

Serverless Mobile Backend as a Service

Marek Sadowski | Developer Advocate | IBM
@blumareks

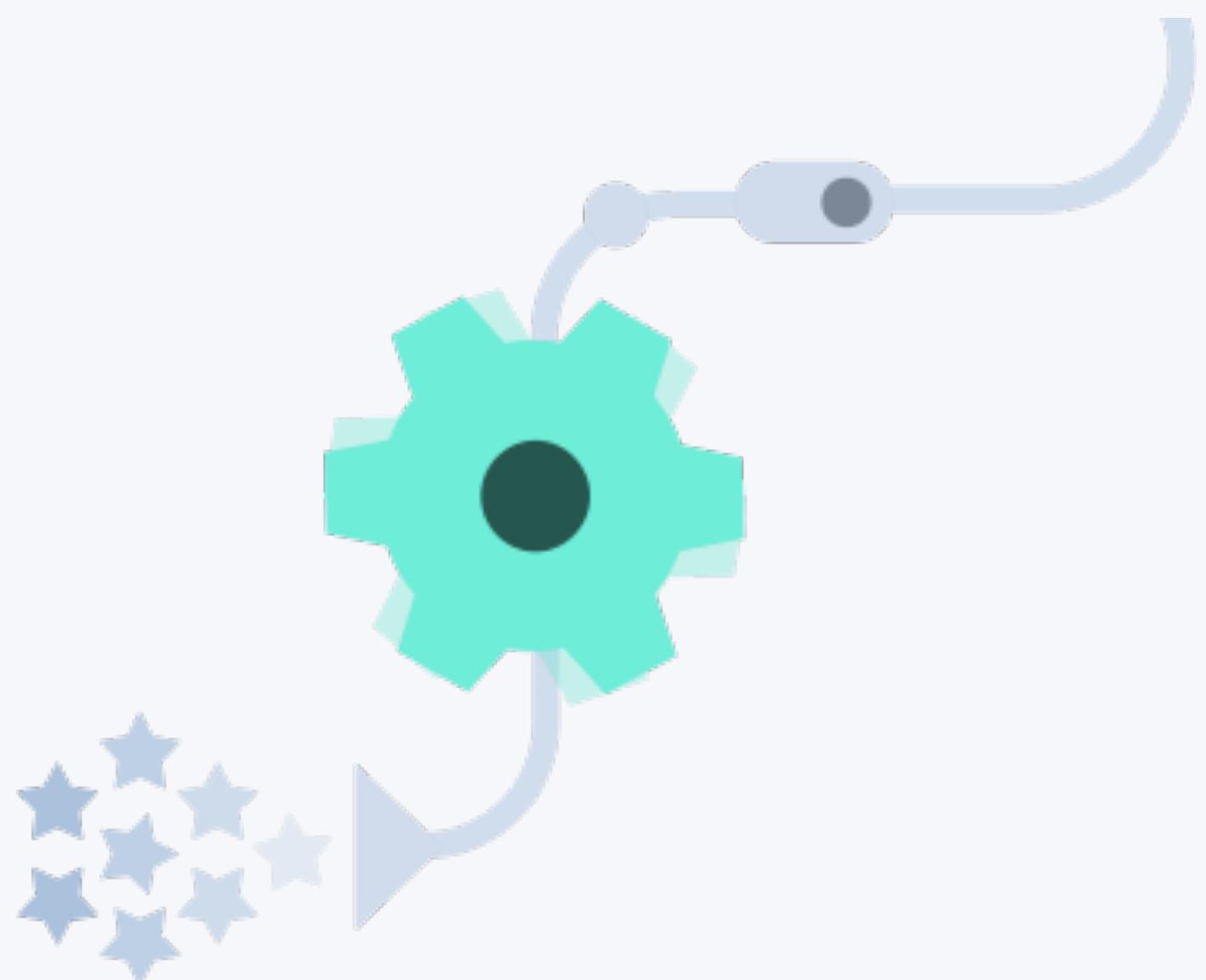
IBM Developer

The Crowdcast webinar

<https://ibm.biz/BdqDWQ>

Agenda

1. What is Serverless Computing ?
2. IBM Cloud Functions
3. Let's code Hello World in IBM Code Functions



Backend as a Service progression

IaaS	CaaS	PaaS	FaaS/Serverless
Functions	Functions	Functions	Functions
Applications	Applications	Applications	Applications
Runtimes	Runtimes	Runtimes	Runtimes
Operating Systems	Operating Systems	Operating Systems	Operating Systems
Infrastructure	Infrastructure	Infrastructure	Infrastructure

Provided by
Developer

Virtualization
Provided by Cloud

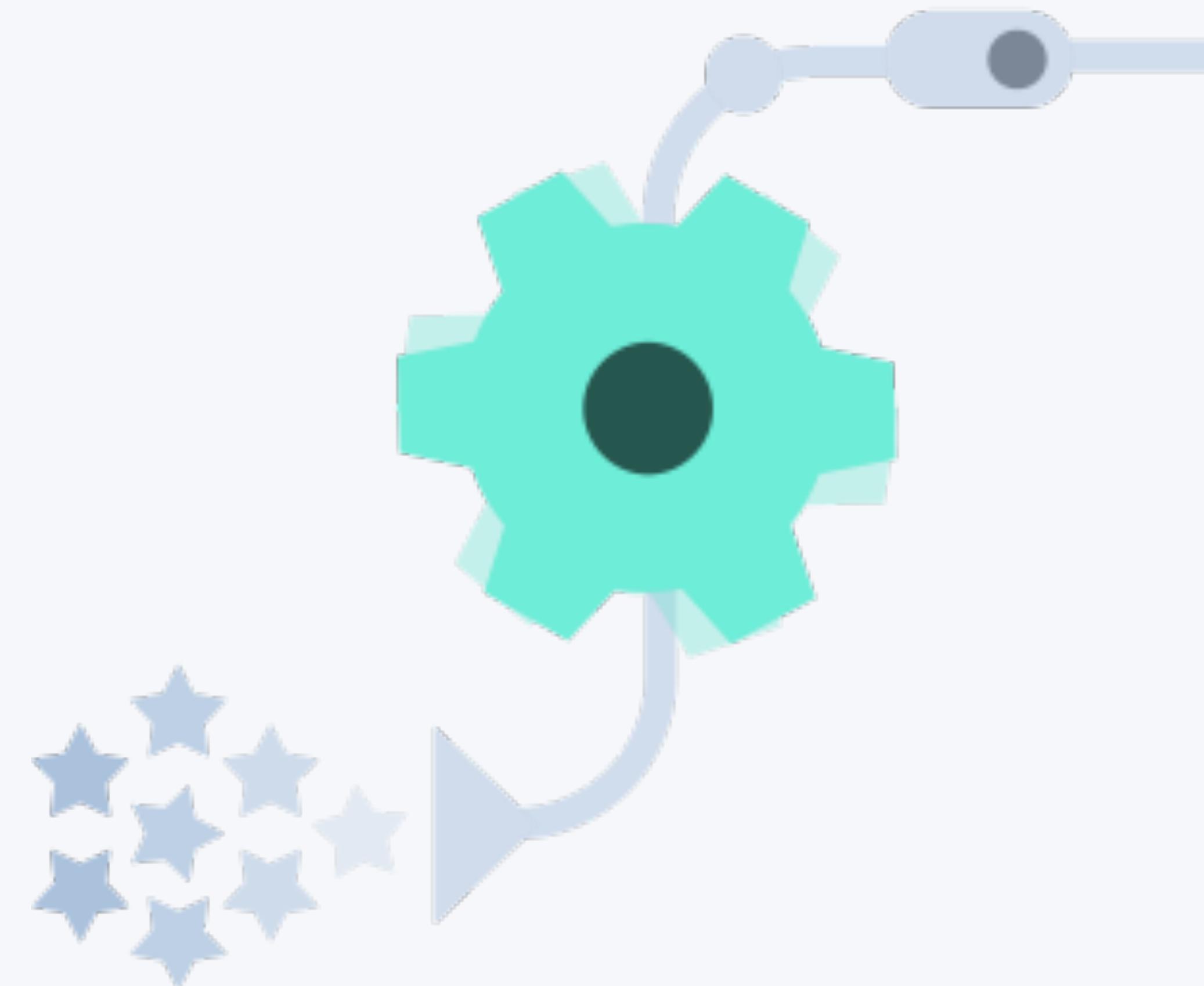
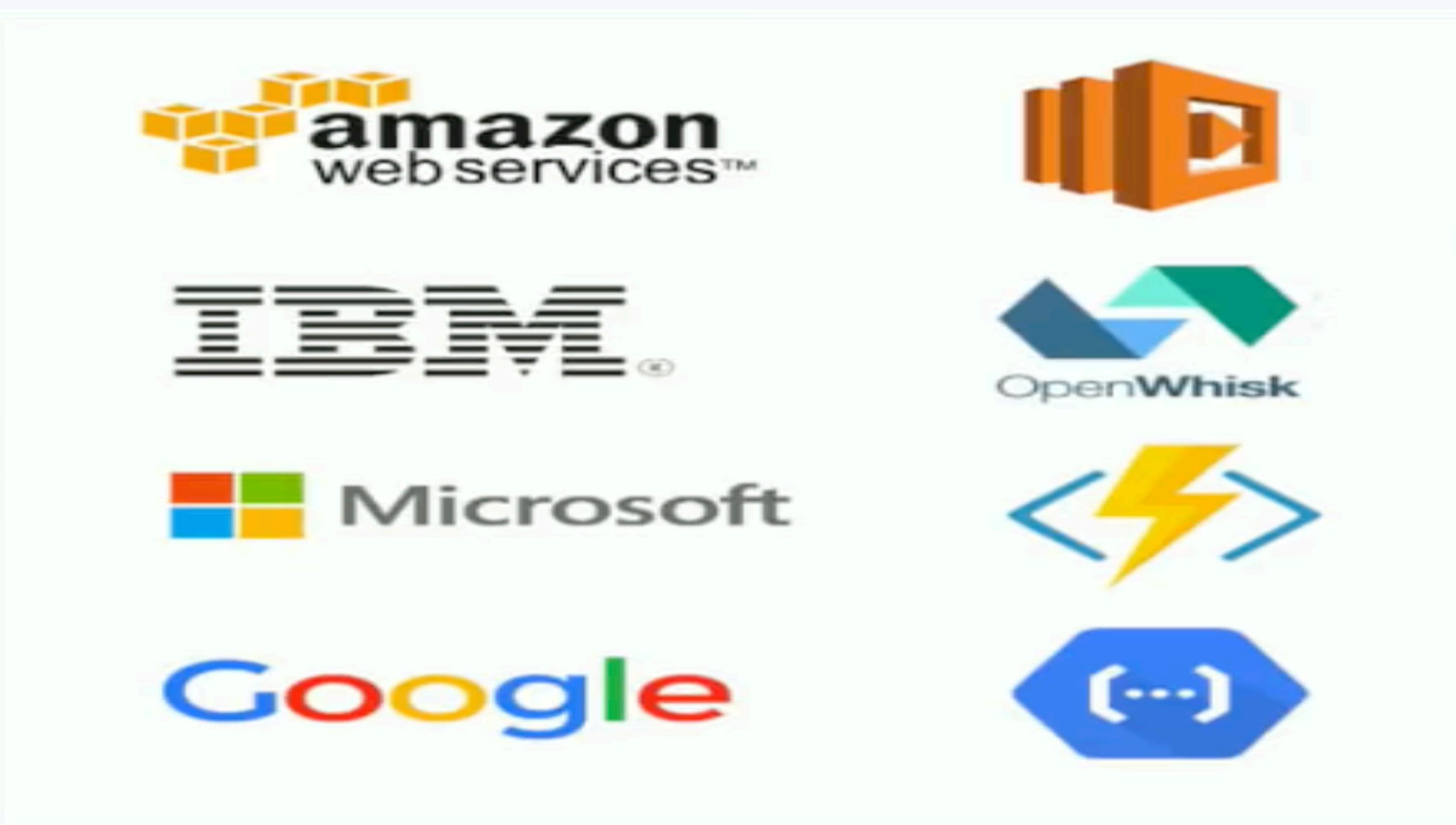
What is Serverless Computing?

- **Requires no management and operation of infrastructure**, enabling developers to focus more narrowly on code/custom business logic.
- **Runs code only on-demand on a per-request basis, scaling transparently with the number of requests being served.**
- **Enables end users to pay only for resources being used, never paying for idle capacity.**

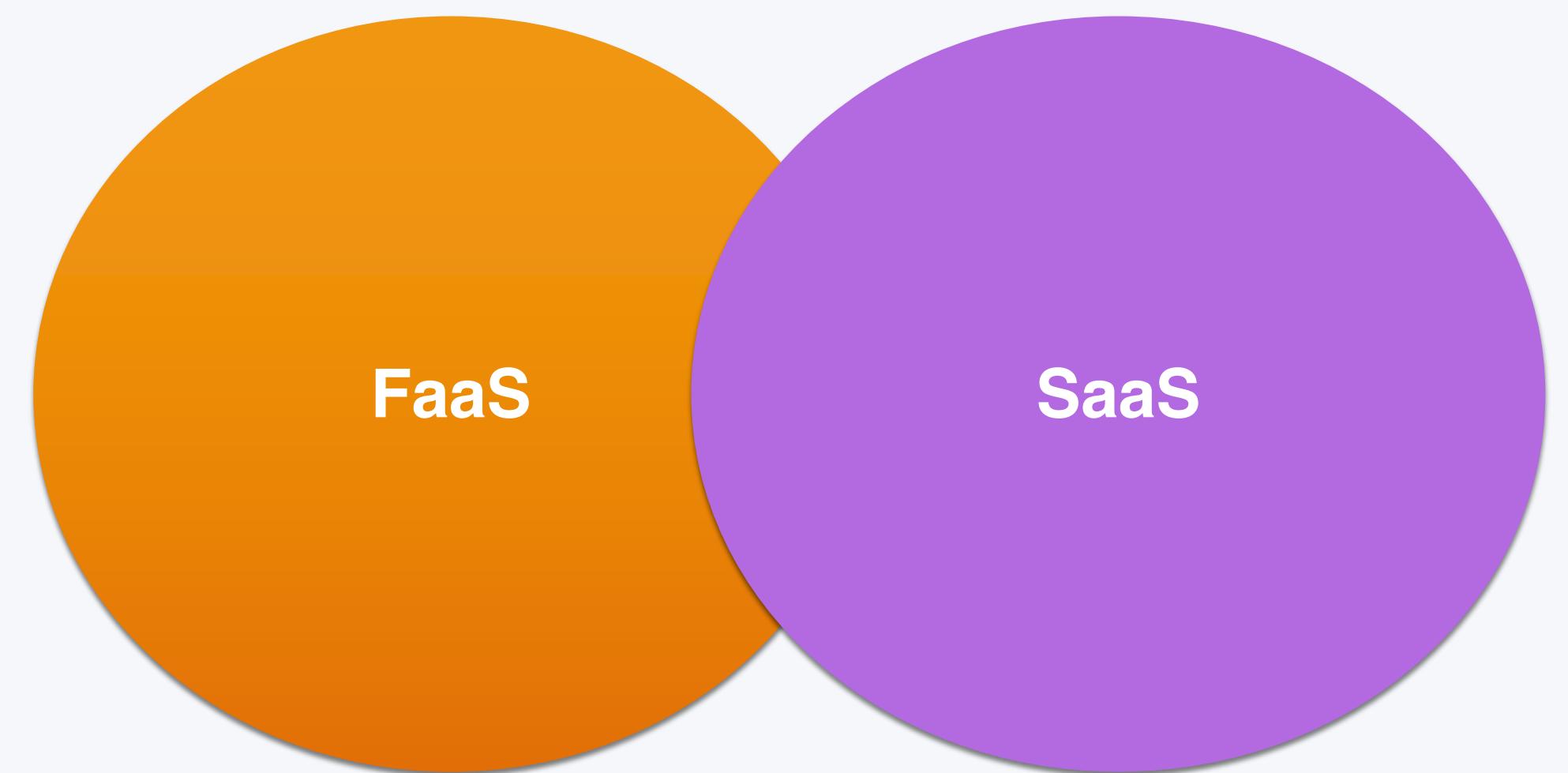
When did Serverless Computing start?

- **AWS Lambda** is an event-driven, serverless computing platform provided by Amazon as a part of Amazon Web Services.
- It is a computing service that runs code in response to events and automatically manages the computing resources required by that code.
- **It was introduced in November 2014**

Serverless computer providers



Serverless and SaaS

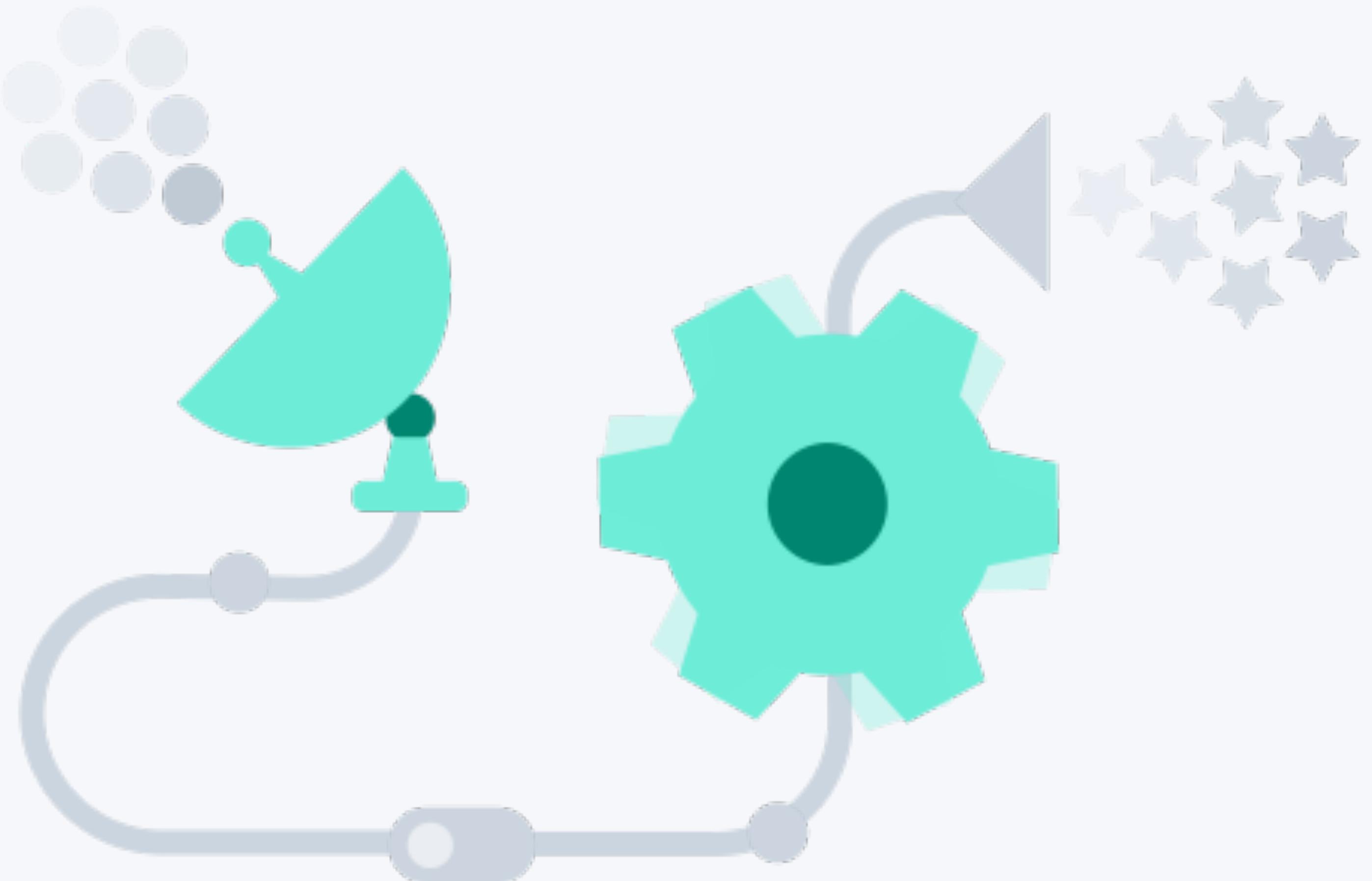


DIY Serverless

- Apache OpenWhisk (IBM Cloud Functions)
- Kubeless
- Fn (Oracle Serverless)
- ...

