GLOBAL INTEGRATION BOOTCAMP

Azure Serverless

Develop Modern Software Architectures

Powered by Microsoft - Saturday March 24th 2018
UK event sponsored by BizTalk360

GLOBAL INTEGRATION BOOTCAMP

Azure Serverless: Modern Software Architectures

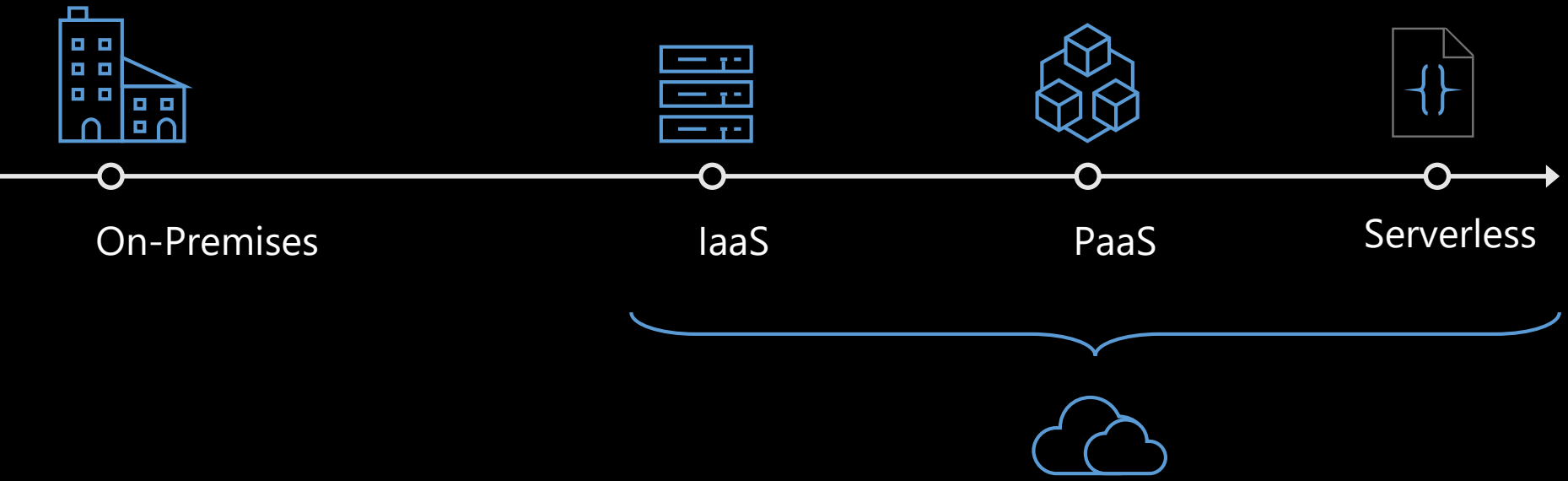
Saffieldin Ali
Senior Premier Field Engineer
Azure Development Advisory Services

About Me

- Saffi Ali
- Sr. Azure Premier Field Engineer

  Development Advisory Services for Azure
- MCSE: Cloud and Infrastructure
- WW Integration Community SME
- Active Community Speaker
- Blockchain Enthusiast

The evolution of application platforms



What is Serverless?

