Building an Ultra-Scalable API Using Azure Functions Without Too Much Worry

Chad Green
DogFoodCon
October 5, 2018

What will be covered

- What is serverless computing?
- Why to consider having a serverless API backend?
- Example of how to develop a serverless architecture
- Potential benefits and pitfalls

Who is Chad Green

Data & Solutions Architect at ProgressiveHealth

Community Involvement

Code PaLOUsa Conference Chair
Louisville .NET Meetup Organizer
Louisville Tech Leaders Meetup Organizer
Louisville Tech Ladies Co-Organizer

Contact Information

✉ chadgreen@chadgreen.com
chadgreen.com
@ChadGreen
in ChadwickEGreen

What is Serverless Computing

Building an Ultra-Scalable API Using Azure Functions Without Too Much Worry

• The evolution of application platforms

On-Premises

What media should I use to keep **backups**?

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

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

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

How can I increase **server** utilization?

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

Who **monitors** my **App**?

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

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

How can I **scale** my app?

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

Do I need a secondary **network connection**?

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

Which **Operating System** should I use?

What **happens** if the power goes out?

How many **servers** do I need?

Who **monitors** my **servers?**

Do I need a **UPS**?

What **storage** do I need to use?

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

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

How can I dynamically configure my app?

• The evolution of application platforms

IaaS

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 application?

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 **Operating System** should I use?

Who **monitors** my application?

• The evolution of application platforms

PaaS

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 application?

• The evolution of application platforms

Serverless

The platform for next generation applications

• What is Serverless?

