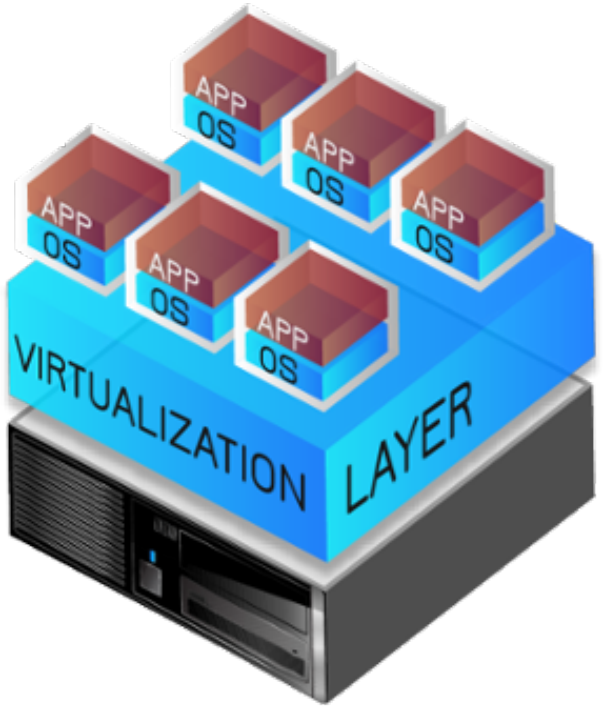
Virtualization, Cloud and Serverless Computing

Outline

- Virtualization
  - Hardware Virtualization
- Cloud Computing
  - PaaS, IaaS
- Serverless Computing



VIRTUALIZATION

Virtualization

- “In computing, virtualization refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources.” Wikipedia

Many Forms of Virtualization

- Server Virtualization
- Storage Virtualization
- Network Virtualization
- Desktop Virtualization
- I/O Virtualization

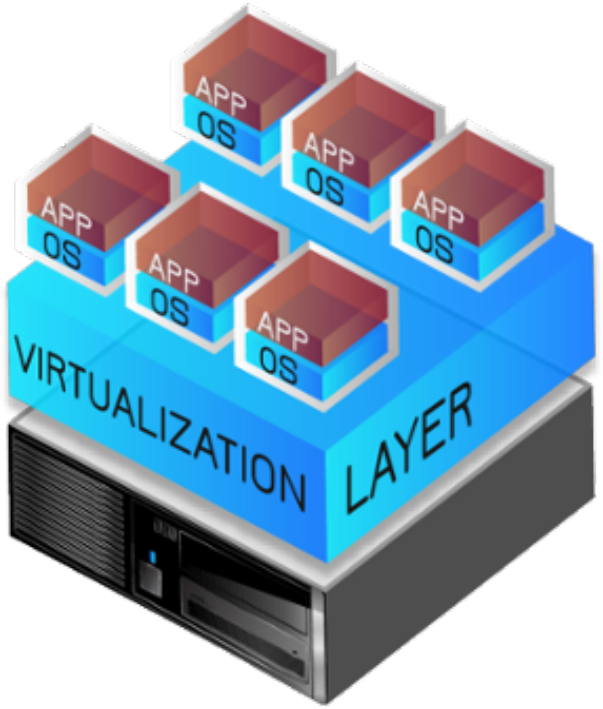
Why Virtualizations?

- Physical Servers: One OS/ Application per server
  - High up-front costs
  - High ongoing costs
  - Most of the server resources are wasted

Source: Wikipedia

What is Server Virtualization?

- The OS is abstracted away from hardware
- Multiple OS and Applications on one Server runs on the virtual layer, also known as Hypervisor



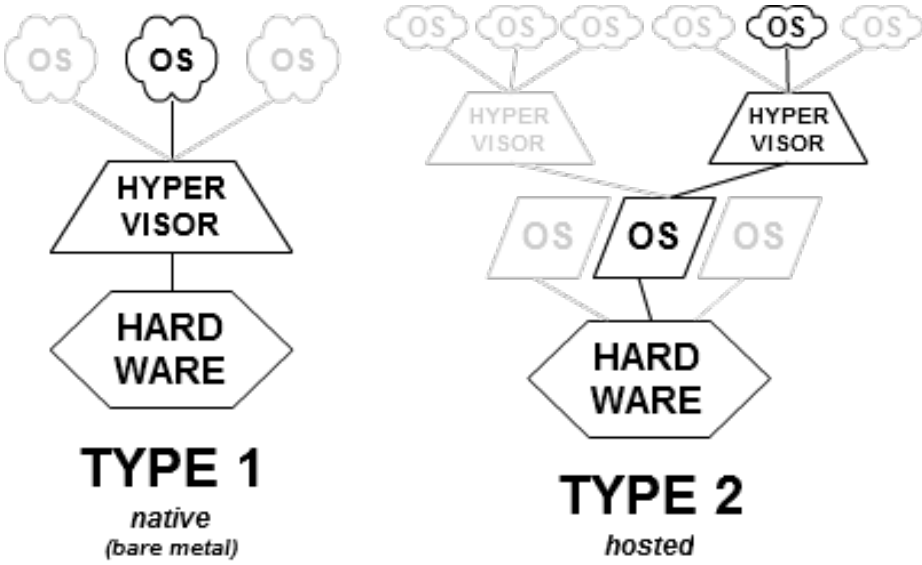
Source: stepupitservices

Hypervisor

- Hypervisor
  - Creates the virtualization layer that makes server virtualization possible
  - Contains the Virtual Machine Monitor (VMM)
- Example of Hypervisors
  - KVM
  - Oracle VirtualBox

Type 1 vs. Type 2 Hypervisor

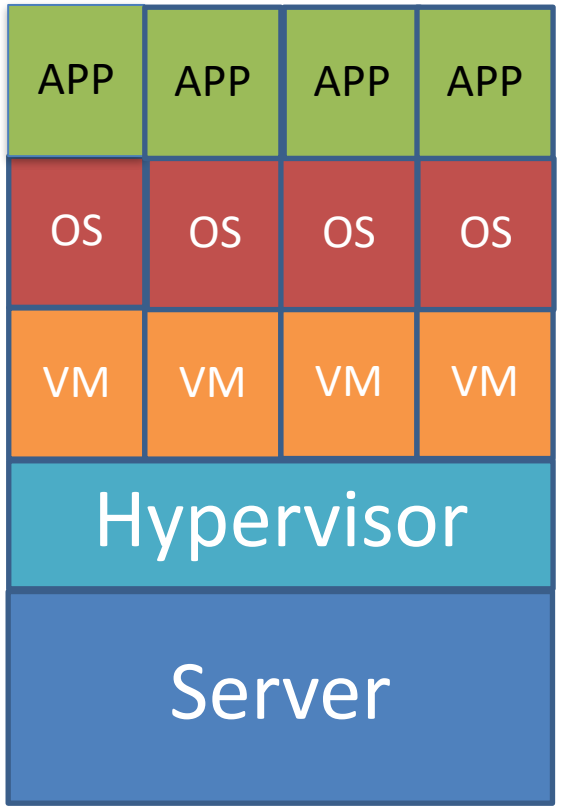
- Type 1 Hypervisor is loaded directly on the hardware (native or bare-metal)
  - Microsoft Hyper-V
  - VMware ESX/ESXi
  - KVM
- Type 2 is loaded on an OS running on the hardware (hosted hypervisor)
  - Oracle VirtualBox
  - VMware Workstation



Source: Wikipedia

Disadvantages of Hypervisors

- The Server's resources are shared among VMs
- Execution of OS consumes resources (CPU/RAM/Disk) which is independent of the app
- OS can be expensive and requires administration



Overview Virtualization

- Server Virtualization most popular form of virtualization
- Type 1 and Type 2 Hypervisors

CLOUD COMPUTING

What is Cloud Computing?