Abstraction
of servers

Event-driven/
instant scale

Micro-billing

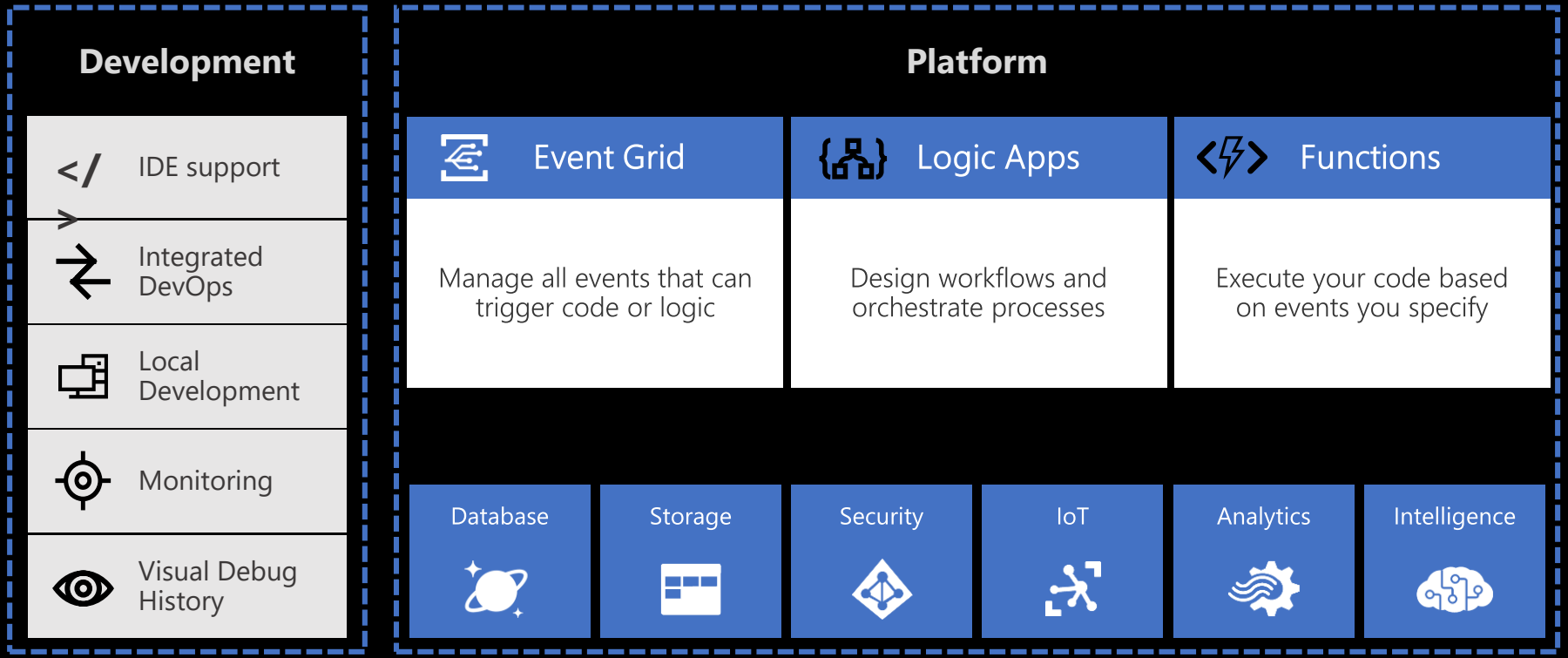
Benefits of Serverless

Manage apps, not servers

Reduced DevOps

Faster time to market

Serverless application platform components



What is the “Functions” programming model?

Function as the unit of work

Function

Function

Function

```
module.exports = function (context, eventHubMessages) {
    context.log(`JavaScript eventhub trigger function called for message array ${eventHubMessages}`);

    eventHubMessages.forEach(message => {
        context.log(`Processed message ${message}`);
    });

    context.done();
};
```

```
}
```