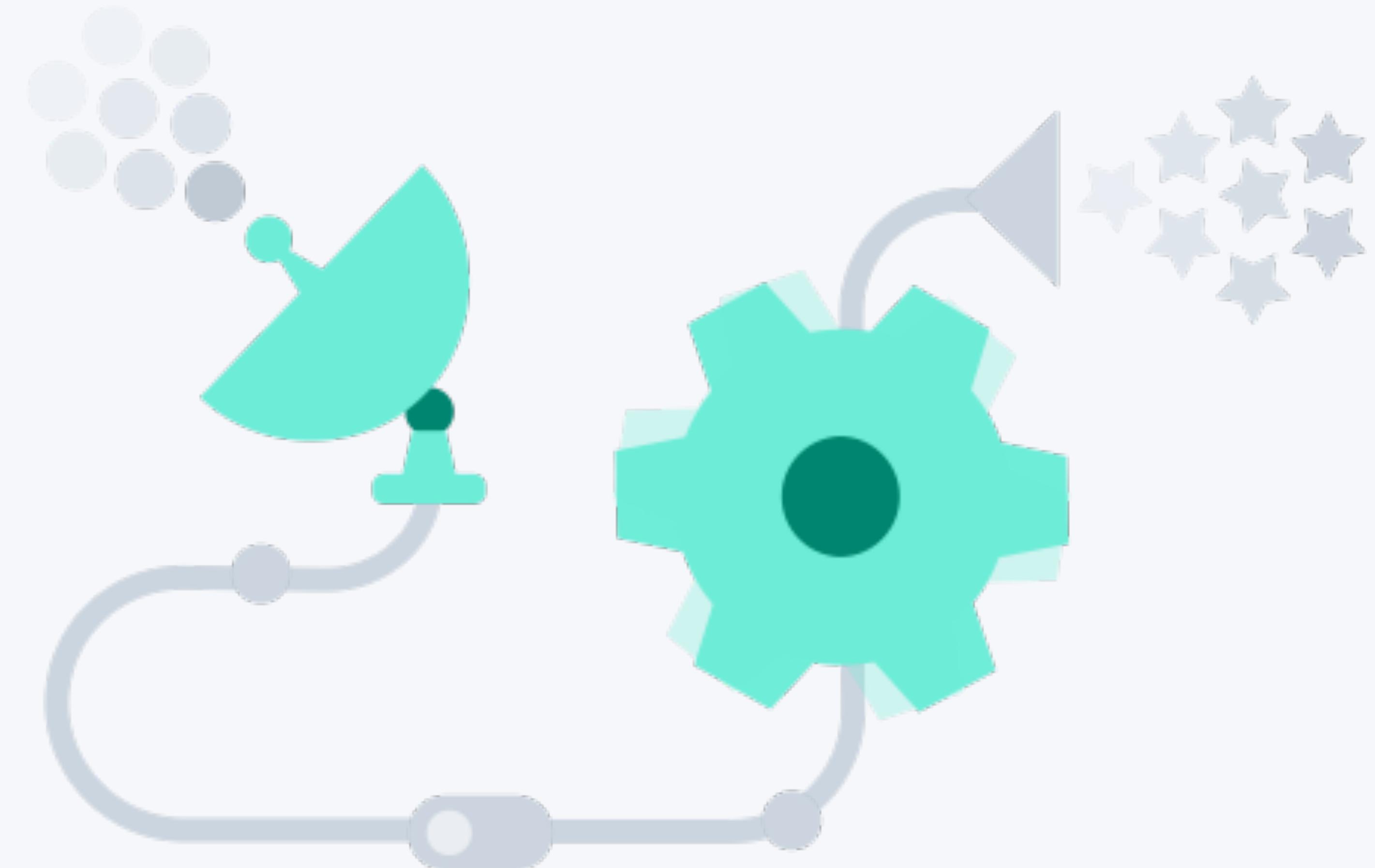
Serverless platform to execute
code in response to events

Developed as
open source software
via Apache Foundation
openwhisk.org



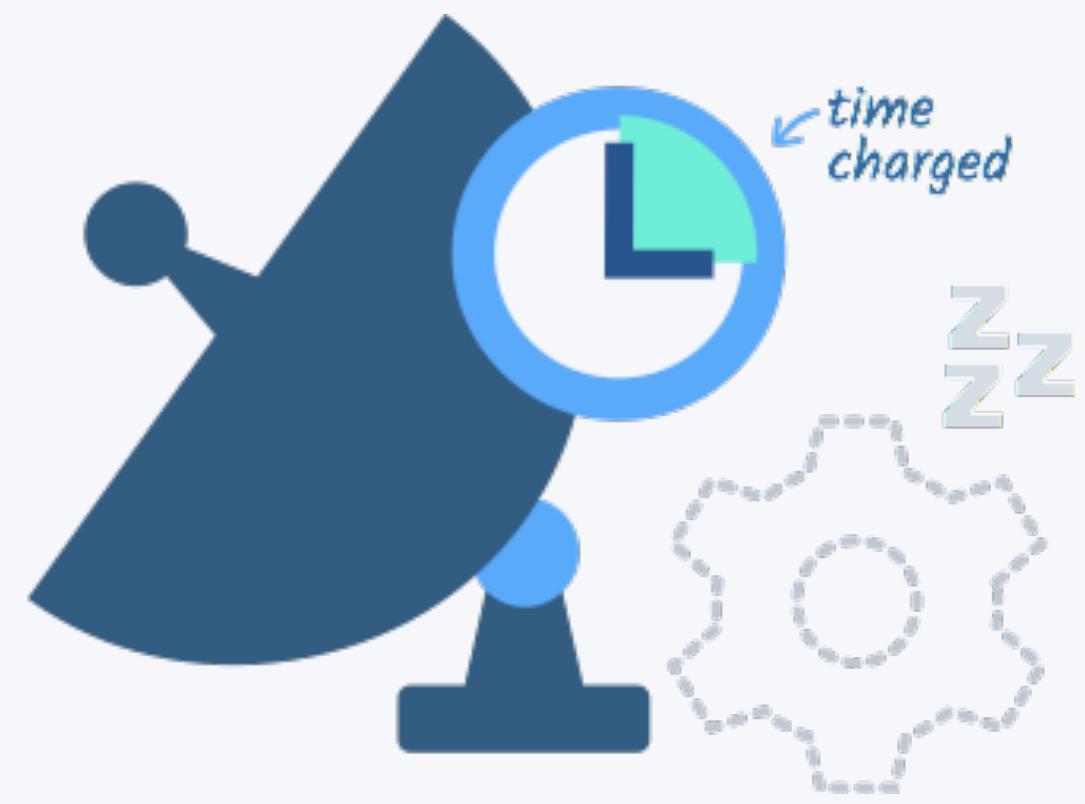
Serverless platform to execute
code in response to events

Managed service on
IBM Cloud
cloud.ibm.com/functions



Granular pricing

Pay only for the exact time your actions run. When an action is not invoked, it's not in memory, so you don't pay anything.



Pricing model

Time an action was running *
memory allocated to action

\$0.000017 per GBs
Free tier: 400,000 GBs



<https://cloud.ibm.com/functions/learn/pricing>

Compare price per request/GBs

Production pricing			
	Requests	GB seconds	Hours / Month
Non-Edge offerings			
Netlify	\$0.000038	NA	1h per 1000 requests
Zeit	NA	NA	\$0.2 / h
IBM OpenWhisk	NA	\$0.000017	NA
Oracle Functions	\$0.0000002	\$0.00001417	NA
Alibaba Functions	\$0.0000002	\$0.00001668	NA
Google Functions	\$0.0000004	\$0.00000025	NA
AWS Lambda	\$0.0000002	\$0.00001667	NA
Azure Functions	\$0.0000002	\$0.000016	NA

<http://serverlesscalc.com/>

IBM Cloud Functions

<https://cloud.ibm.com/functions/>

IBM Cloud Functions

Functions-as-a-Service (FaaS) platform based on Apache OpenWhisk

Run your application code without servers, scale it automatically, and pay nothing when it's not in use.

[Start Creating](#) [Download CLI](#)

Developer Advocacy_Cloud Developer ...
Dallas (CF-Based) [i](#)

The graphic features a central circular icon containing a stylized green 'f' character. To the left of the circle is a blue speech bubble icon with three light gray circles above it. To the right is a teal speech bubble icon with a white '</>' symbol inside. A thin gray line connects the top of the 'f' in the center to the top of the '</>' symbol. The entire graphic is set against a light gray background.

Let's write Hello World!

Create



Quickstart Templates

Get started quickly using one of the Templates. A number of use cases are available, from a hello world action to invoking functions from Cloudant or Event Streams.



Create Action

Actions contain your function code and are invoked by events or REST API calls.



Create Sequence

Sequences invoke Actions in a linear order, passing parameters from one to the next.



Create Trigger

Triggers receive events from outside IBM Cloud Functions and invoke all connected Actions.



Install Packages

Installing Packages installs reusable Actions into your namespace.

1. Action
2. Sequence
3. Trigger
4. Package

Let's write Hello World!

Deploy Template

Template

Get started quickly using one of the Templates. A number of use cases are available, from a hello world action to invoking functions from Cloudant or Event Streams.

[Learn more about Actions](#)

[Learn more about Templates](#)

Choose Template



Hello World
This action will accept a single parameter, which must be a JSON object.

hello world



Cloudant Events
When a Cloudant DB has a document edited or added, log the change in the console.

cloudant



Event Streams
When an Event Streams topic has new data added, log the change in the console.

event streams messagehub



Get HTTP Resource
A web action that is invoked in response to a HTTP event.

web action request



Periodic Slack Reminder
An action that will post to slack based on a periodic trigger.

periodic slack

Let's write Hello World!

Deploy Template

Hello World

This action will accept a single parameter, which must be a JSON object.

This Template creates:
Action helloworld

Add to a Package

Create a Package to place your Template Actions in. Packages act like folders, and contain only Actions and/or Sequences.

Package Name

hello-world

A package by this name already exists.

Action helloworld

Actions are stateless code snippets that run in response to events. They can be invoked manually, via a Trigger, or API calls.

Choose runtime for the actions you will own.

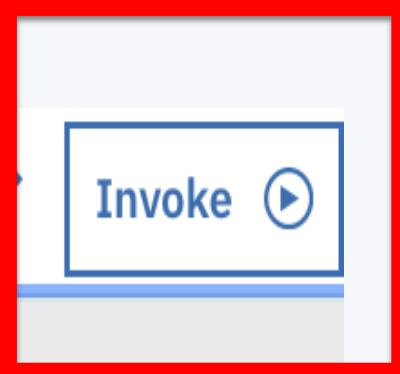
Node.js 10

```
1  /**
2   * main() will be invoked when you Run This Action.
3   *
4   * @param Cloud Functions actions accept a single parameter,
5   *      which must be a JSON object.
6   *
7   * In this case, the params variable will look like:
8   *      { "name": "xxxx" }
9   *
10  * @return which must be a JSON object.
11  *      It will be the output of this action.
12  *
13  */
14
15 function main(params) {
16 if (params.name) {
17   return { greeting: `Hello ${params.name}` };
18 }
19 return { greeting: 'Hello stranger!' };
20 }
21
22 exports.main = main;
23
```

Let's write Hello World!

Code [Node.js 10](#)

View mode - press **Enter** to edit

Change Input [Change](#) [Invoke](#) 

```
1  /**
2  *
3  * main() will be invoked when you Run This Action.
4  *
5  * @param Cloud Functions actions accept a single parameter,
6  *       which must be a JSON object.
7  *
8  * In this case, the params variable will look like:
9  *   { "name": "xxxx" }
10 *
11 * @return which must be a JSON object.
12 *       It will be the output of this action.
13 *
14 */
15 function main(params) {
16   if (params.name) {
17     return { greeting: `Hello ${params.name}` };
18   }
19   return { greeting: 'Hello stranger!' };
20 }
21
22 exports.main = main;
23
```

Let's write Hello World!

Activations	Collapse ↕	Clear 🗑
▼ ✓ helloworld	81 ms	1/12/2020, 08:16:31
Activation ID: cfe5eb9c8d0f4189a5eb9c8d0f018990		
Results: <div style="border: 2px solid red; padding: 10px;"><pre>{ "greeting": "Hello stranger!" }</pre></div>		
Logs: []		

Hands-on

•

Let's sign up to the IBM Cloud

ibm.biz/BdqDWQ

Let's invoke Functions in the IBM Cloud

<https://cloud.ibm.com/functions/>

Pros of Serverless Computing?

- It enables developers to focus on code, not infrastructure.
- Pricing is done on a per-request basis, allowing users to pay only for what they use.
- For certain workloads, such as ones that require parallel processing, serverless can be both faster and more cost-effective than other forms of compute
- Serverless application development platforms provide almost total visibility into system and user times and can aggregate the information systematically

<https://www.ibm.com/cloud/learn/serverless>

Cons of Serverless Computing?

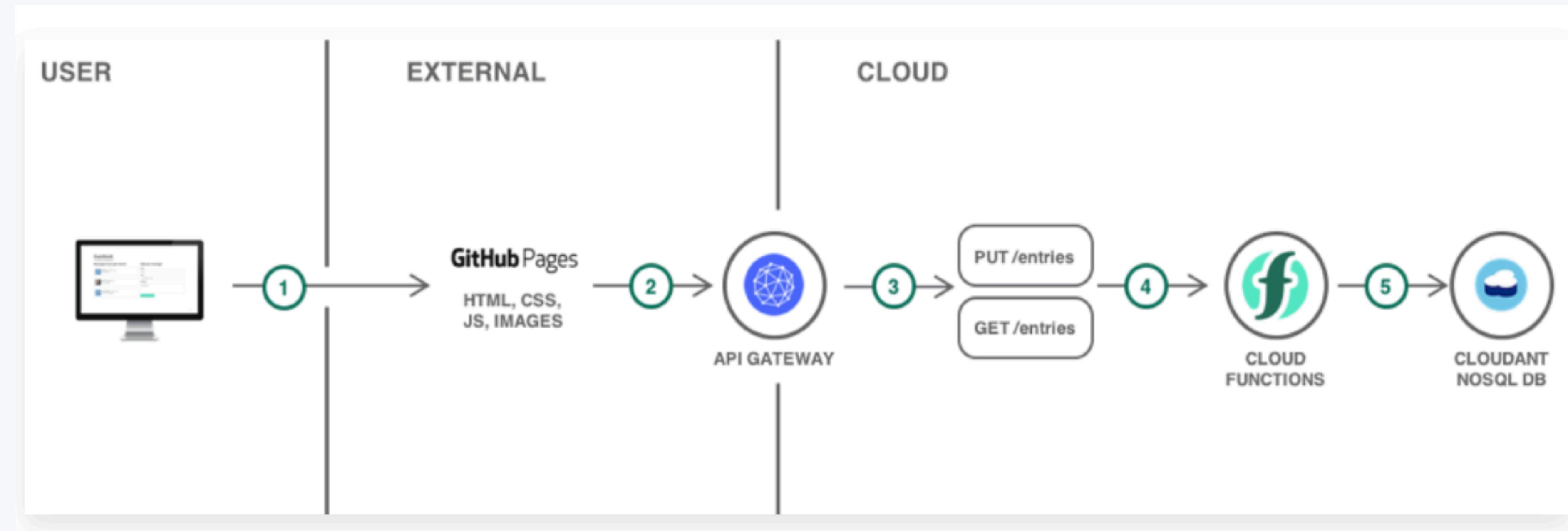
- **Long-running processes:** FaaS and serverless workloads are designed to scale up and down perfectly in response to workload, offering significant cost savings for spiky workloads. But for workloads characterized by long-running processes, these same cost advantages are no longer present and managing a traditional server environment might be simpler and more cost-effective.
- **Vendor lock-in:** Serverless architectures are designed to take advantage of an ecosystem of managed cloud services and, in terms of architectural models, go the furthest to decouple a workload from something more portable, like a VM or a container. For some companies, deeply integrating with the native managed services of cloud providers is where much of the value of cloud can be found; for other organizations, these patterns represent material lock-in risks that need to be mitigated.
- <https://www.ibm.com/cloud/learn/serverless>

Cons of Serverless Computing?

- **Cold starts:** Because serverless architectures forgo long-running processes in favor of scaling up and down to zero, they also sometimes need to start up from zero to serve a new request. For certain applications, this delay isn't much of an impact, but for something like a low-latency financial application, this delay wouldn't be acceptable.
- **Monitoring and debugging:** These operational tasks are challenging in any distributed system, and the move to both microservices and serverless architectures (and the combination of the two) has only exacerbated the complexity associated with managing these environments carefully.