- A computing service you traditionally did local, now the service is performed remotely (off-premises)
- Cloud computing is an approach to computing that leverages the efficient pooling of an on-demand, self-managed, virtual infrastructure

Google Cloud Platform

Advantages of Cloud Computing

- Fast and efficient access to resources that clients actually need
- Pay for what you use
- No initial capital expenditure
- Less need for a big administrator team
- Self-maintenance and fault tolerance resources

Disadvantages of Cloud Computing

- Clients need to trust the cloud providers
- Possible limited access to your data
- Possible conflicts with government regulations and restrictions
- Potential data loss
- Locked in within the cloud provider's specification
- You may encounter unknown costs (check SLA)
- Know who you are dealing with!

Evolution of Cloud Computing

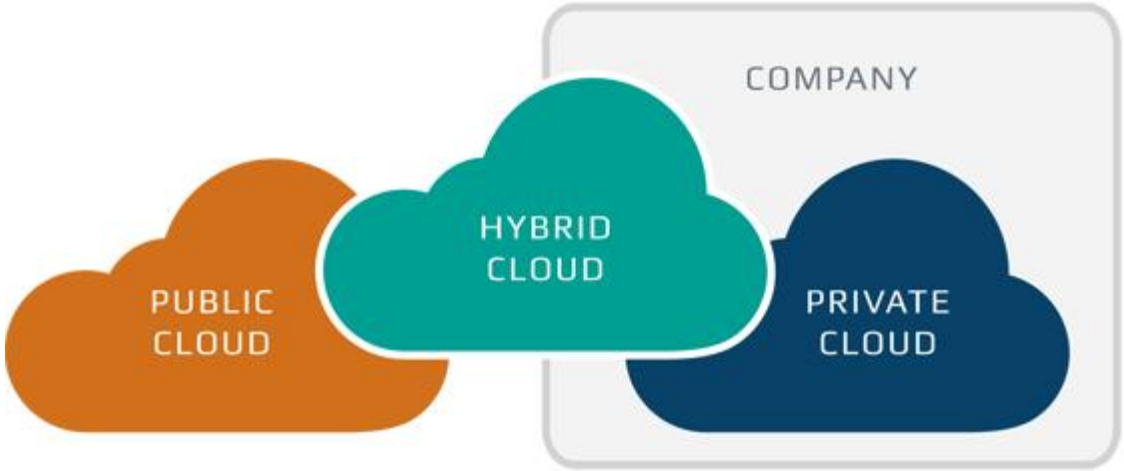
- The 1950s – Mainframe Computing
- 1960 – 1990
  - Internet, VMs, VPNs
- 1999 – Salesforce, the first SaaS
- 2006 – Amazon Web Services
- 2008 – Microsoft Azure
- 2013 – Google Compute Engine
- 2014 – IBM Bluemix

Cloud Comes in Many Shapes!

- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)
- Function-as-a-Service (FaaS)
- Database-as-a-Service (DBaaS)
- Everything-as-a-Service (*aaS, XaaS)

Infrastructure as a Service (IaaS)

- VMs offered as a service
  - Private Clouds run on your premises (e.g., OpenStack)
  - Public Clouds runs over the internet (e.g., AWS)
  - Hybrid Cloud is not a real cloud. It is the cooperation between public and private cloud



Source: Tek-nology Solutions

Virtualization vs. Private Cloud

- Virtualization is required for cloud computing
- Virtualization provides scalability, fault-tolerance, high availability and load balancing
- Private clouds are built on top of virtualizations
- Private clouds provide abstraction of resources, secure multi-tenancy, better separation of concerns

Platform-as-a-Service (PaaS)

- Clients do not need to manage storage, network, OS, database, etc.
- Cloud provider install all the dependencies client need
  - Google App Engine
  - AWS Elastic Beanstalk
  - MS Azure App Service
  - Heroku

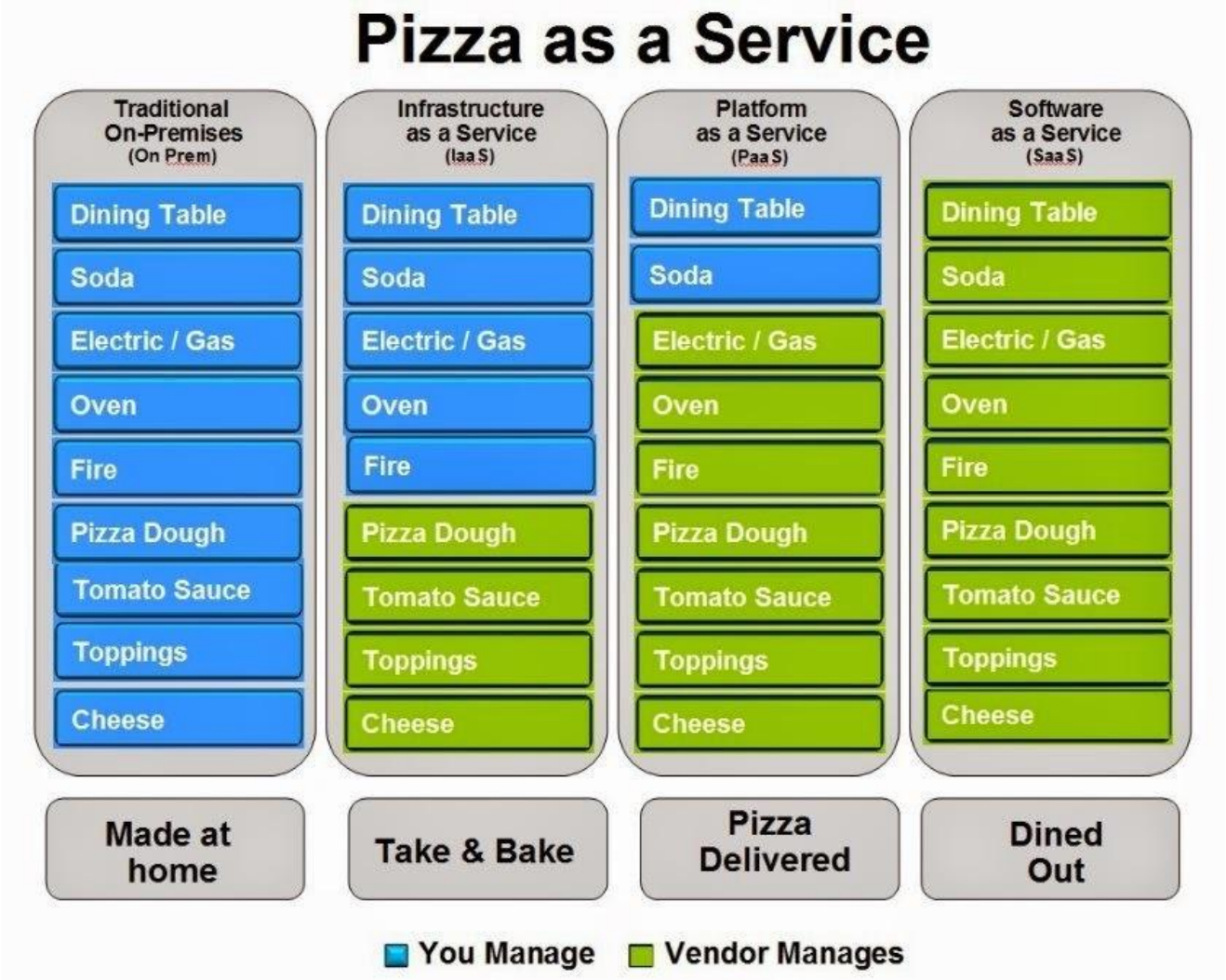
IaaS vs PaaS

- IaaS is simply a VM with some OS installed. Clients are required to install all the packages they need
- PaaS builds upon IaaS. Clients have access to the previously installed and configured resources
- PaaS is suitable solution for fast and easy application design and development

