March 28, 2019 — 8:30 AM - 5:00 PM | Detroit, Michigan

Microsoft

Azure Functions

Learn, architect, and
develop solutions on Azure

#AzureDevDays
for developers, by developers

Learn.

Connect.

Explore.

Serverless in Azure

Functions

Serverless compute

Logic Apps

Serverless workflow

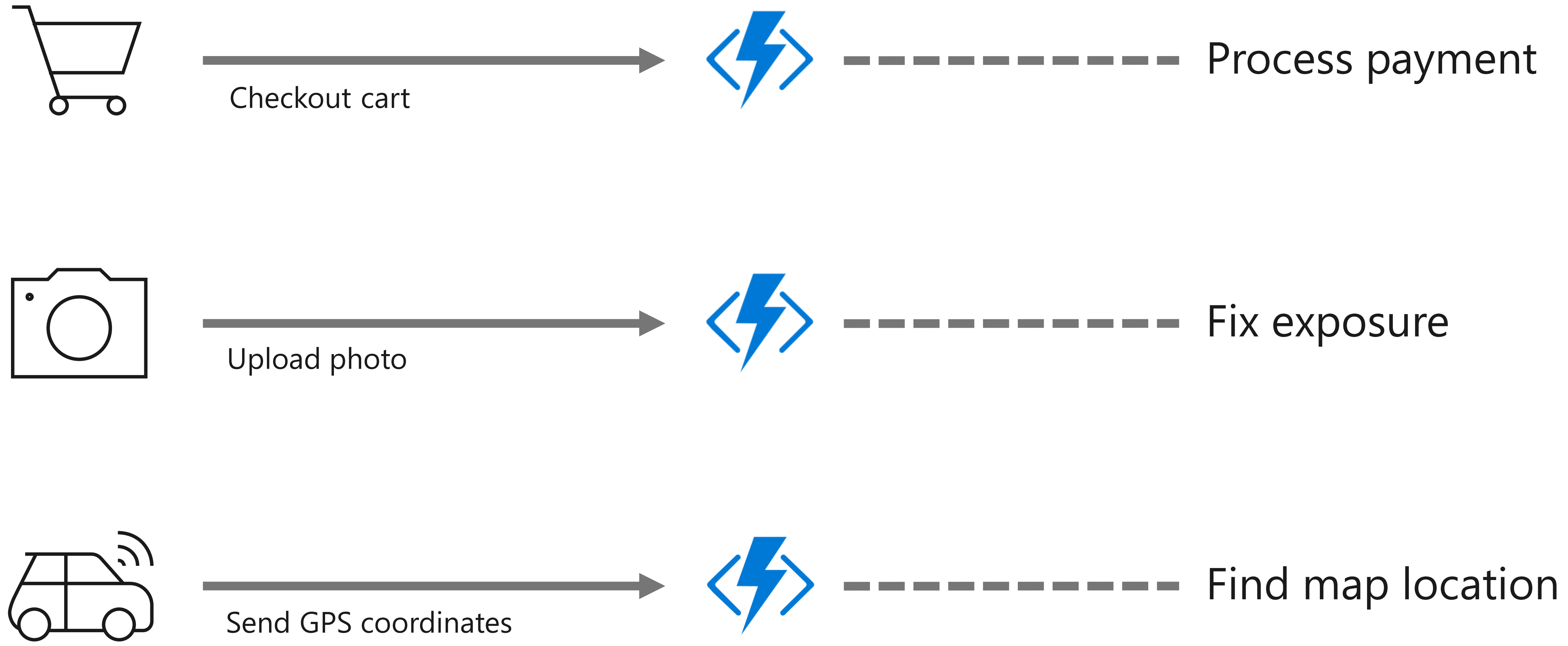
Event Grid

Serverless events

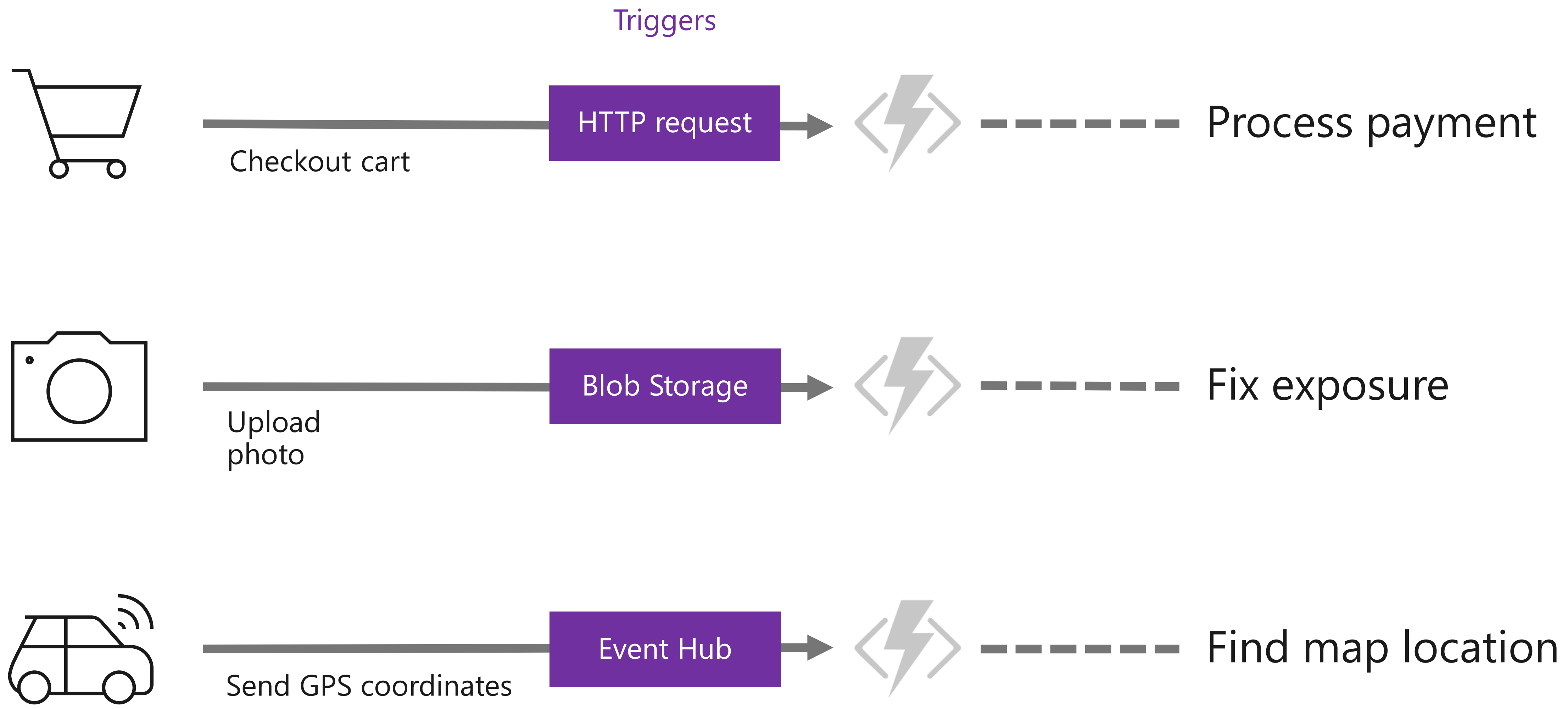
What is an Azure Function?

Typically a small, precise and highly cohesive unit of code that can be *triggered* to run based on an **event** or **schedule**.

