

Real World Serverless

David Schmitz
Senacor Technologies

DAVID SCHMITZ
SENACOR TECHNOLOGIES
@KOENIGHOTZE

4 QUESTIONS

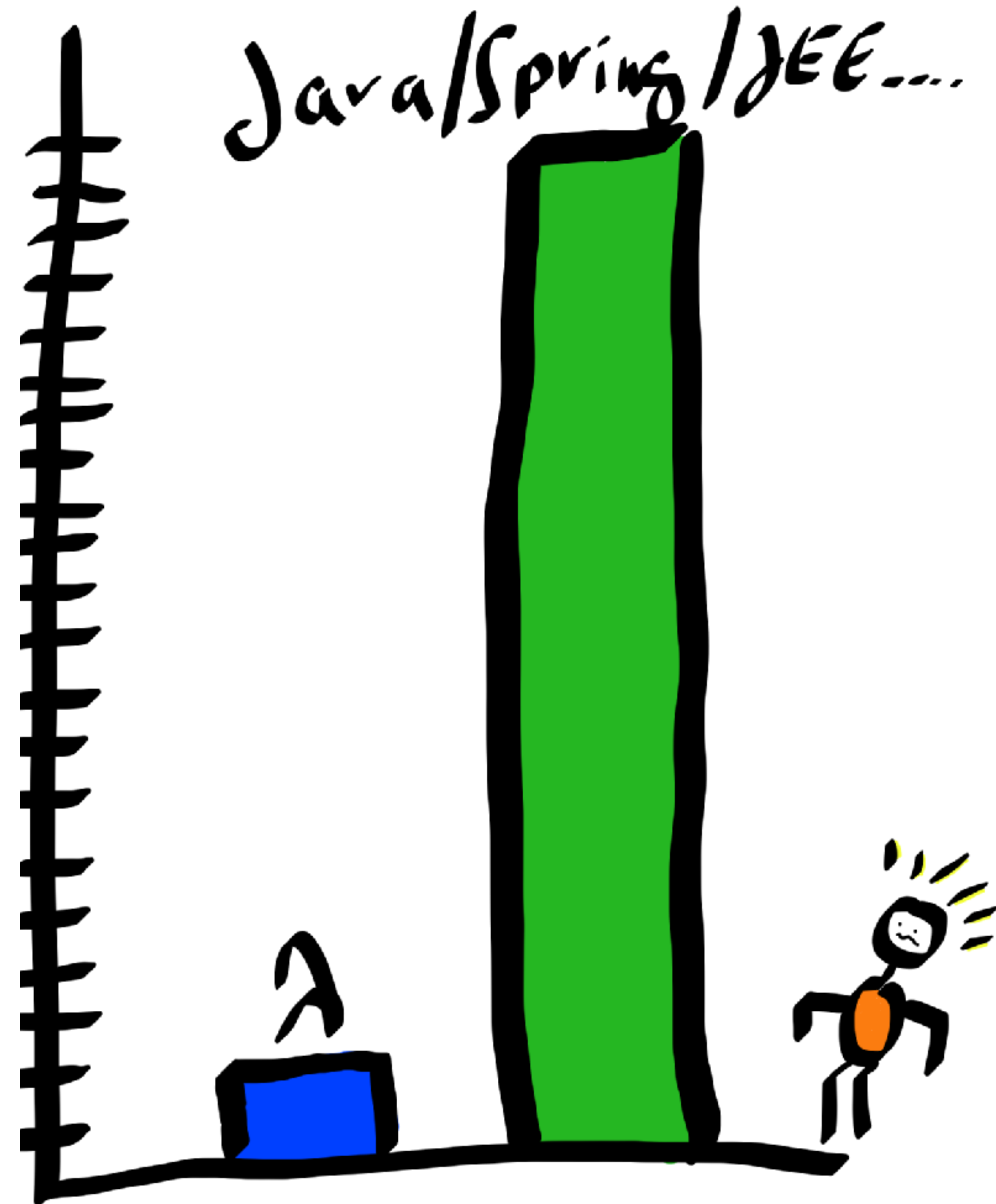
MICROSERVICES
CONTAINERS
CLOUD
SERVERLESS

EXPECTATION MANAGEMENT



VENDOR?

