8½ THINGS ABOUT SERVERLESS

WITH NODE.JS

SLOBODAN STOJANOVIĆ

CTO AT CLOUD HORIZON AND
JS BELGRADE MEETUP ORGANIZER

twitter.com/slobodan_

github.com/stojanovic

#holyjs
@slobodan_


ANOTHER SERVERLESS TALK?

SOME OF THE QUESTIONS I WILL TRY TO ANSWER TODAY

- WHAT IS SERVERLESS?
- HOW DOES IT WORK?
- WHAT ARE THE SERVERLESS PLATFORMS?
- HOW DOES IT WORK WITH NODE.JS?
- HOW TO DEBUG SERVERLESS FUNCTION?
- WHEN SHOULD YOU USE IT?
- IS SERVERLESS SECURE?


I'm ready.

1. WHAT IS SERVERLESS?

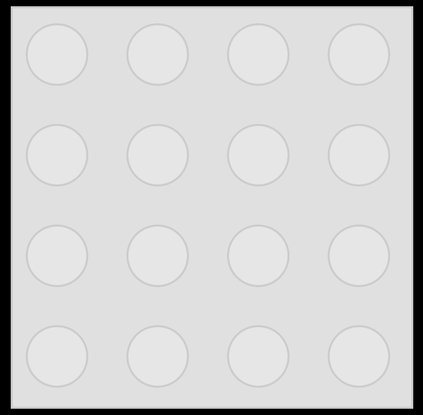
PLATFORM-AS-A-SERVICE (PaaS)

FUNCTION-AS-A-SERVICE (FaaS)

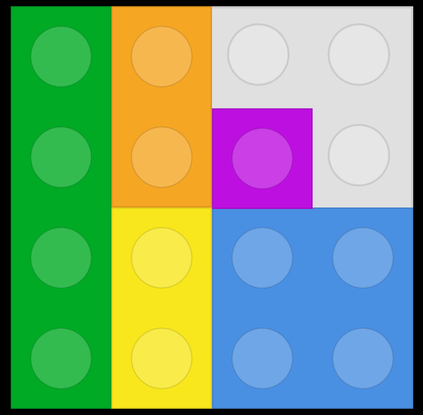

LET'S TRY TO EXPLAIN IT WITH
SOMETHING AS SIMPLE AS
LEGO BRICKS

#holyjs
@slobodan_

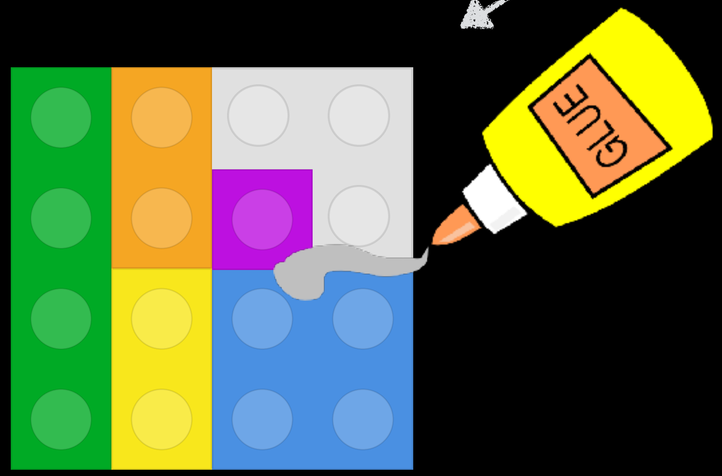
Imagine Lego 4x4 white piece
is a physical server



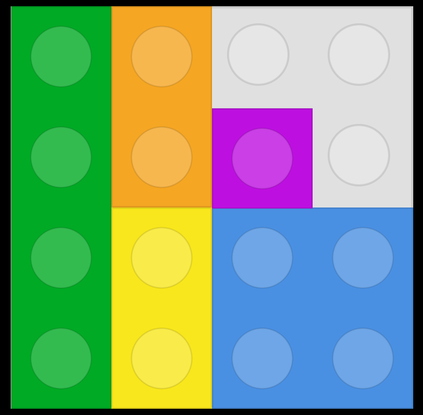
Building and deploying monolith app



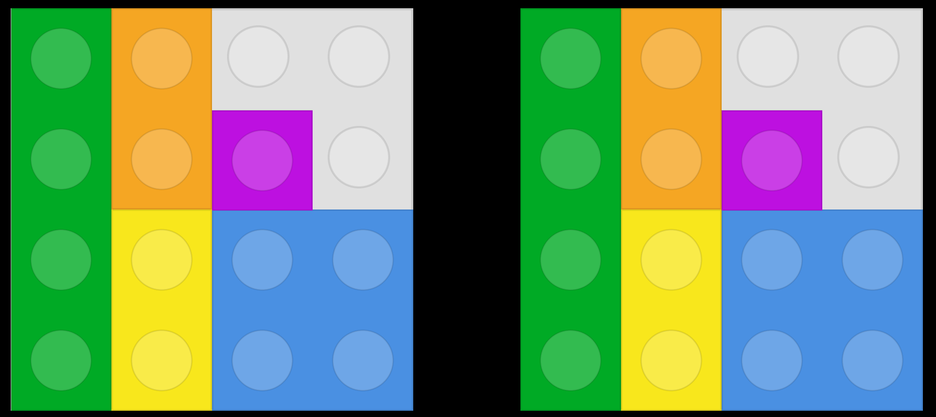
Building and deploying monolith app



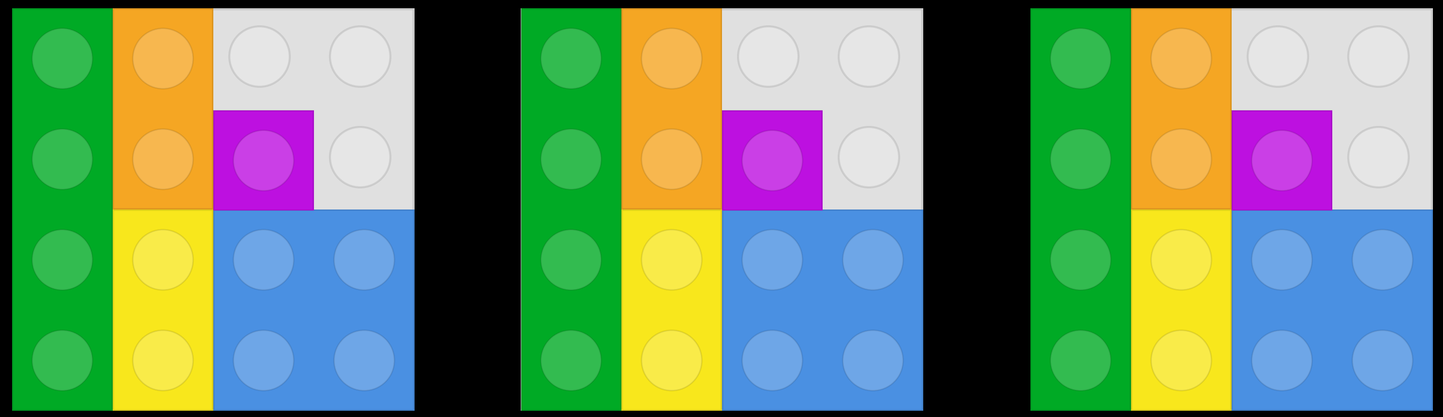
Then people start using it and you need to scale



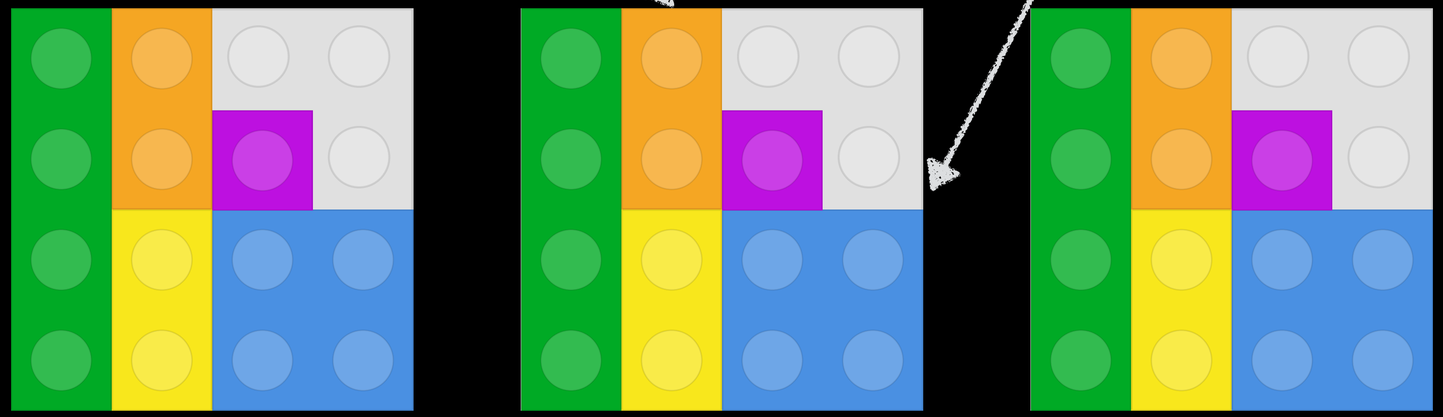
Then people start using it and you need to scale



Then people start using it and you need to scale

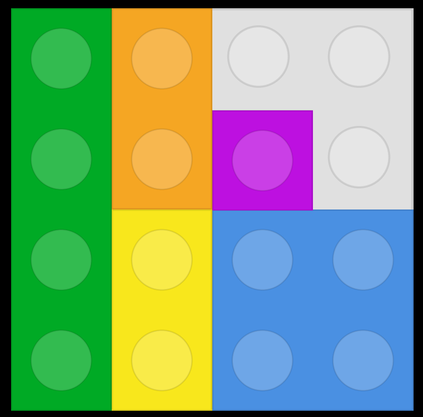


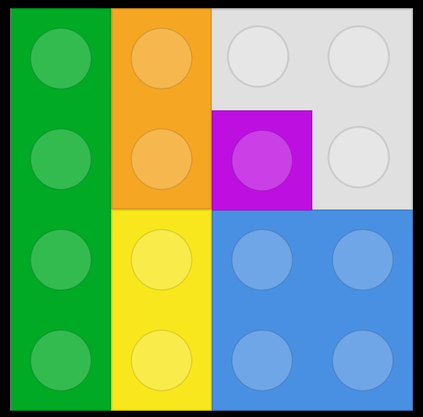
You need to scale everything,
but only green and blue parts
are really used



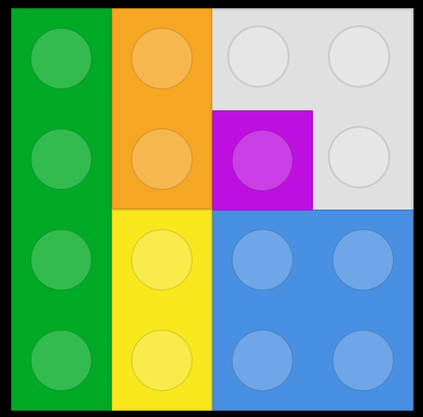
If one part of the app fails whole system will fail

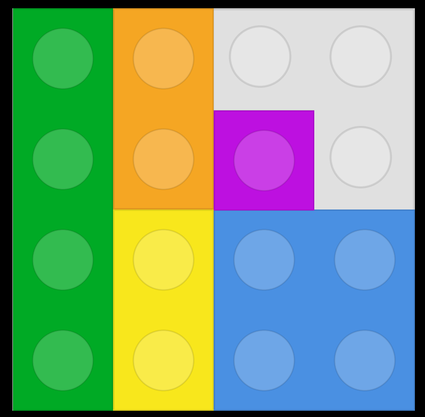
If one part of the app fails whole system will fail





If one part of the app fails whole system will fail

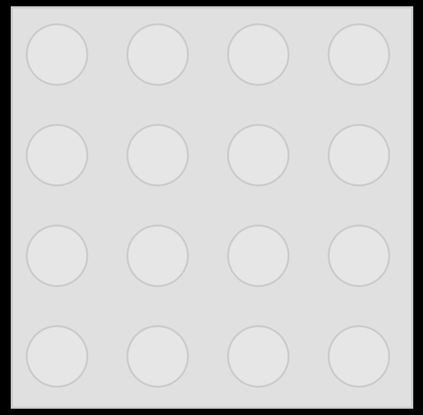




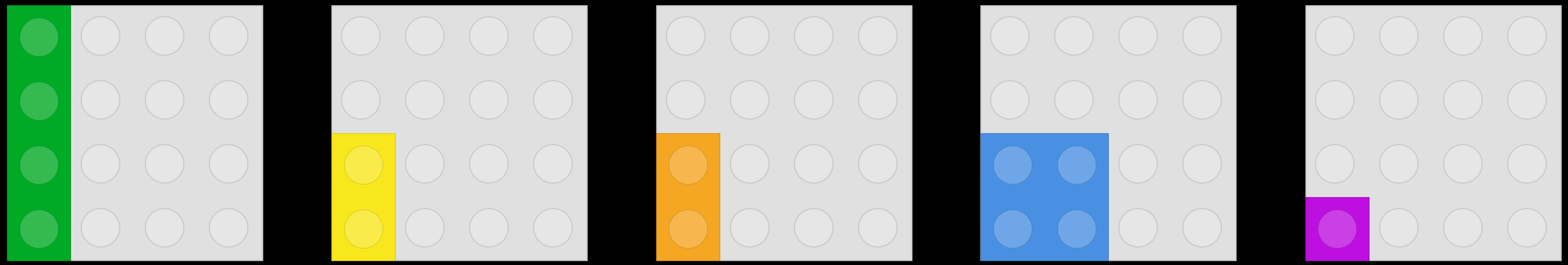
MICROSERVICES
TO THE RESCUE

#holyjs
@slobodan_

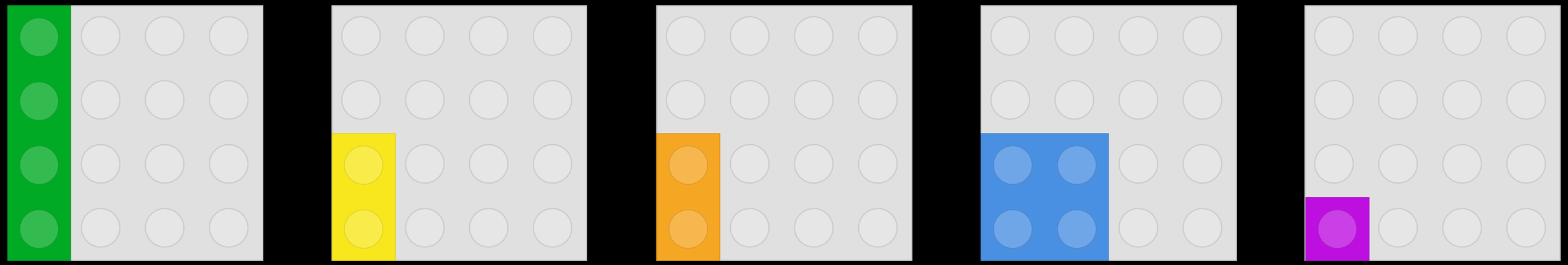
Hello, from our physical server, again!
Now it can be in the cloud, it doesn't matter

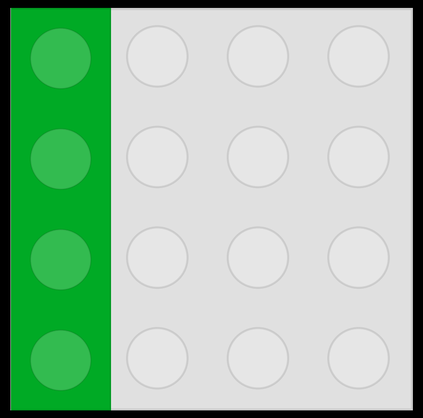


Now each service should be independent, right?

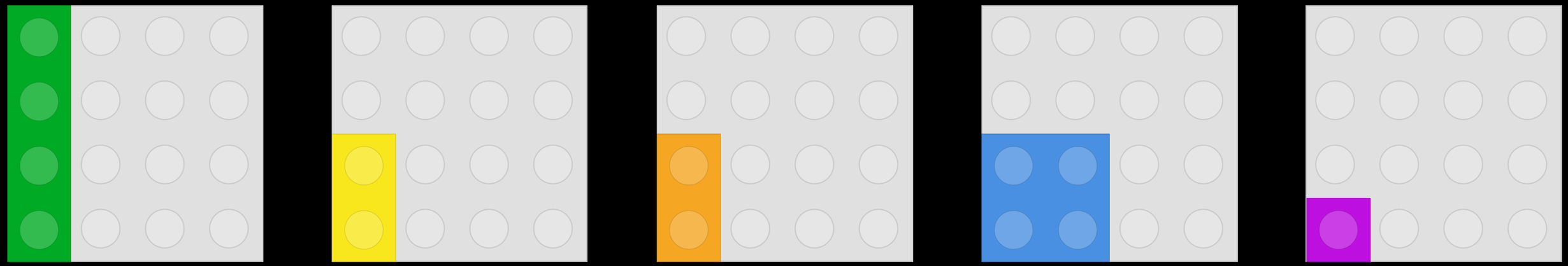


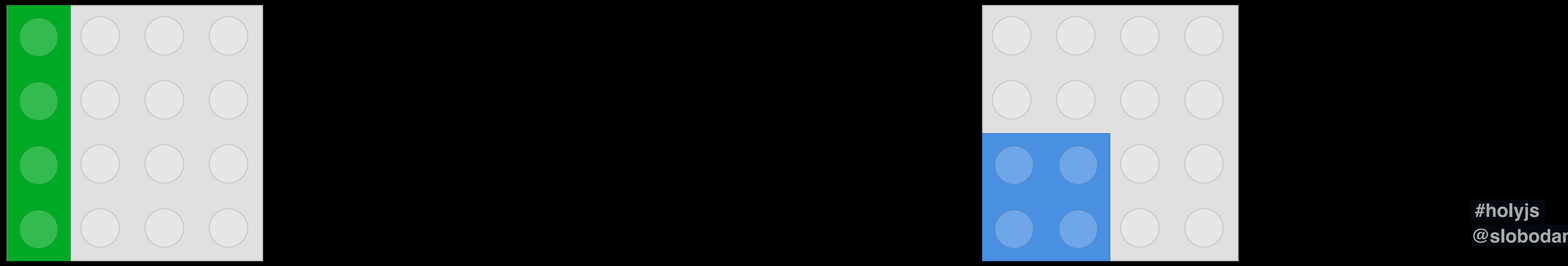
Scaling is easy



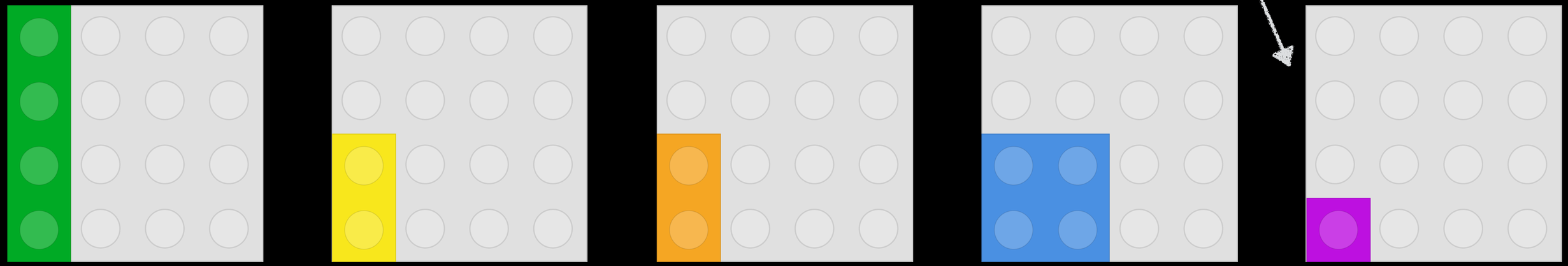


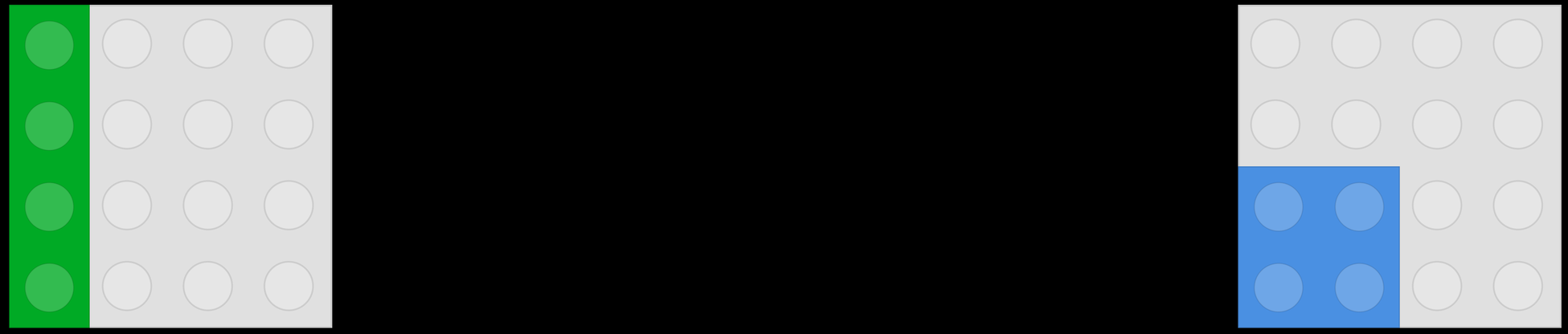
Scaling is easy





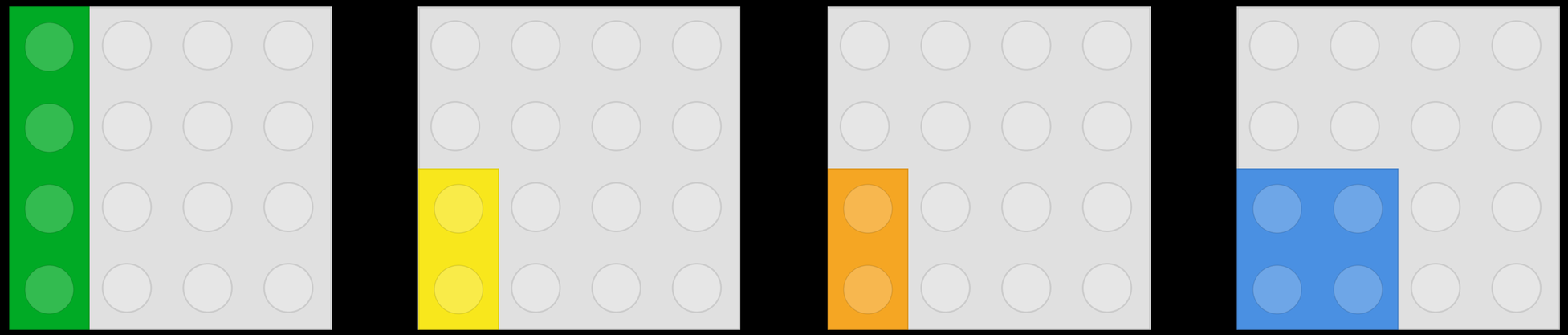
And if no one is using some service
it's simple to remove or replace it

