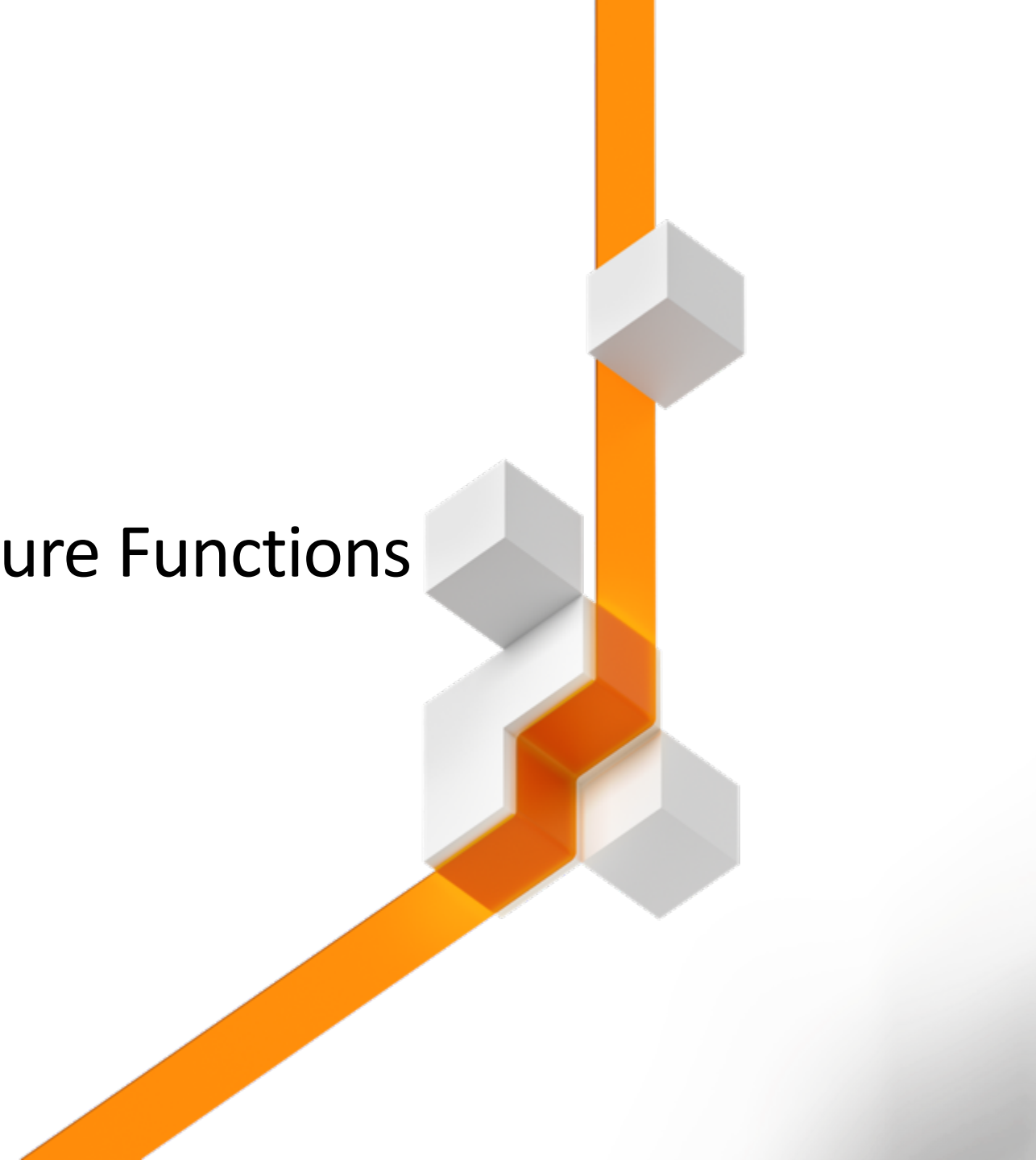
Go Serverless with Java and Azure Functions

Sandra Ahlgrimm

Cloud Advocate | @sKriemhild

Microsoft



What media should I use to keep backup?

What is the right **size** of **servers** for my business needs?

How do I **deploy** new **code** to my **server**?

What happens in case of **server hardware** failure?

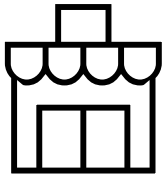
What size of **servers** should I **buy**?

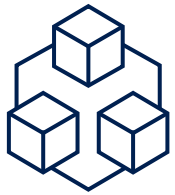
Which packages should be on my **server**?

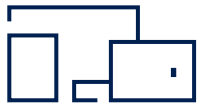
Who **monitors** my **app**?

How often should I backup my **server**?

How can I **scale** my app?







How can I increase **server** utilization?

Do I need secondary network connection?

Who has **physical** access to my **servers**?

Which OS should I use?

Are my **server** in a secure location?

Who **monitors** my **Servers**?

Do I need a UPS?