Software-as-a-Service (SaaS)

- No hardware or software that clients need to install, configure and maintain
- In many cases more affordable than similar options
- Clients have fast access to most recent patches and features
  - Google Docs
  - Salesforce
  - Gmail

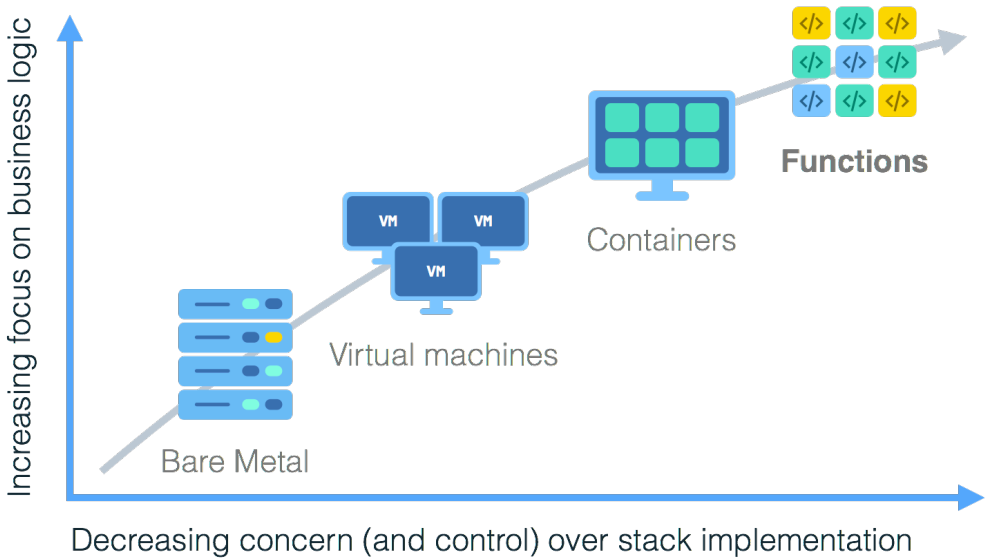
Pizza-as-a-Service



Source: Albert Barron

Overview Cloud Computing

- Cloud computing is using the managed remote computational resources over the network
- IaaS, PaaS, SaaS, XaaS

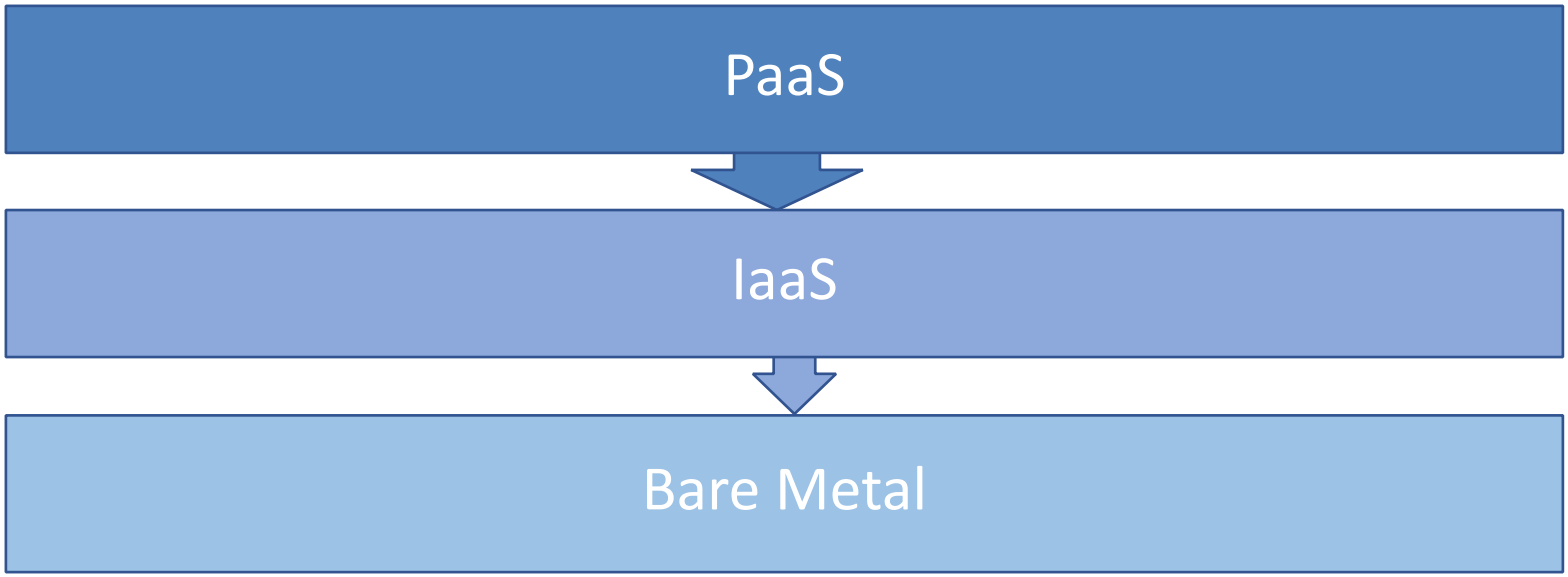


SERVERLESS COMPUTING

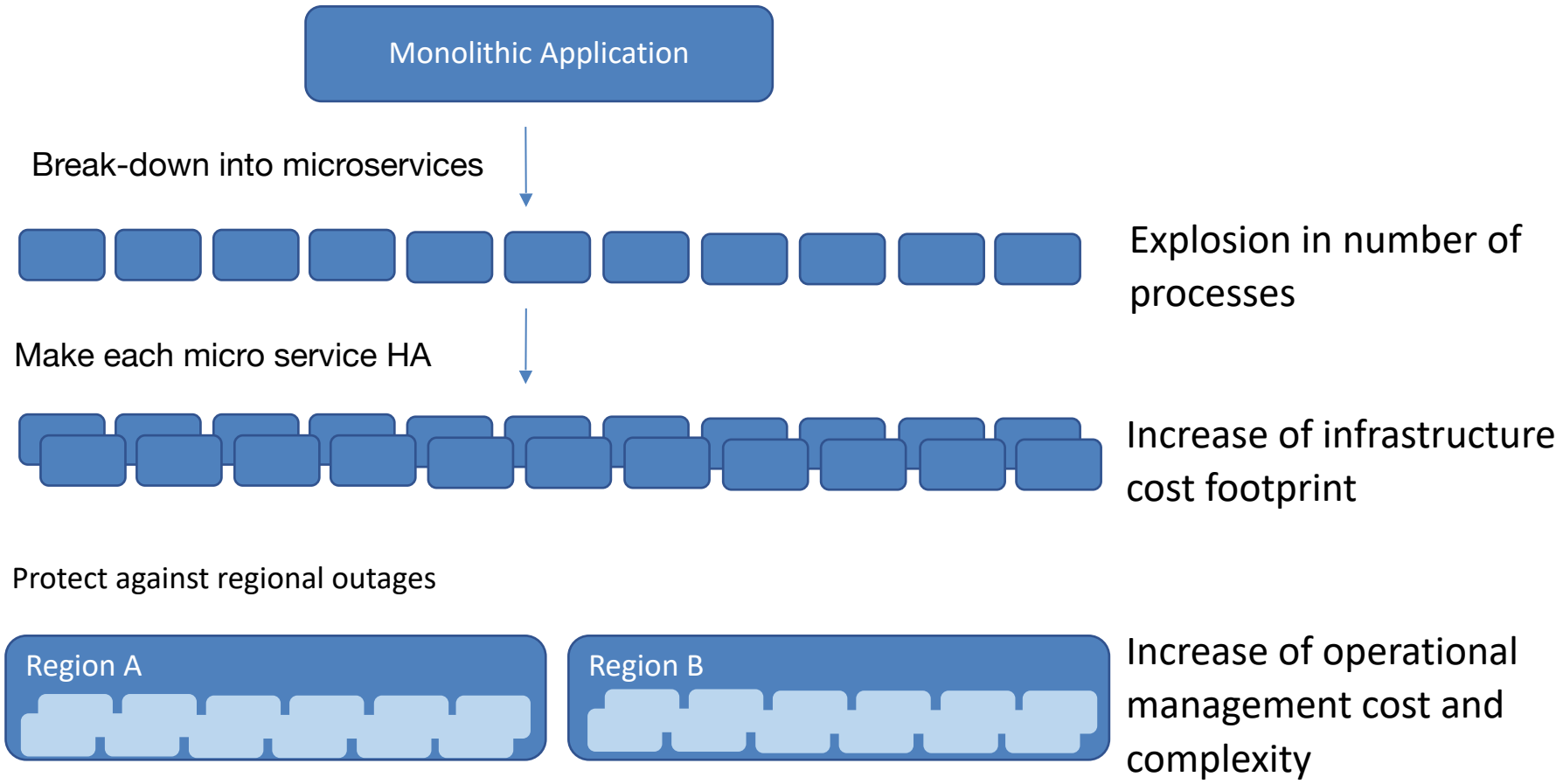
Thanks to Paul Castro, Vatche Ishakian, Vinod Muthusamy and Aleksander Slominski @IBM Research