CLOUD GURU?



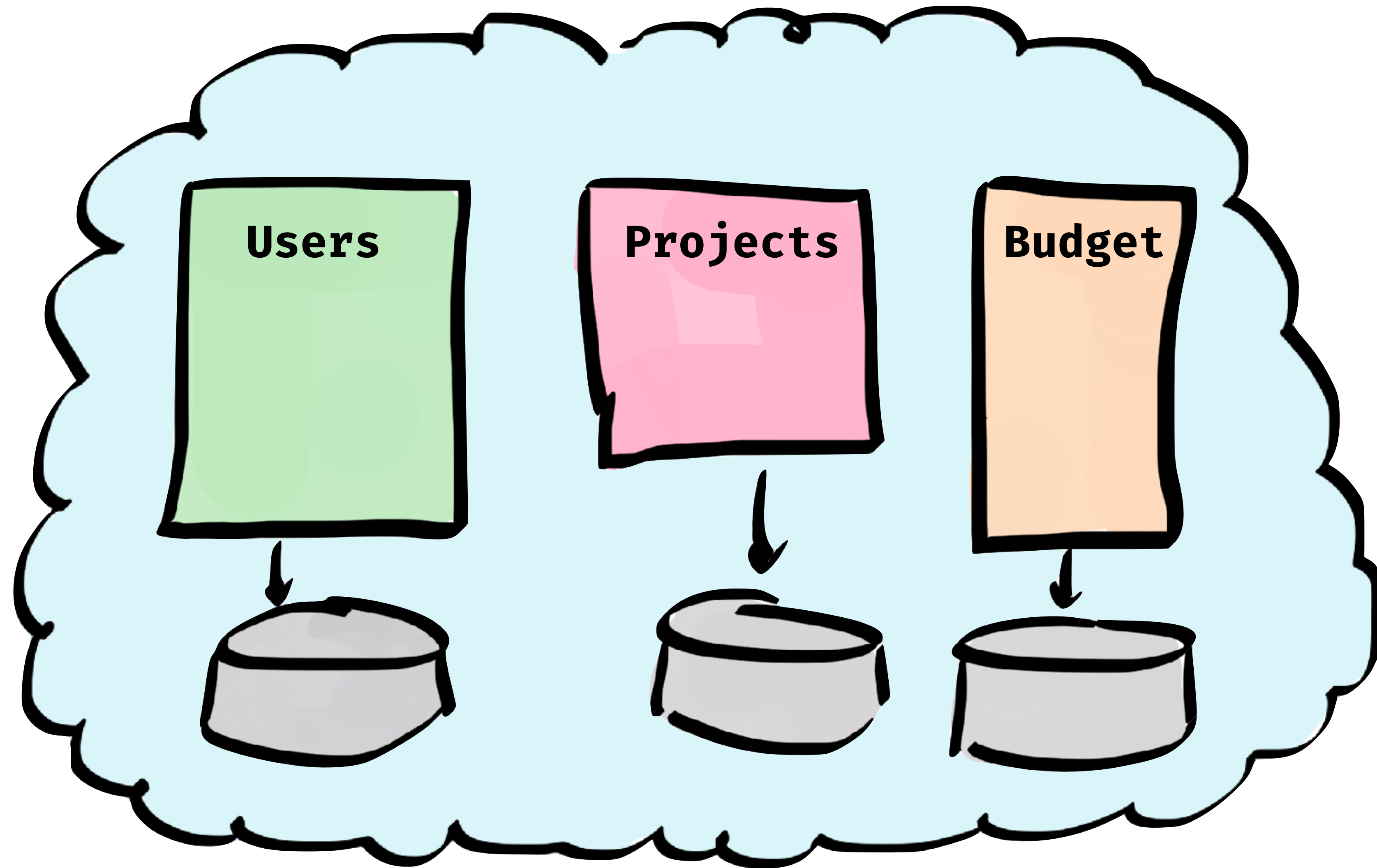