What happens if the power goes out?

What storage I need to use?

How many **servers** do I need?

It takes how long to **provision** a new **server**?

How often should I **patch** my **servers**?

How can I dynamically configure my app?

On-premises

The “evolution” of application platforms

What is the right **size** of **servers** for my business needs?

How can I increase **server** utilization?

How many **servers** do I need?

How can I **scale** my app?

How often should I **patch** my **servers**?

How often should I backup my **server**?

Which packages should be on **my server**?

How do I **deploy** new **code** to my **server**?

Which OS should I use?

Who **monitors** my app?

On-premises

IaaS

The “evolution” of application platforms

What is the right **size** of **“servers”** for my business needs?

How can I increase **“server”** utilization?

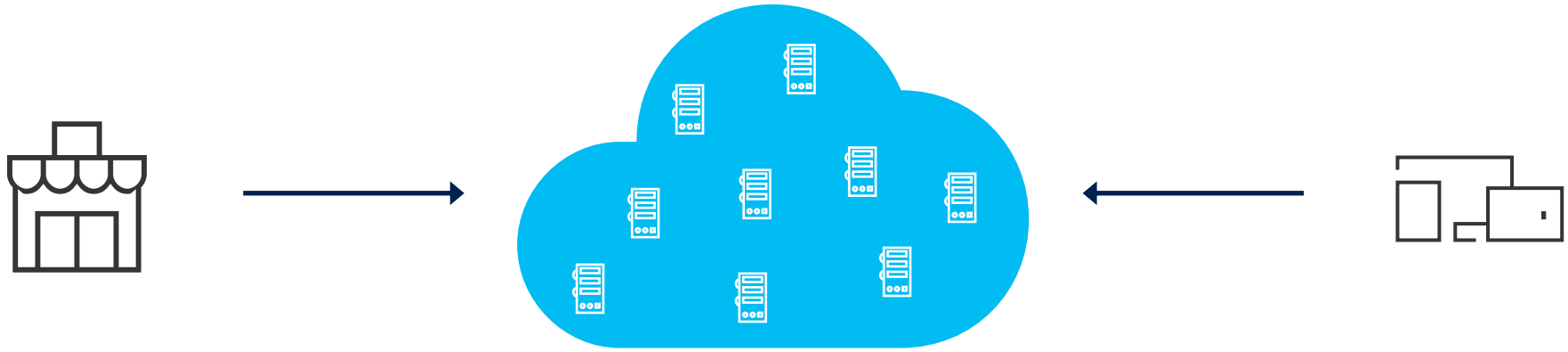
How many **“servers”** do I need?

How can I **scale** my app?

On-premises — IaaS — PaaS

The “evolution” of application platforms

How do I **architect** my app?



Serverless, the platform for cloud native apps

On-Premises — IaaS — PaaS — **Serverless**

The “evolution” of application platforms

What is Serverless?

Abstraction
of servers

Event-driven/
instant scale

Micro-billing

What are the benefits?

Focus

Solve business problems—not technology problems related to undifferentiated heavy lifting

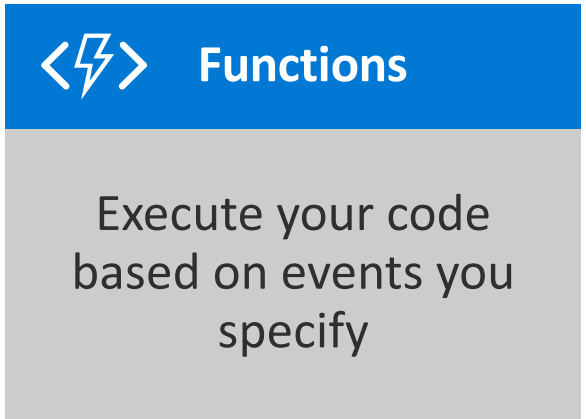
Efficiency

- Shorter time to market
- Fixed costs converted to variable costs
- Better service stability
- Better development and testing management
- Less waste

