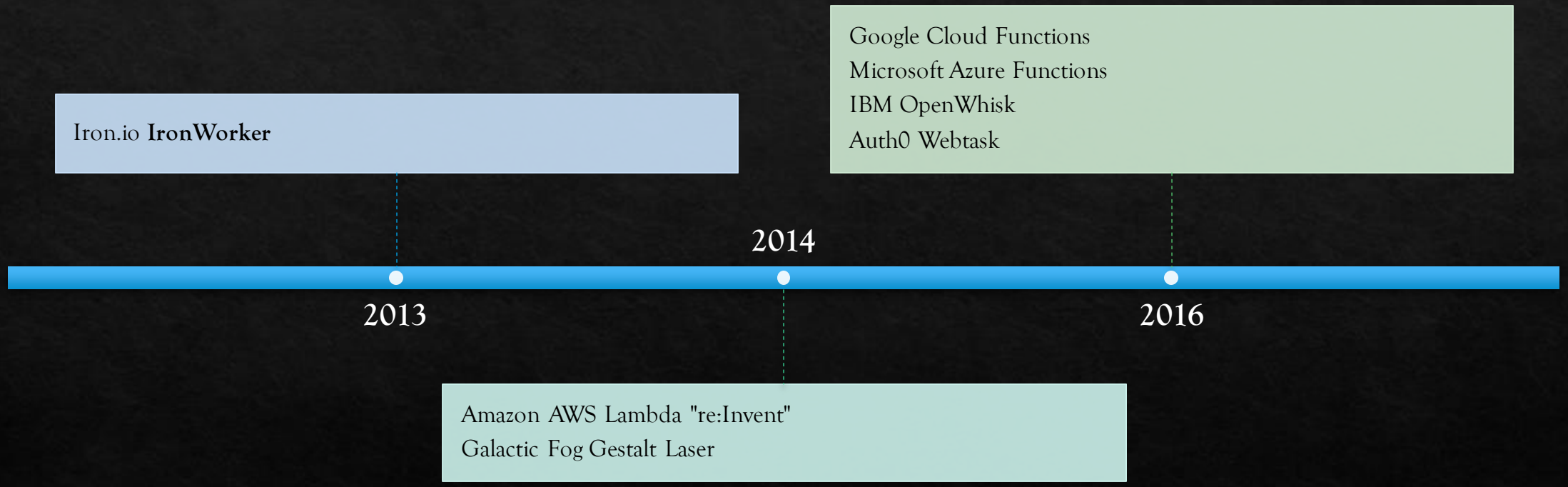

Serverless Computing
How's, what's and why's

Google search term popularity of "serverless"