Abstraction of Servers

Event-Driven/Instant Scale

Micro-Billing

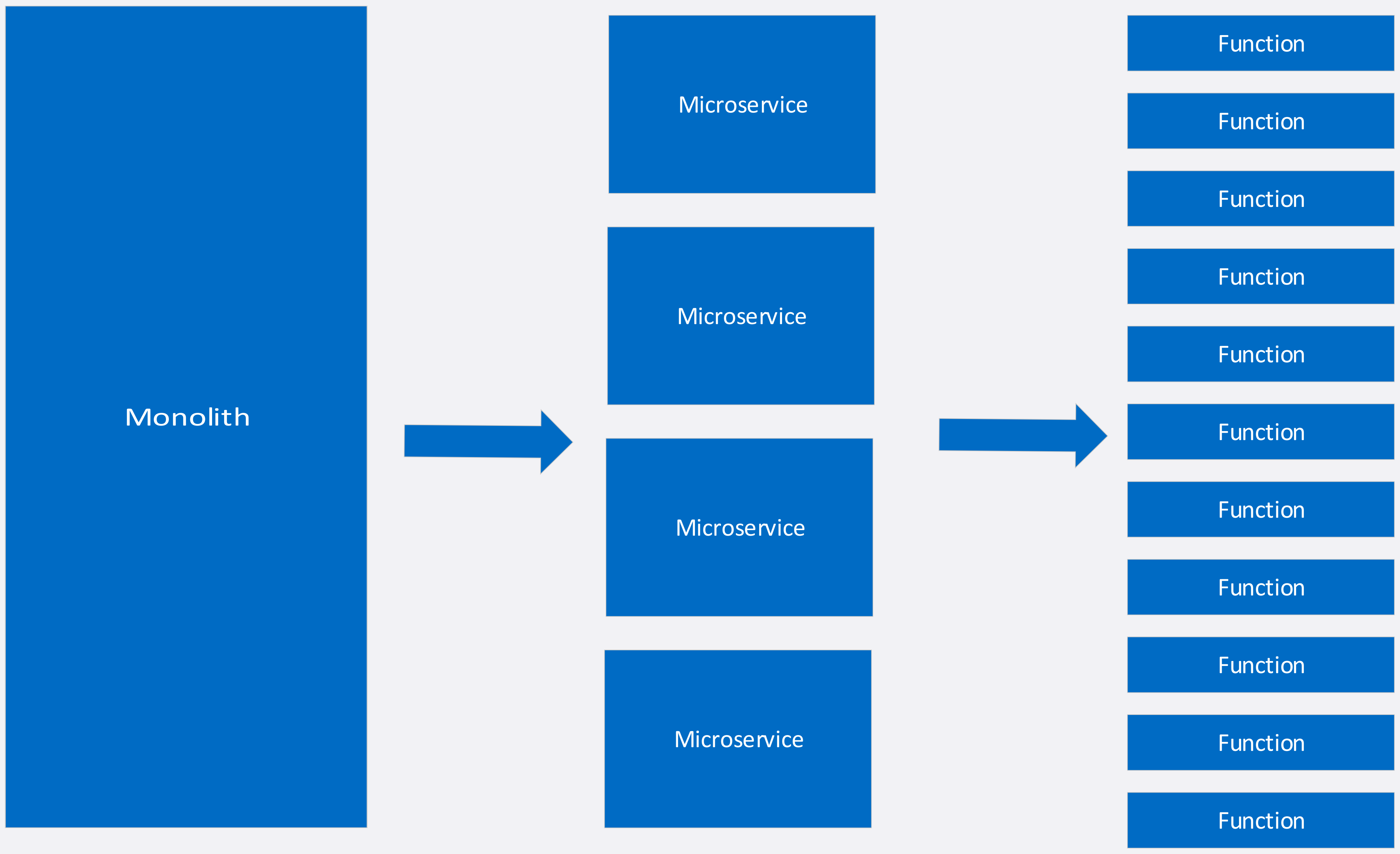
• Benefits of Serverless

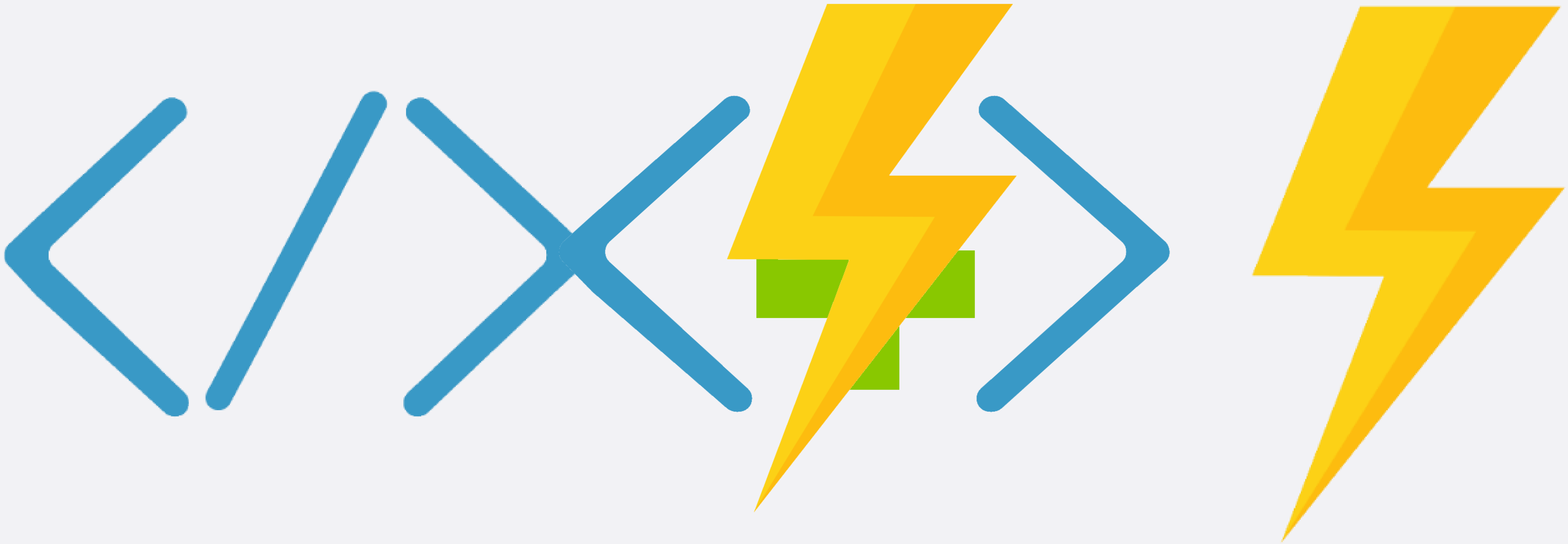
Manage apps not servers