Evolution of Serverless

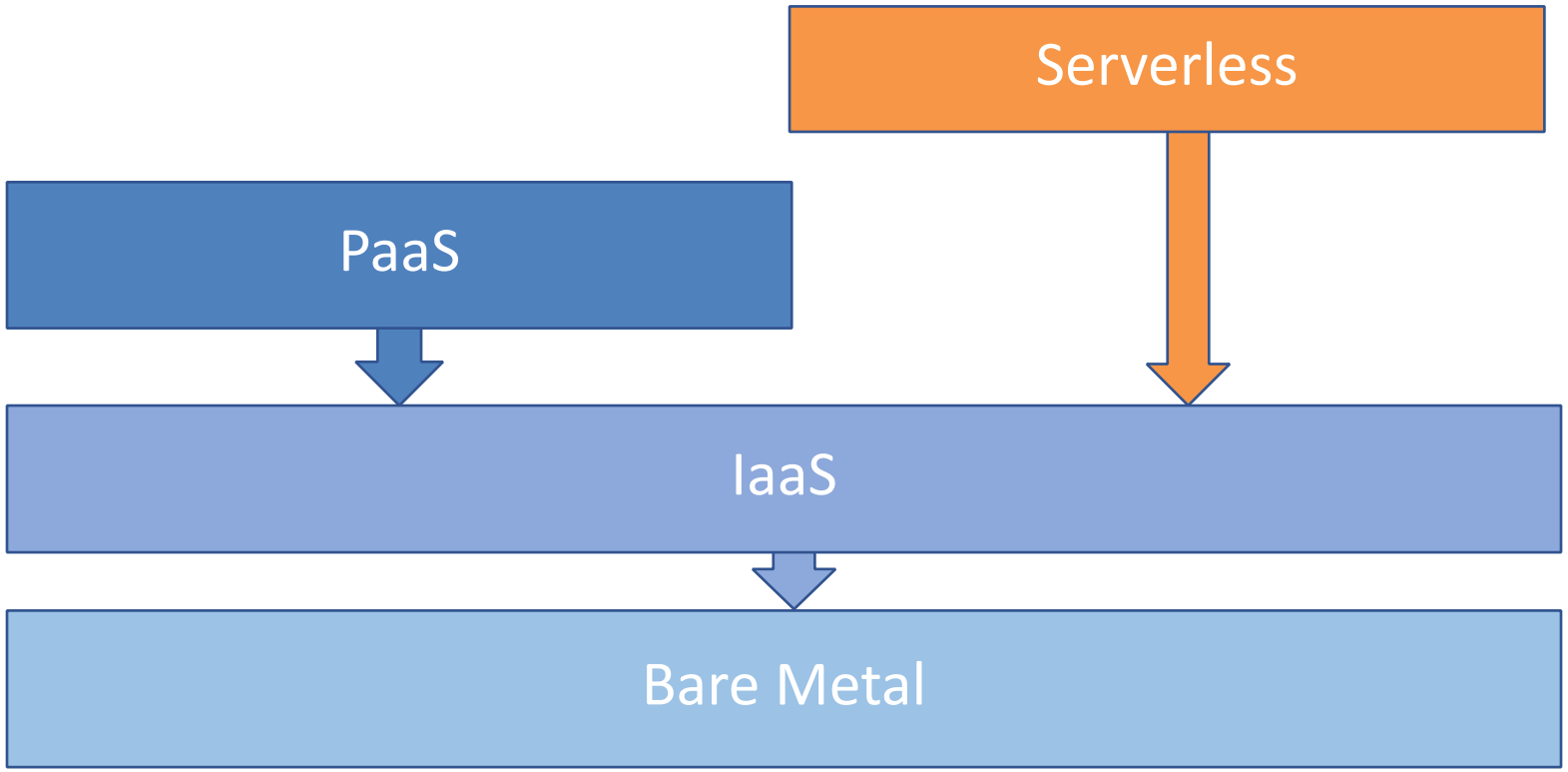
- We have the cloud, Why Serverless computing?



Evolution of Serverless



Serverless Position



What is Serverless?

- A cloud-native platform for short-running, stateless computation, and event-driven applications which scales up and down instantly and automatically and charges for actual usage at millisecond granularity
  - Auto-scalability and maintenance
  - Pay for what you use
- Serverless does not mean no servers, means worry-less servers
- Also known as Function-as-a-Service (FaaS)

What is Serverless Good For?

- Serverless is a good solution for short-running stateless event-driven operations
  - Microservice
  - Mobile Backends
  - Bots, ML Inferencing
  - IoT
  - Modest Stream Processing
  - Service Integration

What is Serverless NOT Good For?

- Serverless is not good for long-running stateful computationally heavy operations
  - Databases
  - Deep Learning Training
  - Heavy-Duty Stream Analytics
  - Spark/Hadoop Analytics
  - Video Streaming
  - Numerical Simulations

