Serverless Azure

Azure Functions

Michał Jankowski

about me

Michał Jankowski

architect / software developer / team leader / traveller / photographer

Objectivity
Bespoke Software Specialists

aim.

Learn how we can use Serverless in Azure in our solutions to improve our productiveness.

way
of working

we should have fun

A bit of theory, then a lot of demos and practice.

I would encourage you to work together and exchange your knowledge.

Great things in business are never done by one person. They're done by a team of people.

Steve Jobs, cofounder of Apple

we will
be working with

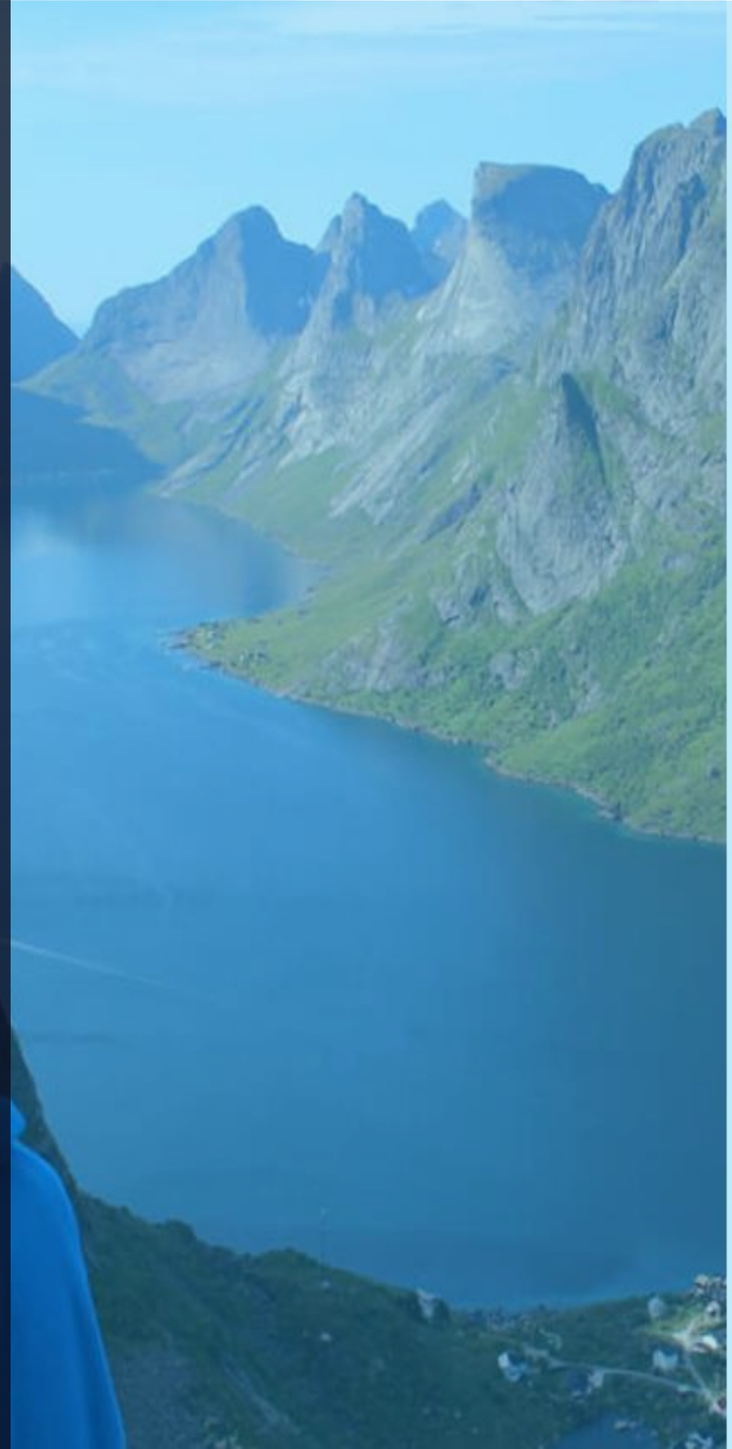
Azure Functions

An event-based serverless compute experience to accelerate your development. Scale based on demand and pay only for the resources you consume.

todo.

We will provide Azure Function backend for registration form that will:

- validate data
- store customer data
- send thank you message
- send SMS to us each time new customer will register



Get The Best Ticket Today!

You can add unlimited fields directly from HTML

Your name

Your surname

Your county

Your e-mail

Your birth year

Send Information

presentation
agenda

theory

Serverless in Azure environment

azure functions

Main part of presentation. You will learn how to develop them correctly and in effective way.

theory.

