Apache OpenWhisk

Made by:
Milos Djordjevic
Natalija Pesic
Mentor:
Dragan Stojanovic

What is OpenWhisk?

An **open source** project driven by **IBM** and **Adobe**, **Apache OpenWhisk** is a robust **Functions-as-a-Service** (FaaS) platform that can be deployed in the cloud or within the data center. When compared to other **serverless** projects, OpenWhisk is a robust, scalable platform designed to support thousands of concurrent triggers and invocations.

Available as open source via Apache openwhisk.org

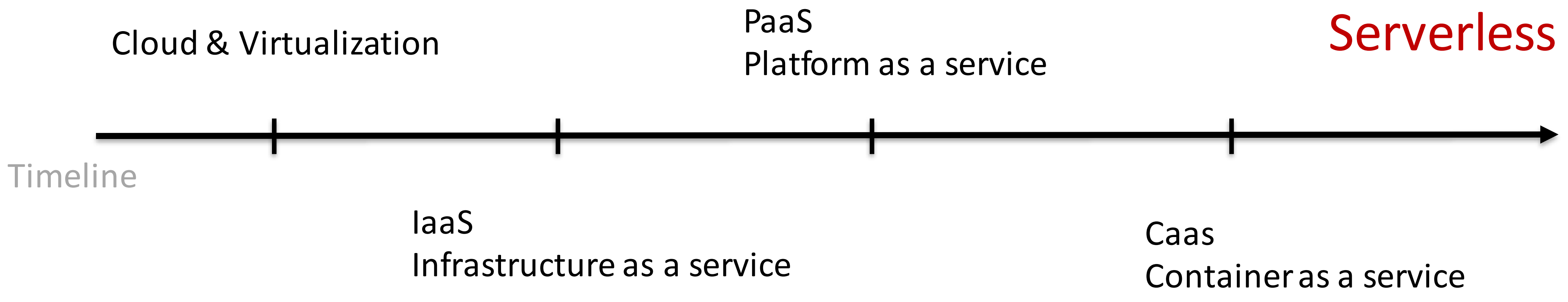
Available as managed service on **IBM Bluemix** **bluemix.net/openwhisk**

Apache OpenWhisk is a **serverless** platform ideally suited to a wide range of scenarios including cognitive, data, IoT, microservices, and mobile workloads.

Apache OpenWhisk is a **serverless**, **open source** cloud platform that executes **functions** in response to events at any scale

Apache OpenWhisk is an open source, distributed **Serverless** platform that executes functions f(x) in response to events at any scale.

Evolution of serverless



Cloud & Virtualization

IaaS
Infrastructure as a service

PaaS
Platform as a service

Caas
Container as a service

Serverless

Timeline

Serverless != No Servers

“serverless” does not mean “without a server”—it means “without managing the server"