Serverless Platforms

- All major cloud providers offer serverless platforms

AWS Lambda

OpenLambda

fission

Kubernetes

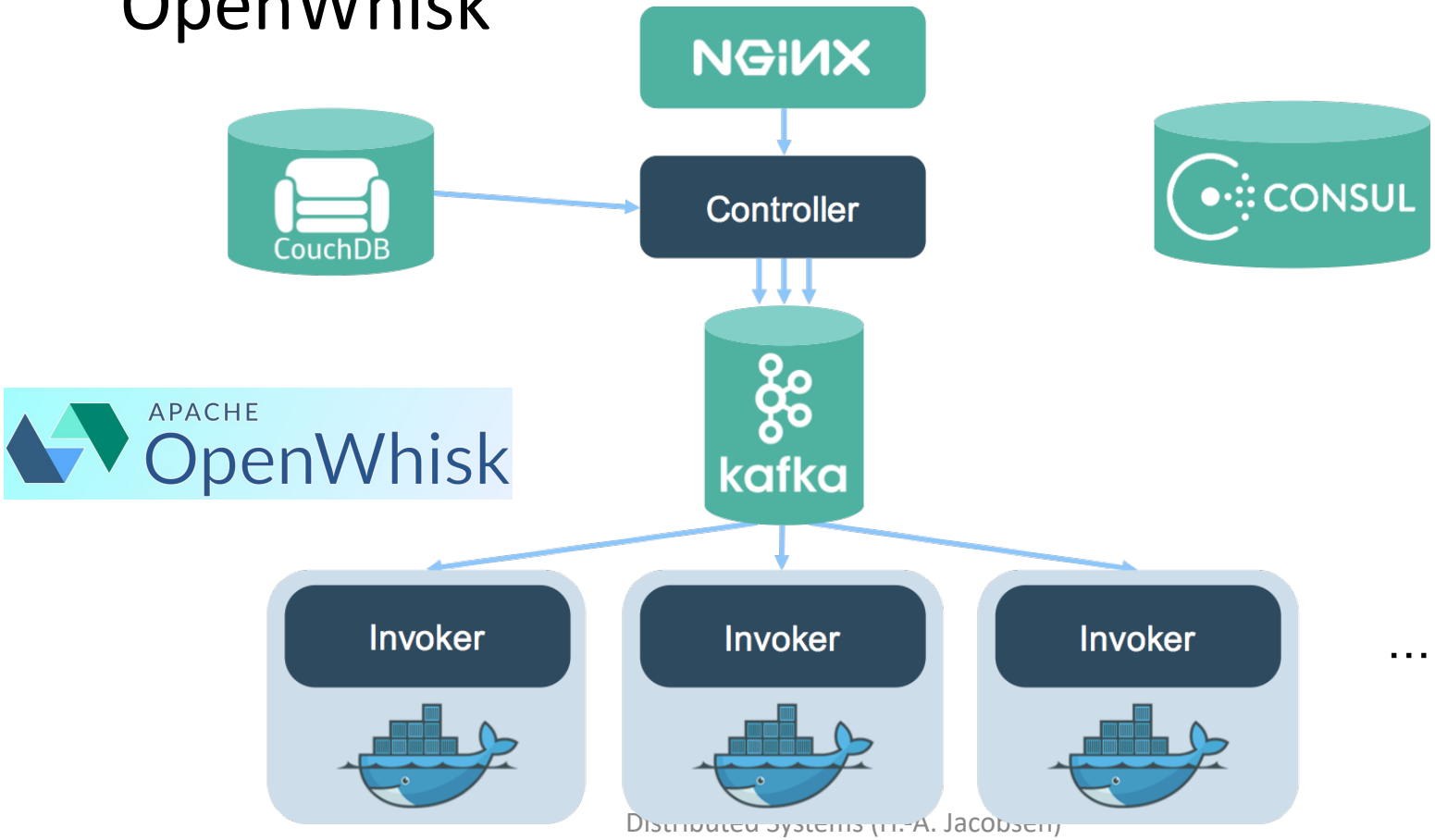
Iron.io

Azure Functions

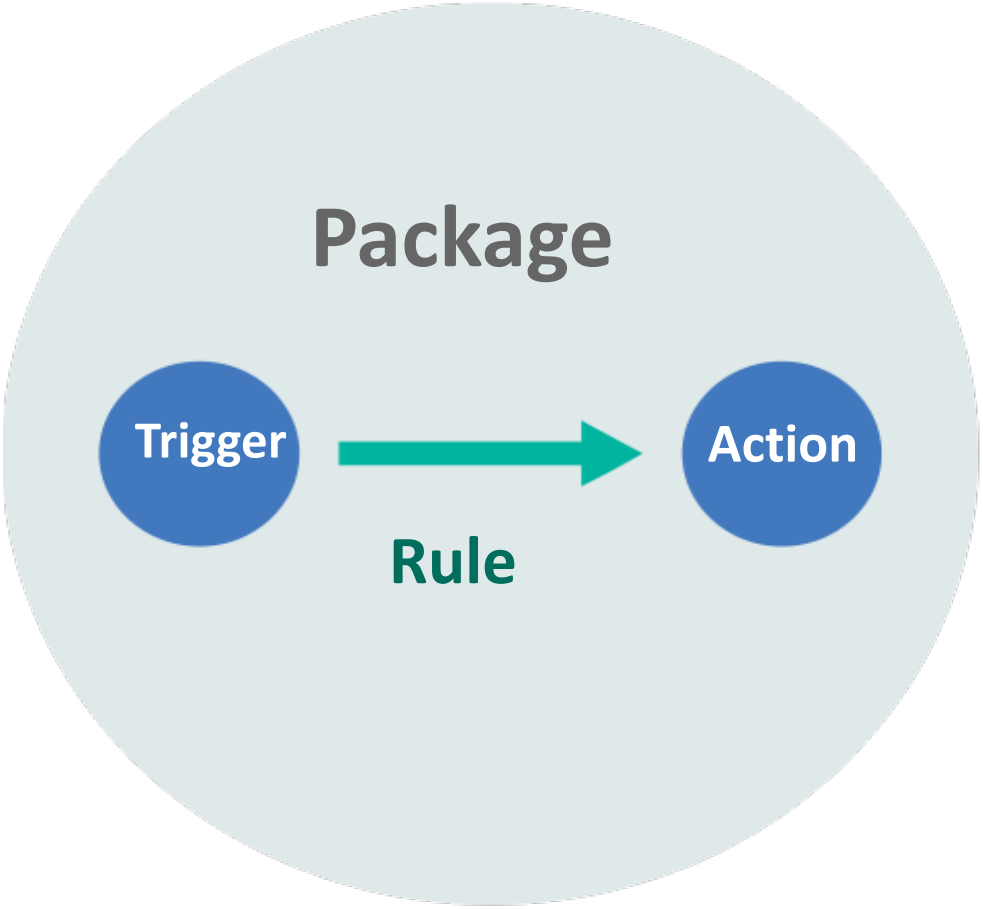
Google Functions

Apache OpenWhisk Serverless Architecture

- IBM Cloud Functions is implemented based on OpenWhisk



OpenWhisk Programming Model



Serverless Actions

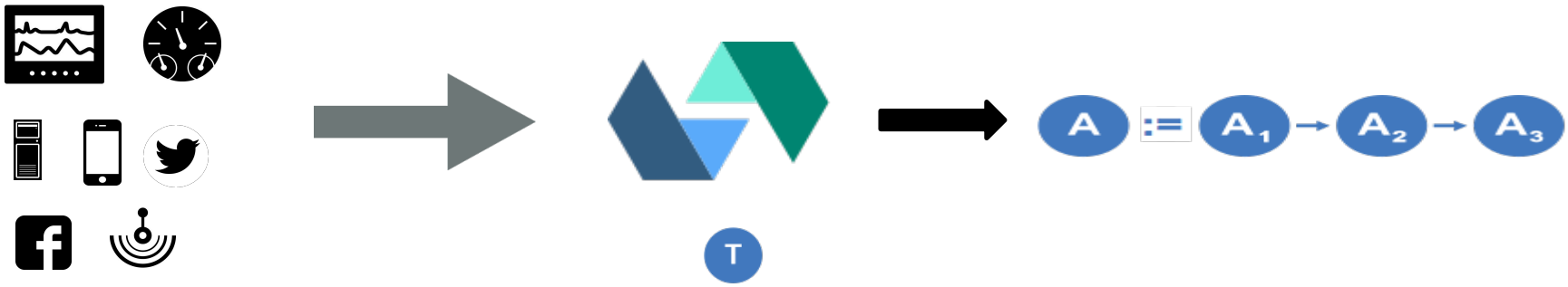
- Action is a stateless function that is execution in response to an event


```
function main(params) {
    console.log("Hello " + params.name);
    return { msg: "Goodbye " + params.name) };
}
```


```
def lambda_handler(event, context):
    print("hello world")
```