<https://www.ibm.com/cloud/learn/serverless>

Understanding the Serverless stack



<https://cloud.ibm.com/docs/tutorials?topic=solution-tutorials-serverless-api-webapp>

<https://www.ibm.com/cloud/learn/serverless>

Serverless for Data Processing

Serverless is well-suited to working with structured text, audio, image, and video data around tasks that include the following:

- Data enrichment, transformation, validation, cleansing
- PDF processing
- Audio normalization
- Image rotation, sharpening, and noise reduction
- Thumbnail generation
- Image OCR'ing
- Applying ML toolkits
- Video transcoding

<https://www.ibm.com/cloud/learn/serverless>

Massively parallel compute/“Map” operations

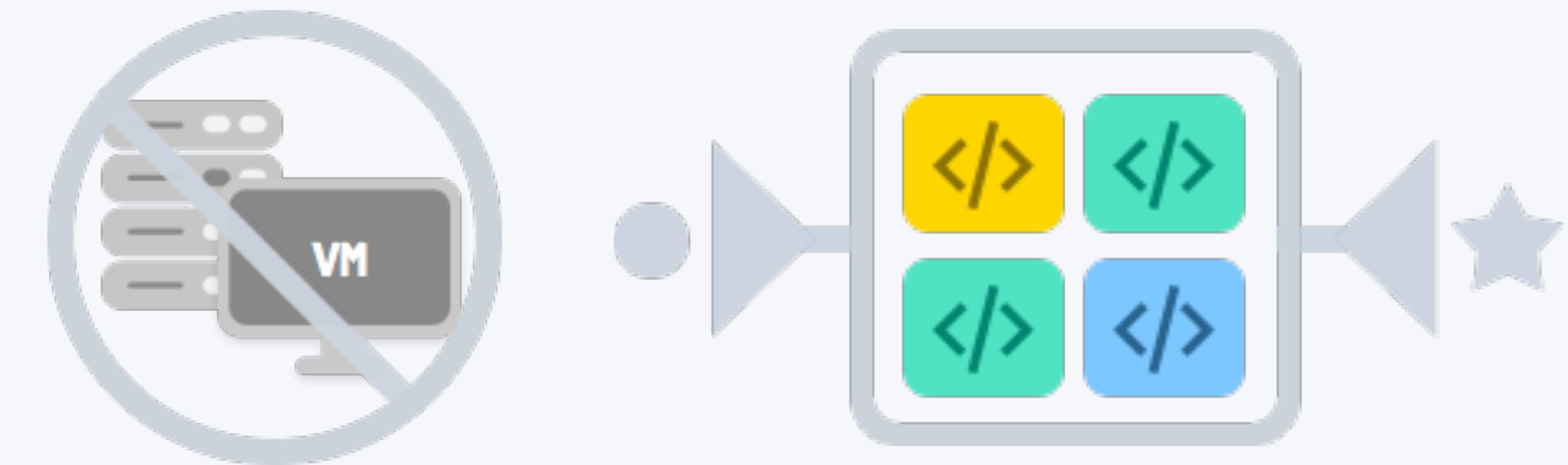
- Any kind of embarrassingly parallel task is very well-suited to be run on a serverless runtime. Each parallelizable task results in one action invocation. Possible tasks include the following:
 - Data search and processing (specifically Cloud Object Storage)
 - Map(-Reduce) operations
 - Monte Carlo simulations
 - Hyperparameter tuning
 - Web scraping
 - Genome processing

Stream Processing Workloads

- **Combining managed Apache Kafka with FaaS and database/storage** offers a powerful foundation for real-time buildouts of data pipelines and streaming apps. These architectures are ideally suited for working with all sorts of data stream ingestions (for validation, cleansing, enrichment, transformation), including:
- Business data streams (from other data sources)
- IoT sensor data
- Log data
- Financial (market) data

Runs code **only** on-demand on
a per-request basis

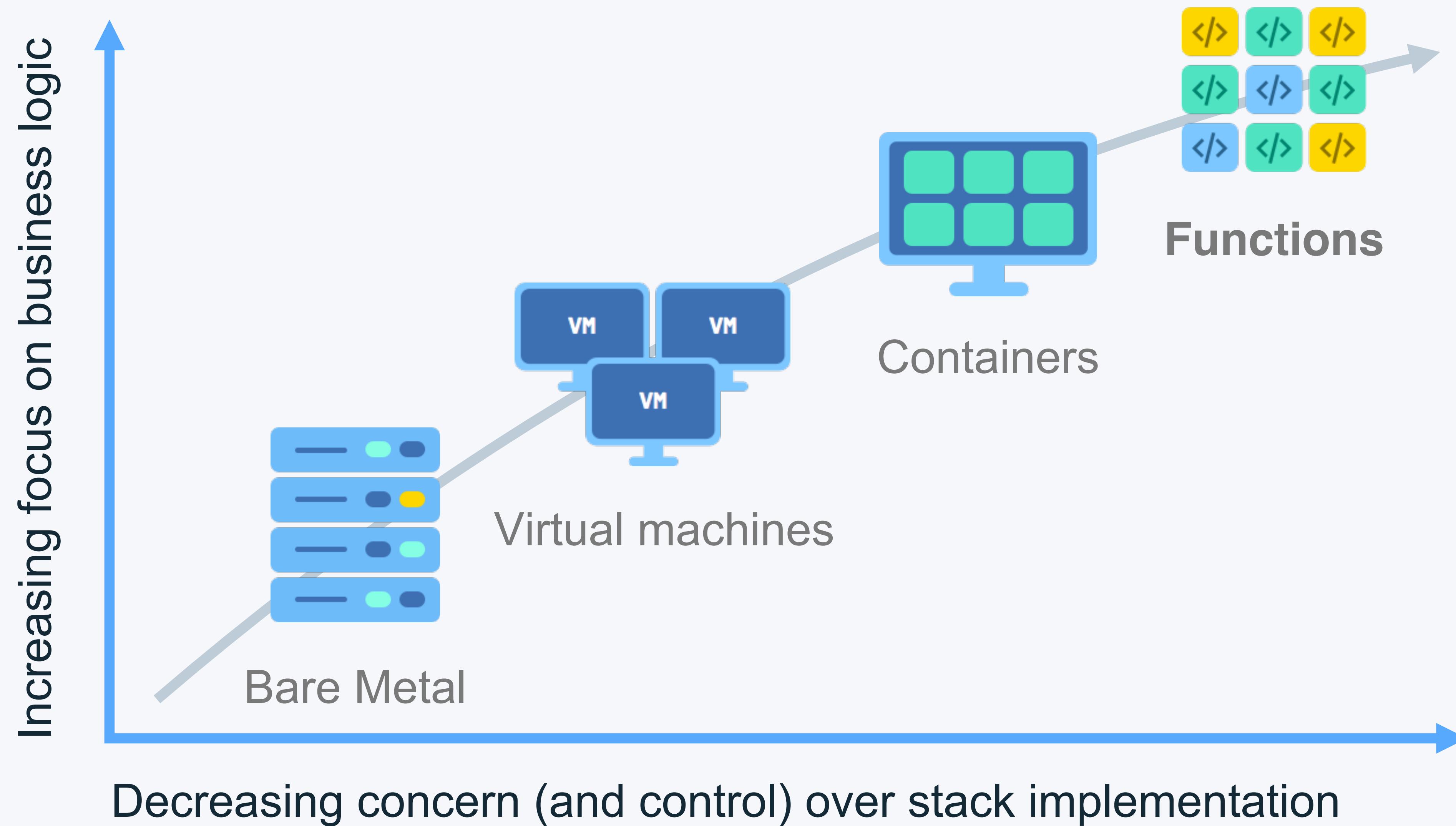
Serverless deployment & operations model



- You write code that executes only in response to events.
- Instances of your functions will start and stop on demand.

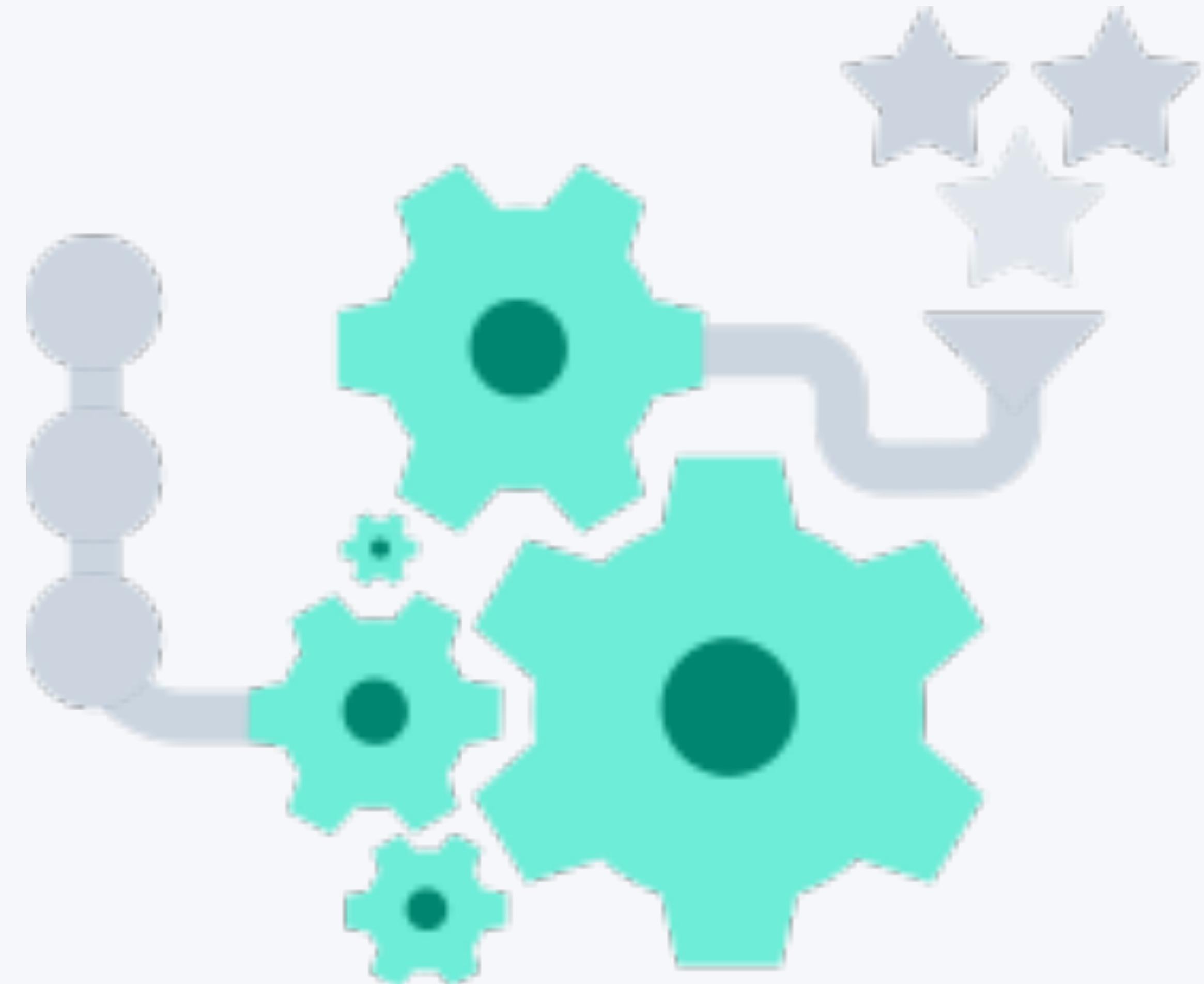
What is Serverless computing (Functions-as-a-Service)?

IBM Cloud Functions



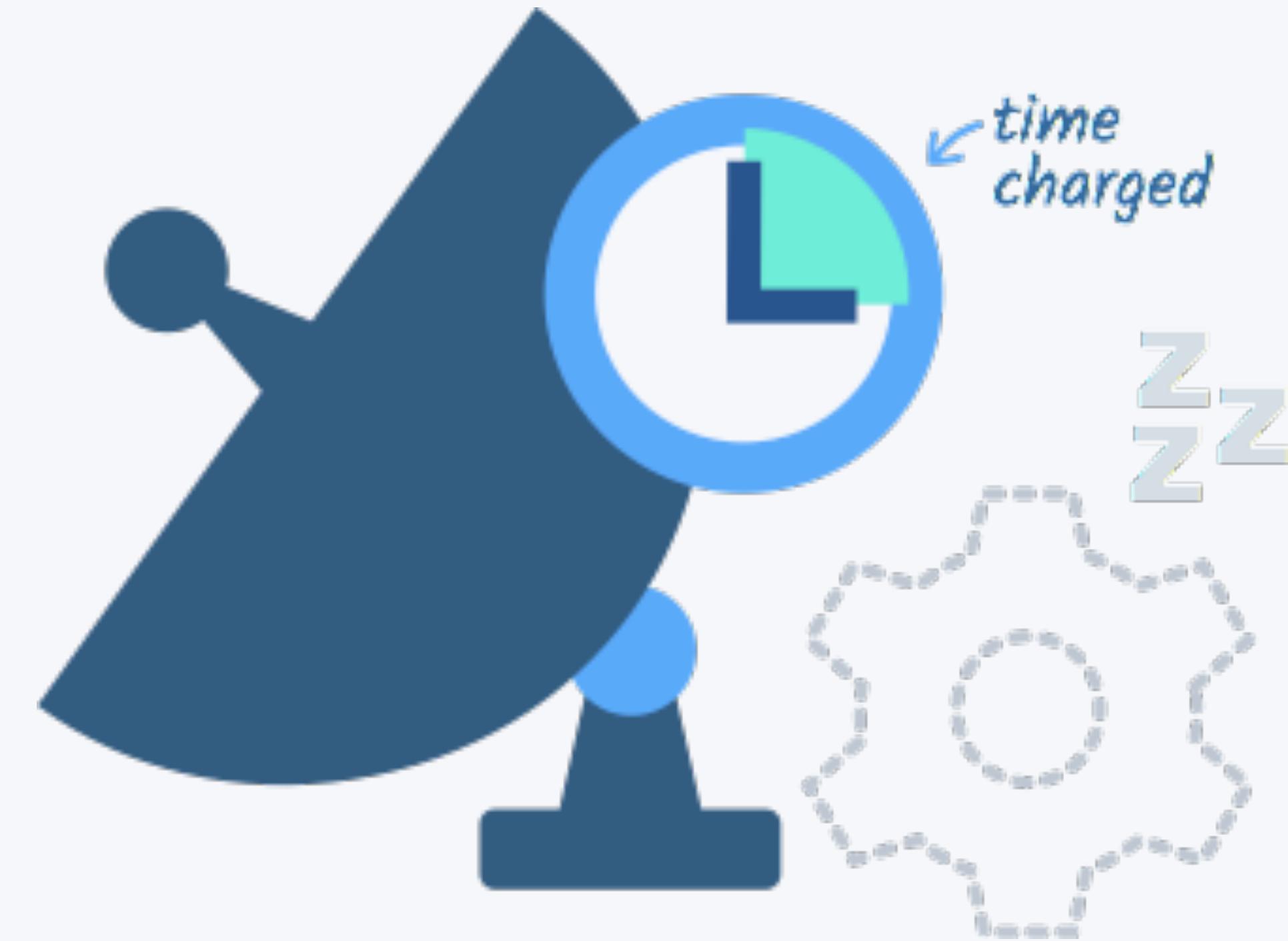
Runs code **only** on-demand on
a per-request basis

Scales on
a per-request
basis



Runs code **only** on-demand on
a per-request basis

Optimal
utilization &
granular pricing



It also means that there's a tight match
between resources consumed and the price you
are charged, down to about 100ms.

Traditional model

Worry about scaling

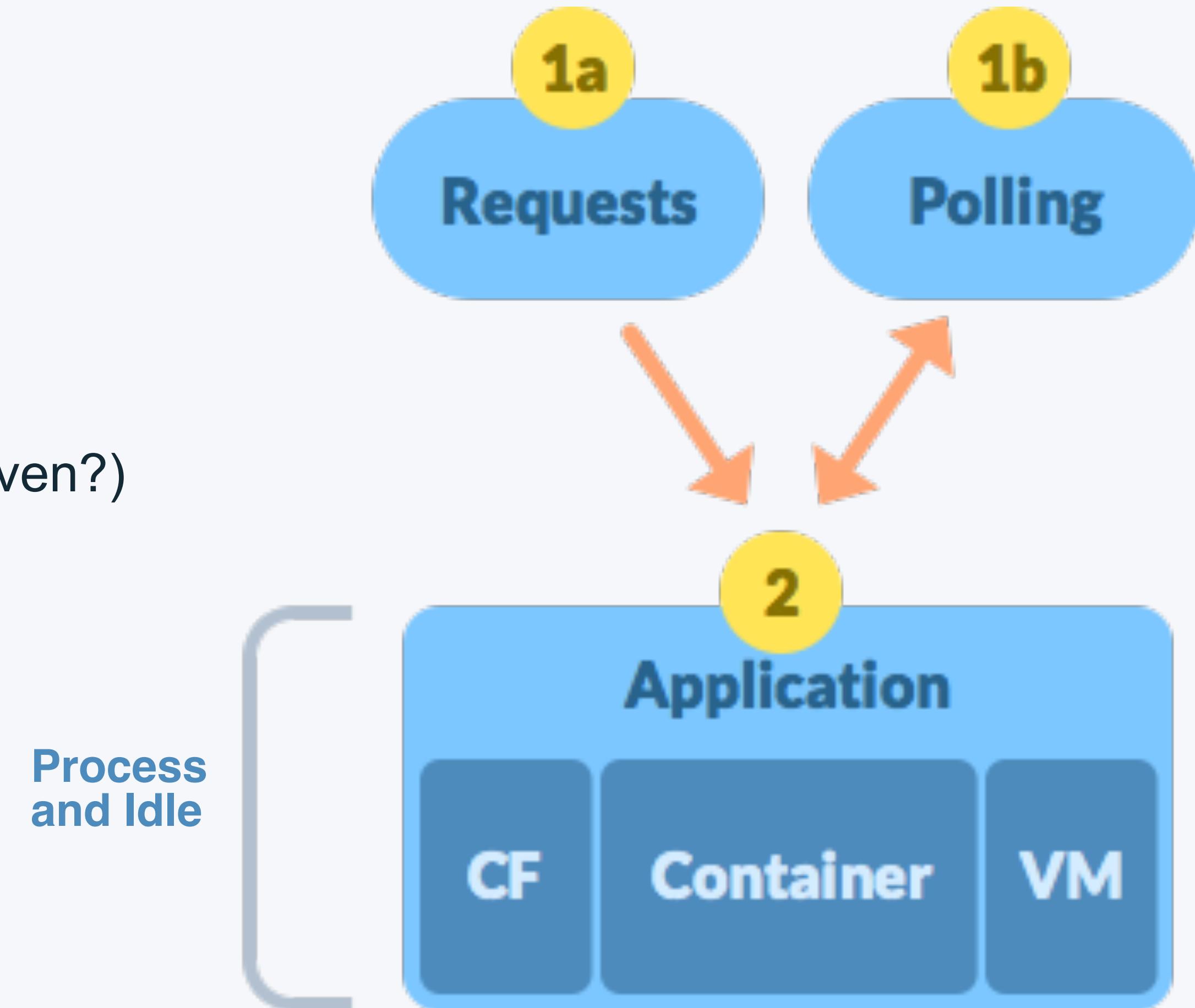
- When to scale? (mem-, cpu-, response time-, etc. driven?)
- How fast can you scale?

Worry about resiliency & cost

- At least 2 processes for HA
- Keep them running & healthy
- Deployment in multiple regions

Charged even when idling / not 100% utilized

Continuous polling due to missing event programming model

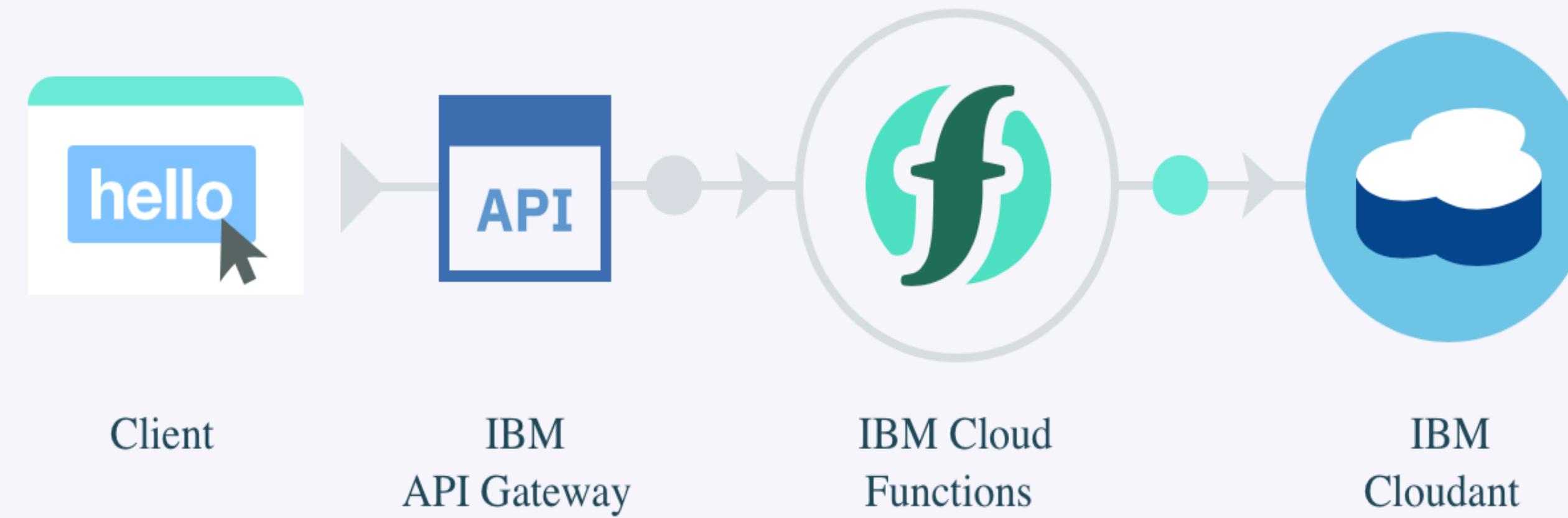


What should I use Cloud functions for ?