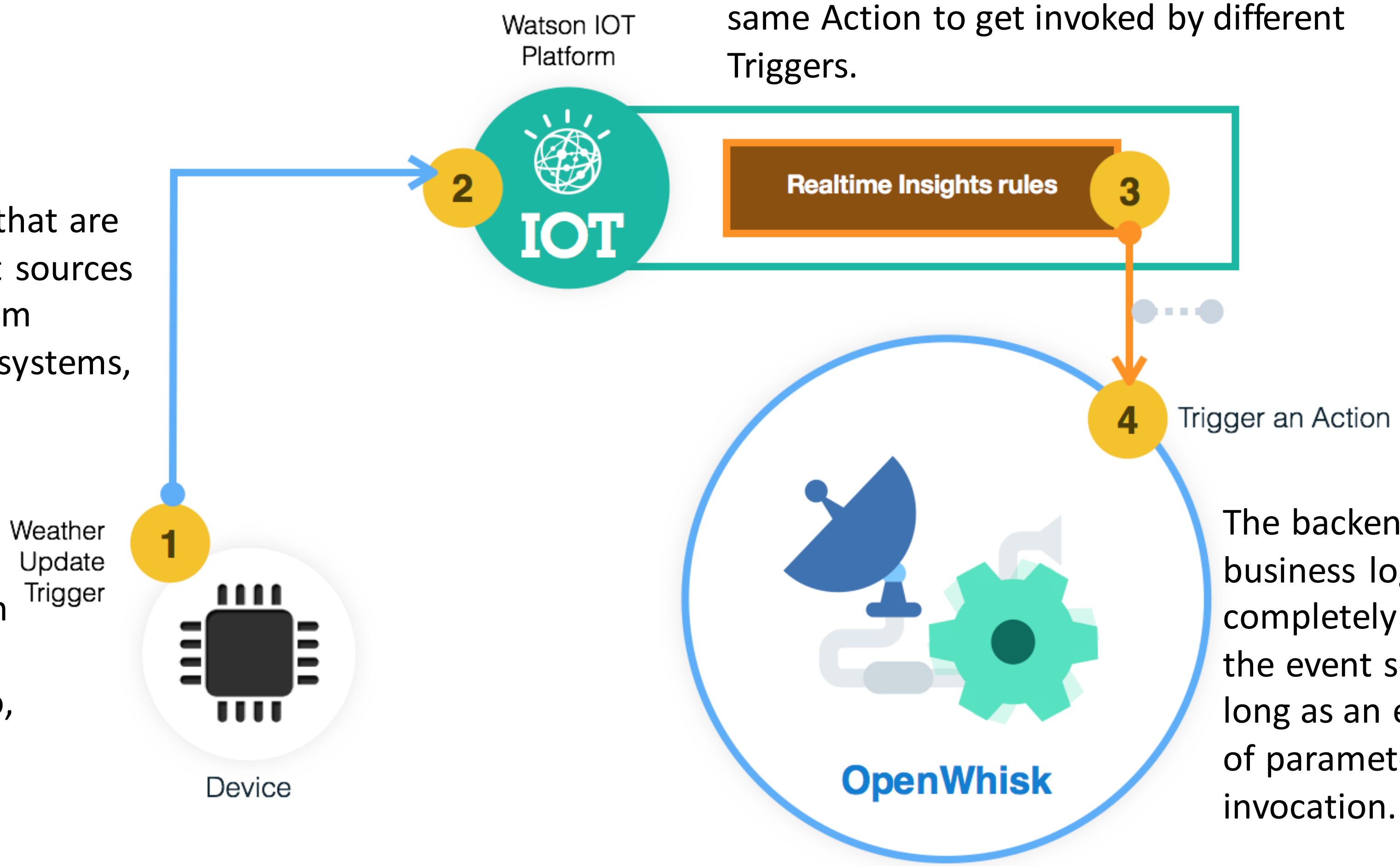
Instead of spending valuable time on designing and maintaining multi-servers scalable architecture OpenWhisk will handle it all for them. So, programmer only care about functions.

Concepts

Rules create a loosely coupled association between Triggers and Actions. This design pattern enables the same Action to get invoked by different Triggers.

Triggers, are endpoints that are explicitly called by event sources such as databases, stream processing engines, file systems, and line-of-business applications.

An external event can be anything ranging from an HTTP request to a data feed such as audio, video, text etc.



The backend functions which hold the business logic are called **Actions**. They are completely autonomous and independent of the event sources. They can be invoked as long as an event source passes the right set of parameters that are essential for the invocation.

Actions, Rules, and Triggers can be created and managed through REST endpoints. All that the event source needs to do to invoke an Action is to call the Trigger REST API.