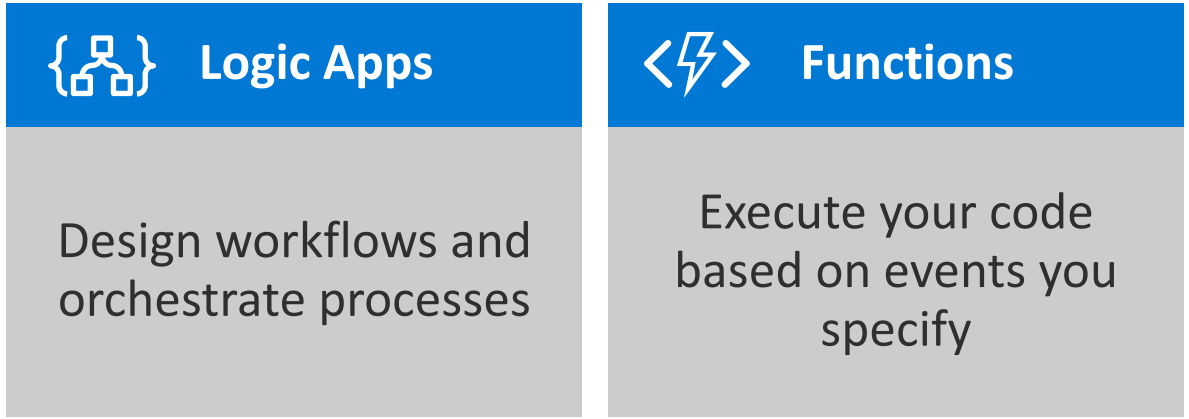
Flexibility

- Simplified starting experience
- Easier pivoting means more flexibility
- Easier experimentation
- Scale at your pace—don't bet the farm on Day 1
- Natural fit for microservices