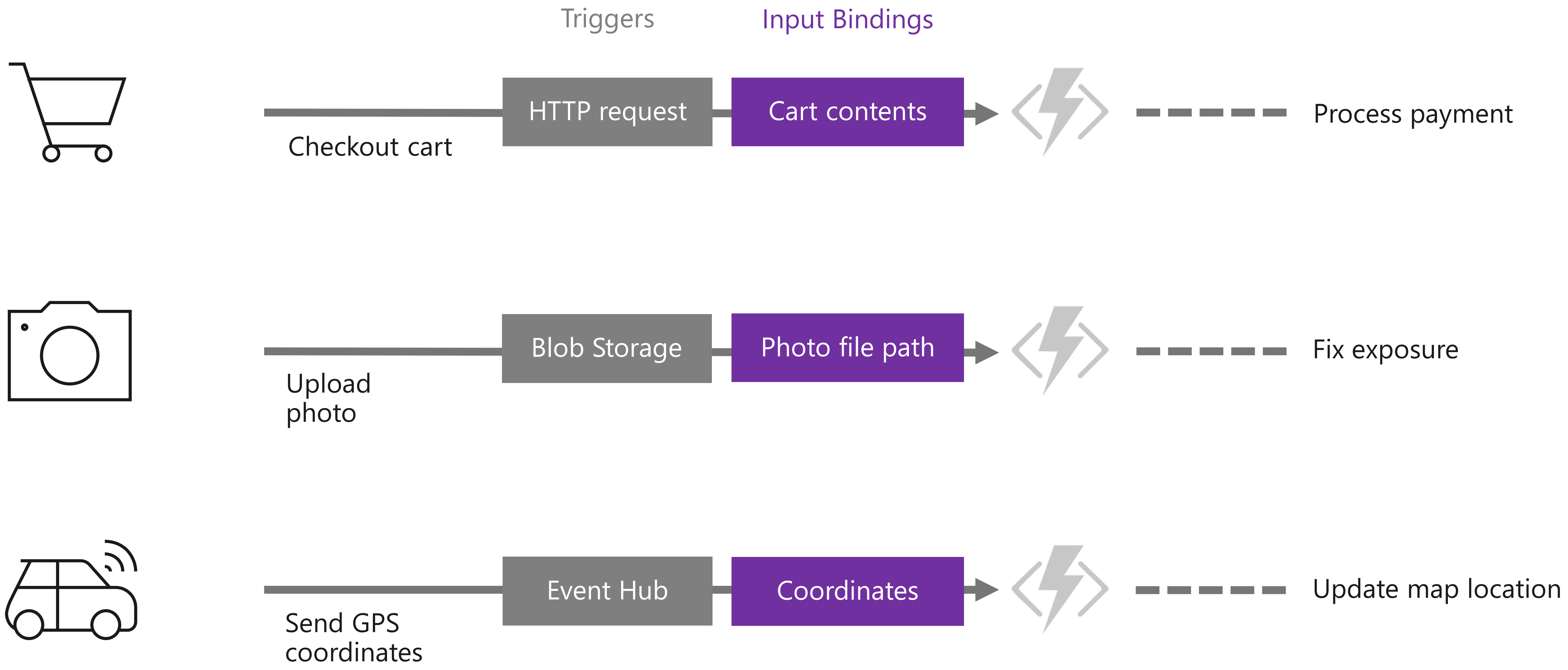
Introducing Azure Functions



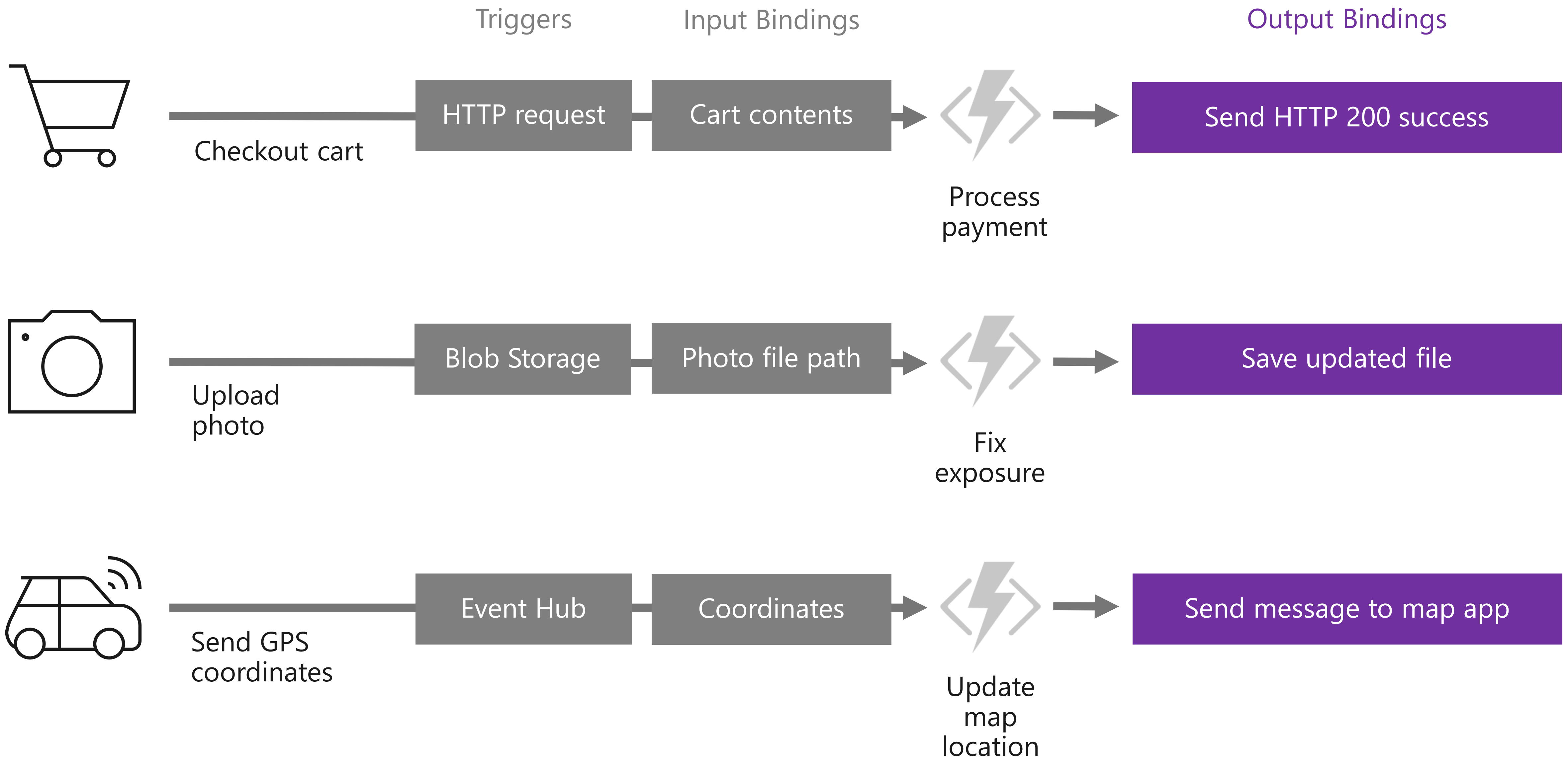
Triggers


Triggers
Checkout cart
HTTP request
Process payment
Upload photo
Blob Storage
Fix exposure
Send GPS coordinates
Event Hub
Find map location

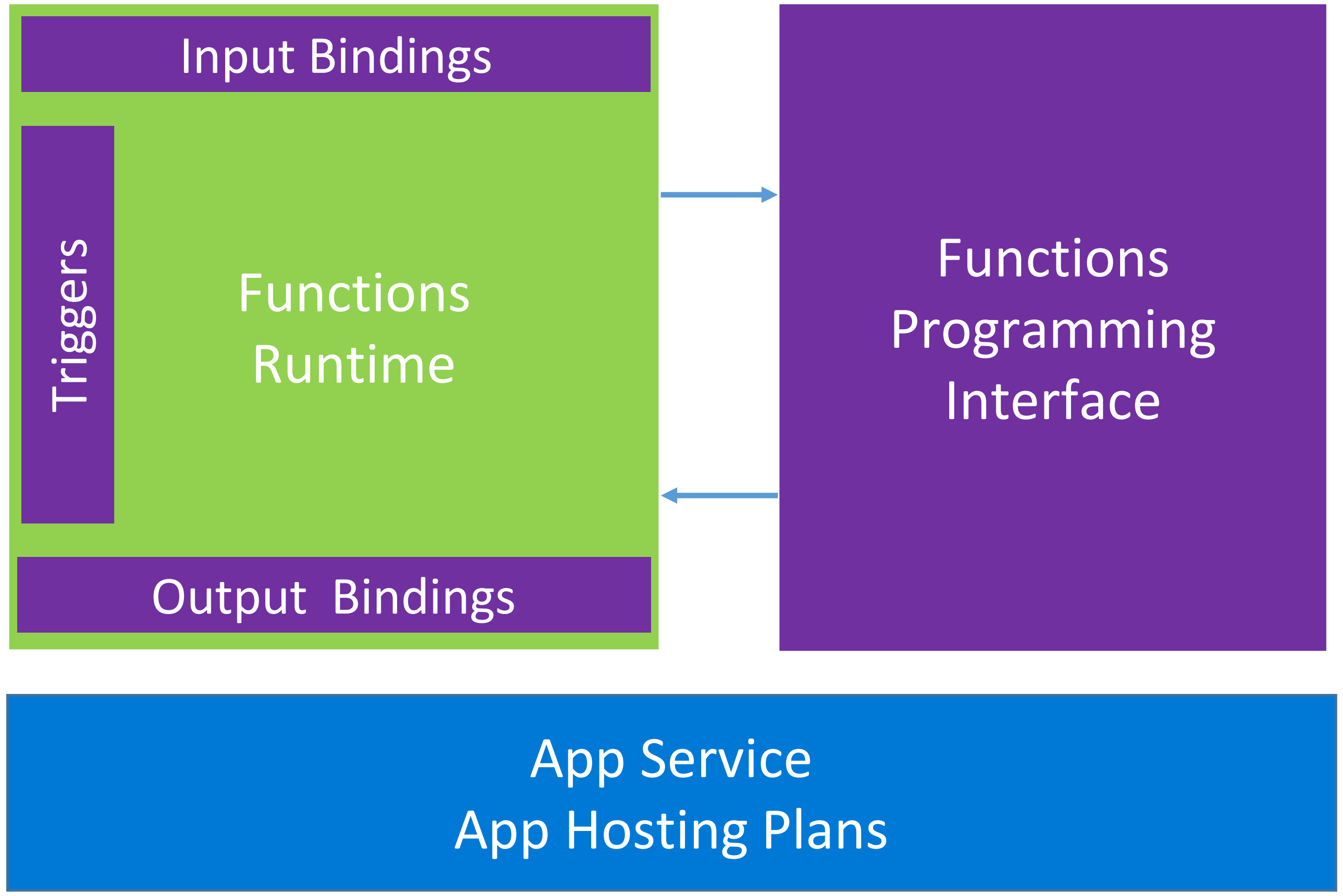
Bindings


Triggers
Input Bindings
HTTP request
Cart contents
Checkout cart
Process payment
Blob Storage
Photo file path
Upload photo
Fix exposure
Event Hub
Coordinates
Send GPS coordinates
Update map location

Output Bindings



Azure Functions Architecture



Hello Azure Functions World! (Web)

Files comprising a C# function…

function.json

Bindings

Defines input and output bindings.

run.csx

Code

Code that runs in response to the trigger.

