Serverless architecture: Functions as a Service

Dzmitry Varabei aka “Dean”

Dzmitry Varabei

Chief Software Engineer

- 10+ years of Software Development
- 7 years in education and knowledge sharing
- 3+ years in community building

https://school.rollingscopes.com/

ThoughtWorks®

TECHNOLOGY RADAR *APRIL '16*

Our thoughts on the technology and trends that are shaping the future

Serverless is coming!

Serverless “hype”

1. News
2. Blog posts
3. Meetups
4. Conferences

…..

CAUTION

USE AT YOUR OWN RISK

“Нужно бежать со всех ног, чтобы только оставаться на месте, а чтобы куда-то попасть, надо бежать как минимум вдвое быстрее!”

(Lewis Carroll, Through the Looking Glass, Chapter 2)

“If you wait by the river long enough, the bodies of frameworks **will float by”** *Sun Tzu*

Chapter I:
The hardest problem

The hardest problem in computer science

- …is, of course, naming.

Serverless architecture!

Just right for front-end devs?

02 WINS
60
Angular.js developer
CAGE
Front-end WINS
FATALITY
???
DevOps ?
Java dev?

NO

NO

NO

Fashionable Naming

Cloud Computing

Infrastructure as a service

Platform as a service

Function as a service

Serverless architecture

AWS Lambda, Phoenix Environments, API Gateway,

……..

Chapter II:
From the Iron Age to the Cloud Age

Self-hosting
(aka On-premise)

You buy and install hardware

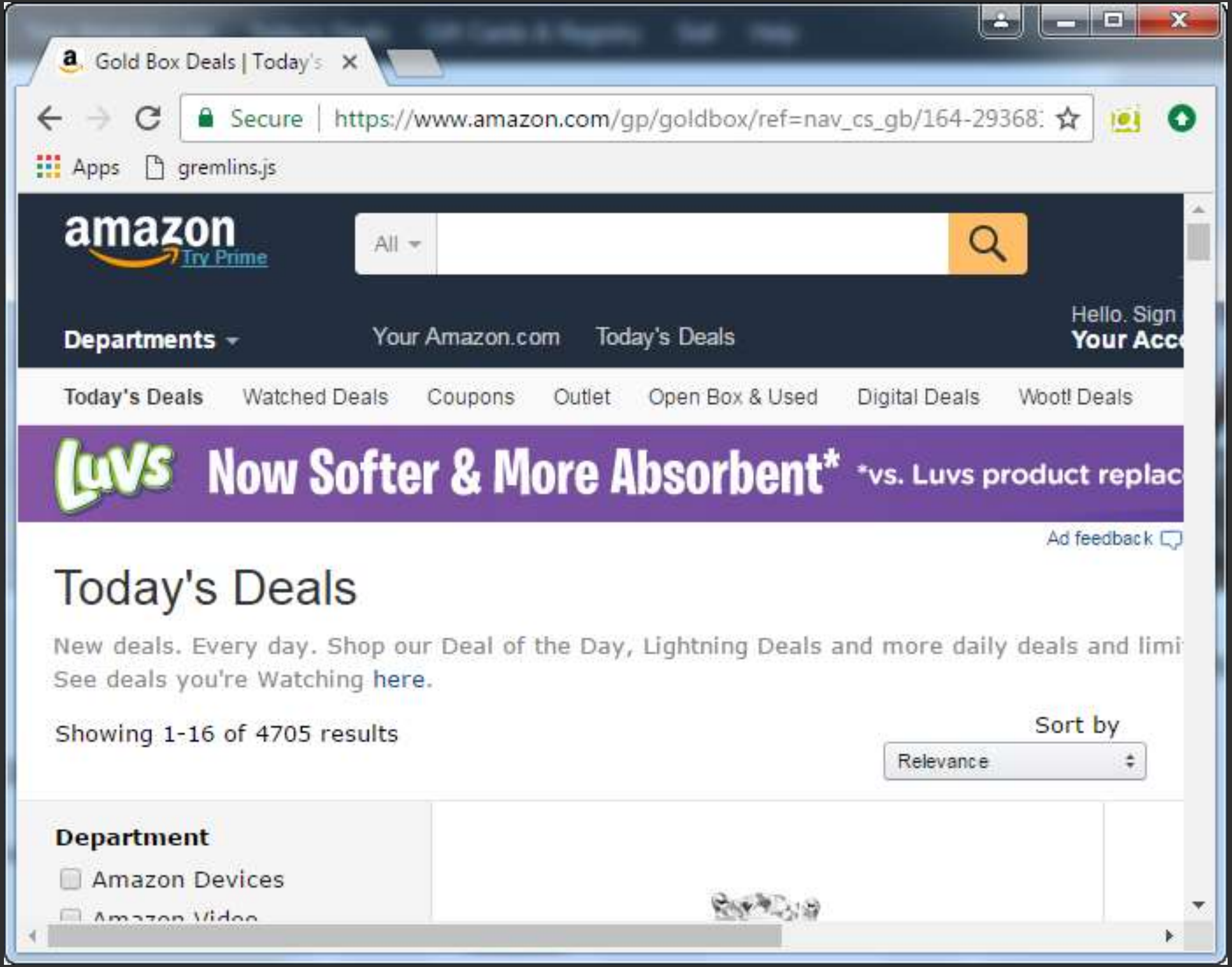
You manage networks

You scale

[Silicon Valley](http://www.imdb.com/title/tt2575988/)

You maintain

In Amazon we trust



Gold Box Deals | Today's

Secure | https://www.amazon.com/gp/goldbox/ref=nav_cs_gb/164-29368

Apps gremlins.js

Today's Deals

New deals. Every day. Shop our Deal of the Day, Lightning Deals and more daily deals and limi
See deals you're Watching here.

Showing 1-16 of 4705 results

Sort by: Relevance

Department

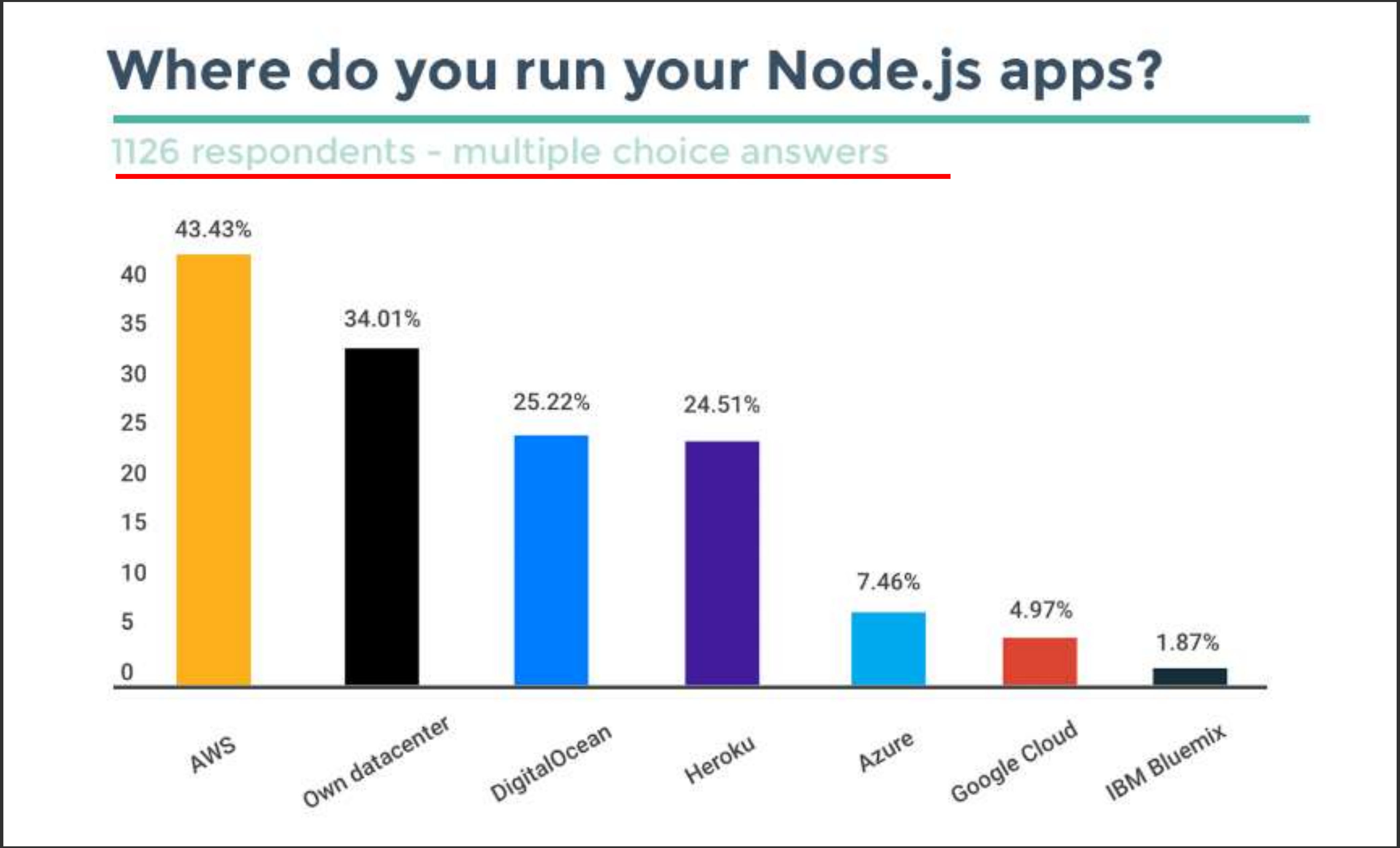
- Amazon Devices
- Amazon Video

Survey

“**How** Developers use **Node.js**”

1126 Respondents

https://blog.risingstack.com/node-js-developer-survey-results-2016/



https://blog.risingstack.com/node-js-developer-survey-results-2016/

Three DevOps Survey

Results: 80-90% AWS/Hubrid/ Private cloud

Private cloud

PayPal OpenStack Private Cloud

Physical servers: 8000+

Total cores: 400 000+

Number of VMs: 82000+

Storage: 2 petabytes

Processed $228 billion in payments last year

https://www.openstack.org/summit/tokyo-2015/videos/presentation/architectures-for-successful-openstack-cloud-deployment

Chapter III:
Cloud computing services

Infrastructure as a service (IaaS)

is a form of cloud computing that provides virtualized computing resources over the Internet.

Self Hosting	
YOU MANAGE	Applications
	Data
	Runtime
	Middeware
	O/S
	Virtualization
	Servers
	Storage
	Networking

VS

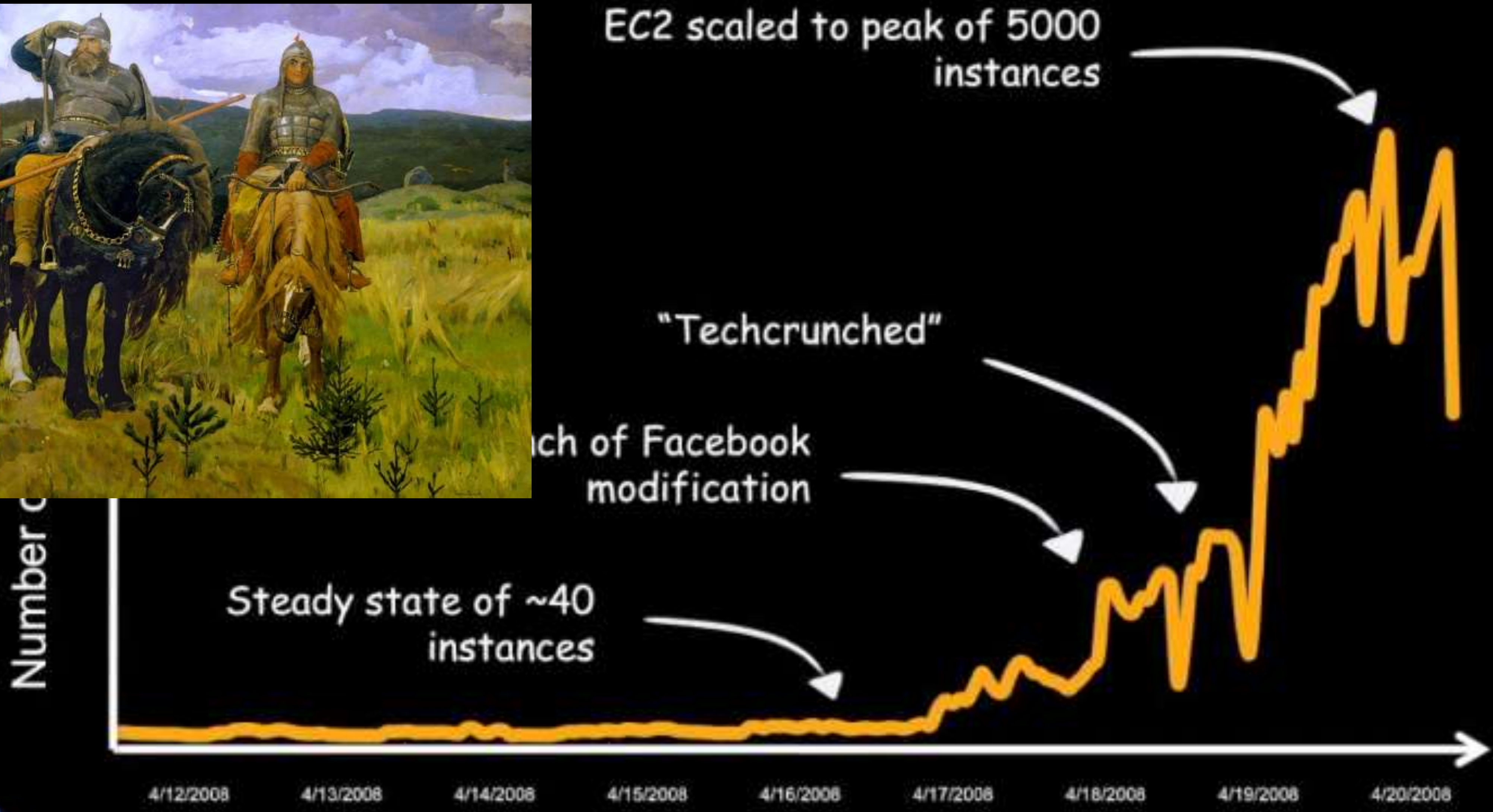
Infrastructure as a service	
YOU MANAGE	Applications
	Data
	Runtime
	Middeware
	O/S
VENDOR	Virtualization
	Servers
	Storage
	Networking

Self Hosting	
YOU MANAGE	Applications Data Runtime Middeware O/S Virtualization Servers Storage Networking

VS

Infrastructure as a service	
YOU MANAGE	
VENDOR	Virtualization Servers Storage Networking

40 servers to 5000 in 3 days



Case Study

IAAS benefits

- Infrastructure in minutes
- Reduced operational cost
- Pay as you go
- Worldwide
- Scalability and flexibility
- Anytime, Anywhere, Access