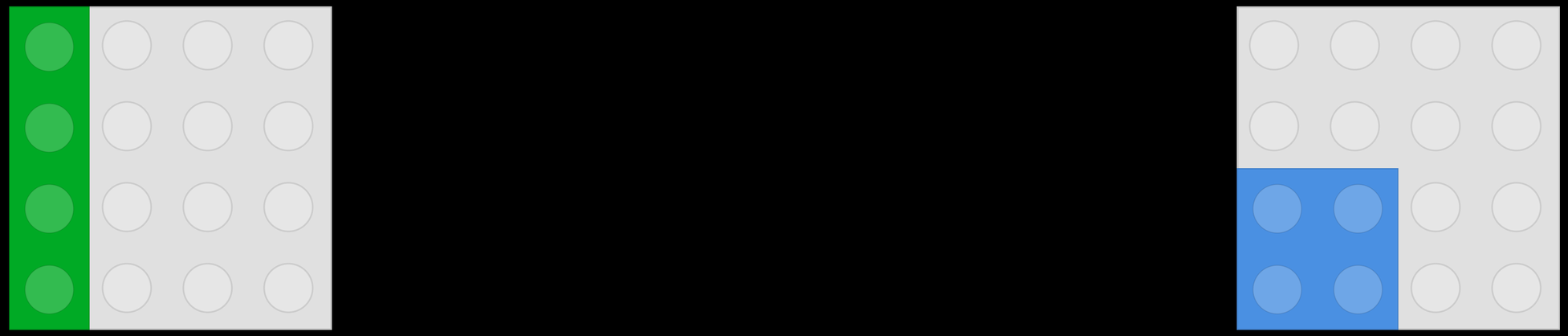




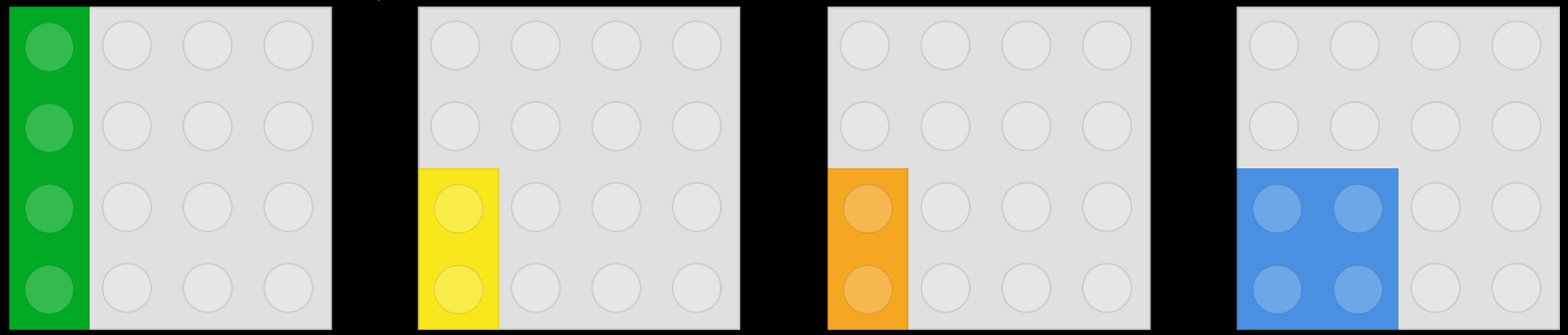
#holyjs
@slobodan_

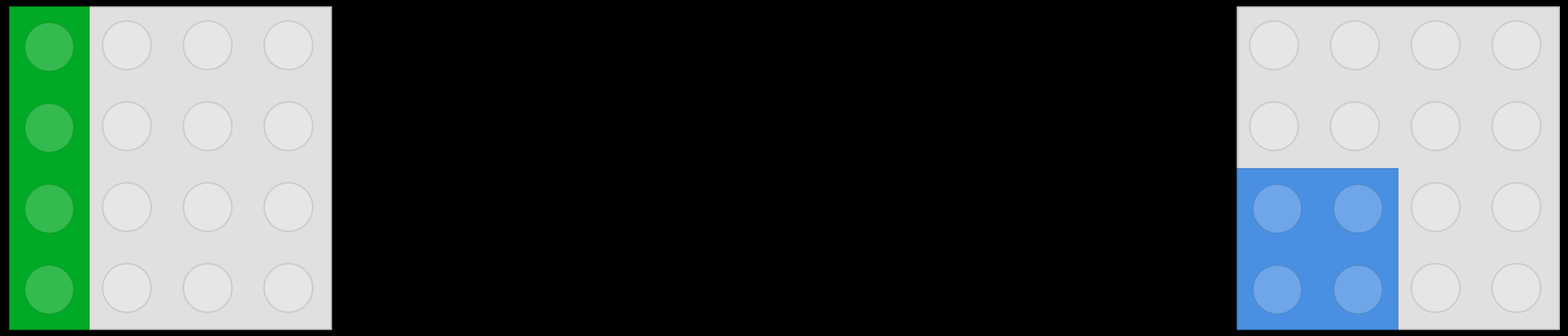
And if no one is using some service
it's simple to remove or replace it



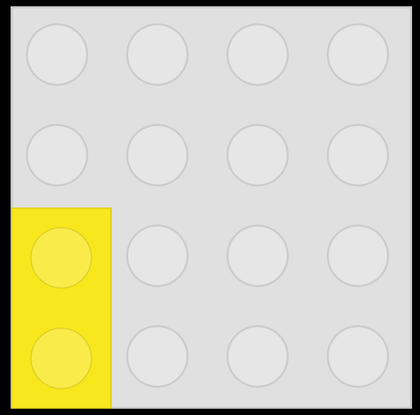


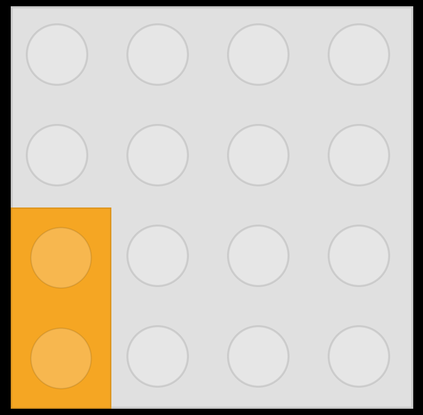
And if some service fails,
it doesn't affect the system

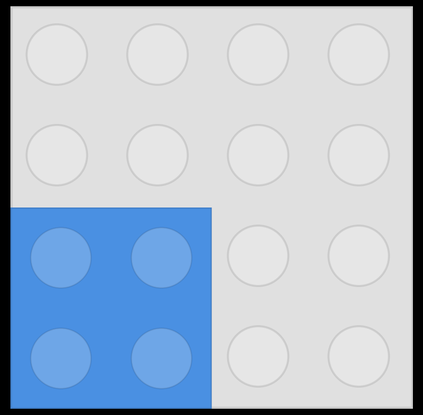


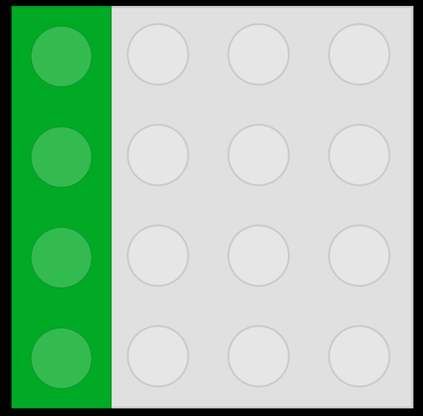


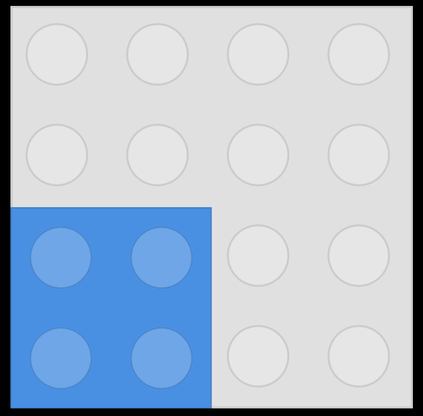
And if some service fails,
it doesn't affect the system



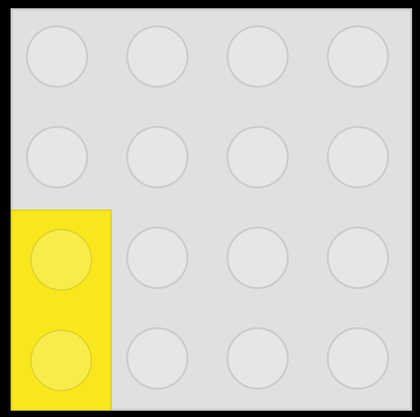


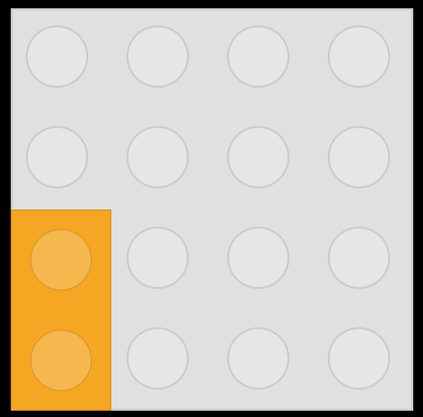


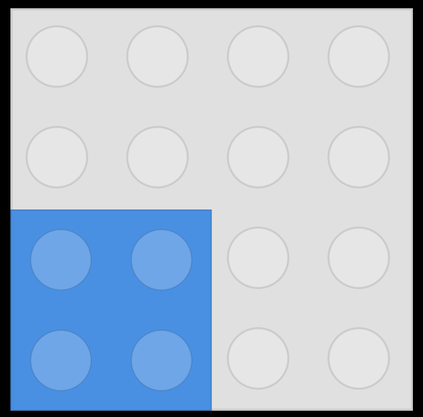


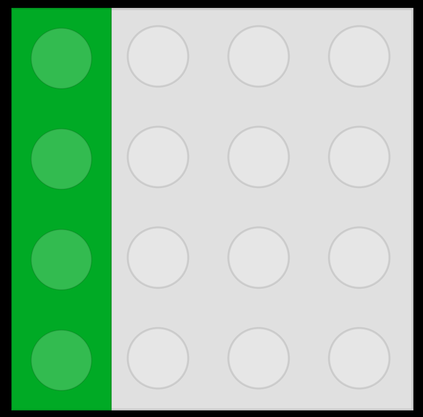


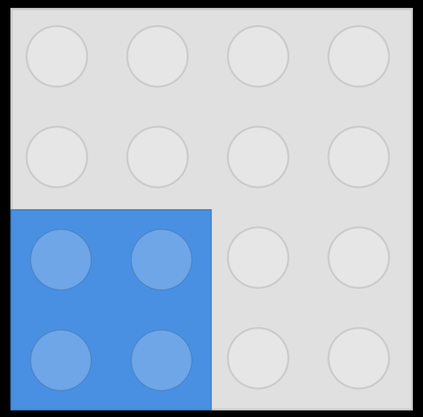
And if some service fails,
it doesn't affect the system





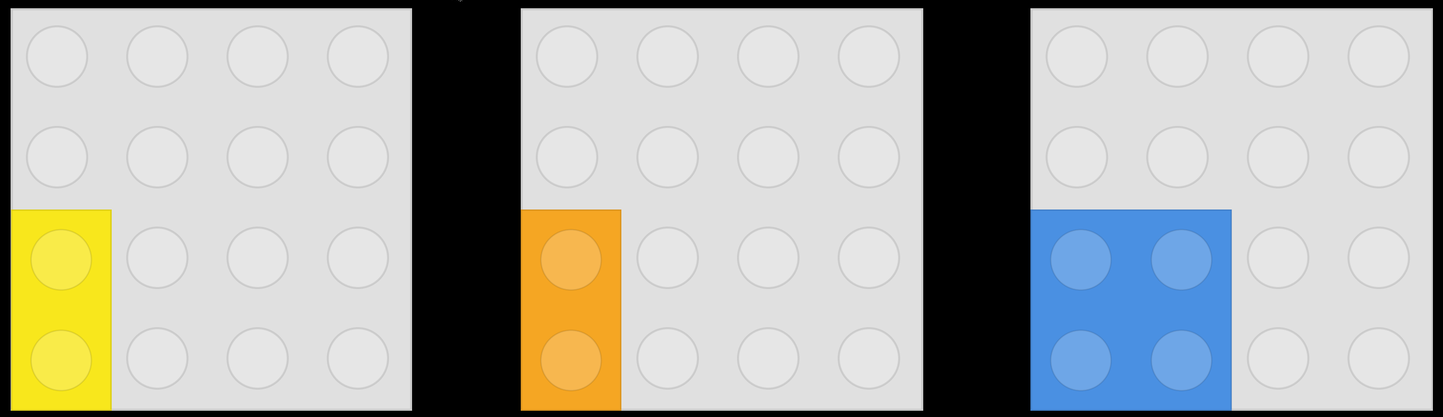


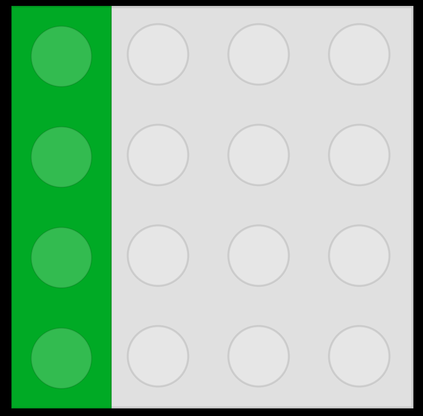


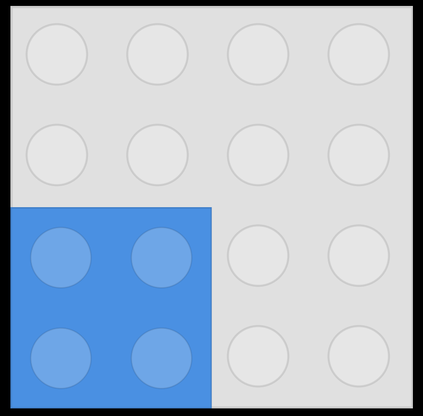


#holyjs
@slobodan_

But what if we didn't scaled it before it fails?

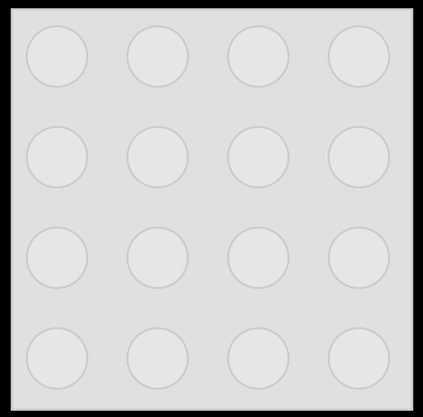




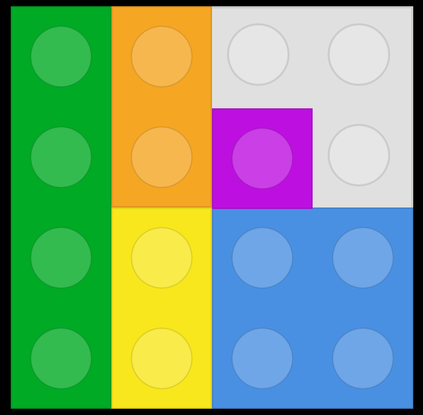


ALSO, ARE SERVERS THAT CHEAP?

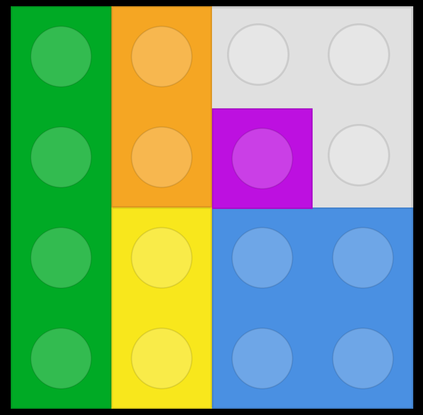
How do we make microservices cheaper?



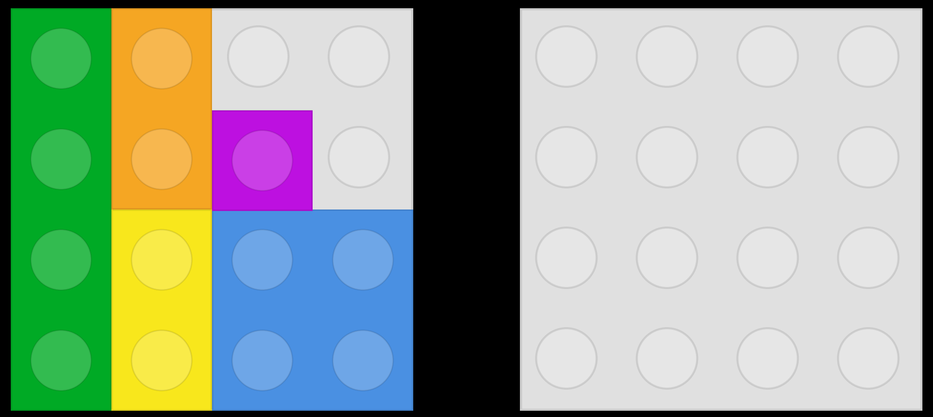
How do we make microservices cheaper?