Reduced DevOps

Faster Time to Market

• Serverless Scale





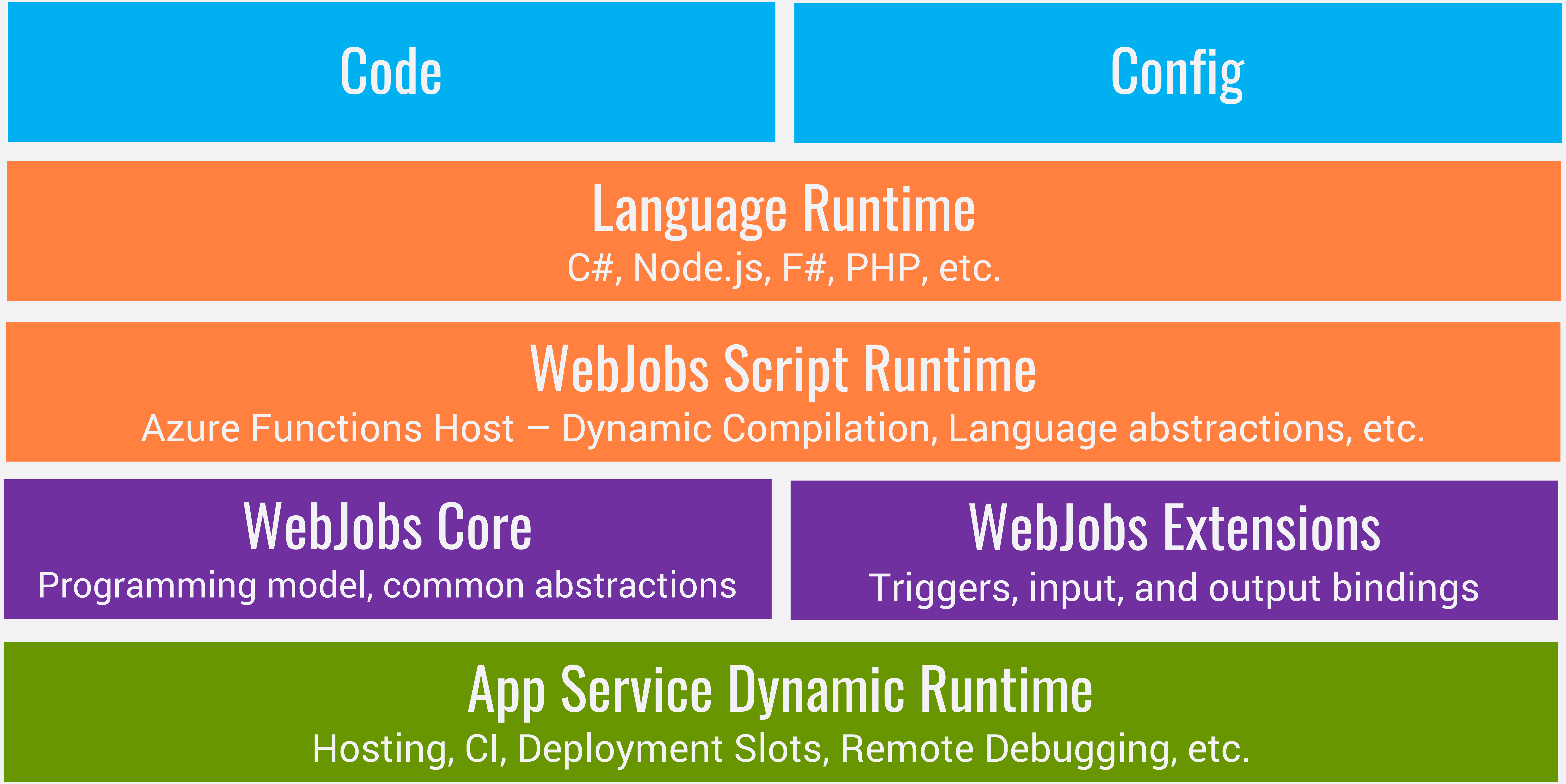
Code

Events + data

Azure **Functions**

Building an Ultra-Scalable API Using Azure Functions Without Too Much Worry

• Azure Functions Architecture



Code

Config

Language Runtime

C#, Node.js, F#, PHP, etc.

WebJobs Script Runtime

Azure Functions Host – Dynamic Compilation, Language abstractions, etc.