Popularity

Popularization



Distinction

SERVERLESS Architecture

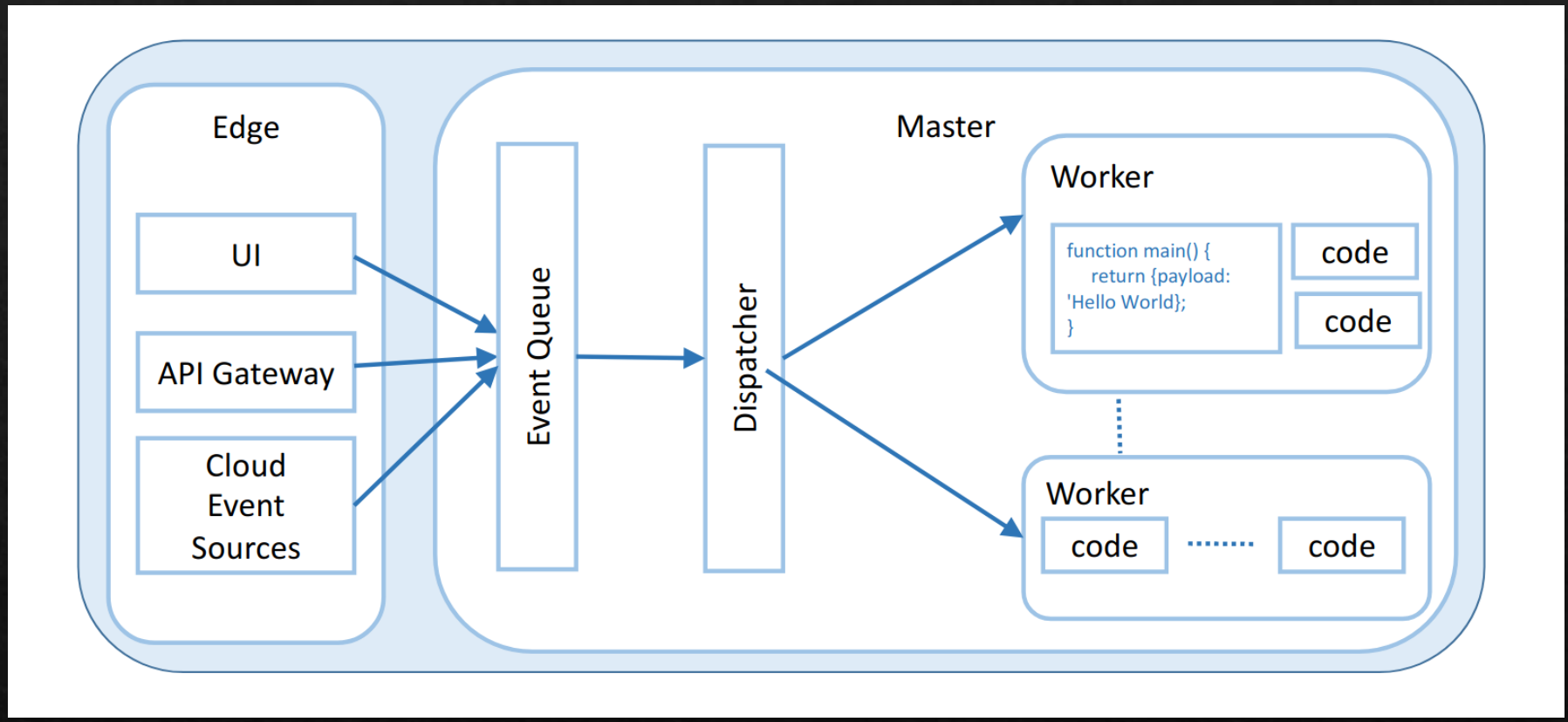
SERVERLESS
COMPUTING

SERVERLESS NETWORK
FILE SYSTEM

Definition

1. Cloud computing execution model
2. Cloud provider runs the server and
3. Dynamically manages machine resources
4. Deployment of stateless functions

Developer Control



Architecture

General unspecific platform architecture

Image from Baldini et al., Serverless Computing: Open Trends and Current Problems
https://arxiv.org/pdf/1706.03178.pdf

Internal Architecture

Based on Containers

That means shared kernel

Function adds last layer

Or mounted if interpreted language

Compiled languages slow first invocation, fast after

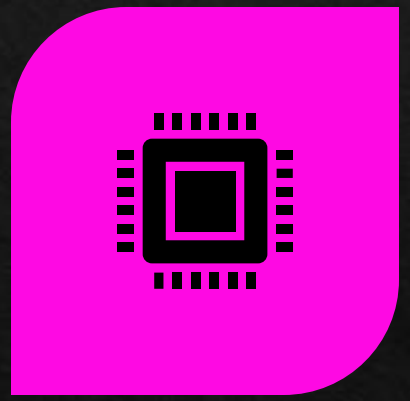
Interpreted languages fast first invocation, not as fast after

Multi-Tenant

Characteristics

- Cost
- Performance and limits
- Programming languages
- Composability
- Deployment
- Security and accounting
- Monitoring and alerting