Supported languages

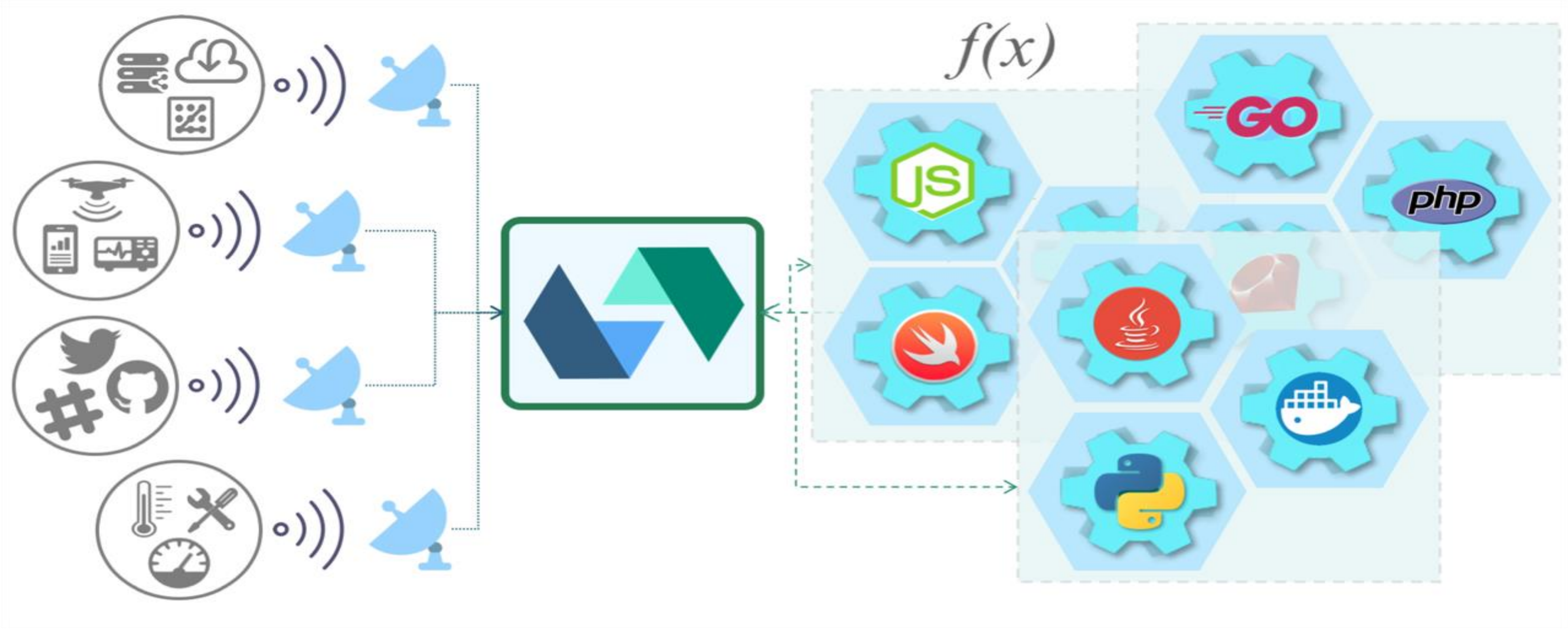
Multi-language Support

Java,
NodeJS,
Python,
PHP,
Ruby,
Go,
Scala,
Swift,
Haskell and
Docker

..and anything that runs on a Docker Container

Event Provider

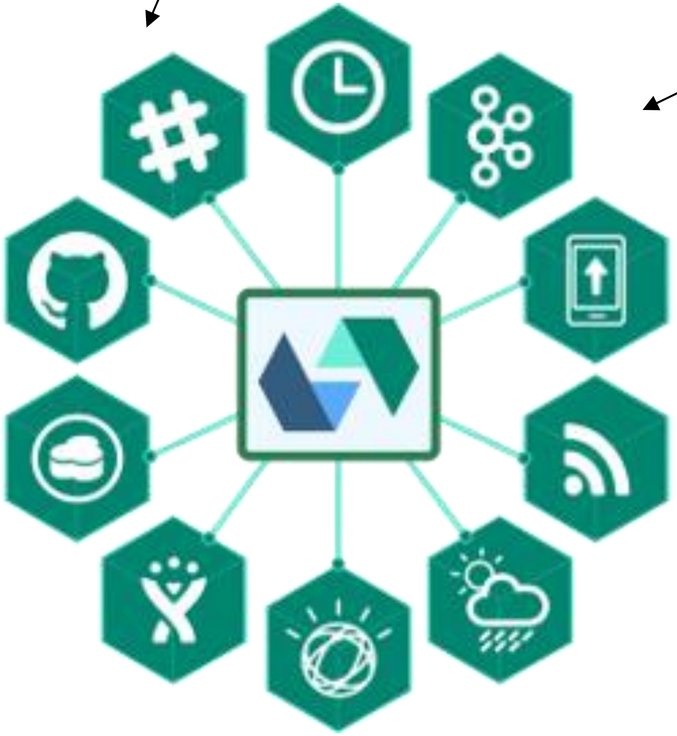
Developers can easily integrate their apps with other third-party services and frameworks that represent event providers (services that generate and provide events). OpenWhisk facilitates the integration of functions with *Kafka* queues, APIs like *IBM Watson*, backend schedulers, services such as push notifications for mobile, workplace messaging like *Slack*, code management services like *Git*.