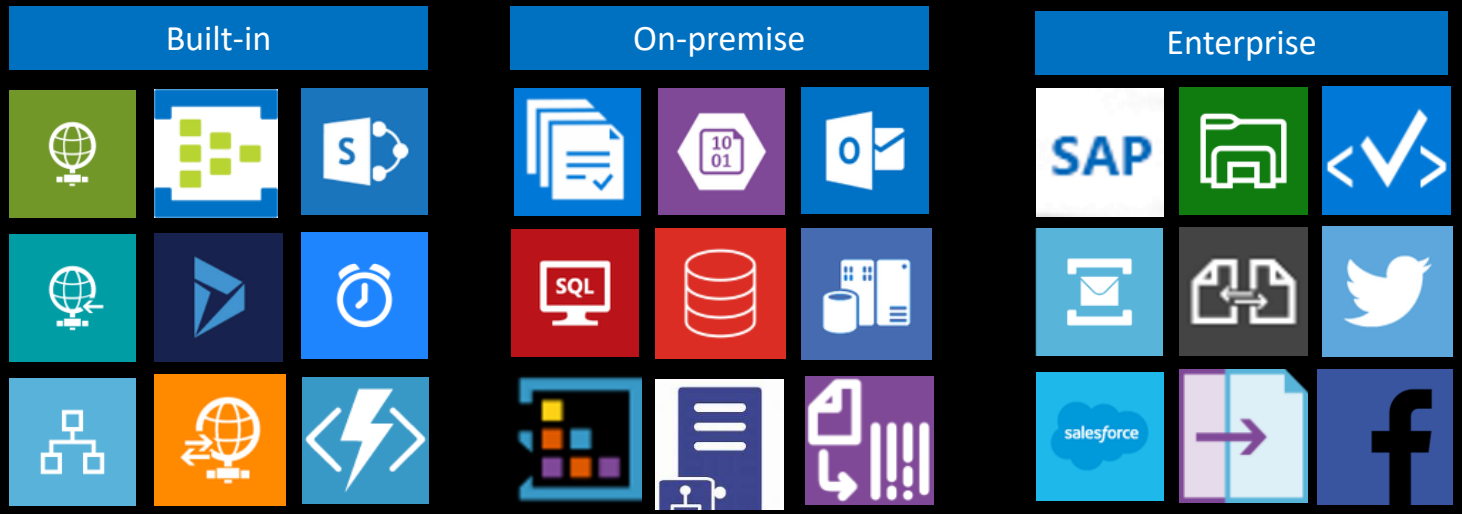
Demo

Using Functions in the Azure Portal

Triggers and Bindings

Type	1.x	2.x	Trigger	Input	Output
Blob Storage	✓	✓[1]	✓	✓	✓
Cosmos DB	✓	✓	✓	✓	✓
Event Grid	✓	✓	✓		
Event Hubs	✓	✓	✓		✓
External File[2]	✓			✓	✓
External Table[2]	✓			✓	✓
HTTP	✓	✓[1]	✓		✓
Microsoft Graph Excel tables		✓		✓	✓
Microsoft Graph OneDrive files		✓		✓	✓
Microsoft Graph Outlook email		✓			✓
Microsoft Graph Events		✓	✓	✓	✓
Microsoft Graph Auth tokens		✓		✓	
Mobile Apps	✓	✓		✓	✓
Notification Hubs	✓				✓
Queue storage	✓	✓[1]	✓		✓
SendGrid	✓	✓			✓
Service Bus	✓	✓	✓		✓
Table storage	✓	✓[1]		✓	✓
Timer	✓	✓	✓		
Twilio	✓	✓			✓
Webhooks	✓		✓		✓