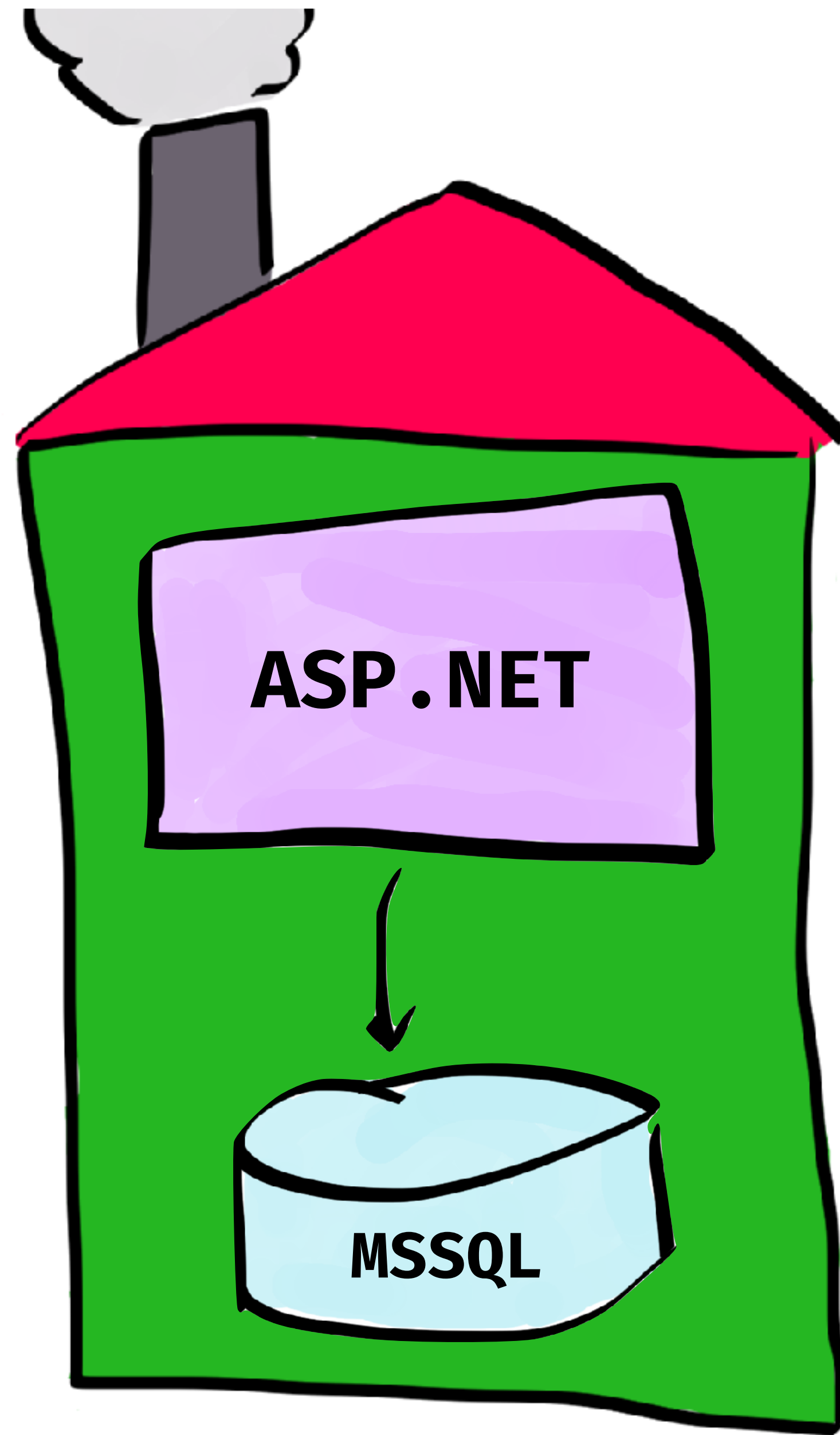
THINGS ARE MOVING
FAST