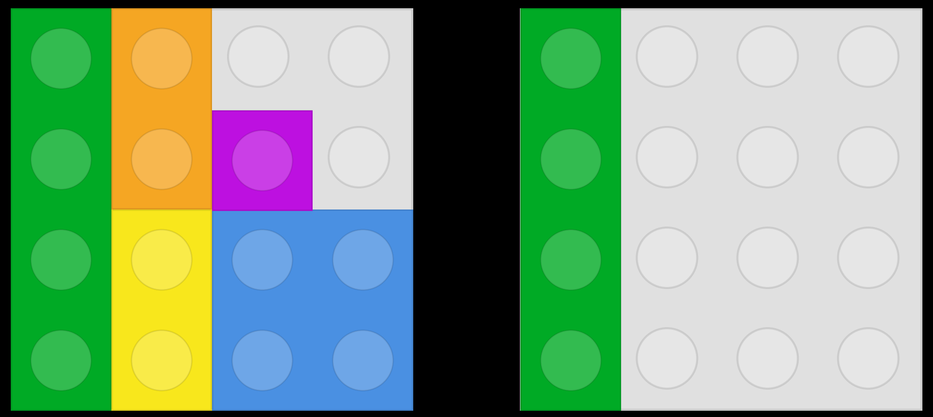
By bundling them together



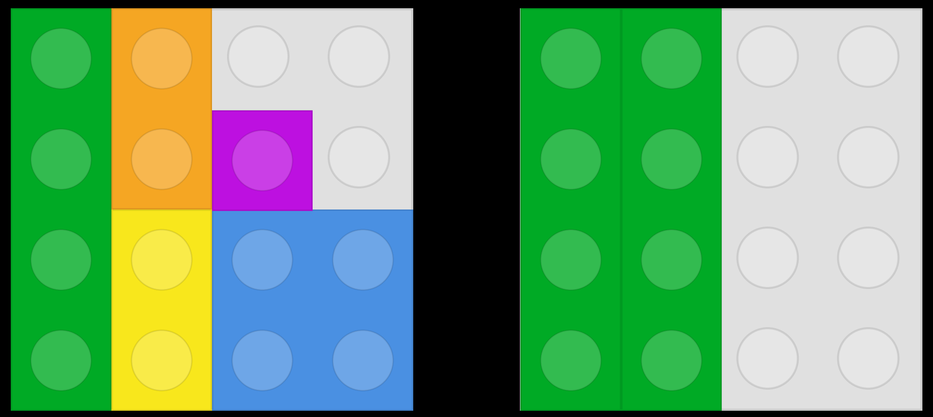
By bundling them together



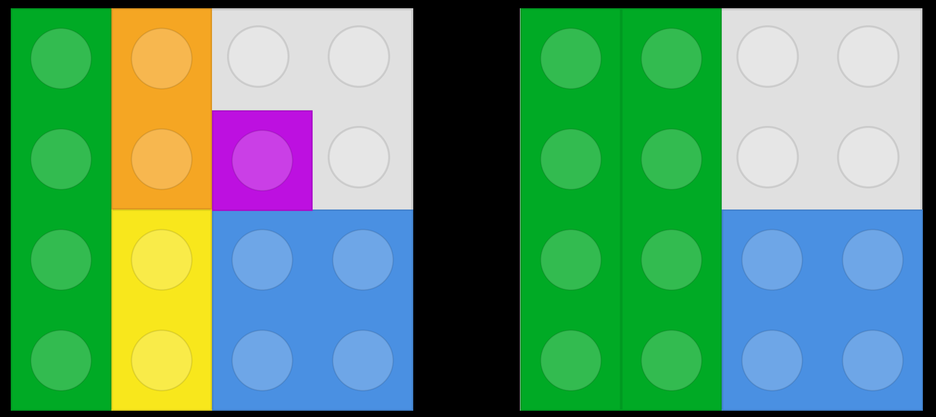
By bundling them together



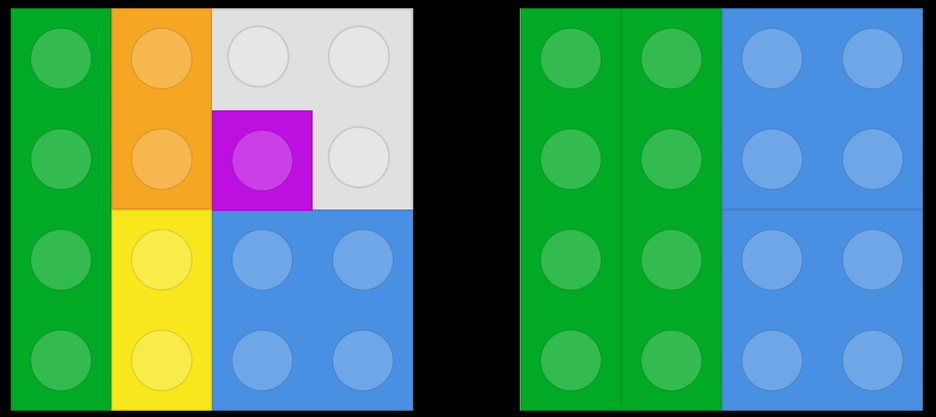
By bundling them together



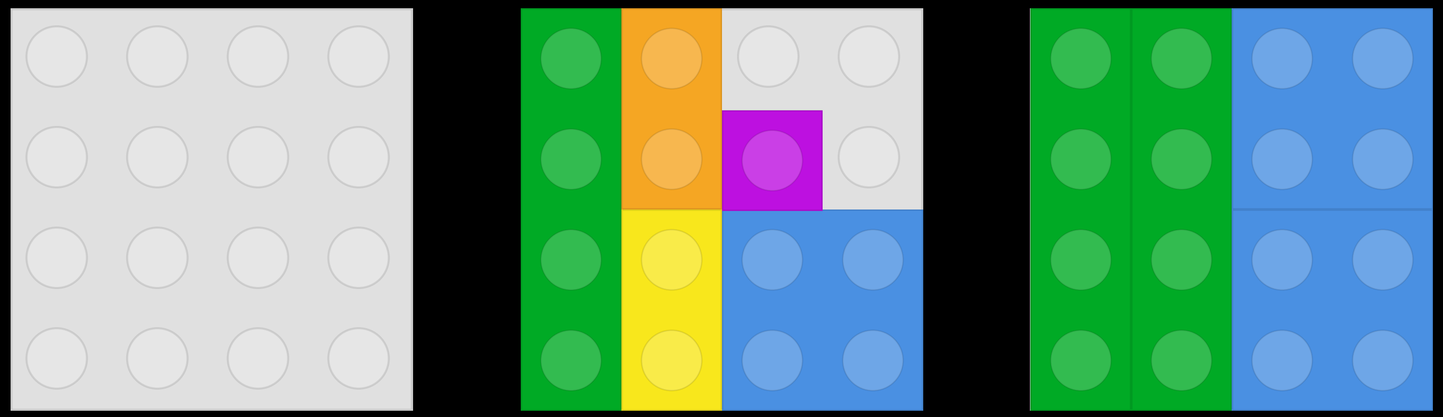
By bundling them together



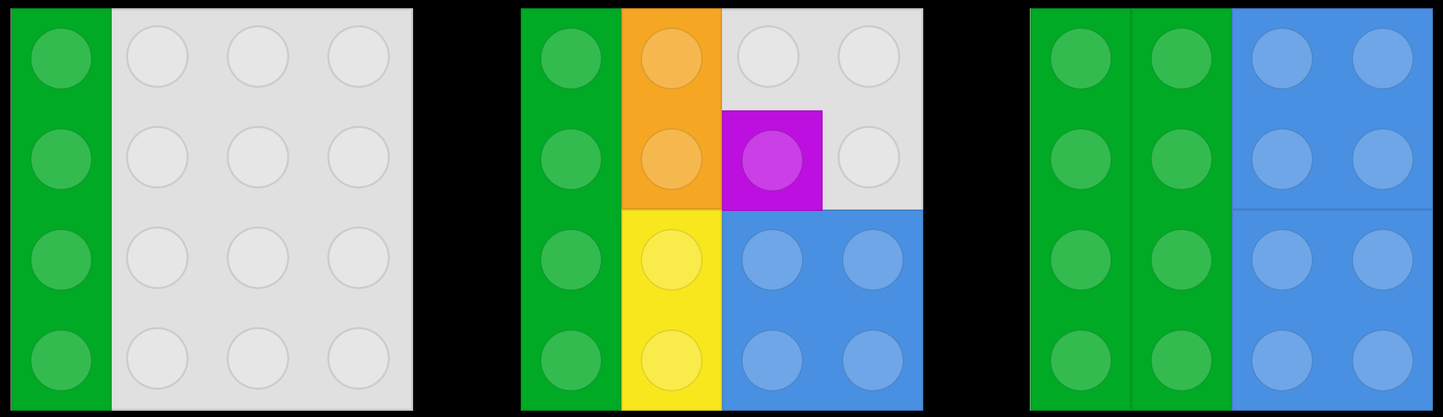
By bundling them together



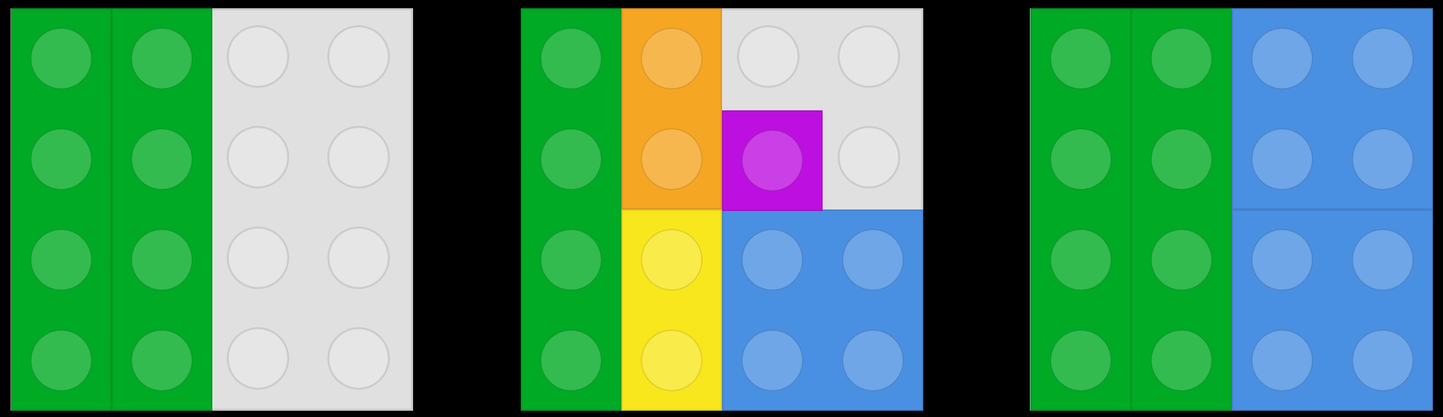
By bundling them together



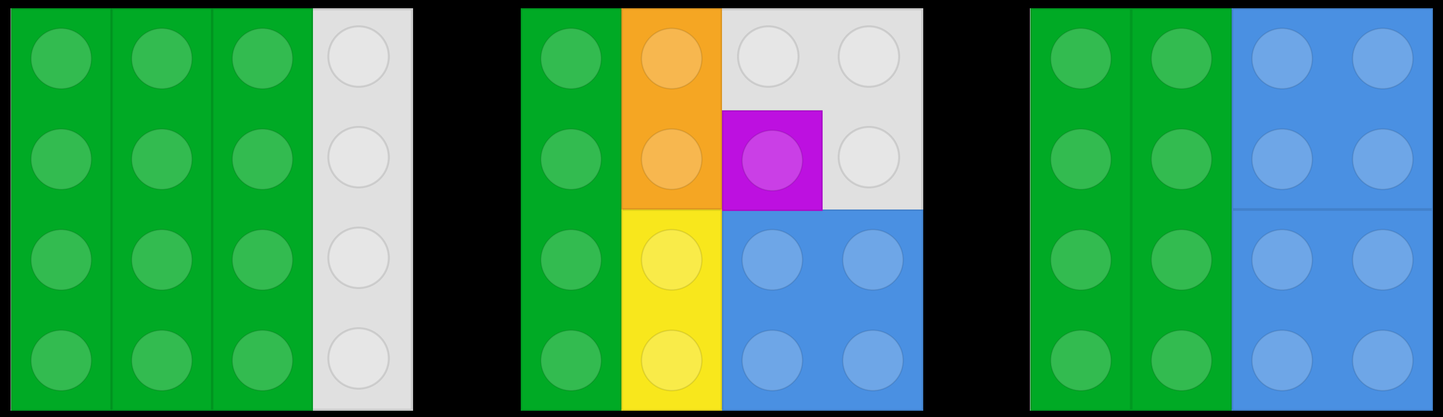
By bundling them together



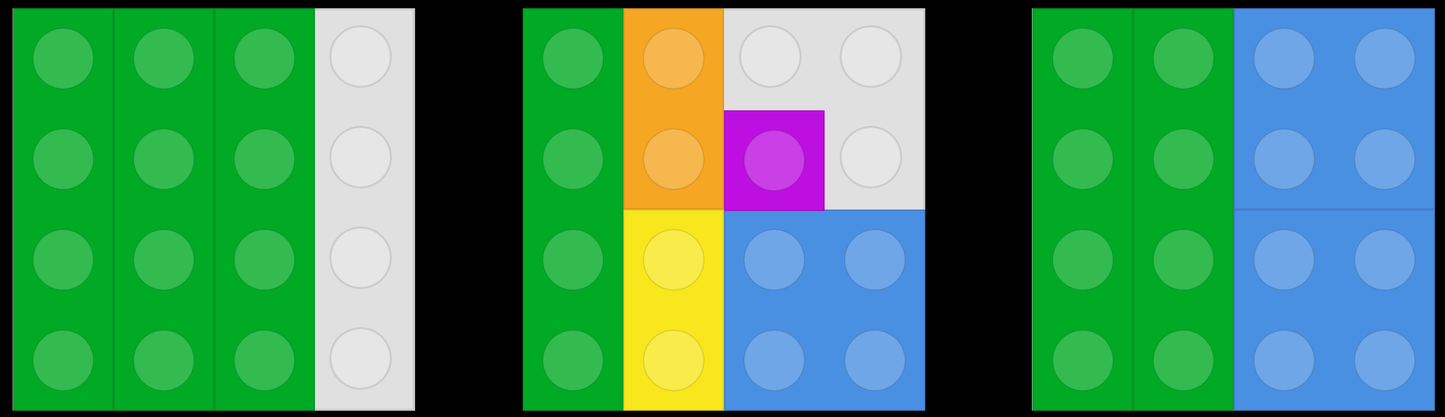
By bundling them together

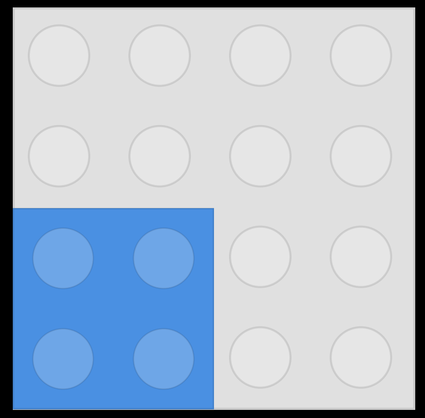


By bundling them together

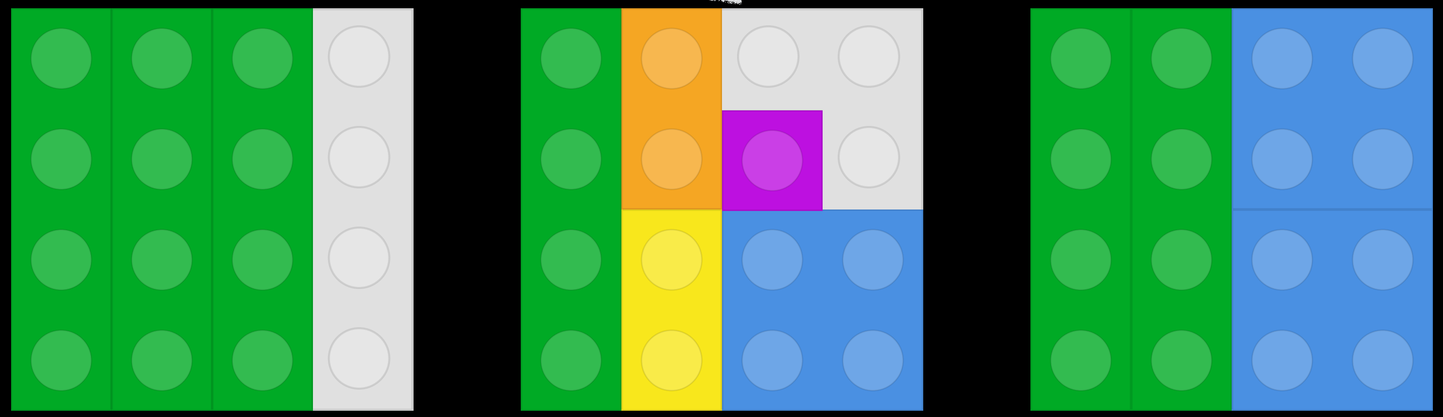


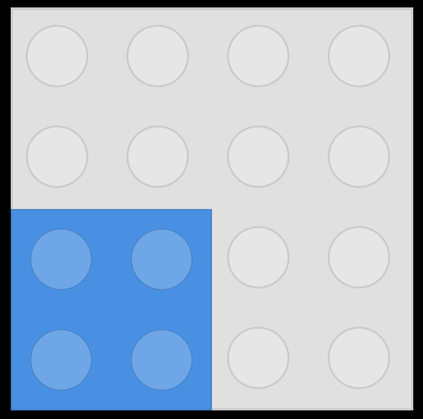
By bundling them together





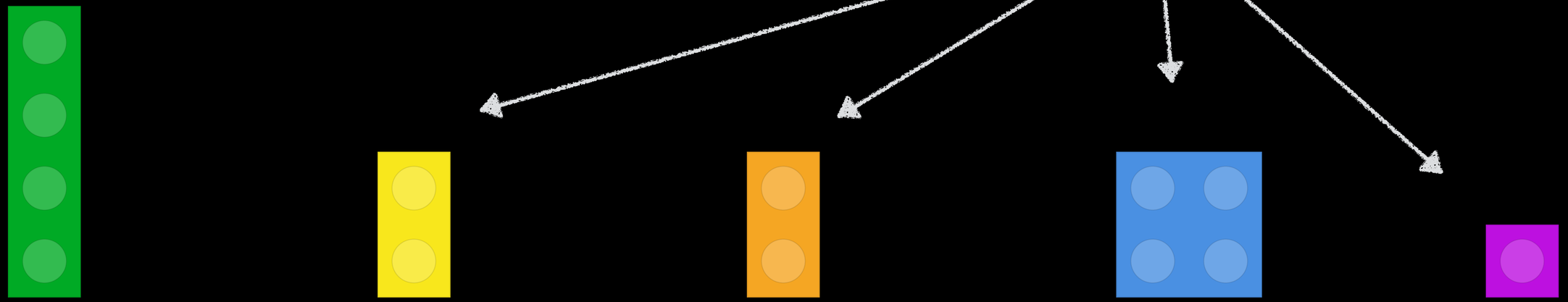
But what if
this one fails?



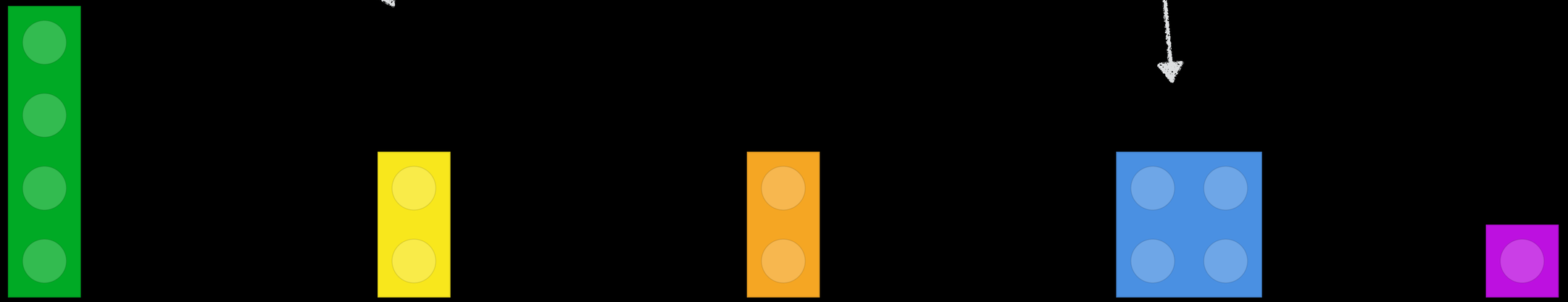


IMAGINE WE DON’T NEED 4X4 PIECE ANYMORE?

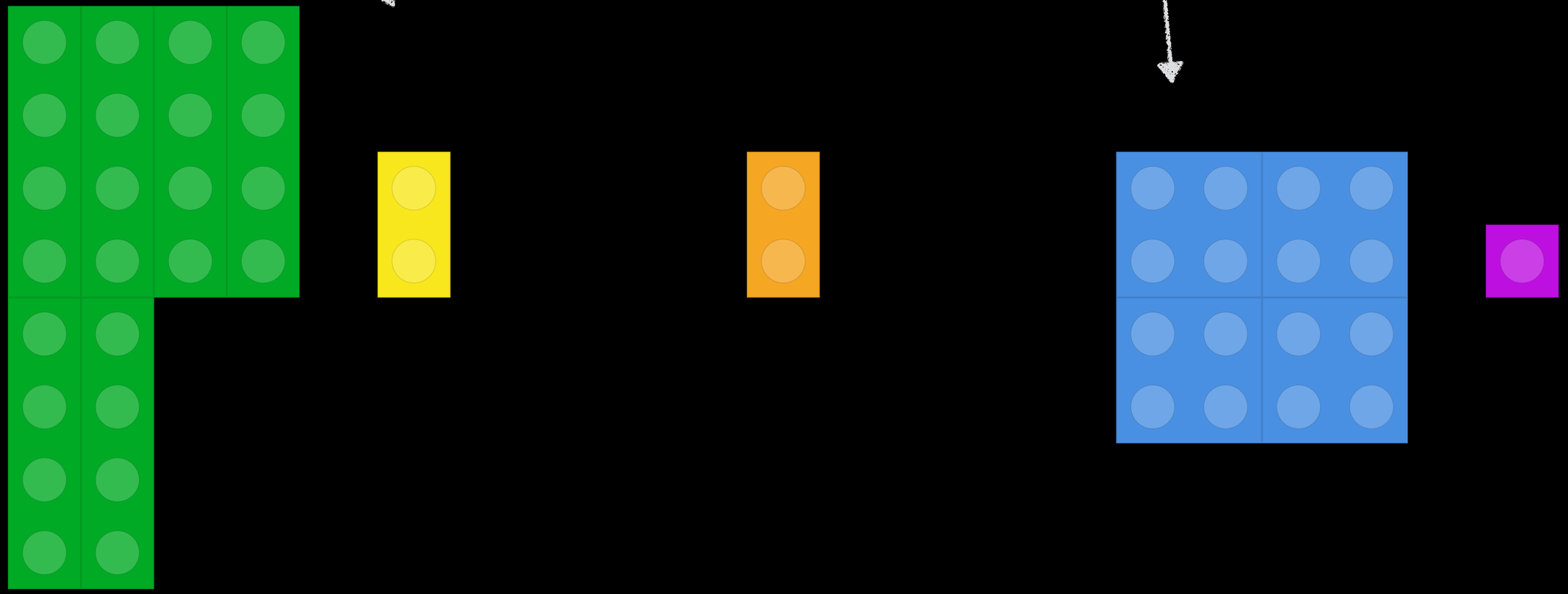
Imagine our microservices without 4x4 white pieces!