More bindings

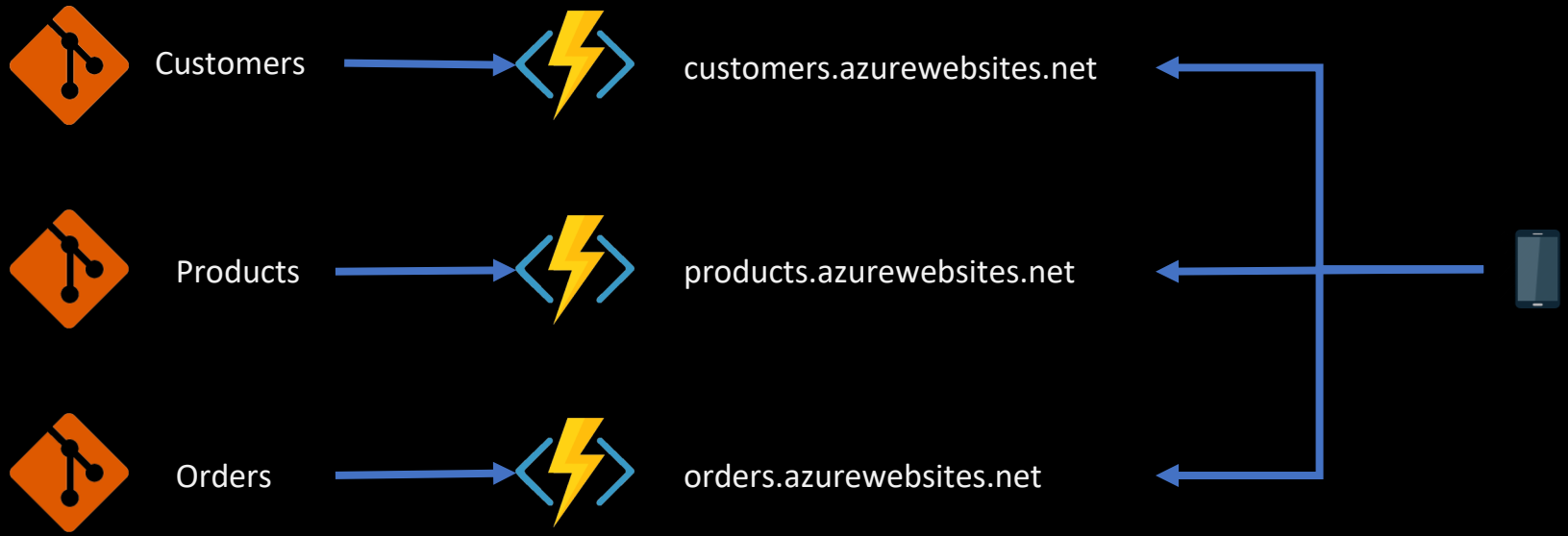


Over 190 LogicApp connectors and ever expanding EventGrid events

Azure Functions Tooling Options

Tools	Windows	MacOS	Linux
Portal (with Kudu)	✓	✓	✓
Visual Studio	✓		
Visual Studio Mac		✓	
Visual Studio Code	✓	✓	✓
Function Core Tools	✓	✓	✓

Deployment and management isolation



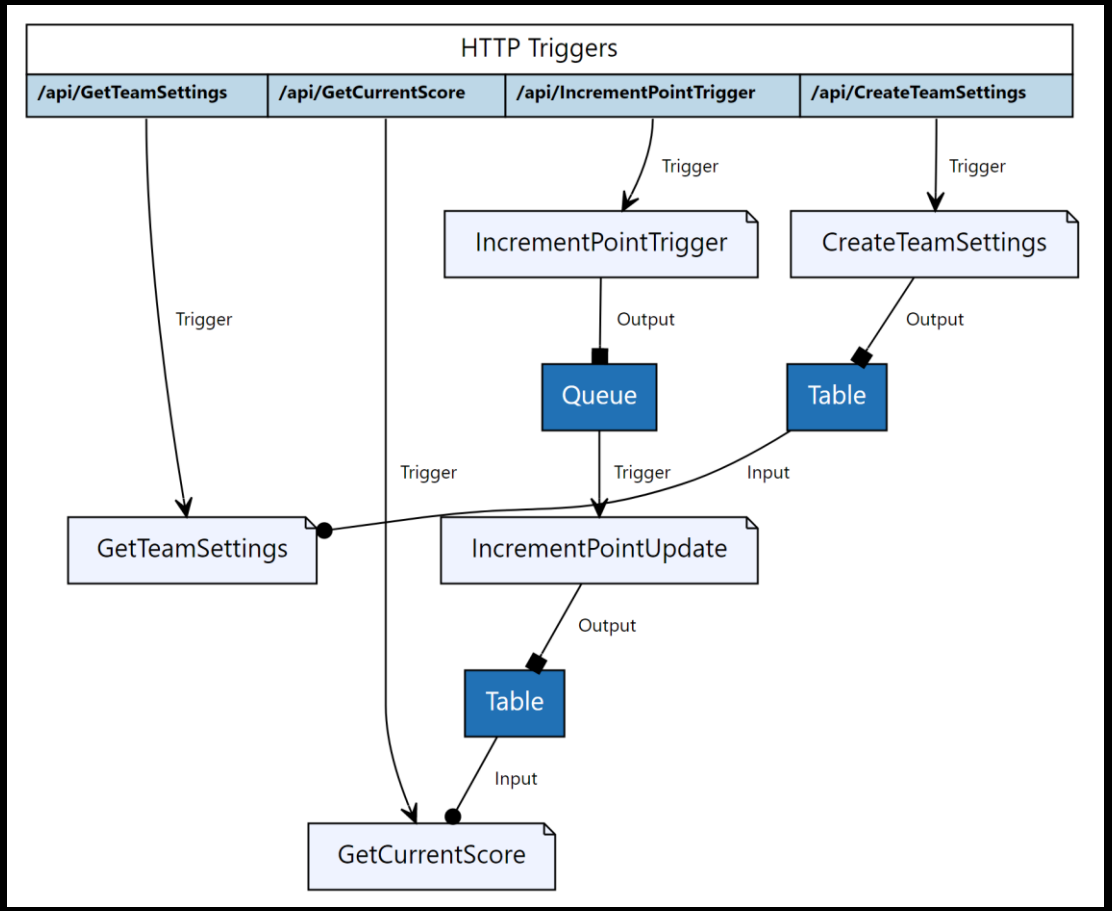
Serverless APIs

Proxies
Security (EasyAuth)
OpenAPI (Swagger)
CORS (Cross Origin Resource Sharing)
Flow & PowerApps integration
API Management