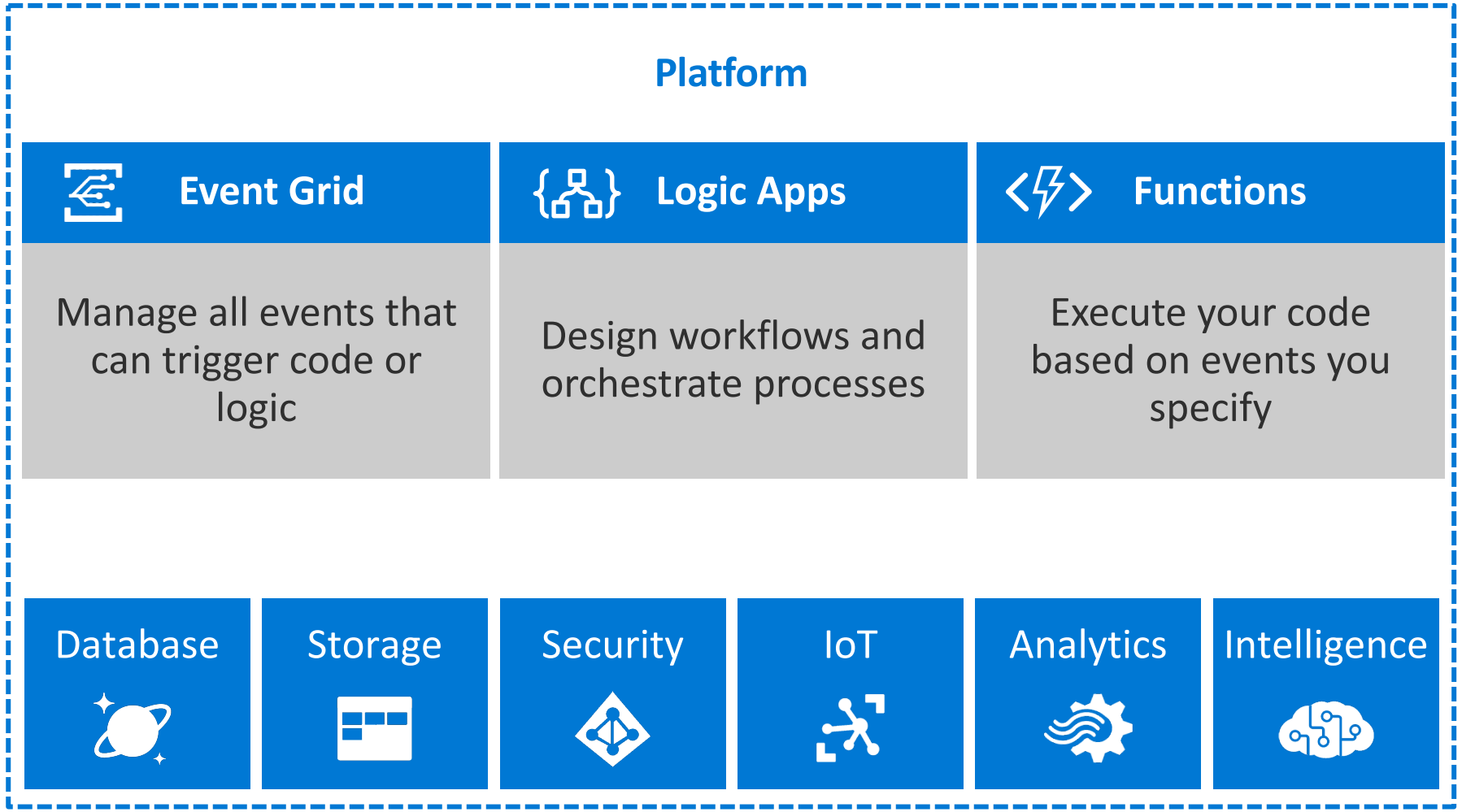
Azure serverless platform components



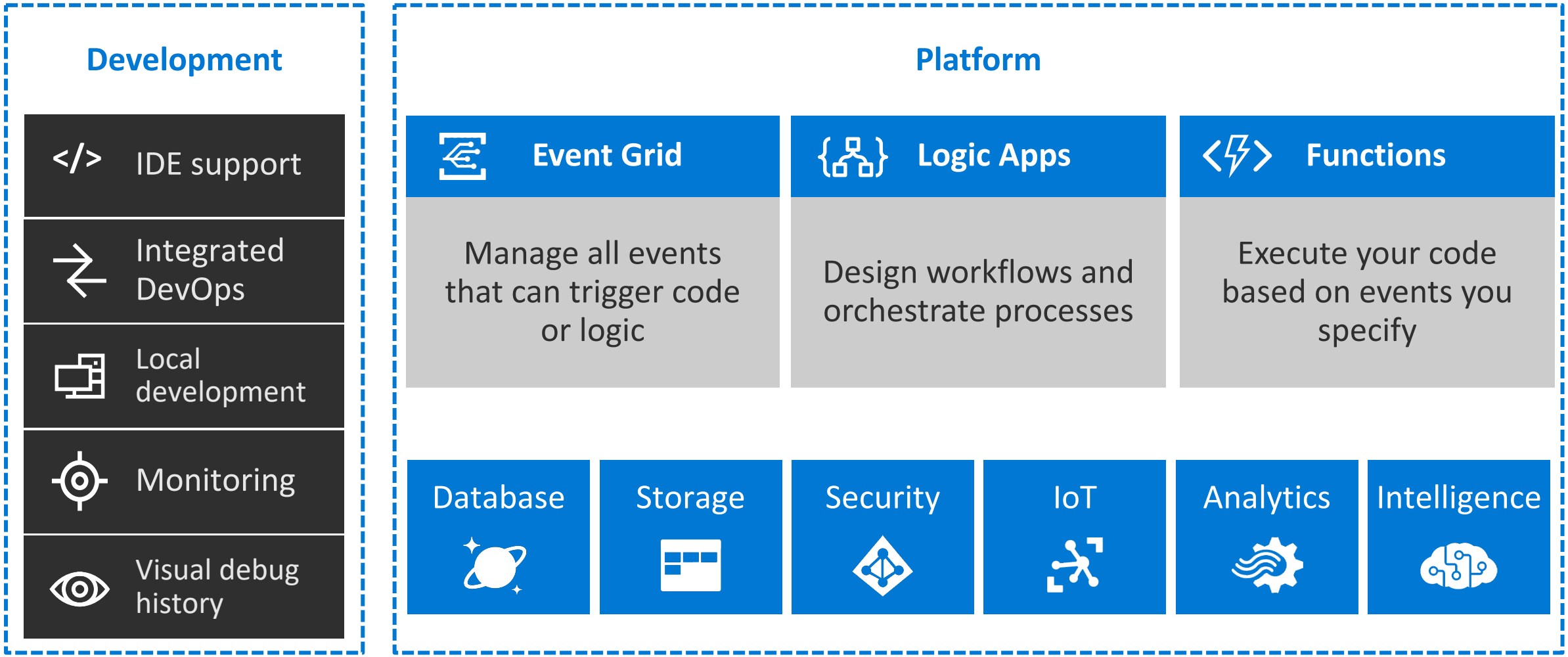
Azure serverless platform components



Azure serverless platform components



Azure serverless platform components



Azure Functions

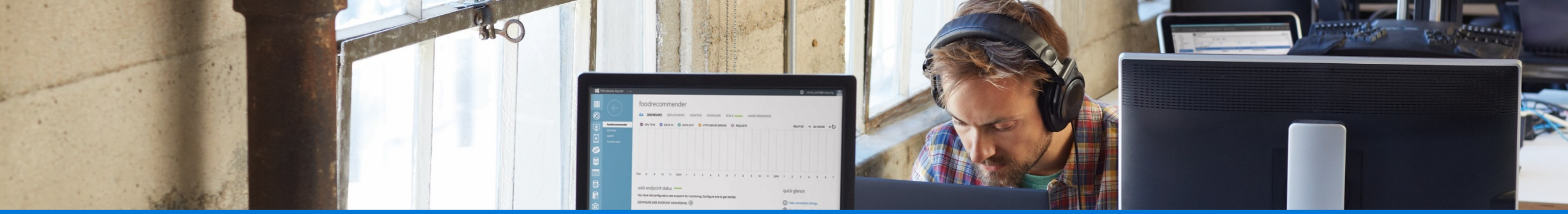
Introducing Functions

Code

Azure Functions

+

Events



Boost development efficiency

Triggers

Use triggers to define how functions are invoked
Avoid hardcoding with preconfigured JSON files
Build serverless APIs using HTTP triggers

Bindings

Connect to data with input and output bindings
Bind to Azure solutions and third-party services
Use HTTP bindings in tandem with HTTP triggers

Proxies

Define one API surface for multiple function apps
Create endpoints as reverse proxies to other APIs
Condition proxies to use variables

Local debugging

Debug your functions locally
Use debugging tools in Azure portal, VS, and VS Code