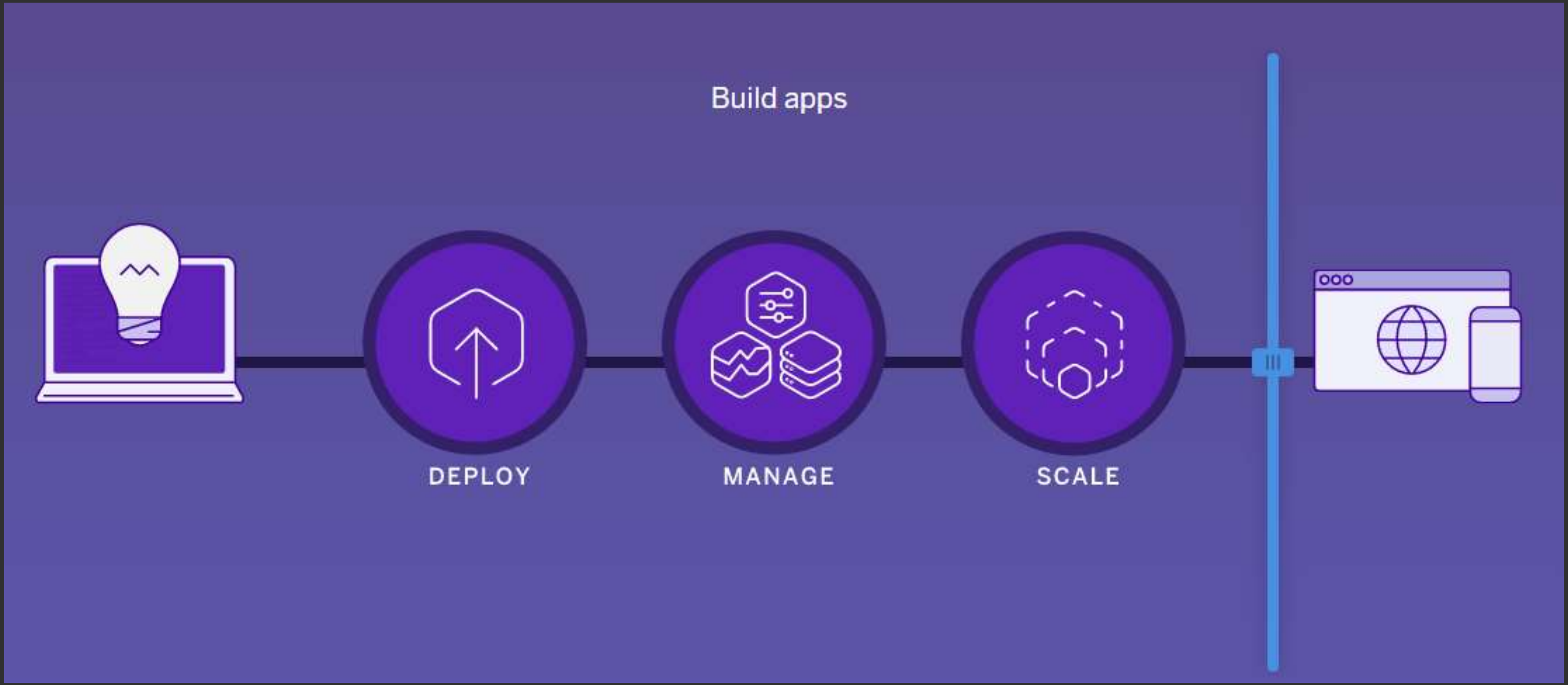
“add dev/prod env, env in japan, add 1000 servers or remove 1000 servers - by clicking a button”

Platform as a service

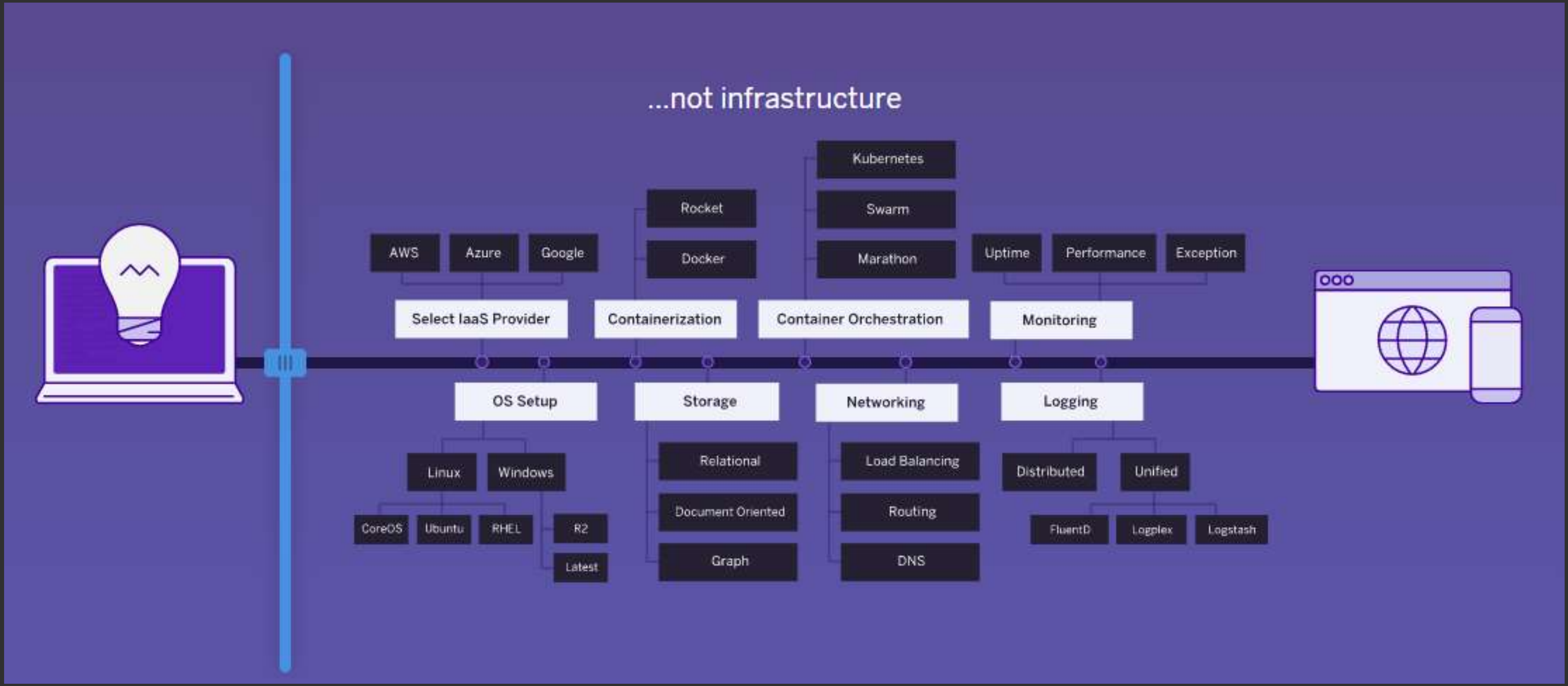
Platform as a service

YOU	Applications Data
VENDOR MANAGE	Runtime Middeware O/S Virtualization Servers Storage Networking

Heroku: “Build apps…



Heroku: "… not infrastructure"



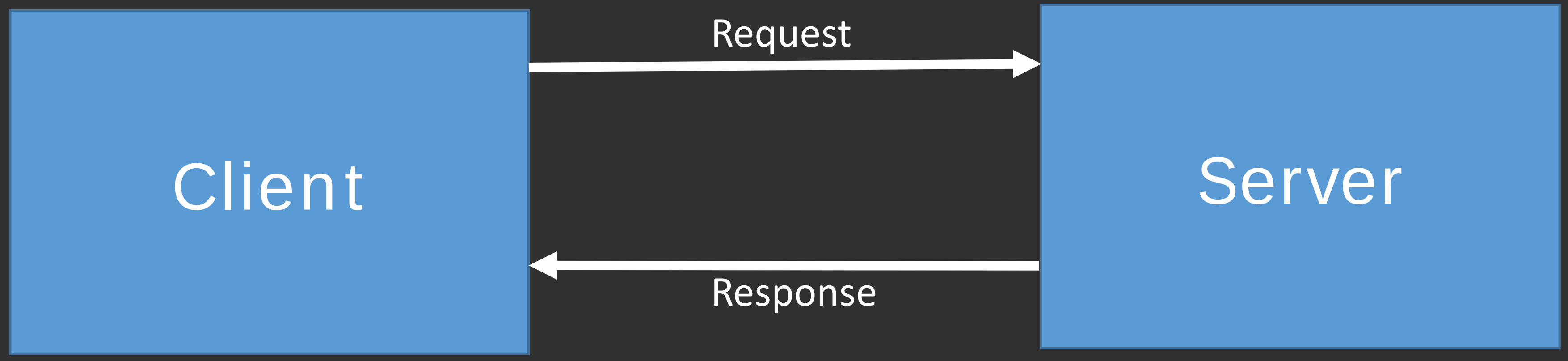
Platform as a service

“Many large organizations see the Cloud and Platform as a Service (PaaS) as an obvious way to standardize infrastructure, ease deployment and operations, and make developers more productive. But it’s still early days…”

(C)ThoughtWorks Radar

Chapter III:
Scalability and cost basics

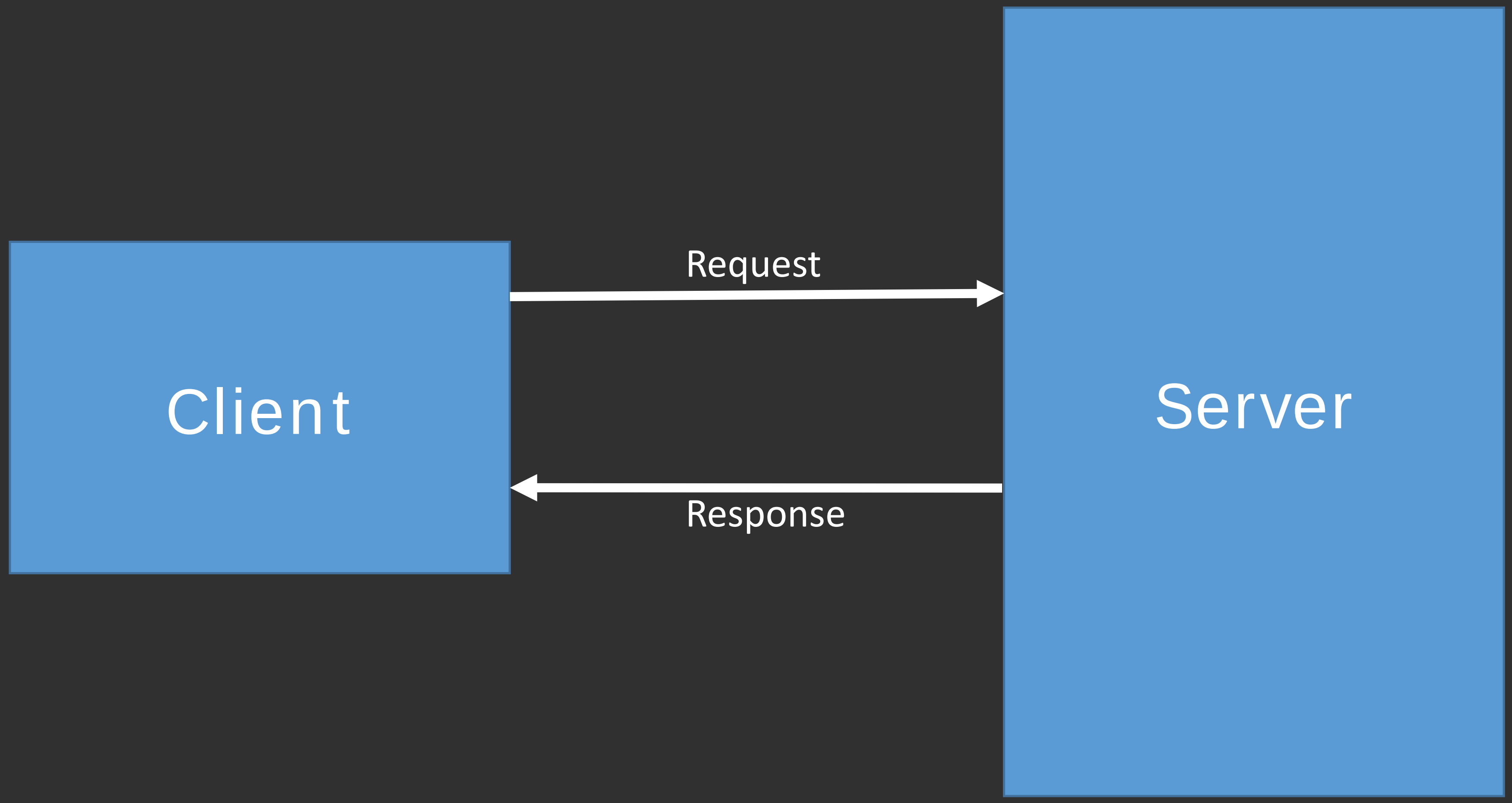
Classic Client-Server Architecture



What about scaling?