Some Event Provider Examples

Slack – messages and meeting schedules as events

Kafka – using topics and publish for creating events



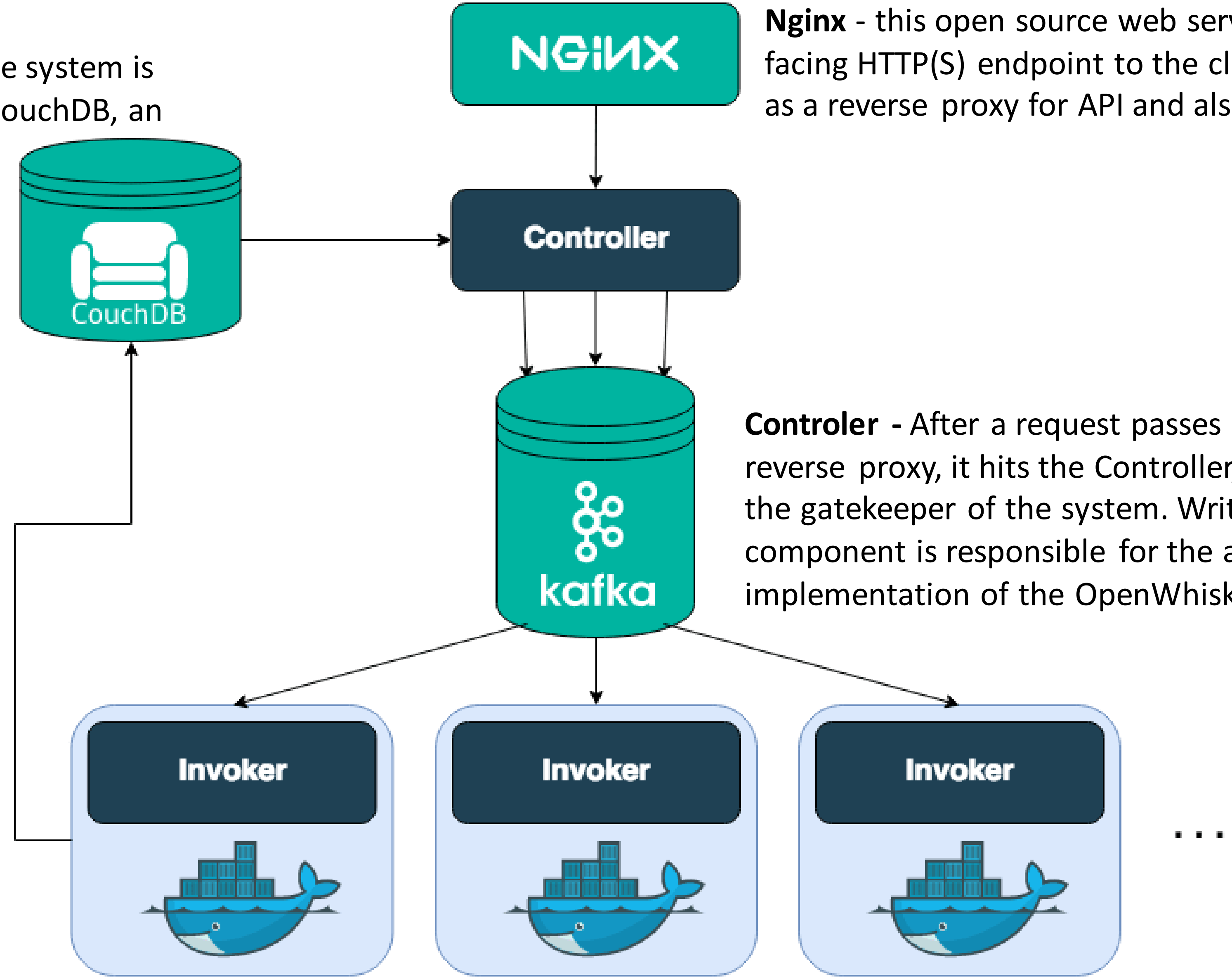
GitHub – web hook (user defined HTTP callback, for example when pushing code to repo)

Push notifications on mobile systems as events

The Building Blocks of OpenWhisk

CouchDB - The state of the system is maintained and managed in CouchDB, an open source JSON data store.

Nginx - this open source web server exposes the public-facing HTTP(S) endpoint to the clients. It is primarily used as a reverse proxy for API and also for terminating SSL.



Controler - After a request passes through the reverse proxy, it hits the Controller, which acts as the gatekeeper of the system. Written in Scala, this component is responsible for the actual implementation of the OpenWhisk API.