types of approaches

01 Do I really know how hardware works? What hardware specification should be delivered?

02 Am I a good system administrator? When I should update servers' OS?

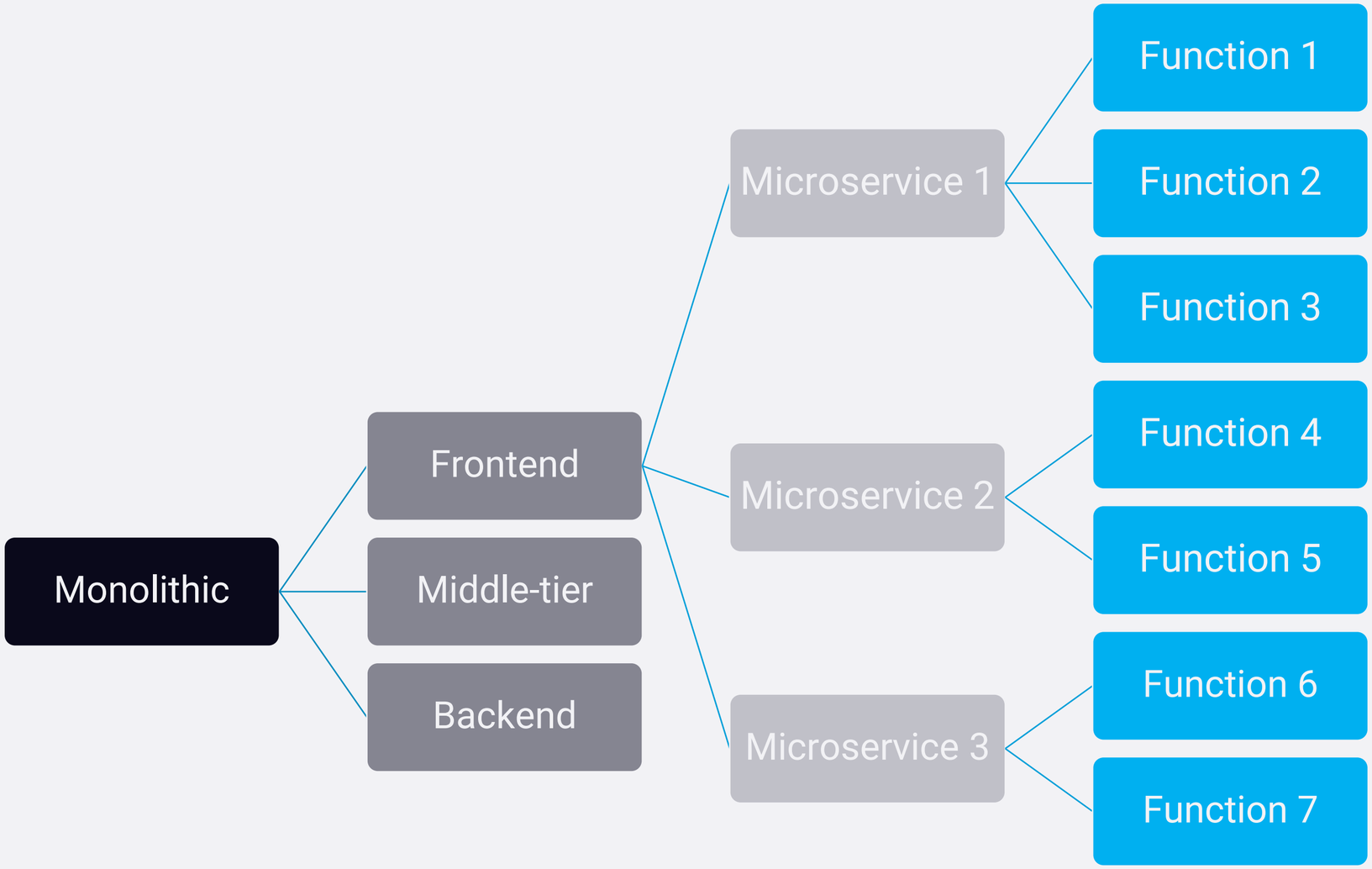
03 Finally I can focus on features development.

On-Premises	IaaS	PaaS	Serverless
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
O/S	O/S	O/S	O/S
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

Managed by us

Managed by vendor

the evolution
of application platforms



serverless characteristics

server abstraction

There is no server managing tasks.

event driven

Function does not work when there is no event triggering it. It can also instantly scale up.

microbilling

Pay only when there are events. But think about DDOS on your wallet.

productivity

Reduce tasks related to infrastructure. You can focus on development activities.

focus on features

And then you are able to focus on business logic of your app.

faster time to market

All items mentioned together allow you to reduce time to market.

serverless
in Azure

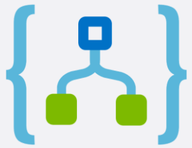
Azure Functions

An event-based serverless compute experience to accelerate your development. Scale based on demand and pay only for the resources you consume.

Event Grid

A single service for managing routing of all events from any source to any destination. Designed for high availability, consistent performance and dynamic scale. Event Grid lets you focus on your app logic rather than infrastructure.

serverless
in Azure



Logic Apps

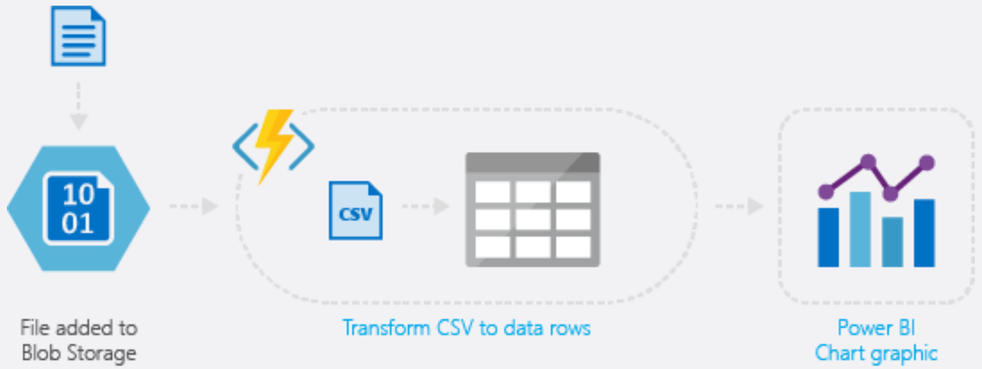
Provide a way to simplify and implement scalable integrations and workflows in the cloud. It provides a visual designer to model and automate your process as a series of steps known as a workflow.