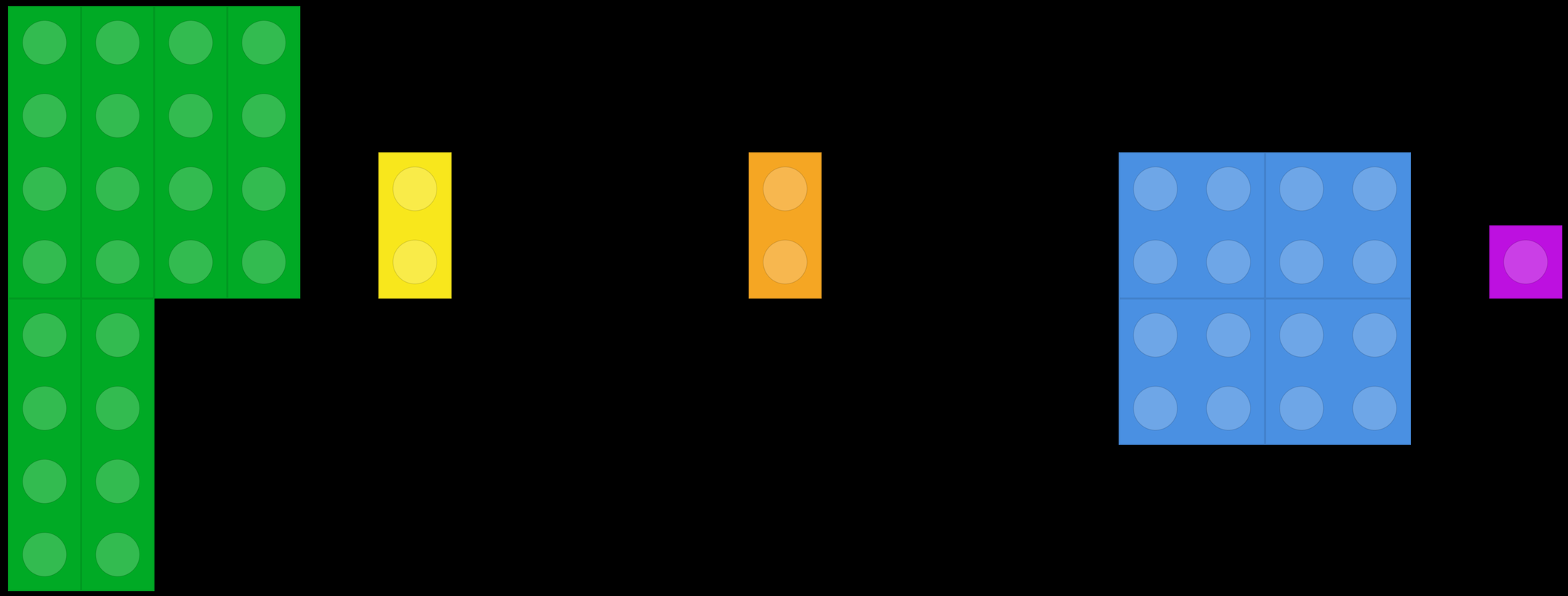
Than scaling is simply
adding more services



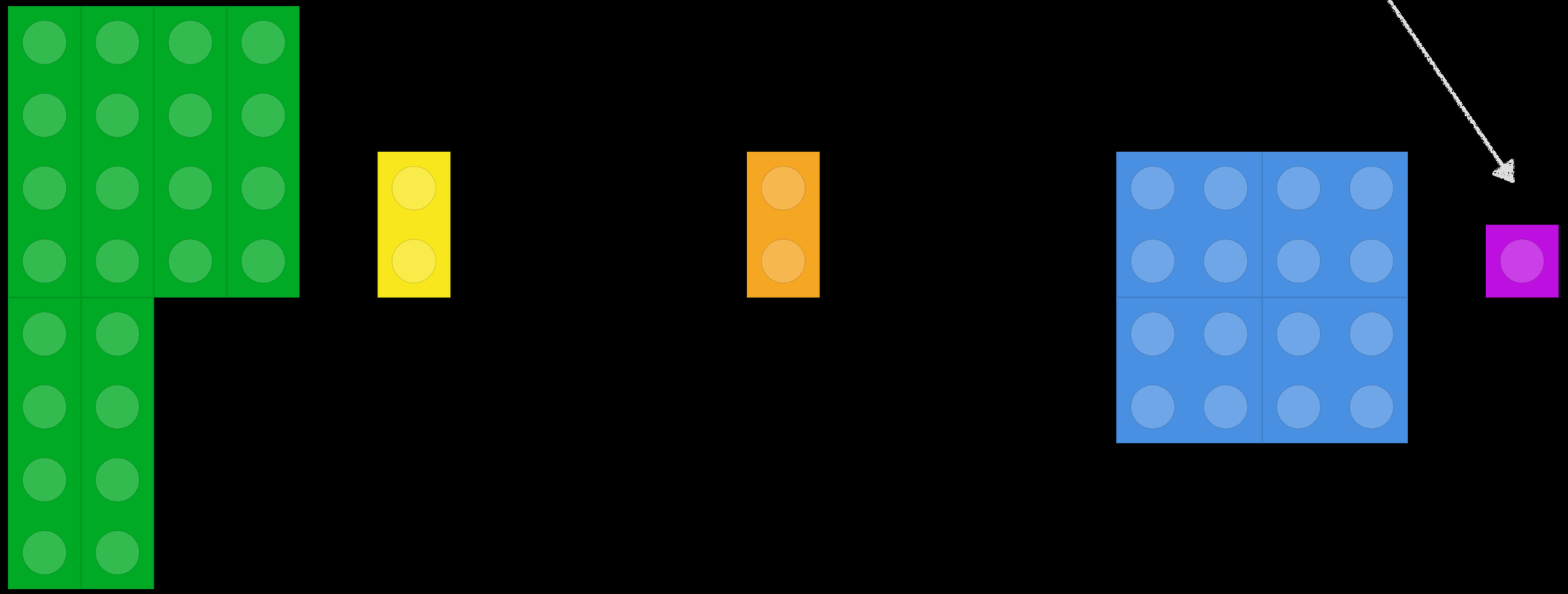
Than scaling is simply
adding more services



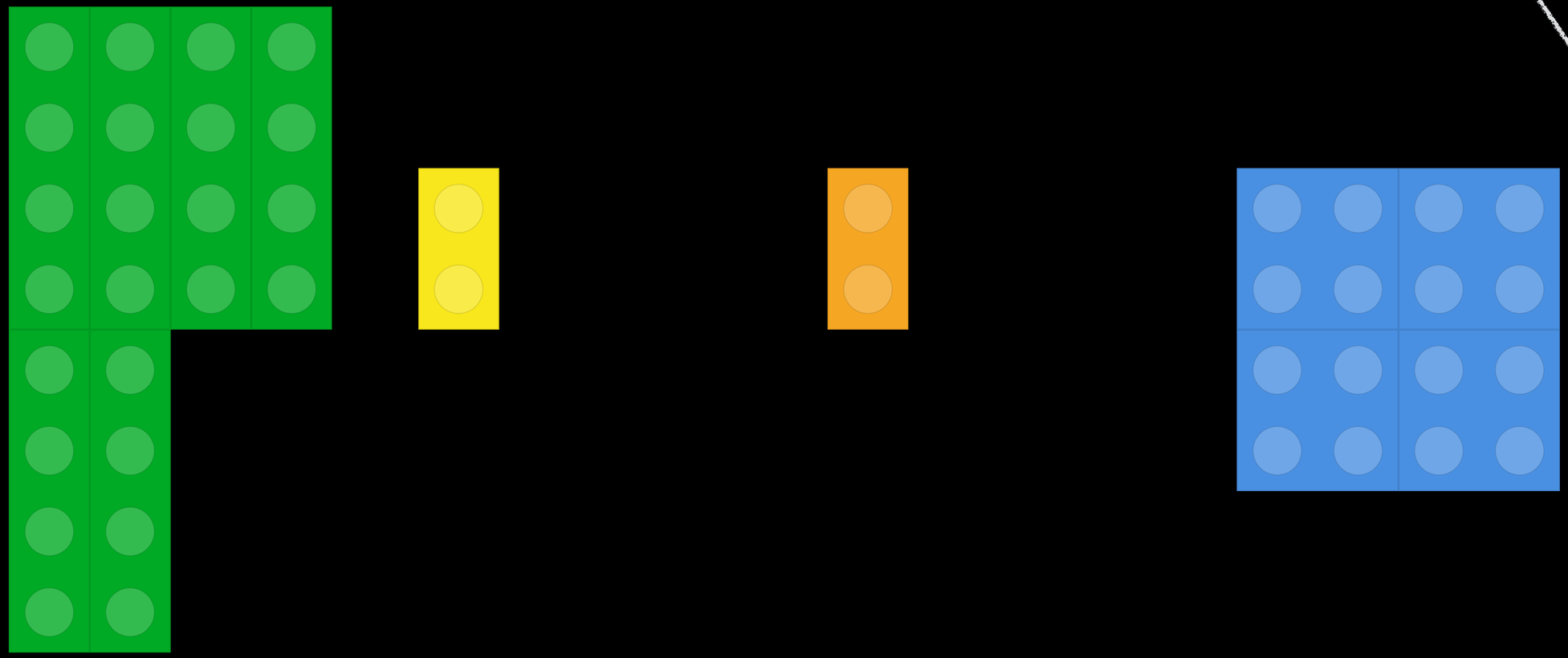
And, if we don't need some service...



And, if we don't need some service...



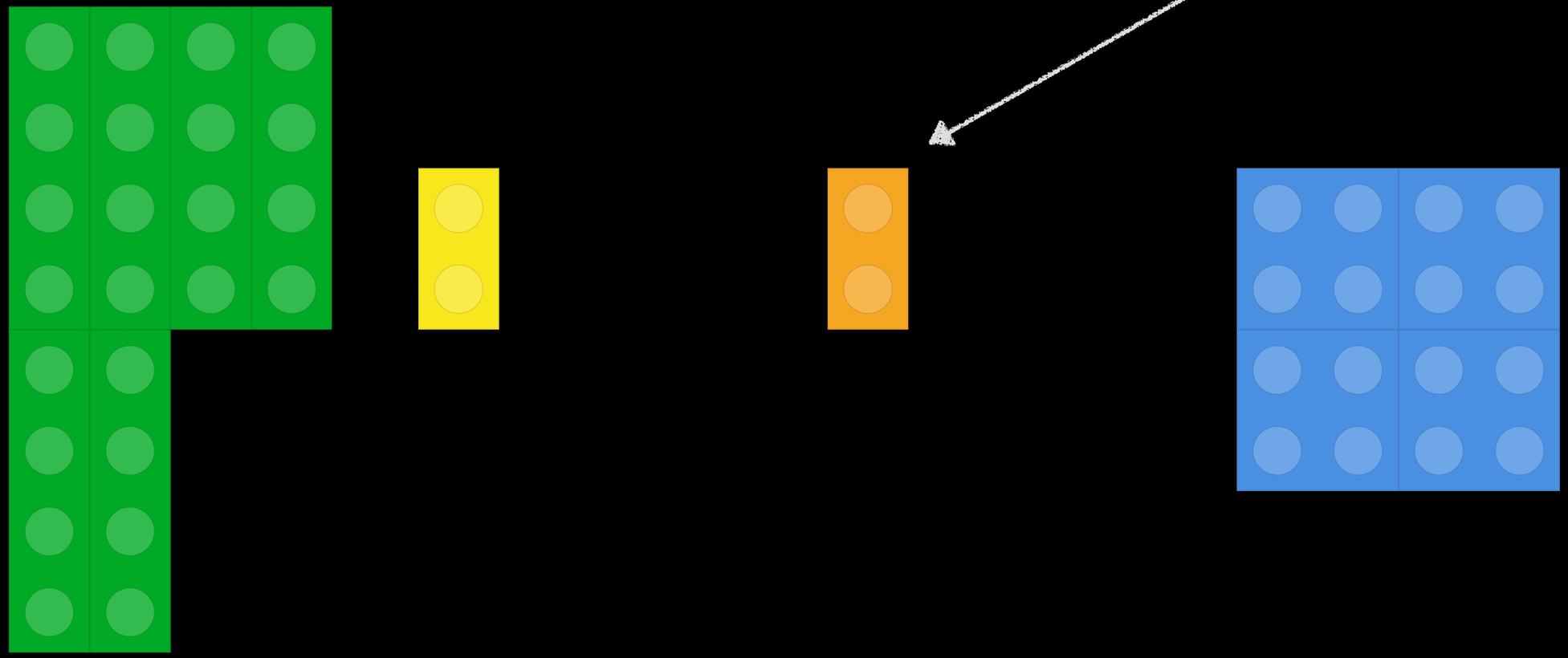
And, if we don't need some service...



NOW IMAGINE THAT

SCALING IS DONE AUTOMATICALLY

If something fails it's replaced automatically



ALSO IMAGINE THAT YOU PAY
ONLY FOR THE PARTS THAT YOU ARE
USING AT THE MOMENT

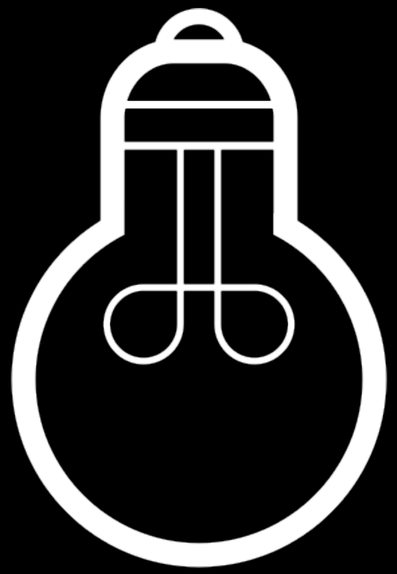

POLICE BOX

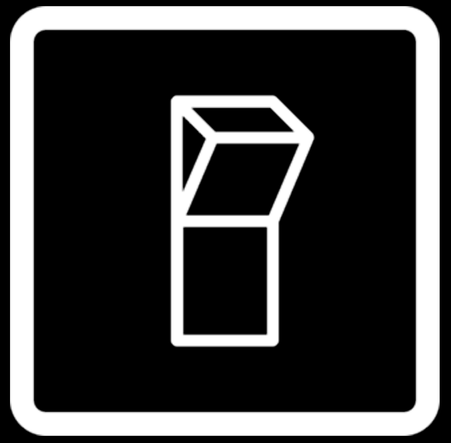
WELL, WE CALL THAT THING SERVERLESS!

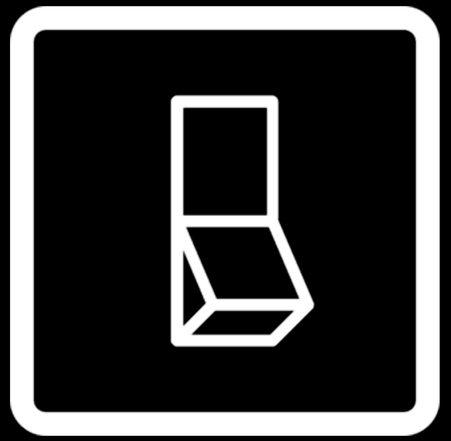

CORNELIAPORNELIA

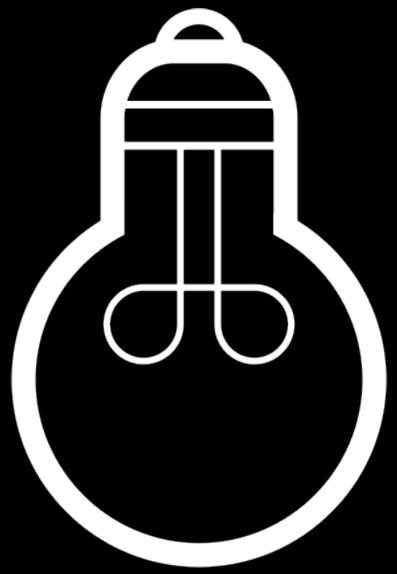
2. HOW DOES IT WORK?

