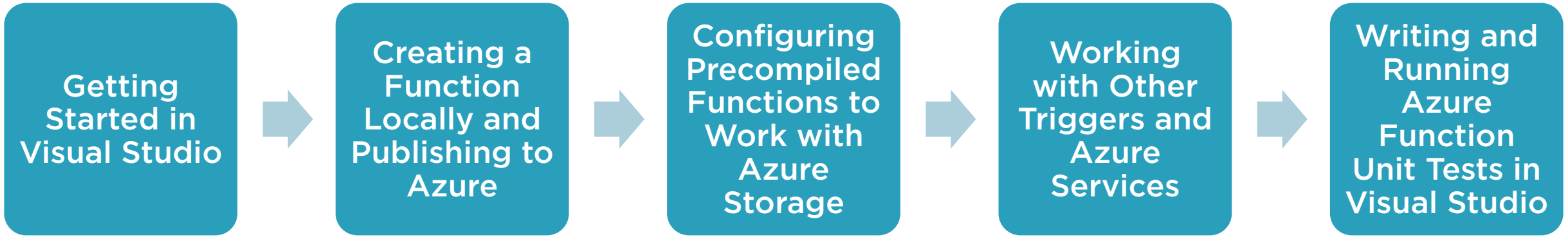
Writing and Testing Precompiled Azure Functions in Visual Studio 2017

GETTING STARTED IN VISUAL STUDIO

Jason Roberts
.NET MVP

@robertsjason dontcodetired.com

Course Overview



Overview

Why Azure Functions?

Core concepts overview

An overview of creating precompiled functions in Visual Studio

An introduction to Azure Function code attributes

Setting up the development environment

Creating an Azure Functions project in Visual Studio

Why Azure Functions?

- Fully managed compute environment
- High reliability
- Security
- Auto-scaling
- Pay-per-use (Consumption plan)
- Less boilerplate code to write/maintain
- Integrate easily with range of services

Benefits of Precompiled Functions

Normal C# classes, not .csx C# script files

Familiar environment of Visual Studio

- IntelliSense
- Debugging, breakpoints, etc.
- NuGet package manager
- Test Explorer
- 3rd party tools (e.g. ReSharper, etc.)

Local Azure Functions runtime environment

Create unit test projects

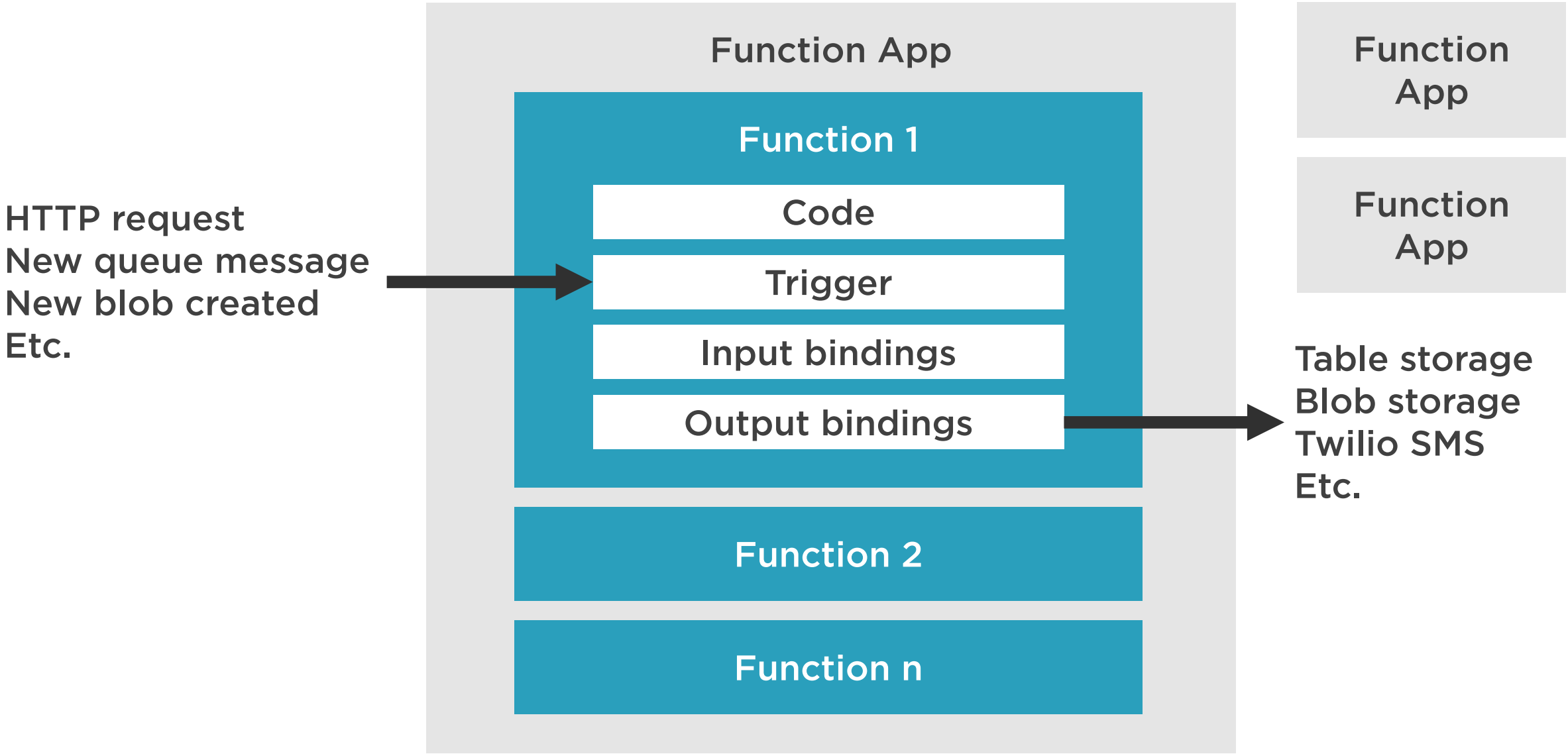
Better cold-start performance

- .csx compiled when first request

“Azure Functions Fundamentals”

by Mark Heath

Core Concepts Overview



An Overview of Creating Precompiled Functions in Visual Studio

Create Azure account

Create new Azure Functions project in Visual Studio

Add one or more functions to the project

- Select trigger type
- Create function C# class file
- Basic trigger information configured
- Trigger and binding attributes

Compile

Run & debug functions locally

Add unit test project

Publish functions to Azure

Azure Functions Core Tools command line interface

github.com/Azure/azure-functions-cli

An Introduction to Azure Function Code Attributes

```
[FunctionName("ValidateApplication")]
```

```
[HttpTrigger] [BlobTrigger] [QueueTrigger] …
```

```
[Queue] [Blob] [Table] [ServiceBus] [EventHub] …
```

```
[FunctionName("MakeApplication")]
public static async Task<HttpResponseMessage> Run(
    [HttpTrigger(AuthorizationLevel.Anonymous, "post", Route = null)]
    HttpRequestMessage req,
    [Blob("submitted-applications/{rand-guid}")]
    TextWriter applicationBlobStorage,
    TraceWriter log)
{
    // function code
}
```

Summary

Why Azure Functions?

- Pay-per-use
- Auto-scaling

Core concepts

- Function code
- Triggers & bindings
- Function apps

An overview of creating precompiled functions in Visual Studio

[FunctionName] [HttpTrigger]

Setting up the development environment

Created an Azure Functions project in Visual Studio

Next:

Creating a Function Locally and Publishing to Azure