Benefits

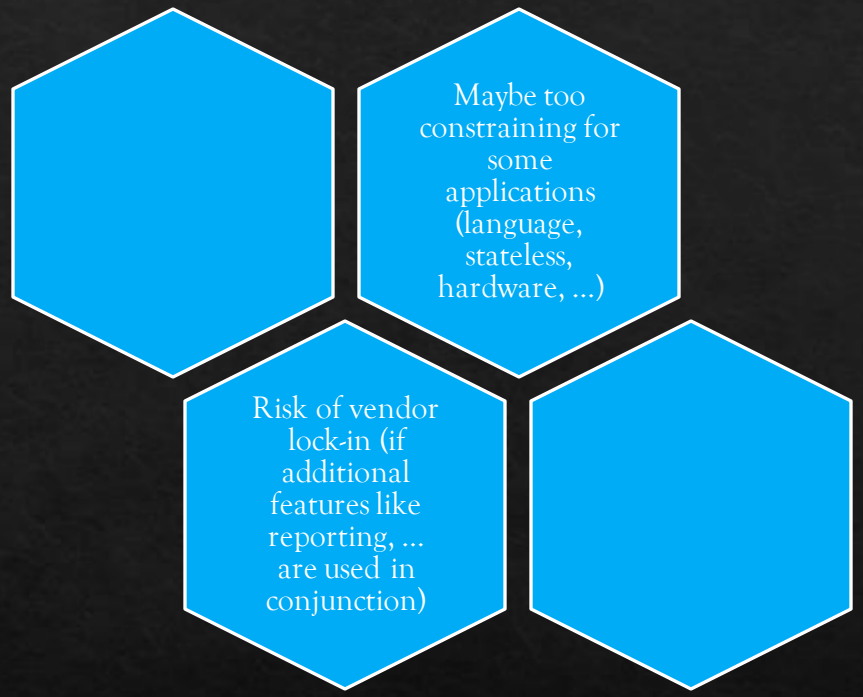


OPTIMISE FUNCTION BY COST (RATHER THAN BY LATENCY, SCALABILITY, AND ELASTICITY)

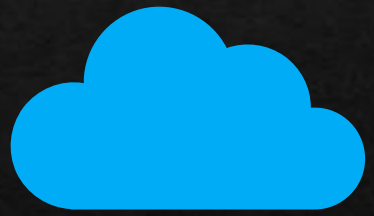
NO LONGER MANAGE SERVER OR INFRASTRUCTURE

STATELESS FUNCTIONS ENABLE PROVIDER (TO PATCH SERVER OR MOVE TENANT, NO NEED TO WAIT)

Drawbacks

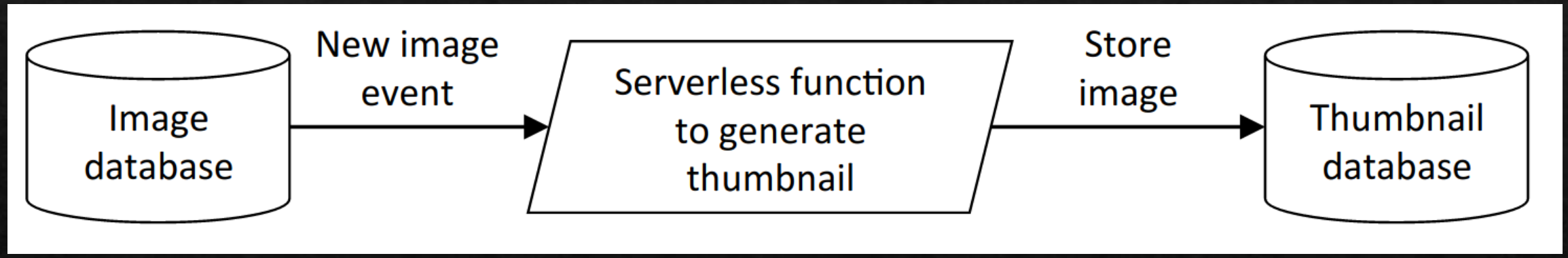

Maybe too constraining for some applications (language, stateless, hardware, …)
Risk of vendor lock-in (if additional features like reporting, … are used in conjunction)

Use Cases



Bursty, compute intensive workload

Avoid IO operations



Event processing

Hello World of AWS Lambda: Event driven image processing

Image from Baldini et al., Serverless Computing: Open Trends and Current Problems
https://arxiv.org/pdf/1706.03178.pdf