WebJobs Core

Programming model, common abstractions

WebJobs Extensions

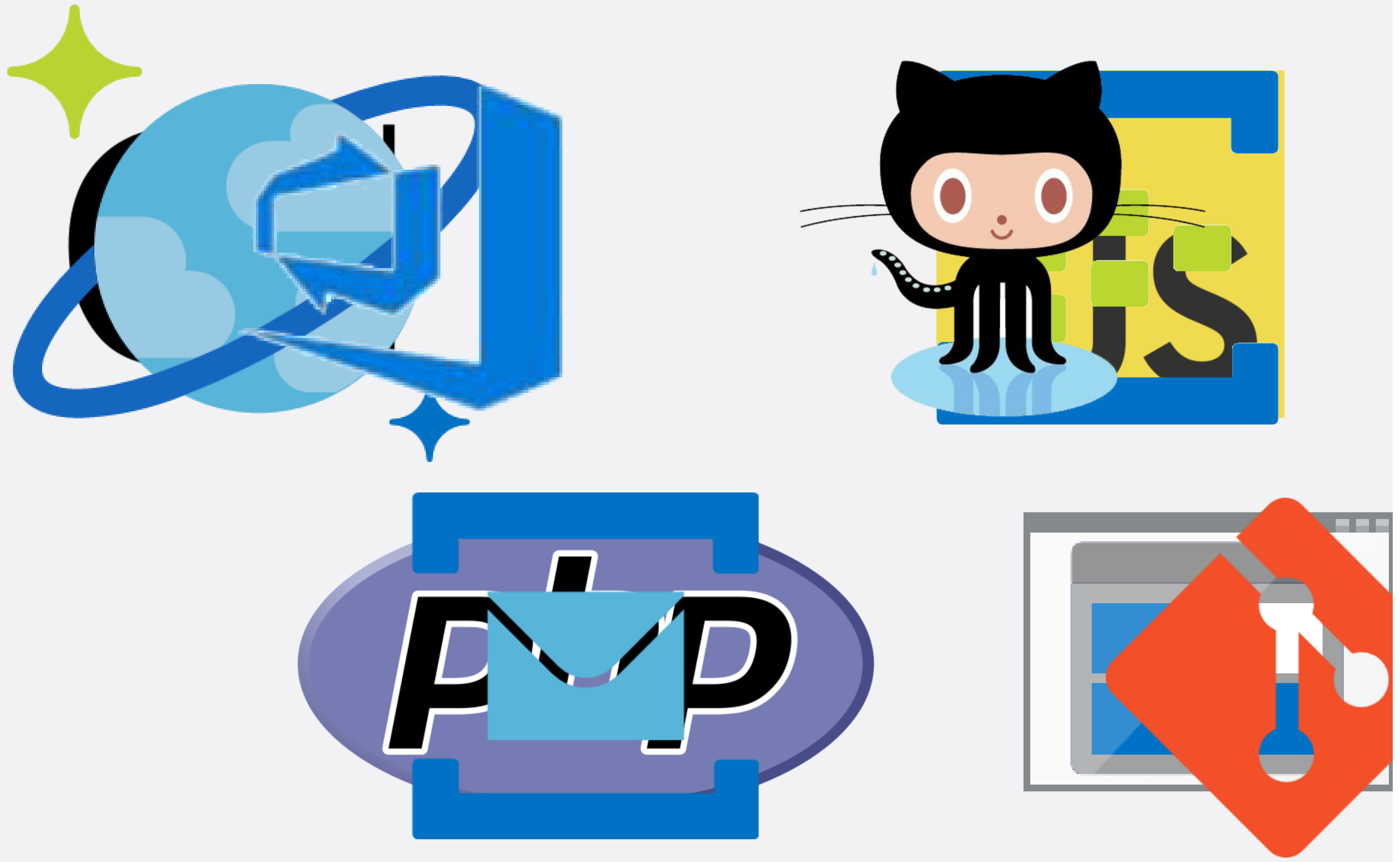
Triggers, input, and output bindings

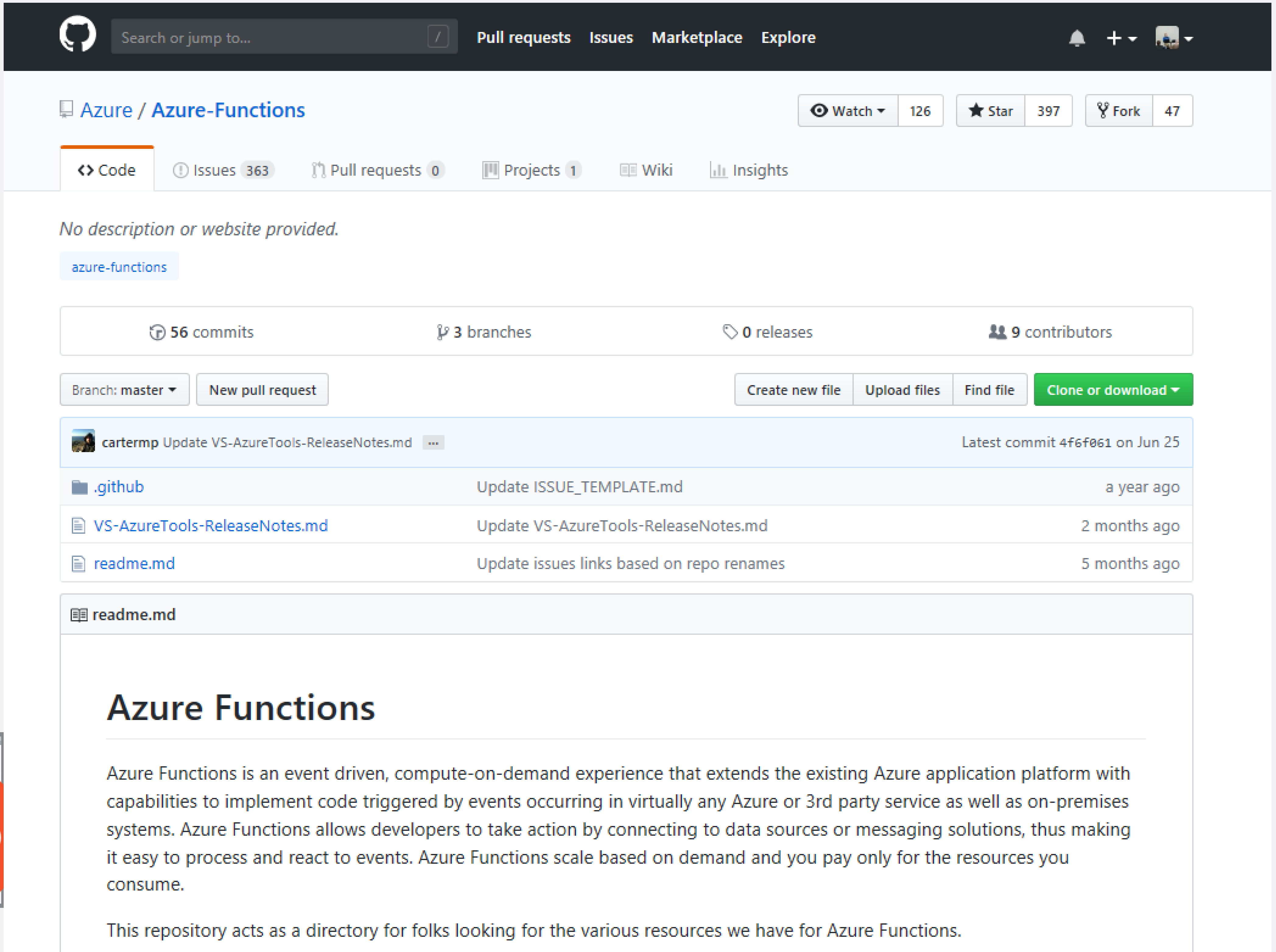
App Service Dynamic Runtime

Hosting, CI, Deployment Slots, Remote Debugging, etc.

• Features of Azure Functions

- Choice of language
- Pay-per-use pricing model
- Bring your own dependencies
- Integrated security
- Simplified integration
- Flexible development
- Open-source





Azure / **Azure-Functions**

No description or website provided.

56 commits | 3 branches | 0 releases | 9 contributors

cartermp Update VS-AzureTools-ReleaseNotes.md ...		Latest commit 4f6f061 on Jun 25
.github	Update ISSUE_TEMPLATE.md	a year ago
VS-AzureTools-ReleaseNotes.md	Update VS-AzureTools-ReleaseNotes.md	2 months ago
readme.md	Update issues links based on repo renames	5 months ago

readme.md

Azure Functions

Azure Functions is an event driven, compute-on-demand experience that extends the existing Azure application platform with capabilities to implement code triggered by events occurring in virtually any Azure or 3rd party service as well as on-premises systems. Azure Functions allows developers to take action by connecting to data sources or messaging solutions, thus making it easy to process and react to events. Azure Functions scale based on demand and you pay only for the resources you consume.

This repository acts as a directory for folks looking for the various resources we have for Azure Functions.

Why consider having a serverless API backend?

What does Scaling Mean?

Handling more Transactions

What does **Scaling** Mean?

~~**Handling more Transactions**~~

~~**per second**~~

Making more **Customers**

happy per second

Scalability Basics

Vertical Scaling

- Simple way to scale most software is simply run it on a more powerful machine
- Code performance improvements
- Many drawbacks
  - Costs rarely linear
  - Does not address redundancy

Horizontal Scaling

- Multiple instances of the application
- Can scale massively
- Introduces redundancy

Basic Principles to Follow

- Stateless
- Coarse Grained API
- Embrace Failure
- Avoid instance specific configuration
- Simple automated deployment
- Monitoring
- KISS – Keep It Small and Simple

Design Goals

- Distribute API Development
- Support multiple languages
- Minimize latency
- Minimize deployment risks

Example Architecture

Building an Ultra-Scalable API Using Azure Functions Without Too Much Worry

Variable Barrier Reach System Needs

- Dynamic Placement
  - Physical Demand Analysis
  - Post Offer/Transfer Evaluation
  - Variable Barrier Reach Evaluation