Demo: aka.ms/GIB2018

Azure Functions Game

Serverless Game



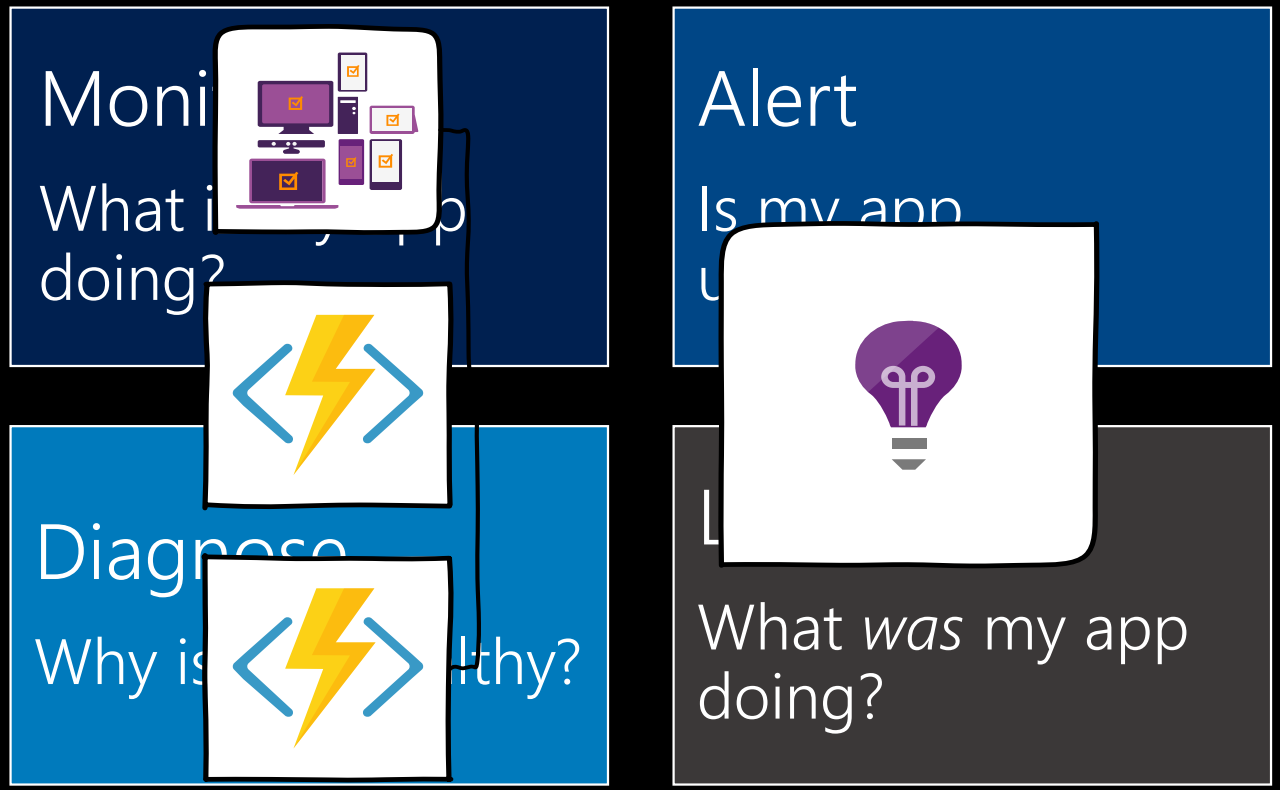
Azure Functions Bindings Extensibility

```
using System.Net;
using System.Net.Http;
using Microsoft.Azure.WebJobs;
using Microsoft.Azure.WebJobs.Host;
using SampleExtension;

namespace SampleFunctionApp
{
    public static class HttpTriggerSlack
    {
        [FunctionName("HttpTriggerSlack")]
        public static string Run(
            [HttpTrigger] SlackMessage message,
            [Slack(WebHookUrl = "SlackWebHook")] out SlackMessage slackMessage,
            TraceWriter log)
        {
            slackMessage = message;

            return "Ok";
        }
    }
}
```