API composition

Offloading API calls

and glue logic from mobile app to backend

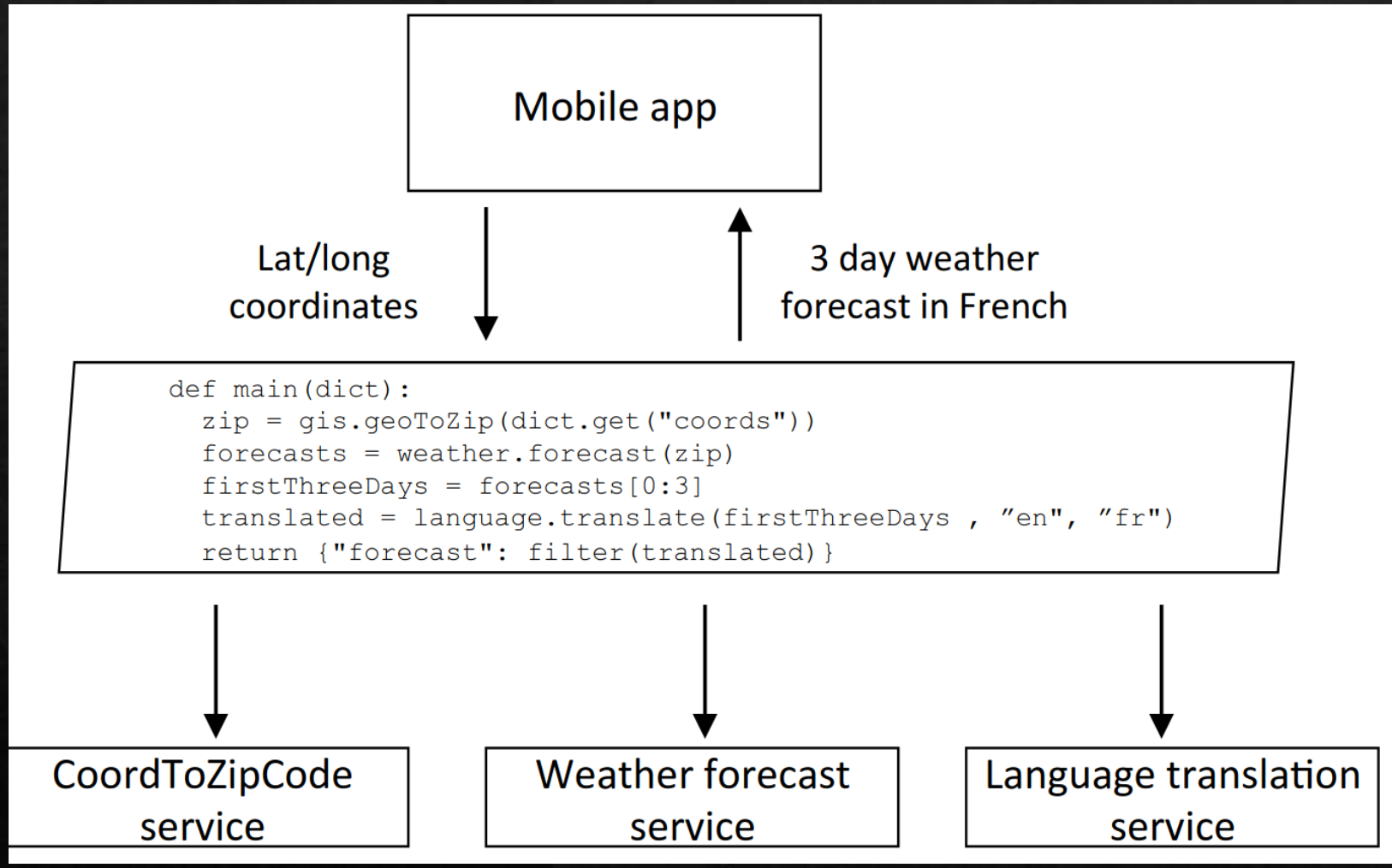
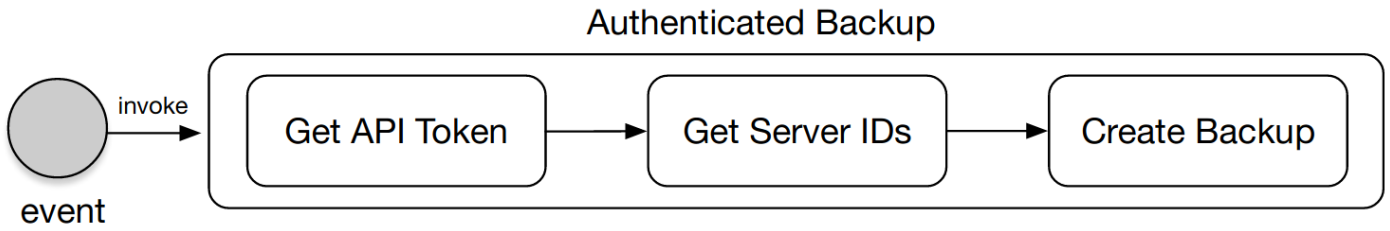