Bindings defined in function.json

```json
{
  "bindings": [
    {
      "authLevel": "function",
      "name": "req",
      "type": "httpTrigger",
      "direction": "in",
      "methods": [
        "get",
        "post"
      ]
    },
    {
      "name": "$return",
      "type": "http",
      "direction": "out"
    }
  ],
  "disabled": false
}
```

Name of the *binding* to use in code

Type of *trigger*

Sets it as an *input binding*

Sets it as an *output binding*

Function code in run.csx

```
using System.Net;

public static async Task<HttpResponseMessage> Run(HttpRequestMessage req, TraceWriter log)
{
    log.Info("C# HTTP trigger function processed a request.");

    // parse query parameter
    string name = req.GetQueryNameValuePairs()
        .FirstOrDefault(q => string.Compare(q.Key, "name", true) == 0)
        .Value;

    if (name == null)
    {
        // Get request body
        dynamic data = await req.Content.ReadAsAsync<object>();
        name = data?.name;
    }

    return name == null
        ? req.CreateResponse(HttpStatusCode.BadRequest, "Please pass a name on the query string or in the request body")
        : req.CreateResponse(HttpStatusCode.OK, "Hello " + name);
}
```

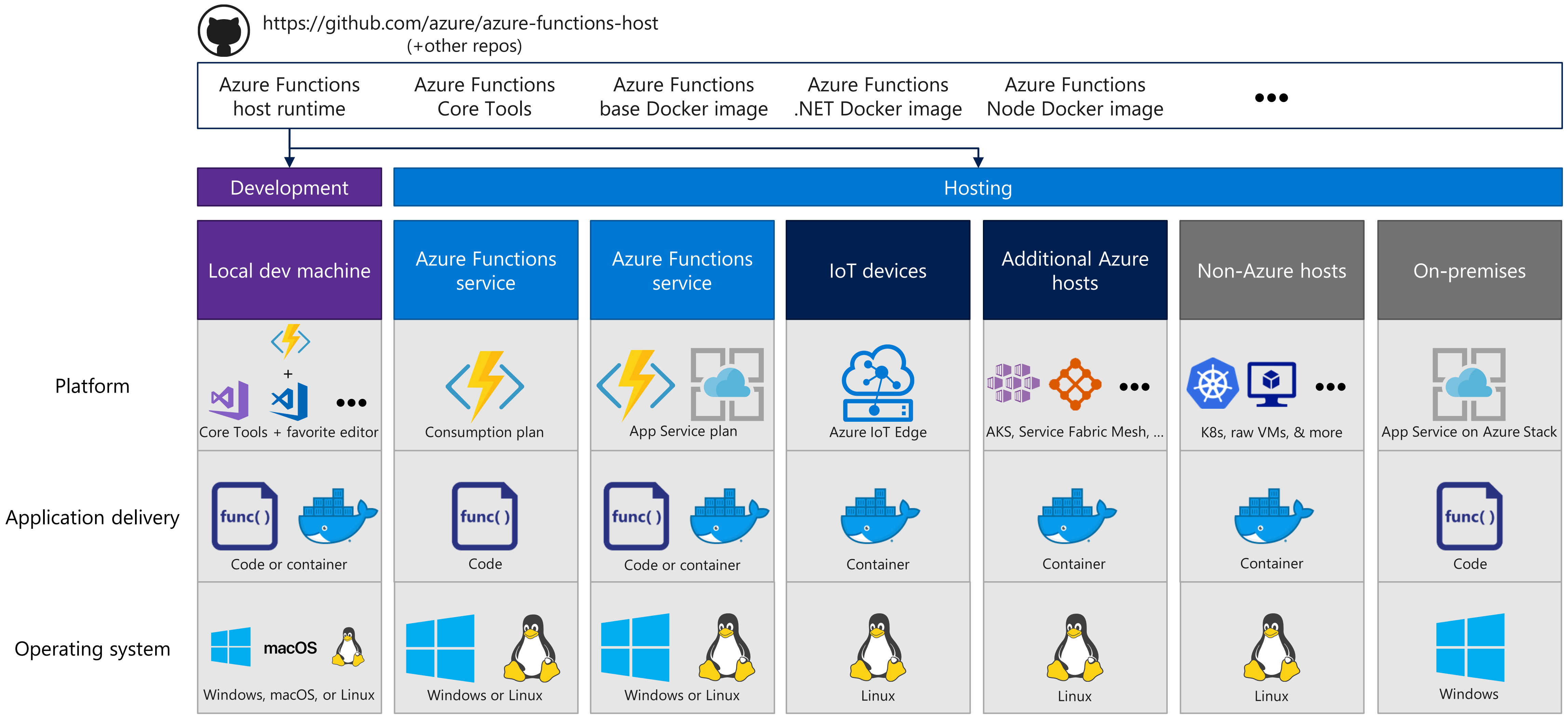
Read from *input binding*

Read from *input binding*

Write to *output binding*

Introducing Azure Functions 2.0

Functions everywhere



https://github.com/azure/azure-functions-host
(+other repos)

Azure Functions host runtime | Azure Functions Core Tools | Azure Functions base Docker image | Azure Functions .NET Docker image | Azure Functions Node Docker image | •••

Development | Hosting

	Development	Hosting					
	Local dev machine	Azure Functions service	Azure Functions service	IoT devices	Additional Azure hosts	Non-Azure hosts	On-premises
Platform	Core Tools + favorite editor	Consumption plan	App Service plan	Azure IoT Edge	AKS, Service Fabric Mesh, ...	K8s, raw VMs, & more	App Service on Azure Stack
Application delivery	Code or container	Code	Code or container	Container	Container	Container	Code
Operating system	Windows, macOS, or Linux	Windows or Linux	Windows or Linux	Linux	Linux	Linux	Windows

Language options

Generally available

.NET

node JS

Public preview

Java

Public preview

New!

python

More on the way!

Bindings and integrations

Functions 1.0

Microsoft.NET.Sdk.Functions (.NET Framework 4.6)

- HTTP
- Timer
- Storage
- Service Bus
- EventHubs
- Cosmos DB

Functions 2.0

Microsoft.NET.Sdk.Functions (.NET Standard 2.0)

- HTTP
- Timer

Microsoft.Azure.WebJobs.Extensions.Storage 3.0.0

Microsoft.Azure.WebJobs.Extensions.ServiceBus 3.0.0

Microsoft.Azure.Webjobs.Extensions.EventHubs 3.0.0

Microsoft.Azure.WebJobs.Extensions.CosmosDB 3.0.0

Microsoft.Azure.Webjobs.Extensions.EventGrid 2.0.0

Microsoft.Azure.WebJobs.Extensions.DurableTask 1.4.0

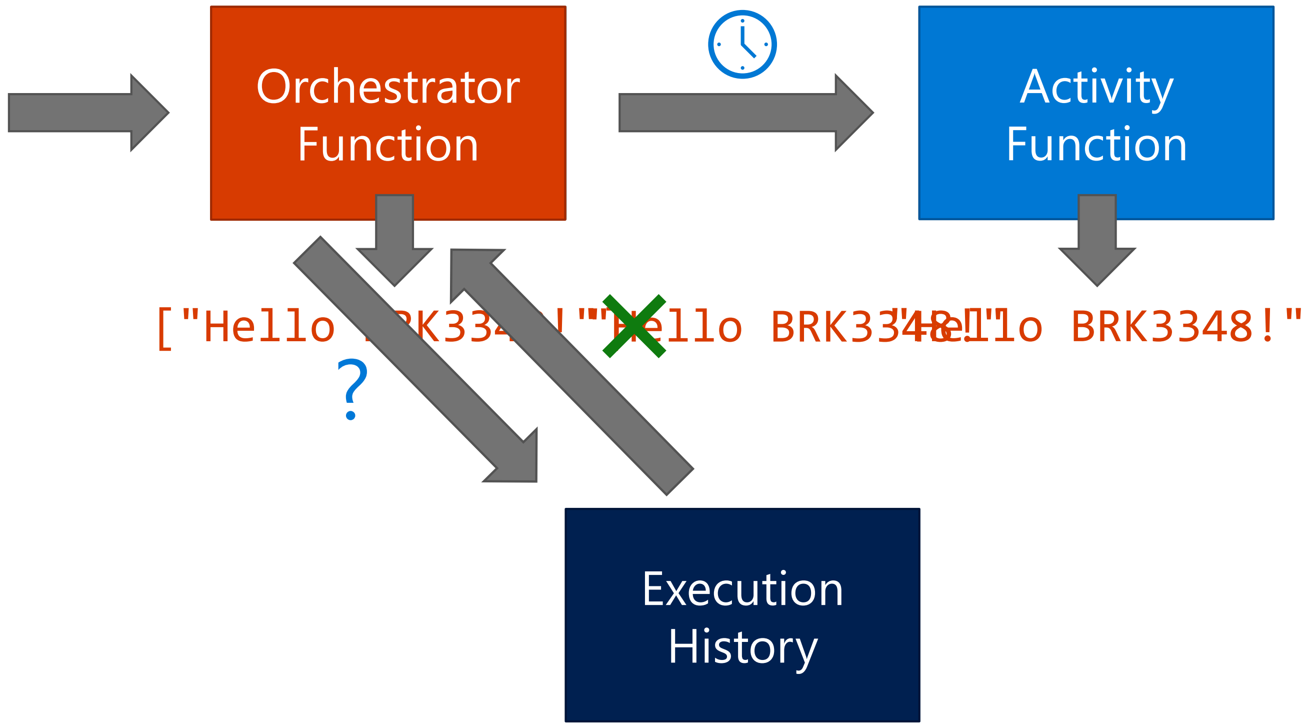
Microsoft.Azure.Webjobs.Extensions.MicrosoftGraph 1.0.0-beta

