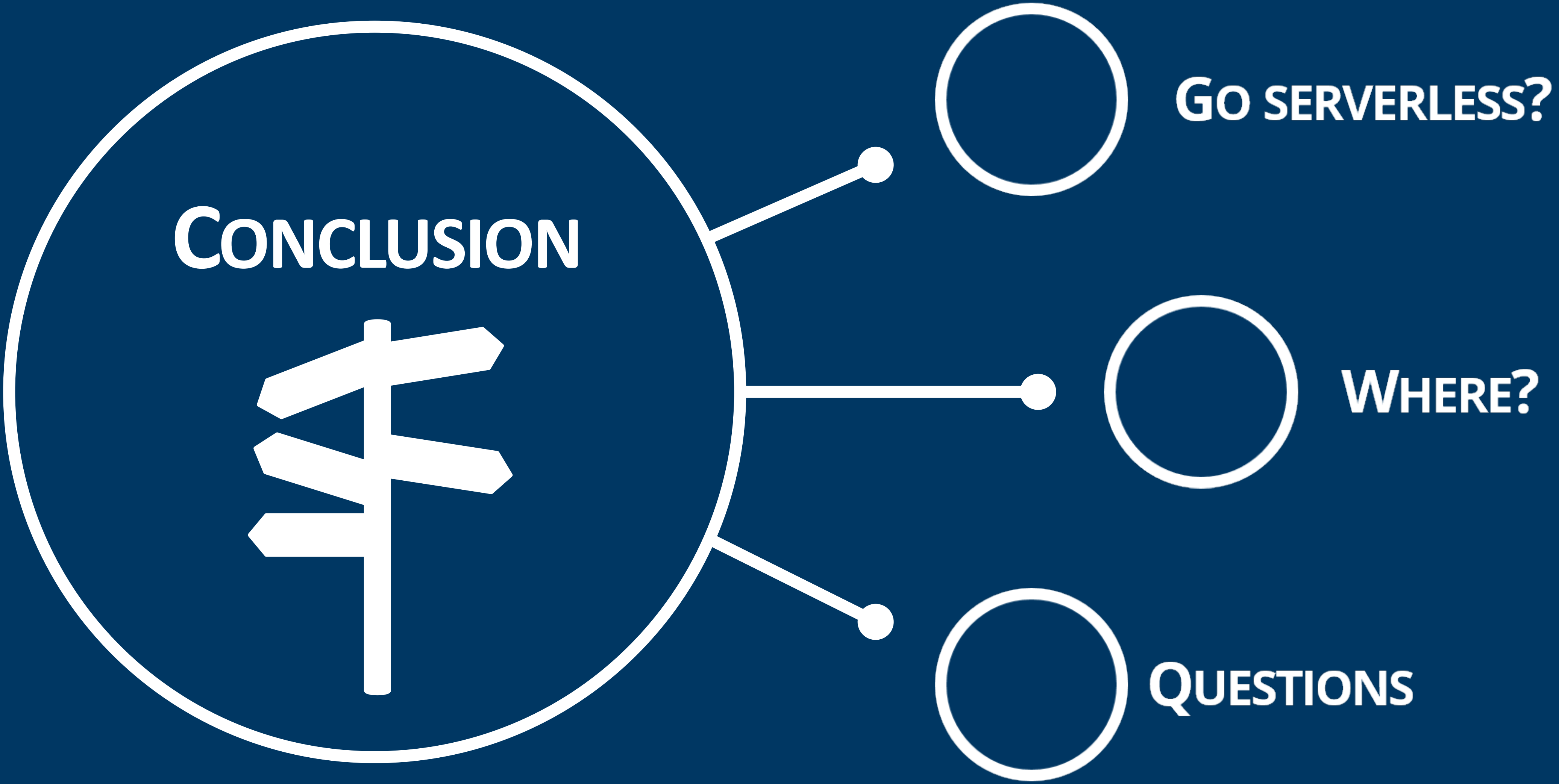
STANDARDS

The serverless space is getting increasingly diverse. Standards are crucial for portability and interoperability.