Serverless Backends

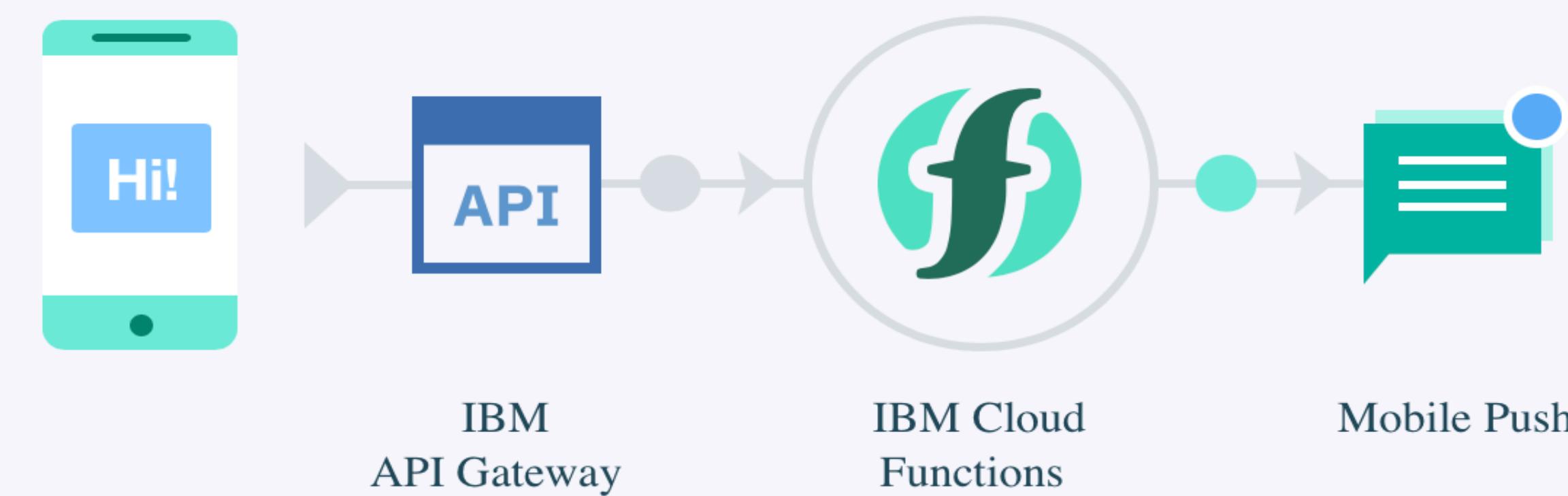
Expose application logic by implementing serverless microservices. Simply map your functions to well-defined API endpoints any client can call by making use of Web Actions or our latest API Gateway integration.

•



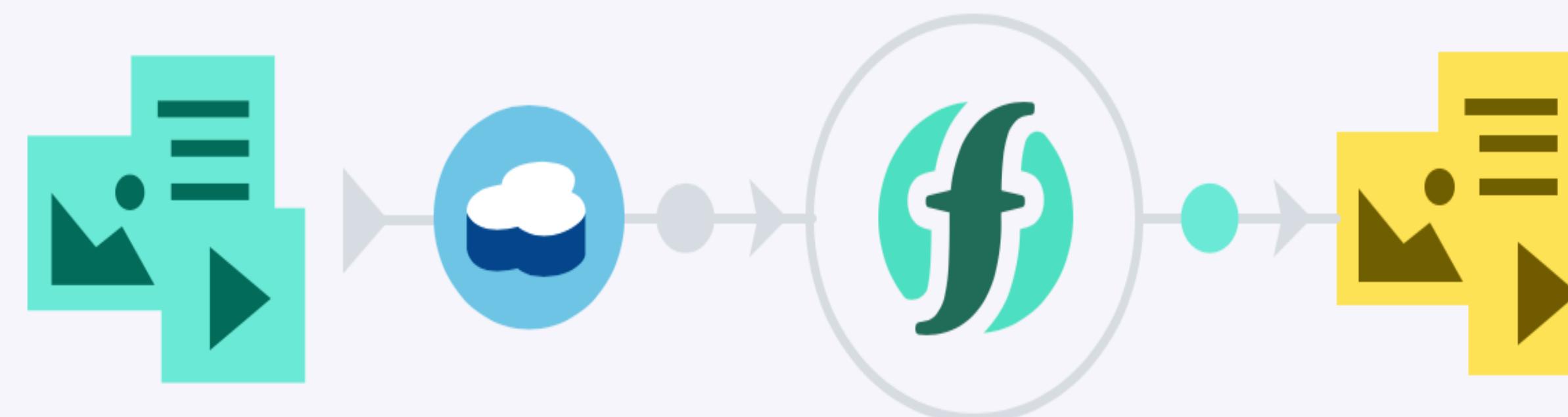
Mobile Backend

Allow mobile developers to easily access server-side logic and to outsource compute-intensive tasks to a scalable cloud platform. Let them implement functions in languages like Swift and easily consume server-side functions using our iOS SDK.



Data Processing

Execute code whenever data is updated in your datastore. Easily automate processes like audio normalization, image rotation, sharpening, noise reduction, thumbnail generation, or video transcoding.

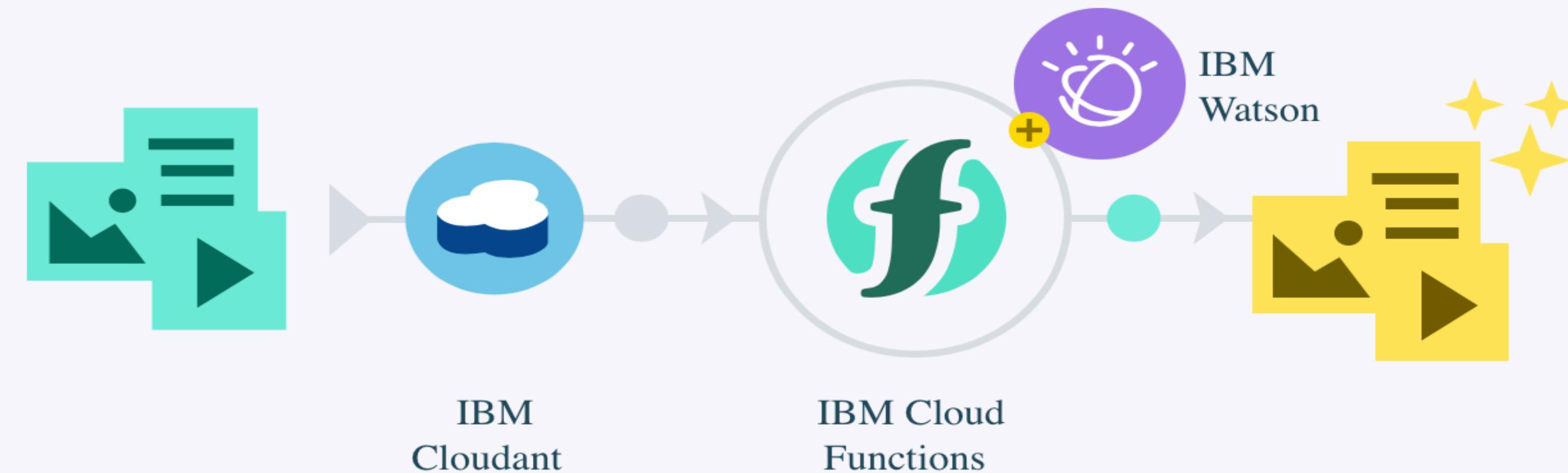


IBM
Cloudant

IBM Cloud
Functions

Cognitive Data Processing

Analyze data as soon as it becomes available. Let your function make use of powerful cognitive services like IBM Watson to detect objects or people appearing in images or videos.



IBM
Cloudant

IBM Cloud
Functions

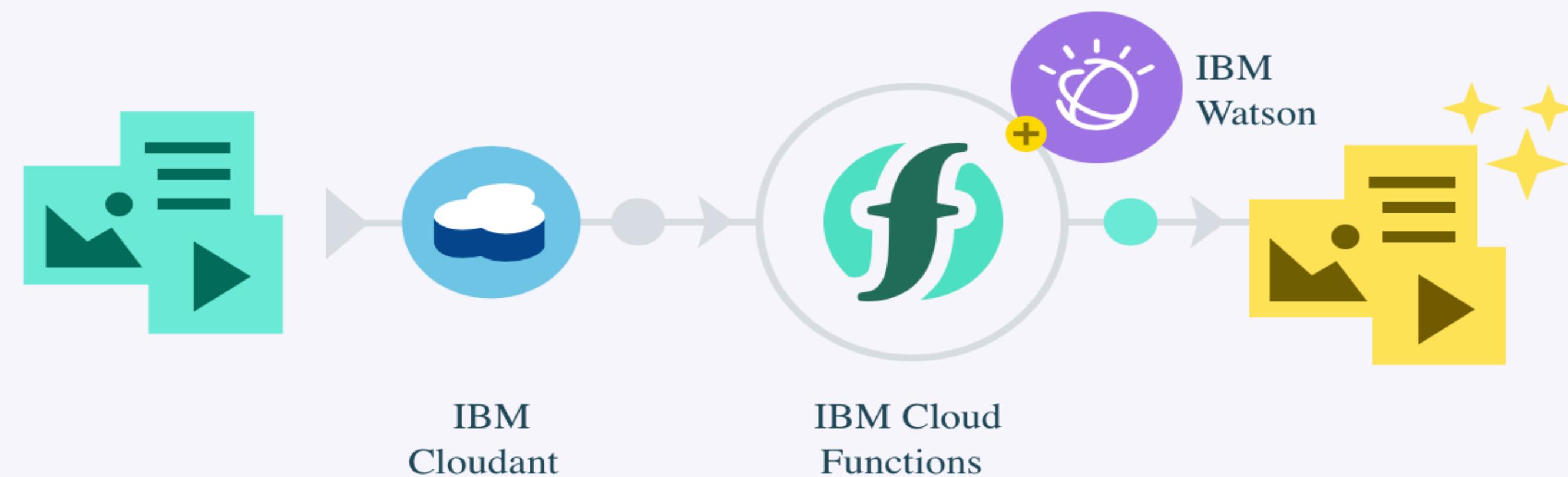
Case-Study: With Skylink, a drone is continuously taking and analyzing pictures while flying using an IBM Watson service.

Coding 2

- Python - **ibm.biz/sv-serverless-2020**
- Watson – visual recognition

Cognitive Data Processing

Analyze data as soon as it becomes available. Let your function make use of powerful cognitive services like IBM Watson to detect objects or people appearing in images or videos.



IoT Ready

React to and process IoT sensor data. Let any IoT device send data to our IBM Watson IoT platform and define cloud rules to call your functions and execute custom application logic.



Case-Study: GreenQ: Cloud Functions for a Smart City Data Application.

Event Stream Processing

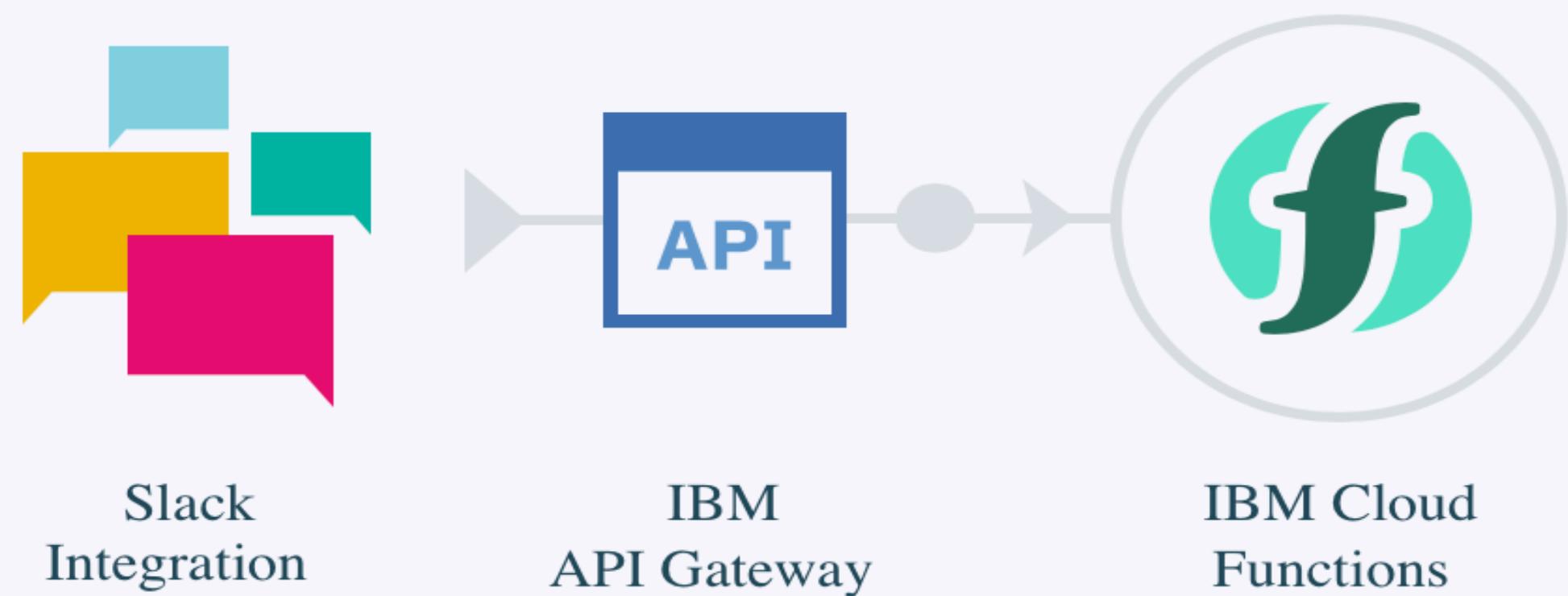
Process data in-motion and make your functions react to incoming messages. Benefit from our IBM Event Streams—an IBM Cloud managed Apache Kafka integration.



Conversational Scenarios

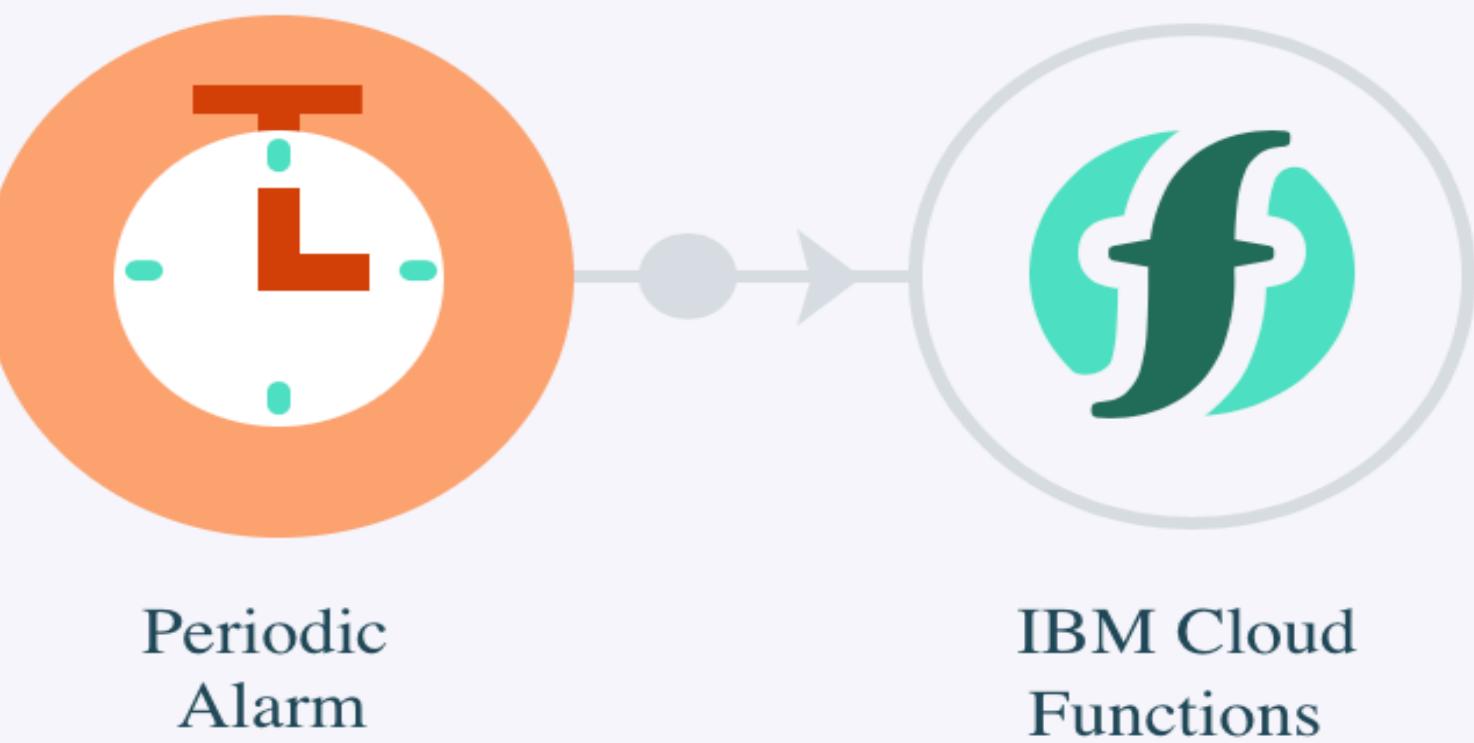
Implement serverless conversational applications, like chatbots, by passing chat messages to your functions for further processing.

[Go to tutorial](#)



Scheduled Tasks

Execute your functions periodically. Define schedules following a cron-like syntax to specify when actions are supposed to be executed.



IBM Cloud Functions

Functions-as-a-Service (FaaS) platform based on Apache OpenWhisk

Develop your functions directly in one of the [natively supported languages](#), or run code in any other language (including compiled Go, C, etc. binaries) by providing us with a Docker container.



Python



.NET



Ruby



Any language
(via Docker)

Event Providers

Benefit from an ecosystem of event consumers and emitters from different areas like analytics, cognitive, data, IoT, mobile, and more. Using our open event-provider interface, [you can enable](#) any service you would like to use.