Durable Functions

```
var outputs = new List<string>();

outputs.Add(await context.CallActivityAsync<string>("SayHello", "BRK3348"));

return outputs;
```



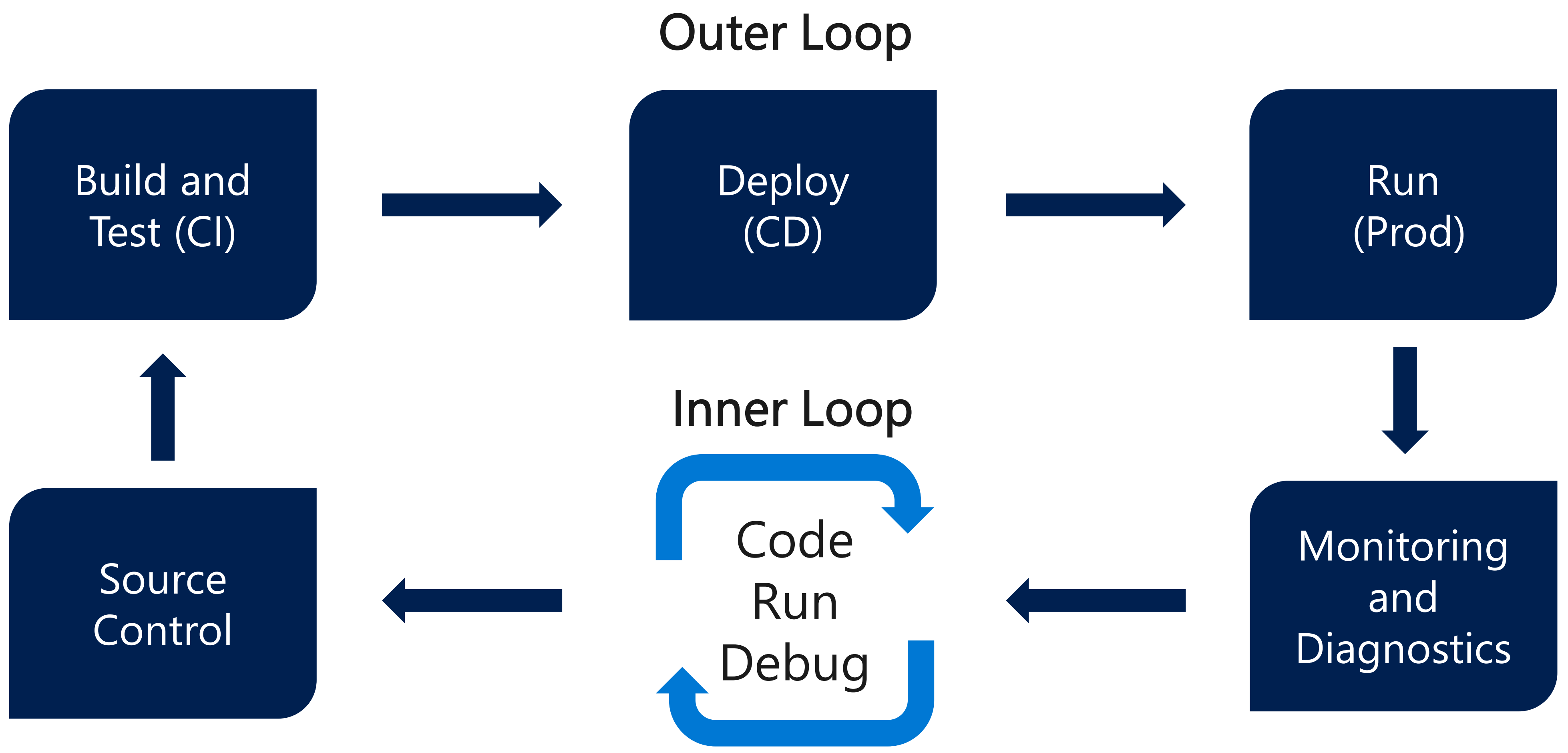
History Table

Orchestrator Started
Execution Started
Task Scheduled, SayHello, "BRK3348"
Orchestrator Completed
Task Completed, "Hello BRK3348!"
Orchestrator Started
Execution Completed, ["Hello BRK3348!"]
Orchestrator Completed

Microsoft Azure

DevOps and Serverless

Inner and Outer Loop Development



Available tools of Azure Functions

Local Tools	**Deployment Center (Kudu)**	**Azure DevOps**	**Other CI/CD**
Quickly publish to production	App Services powered CI/CD	Fully managed CI/CD	Any other CI/CD tool (Jenkins, Octopus, Travis)
Best Suited – Quickly validate code works in the cloud	**Best Suited –** One-click deploy from GitHub/source	**Best Suited –** Production CI/CD with various environments	**Best Suited –** Integrated serverless with existing tools and processes
Watch out – “Friends don’t let friend right-click publish”	**Watch out –** Not as customizable as Azure DevOps pipelines	**Watch out –** Web Deploy vs Run from Package	**Watch out –** Documentation and samples are limited
Tip – Use the ‘run from package’ feature	**Tip –** Use the new “Deployment Center” section	**Tip –** Can call functions as release gates	**Tip –** Use the ‘run from package’ publish gesture

Inner Loop Best Practices

- Write unit tests for your functions
  - How should this behave on success?
  - How should this behave on failure?
  - Mock external systems and side-effects
- Environment variables == local.settings.json == Application Settings
  - local.settings.json when local
  - Application Settings when published

Outer Loop Best Practices

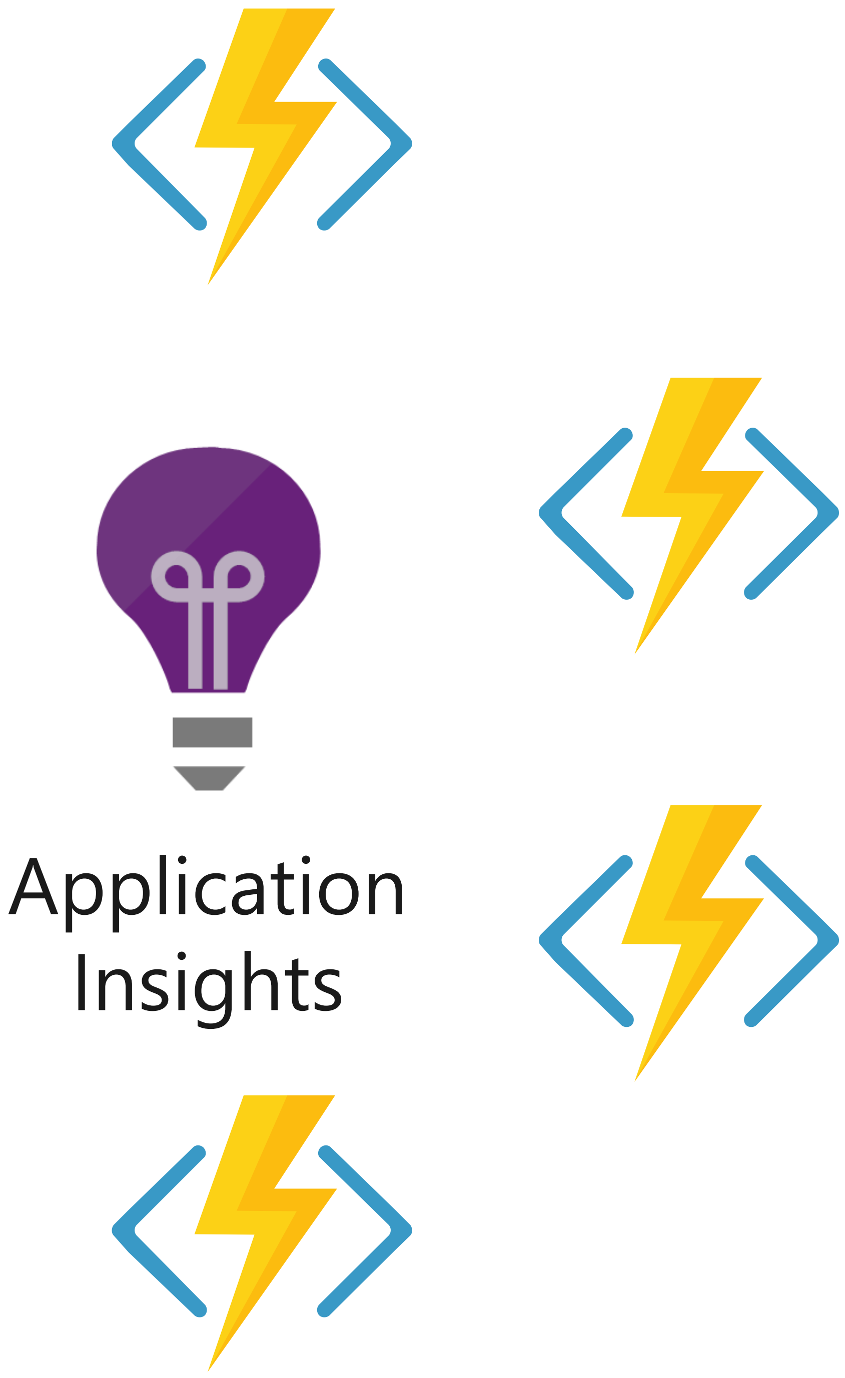
- Every function should be checked into source control
- Feature / Bug branches should be tested before merge into master
  - Code compiles
  - Unit tests pass
- Deploy to stages
  - Development, Pre-Prod, Prod
  - Use release gates and approval (auto or manual) between stages
- Use slots where appropriate
  - Optimized today for HTTP scenarios
  - Be aware of scaling impacts to consumption functions
- Enable Application Insights