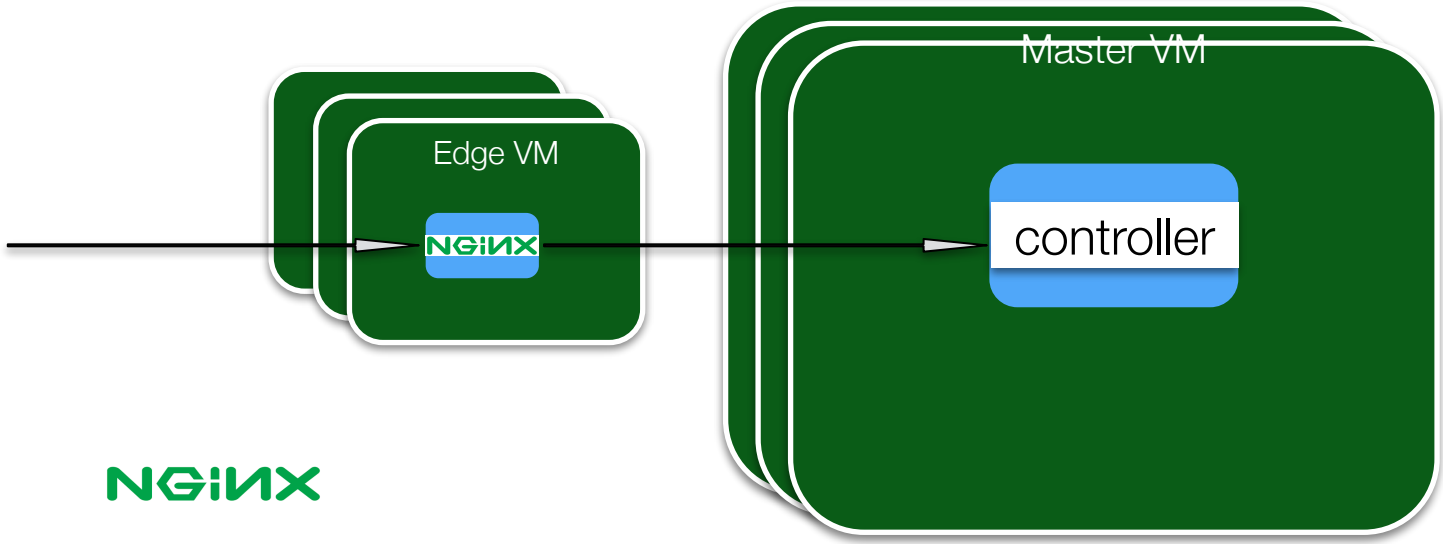
Triggers, Rules and Sequences

- An event triggers the execution of an action or a sequence of actions
- A Rule maps a a trigger to an action

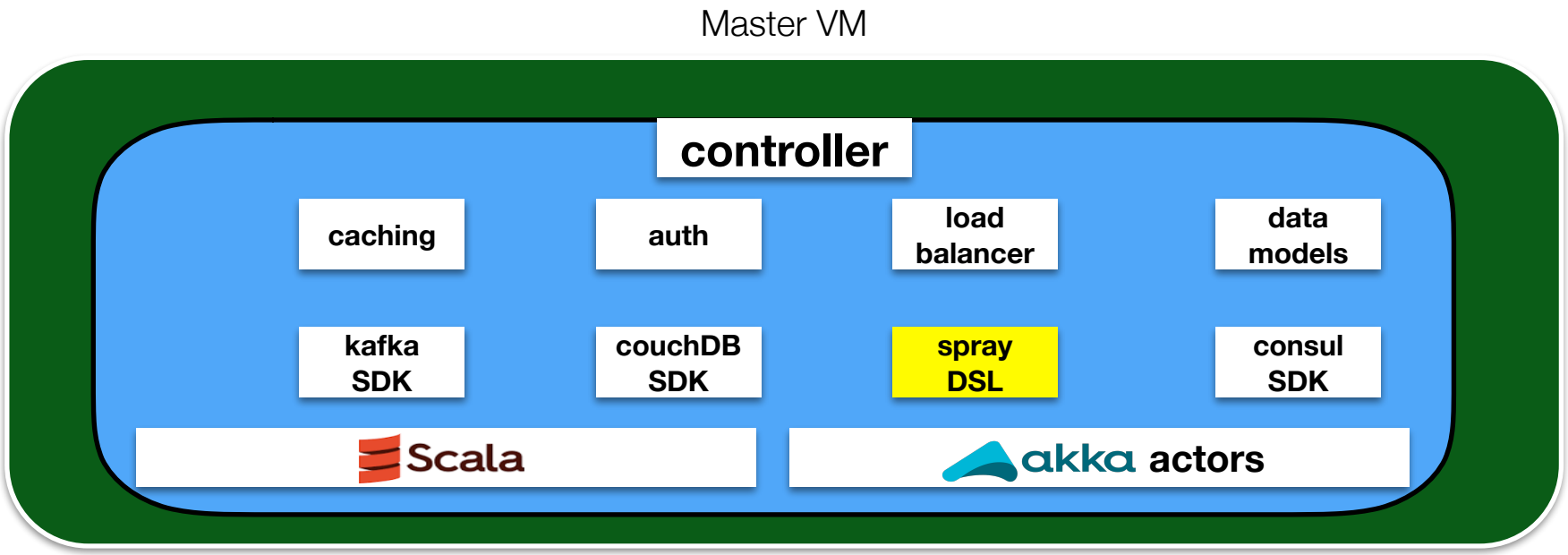


OpenWhisk: Step 1. Entering the system

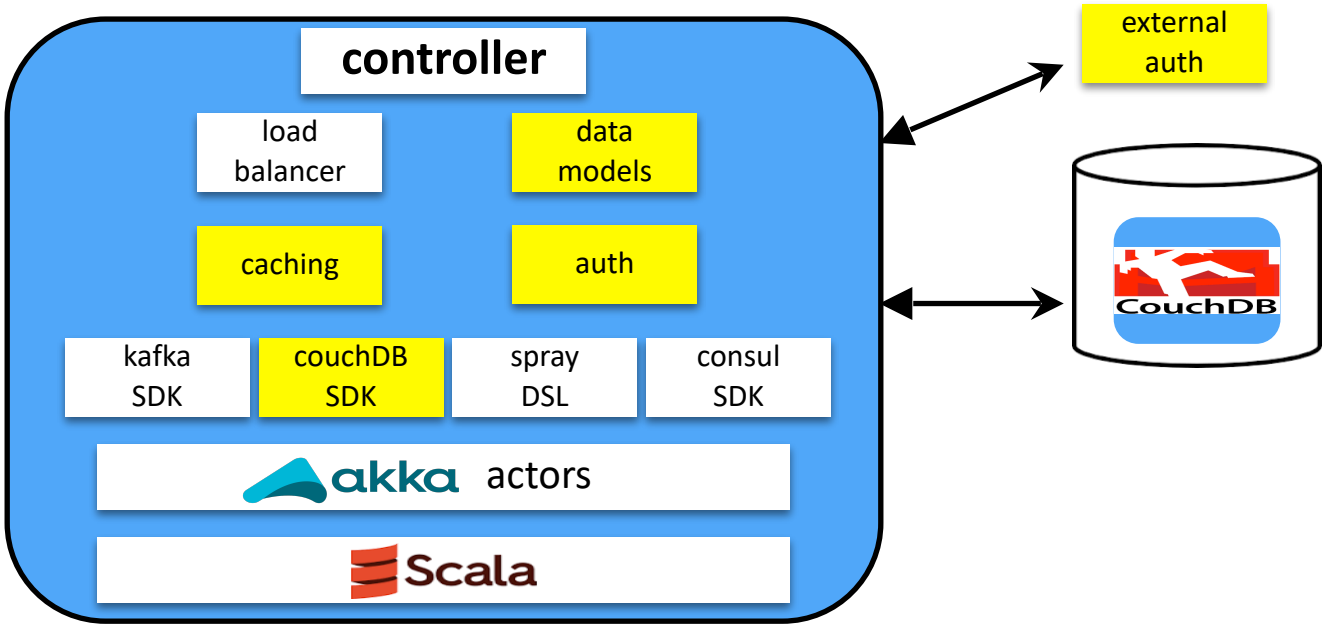
```
POST /api/v1/namespaces/myNamespace/actions/myAction
```