Github



Cloudant



Event
Streams

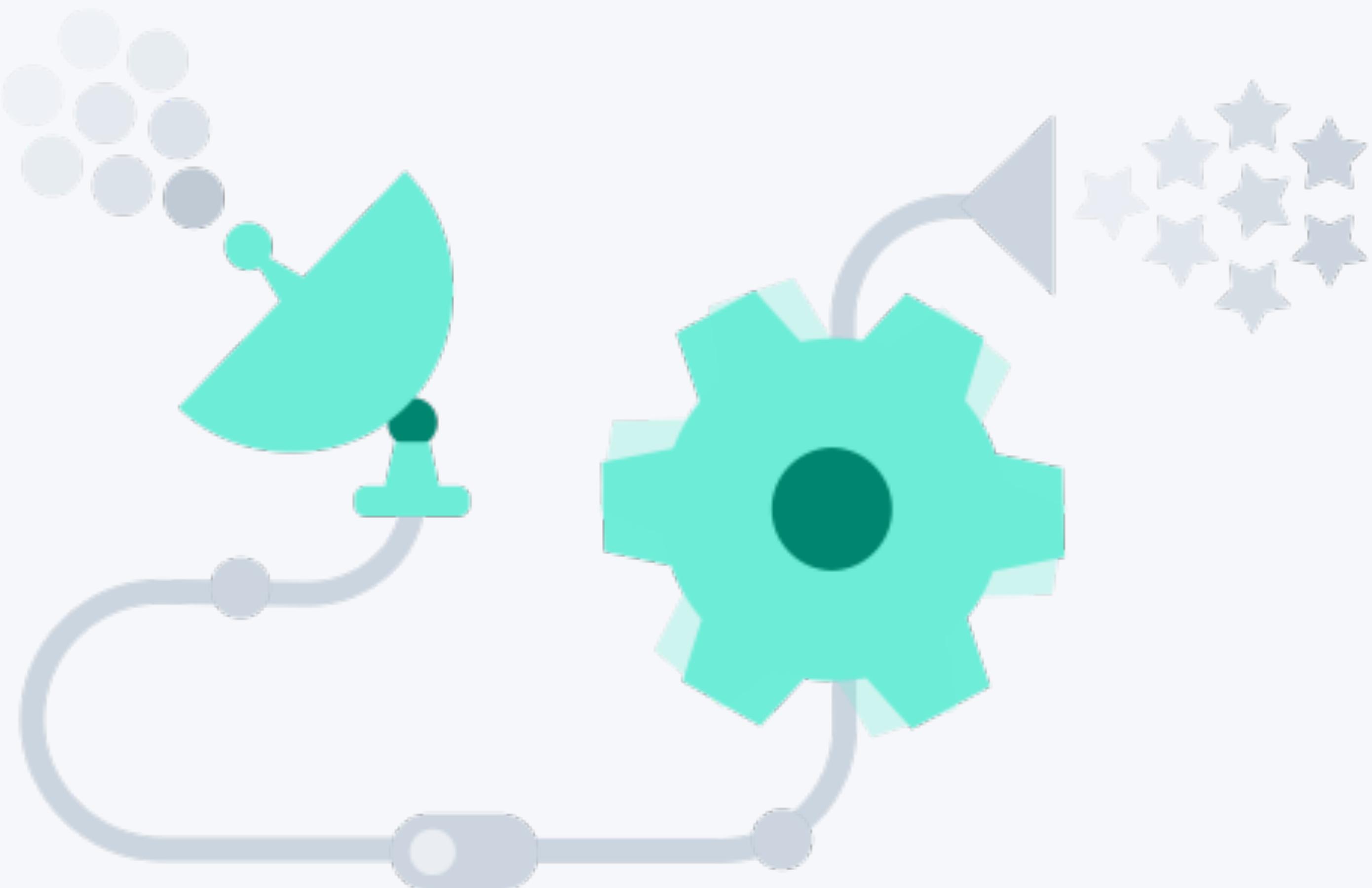


Mobile Push



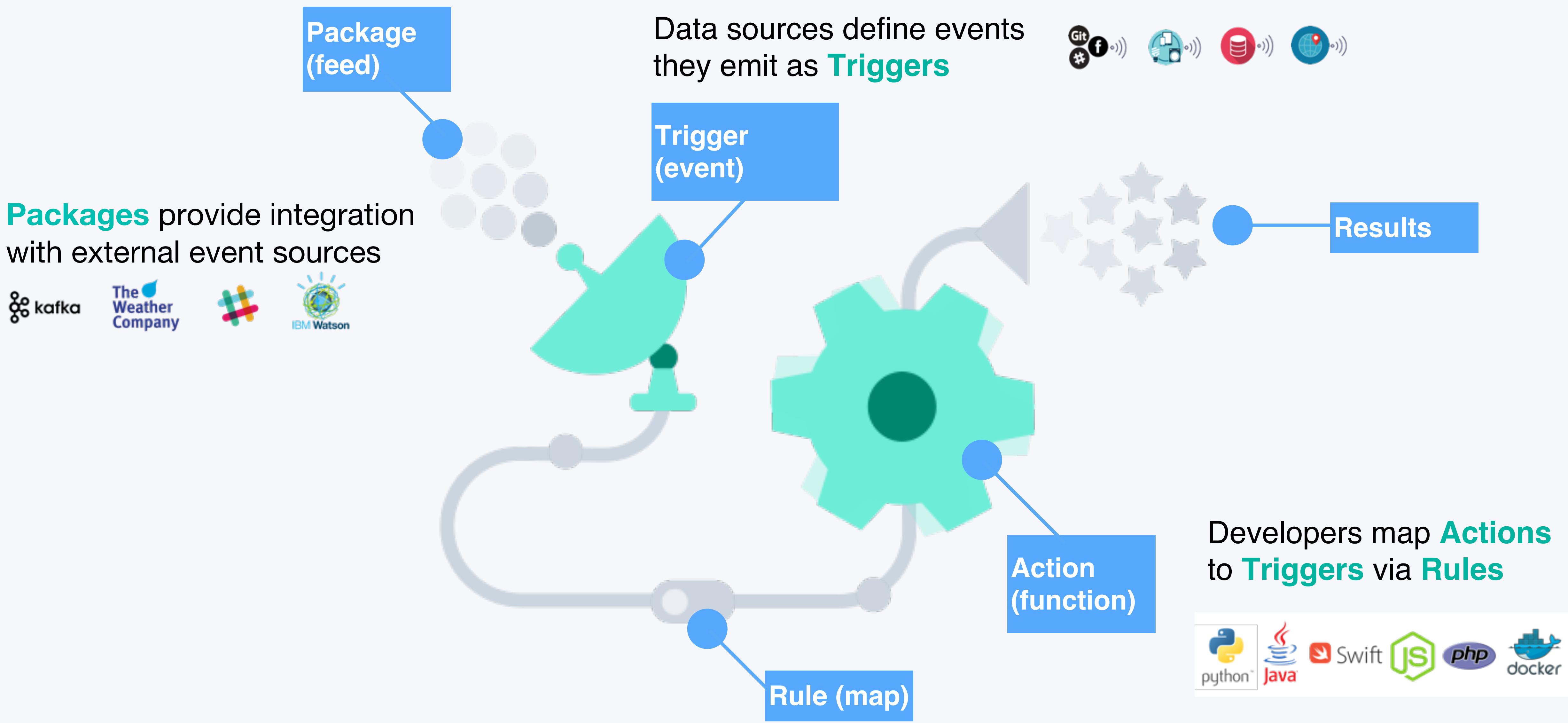
Periodic
Alarm

Serverless
platform to
execute code in
response to
events



What is IBM Cloud Functions (Apache OpenWhisk)?

IBM Cloud Functions



Supported Languages

Multi-language Support

JS/NodeJS 10

Swift 4

Java

GO

Python 3

PHP 7

Docker – *everything goes!*

Community Efforts

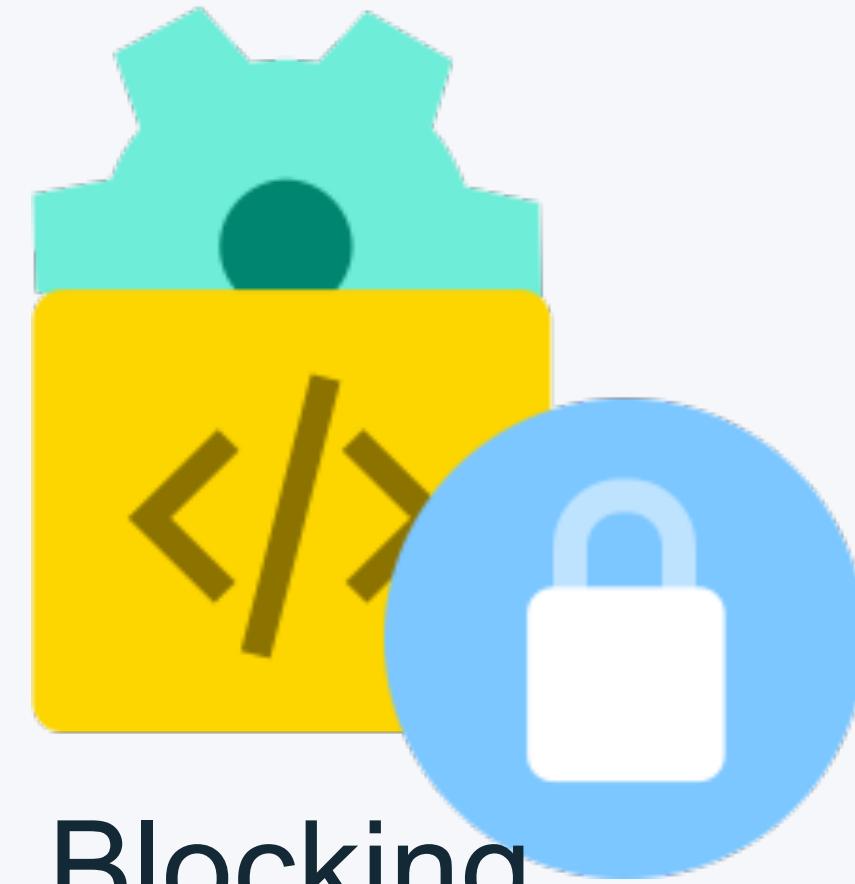
Haskell

Scala

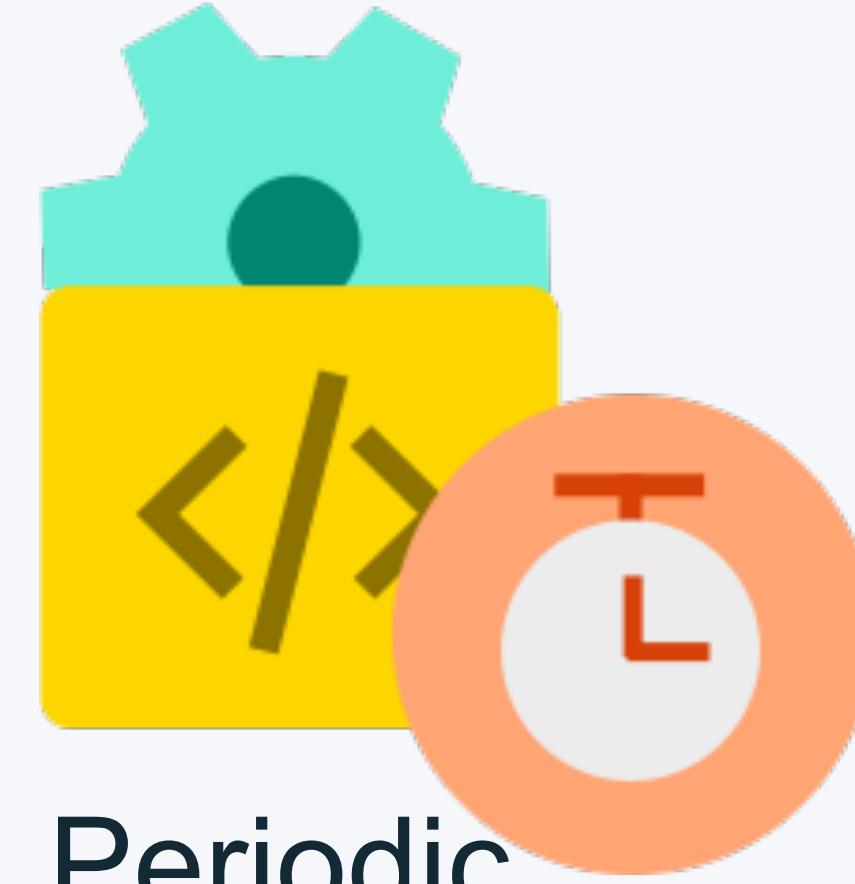
...