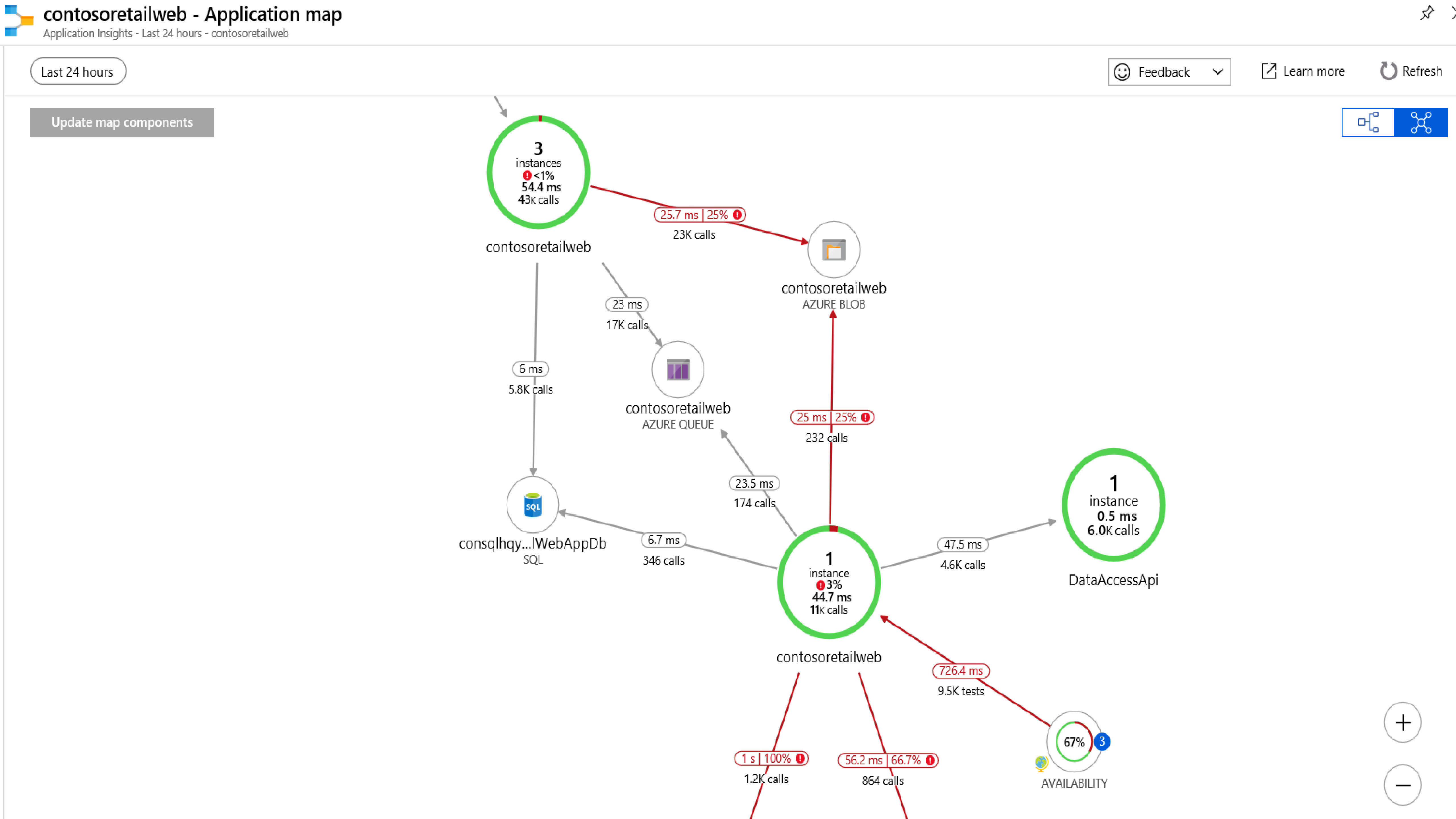
Monitoring

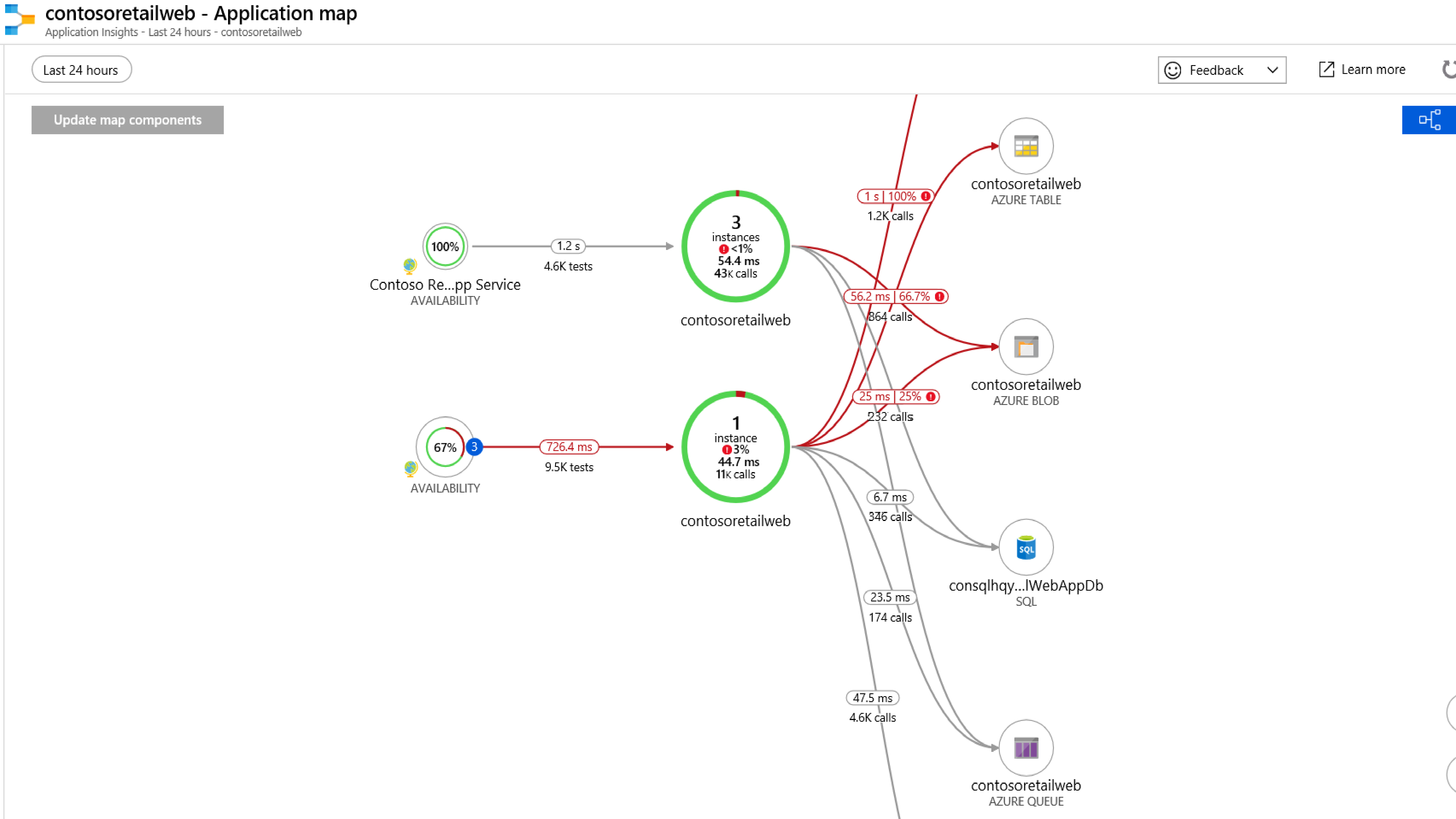
Gain real-time observability

Analyze and debug traces and metrics

View dependencies and relationships with AppMap




contosoretailweb - Application map
Application Insights - Last 24 hours - contosoretailweb
Last 24 hours
Feedback
Learn more
Refresh
Update map components
3 instances <1% 54.4 ms 43K calls
contosoretailweb
25.7 ms | 25%
23K calls
contosoretailweb
AZURE BLOB
23 ms
17K calls
6 ms
5.8K calls
contosoretailweb
AZURE QUEUE
25 ms | 25%
232 calls
23.5 ms
174 calls
consqlhqy...lWebAppDb
SQL
6.7 ms
346 calls
1 instance 3% 44.7 ms 11K calls
contosoretailweb
47.5 ms
4.6K calls
1 instance 0.5 ms 6.0K calls
DataAccessApi
726.4 ms
9.5K tests
1 s | 100%
1.2K calls
56.2 ms | 66.7%
864 calls
67%
3
AVAILABILITY


contosoretailweb - Application map
Application Insights - Last 24 hours - contosoretailweb
Last 24 hours
Feedback
Learn more
Update map components
100%
Contoso Re...pp Service
AVAILABILITY
1.2 s
4.6K tests
3
instances
<1%
54.4 ms
43K calls
contosoretailweb
67%
3
AVAILABILITY
726.4 ms
9.5K tests
1
instance
3%
44.7 ms
11K calls
contosoretailweb
1 s | 100%
1.2K calls
56.2 ms | 66.7%
864 calls
25 ms | 25%
232 calls
6.7 ms
346 calls
23.5 ms
174 calls
47.5 ms
4.6K calls
contosoretailweb
AZURE TABLE
contosoretailweb
AZURE BLOB
consqlhqy...lWebAppDb
SQL
contosoretailweb
AZURE QUEUE