Development tools

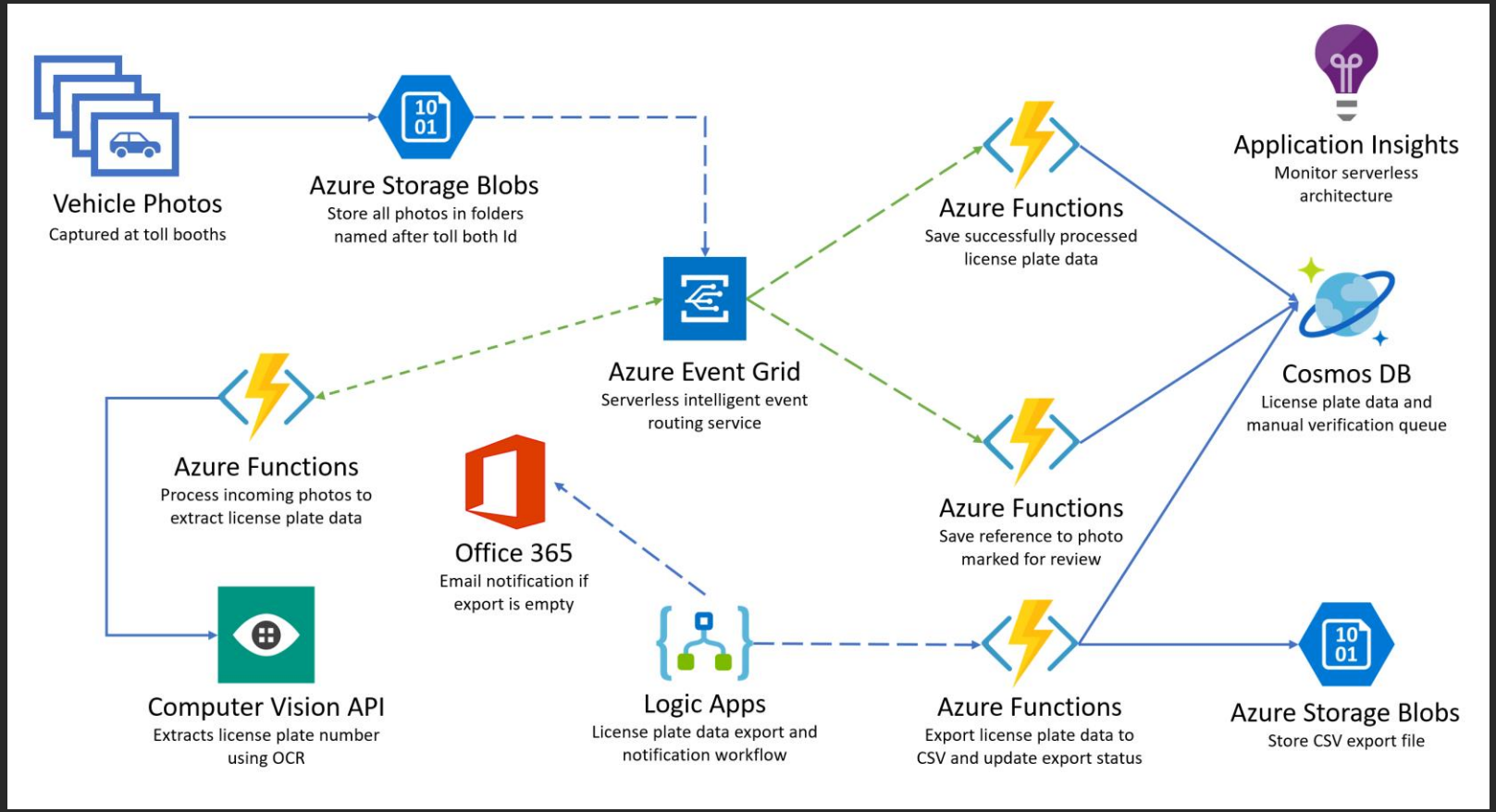
Monitoring: App Insights
Local debugging: VS, VS Mac, VS Code
Run locally: Azure Functions Core Tools
Publish: VSTS, Maven, Kudu
CI/CD: GitHub, BitBucket, Jenkins…

Monitoring with Application Insights



Let’s get real

Tollbooth number plate recognition



Serveless solution

License plate processing serverless components

- Orchestrate event-driven activities with Event Grid
- Use Azure Functions for serverless compute with the Consumption plan
- Research and test downstream services to see if they can handle high demand – implement rate-limiting and resiliency strategy as needed
- Use Cosmos DB to store license plate data