Image from Baldini et al., Serverless Computing: Open Trends and Current Problems

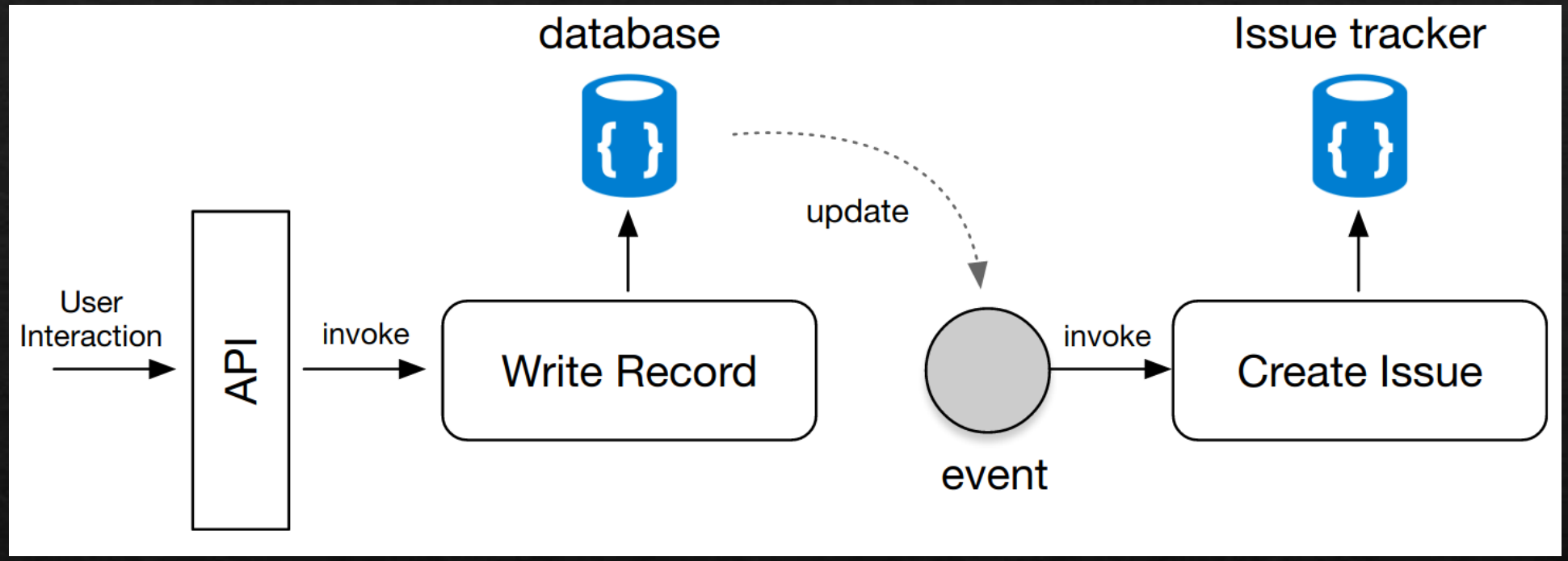
https://arxiv.org/pdf/1706.03178.pdf



API aggregation

Reducing the number of API calls required for a mobile client

Image from Baldini et al., Serverless Computing: Open Trends and Current Problems https://arxiv.org/pdf/1706.03178.pdf