Azure Functions Demo

Portal
Visual Studio Code
Visual Studio
Azure DevOps

Microsoft Azure

Serverless Security

The shared responsibility model

	Physical security	Host infrastructure	Operating system	Network controls	Identity & access management	Logical application server	Application logic	Client endpoints	Data classification
On-prem	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
IaaS	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜
PaaS	🟦	🟦	🟦	🟦/⬜	🟦/⬜	🟦/⬜	⬜	⬜	⬜
Serverless	🟦	🟦	🟦	🟦/⬜	🟦/⬜	🟦	⬜	⬜	⬜
SaaS	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦/⬜	⬜

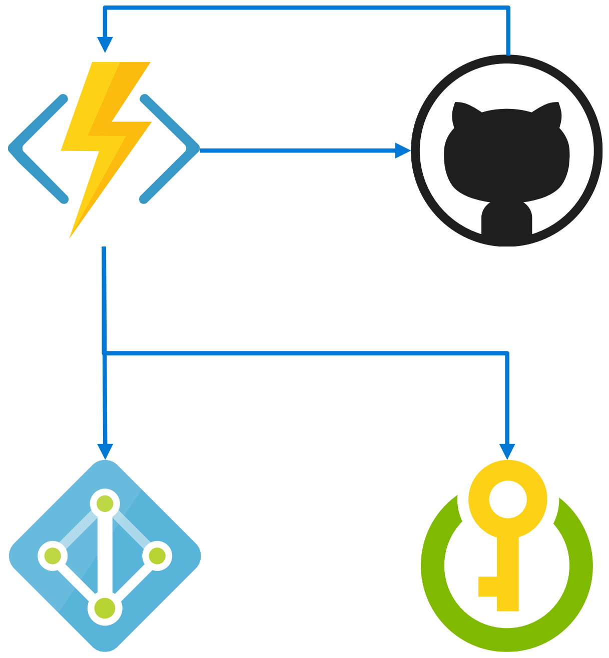
🟦 Cloud provider ⬜ Cloud customer

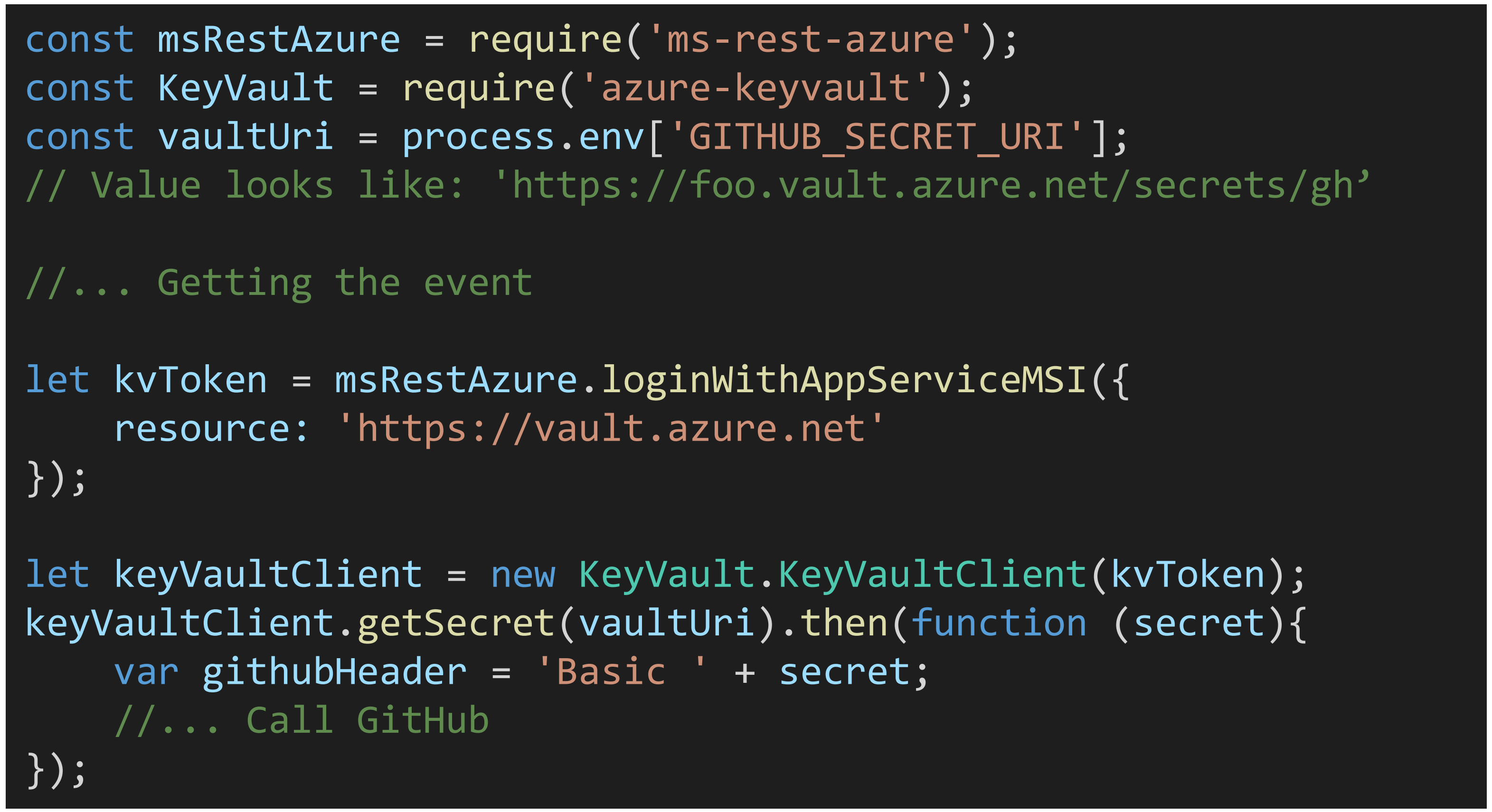
Spot the vulnerability!

```
module.exports = function (context, payload) {
    if (payload.action != "opened") {
        context.done();
        return;
    }
    var comment = { "body": "Thank you for your contribution! We will get to it shortly." };
    if (payload.pull_request) {
        var pr = payload.pull_request;
        context.log(pr.user.login, " submitted PR#", pr.number, ": ", pr.title);
        SendGitHubRequest(pr.comments_url, comment, context); // posting a comment
    }
    context.done();
};

function SendGitHubRequest(url, requestBody, context) {
    var request = require('request');
    var githubCred = 'Basic ' + 'mattchenderson:8e254ed4';
    request({
        url: url,
        method: 'POST',
        headers: {
            'User-Agent': 'mattchenderson',
            'Authorization': githubCred
        },
        json: requestBody
    }, function (error, response, body) {
        if (error) {
            context.log(error);
        } else {
            context.log(response.statusCode, body);
        }
    });
}
```

Secrets management





```
const msRestAzure = require('ms-rest-azure');
const KeyVault = require('azure-keyvault');
const vaultUri = process.env['GITHUB_SECRET_URI'];
// Value looks like: 'https://foo.vault.azure.net/secrets/gh’

//... Getting the event

let kvToken = msRestAzure.loginWithAppServiceMSI({
    resource: 'https://vault.azure.net'
});

let keyVaultClient = new KeyVault.KeyVaultClient(kvToken);
keyVaultClient.getSecret(vaultUri).then(function (secret){
    var githubHeader = 'Basic ' + secret;
    //... Call GitHub
});
```