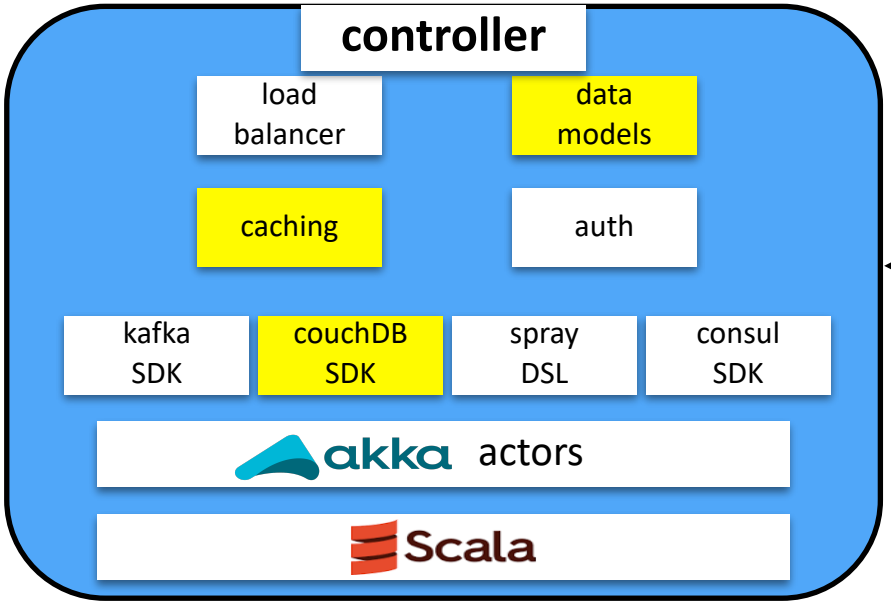
OpenWhisk: Step 2. Handle the request

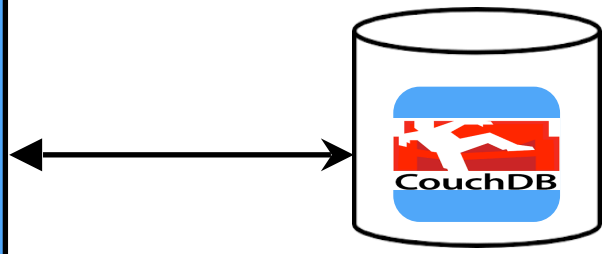


OpenWhisk: Step 3. Authentication + Authorization



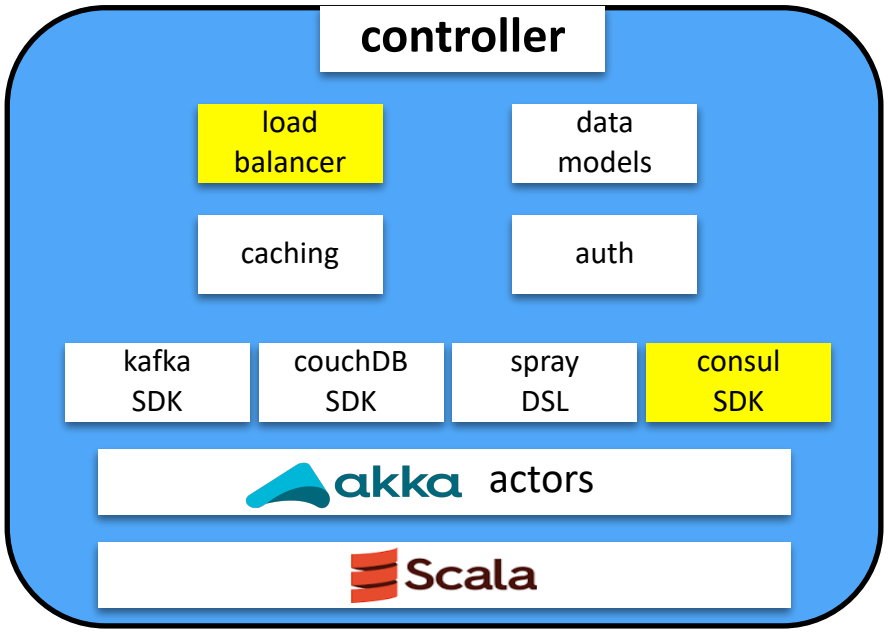
OpenWhisk: Step 4. Get the action

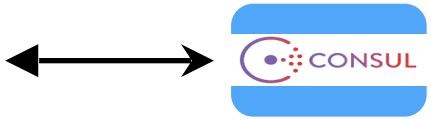




- Actions stored in CouchDB

OpenWhisk: Step 5. Looking for a home



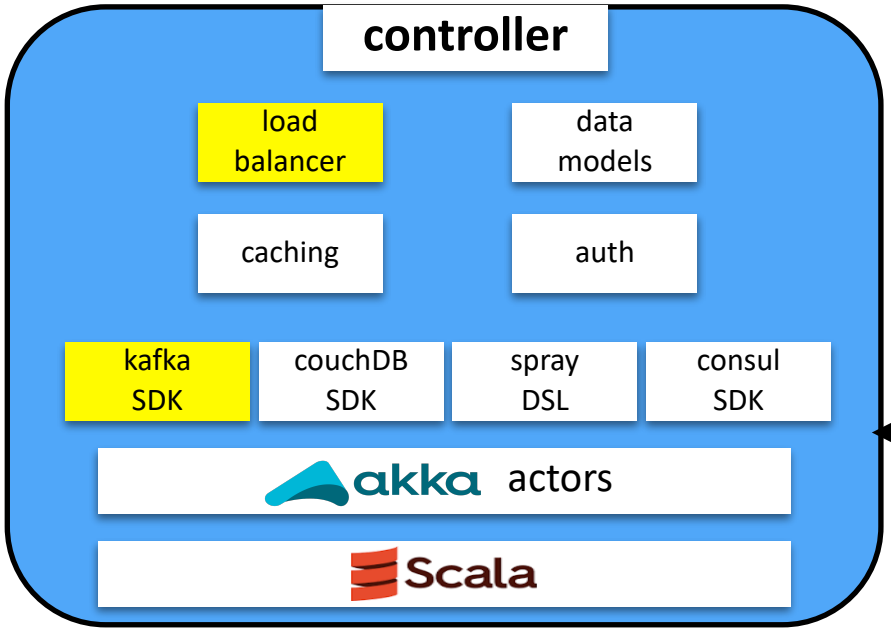


Why CONSUL *?*

- Sequentially consistent KV store
- Replication, Fault Tolerance
- Health Check / Monitoring utilities

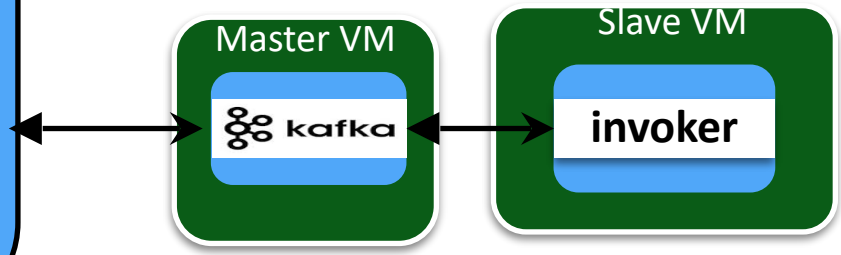
- Load balancer: find a slave to execute
- Slave health, load stored in Consul

OpenWhisk: Step 6. Get in line!