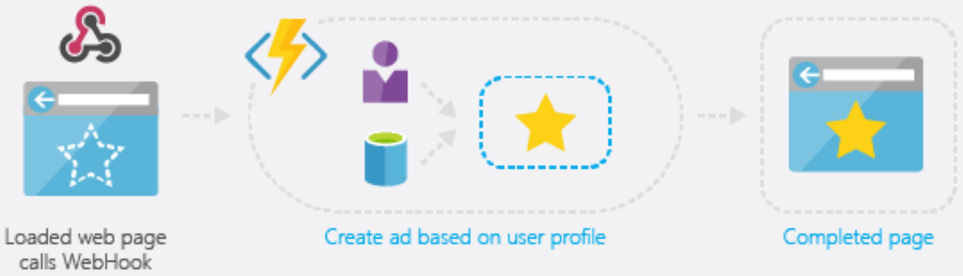
Flow

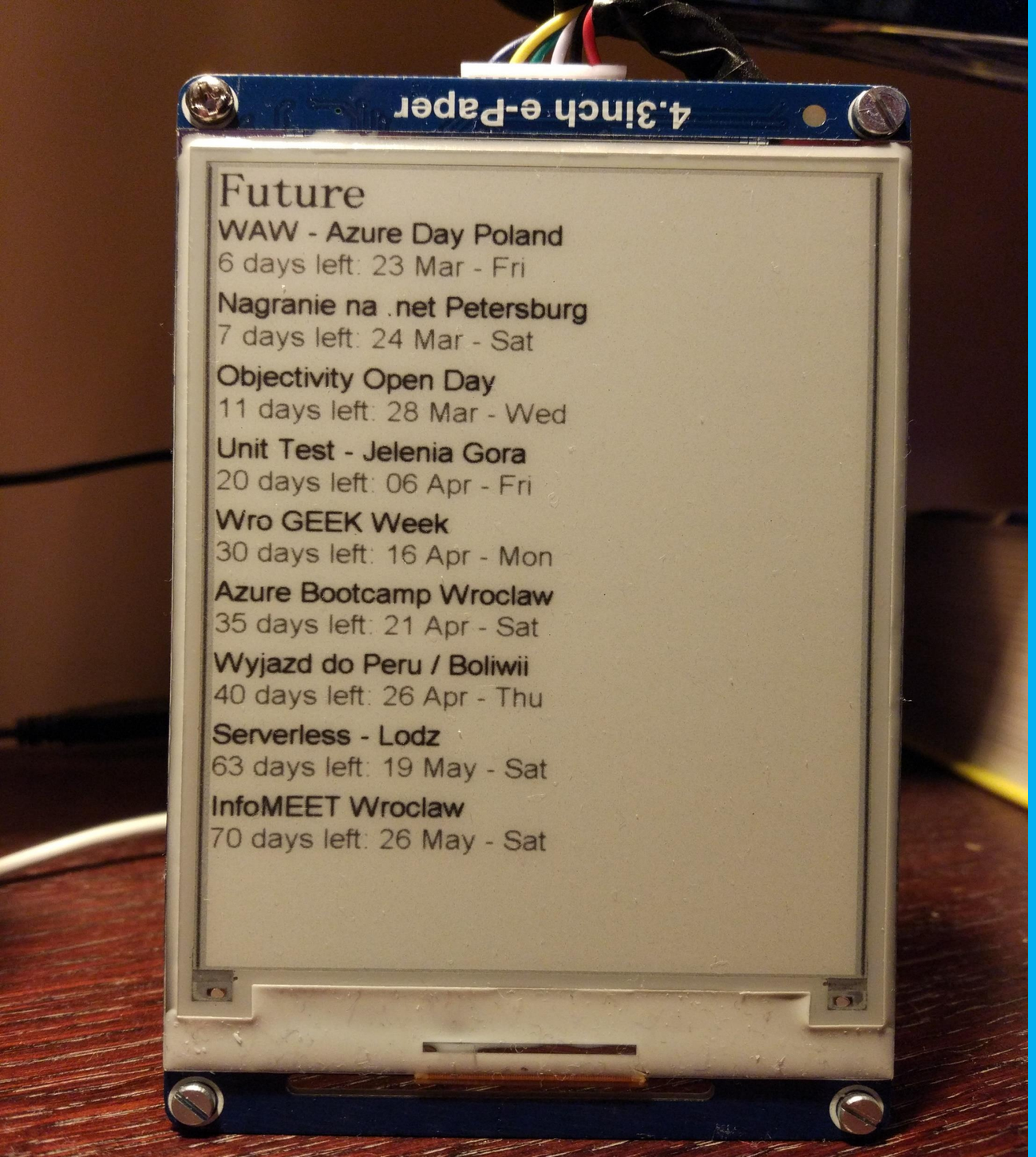
Is a service that allows you to create automated workflows between your favourite applications and services to synchronize files, get notifications, collect data, and more.

azure
functions.

common scenarios







- Azure Function as a glue
- My favourite usage scenario
- Arduino
  → Azure Function
  → ToDoist

key features

choice of languages

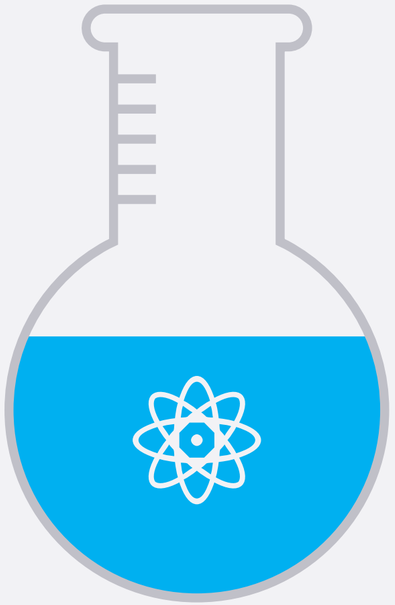
C#, F#, Node.js, Python, PHP, batch, bash, or any executable

pay-per-use pricing model

Consumption plan vs *App Service* plan

bring your own dependencies

Functions supports *NuGet* and *NPM*



open-source

The Functions runtime is open-source and available on GitHub

integrated security

Protect HTTP-triggered functions with OAuth providers such as AAD, Facebook, Google, Twitter, and Microsoft Account

simplified integration

Defined services can trigger your function or can serve as input and output for your code.

flexible development

Set up continuous integration and deploy your code through GitHub, Visual Studio Team Services, and other supported development tools.

time of starting

We will need additional time for our function start. Normal application are always ready for response.

think about state

Functions are stateless. You should save somewhere state if you need.