Serveless solution

License plate OCR

- Use the Cognitive Services Computer Vision API and its built-in OCR capabilities
- The image processing function can make a REST call to send the photo and read the JSON data in return

Serveless solution

Data export workflow

Create a Logic App with the following:

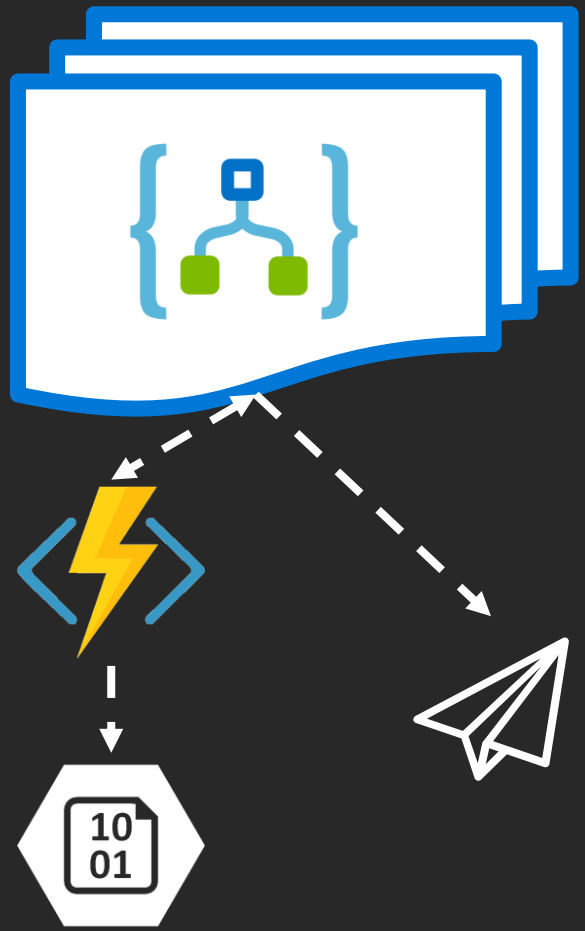
Recurrence trigger that executes every hour

then

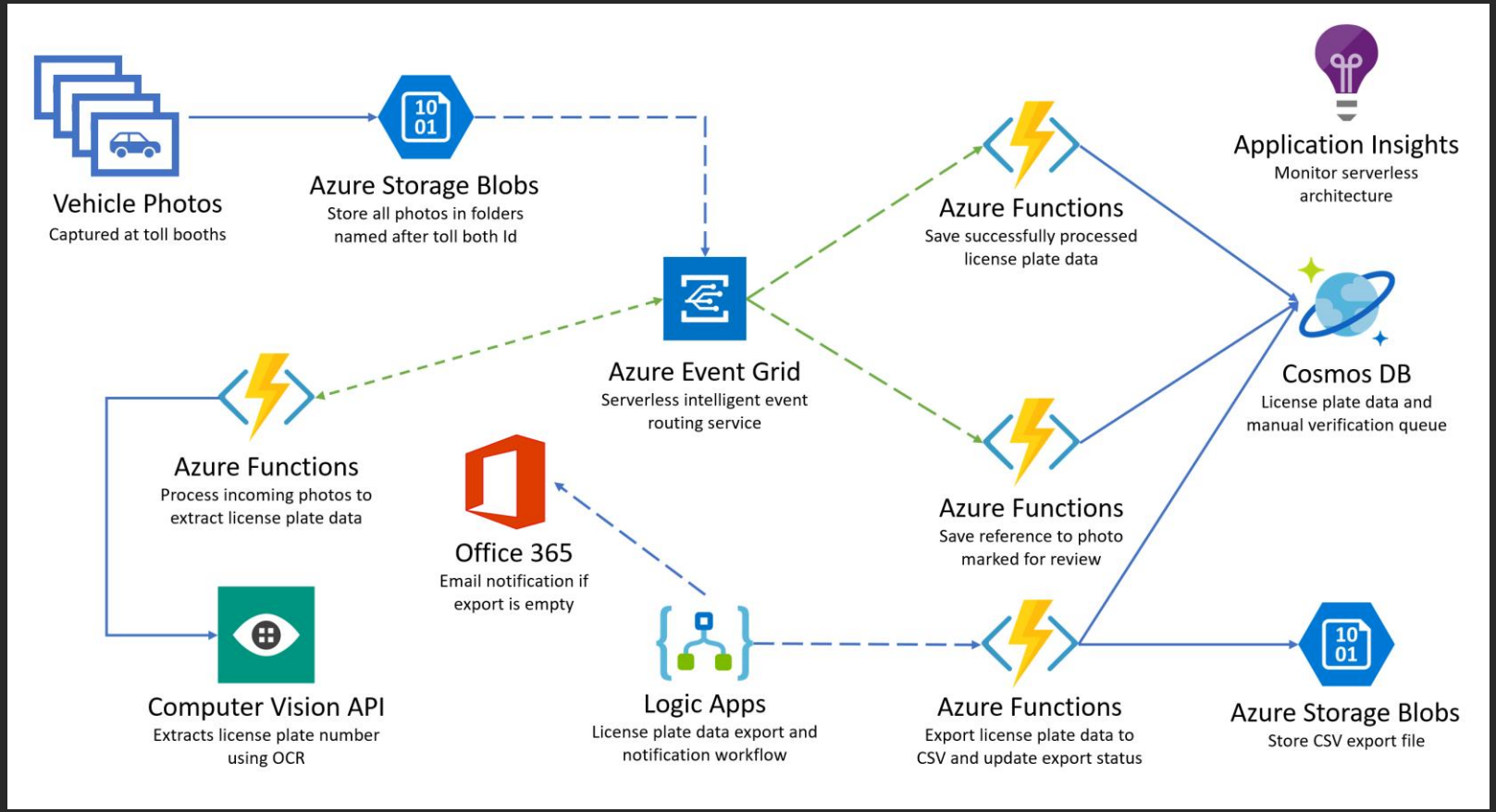
Executes Azure function that exports CSV to storage