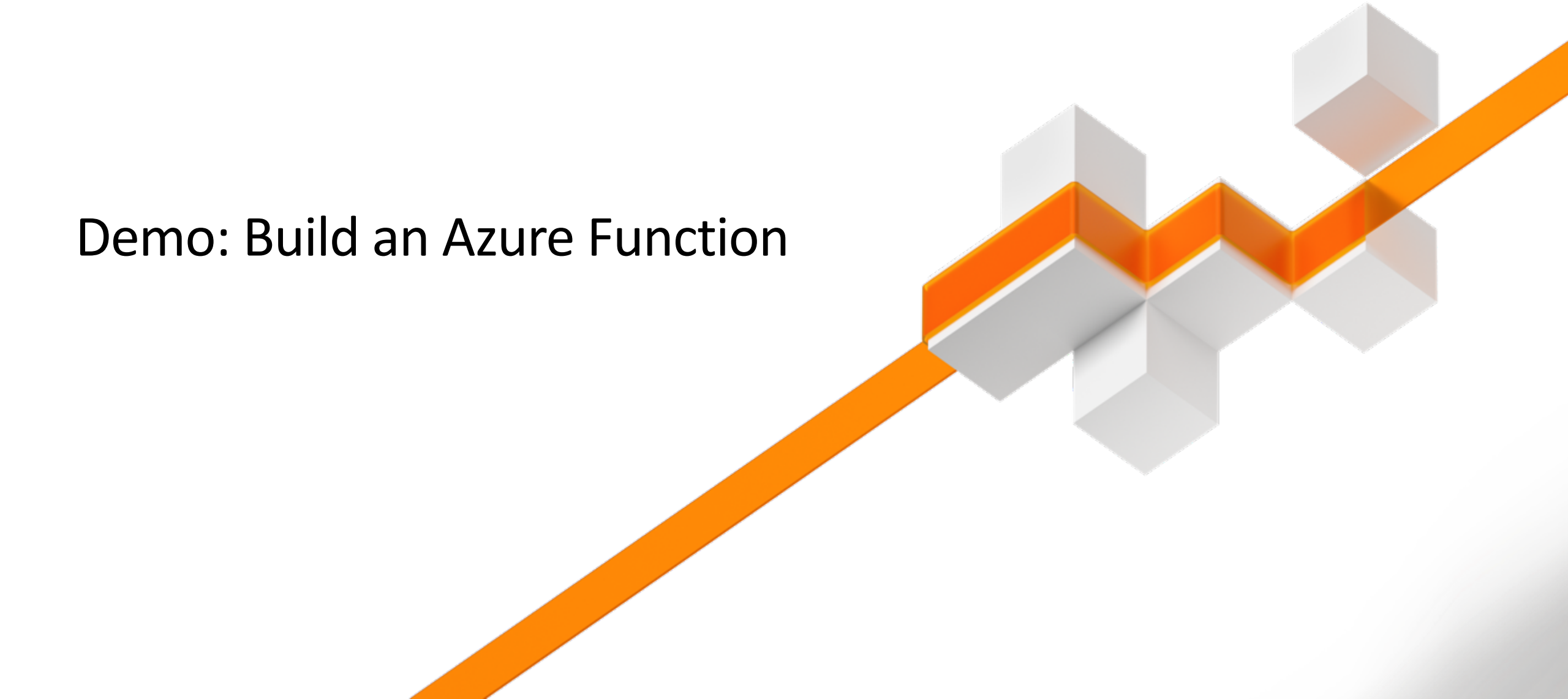
CI/CD

Save time with built-in DevOps
Deploy functions using App Service for CI
Leverage Microsoft, partner services for CD

Monitoring

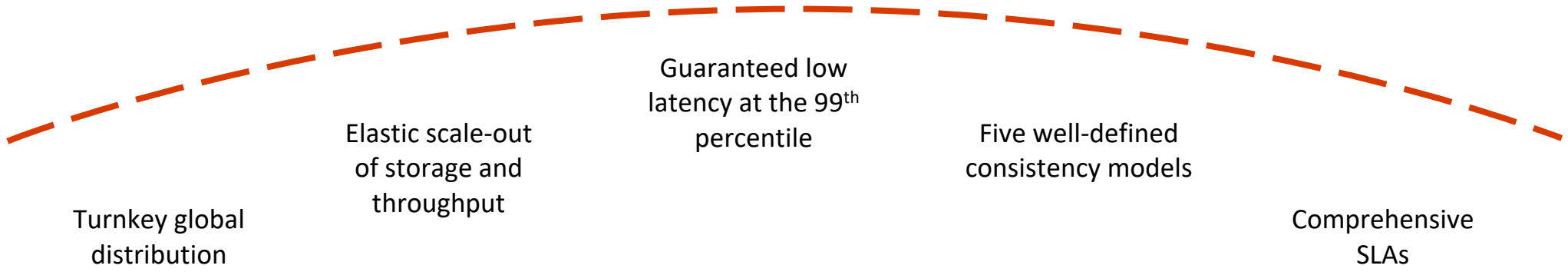
Integrate with Azure Application Insights
Get near real-time details about function apps
See metrics around failures, executions, etc.