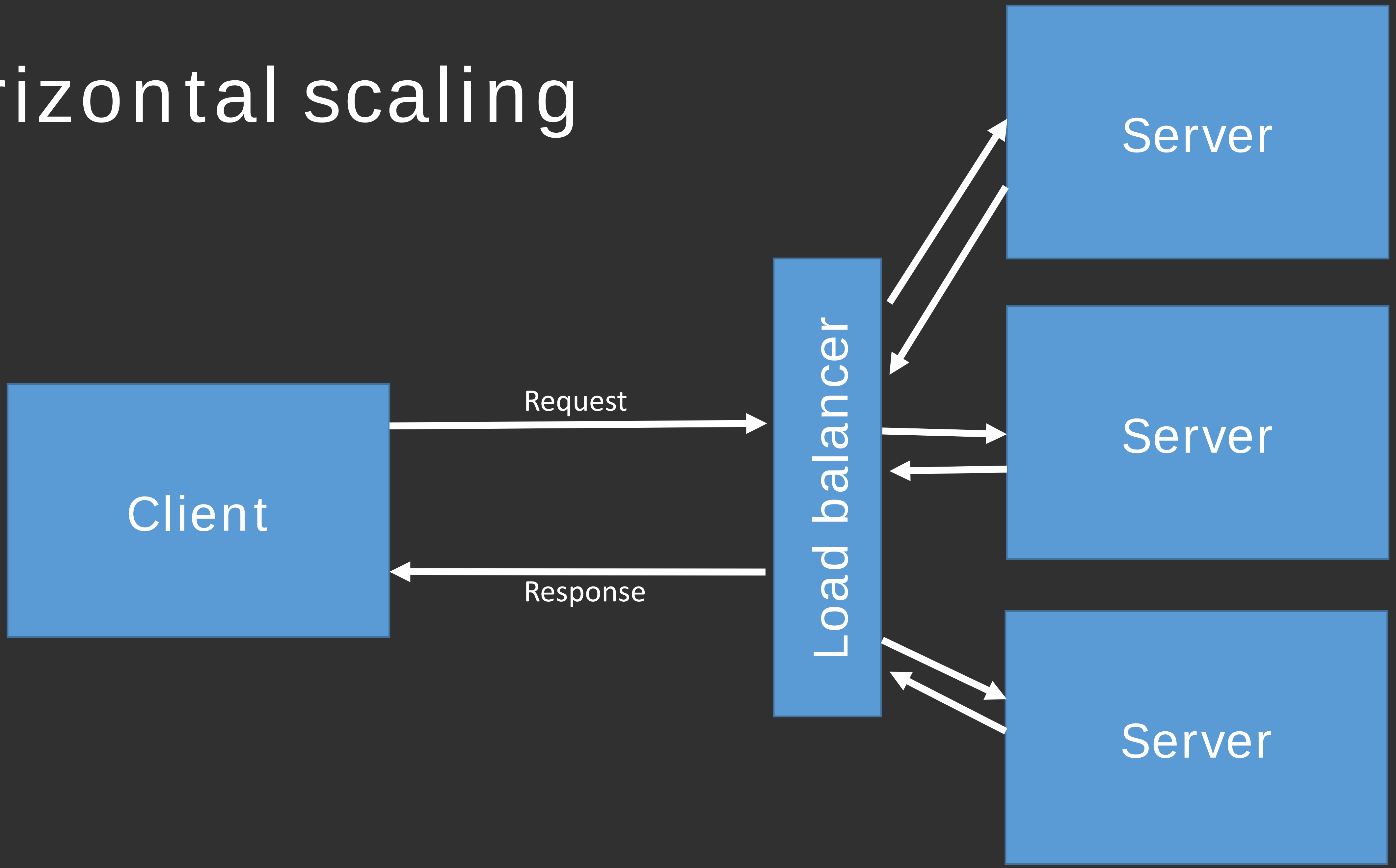
10 users -> 10000+ users

Vertical scaling



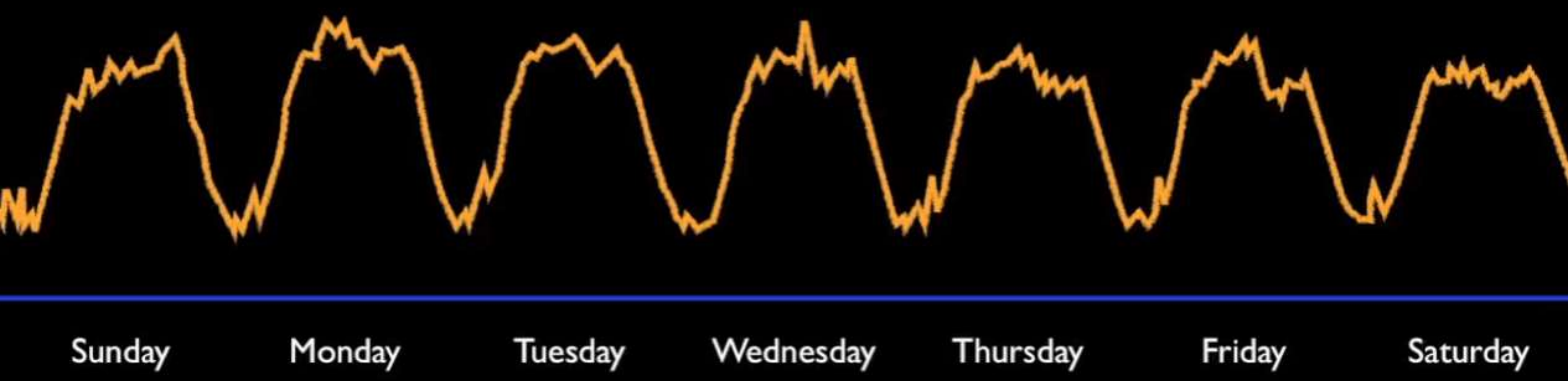
That’s enough for 90 percent of projects

Horizontal scaling



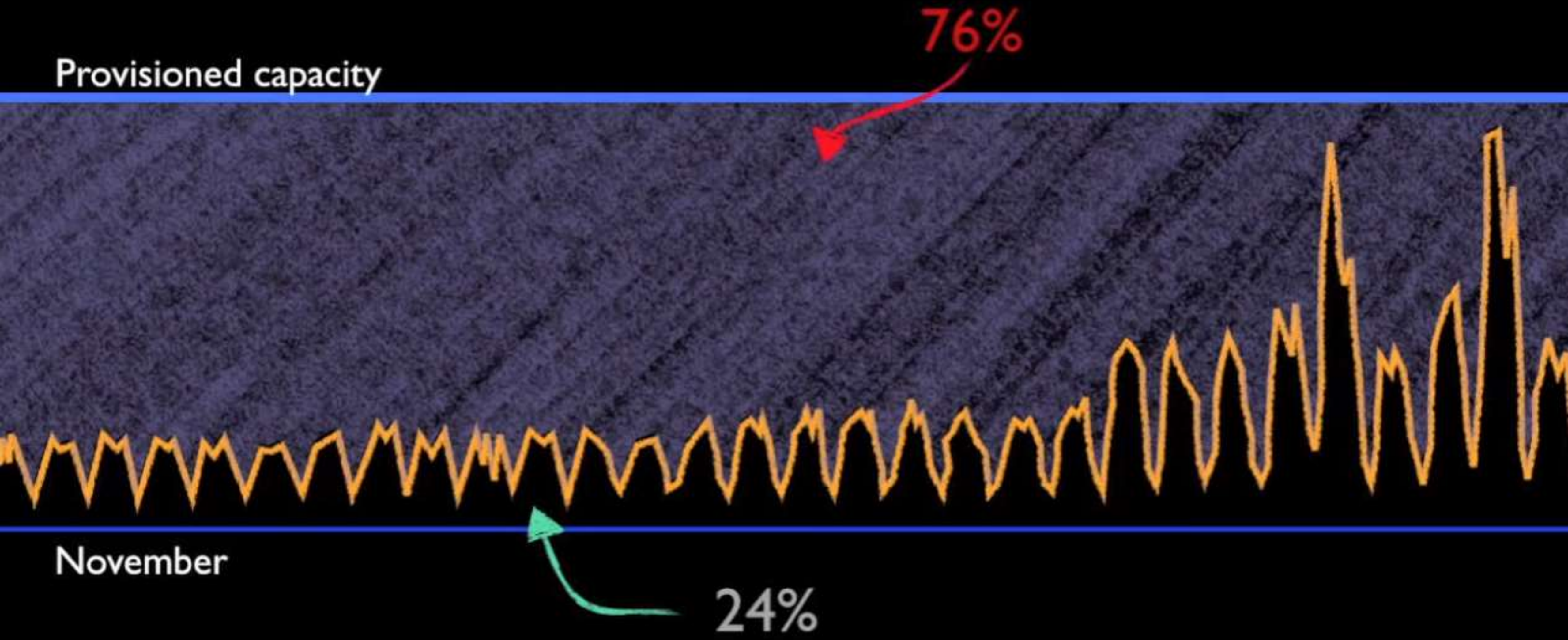
Traffic story

Typical weekly traffic to Amazon.com



https://www.youtube.com/watch?v=DERzYnthq1s

November traffic to Amazon.com



https://www.youtube.com/watch?v=DERzYnthq1s

Chapter IV: Function as a service

Serverless architecture

1. Infinite scaling
2. Only pay for the compute that you need
3. No servers to manage

Function as a service

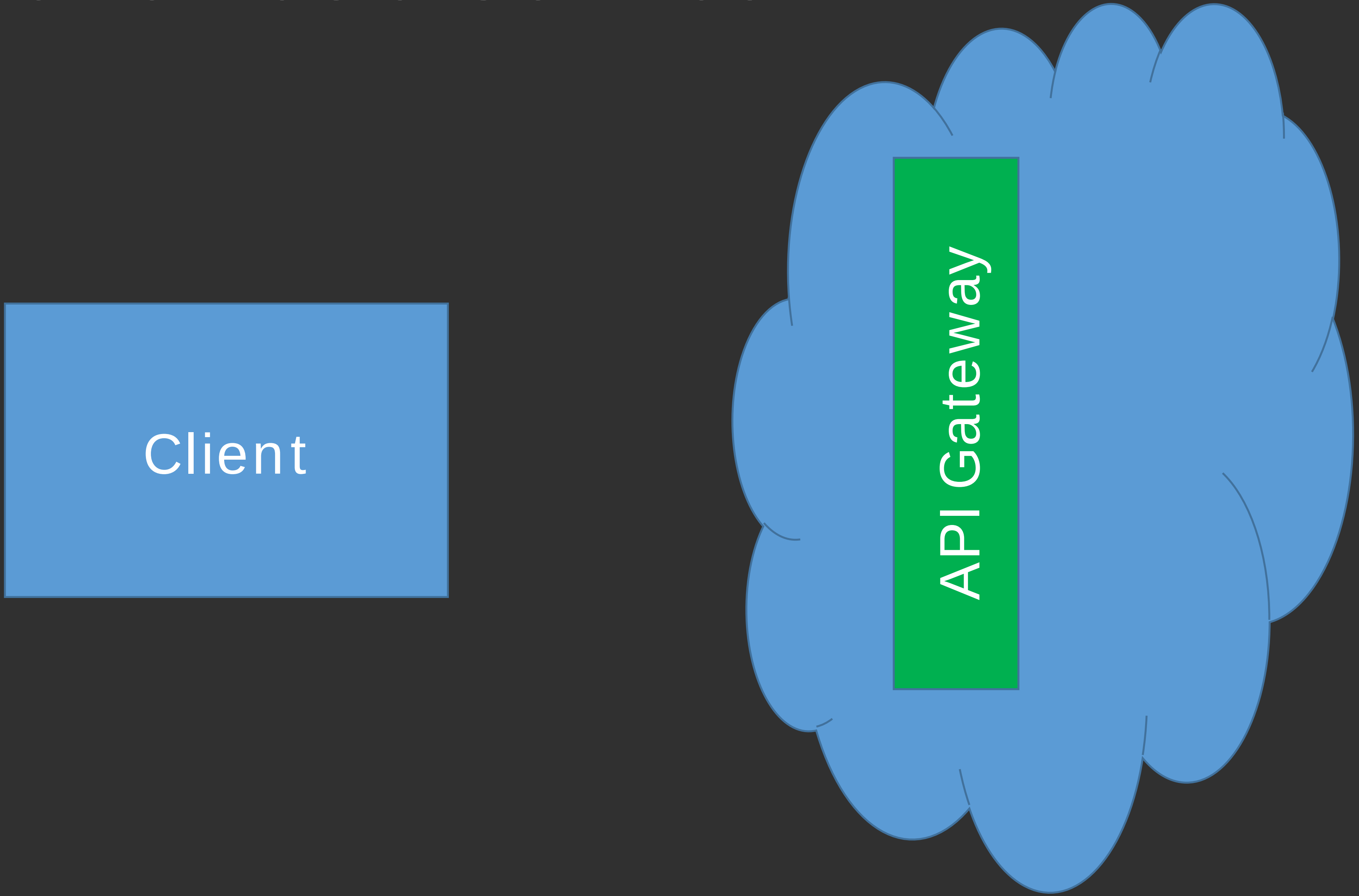
YOU	Applications
	Data
VENDOR MANAGE	Runtime
	Middeware
	O/S
	Virtualization
	Servers
	Storage
	Networking

Function as a Service

“ Magic r power that comes into existence on request and dissapears immediatly after use”