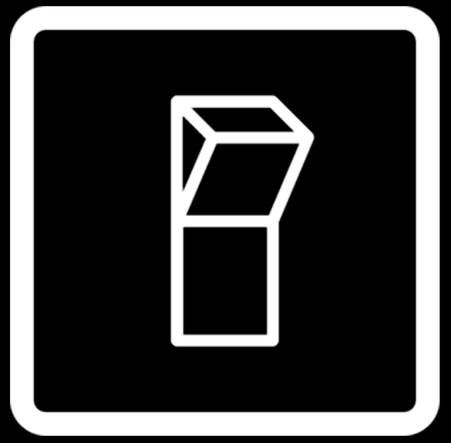
EVENT-DRIVEN

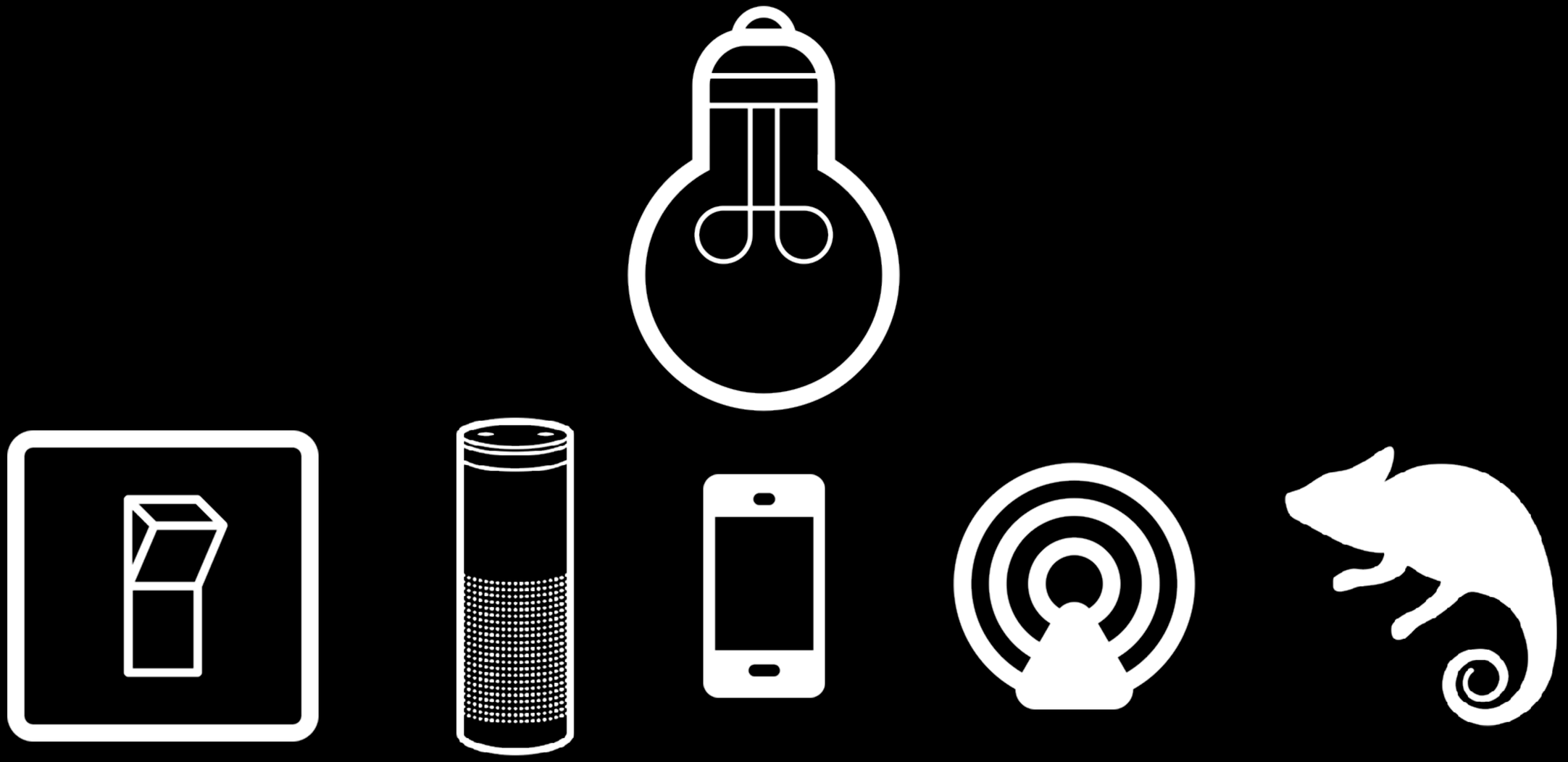


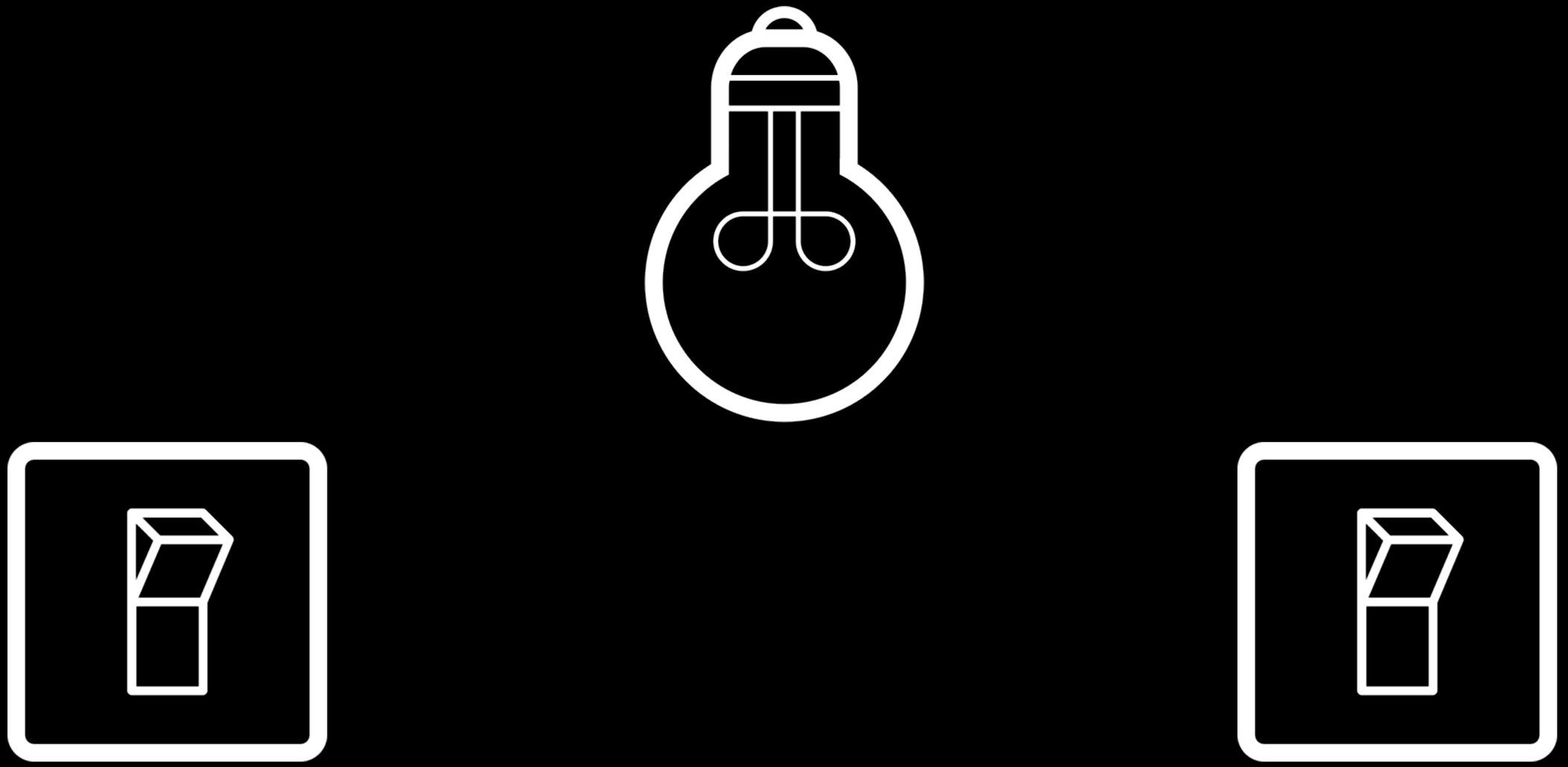


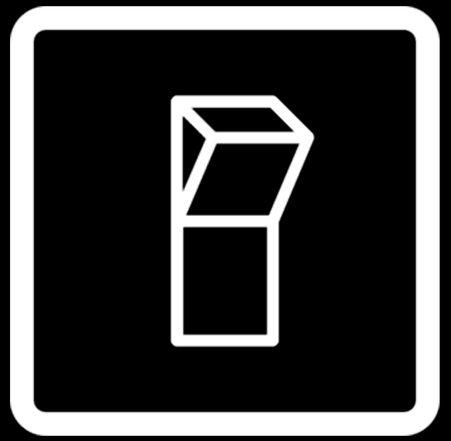


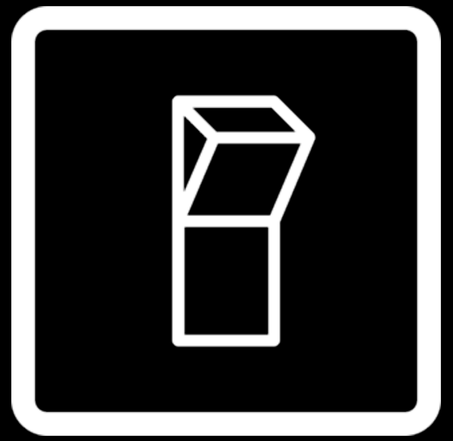




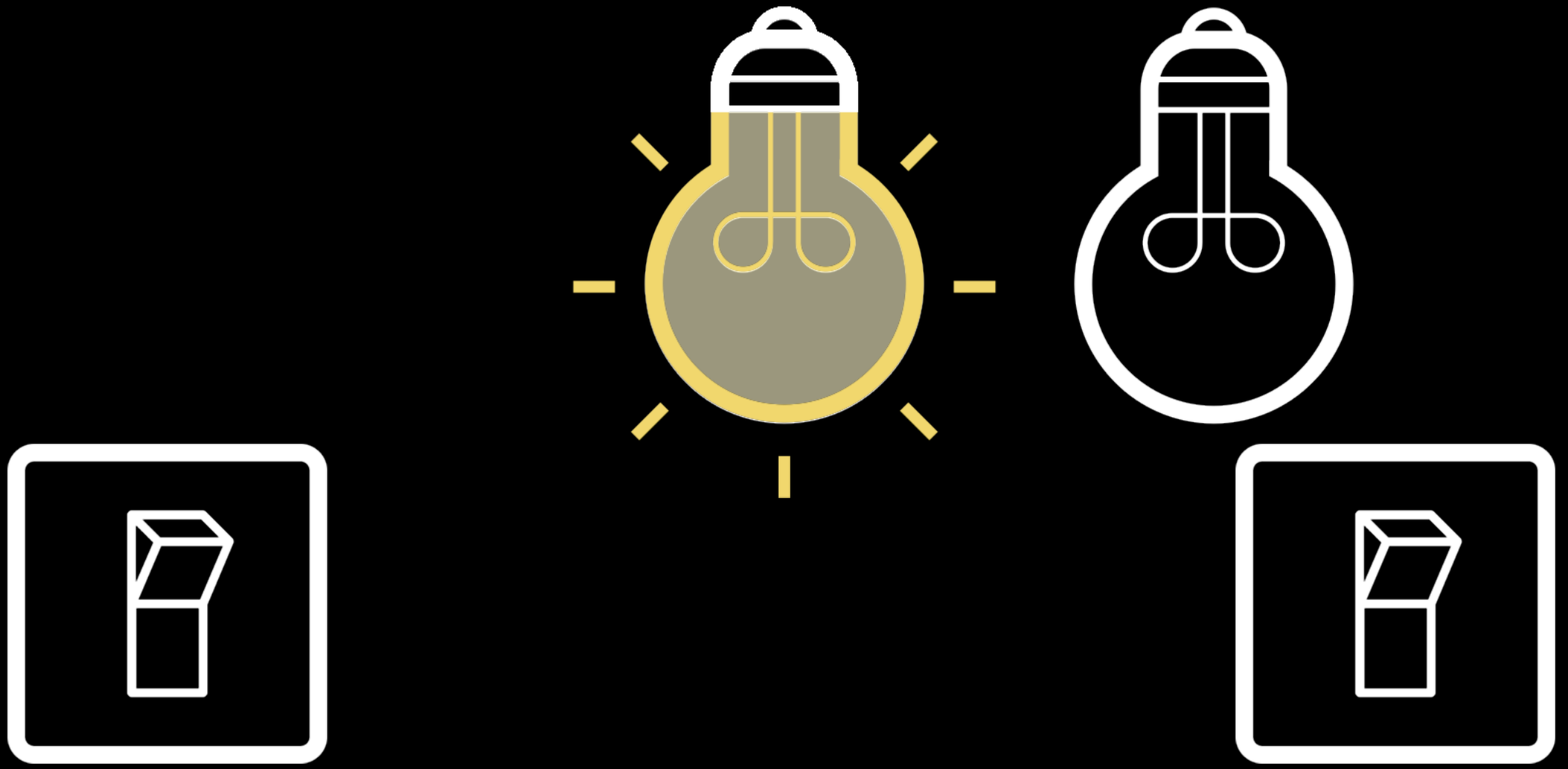


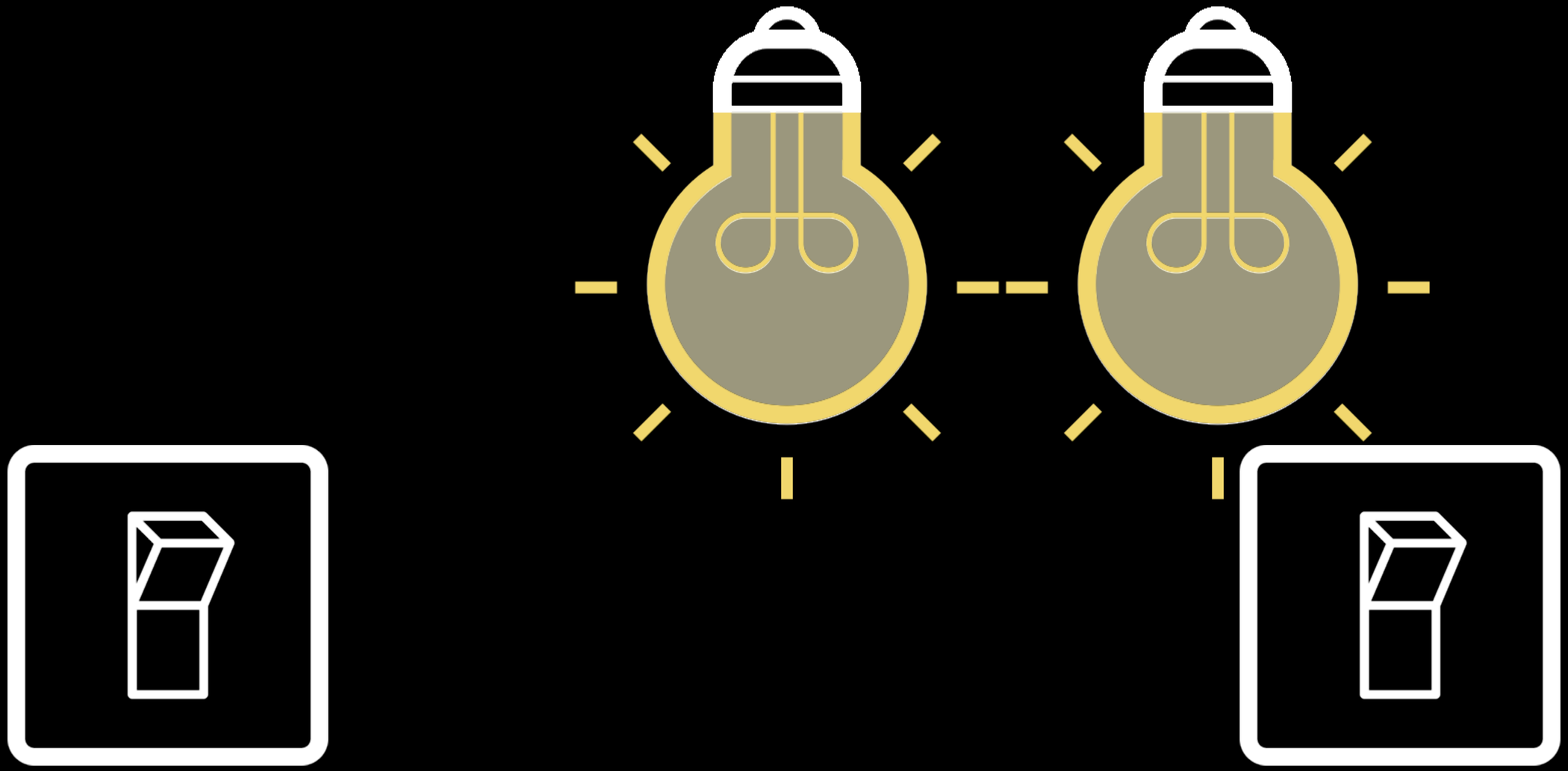


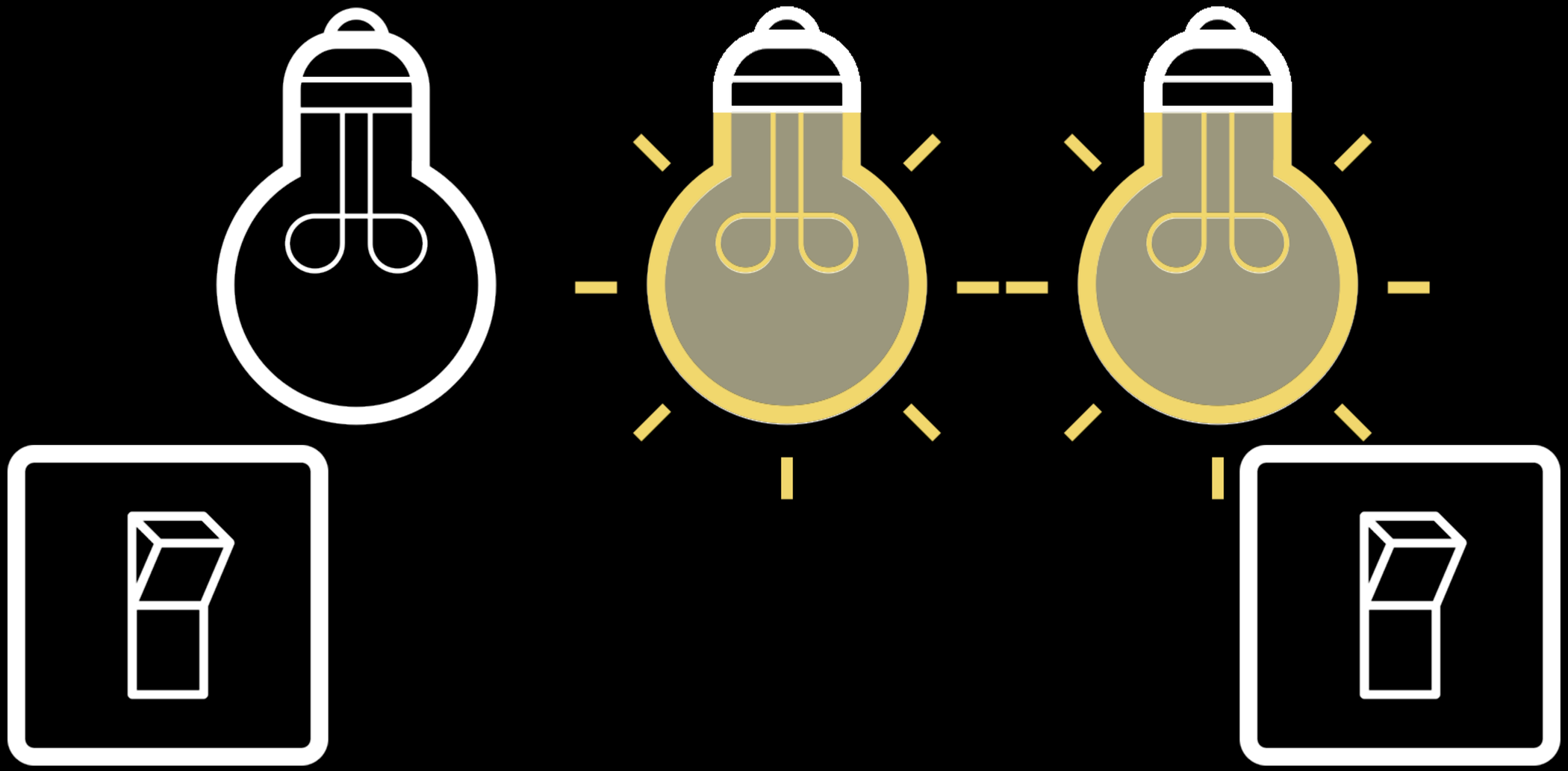


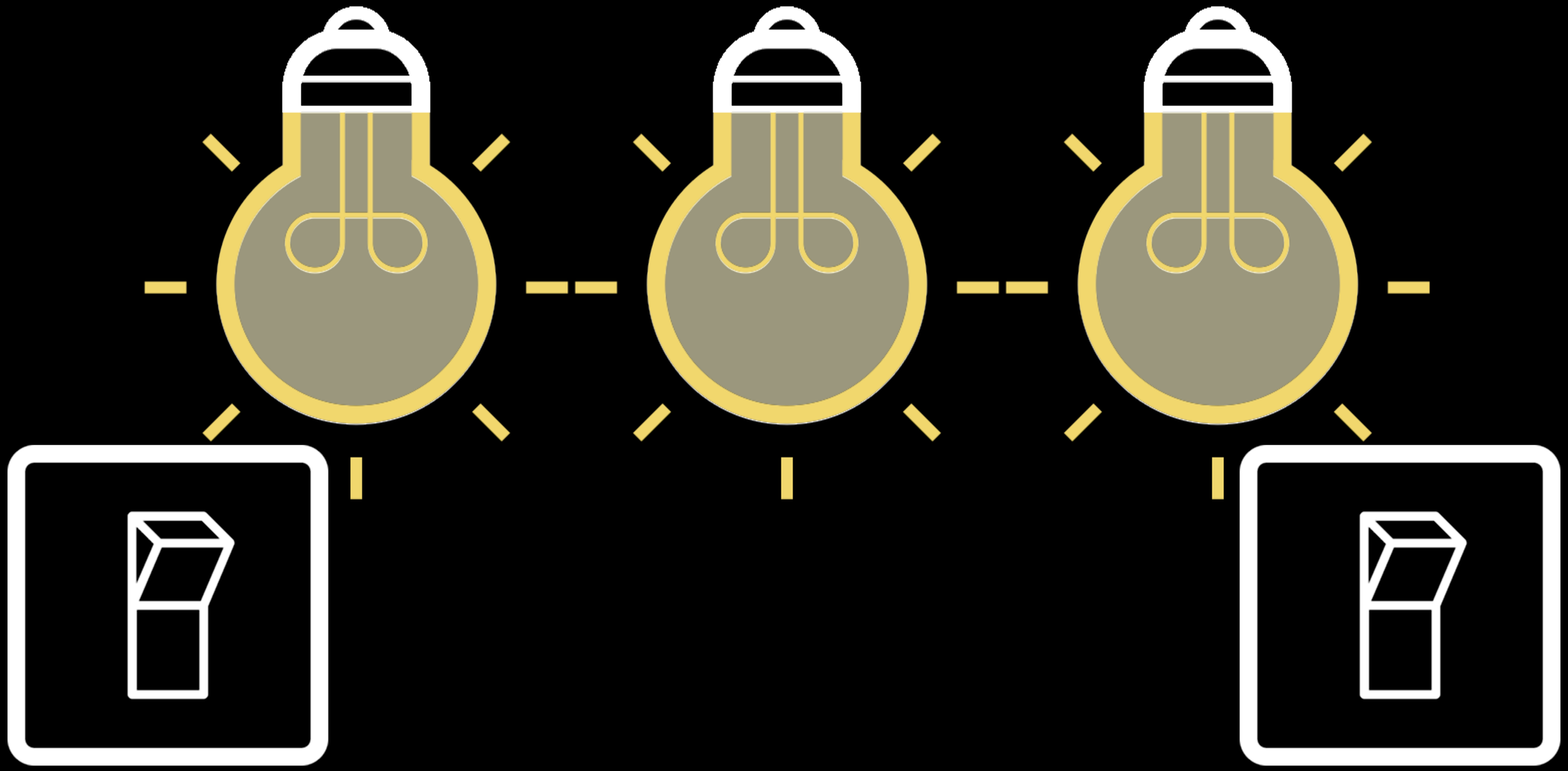


#holyjs
@slobodan_



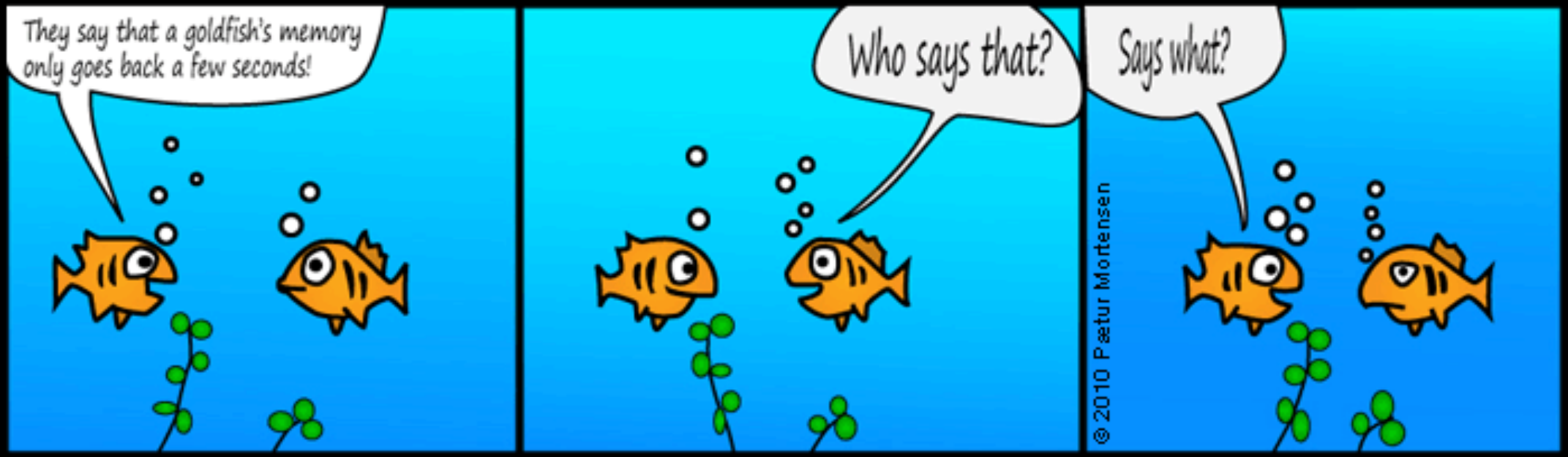






“LIGHTBULB” == FUNCTION

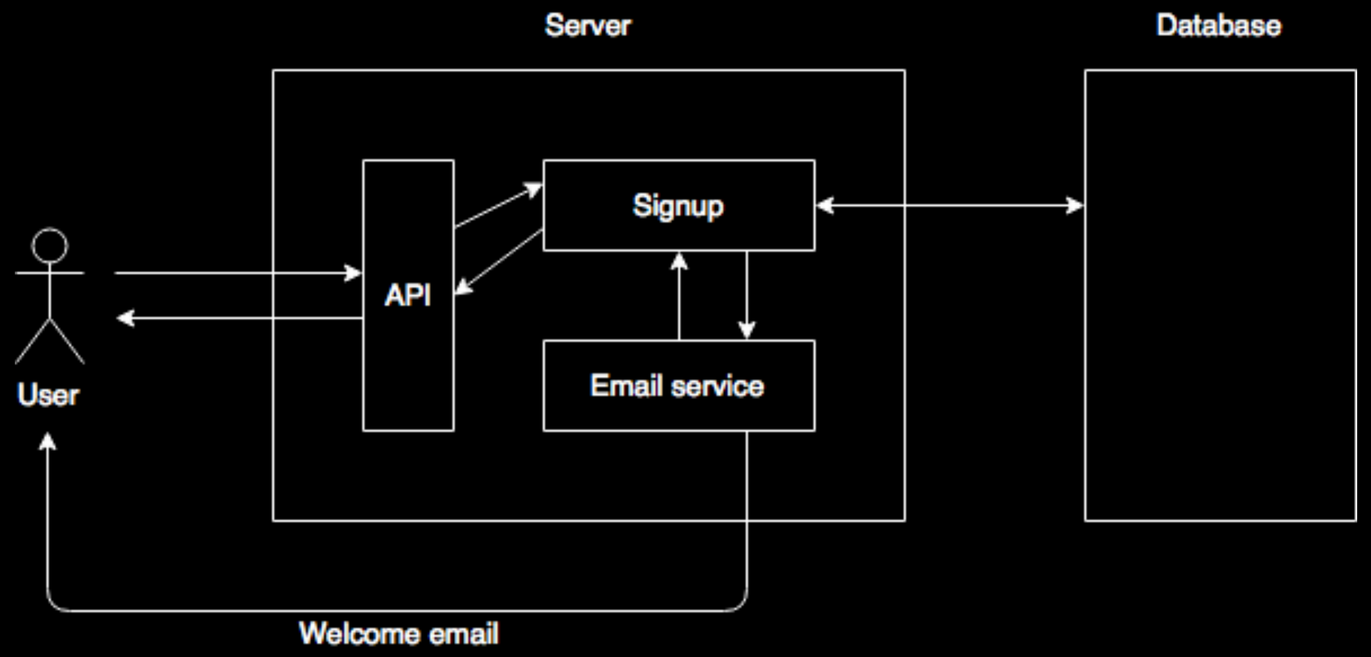
STATELESS

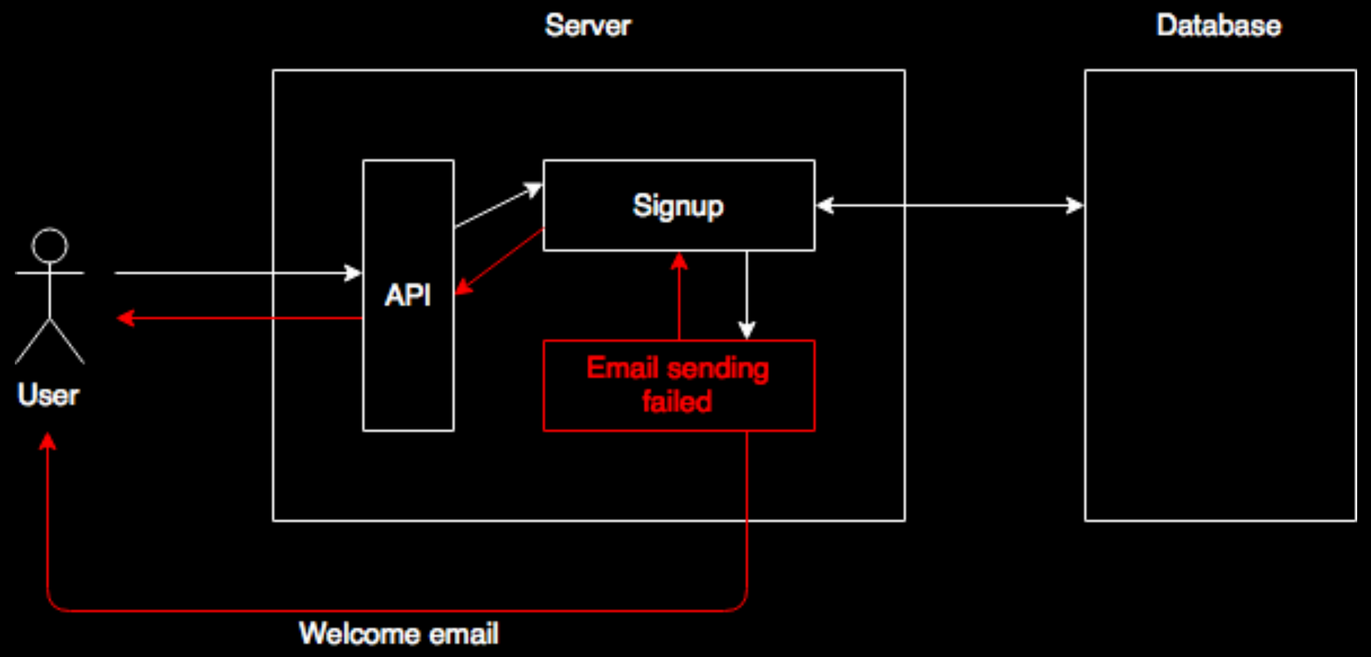


SERVERLESS IS LIKE A GOLDEN FISH

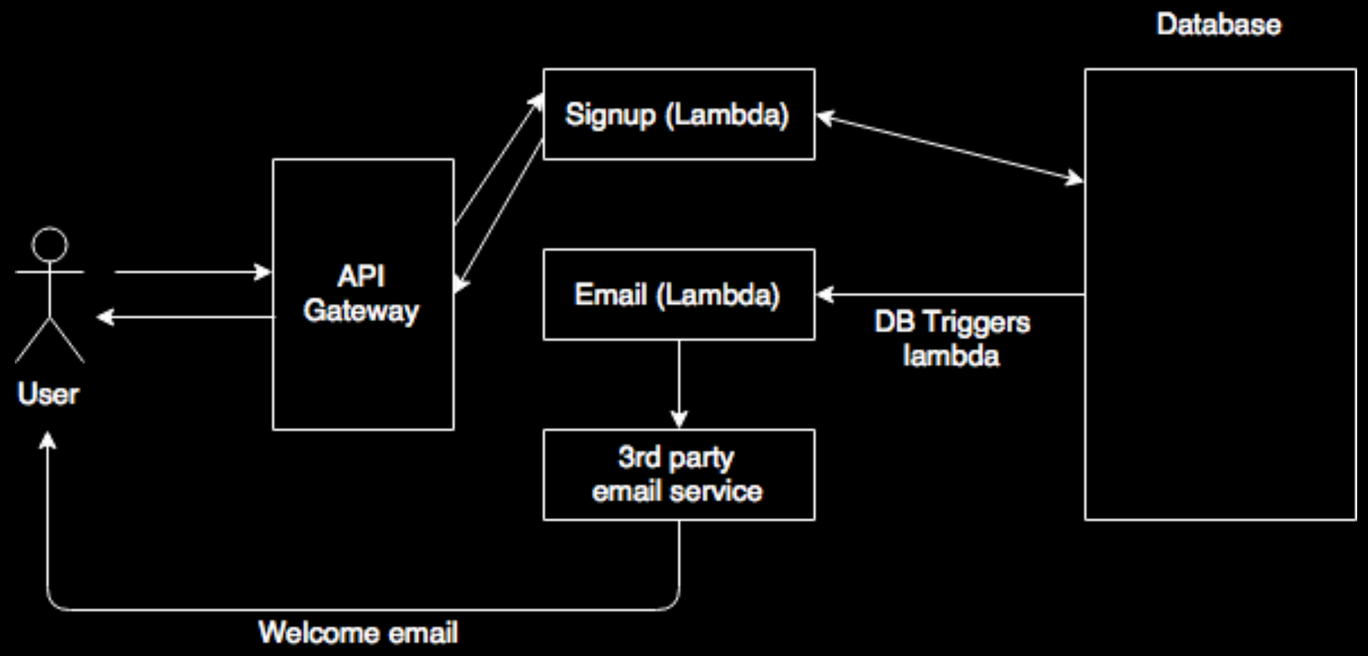
#holyjs
@slobodan_

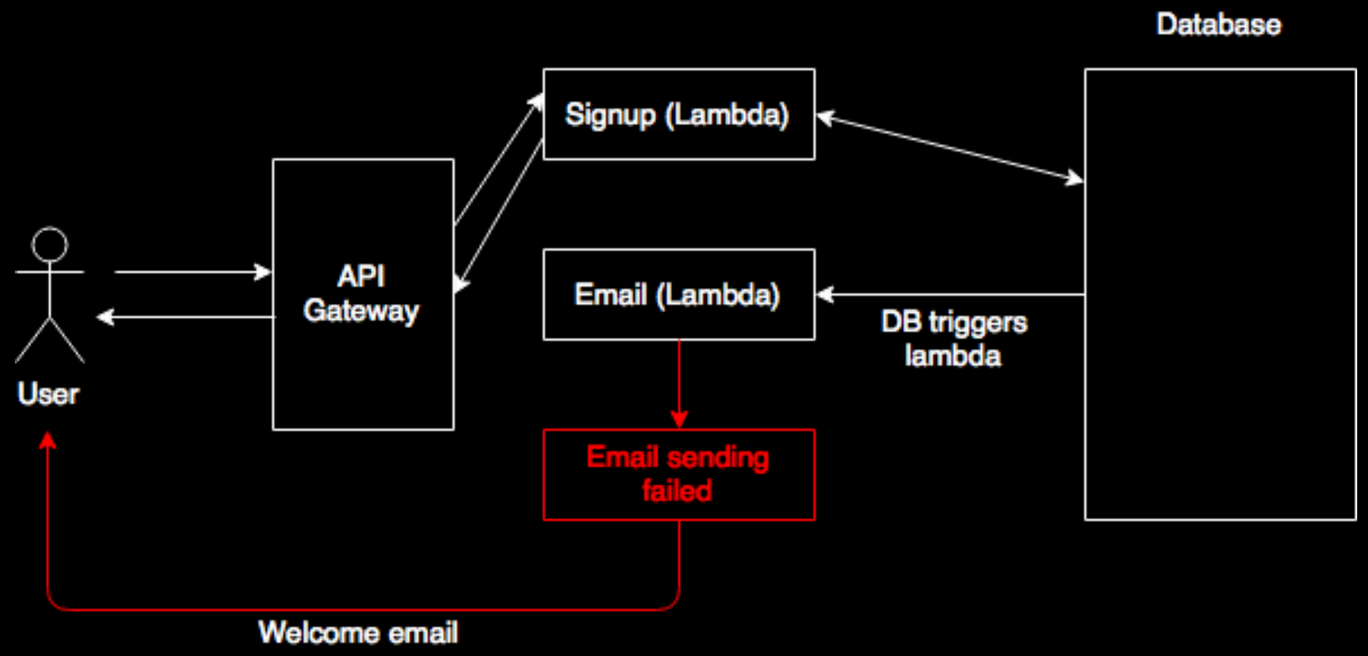
BUT WHAT ABOUT REAL SYSTEMS?


Server
Database
Signup
API
User
Email service
Welcome email


Server
Database
Signup
API
User
Email sending
failed
Welcome email

SERVERLESS APP


Database
Signup (Lambda)
API
Gateway
Email (Lambda)
DB Triggers
lambda
User
3rd party
email service
Welcome email


Database
Signup (Lambda)
API
Gateway
Email (Lambda)
DB triggers
lambda
User
Email sending
failed
Welcome email

WHAT CAN TRIGGER SERVERLESS FUNCTION

- HTTP REQUEST (API)
- FILE
- NOTIFICATION
- MAIL
- STREAM
- MANY OTHER THINGS


gif-finder.com

3. WHAT ARE THE MOST IMPORTANT SERVERLESS PLATFORMS?

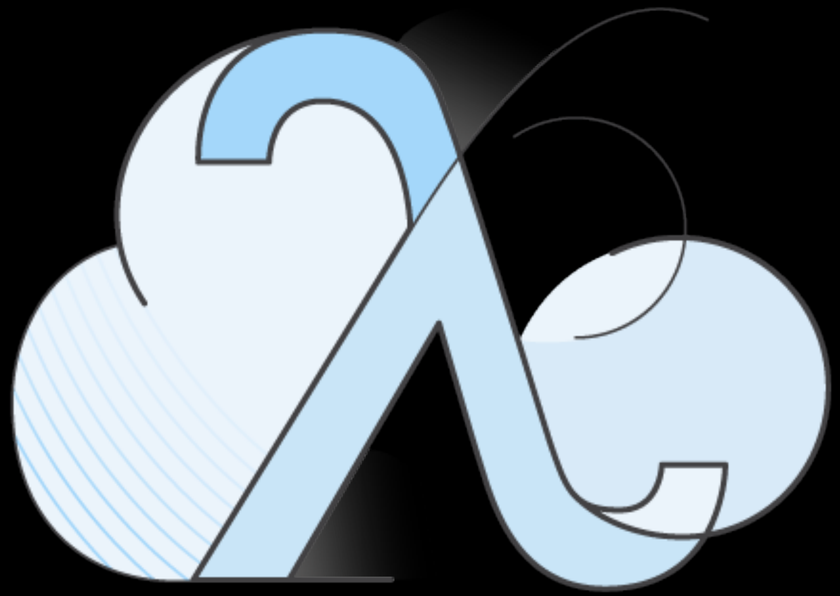
Microsoft Azure

Google Cloud Platform

Microsoft Azure

Google Cloud Platform


amazon
web services



AWS LAMBDA

- $0.20 PER 1 MILLION REQUESTS
- FIRST MILLION REQUESTS / MONTH ARE FREE
- $0.00001667 FOR EVERT GB-SECOND
- FIRST 400.000 GB-SECONDS / MONTH ARE FREE

- TIMEOUT UP TO 300 SECONDS
- 128MB TO 1.5GB OF MEMORY