Kafka - Apache Kafka is typically used for building real-time data pipelines and streaming applications. OpenWhisk takes advantage of Kafka to connect Controller with Invokers.

The Building Blocks of OpenWhisk

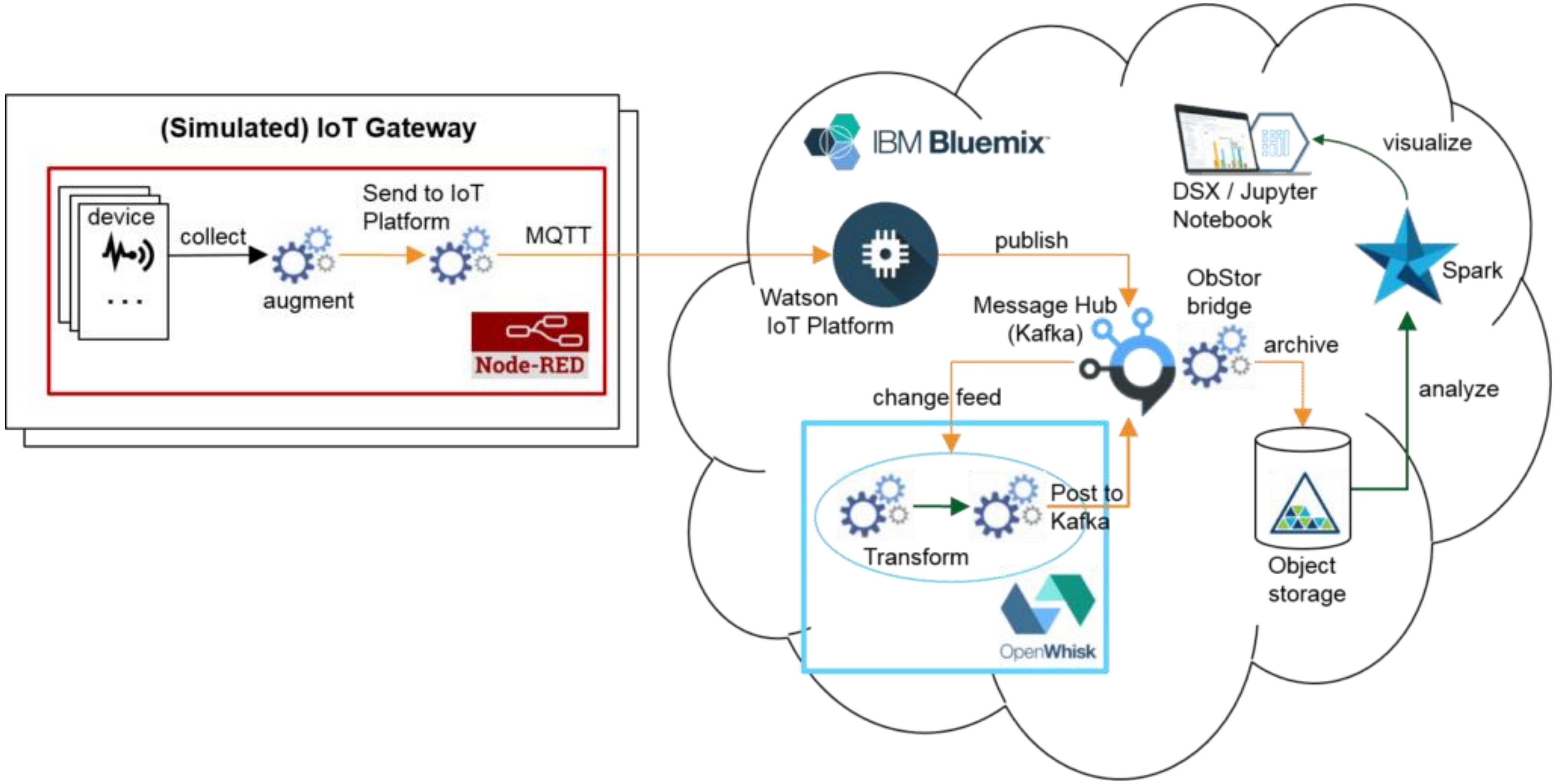
Invoker - Written in Scala. It spins up a new Docker container that acts as the unit of execution for the chosen Action. The Invoker copies the source code from CouchDB and injects that into the Docker container.