(c) ThoughtWorks Radar

Function as a Service

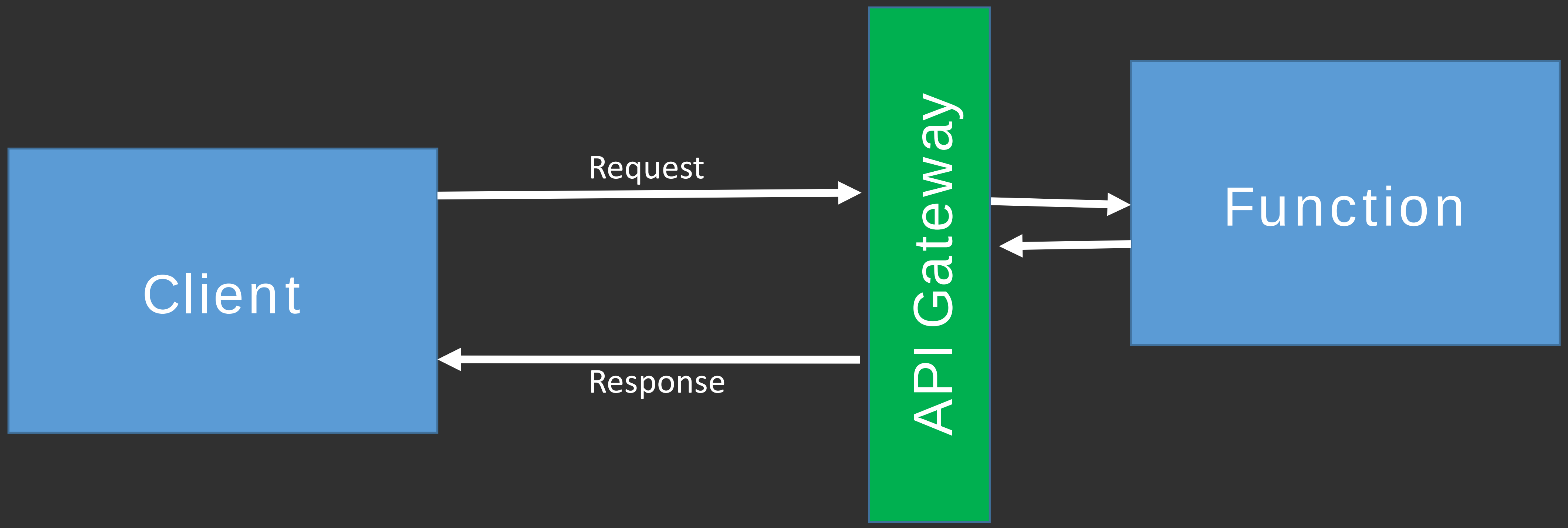


Amazon API Gateway

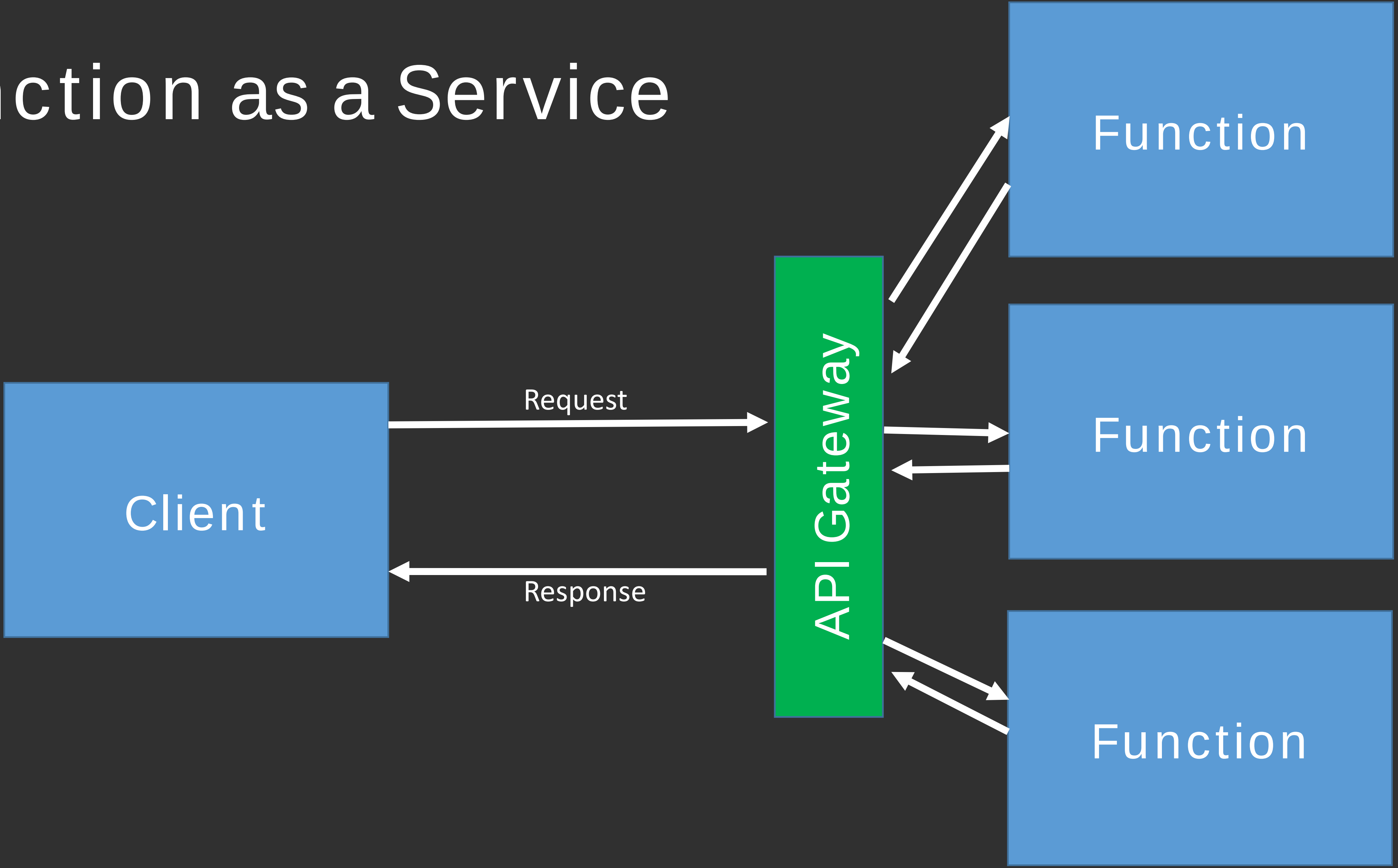
- Fully managed and scalable **RESTful** API gateway service
- Powered by our **content delivery network** via 59 global edge locations

https://www.youtube.com/watch?v=YEWtQsqIYsk

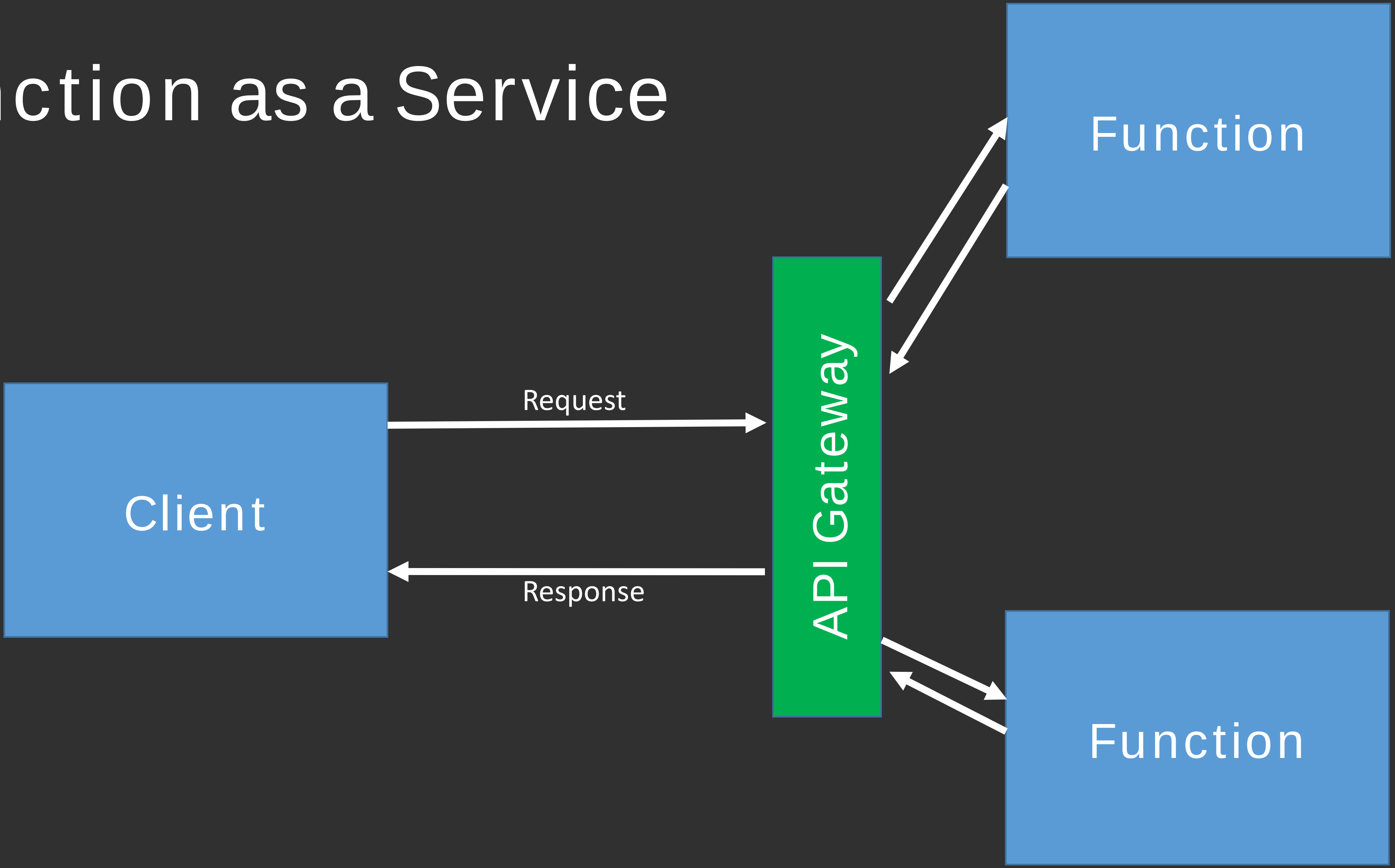
Function as a Service



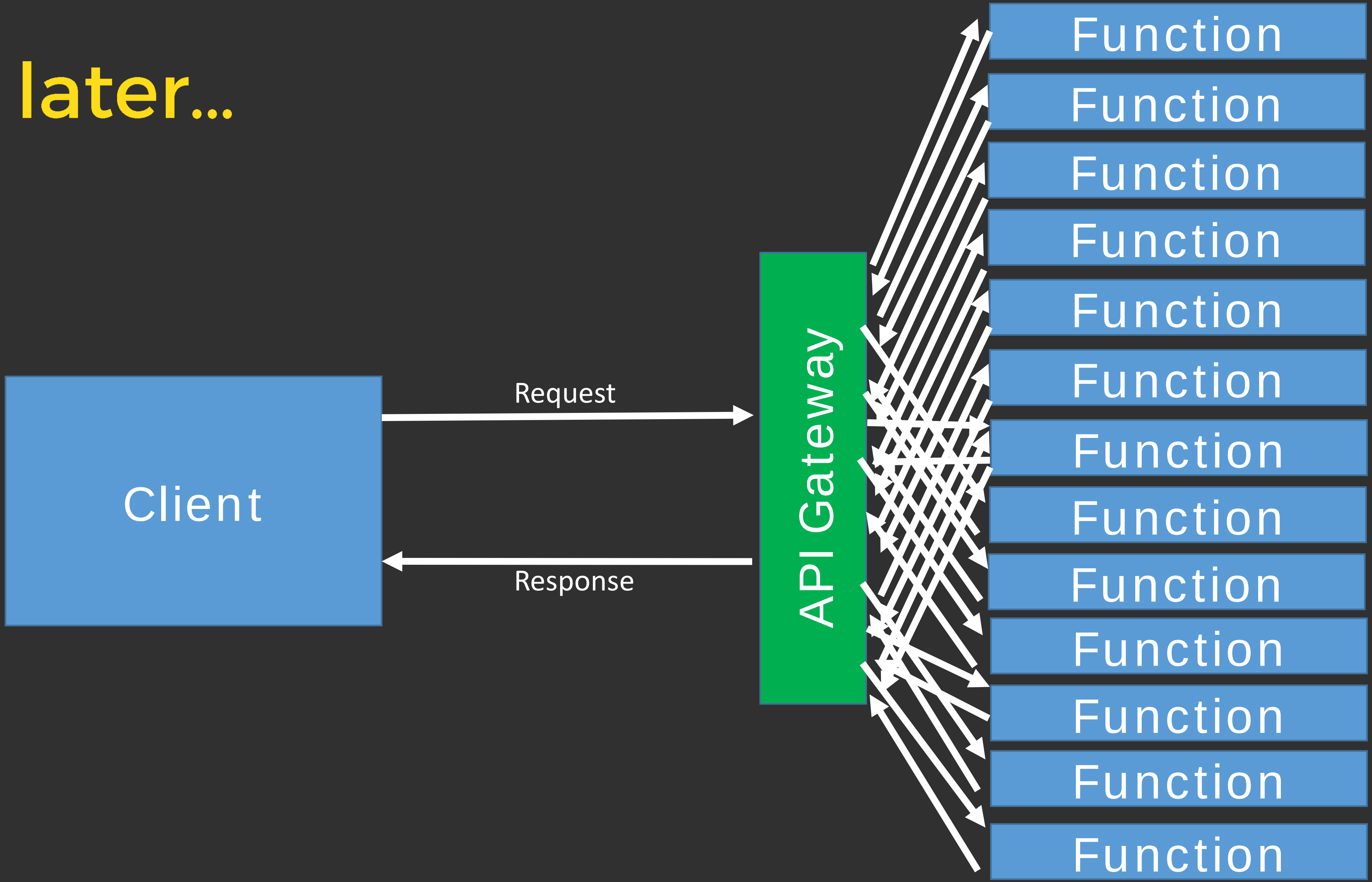
Function as a Service



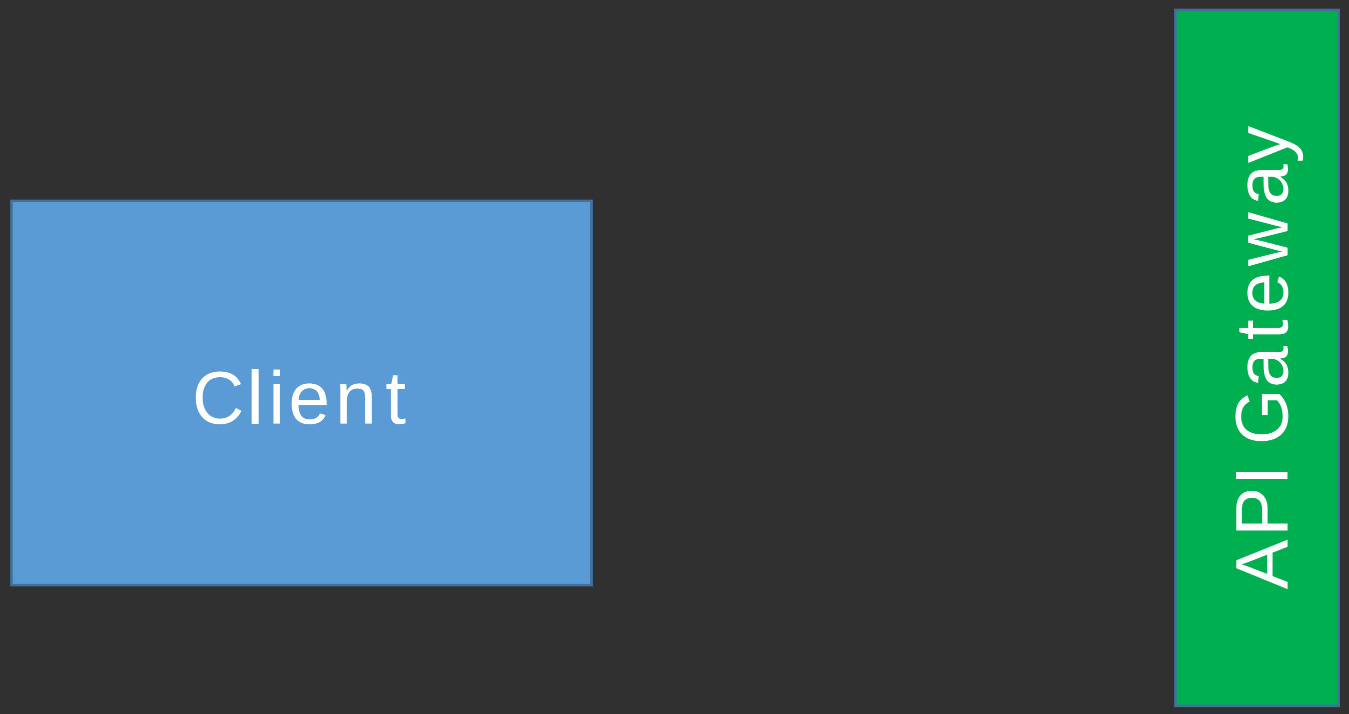
Function as a Service



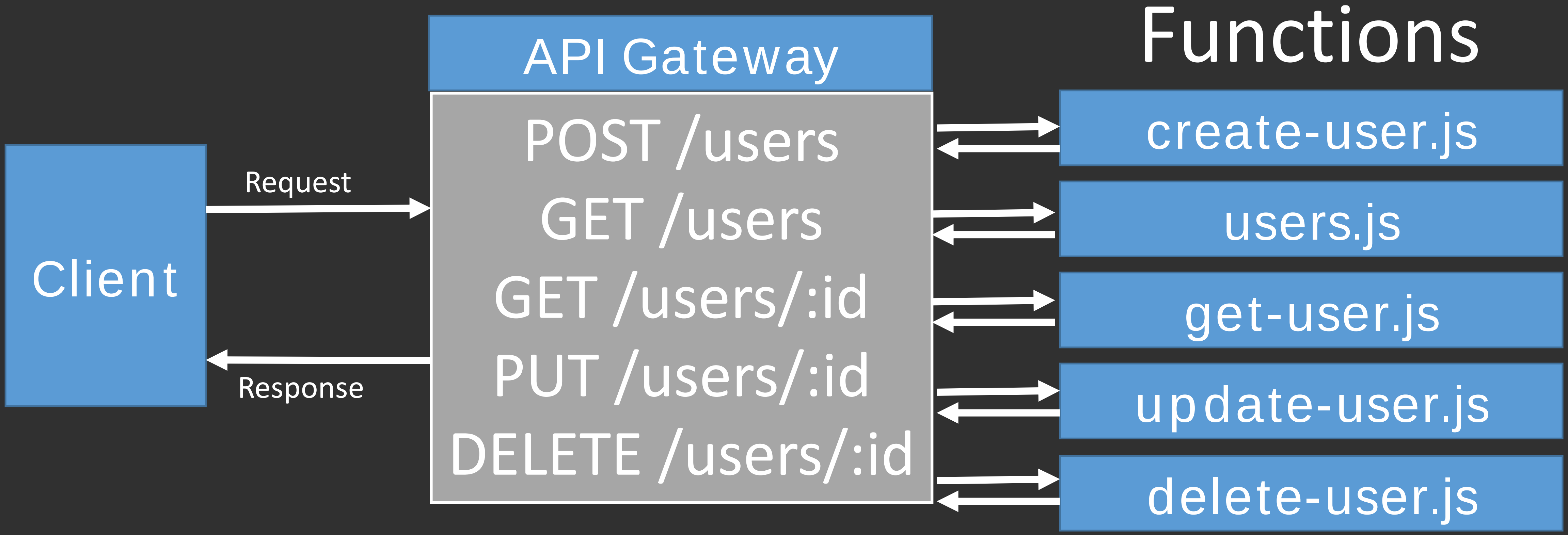
30s later…


Client
Request
Response
API Gateway
Function
Function
Function
Function
Function
Function
Function
Function
Function
Function
Function
Function
Function