Flow control

Batched invocation for issue tracking

Image from Baldini et al., Serverless Computing: Open Trends and Current Problems
https://arxiv.org/pdf/1706.03178.pdf

Different Eco Systems

Comparison

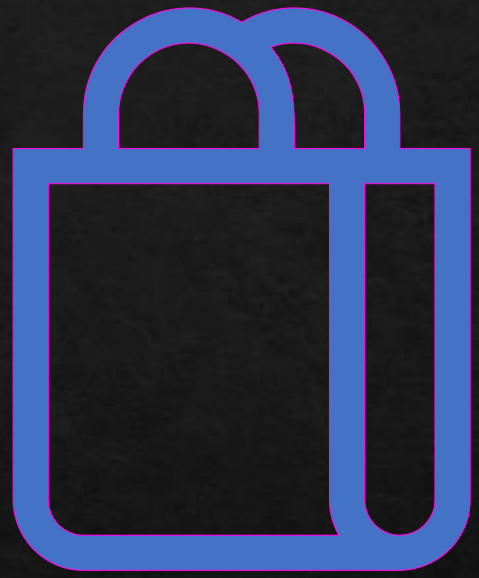
	AWS Lambda	Microsoft Azure	Google Cloud Functions
Memory (MB)	64 * k in (2,...,47)	1536	128 * k in (1,...,32)
CPU	Proportional to Memory	Unknown	Proportional to Memory
Language	Node, Python, Ruby, Java, Go, .Net Others via custom runtime	C#, Java, JavaScript, TypeScript, Python, Powershell	Nodejs, Python, Go, Java
Runtime OS	Amazon Linux	Windows 10, Linux	Debian 8
Local disk (MB)	512	500	> 512
Timeout (sec)	900 (configurable from default: 3)	600	540 (configurable from default: 60)
Billing factor	Execution time Allocated memory	Execution time Consumed memory	Execution time Allocated memory Allocated CPU

As of 2020-11-06:
AWS: https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-limits.html
Azure: https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/azure-subscription-service-limits
Google: https://cloud.google.com/functions/quotas

Many other:

- IBM OpenWhisk
- Iron.io Ironworker
- Auth0 Webtask
- Galactic Fog Gestal Laser
- ...

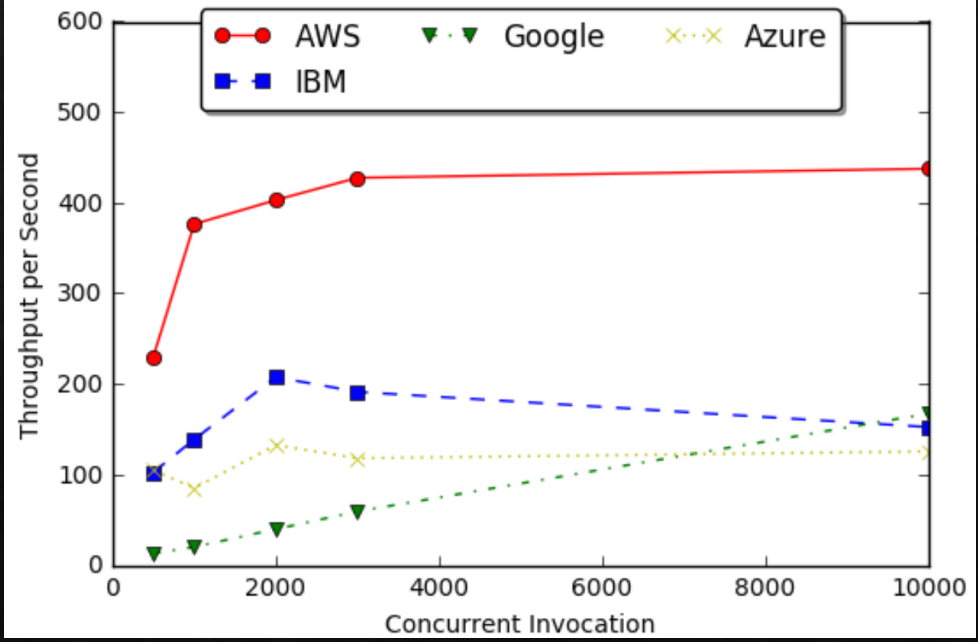
according to:
Lynn et al.: A Preliminary Review of Enterprise Serverless Cloud Computing Fucntion as a Service Platforms
https://ieeexplore.ieee.org/abstract/document/8241104