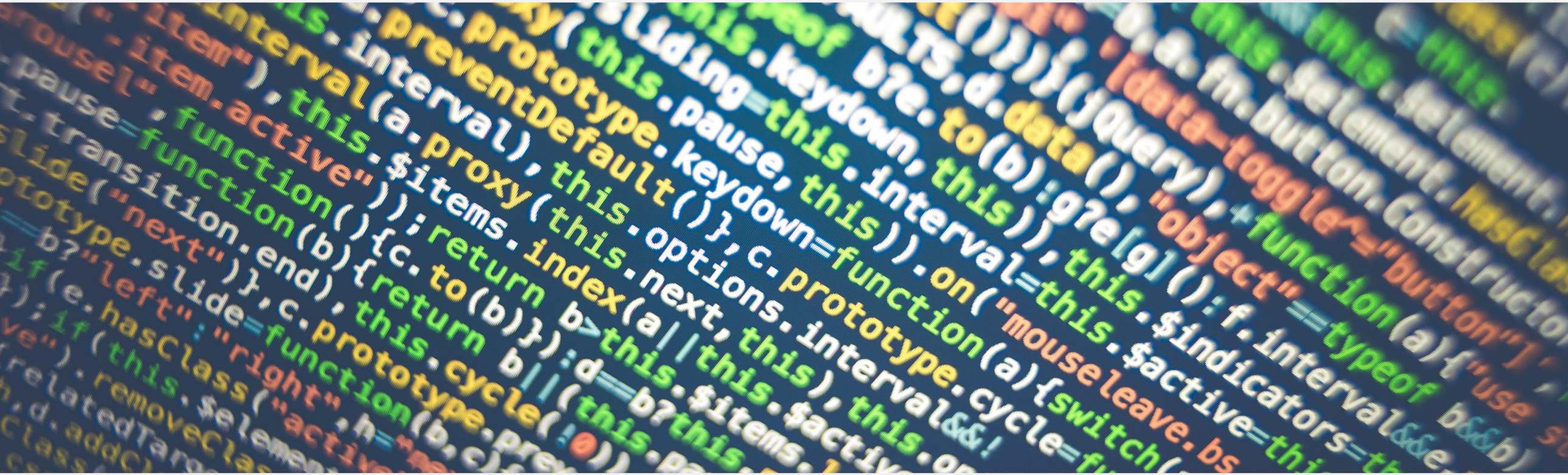
local environment

It is not so easy to start your function locally and it can be run only under Windows.

vendor locking

It will be hard to change your vendor in latter stage of your application life.

maybe
we should start coding



demo

1.

triggers
& bindings.

triggers & bindings

type	1.x	2.x	trigger	input	output
Blob Storage	+	+	+	+	+
Cosmos DB	+	+	+	+	+
Event Grid	+	+	+	-	-
Event Hubs	+	+	+	-	+
HTTP	+	+	+	-	+
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	+	+	+	-	+
SendGrid	+	+	-	-	+
Service Bus	+	+	+	-	+
Table storage	+	+	-	+	+
Timer	+	+	+	-	+
Twilio	+	+	-	-	+
Webhooks	+	-	+	-	+

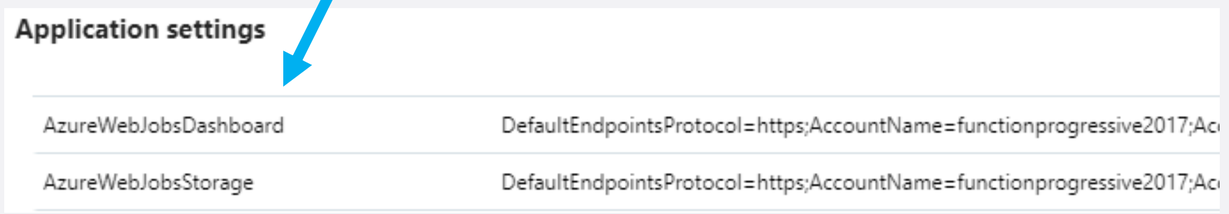
classic approach

run.csx:

```
public static void Run(string myQueueItem, TraceWriter log)
{
  log.Info($"C# Queue trigger function processed: {myQueueItem}");
}
```

function.json:

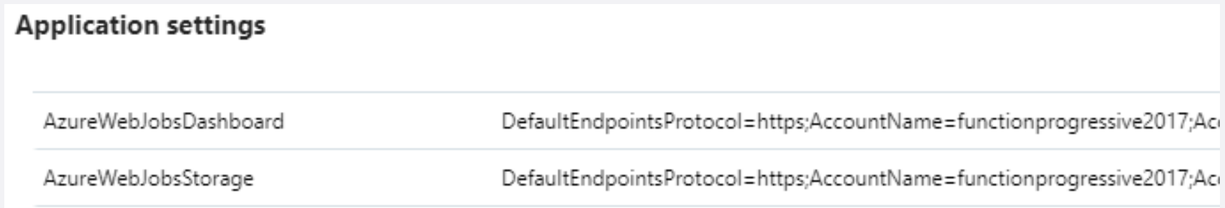
```
{
  "bindings": [
    {
      "name": "myQueueItem",
      "type": "queueTrigger",
      "direction": "in",
      "queueName": "myqueue-items",
      "connection": "AzureWebJobsDashboard"
    }
  ],
  "disabled": false
}
```



01

new approach

```
public static class DemoFunction
  {
    [FunctionName(„DemoFunction")]
    public static void Run(
      [QueueTrigger("myqueue-items", Connection = "AzureWebJobsDashboard")]string myQueueItem,
      TraceWriter log)
      {
        log.Info($"C# Queue trigger function processed: {myQueueItem}");
      }
  }
```



02

tools.

there are some options

VS 2017 v15.6.x

Native support for Azure Functions. Possibility of pre-compilation, deployment and debugging

VS Code

Light code editor. You need to use other tools to be able to do everything that VS 2017 can do.

azure-function-core-tools

CLI that helps you working with Azure Functions locally

cold start.

After some time without any action your function will be turned off / terminated.

Then next request will take longer – function will need to recover environment, reload dependencies and compile it again.

Solution for that is **pre-compilation** of your function.

pre-compilation.

1. We can use full features on Visual Studio, including IntelliSense.
2. We can easily write unit test codes.
3. We can easily attach function codes to existing CI/CD pipelines.
4. We can easily migrate the existing codebase with barely modifying them.
5. We don’t need project.json for NuGet package management.
6. We can reduce the total amount of cold start time by removing compiling on-the-fly when requests hit to the Functions.

demo

2.

do we design it correctly

“Many of the solutions that we consider best practice are solutions for problems that no longer apply”

Gojko Adzic, MindMup / Claudia.js

do we design it **correctly**

01 architecture optimised for reserved resources vs bundling into apps

02 paid for reserved resources vs utilised capacity

03 optimized for quick fail-over vs time to start

04 difficult to replicate "production" vs multiple version of functions

05 layered architecture vs smart composition

06 high vs low cost

Play arbitrage with different charging models

summary.