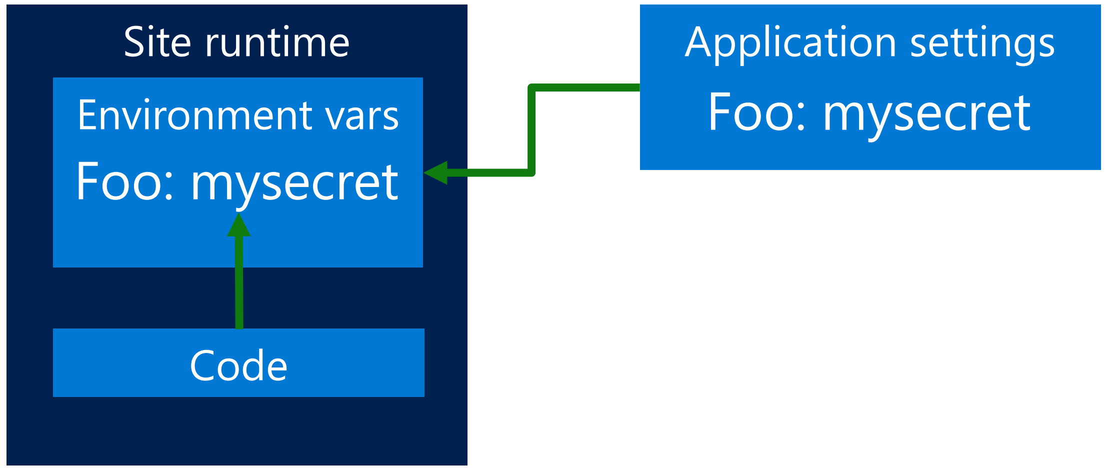
Coming soon: Key Vault references

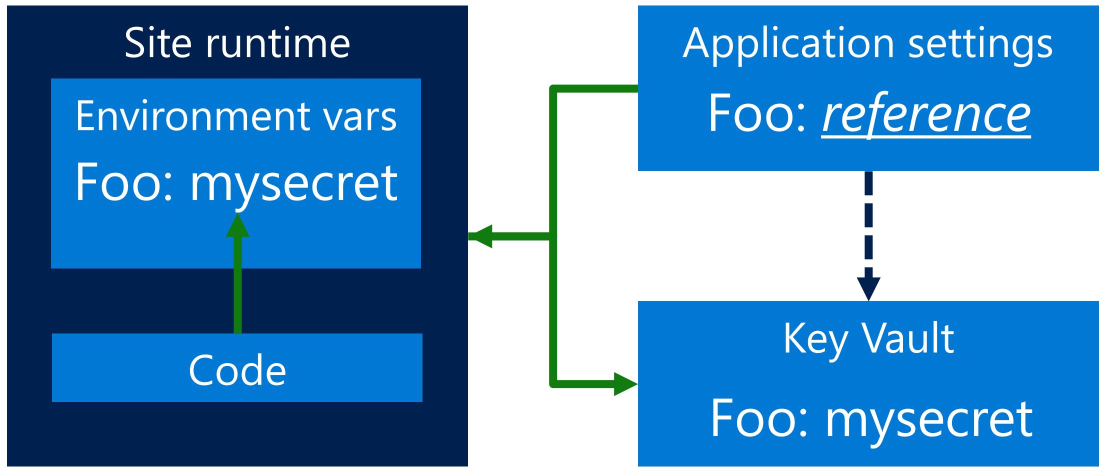
@Microsoft.KeyVault(SecretUri=https://**myvault**.vault.azure.net/secrets/**mysecret**/**mysecretversion**)

Gets secrets out of App Settings and into secrets management

Leverages the managed identity of your function app

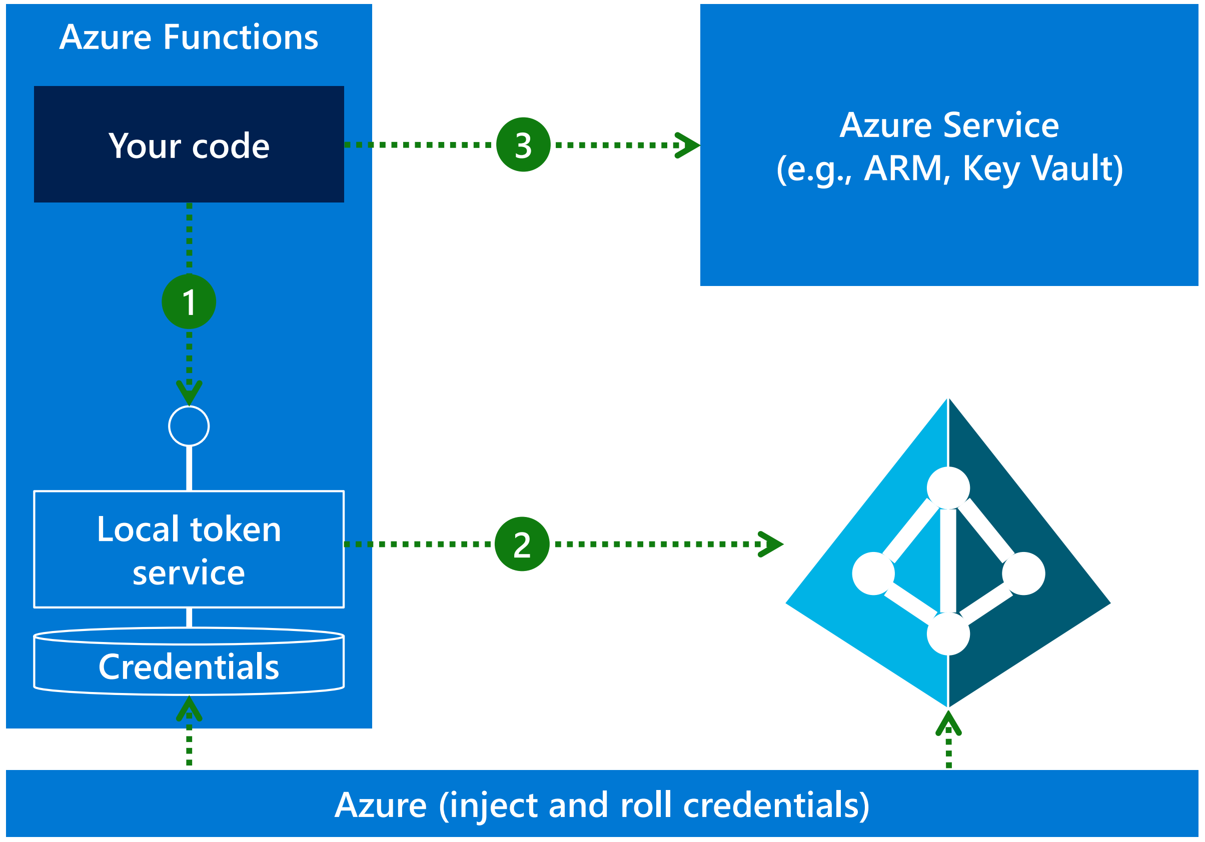
Versions will be required at initial preview (goal of auto-rotation)



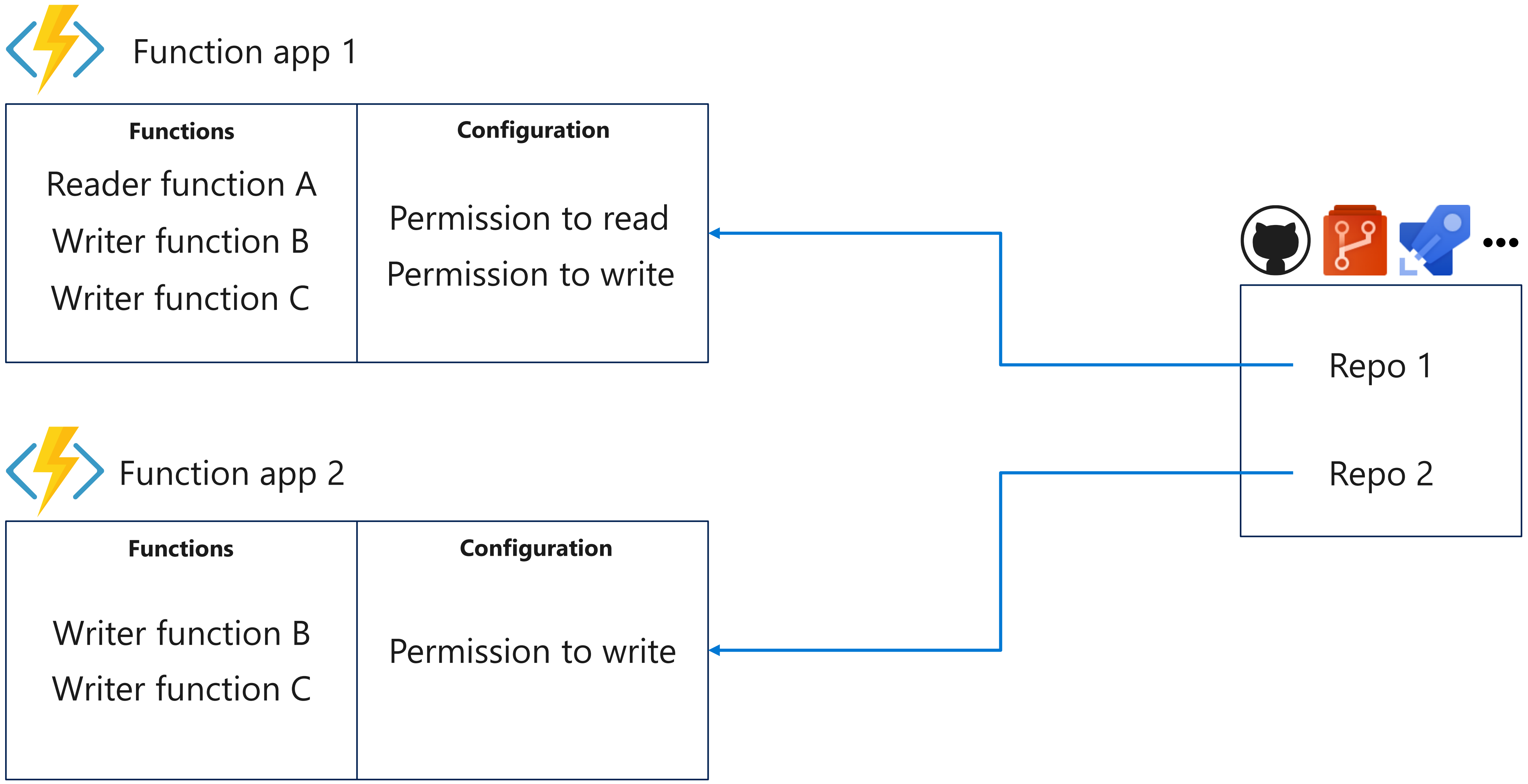


Managed identities for Azure Functions

- Keep credentials out of code
- Auto-managed identity in Azure AD for Azure resource
- Use local token endpoint to get access tokens from Azure AD
- Direct authentication with services, or retrieve creds from Azure Key Vault

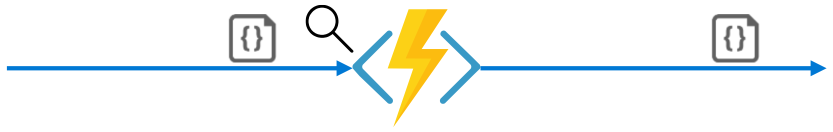

Azure Functions
Your code
1
Local token service
Credentials
3
Azure Service (e.g., ARM, Key Vault)
2
Azure (inject and roll credentials)

Grouping and permissions



Inputs AND outputs

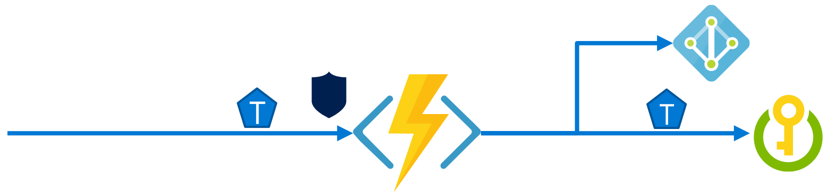
Am I validating inputs and preventing injection attacks?