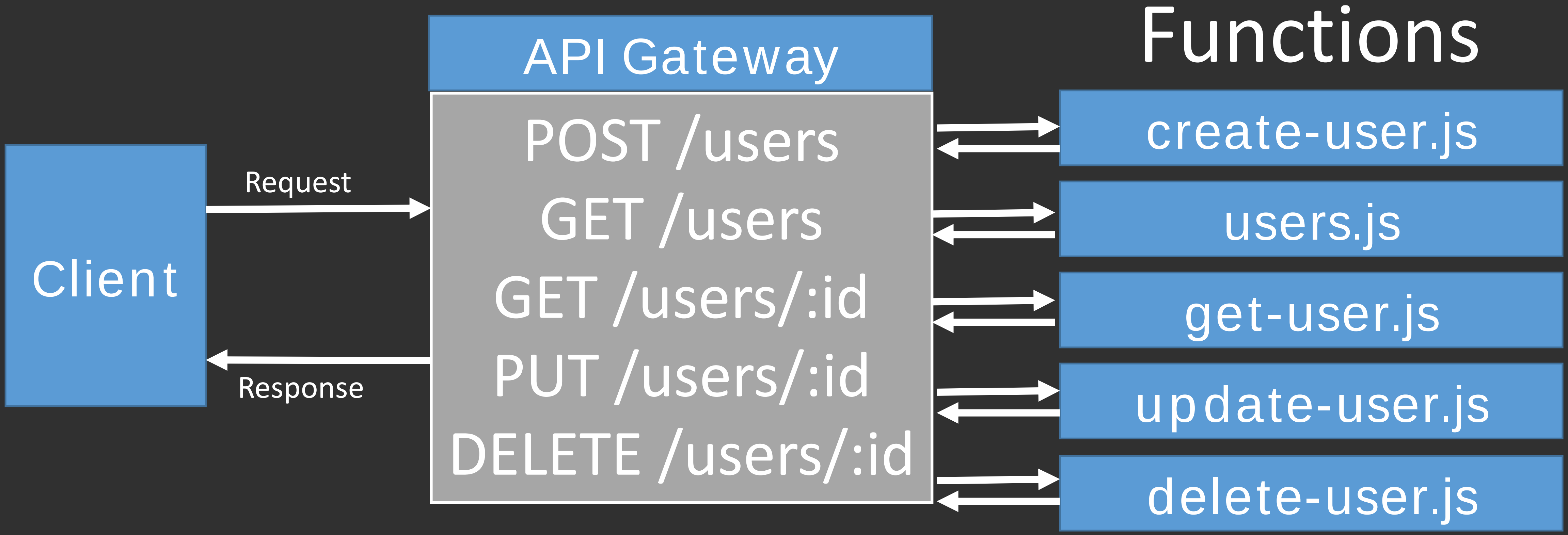
60s later...



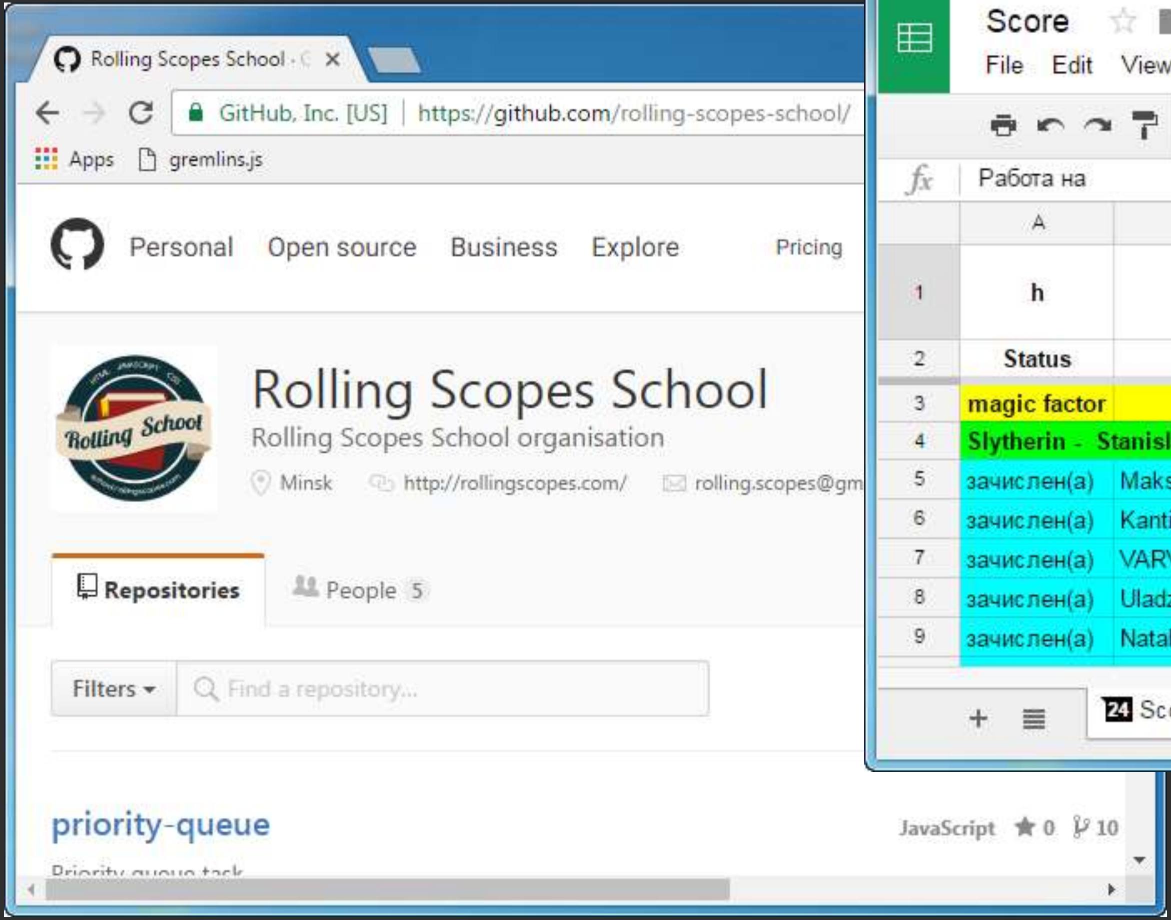
Nanoservices



Granular Deploy, Update, Scale, Reliability, etc.



Use case #1



Rolling Scopes School · GitHub

GitHub, Inc. [US] | https://github.com/rolling-scopes-school/

Apps gremlins.js

Personal Open source Business Explore Pricing

Rolling Scopes School

Rolling Scopes School organisation

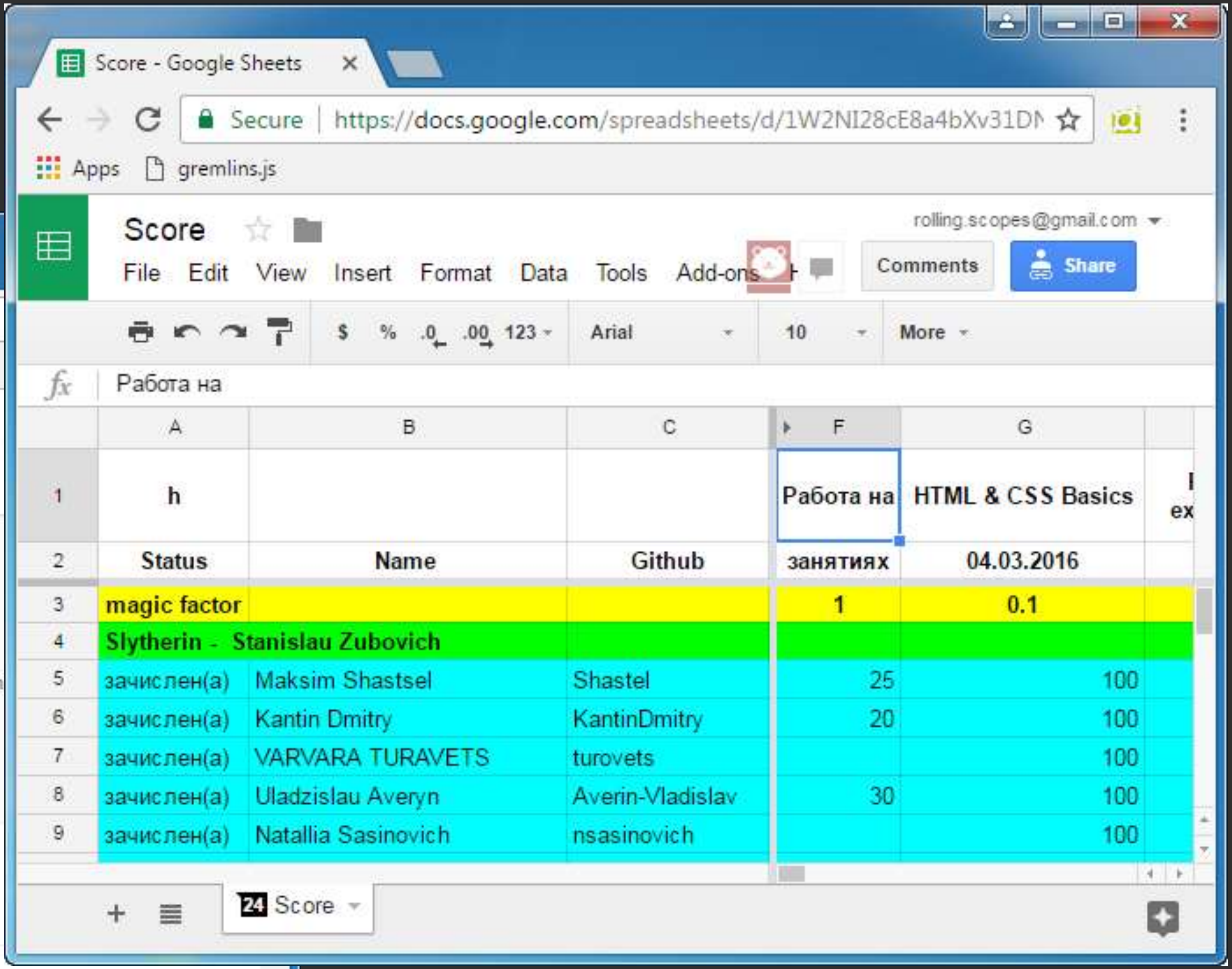
Minsk http://rollingscopes.com/ rolling.scopes@gm

Repositories People 5

Filters Find a repository...

priority-queue JavaScript ★ 0 10

Priority queue task



Score - Google Sheets

Secure | https://docs.google.com/spreadsheets/d/1W2NI28cE8a4bXv31DN

Apps gremlins.js

Score rolling.scopes@gmail.com

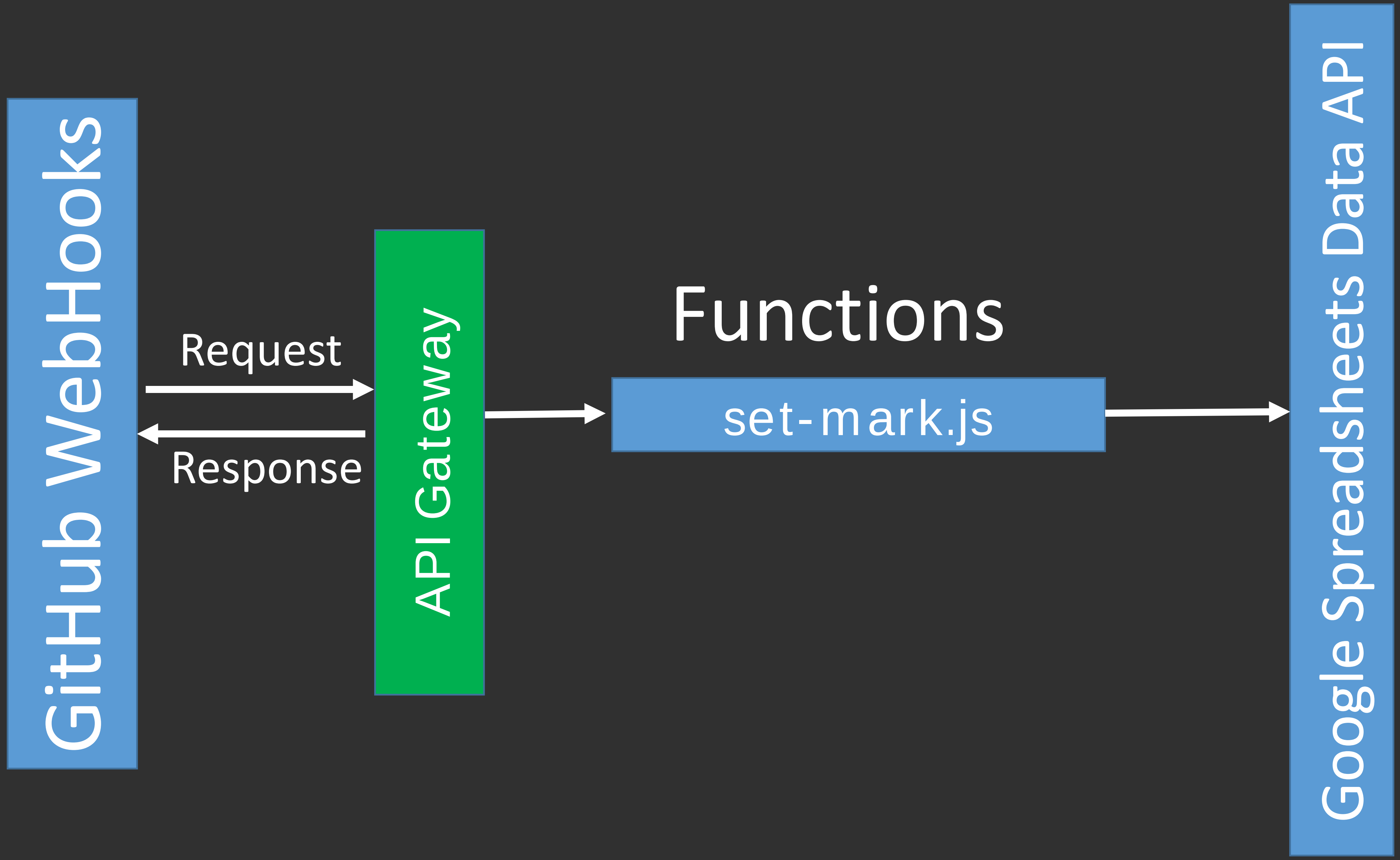
File Edit View Insert Format Data Tools Add-ons Comments Share