We have learnt possibilities of Azure Serverless environment

You should know how to develop your solution

You should be aware of new challanges with this apporach

more information

- https://docs.microsoft.com/en-us/azure/azure-functions/
- https://github.com/Azure/azure-functions-core-tools
- https://github.com/Azure/Azure-Functions
- https://github.com/Azure/azure-webjobs-sdk-extensions/

do you have any questions?

www.jankowskimichal.pl

@JankowskiMichal

github.com/MichalJankowskii

thank
you

www.jankowskimichal.pl

@JankowskiMichal

github.com/MichalJankowskii

it’s time for YOU

is your
environment ready

applications needed

- Azure account
- Visual Studio Code
- Visual Studio 2017 version 15.6.x with Azure Functions Tools for Visual Studio extension installed
- azure-functions-core-tools
- Azure Storage Explorer
- Postman

links to installers

https://goo.gl/Fq932B

this is time
for you

1. Create your first function in Azure Portal (*):
    a. Check compilation and errors
    b. Execute your function
2. Create simple calculator API that will support addition, subtraction, multiplication and division (*)
3. Create function that will execute every 30 seconds and log time
    a. Add possibility to change time format without changing the function code
4. Change calculator URLs to more readable form by using Azure Functions Proxies
5. Create Swagger documentation for calculator API

https://goo.gl/rZx11C

(*) mandatory

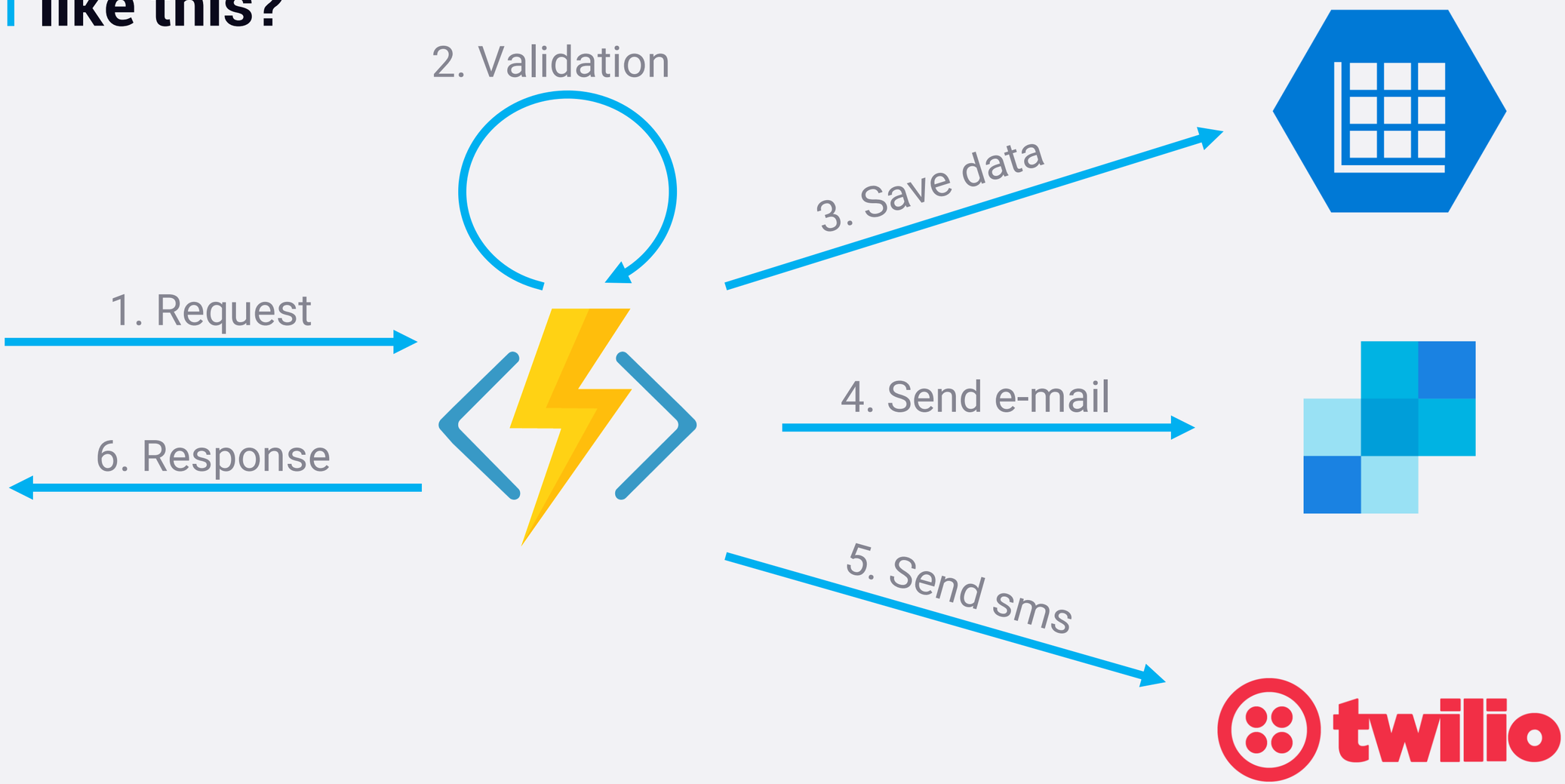
this is time for you

We will provide Azure Function backend for registration form that will:

1. Validate data
2. Store customer data
3. Send thank you message
4. Send SMS to you each time new customer will register

https://goo.gl/q2gChq

is your design like this?