Demo: Build an Azure Function

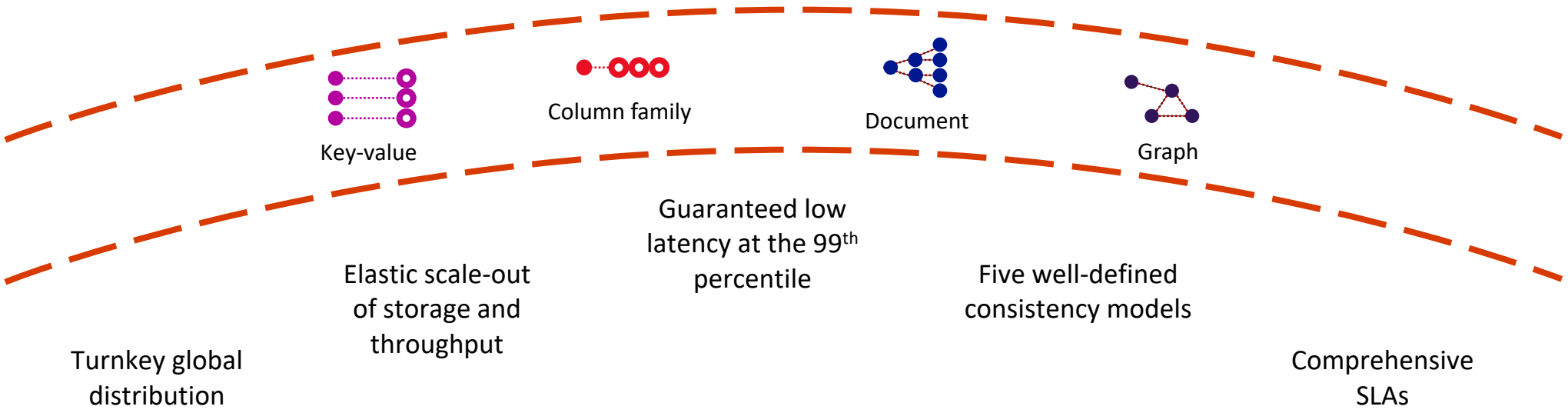


Azure Cosmos DB

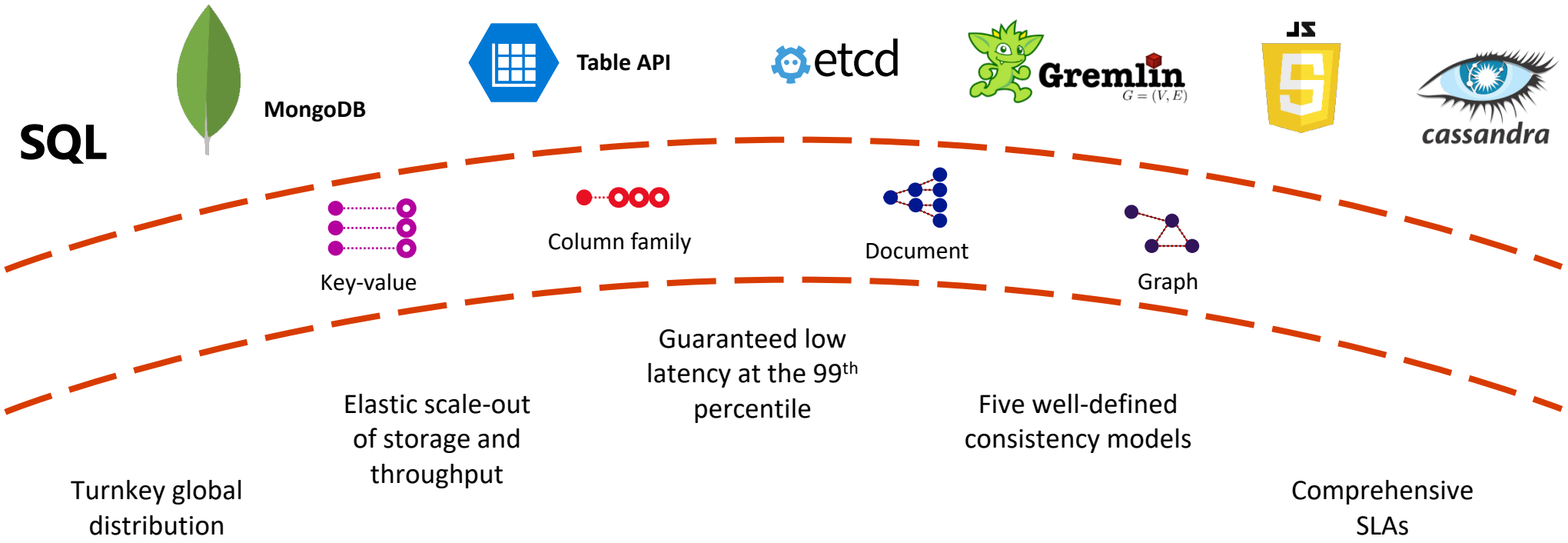
Azure Cosmos DB



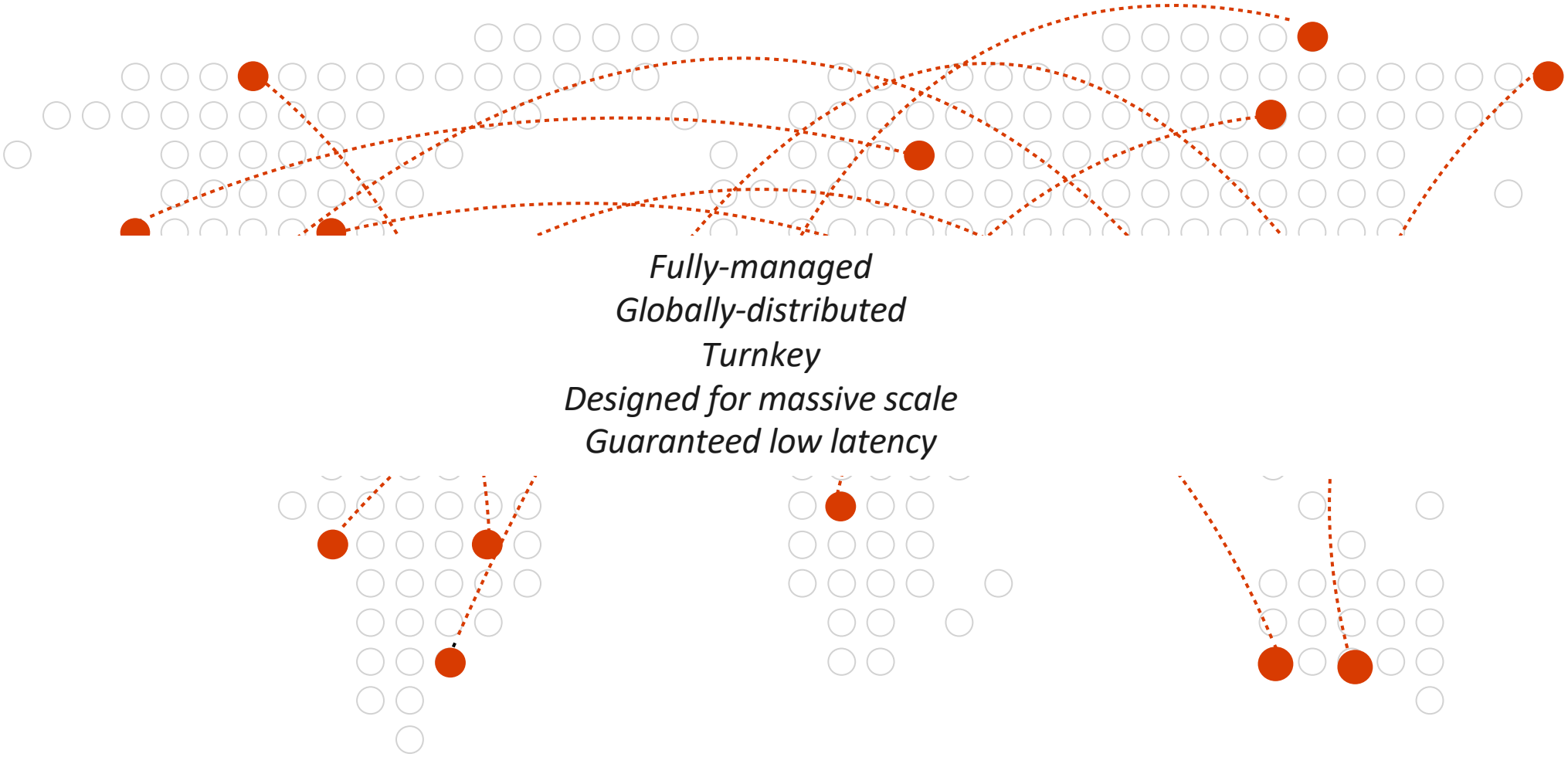
Azure Cosmos DB



Azure Cosmos DB

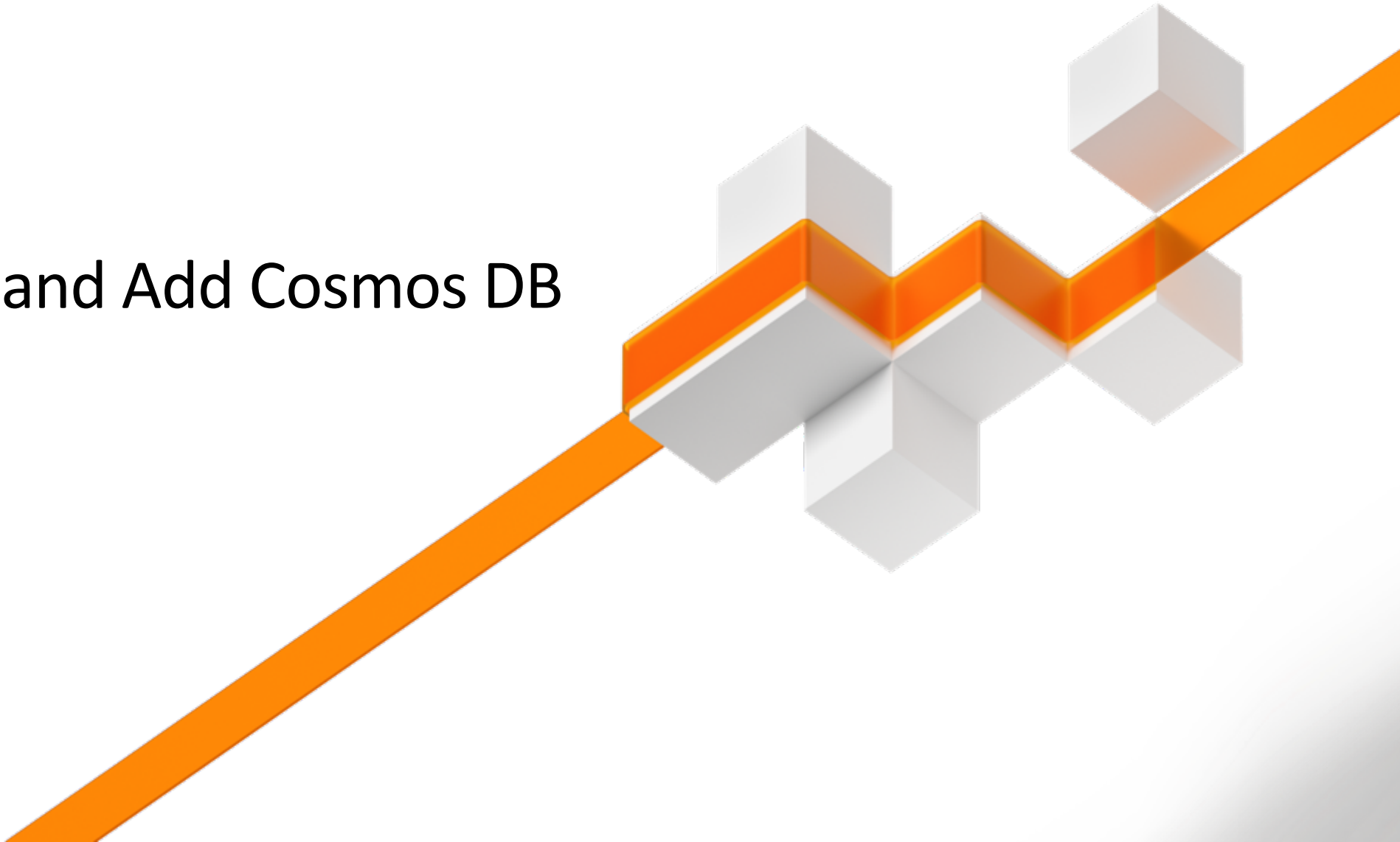


Azure Cosmos DB



Fully-managed
Globally-distributed
Turnkey
Designed for massive scale
Guaranteed low latency

Demo: Create and Add Cosmos DB



aka.ms/serverless-cosmo

What is Azure Functions?

An event-based, serverless compute experience that accelerates app development

Azure Functions = FaaS++

Integrated programming model

Use built-in triggers and bindings to define when a function is invoked and to what data it connects

Enhanced development experience

Code, test and debug locally using your preferred editor or the easy-to-use web based interface including monitoring

Hosting options flexibility

Choose the deployment model that better fits your business needs without compromising development experience

Focus on code, not plumbing

No infrastructure management

Auto-scale based on your workload

No wasted resources, pay only for what you use

FaaS is at the center of serverless

Functions-as-a-Service programming model use functions to achieve true serverless compute

Single responsibility

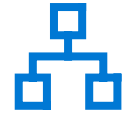
Functions are single-purposed, reusable pieces of code that process an input and return a result

Short lived

Functions don't stick around when finished executing, freeing up resources for further executions

Stateless

Functions don't hold any persistent state and don't rely on the state of any other processes

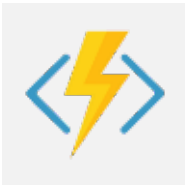


Event driven & scalable

Functions respond to predefined events, and are instantly replicated as many times as needed

Homework

Get free Resources on Azure aka.ms/az-free



Azure Functions aka.ms/javafunction

Functions with Cosmos DB aka.ms/homework

THANK YOU

Sandra Ahlgrimm

Cloud Advocate | @sKriemhild

Microsoft