Arial 10 More

fx Работа на

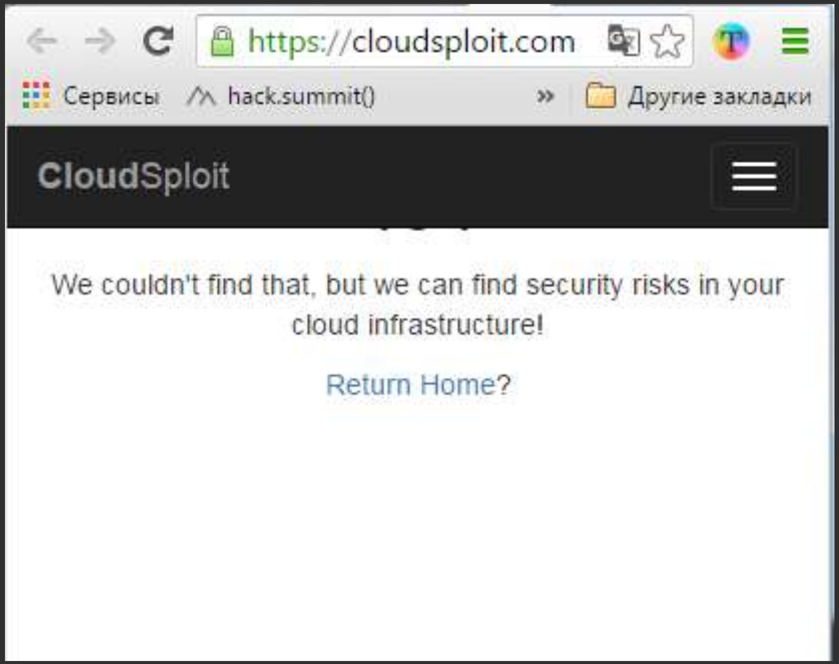
	A	B	C	F	G
1	h			Работа на	HTML & CSS Basics
2	Status	Name	Github	занятиях	04.03.2016
3	magic factor			1	0.1
4	Slytherin - Stanislau Zubovich				
5	зачислен(а)	Maksim Shastsel	Shastel	25	100
6	зачислен(а)	Kantin Dmitry	KantinDmitry	20	100
7	зачислен(а)	VARVARA TURAVETS	turovets		100
8	зачислен(а)	Uladzislau Averyn	Averin-Vladislav	30	100
9	зачислен(а)	Natallia Sasinovich	nsasinovich		100

Score


GitHub WebHooks
Request
Response
API Gateway
Functions
set-mark.js
Google Spreadsheets Data API

Use case #2

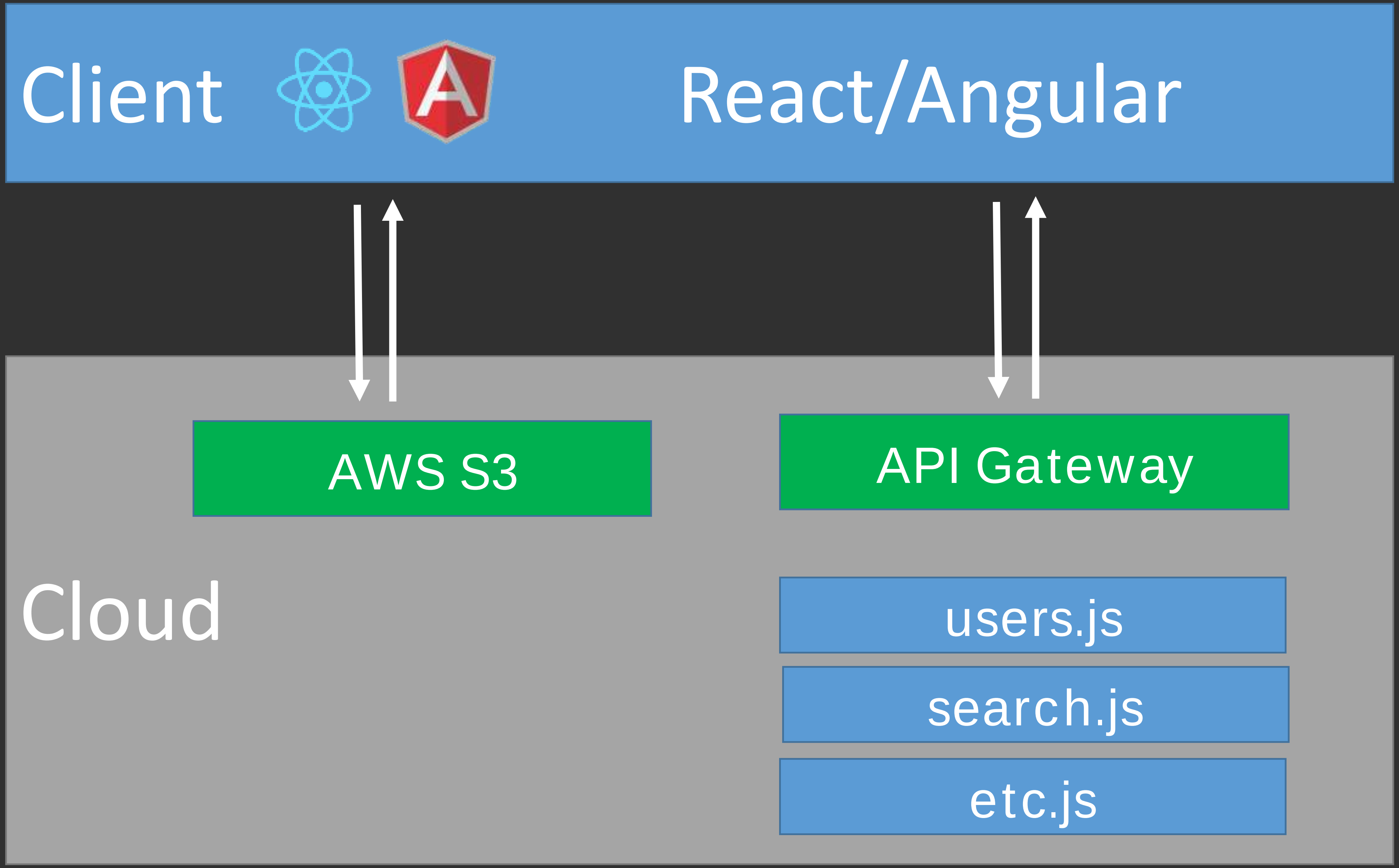
Story: We Made the Whole Company "Serverless"

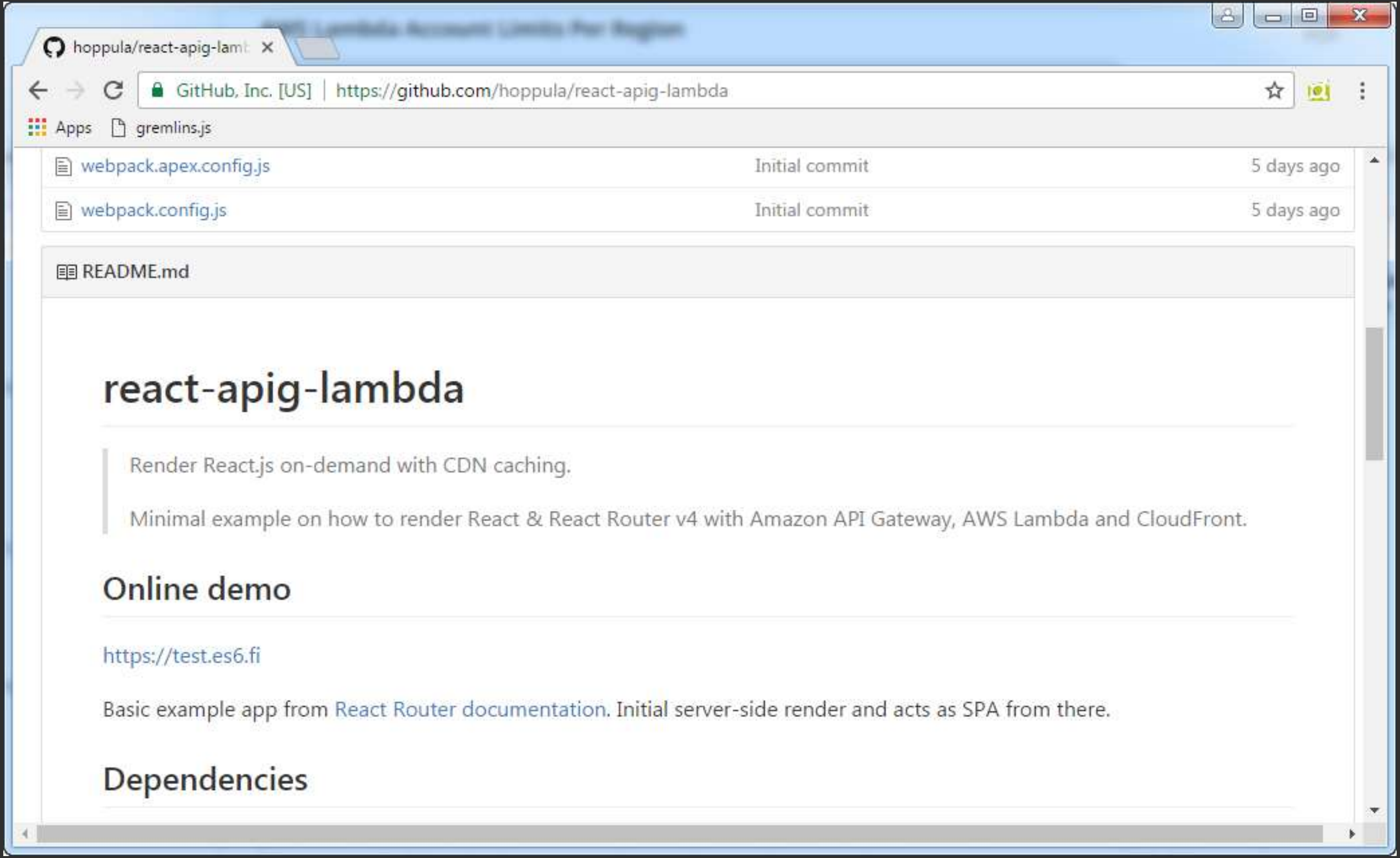


We couldn't find that, but we can find security risks in your cloud infrastructure!

Return Home?

https://blog.cloudsploit.com/we-made-the-whole-company-serverless-5a91c27cd8c4#.asm49udxm


Client
React/Angular
AWS S3
API Gateway
Cloud
users.js
search.js
etc.js



webpack.apex.config.js	Initial commit	5 days ago
webpack.config.js	Initial commit	5 days ago

README.md

react-apig-lambda

> Render React.js on-demand with CDN caching.
>
> Minimal example on how to render React & React Router v4 with Amazon API Gateway, AWS Lambda and CloudFront.

Online demo

https://test.es6.fi

Basic example app from React Router documentation. Initial server-side render and acts as SPA from there.

Dependencies

Function as a Service

Functions are the unit of deployment and scaling

No machines, VMs or containers visible in the programming model

Scales per request, Users cannot over- or under-provision capacity

Service Providers

Amazon Lambda

IBM OpenWhisk

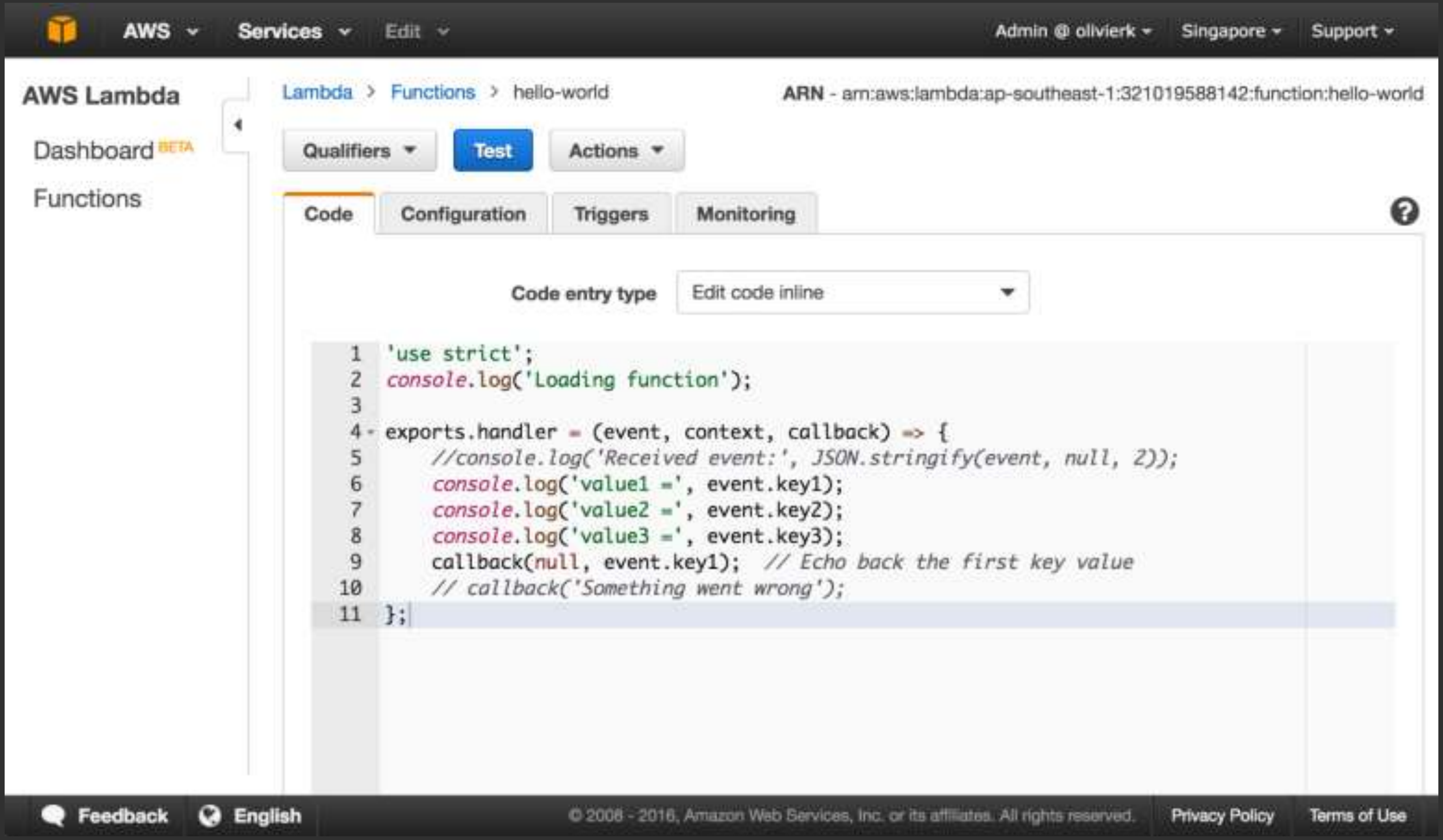
Google Cloud Functions

Microsoft Azure Functions

….