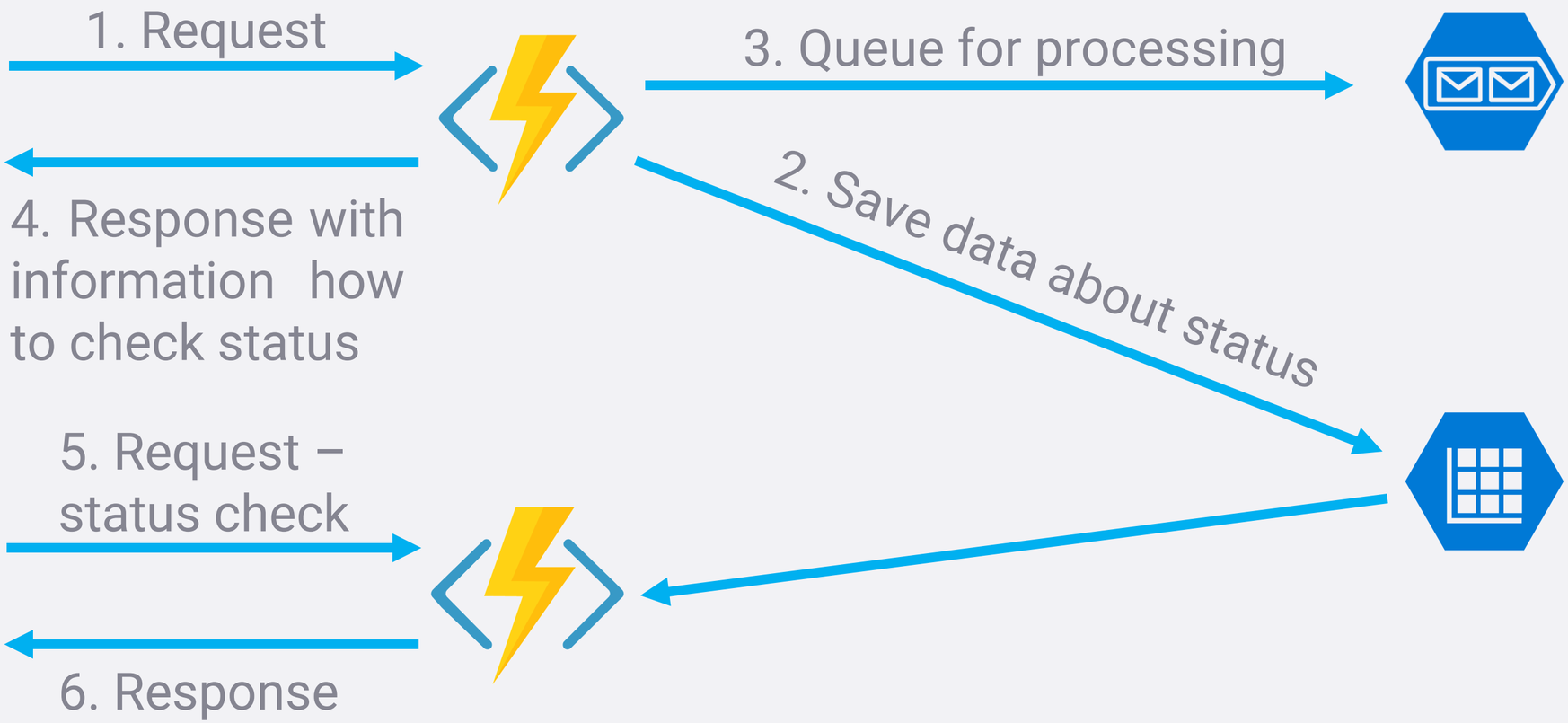


demo

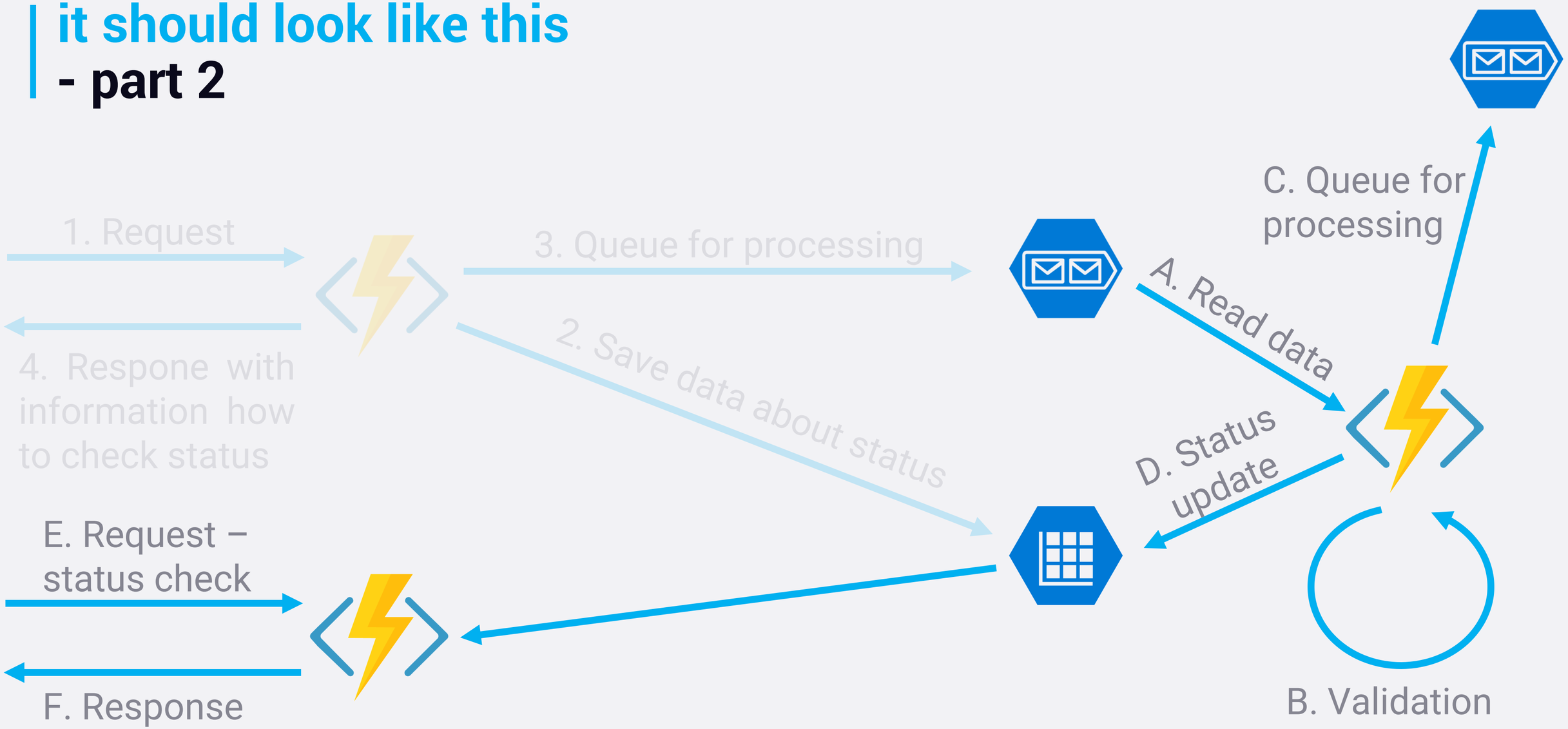
3.