OpenWhisk uses **Consul** as the single source of truth accessible by every component of the system. It also provides service discovery capabilities making it easy for the Controller to discover the entities that will invoke an Action.

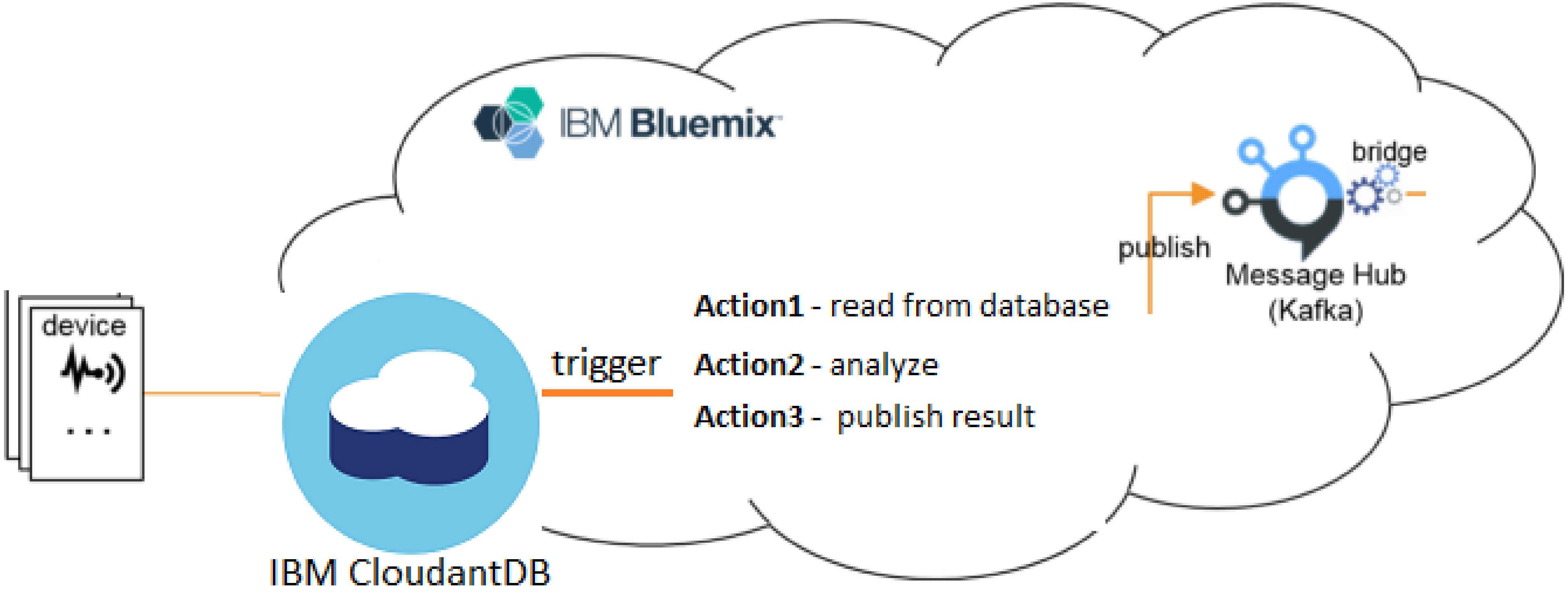
Docker
Apache OpenWhisk is built on top of proven open source technologies including Docker, which plays a very important role. Almost all the components of OpenWhisk are packaged and deployed as containers. From Nginx to Kafka to Consul, everything in the platform runs as a container.
Refer to the available images on Docker Hub for a list of OpenWhisk container images.

What is IBM Bluemix OpenWhisk?

IBM BlueMix OpenWhisk is the FaaS Functions as a Service programming platform of IBM based on the open source serverless project OpenWhisk.



Demo app



Action1 - read from database

Action2 - analyze

Action3 - publish result

Sources:

- https://medium.com/openwhisk/openwhisk-for-a-smart-city-data-application-dccd7894e0e1
- https://www.8bitmen.com/what-is-apache-openwhisk-why-use-it-everything-you-should-know-about-it/
- https://thenewstack.io/behind-scenes-apache-openwhisk-serverless-platform/
- https://www.oreilly.com/library/view/learning-apache-openwhisk/9781492046158/ch01.html
- https://openwhisk.apache.org/
- https://www.youtube.com/watch?v=5bOnGxgda1I