AWS Lambda

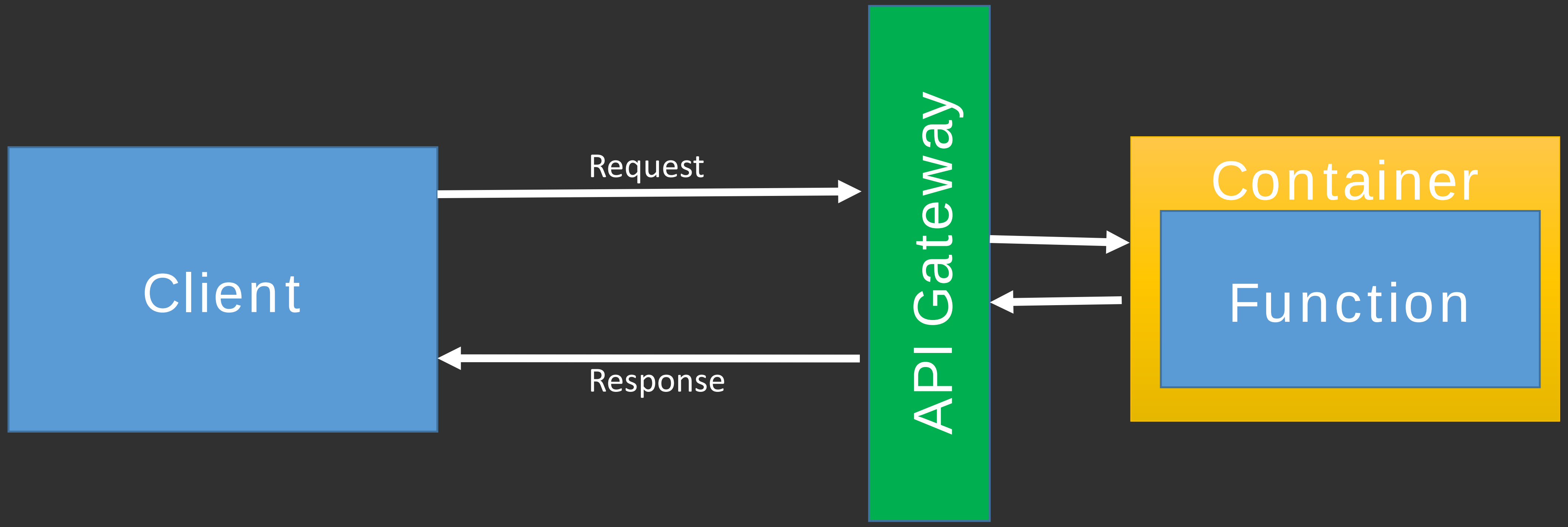
Supports
JavaScript,
Java
Python



```
'use strict';
console.log('Loading function');

exports.handler = (event, context, callback) => {
    //console.log('Received event:', JSON.stringify(event, null, 2));
    console.log('value1 =', event.key1);
    console.log('value2 =', event.key2);
    console.log('value3 =', event.key3);
    callback(null, event.key1);  // Echo back the first key value
    // callback('Something went wrong');
};
```

Lambda container



Start-up latency

Node 10-100 ms

Java ??? ms

Cold start / Hot start / Container reuse

If start time is crucial, create a scheduled function to keep the Lambda function “warm” **(c) AWS**

What is Lambda container?

Resource	Default Limit
Ephemeral disk capacity	512 MB
Invoke request body payload size (RequestResponse)	6 MB
Invoke response body payload size (RequestResponse)	6 MB

http://docs.aws.amazon.com/lambda/latest/dg/limits.html

Maximum execution duration per request: 5 min

AWS Lambda Deployment Limits

MAX 50Mb (zip) per Function

Default concurrent executions limits

Per Region Per Account

100

NoOps?

Serverless != DevOpsLess

AWS CloudFormation

Phoenix Environments

Using automation, we can create whole environments - including network configuration, load balancing and firewall ports - for example by using CloudFormation in AWS. We can then prove that the process works by tearing the environments down and recreating them from scratch on a regular basis.

(c) ThoughtWorks Radar

Fine-Grained Pricing

Free Tier

1M requests and 400,000 GBs of compute.

Every month, every customer.

- Compute time in 100ms increments
- Low request charge
- No hourly, daily or monthly minimums
- No per-device fees

- **Never pay for idle!**

Gigabytes per second (GBps)

Price =

function's RAM size X execution time (sec)

Example: 1GB RAM X 5 min = 0.005001$

ROLLINGSCOPES.COM ☺

Billing: $11.43/month

Instance utilization: 4%/month ☹

Fond of performance optimization?

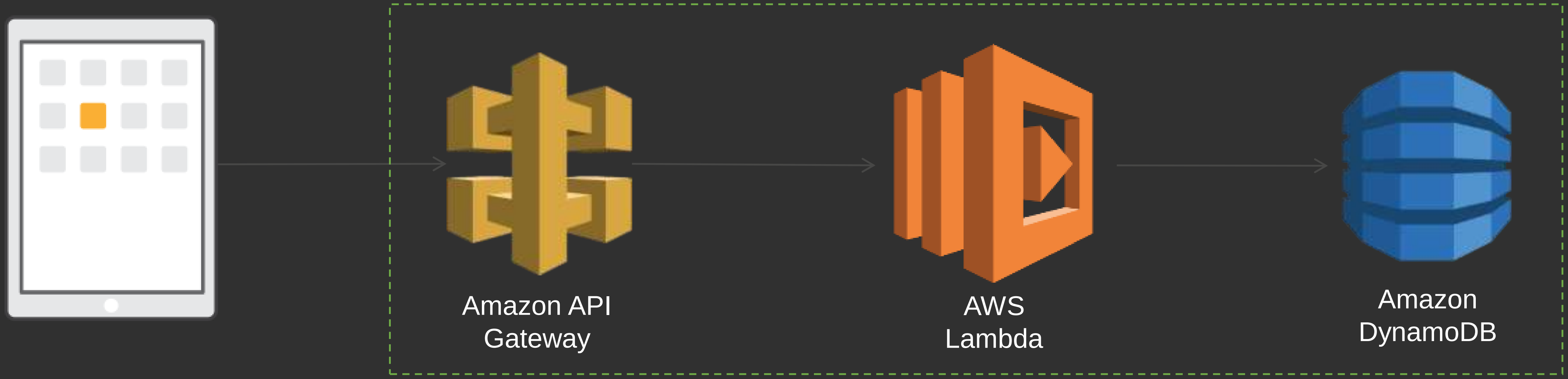
Direct relationship between cost and performance ☺

Serverless architecture

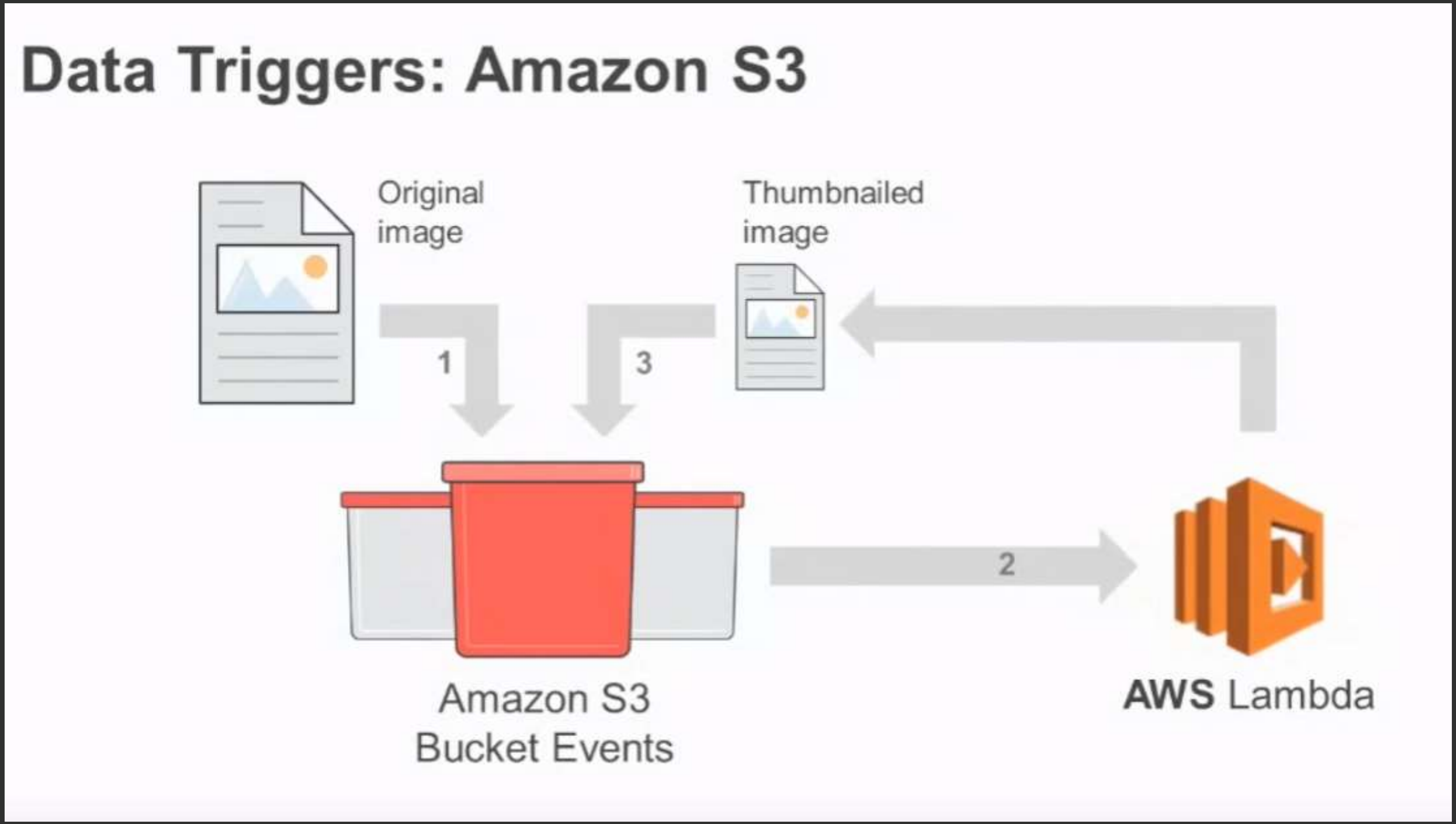
Serverless was first used to describe applications that significantly or fully depend on 3rd party applications / services (‘in the cloud’) to manage server-side logic and state.

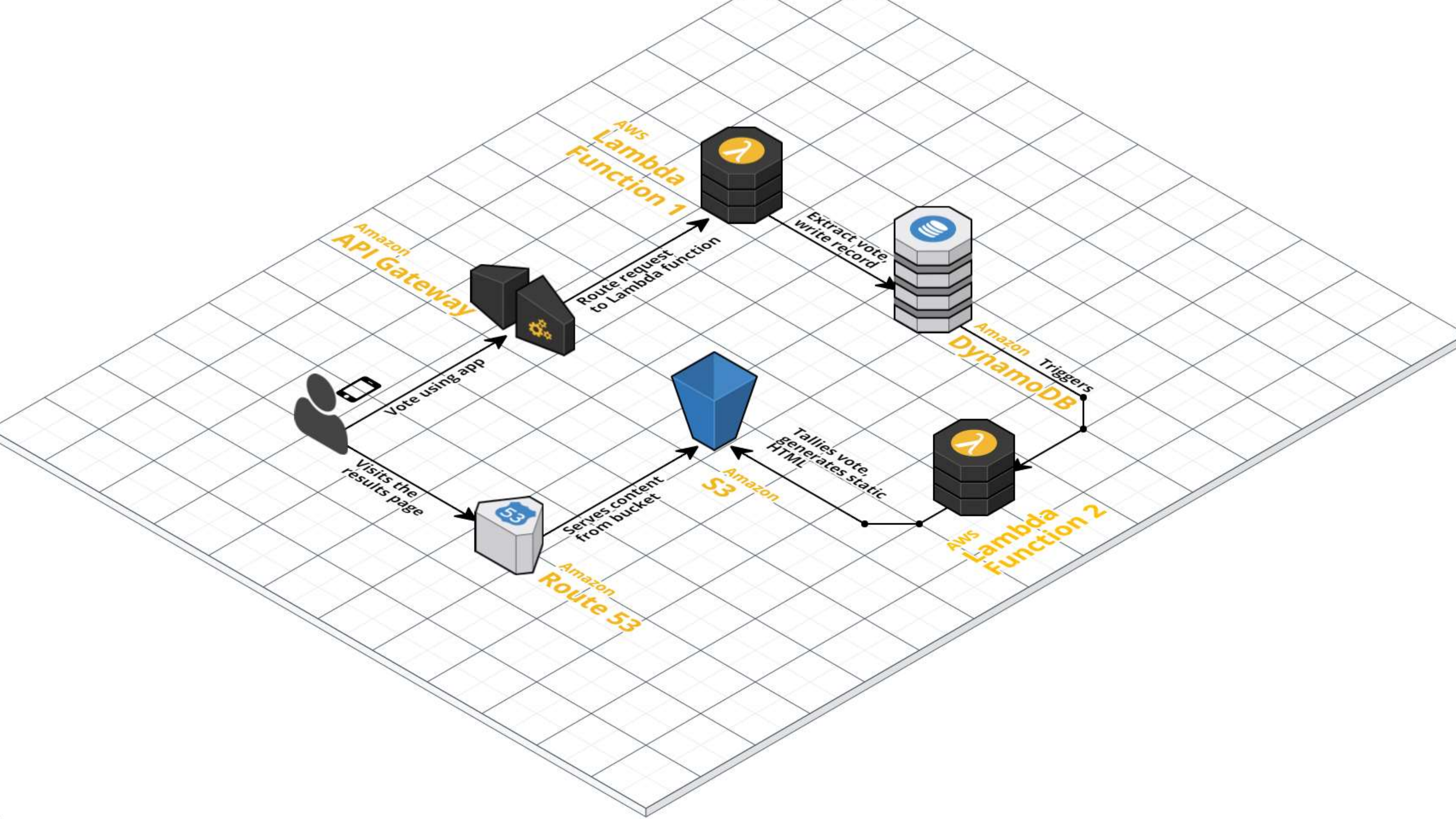
http://martinfowler.com/articles/serverless.html