... and more to come

Support for different invocation models



Blocking

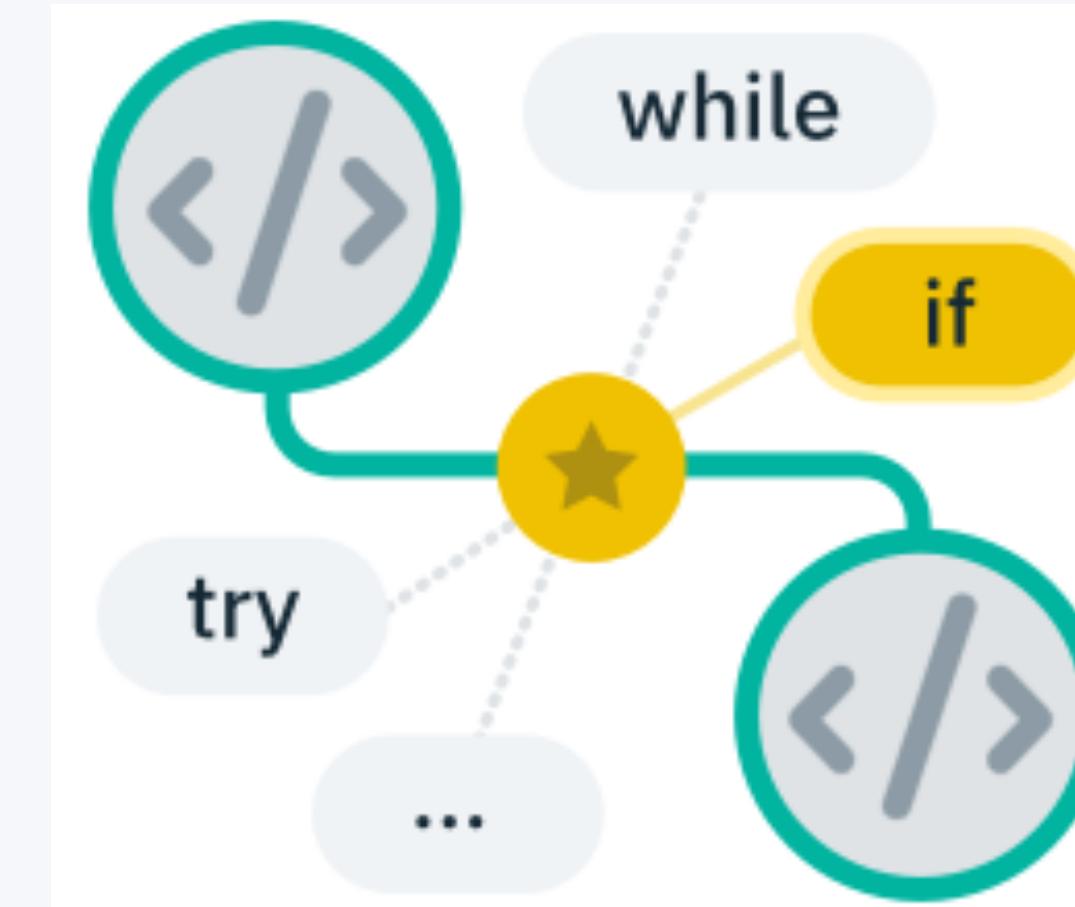


Periodic

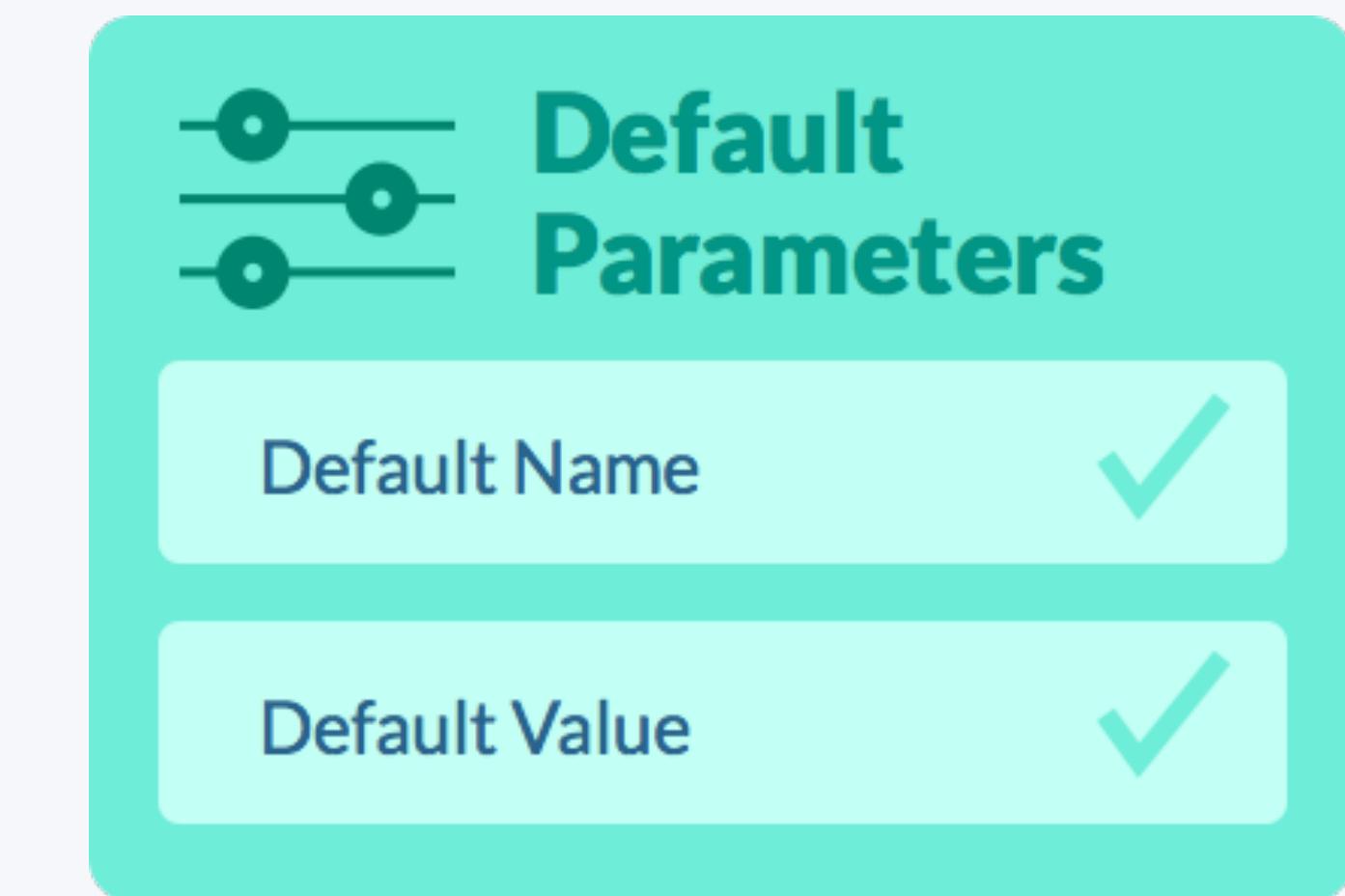


Non-blocking

Supports higher-level programming constructs

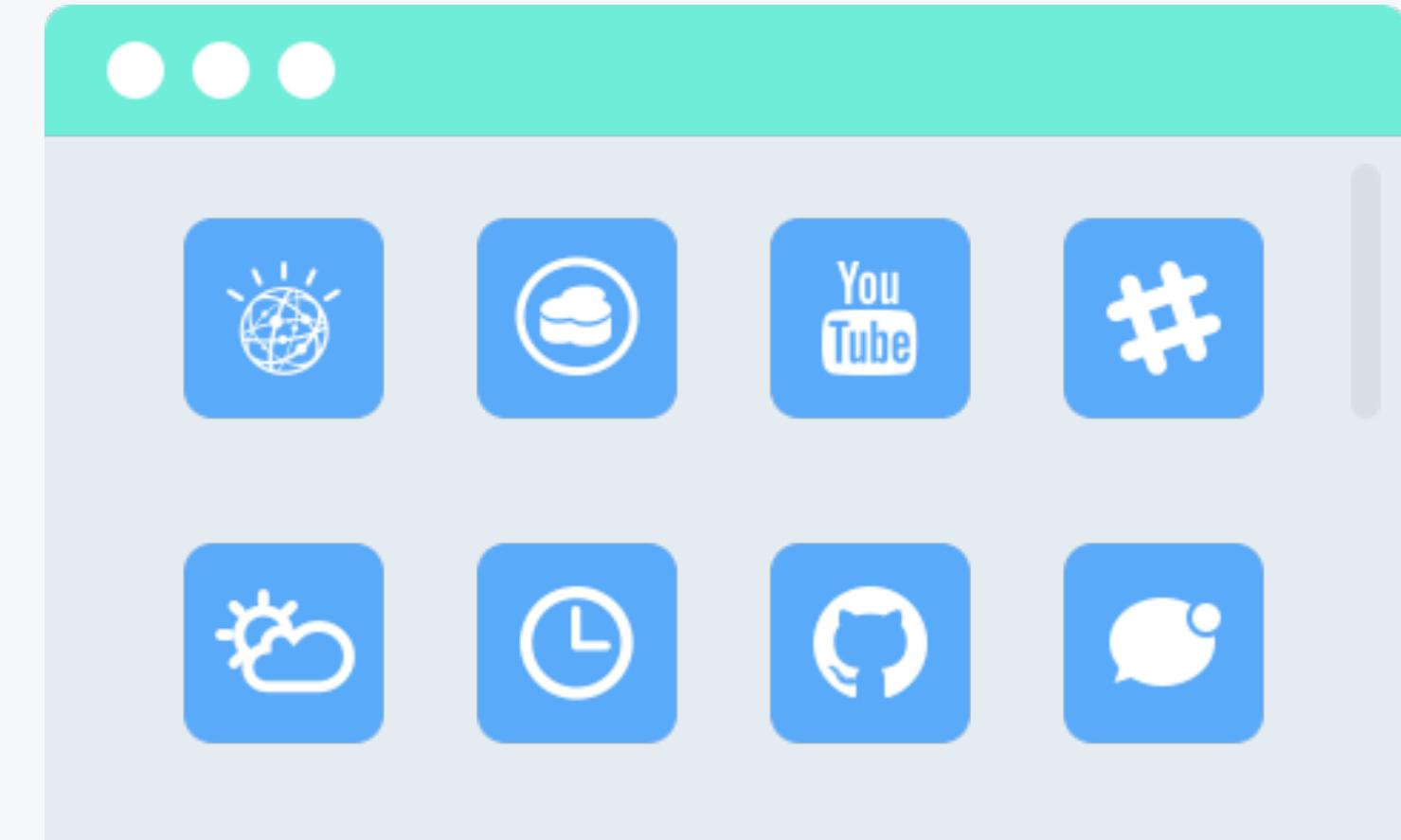


Sequencing
Conditionals
Loops
Error handling



Parameter
binding

Event Provider



Open event emitter
(consumer ecosystem)

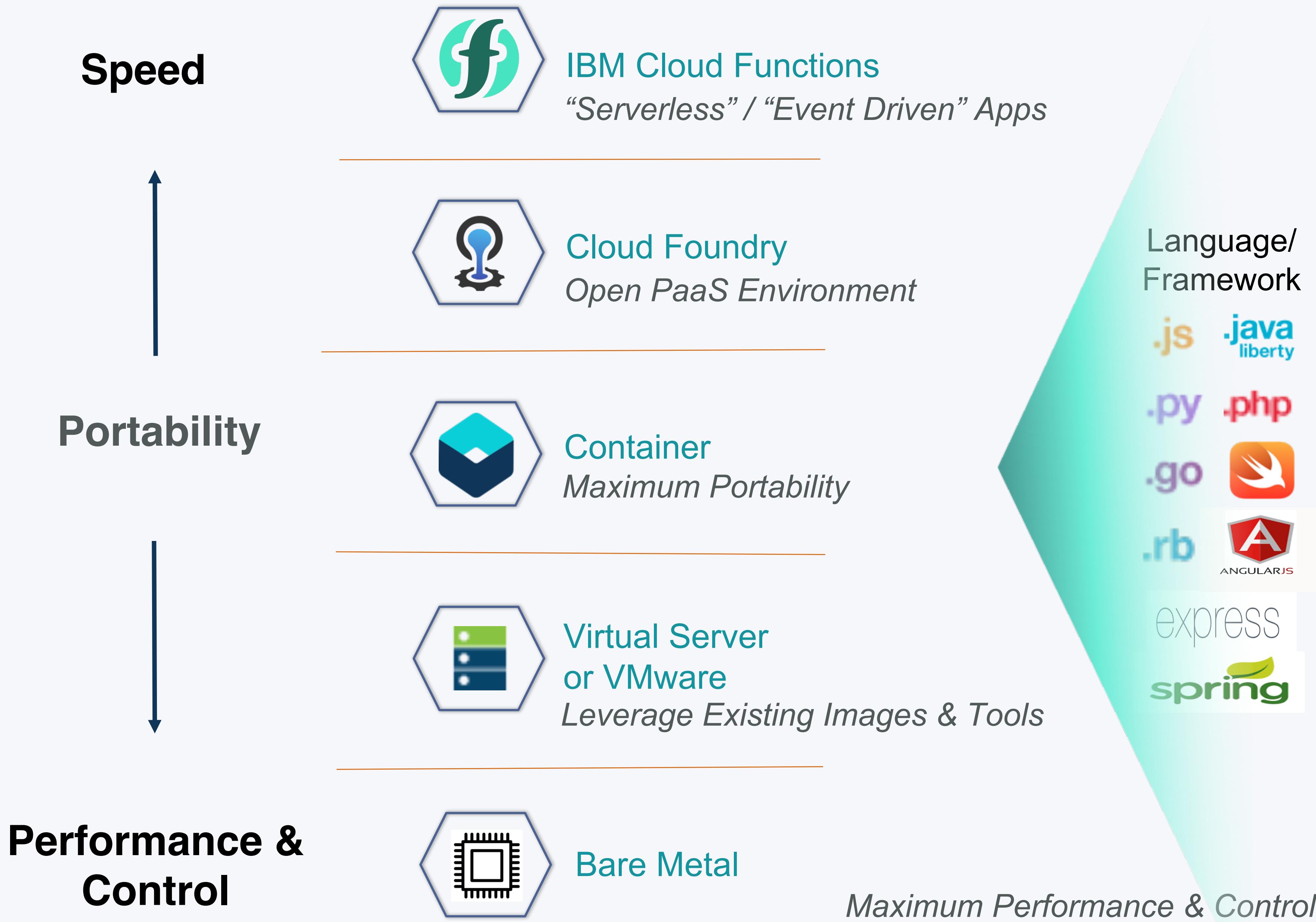


Open interface
for event emitters

Serverless Framework support

Allows packaging of pieces of a serverless application into a single project and deploy it in a vendor-agnostic way.







Full control over infrastructure
and maximum portability

Containers-as-a-Service

- **Control** over runtime environment (runtimes, versions, minimal OS).
- Greater **reusability and portability** of container images.
- Great fit for bringing containerized **apps and systems to the cloud**.

- More responsibility over package configuration (security patches).
- Need to understand distributed systems.



Focus on the application and let
the platform handle the rest

Platform-as-a-Service

- No need to manage underlying OS.
- **Buildpacks provide influence over the runtime**, giving as much or as little control (**sensible defaults**) as desired.
- Great fit for many **existing web apps** with a stable programming model.

- Loss of control over operating system, possibly at the mercy of buildpack versions.
- Limited to HTTP/HTTPS

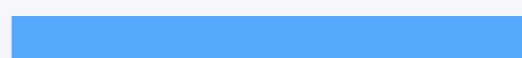


**Auto-scaled, event-driven applications
that respond to a variety of triggers**

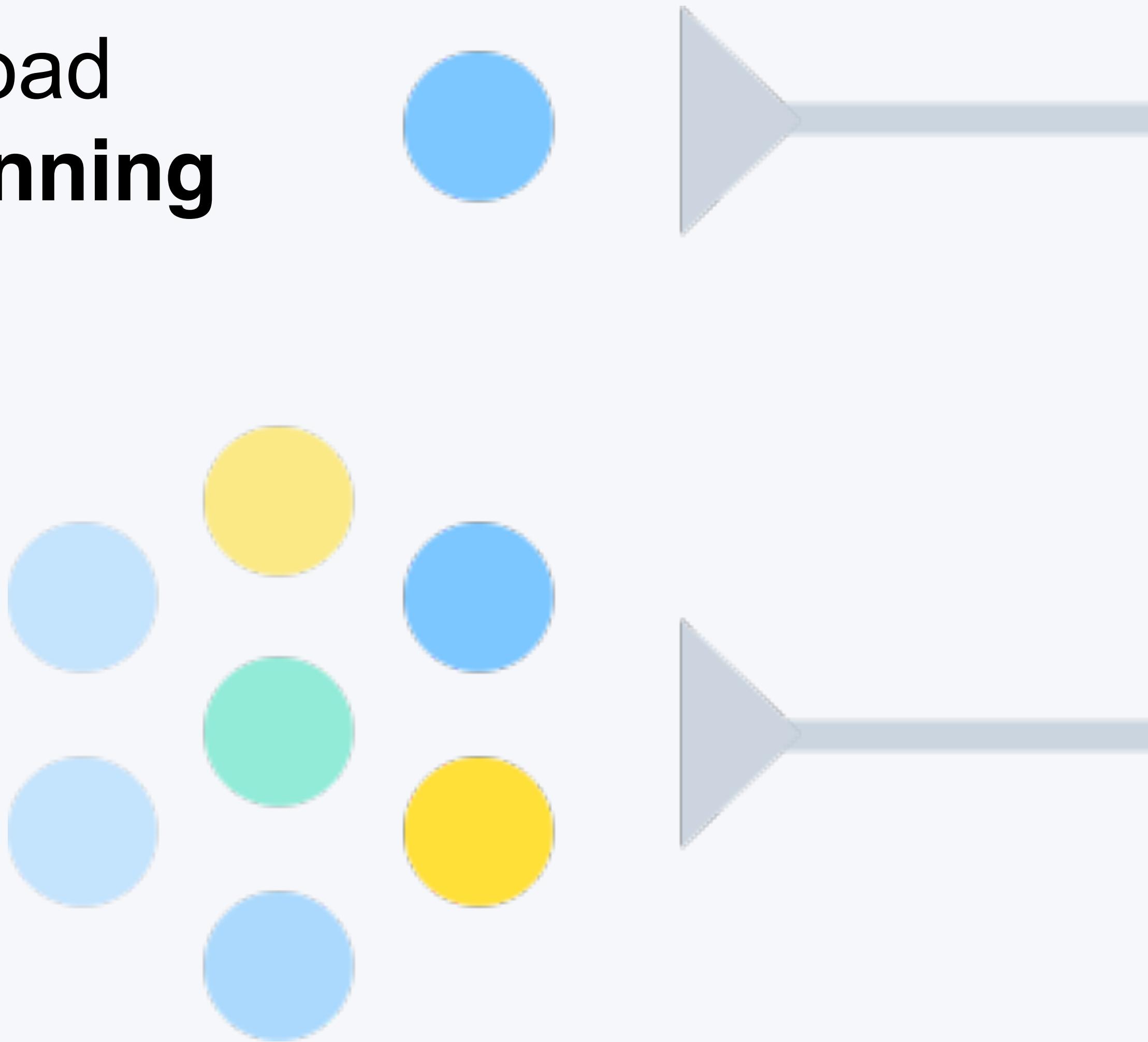
Functions-as-a-Service

- OS, runtime, and even **container lifecycle is completely abstracted** (serverless).
- **Autoscales** in response to demand, with an associated **granular cost model**.
- Great fit for emerging, non-HTTP, **event-driven workloads** involving IoT, data, messages.
- Emerging computing model, rapid innovation with less comprehensive and stable documentation, samples, tools, and best practices.

Volatile and/or **event-driven** workload
that can be split in smaller **short-running**
pieces.



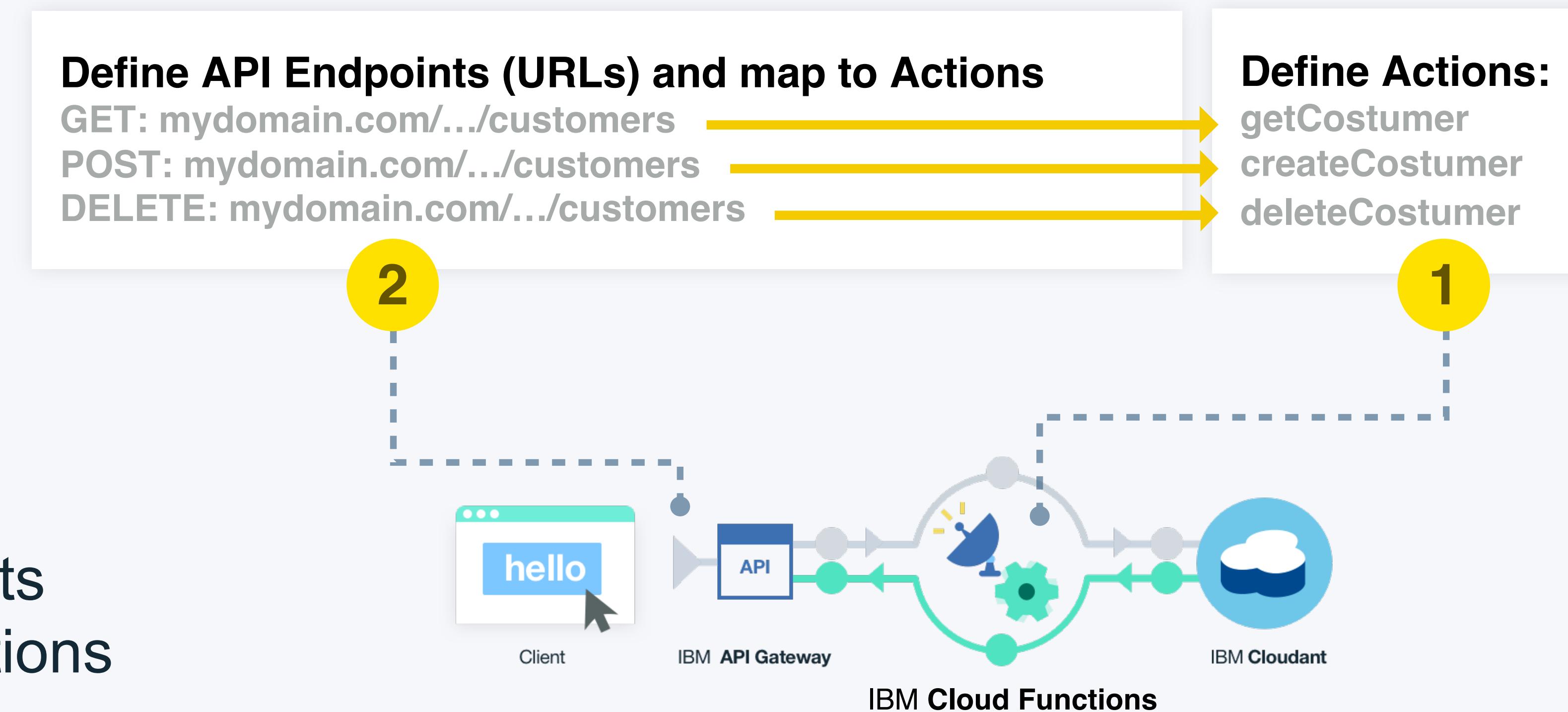
Suited for sporadic as well
as heavy load scenarios.