- 500MB OF NON-PERSISTENT STORAGE (/TMP)
- DEFAULT CONCURRENT EXECUTION LIMIT: 1000

WHAT PROGRAMMING LANGUAGE YOU NEED TO USE FOR SERVERLESS?

AWS LAMBDA

- NODE.JS
- PYTHON
- JAVA
- C#
- SHELL SCRIPTS
- ANY EXECUTABLE

AZURE FUNCTIONS

- NODE.JS
- C#
- F#
- PYHON
- PHP
- SHELL SCRIPTS
- ANY EXECUTABLE

GOOGLE CODE FUNCTIONS

JUST NODE.JS AT THE MOMENT

#holyjs
@slobodan_

4. WHY IS IT IMPORTANT?

MINIMAL MAINTENANCE

FINANCIAL INCENTIVE

https://gojko.net/2016/08/27/serverless.html

TRUE MICROSERVICES

AUTO SCALING AND FAILOVER

FOCUS ON BUSINESS LOGIC, NOT CONFIGURATION


COMEDY
CENTRAL

5. HOW DOES IT WORK WITH NODEJS?


4GIFS
.com

PROS

- GOOD STARTUP PERFORMANCE
- SMALL MODULES
- PERFORMANCE WITH LOW CPU AND MEMORY
- SINGLE THREAD
- EASY TO LEARN

CONS

- ASYNC
- MADE FOR SCALABILITY
- NODE_MODULES SIZE
- EASY TO FUCKUP

```
'use strict'
console.log('Loading event')

exports.handler = (event, context, callback) => {
  console.log('"Hello": "World"')
  callback(null, {"Hello": "World"})
}
```

hello-world.js

HOW TO DEPLOY IT TO AWS

- CREATE LAMBDA FUNCTION
- UPLOAD ZIP FILE
- CREATE API GATEWAY
- SET TRIGGER
- SET PERMISSIONS
- ~11 MINS READ: http://amzn.to/2kcilEt

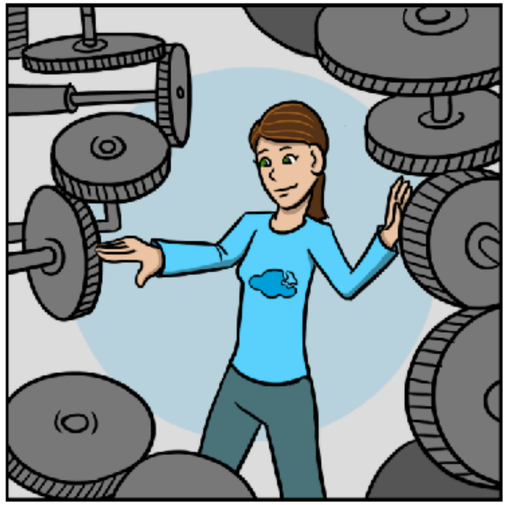
FRAMEWORKS

- SERVERLESS FRAMEWORK
- CLAUDIA.JS
- APEX

CLAUDIAJS.COM

> 69.000 DOWNLOADS

EVENT-DRIVEN
MICRO-SERVICES



AUTO-SCALING
WEB APIS

MULTI-PLATFORM
CHAT-BOTS

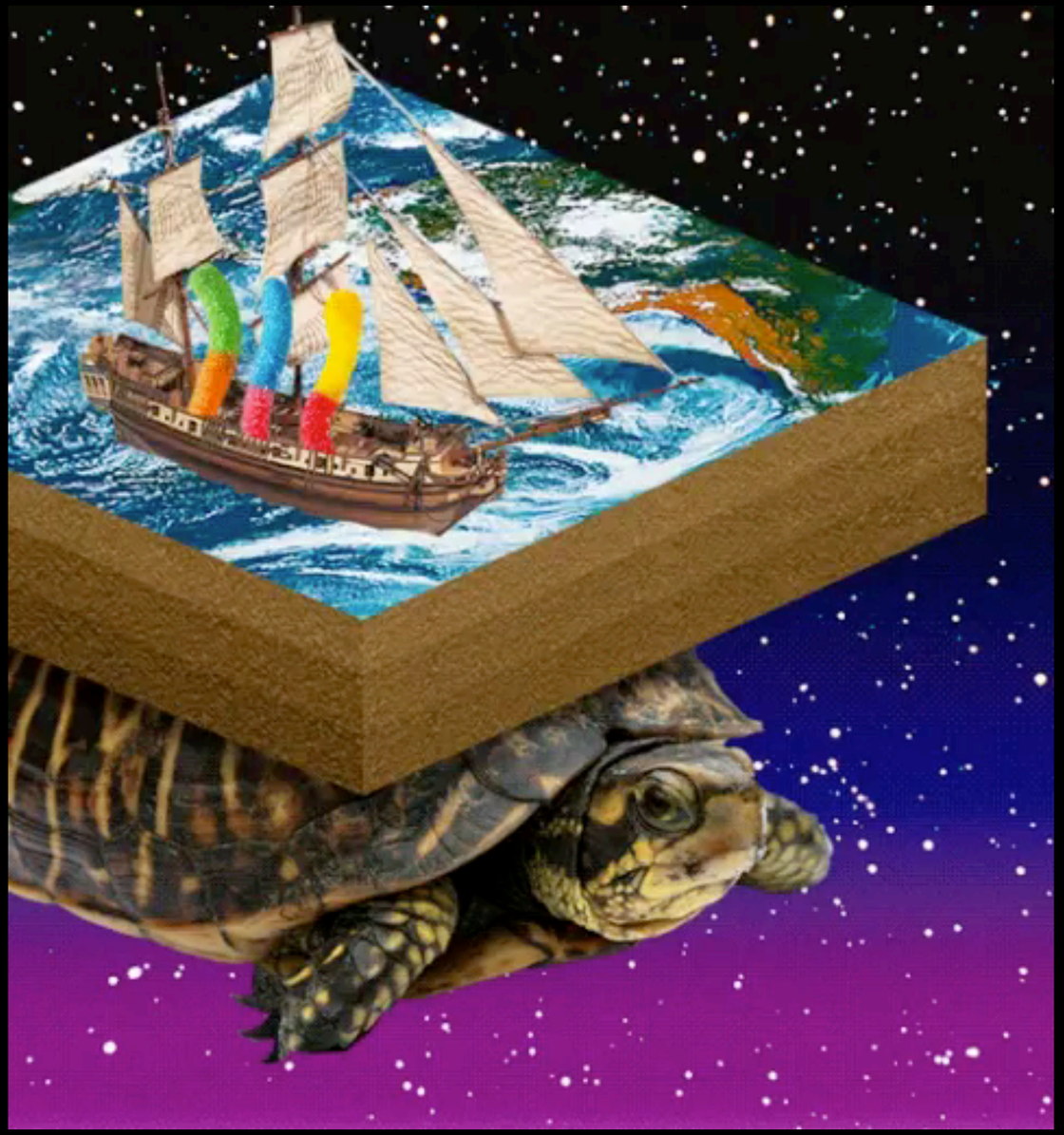
#holyjs
@slobodan_

MAIN CHARACTERISTICS

- DEPLOY AND UPDATE WITH SINGLE COMMAND
- USE STANDARD NPM MODULES
- NO BOILERPLATE, JUST FOCUS ON YOUR WORK
- MANAGE MULTIPLE VERSIONS EASILY
- GET STARTED QUICKLY, FLAT LEARNING CURVE

FLAT LEARNING CURVE

HOW FAST CAN WE START?