Usual scenario (for web)

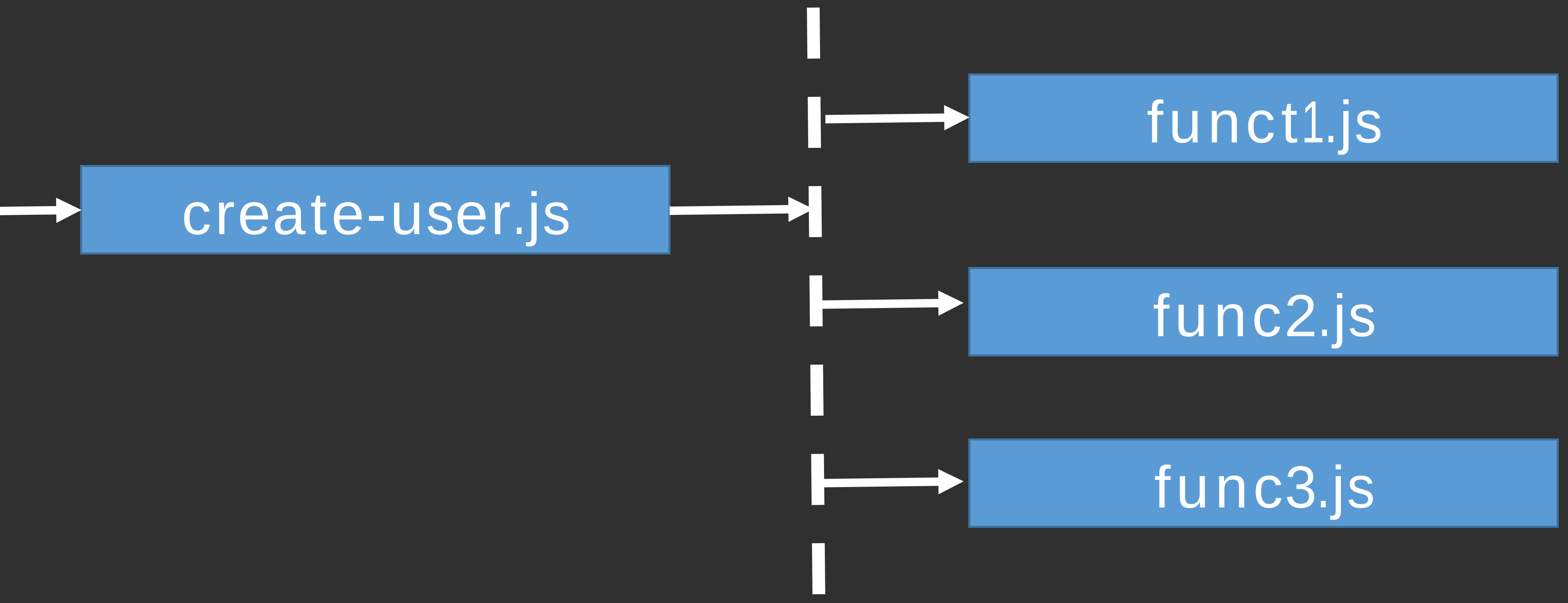


Event-driven Compute in the Cloud




AWS
Lambda
Function 1
Amazon
API Gateway
Route request
to Lambda function
Extract vote,
write record
Amazon
DynamoDB
Triggers
Vote using app
Tallies vote,
generates static
HTML
Amazon
S3
Visits the
results page
Serves content
from bucket
Amazon
Route 53
AWS
Lambda
Function 2

SNS Events



Topic name

Serverless Reference Architectures with AWS Lambda

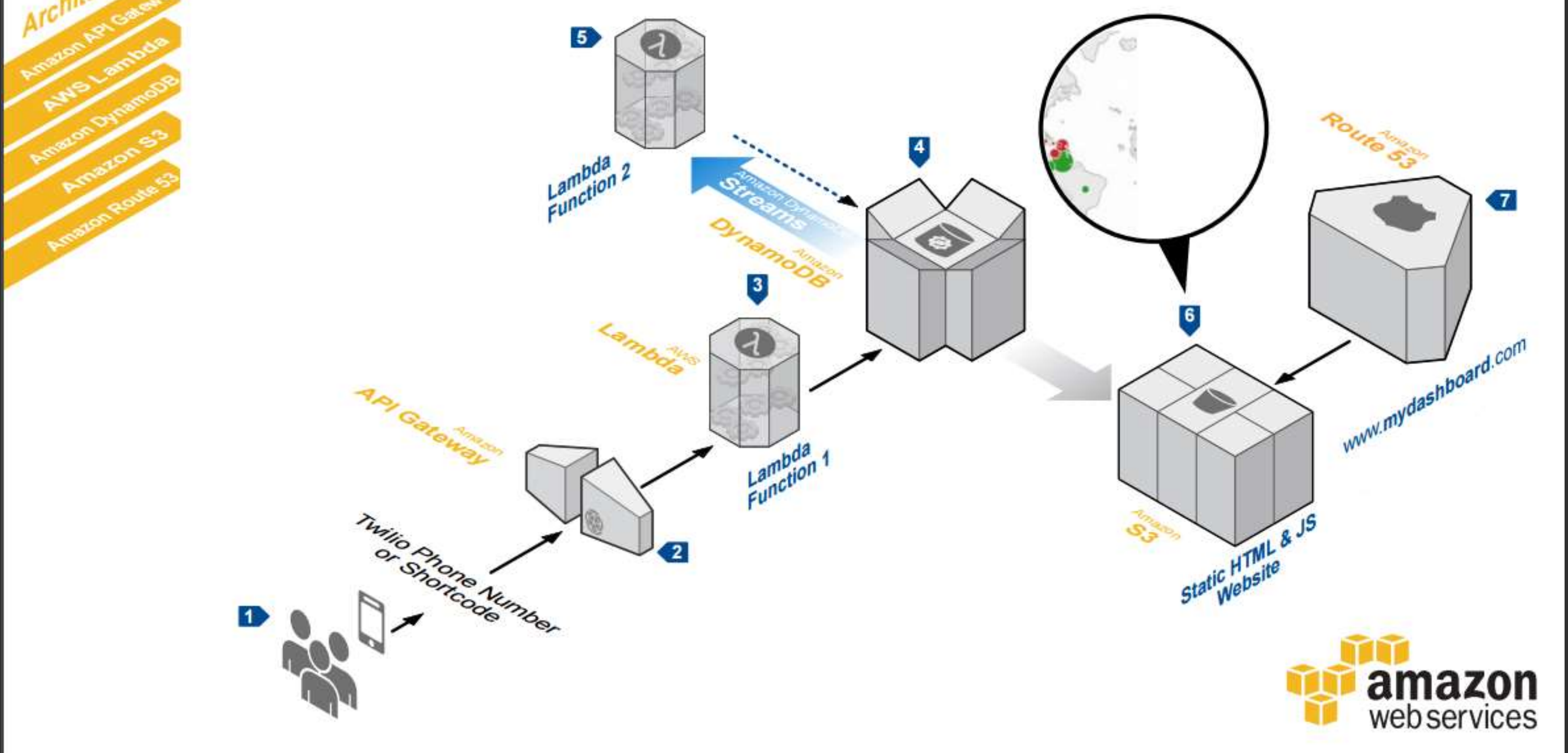
- Web Applications Serverless Reference Architecture
- Mobile Backend Serverless Reference Architecture
- Real-time File Processing Serverless Reference Architecture
- IoT Backend Serverless Reference Architecture
- Real-time Stream Processing Serverless Reference Architecture

http://www.allthingsdistributed.com/2016/06/aws-lambda-serverless-reference-architectures.html

AWS LAMBDA: REAL-TIME VOTING APPLICATION

Consider a dynamic web application that receives votes in real-time. Traditionally, architecting such applications meant building out your infrastructure to support both "spiky" and sustained usage over a finite amount of time. In most cases, this required your operations team to overprovision resources, leading to waste outside of high-volume voting periods.

By combining AWS Lambda with other serverless AWS services such as Amazon API Gateway, you can build a powerful, highly available web application that automatically scales up and down to handle large amounts of concurrent votes - all with zero administrative effort required. You can also store and analyze your data in fault-tolerant services like Amazon DynamoDB and Amazon Simple Storage Service.



System

1 Users text a vote to a phone number or shortcode provided by a third party like Twilio.

function extracts the vote from the message content and writes the result and any

4 metadata into a table in **Amazon DynamoDB**. This table has DynamoDB Streams enabled, which allows you to track changes to your tables on a rolling basis.

6 A dashboard to display a summary of votes is created using HTML and JavaScript, and hosted as a static website in **Amazon Simple Storage Service (Amazon S3)**. uses the AWS Javascript SDK to query the DynamoDB table and display the voting results in

ly, **Amazon Route 53** is used as a DNS provider to create a hosted zone pointing a custom domain name to the Amazon S3 bucket.

https://github.com/awslabs/lambda-refarch-webapp

Drawbacks

Massive Vendor Lock-In

How to debug?

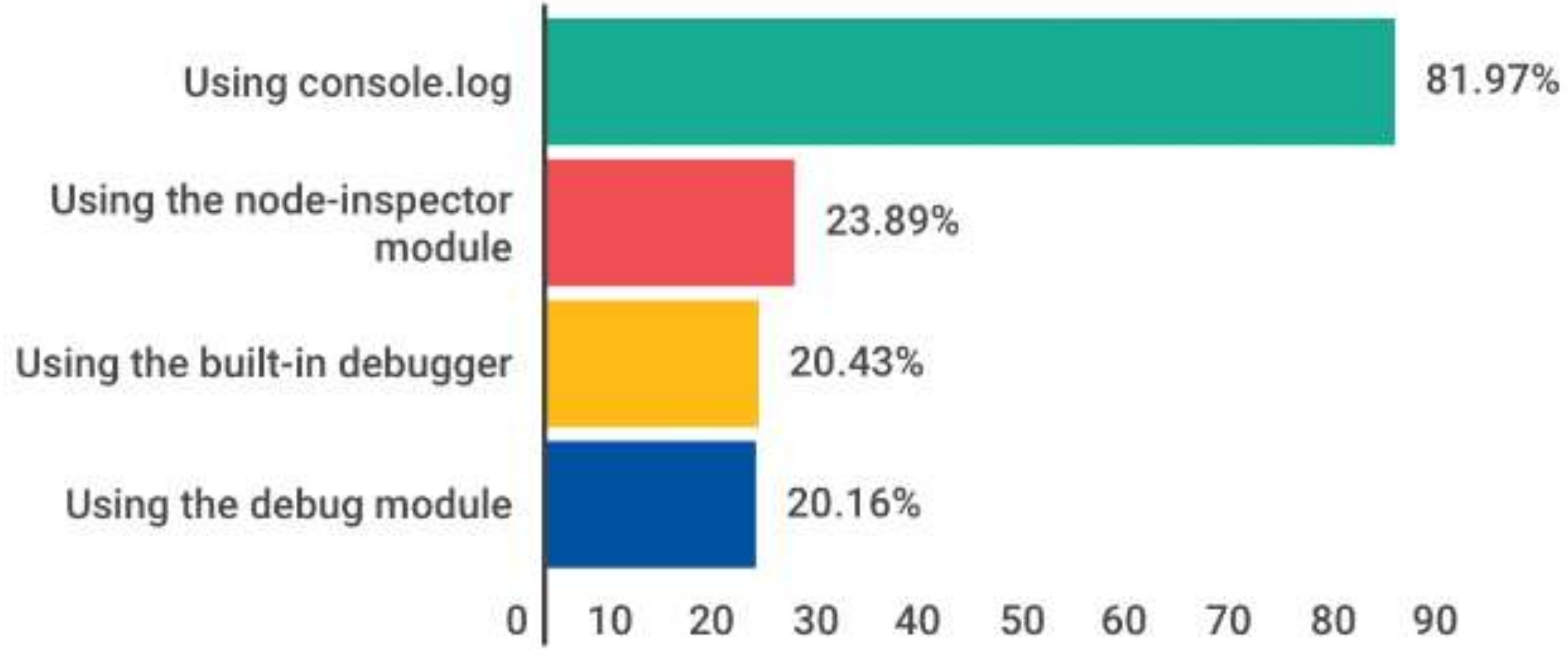
“It is also INCREDIBLY irritating and challenging to debug in a practically infinite number of insanity-inducing ways”

“You cannot ssh into your Lambda.”

“You cannot inspect your program as it’s running. Forget debugging techniques that we’ve accumulated over the years, they don’t apply in this post-Kansas serverless world. Forget dtrace, forget perf, forget flamegraphs. Your debugging arsenal is limited to good old printf debugging. No, scratch that, I should’ve said “good old CloudWatch” debugging, which is a few notches below printf-debugging, because it lags. Hooray!”

How do you debug your applications?

1126 respondents - multiple choice answers



https://blog.risingstack.com/node-js-developer-survey-results-2016/

Serverless === Stateless

No in-proc caching or state

Limited Environments

- No local env
- One account for QA, UAT, PROD
- Integration testing difficulties

- Self DDOS ☺

Thank you!

Useful links

AWS Free Tier

https://aws.amazon.com/ru/free/

AWS Lambda Developer Guide

http://docs.aws.amazon.com/lambda/latest/dg/lambda-dg.pdf

Serverless Architectures

http://martinfowler.com/articles/serverless.html

Useful links

What is AWS?
https://www.youtube.com/watch?v=DERzYnthq1s

Your First Week on Amazon Web Services
https://www.youtube.com/watch?v=7CiHBcqw6zc

https://github.com/JustServerless/awesome-serverless
https://github.com/donnemartin/awesome-aws

https://acloud.guru/course/serverlessconf-nyc-2016/dashboard

Useful links

AWS May 2016 Webinar Series - Deep Dive on Serverless Web Applications

https://www.youtube.com/watch?v=fXZzVzptkeo

AWS April Webinar Series - AWS Lambda: Event-driven Code for Devices and the Cloud

https://www.youtube.com/watch?v=YEWtQsqIYsk