good
practices.

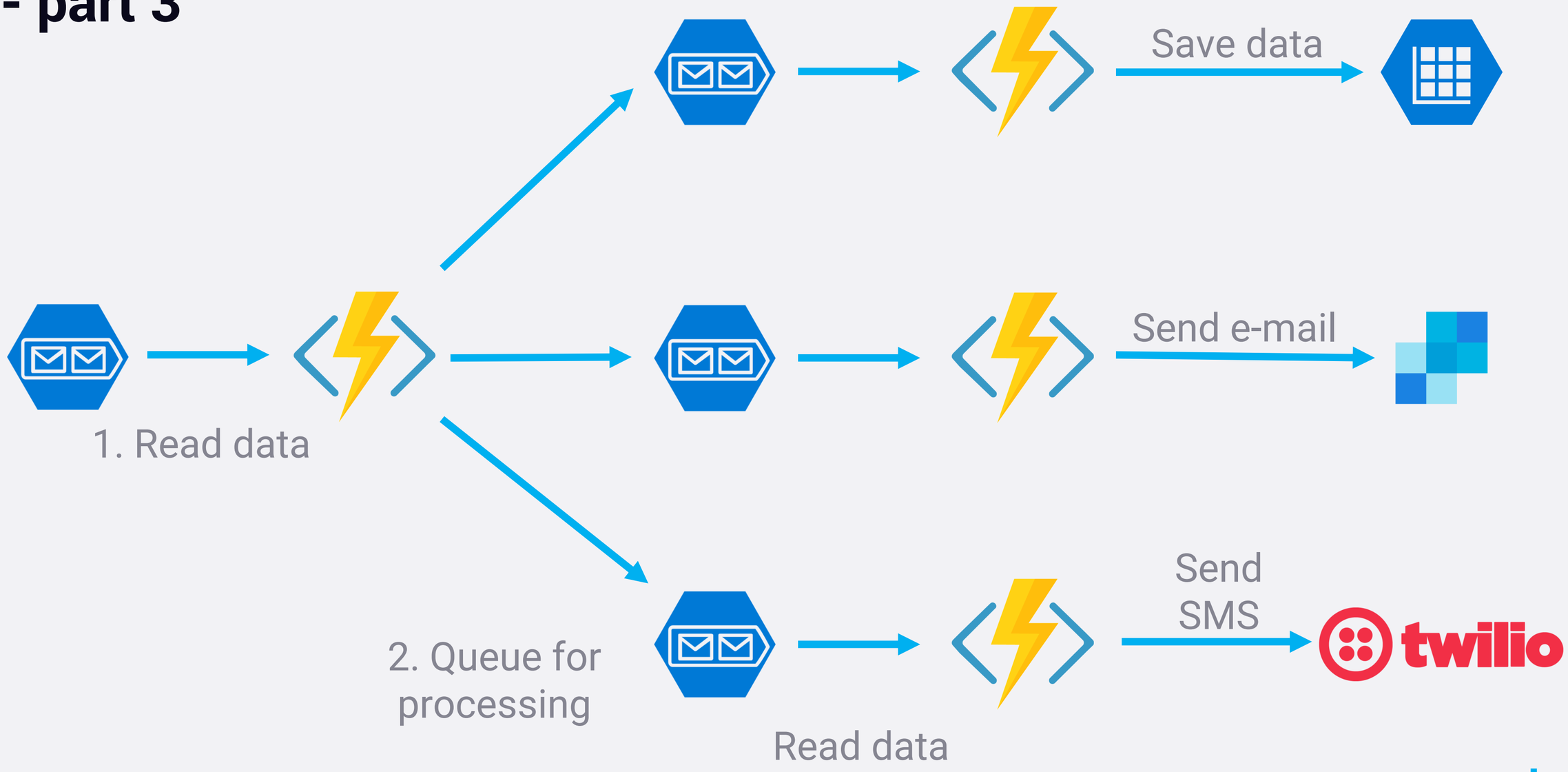
it should look like this
- part 1



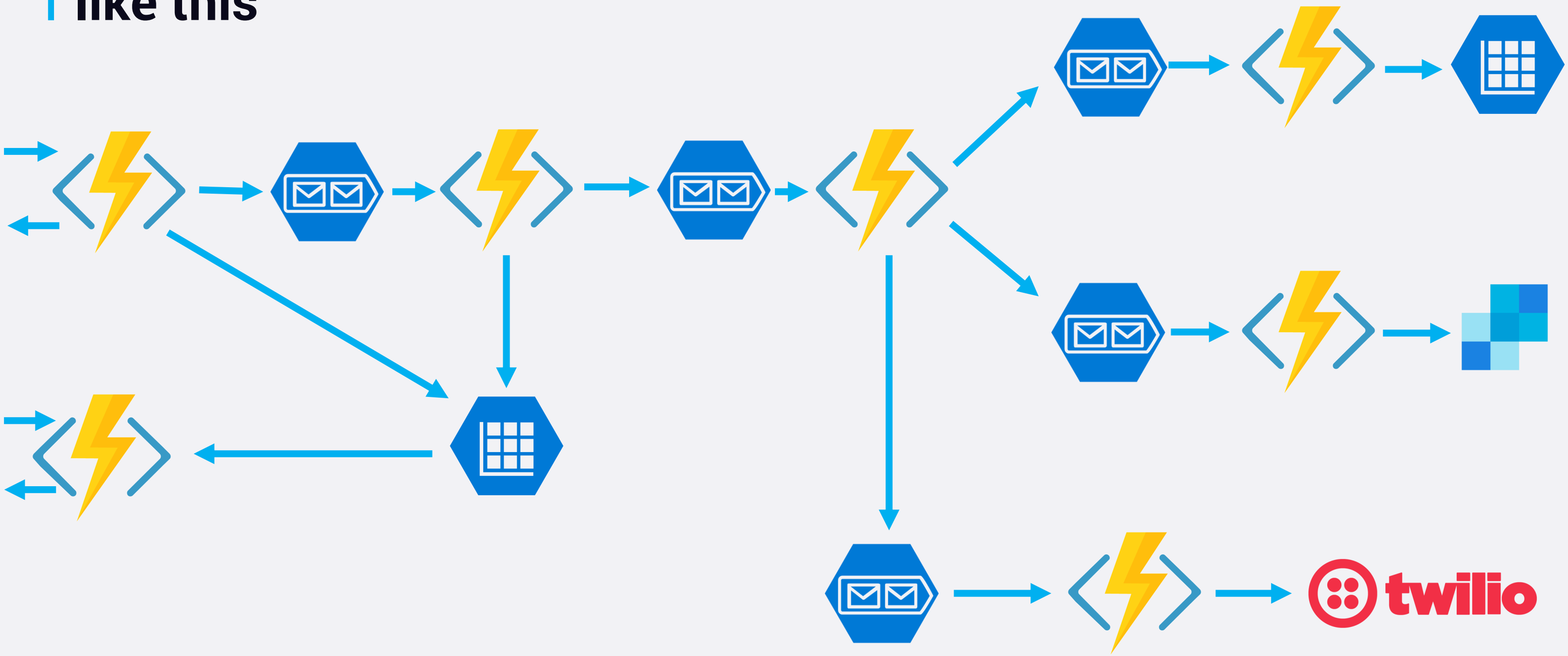
it should look like this - part 2



it should look like this
- part 3

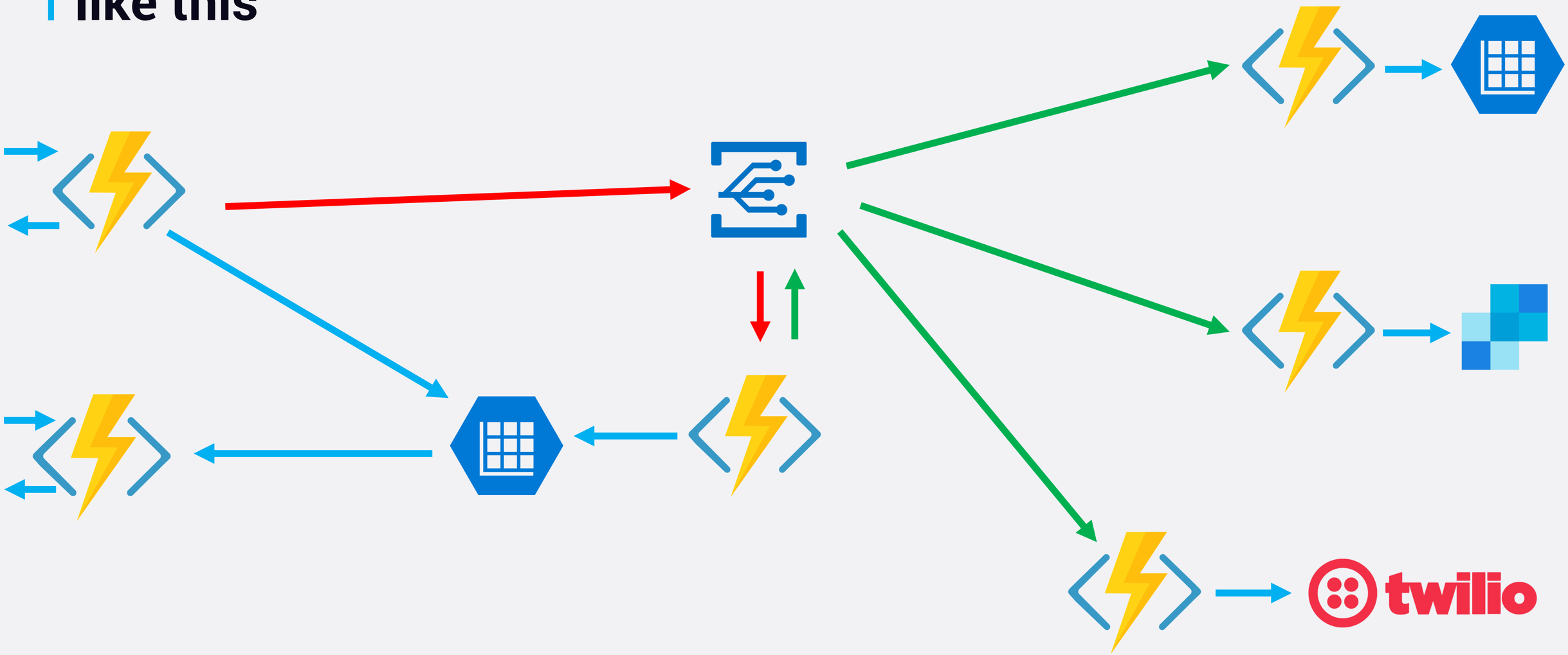


it should look like this



demo
4.

it should look
like this



do you have any questions?

www.jankowskimichal.pl

@JankowskiMichal

github.com/MichalJankowskii

thank
you

www.jankowskimichal.pl

@JankowskiMichal

github.com/MichalJankowskii