Performance and Price Comparison

4 short examples

Function Throughput on Concurrent Invocations



CPU Performance

Provider	GFLOPS per function	TFLOPS in total of 3000
AWS	19.63	66.30
Azure	2.15	7.94
Google	4.35	13.04
IBM	3.19	12.30

Lee et al: Evaluation of Production Serverless Computing Environments
https://ieeexplore.ieee.org/abstract/document/8457830

Median Write/Read Speed (MB/s)

Provider	100 Concurrent		1 Concurrent	
	Write	Read	Write	Read
AWS	39.49	92.95	82.98	152.98
Azure	-	-	44.14	423.92
Google	3.57	54.14	9.44	55.88
IBM	0.50	33.89	7.86	68.23

Building Binary Tree with Cost-Awareness

Platform	RAM	Cost/Sec	Elapsed Second	Total Cost (Rank)
AWS Lambda	3008MB	$4.897e-5	20.3	$9.9409e-4 (6)
AWS EC2 (t2.micro)	1GiB	$3.2e-6	29.5	$9.439e-05 (3)
Azure Functions	192MB	$3e-6	71.5	$2.145e-4 (4)
Azure VM	1GiB	$3.05e-6	88.9	$2.71145e-4 (5)
Google Functions	2GB	$2.9e-5	34.5	$0.001 (7)
Google Compute (f1-micro)	600MB	$2.1e-6	19.2	$4.0319e-05 (1)
IBM OpenWhisk	128MB	$2.2125e-6	34.2	$7.5667e-05 (2)

Lee et al: Evaluation of Production Serverless Computing Environments
https://ieeexplore.ieee.org/abstract/document/8457830

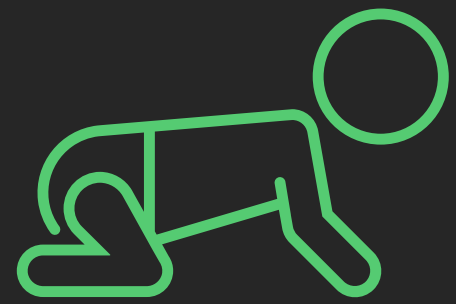
Examples

(only AWS Lambda)

Getting started on AWS Lambda

Register AWS Account (CreditCard needed)

That's it!



https://aws.amazon.com/lambda

Example 1

AWS LAMBDA CONSOLE

AWS LAMBDA DESIGNER

https://console.aws.amazon.com/lambda/home

https://docs.aws.amazon.com/lambda/latest/dg/getting-started-create-function.html

Example 2

INSTALLING AWS CLI

CREATING HELLOWORLD

https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html

https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-awscli.html

https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-configure.html

Example 3

AWS Toolkit Plugin

(here: IntelliJ)

https://aws.amazon.com/intellij/

Example 4

Kotlin: Kotless

https://github.com/JetBrains/kotless

Example 5

PHP: BREF

https://bref.sh/

Further Read

1. AWS Whitepaper:
    1. https://d1.awsstatic.com/whitepapers/serverless-architectures-with-aws-lambda.pdf
2. Any of the mentioned resources
3. Google scholar: `Serverless Computing`
4. Resources and examples of this Github repo:
    1. https://github.com/HartmannS/serverless_presentation