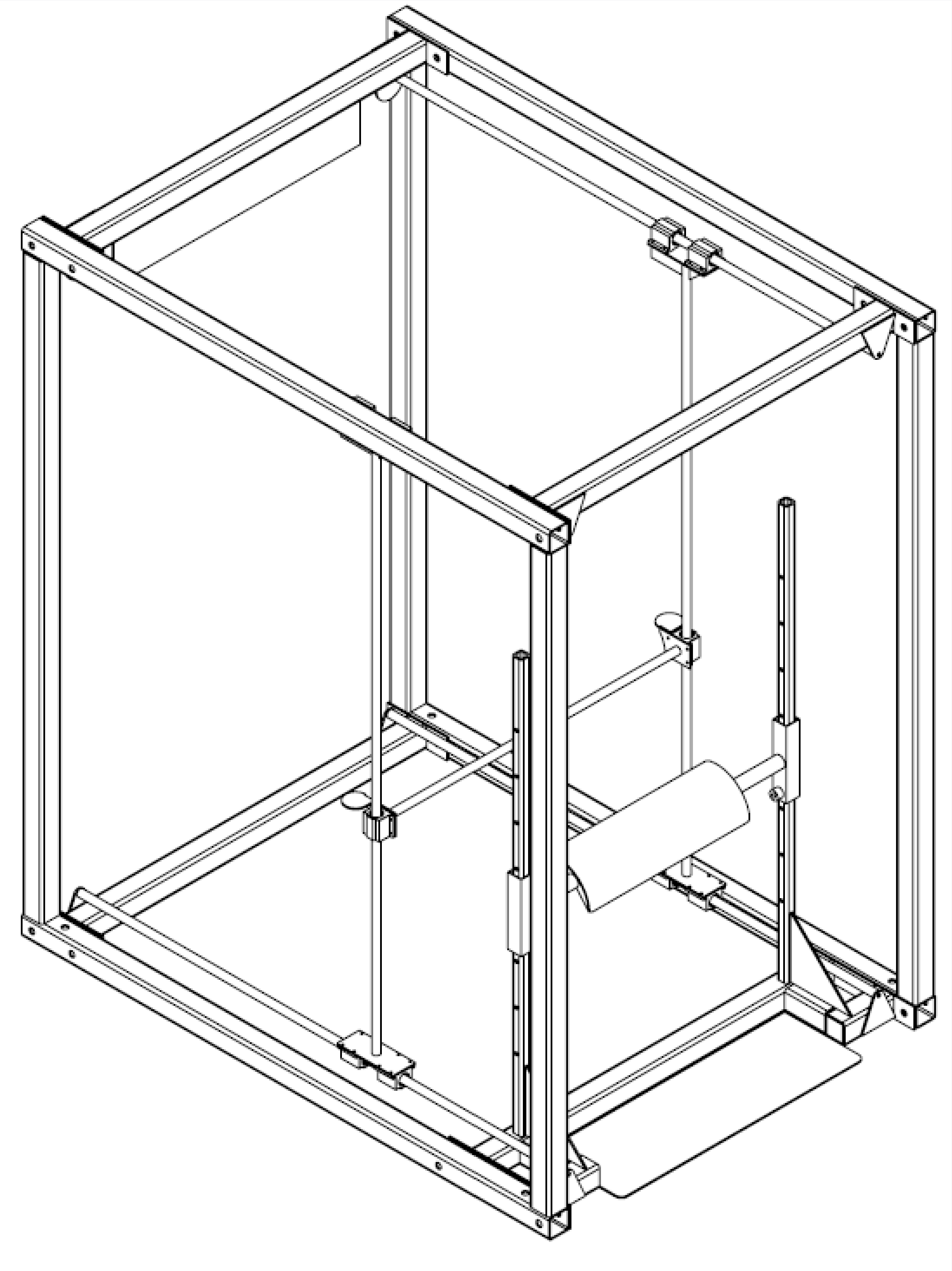
- Desire to automate collection of reach points