then

Uses conditional logic to evaluate response code from the function and send email using Office 365 Outlook connector



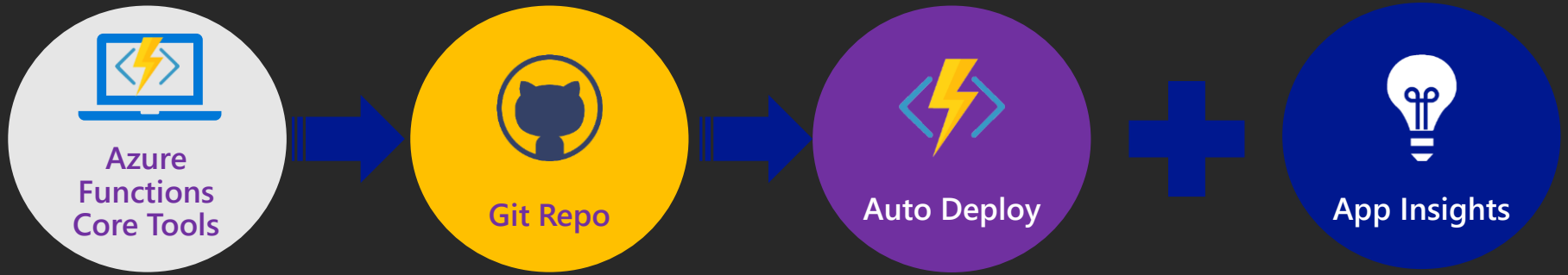
Tollbooth number plate recognition



Serveless solution

Monitoring and DevOps

- Develop & debug Function Apps locally with Azure Functions Core Tools
- Automate function deployments through App Service continuous integration – use integrated source repo like GitHub, DropBox or VSTS
- Monitor all executing serverless components with App Insights, in real-time, and use it to configure alerts and view historical telemetry



Customer questions

- How can the serverless components talk to each other?
- Will the dynamic scalability of the serverless components end up costing us a lot of money?
- How do we combat against erroneous image processing?

Azure Functions is an Open Source project

https://github.com/Azure/Azure-Functions

Azure WebJobs SDK script - the Azure Functions runtime

Azure WebJobs SDK - the "core" of the Azure Functions runtime and many bindings

Azure WebJobs SDK extensions - the repositories of many bindings

Azure Functions CLI - the command line tool for Azure Functions

Azure Functions Portal - the UX for the Functions development portal

Azure Functions templates - the templates which show up the Azure Functions portal

Azure Functions samples - repository for some samples on how the runtime works


Microsoft