prepare

- `npm i claudia -g`
- SET UP AWS CREDENTIALS (ONE TIME)
- `mkdir claudia-api && cd $_`
- `npm init -f`
- `npm i claudia-api-builder -S`

```
'use strict'

const Api = require('claudia-api-builder')
const api = new Api()
const jokes = require('./jokes.json')

api.get('/', () => jokes)

module.exports = api
```

api.js

```
claudia create \
  --region eu-central-1 \
  --api-module api
```

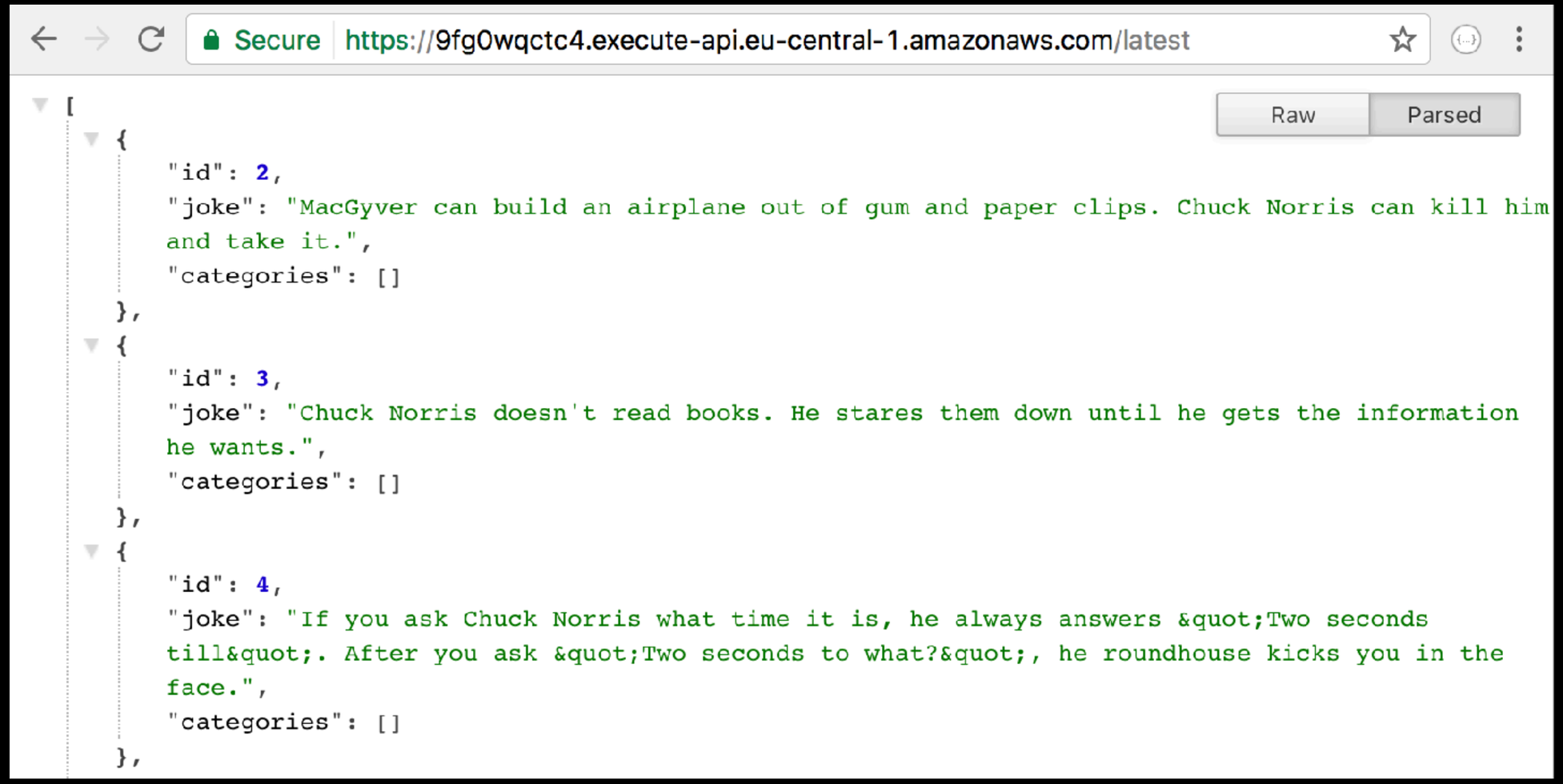
DEPLOY

```
{
  "lambda": {
    "role": "chuck-norris-api-executor",
    "name": "chuck-norris-api",
    "region": "eu-central-1"
  },
  "api": {
    "id": "d1weoszyg7",
    "module": "api",
    "url": "https://9fg0wqctc4.execute-api.eu-
           central-1.amazonaws.com/latest"
  }
}
```

RESULT

bit.ly/holyjs-chuck-api

bit.ly/holyjs-chuck-api



```
'use strict'

const Api = require('claudia-api-builder')
const api = new Api()
const jokes = require('./jokes.json')

api.get('/', () => jokes)

api.get('/{id}', req => jokes.filter(joke =>
  joke.id === req.pathParams.id
)))

api.get('/random', () => jokes[
  Math.floor(Math.random() * jokes.length)
])

module.exports = api
```

api.js

#holyjs
@slobodan_

```
'use strict'

const Api = require('claudia-api-builder')
const api = new Api()
const jokes = require('./jokes.json')

api.get('/', () => jokes)

api.get('/{id}', req => jokes.filter(joke =>
  joke.id === req.pathParams.id
)))

api.get('/random', () => jokes[
  Math.floor(Math.random() * jokes.length)
])

module.exports = api
```

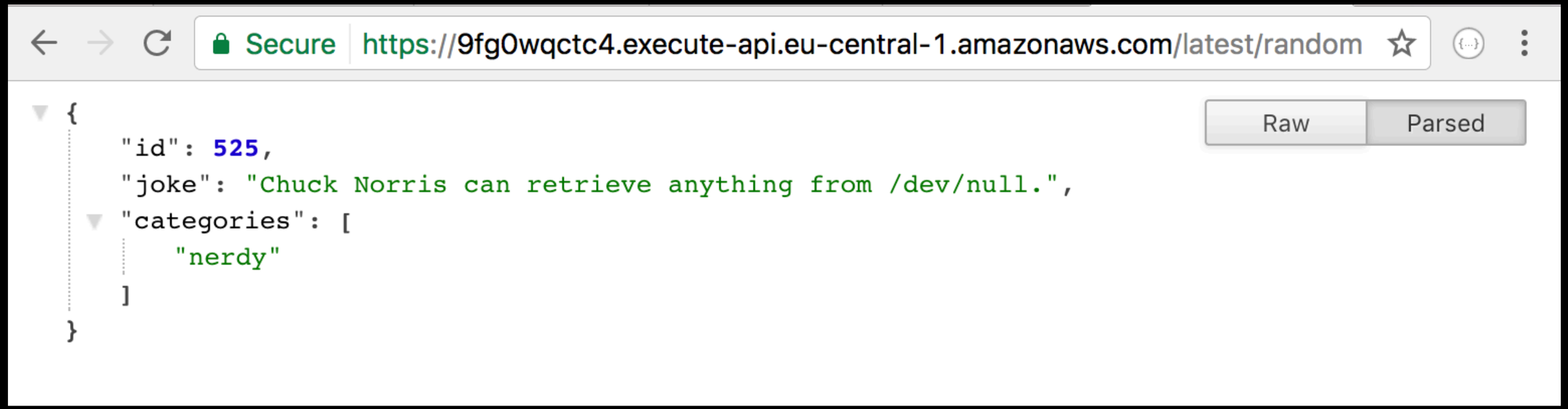
api.js

#holyjs
@slobodan_

```
claudia update
```

UPDATE

/random



bit.ly/holyjs-chuck-api

/42

bit.ly/holyjs-chuck-api

#holyjs
@slobodan_

```
// …

api.get('/{id}', req => {
  console.log(req.pathParams)

  return jokes.filter(joke =>
    joke.id === req.pathParams.id
  )
})

// …
```

api.js

6. HOW TO DEBUG SERVERLESS FUNCTION

WHERE CAN YOU SEE CONSOLE.LOG?

CLOUD WATCH

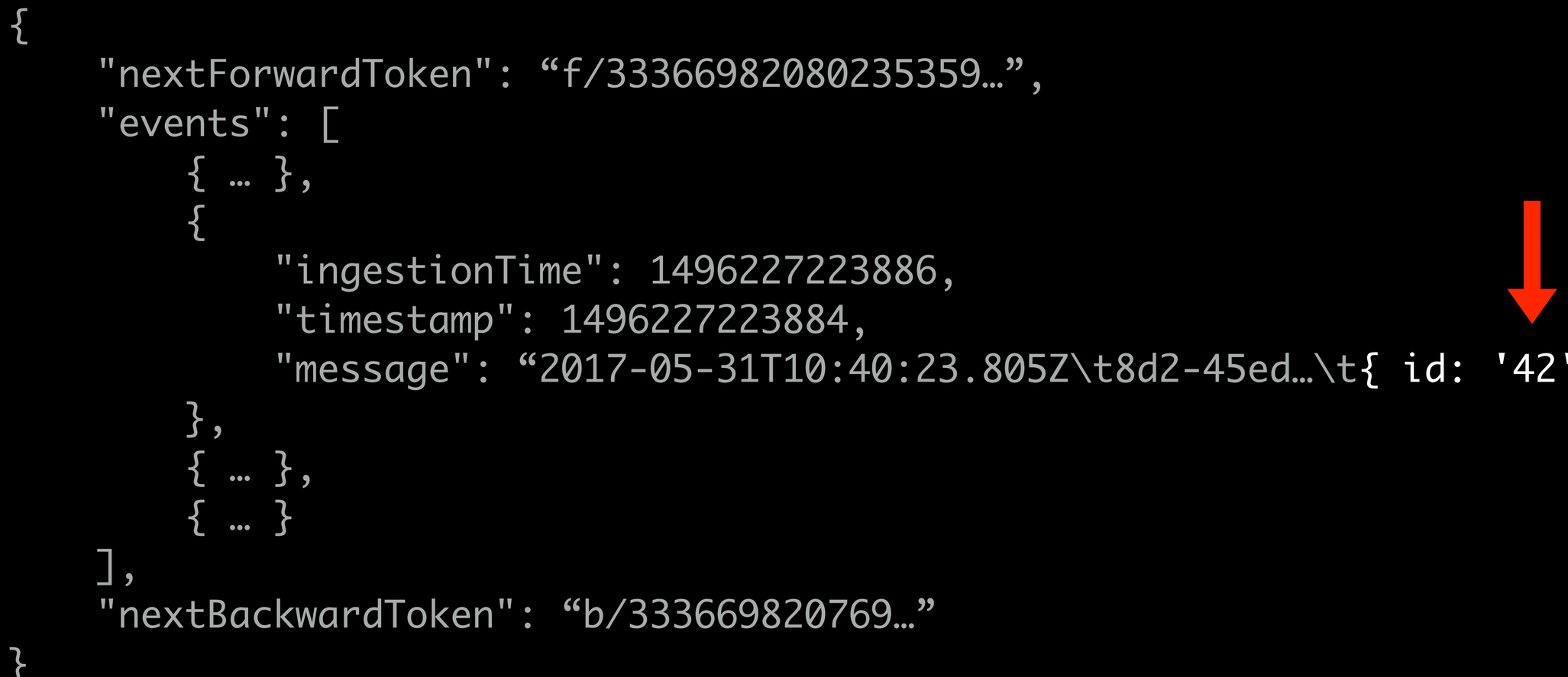
MONITORING SERVICE FOR AWS CLOUD RESOURCES AND THE APPS YOU RUN ON AWS

YOU CAN USE IT VIA
WEB CONSOLE, AWS CLI, ETC.

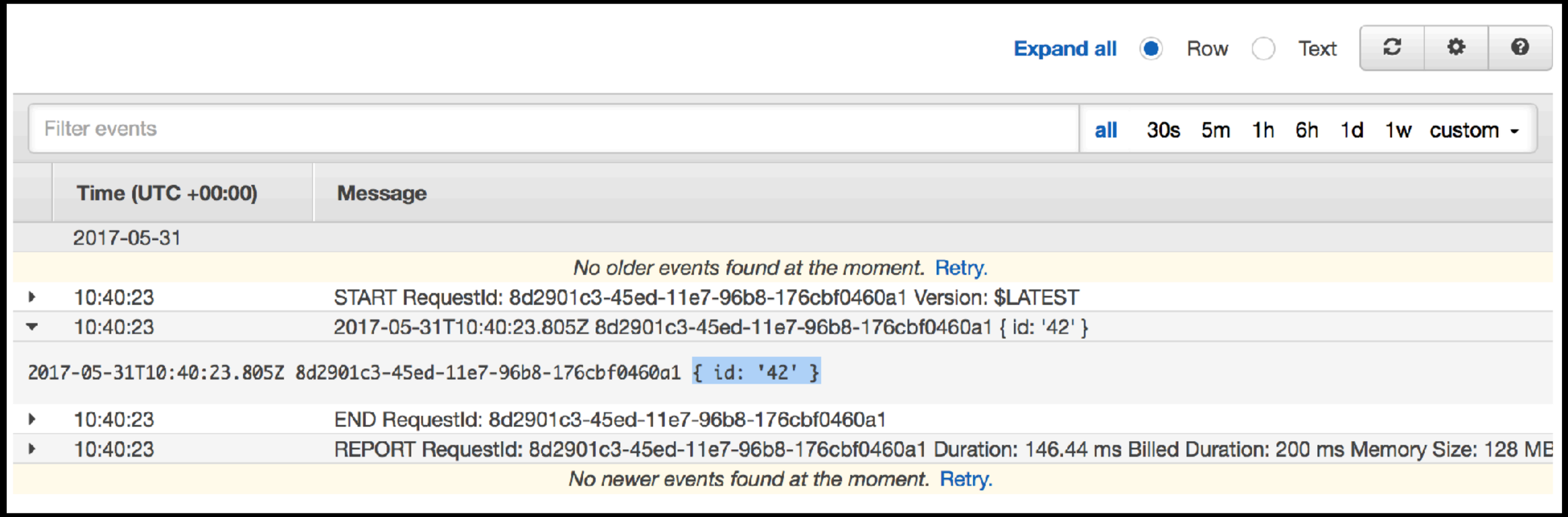
```
aws logs get-log-events \
  --log-group-name /aws/lambda/chuck-norris-api \
  --log-stream-name 2017/05/31/…\
  --output json
```

```
{
    "nextForwardToken": “f/33366982080235359…”,
    "events": [
        { … },
        {
            "ingestionTime": 1496227223886,
            "timestamp": 1496227223884,
            "message": “2017-05-31T10:40:23.805Z\t8d2-45ed…\t{ id: '42' }\n"
        },
        { … },
        { … }
    ],
    "nextBackwardToken": “b/333669820769…”
}
```

OUTPUT


{
"nextForwardToken": “f/33669820802353 59…”,
"events": [
{ … },
{
"ingestionTime": 1496227223886,
"timestamp": 1496227223884,
"message": “2017-05-31T10:40:23.805Z\t8d2-45ed…\t{ id: '42' }\n"
},
{ … },
{ … }
],
"nextBackwardToken": “b/333669820769…”
}

OUTPUT


Expand all
Row
Text
Filter events
all 30s 5m 1h 6h 1d 1w custom
Time (UTC +00:00)
Message
2017-05-31
No older events found at the moment. Retry.
10:40:23 START RequestId: 8d2901c3-45ed-11e7-96b8-176cbf0460a1 Version: $LATEST
10:40:23 2017-05-31T10:40:23.805Z 8d2901c3-45ed-11e7-96b8-176cbf0460a1 { id: '42' }
2017-05-31T10:40:23.805Z 8d2901c3-45ed-11e7-96b8-176cbf0460a1 { id: '42' }
10:40:23 END RequestId: 8d2901c3-45ed-11e7-96b8-176cbf0460a1
10:40:23 REPORT RequestId: 8d2901c3-45ed-11e7-96b8-176cbf0460a1 Duration: 146.44 ms Billed Duration: 200 ms Memory Size: 128 ME
No newer events found at the moment. Retry.

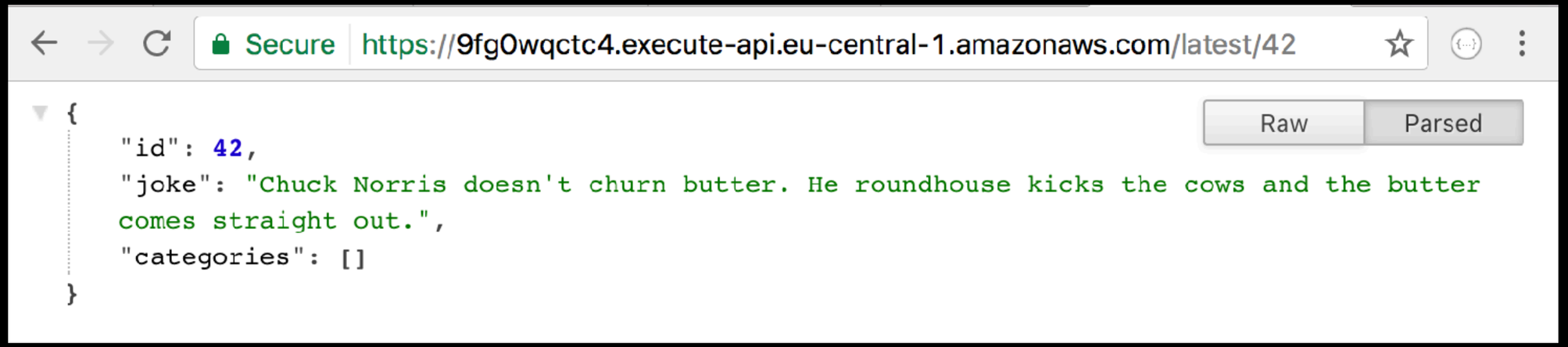
```
// …

api.get('/{id}', req => jokes.filter(joke =>
  joke.id == req.pathParams.id
)[0])

// …
```

api.js

/42



bit.ly/holyjs-chuck-api

#holyjs
@slobodan_

DEBUGGING IS NOT FUN.
HOW CAN WE DO LESS DEBUGGING?

BY WRITING TESTS

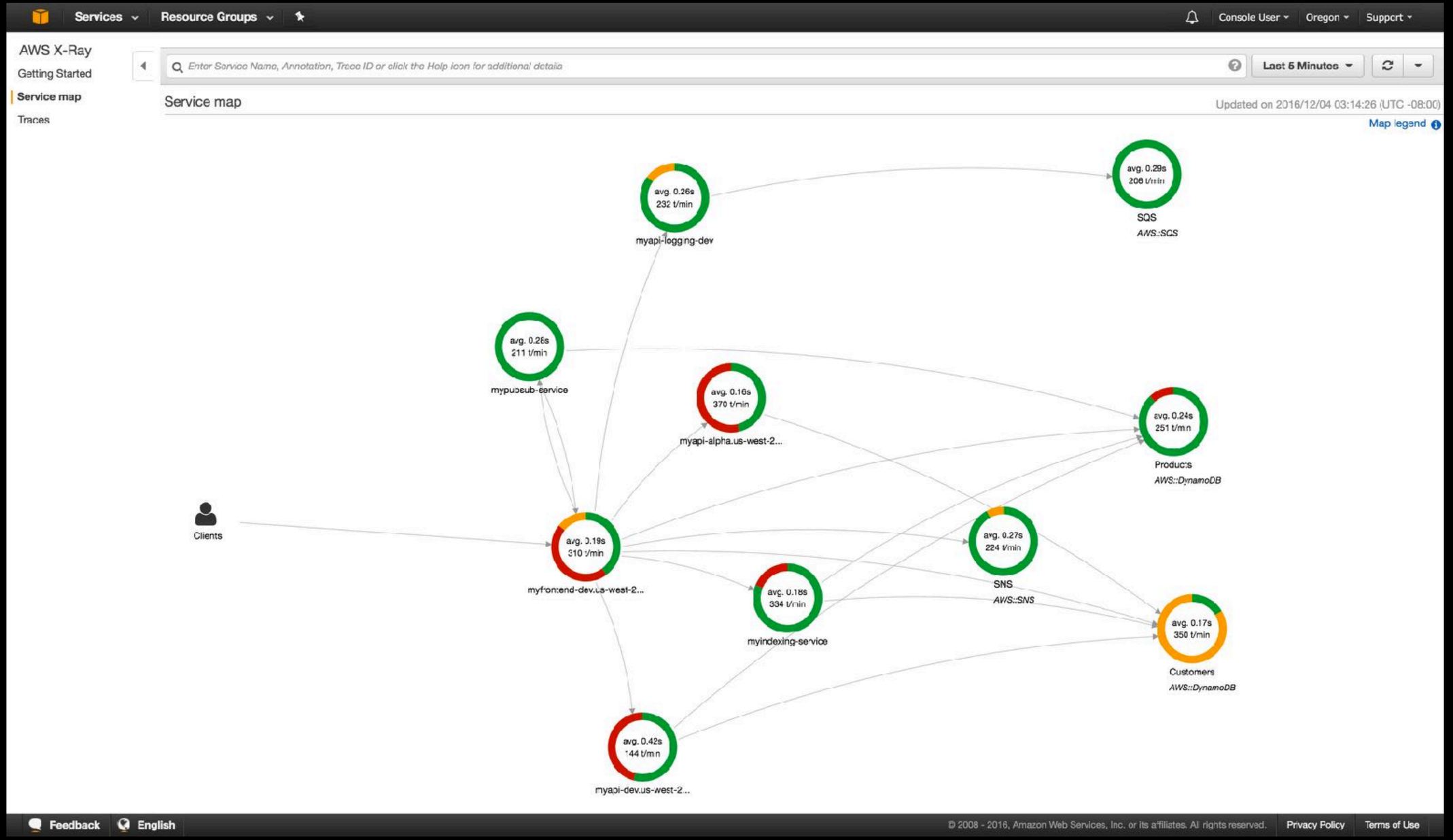
BUT HOW DO YOU TEST SERVERLESS FUNCTIONS?

AS ANY OTHER NODE.JS LIBRARY 😎

EXAMPLE?