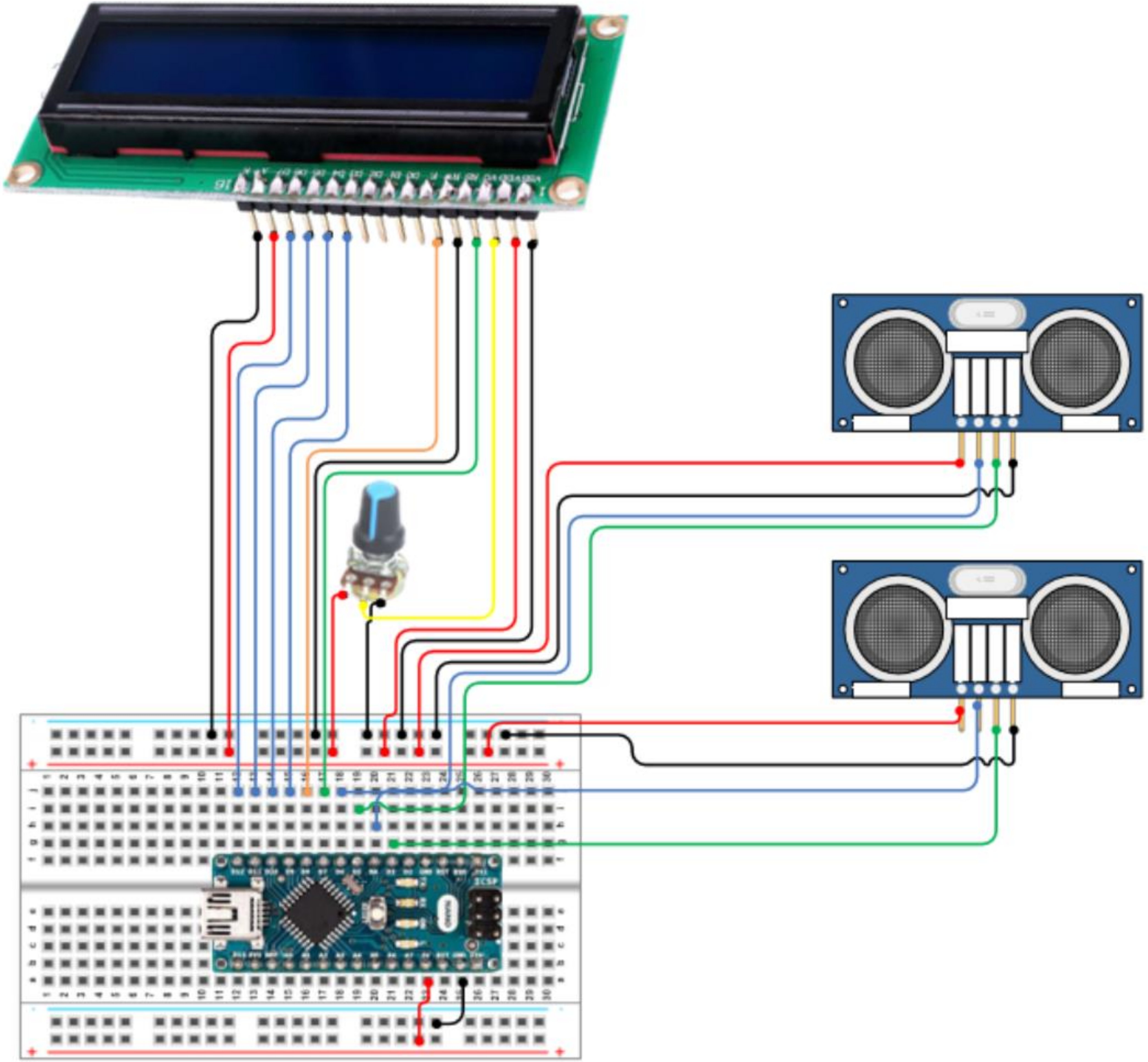
Variable Barrier Reach System Needs

- Track reach capabilities at incremental height differences by plotting the reach paths into X/Y graphical data that can be analyzed accordingly

- Arduino ultrasonic sensors to plot X/Y positions

- Plot data downloaded directly to connected tablet





Variable Barrier Reach System Concerns

- Geographically placed locations
- Limited WiFi
- Large candidate pools

Code Demos

Function Naming

AppName-Entity-Version-AzureRegion[-Environment]

PHF-EmployeeType-V1-USE2-DEV

thank you.