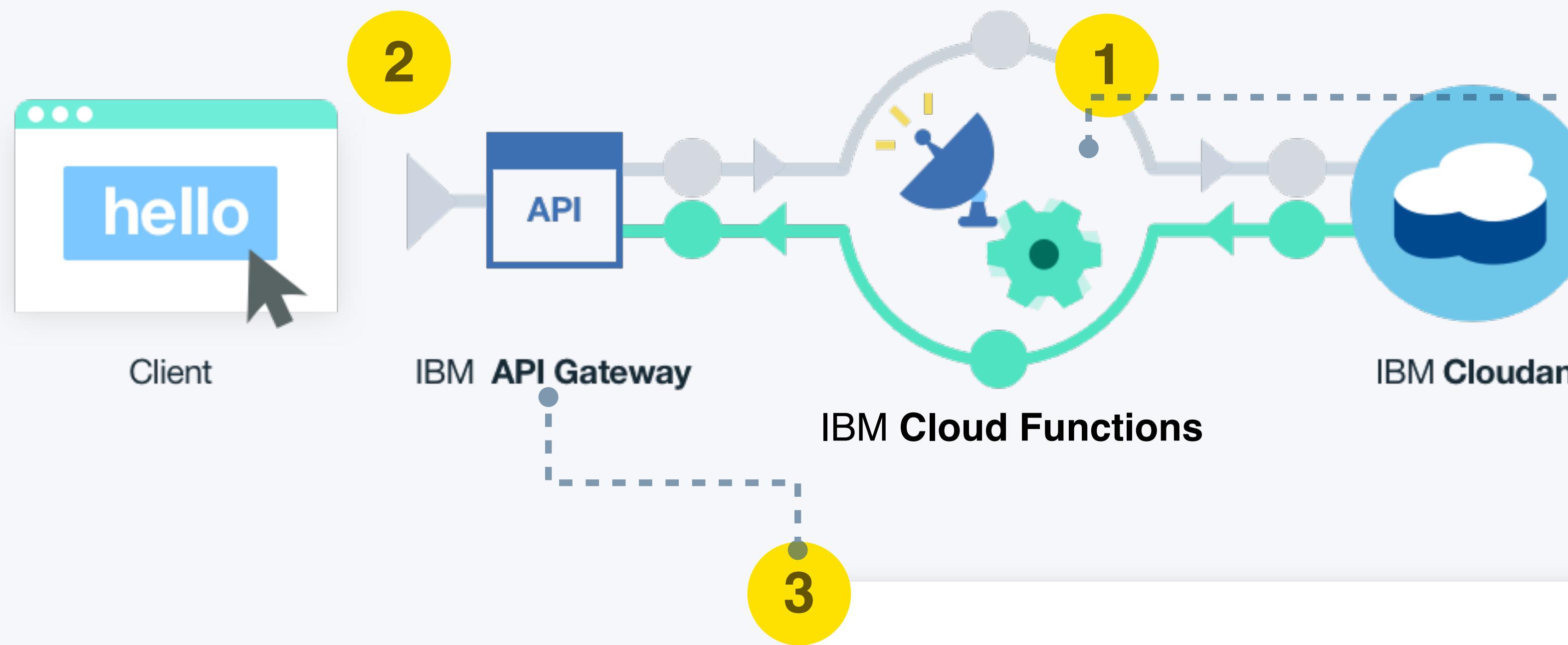
Serverless microservice APIs/backend

API Gateway support

Allows to map API endpoints to IBM Cloud Functions actions



Available for free, without limits

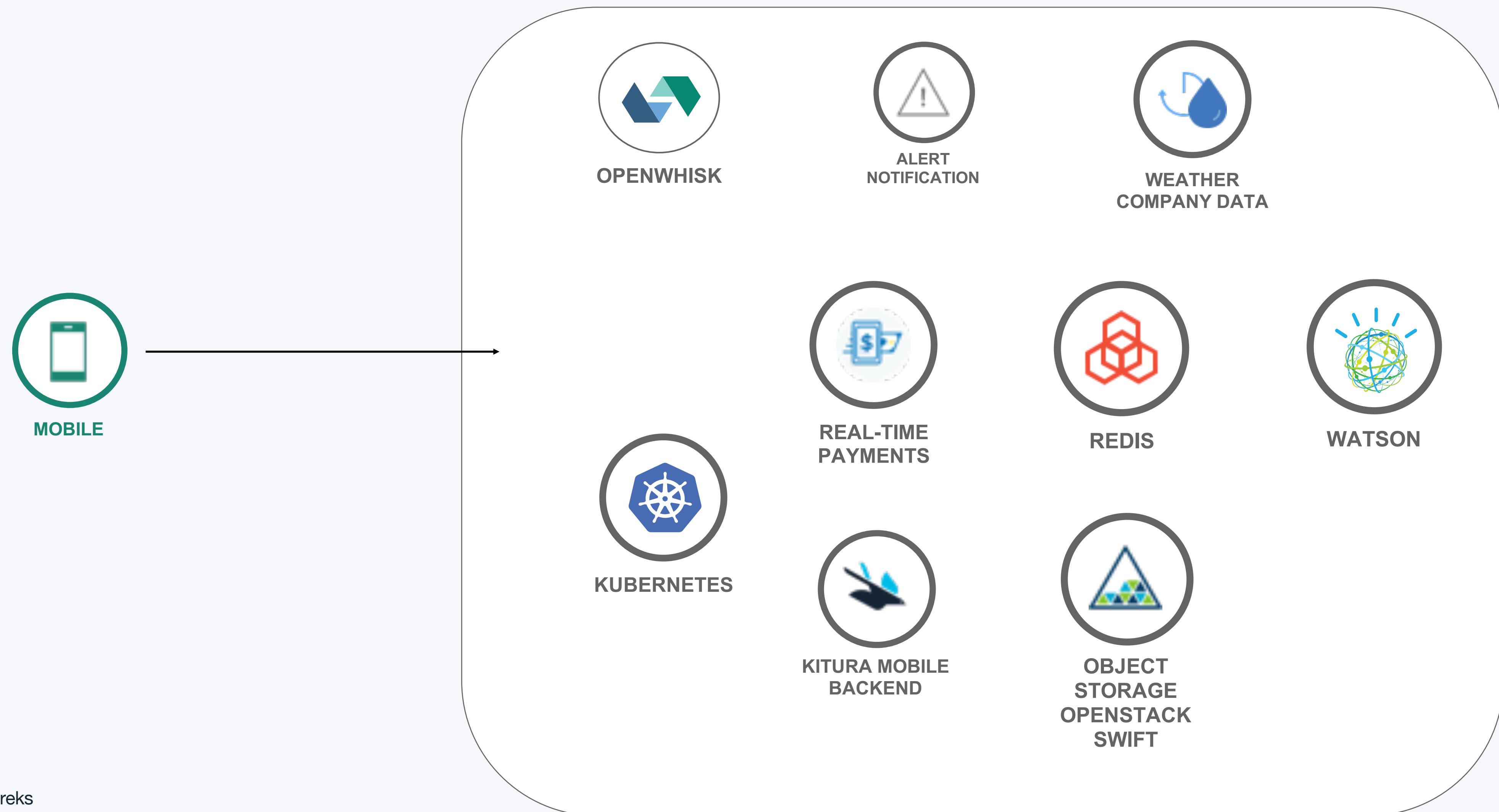


Defined Actions:

- getCostumer
- createCostumer
- deleteCostumer

Easy to add and edit:
Security (API key, API secret, OAuth validation, CORS)
Rate-Limiting
Map actions to API endpoints (OpenAPI Doc creation)
Easy socialization (sharing, API key creation)
Analytics (API calls, errors, response time)
Test your API (API Explorer)
Upload Swagger/OpenAPI Doc

The Kubernetes and FaaS



IBM Cloud Functions Shell

```
> $ ls
```

14205e3f64724676a05...	foo	74ms	ok
fbe12a1e0f1a4d22a12...	foo	3ms	ok
ed1f34092ddc43789f3...	foo	3ms	ok
79698e22220144e1a98...	foo	91ms	ok
7be10a84f5864fbba10...	bar	3ms	ok
c008ee7b92a5480388e...	bar	2ms	ok
4d201f524a2f4e09a01...	bar	3ms	ok
fb51e5549dc4439f91e...	bar	61ms	ok
85956eb2aae844f7956...	seq-2	72ms	ok
1f9a5e04c6be4af9a5...	seq-1	63ms	ok

SUMMARY TIMELINE GRID Showing 1-10 < >

```
ok
> grid
ok
> screenshot
/
```

OPENWHISK.NG.BLUEMIX.NET VMRC_WSKNG_TEST HELP

GRID Recent Activity

fastest 320 slowest 4

failures 1

Showing 400 activations from 1/29/2018, 3:44:45

foo

Summary Timeline Grid

grid

SUMMARY

Recent Activity

Showing 400 activations from 1/29/2018, 3:44:49 PM spanning ~8d

	SELECT RANGE	20ms	173ms	325ms	478ms	Successes	Failures	
helloapp						25th percentile ►	3	-
zombie						median ►	2	-
openwhisk-composer conductor							5	-
seq							1	-
foo							378	-
bar							4	-
bomb							0	1
p a							4	-
seq-2							1	-
seq-1							1	-

Summary Timeline Grid



Serverless code patterns

 <p>Code Pattern Build a Knative serverless web application Nov 19, 2019 →</p>	 <p>Code Pattern Create an Alexa skill with serverless and a conversation Jul 16, 2019 →</p>	 <p>Code Pattern Serverless image processing with Cloud Object Storage Jul 10, 2019 →</p>
 <p>Code Pattern Enhance customer helpdesks with Smart Document Understanding Jul 08, 2019 →</p>	 <p>Code Pattern Create Box Custom Skills with Watson AI May 22, 2019 →</p>	 <p>Code Pattern Create serverless functions to send push notifications Apr 12, 2019 →</p>
 <p>Code Pattern Mirror game app showcases 15 cloud technologies and components Feb 05, 2019 →</p>	 <p>Code Pattern Archived Accelerate AI data preprocessing with PyWren and IBM Cloud Functions Feb 01, 2019 →</p>	 <p>Code Pattern Securely integrate serverless functions with on-premises resources Dec 26, 2018 →</p>

<https://developer.ibm.com/technologies/serverless/patterns/>

THANK YOU

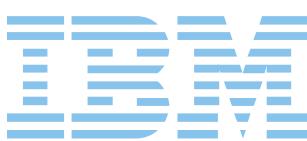
@blumareks

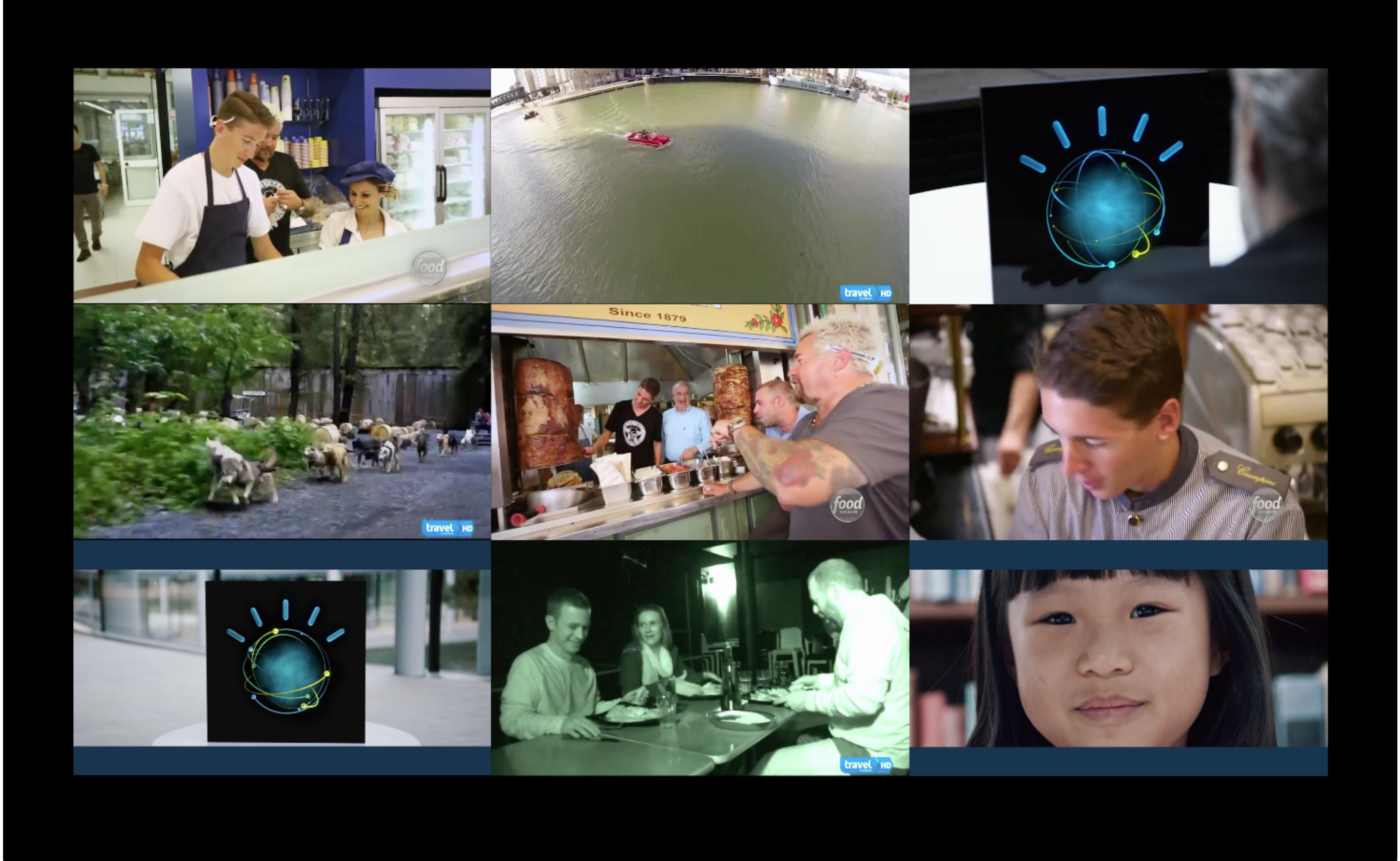
IBM Developer

2.5 quintillion bytes of data every day

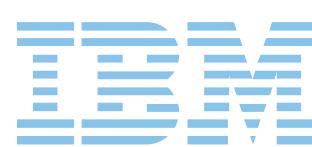
400 hours of video uploaded every minute

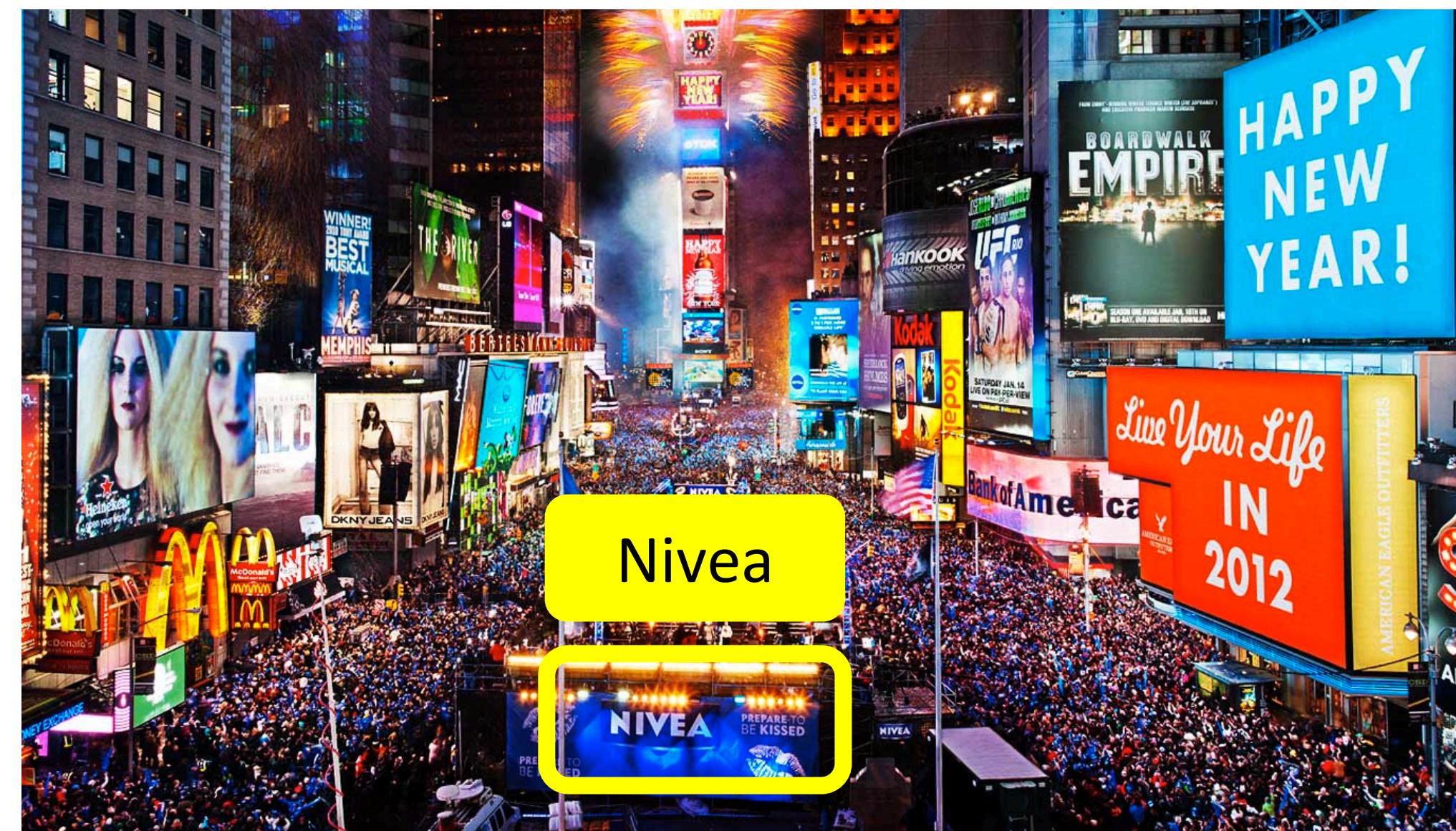
Video will account for 80 to 90% of all consumer Internet traffic by 2019





What a large amount of data to analyze!





Places, products, objects, texts

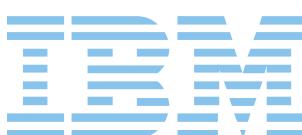
Improve search results

Better navigation

Make recommendations

Detect brands

Use the data to improve search, navigation, recommendation, brand resonance

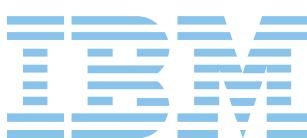


Scale with the number of videos

Run asynchronously

Several independent actions

A good fit for a serverless architecture



iPad Air 2 - iPad Air 2 / iOS 9.3 (13E230)

< Dark Vision

Food Network - Part 1.mp4



Food Network - Part 1.mp4

2 months ago

4025 views

★★★☆☆

tags

person food flower sport sign Face Winter Scene

Indoors Shoes Blue Boating Fishing Human

Mixed Color Room Dental Care Room Male Adult

People in the video



Guy Fieri

Age 35-44

Related videos



Travel Channel - CLIP 1....

6027 views

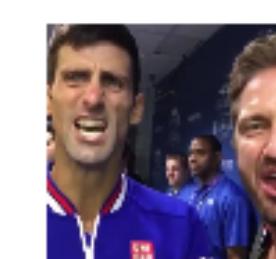
★★★★☆



Travel Channel - CLIP 2....

6027 views

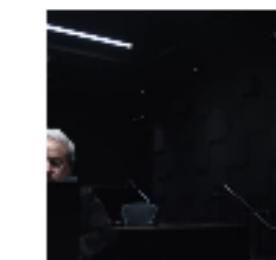
★★★★☆



Novak Djokovic and Ger...

4074 views

★★★★☆



Ridley Scott + IBM Wats...

3045 views

★★★★☆

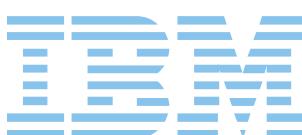


Food Network - Part 2.m...

4025 views

★★★★☆

Using the data we discovered, we can deliver an improved user experience.