CLAUDIA-BOT-BUILDER SPECS

AWS X RAY



#holyjs
@slobodan_

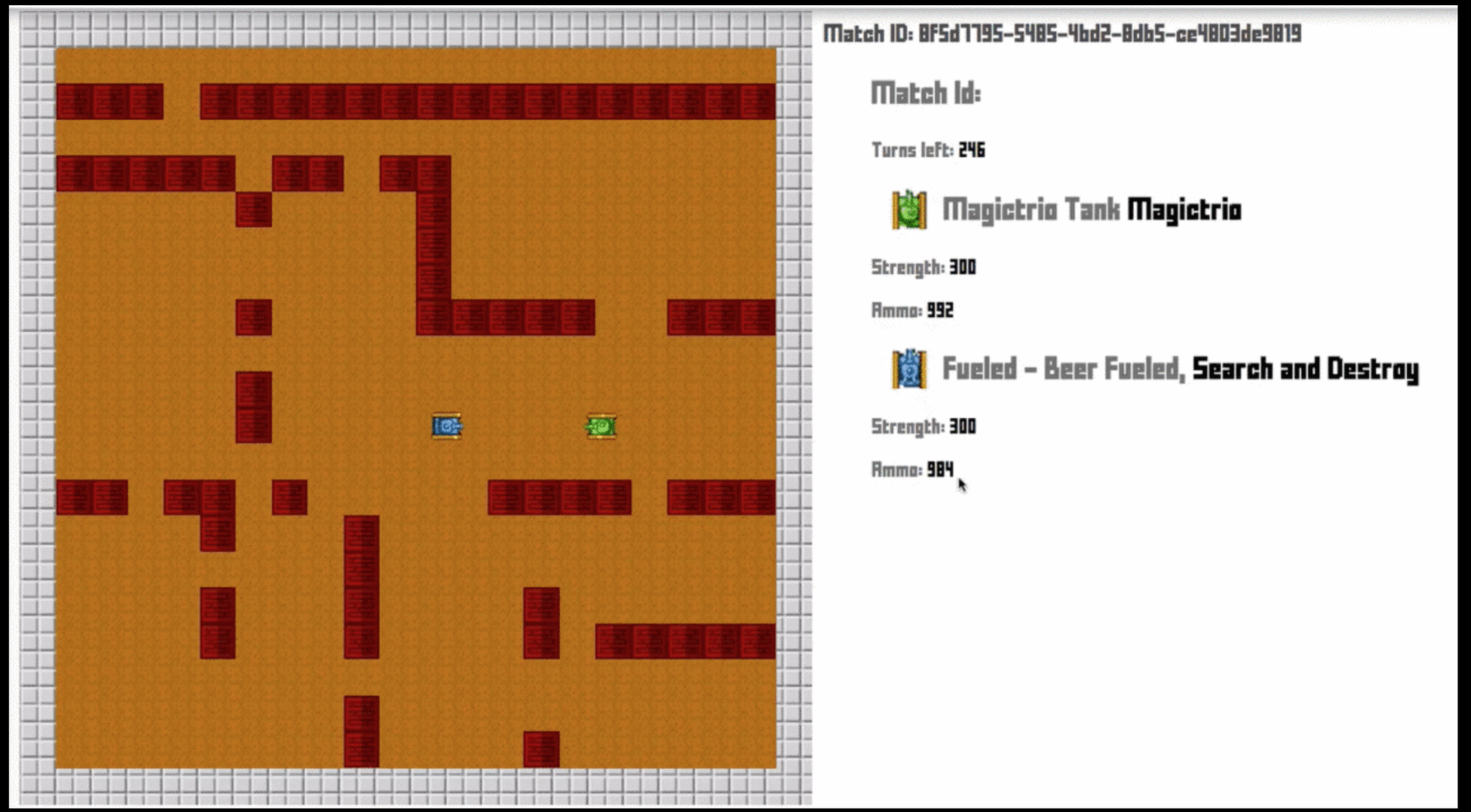
EASIEST WAY TO LEARN SERVERLESS?

GO AND PLAY WITH IT!

#holyjs
@slobodan_

FUNNIEST WAY TO LEARN SERVERLESS?

SERVERLESS.CAMP



#holyjs
@slobodan_


SENORGIF.COM

7. WHEN SHOULD I USE SERVERLESS?

LET'S START WITH WHEN YOU SHOULD NOT USE SERVERLESS?

- REAL-TIME APPS WITH WEB SOCKETS
- LOW-LATENCY APPS
- WHEN YOU NEED CUSTOM SERVER CONFIG
- WHEN YOU NEED COMPLIANCE
- LONG RUNNING TASKS
- COMPEX COMPUTING
- WHEN YOU NEED TO PROVIDE SLA

OK, BUT WHAT IS IT GOOD FOR?

- WEB API
- BACKEND FOR THE SINGLE PAGE APP
- CHATBOTS
- QUICK PROTOTYPES THAT CAN EVOLVE TO A REAL SOLUTIONS
- PROCESSING FILES OR ANY KIND OF DATA
- IoT
- MANY, MANY OTHER THINGS

HOW CAN YOU START USING IT TODAY?

SIDE PROJECT OR SOMETHING SMALL

AWS LAMBDA + AWS ECOSYSTEM

MOVE SOME NON-ESSENTIAL FEATURE TO SERVERLESS

EXISTING APP + FEATURE IN AWS LAMBDA

8. IS SERVERLESS SECURE?

AWS LAMBDA SECURITY

- SERVER “EXISTS” FOR FEW HUNDREDS OF MS
- READ-ONLY, EXCEPT /TMP
- NO DEDICATED SERVER
- SANDBOX / CONTAINERS
- NO CONFIGURATION
- AUTO SCALING
- AWS ECOSYSTEM

PROBLEMS

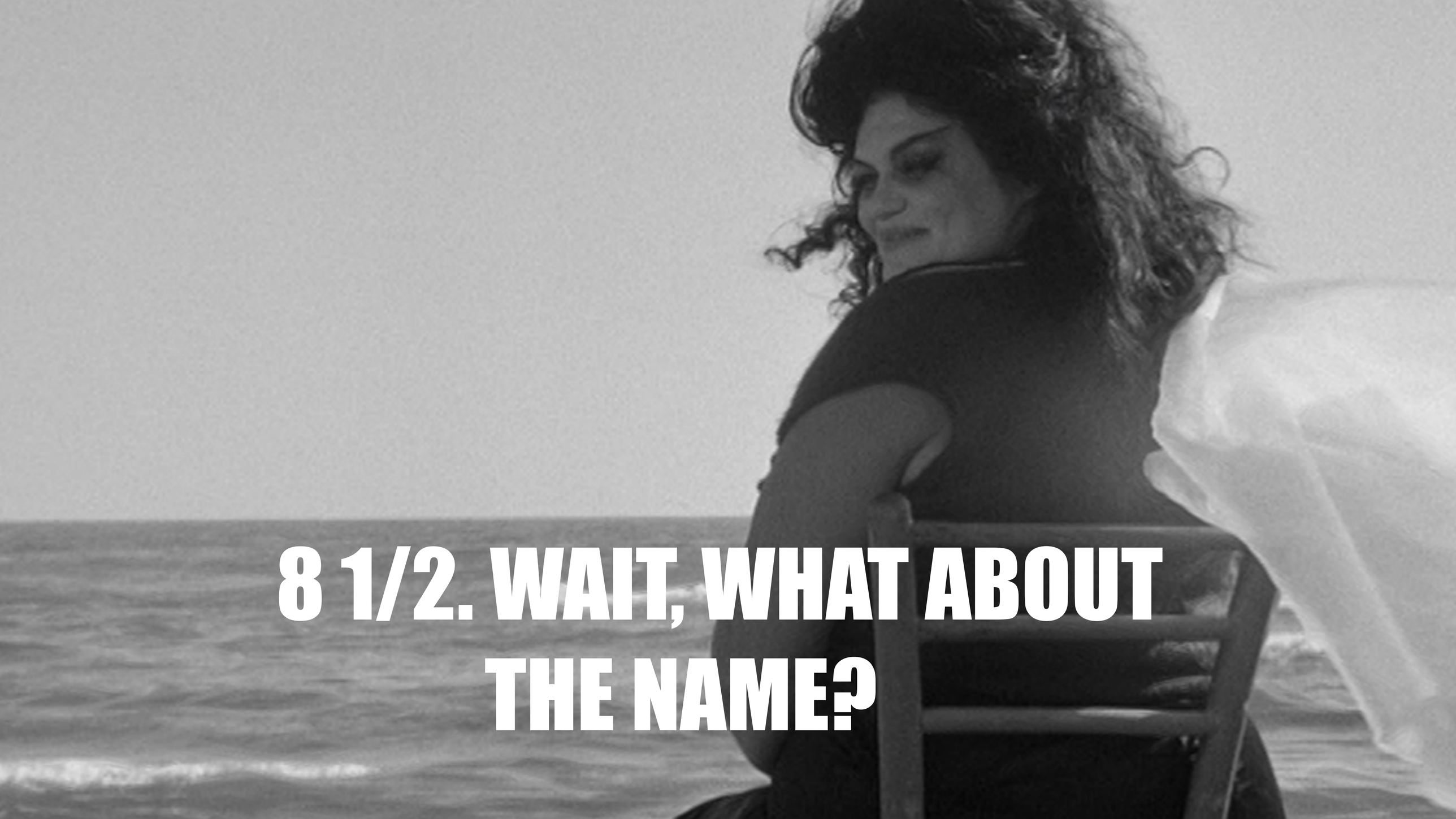
- BAD CODE
- BAD PERMISSIONS
- 3RD PARTY LIBRARIES
- PROBLEMS WITH OTHER BUILDING BLOCKS, SUCH AS SQL INJECTION
- KEYS AND SECRETS

SO, IS IT SECURE?

IT DEPENDS ON YOU

THERE ARE VULNERABLE PARTS,
BUT IT'S A BETTER SET OF PROBLEMS
TO DEAL WITH

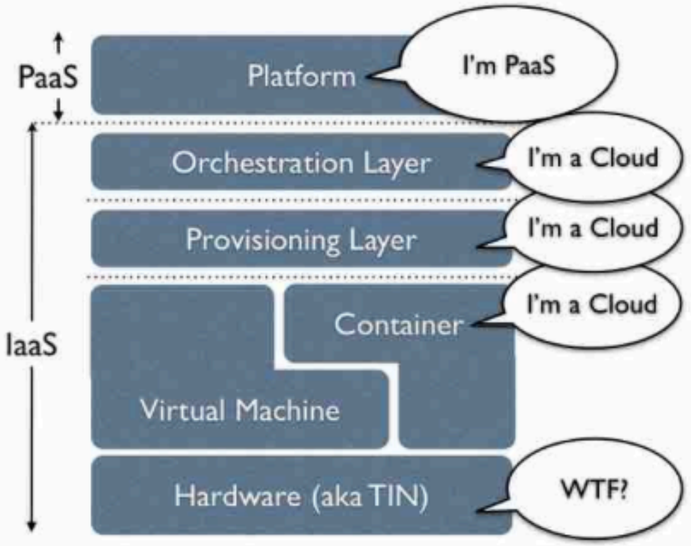

3½


8 1/2. WAIT, WHAT ABOUT
THE NAME?

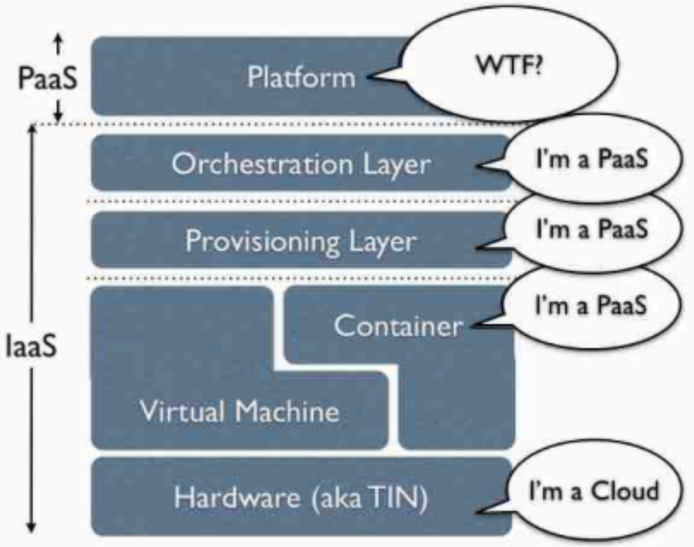
LESS
IS
MORE

SERVERLESS
IS
MORE SERVERS

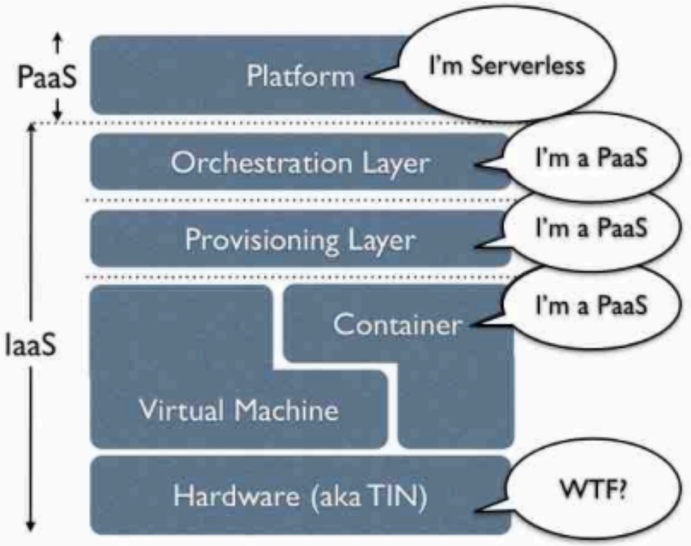
Sometimes, the best thing you can do in technology is change your name



2011



2013



2016

swardley @swardley · Apr 17

"But serverless still has servers" ... it's not the point. Here's the best diagram I have to explain what "serverless" is pic.twitter.com/LagdrqH0Pc

9 194 277

#holyjs
@slobodan_

2012

“WHY THE FUTURE OF SOFTWARE AND APPS IS SERVERLESS”

BY KEN FROMM

2014

AWS LAMBDA

2015

API GATEWAY

“SERVERS ARE DEAD..”

BY ANT STANLEY

JAWS
BECOMES
SERVERLESS FRAMEWORK

2016

“SERVERLESS” IS JUST A NAME.
WE COULD HAVE CALLED IT “JEFF”

PAUL JOHNSTON

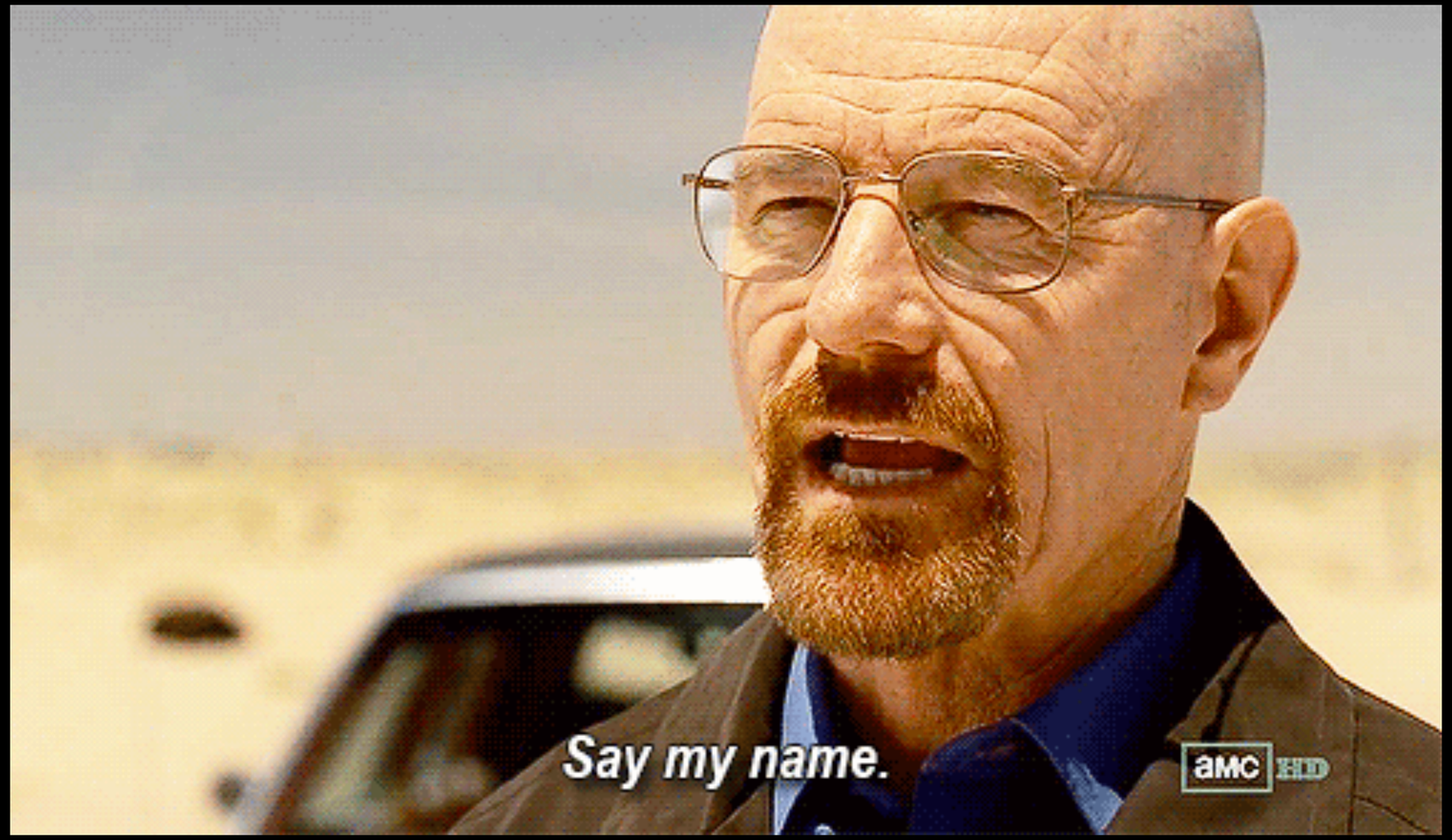
“IT IS SERVERLESS THE SAME WAY
WiFi IS WIRELESS”

GOJKO ADŽIĆ

#holyjs
@slobodan_

NOW

SERVERLESS IS EVERYWHERE

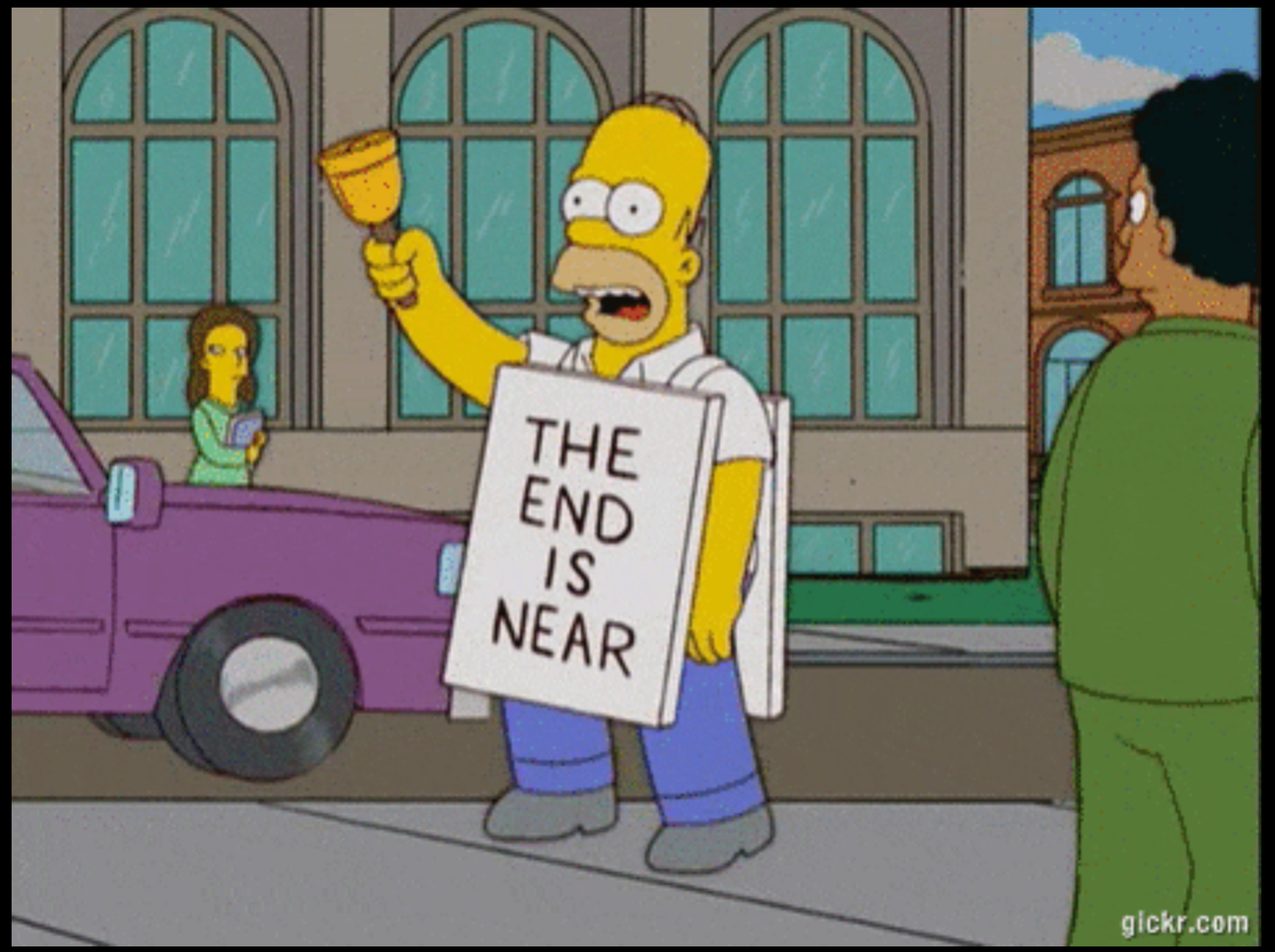

Say my name.
AMC HD

SUMMARY

- SERVERLESS IS FULLY MANAGED FaaS THAT HELPS YOU TO FOCUS ON YOUR BUSINESS LOGIC
- EVENT-DRIVEN - MANY THINGS CAN TRIGGER FUNCTION
- AUTO SCALING AND AUTO FAILOVER
- IT WORKS NICE WITH NODE.JS
- 3 MAJOR NODE.JS FRAMEWORKS
- IT'S TRICKY TO DEBUG, SO WRITE TESTS
- GO AND TRY IT, IT'S EASY TO LEARN 🖖

ALSO, SERVERS STILL EXIST
BUT THEY ARE NOT YOUR PROBLEM

ALSO, SERVERS STILL EXIST
BUT THEY ARE NOT YOUR PROBLEM


THE END IS NEAR
gickr.com


THAT'S IT.
THANKS!


КОНЕЦ

QUESTIONS?

#holyjs

@slobodan_

bit.ly/holyjs-serverless