CLOUD NATIVE
COMPUTING FOUNDATION

cloudevents

Source: Cloud Native Computing Foundation; cloudevents.io



Conclusion

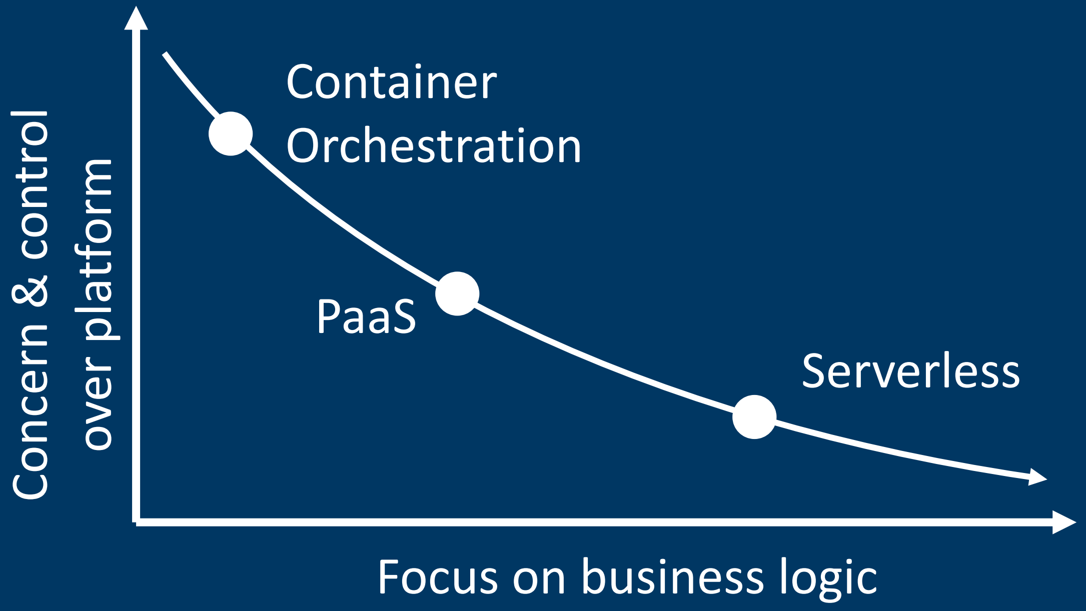
Go serverless?

Where?

Questions

Go serverless?

To determine the best cloud native deployment model, each approach should be evaluated. Hybrid scenarios should also be considered.



Feeatures & Capabilities

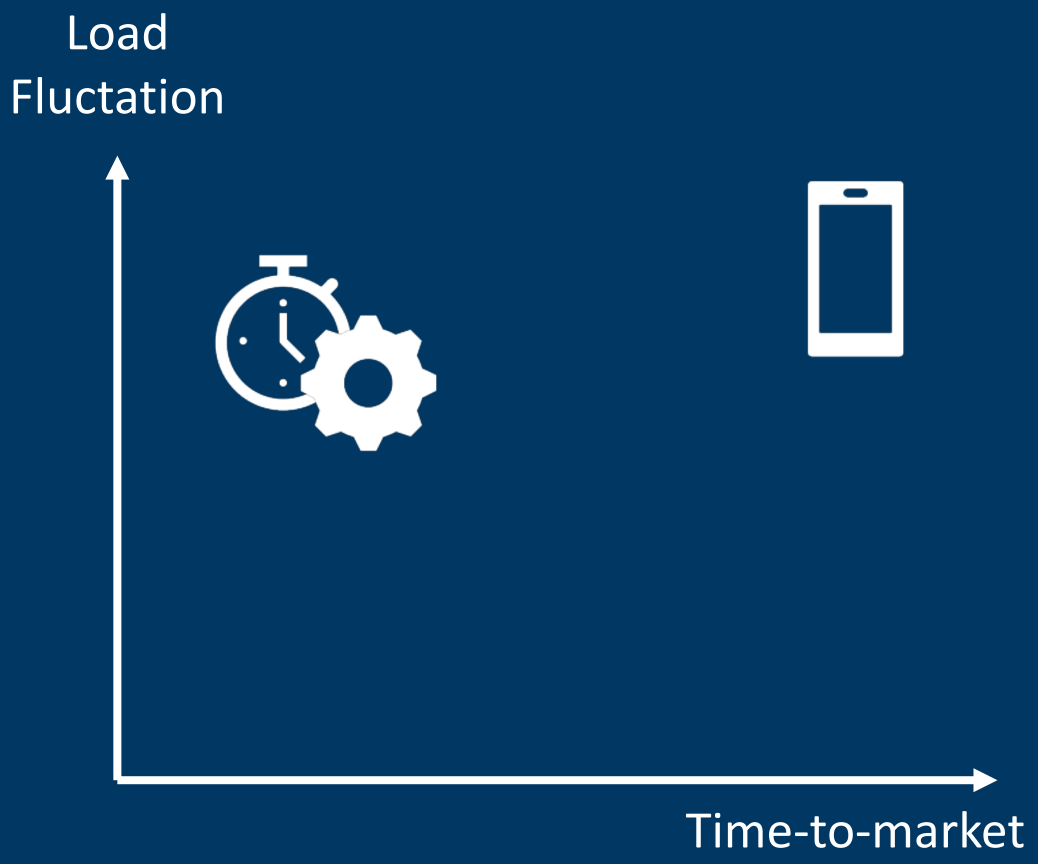
Operational Aspects

Total Cost of Ownership

Source: CNCF WG-Serverless Whitepaper v1.0 (2018)

Where?

Workloads with fluctuating or difficult to predict load and/or a need for a short time-to-market are prime candidates for serverless deployments.



QUESTIONS