Why kafka *?*

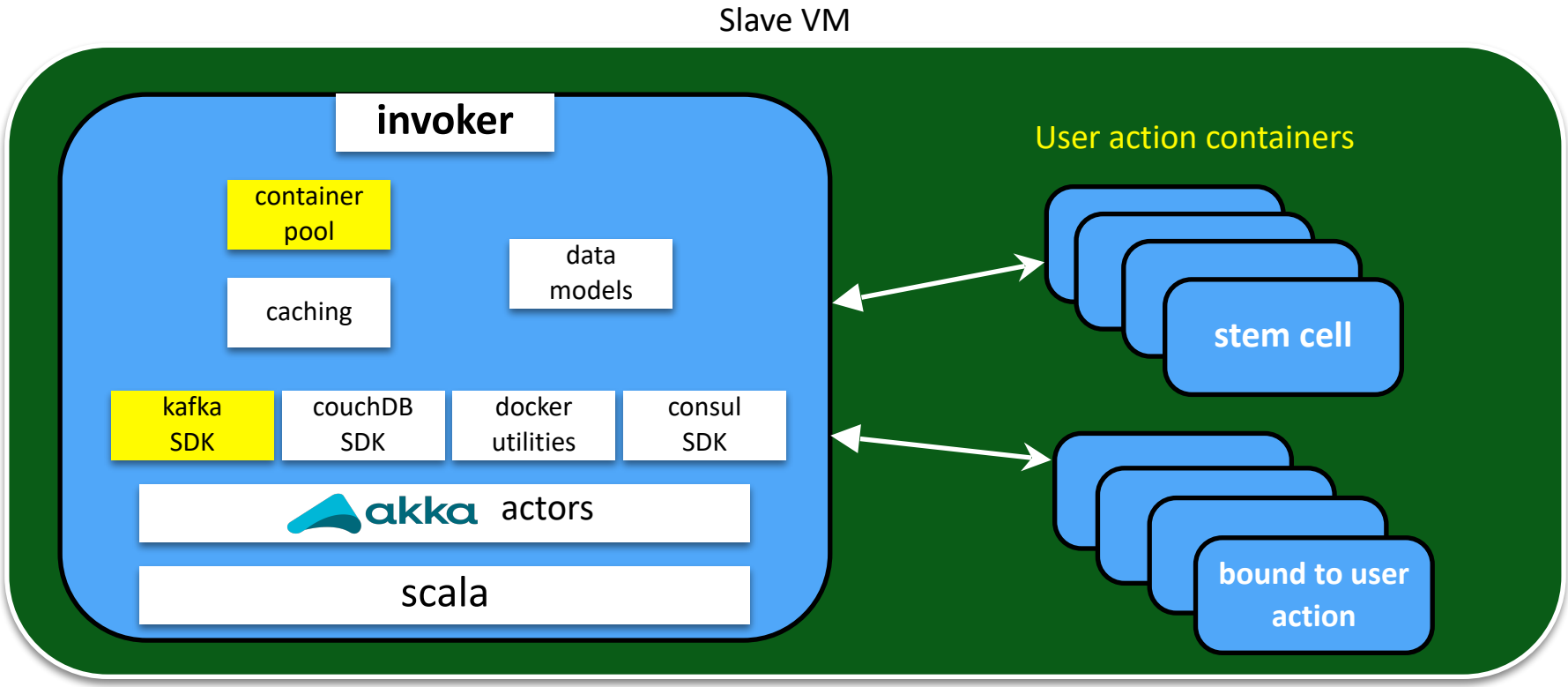
- High throughput fault-tolerant queues
- *Point-to-point* messages via topics
- Explicit load balancing



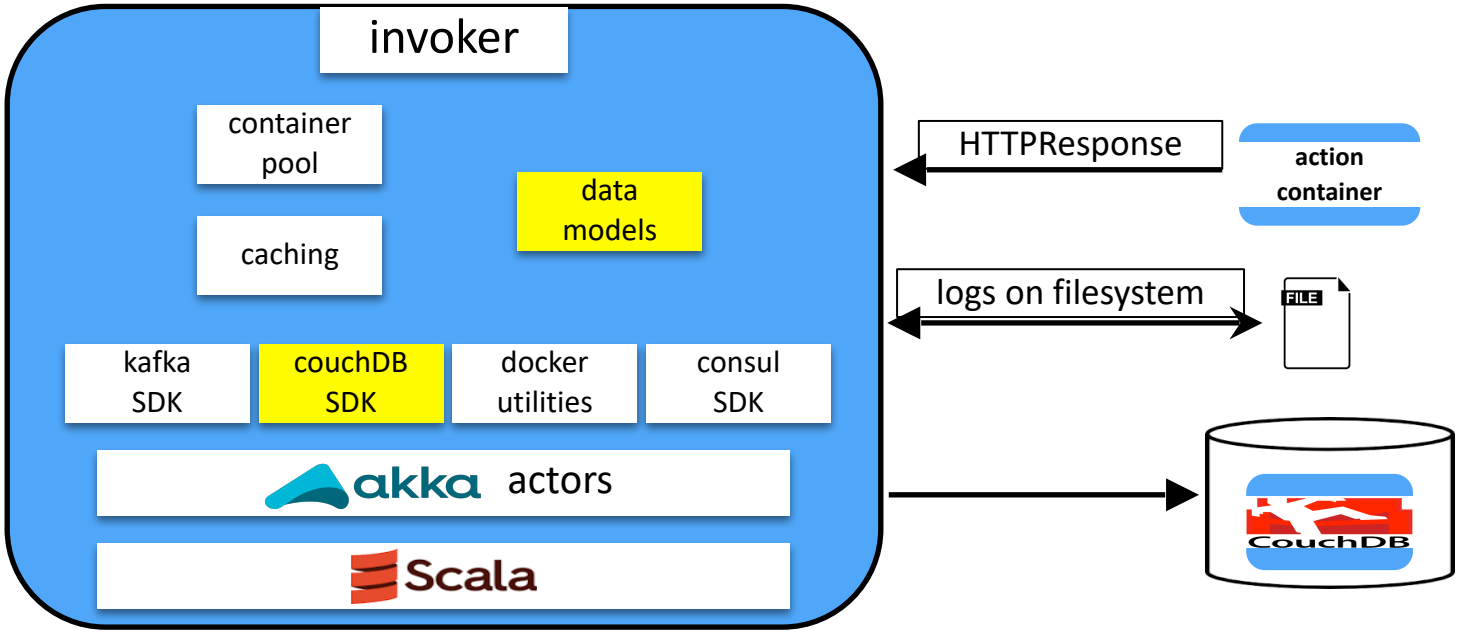
kafka Post request to execute to queue in

OpenWhisk: Step 7. Get to work!

- Each user action gets it own container (isolation)
- Containers may be reused
- Container pool allocates and garbage collects containers



OpenWhisk: Step 8. Store the results.



Overview Serverless Computing

- An auto-scalable cloud resource for hosting and executing stateless functions
- Makes development more cost and time efficient
- A good fit for short-lived stateless microservice
- All primary cloud providers offer serverless platforms