Summary

[Relaunch analysis](#)



Watson Visual Recognition

News Studio 76% Mixed Color 72%
Indoors 70%

Alchemy Face Detection

1 Novak Djokovic 92% MALE 35-44 51%
2 Gerard Butler 88% MALE 35-44 52%

Alchemy Image Keywords

people 99%

Summary

[Relaunch analysis](#)



Watson Visual Recognition

People Activity 70% Scene 70% Car 71%
Indoors 74% Human 73% Meat Eater 70%
Head and Shoulders 73% Mixed Color 75%
Wrestling 73% Person View 71%
Air Vehicle 73% Vehicle 74% Landmark 73%
Outdoors 77% Man Made Scene 71%
Truck 74% Face 74% Brown 76%
Boating 71% Water Sport 76%
Light Truck 70% Food 74% Team Sport 71%

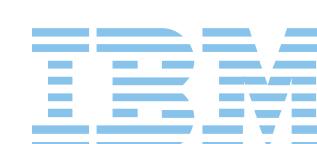
Alchemy Face Detection

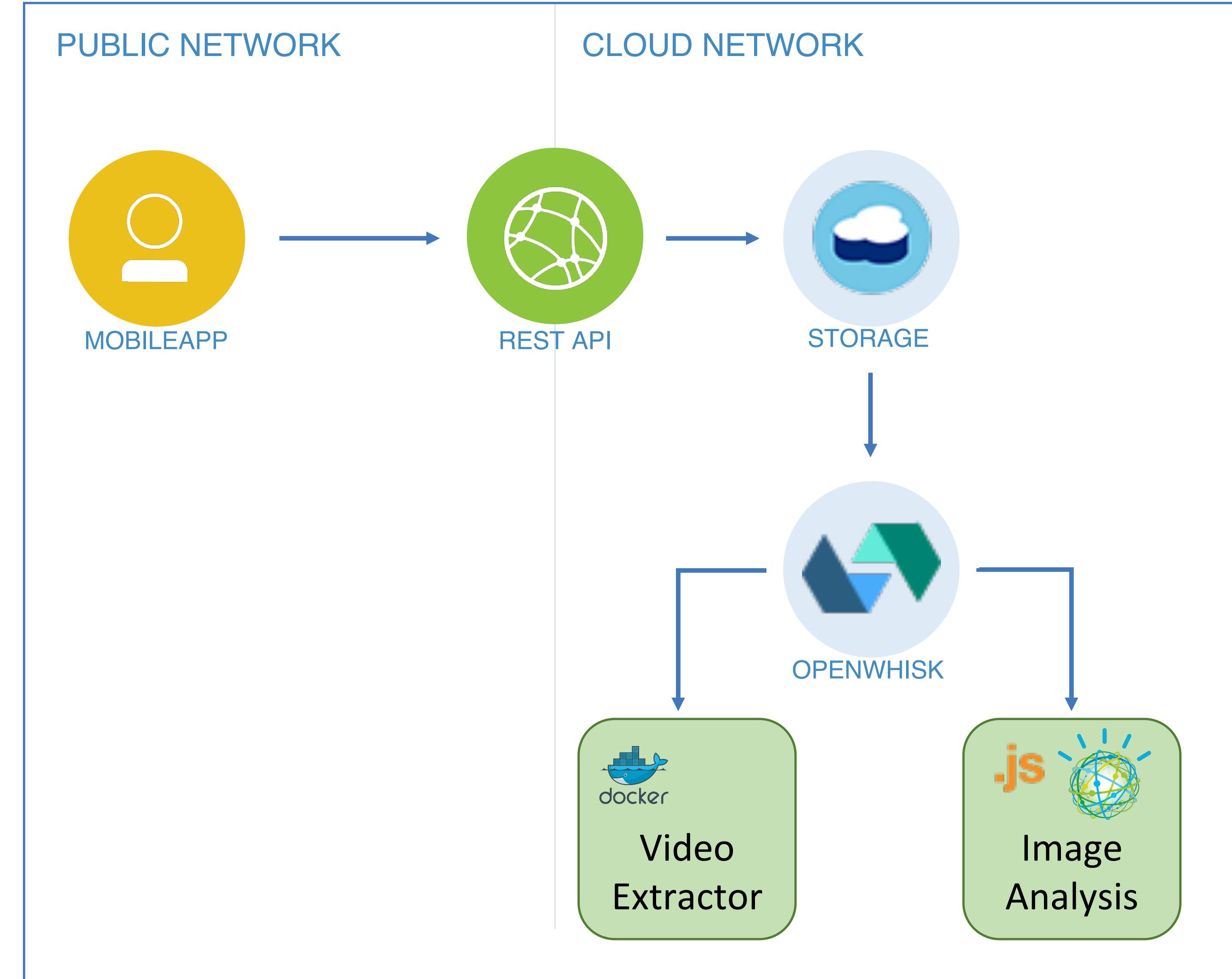
No known face detected

Alchemy Image Keywords

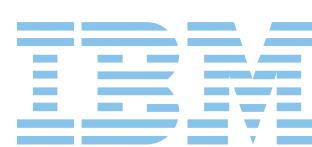
person 100% car 100% vehicle 100%
auto 97% street 62% vintage 69%
crowd 100% animal 100% mammal 98%
people 96% building 98% sky 100%
blue sky 100% eiffel 77% ship 85%
boat 100% racing 71% photobooth 85%
bus 98% cloud 77%

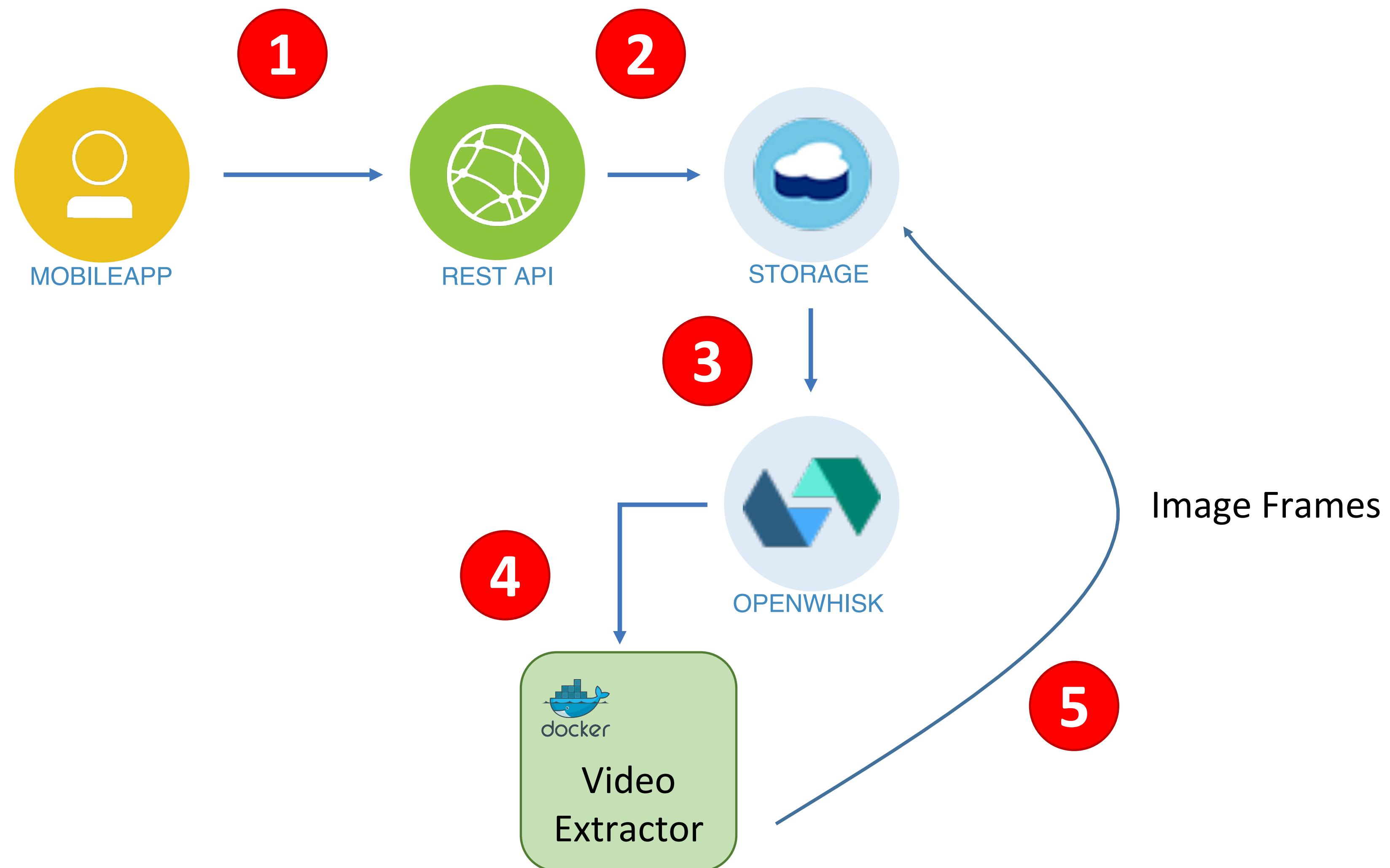
More results for other videos.



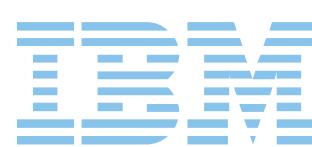


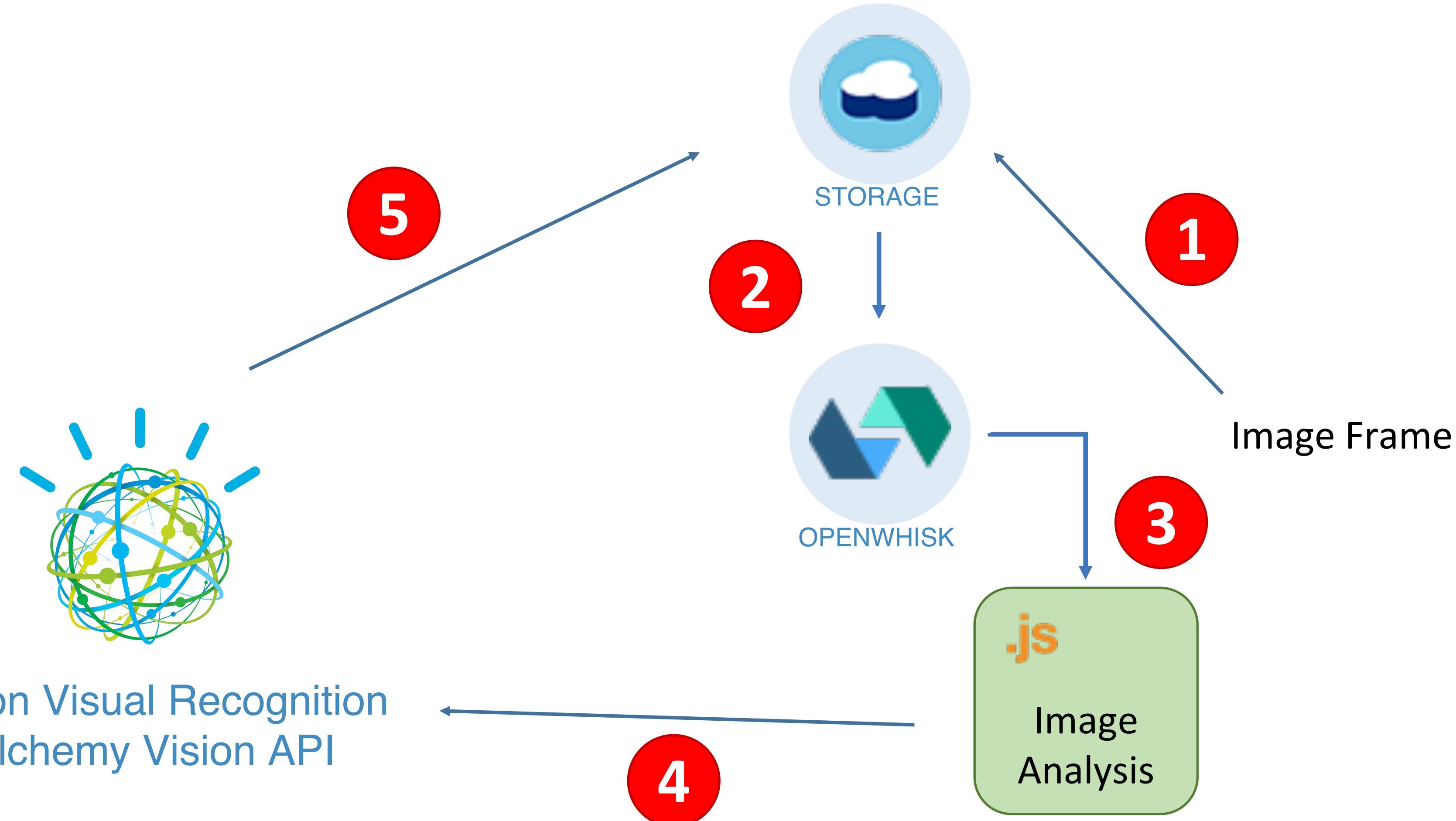
Built with IBM Bluemix, OpenWhisk, Watson





What happened: extracting Frames with a Docker action



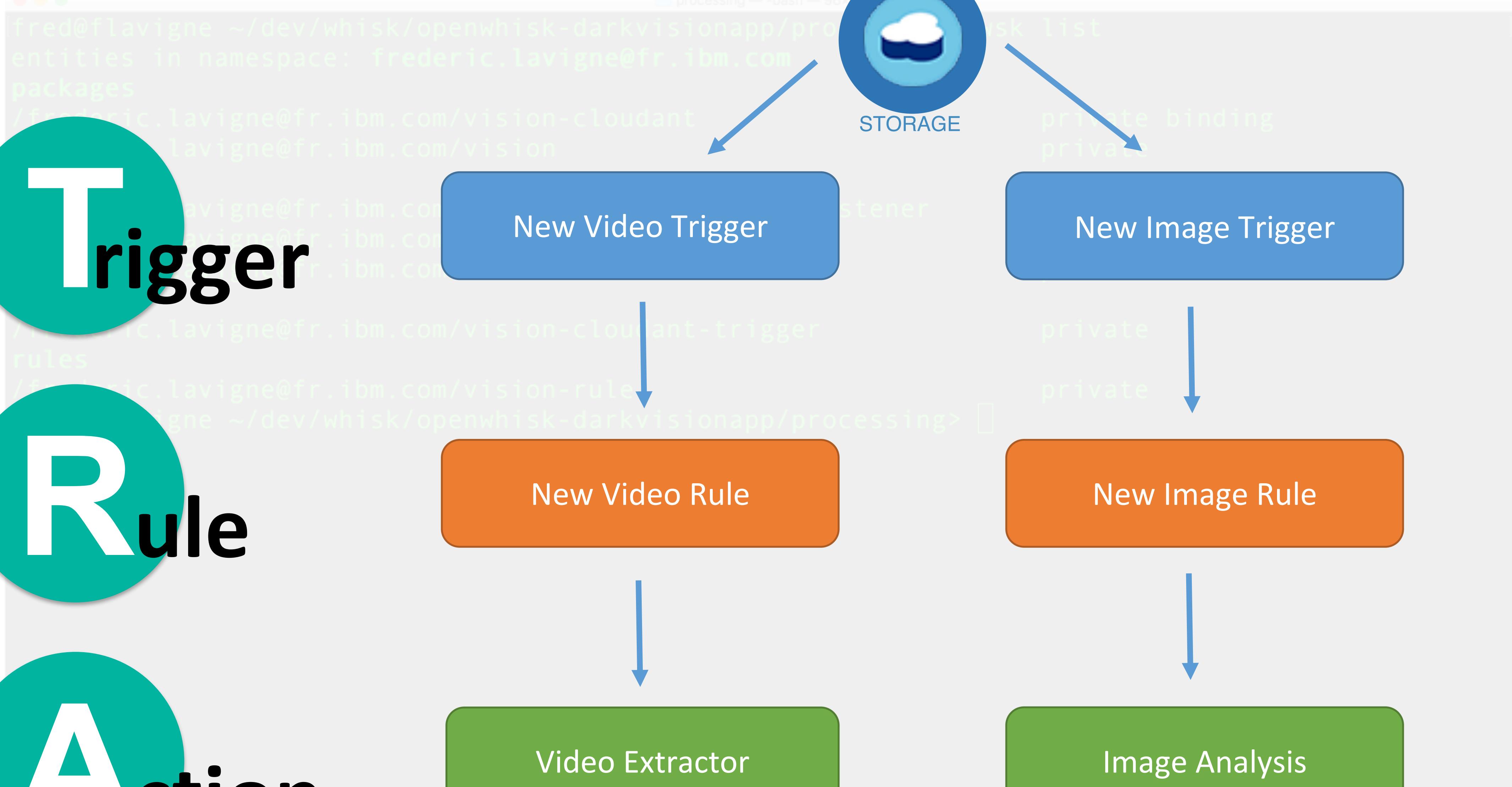


What happened: analyzing frames with a JavaScript action

Trigger

Rule

Action



OpenWhisk artifacts used by Dark Vision



No server to deploy

Single purpose polyglot actions

Event-driven

Dark Vision video processing is a good serverless citizen!



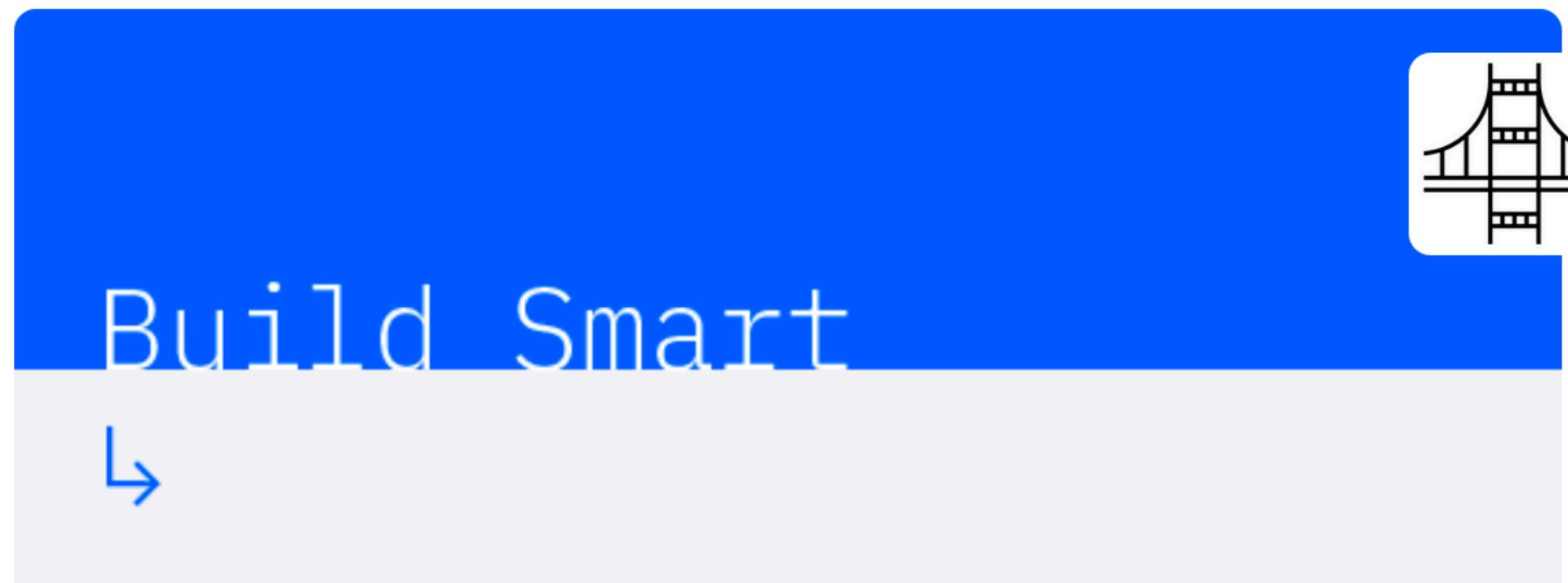
Get the code

<https://github.com/IBM-Cloud/openwhisk-darkvisionapp>



THANK YOU!

- Q&A
- The link: ibm.biz/sv-serverless-2020
- Some more examples – a real time voting app:
<https://developer.ibm.com/tutorials/build-a-real-time-voting-application/>



Part of **IBM Developer** – 40 groups [?](#)

IBM Developer SF Bay Area

📍 San Francisco, CA

👤 8,072 members · Public group [?](#)

👤 Organized by Angie K and 6 others

IBM Developer

IBM

Share: [f](#) [t](#) [in](#)