Sanitization

Am I validating outputs?

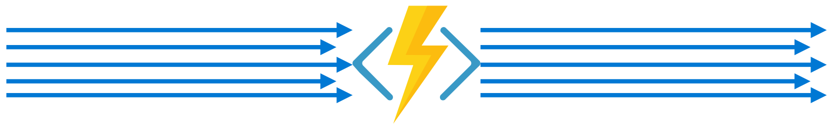
Am I applying proper authorization checks?



Permissions

Am I granting proper roles and permissions? Am I enforcing least privilege?

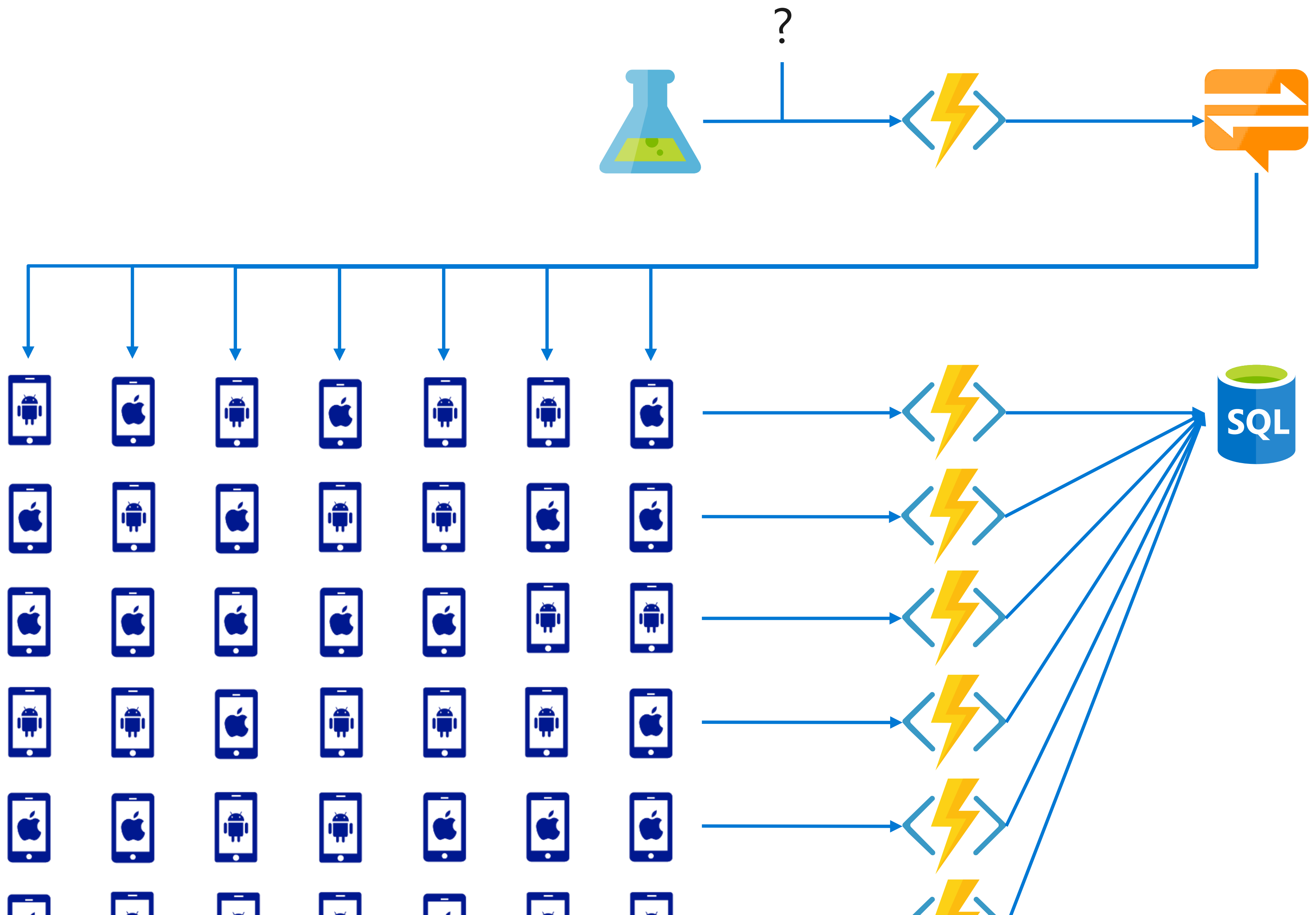
Can my app scale well in response to new events?



Scalability

Can my downstream resources keep up with my scale?

When scaling goes wrong



Serverless security best practices

- Standard PaaS / web app security is still a must-have
- New security tooling options needed

- More secrets, more secret management
- Permissions and grouping – remember least privilege
- Mind both inputs and outputs – the app is only as secure as its weakest link

- Networking solutions need development, but…

Hosting and Connectivity

Functions hosting options

	Development	Hosting					
	Local dev machine	Azure Functions service	Azure Functions service	IoT devices	Additional Azure hosts	Non-Azure hosts	On-premises
Platform	Core Tools + favorite editor	Consumption plan	App Service plan	Azure IoT Edge	AKS, Service Fabric Mesh, …	K8s, raw VMs, & more	App Service on Azure Stack
Application delivery	Code or container	Code	Code or container	Container	Container	Container	Code
Operating system	Windows, macOS, or Linux	Windows or Linux	Windows or Linux	Linux	Linux	Linux	Windows

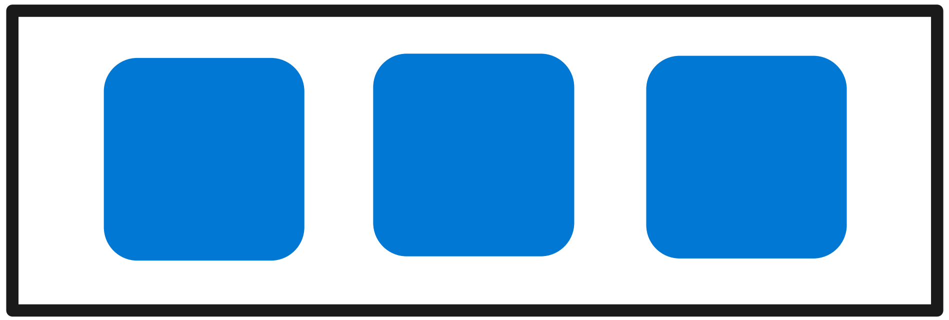
Azure Functions Hosting Options

Consumption

- Rapid scale out
- "Unbounded" scale out
- No VNet connectivity available
- 10 minute execution
- Small instance size
- Scale to zero

App Service Plan / Environment

- Auto-scale out (~5 min)
- Fixed scale out(Max=10 nodes)
- VNet connectivity / hybrid
- Unlimited execution duration
- Premium instance size
- Always on



Azure Functions Hosting Options

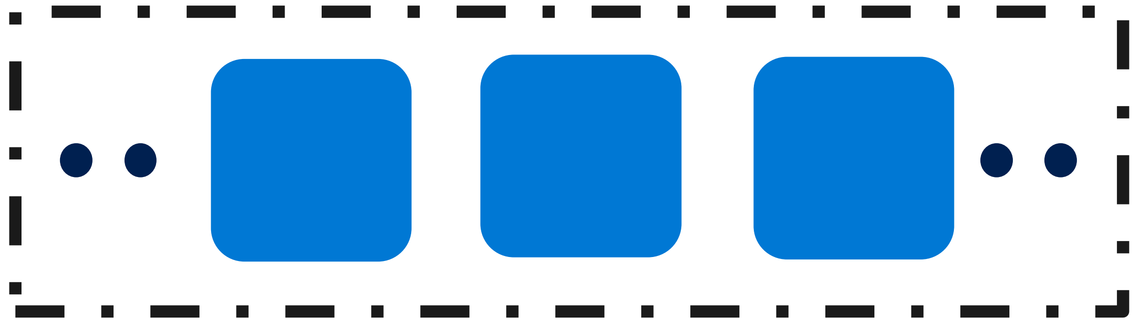
Consumption

- Rapid scale out
- "Unbounded" scale out
- No VNet connectivity available
- 10 minute execution
- Small instance size
- Scale to zero (cold start)

PRIVATE PREVIEW

Functions premium plan

- Rapid scale out
- "Unbounded" scale out
- VNet connectivity / hybrid
- Unlimited execution duration
- Premium instance size
- Always on



Takeaways and next actions

- Functions 2.0 – greater flexibility and control
  - Go create your first 2.0 function!
  - Sign up for the Python private preview: <<>>
- DevOps – keep both inner loop and outer loop in mind
  - Deployment Center and Azure DevOps are powerful tools!
  - Test samples: https://github.com/jeffhollan/functions-test-samples
- Security in serverless – familiar issues, but at different scale
  - Key Vault App Settings coming soon
- Premium Functions – networking, reserved instances, & more
  - Sign up for the private preview: http://aka.ms/functionspremium

Microsoft