**“Some of the things we learned,
while moving an application
(partially) to AWS Lambda.”**

–Me



EMPLOYEE ADMINISTRATION PLATFORM

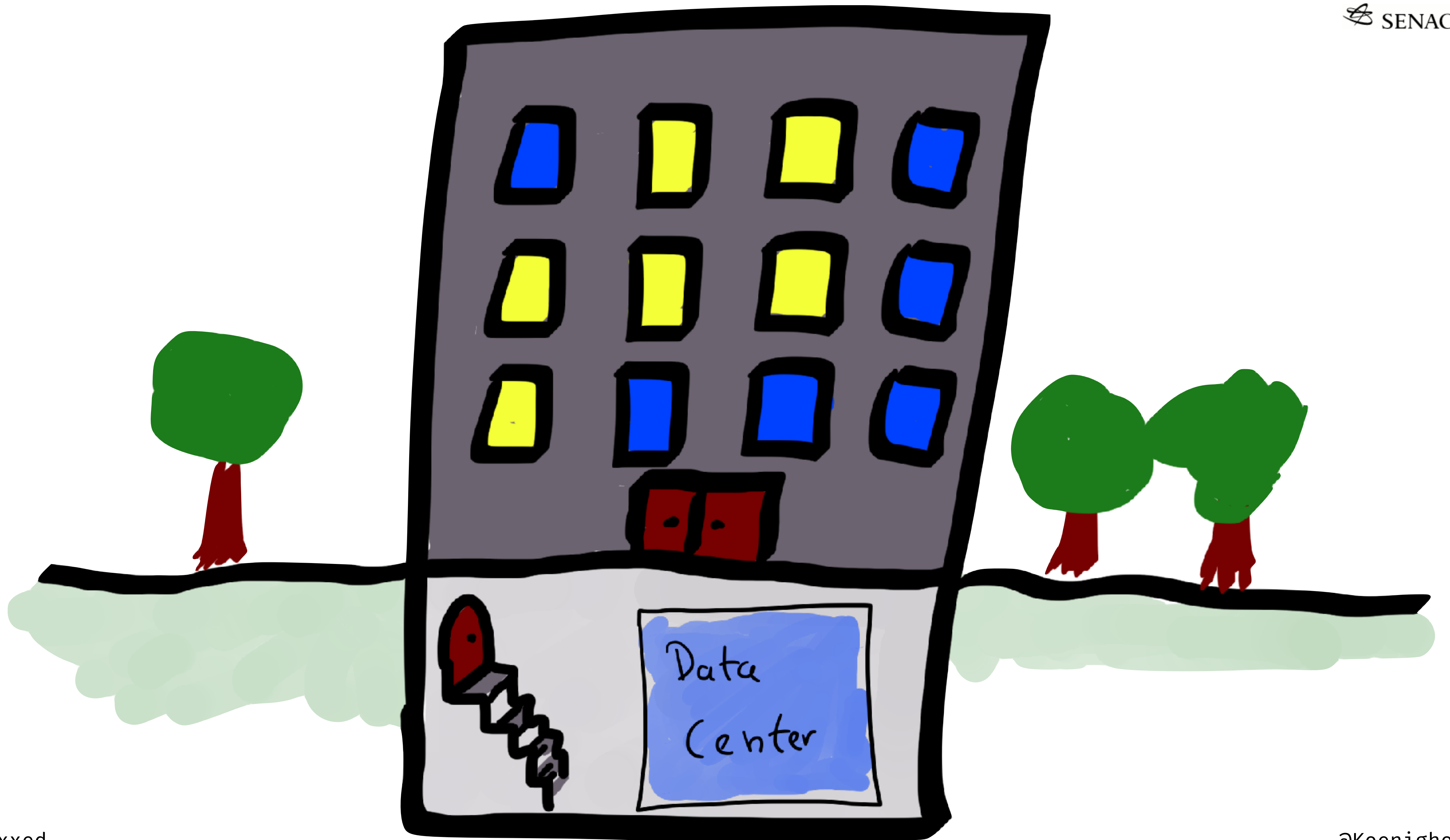


RUNTIME MATTERS
SCALE(LAMBDA) >>> REST
LAMBDA-TIMTOWTDY
DEPLOYMENT STILL SUCKS
SECURITY IS A THING
LAMBDA WITCHCRAFT



AWS LAMBDA 1-0-1

Ye olde days



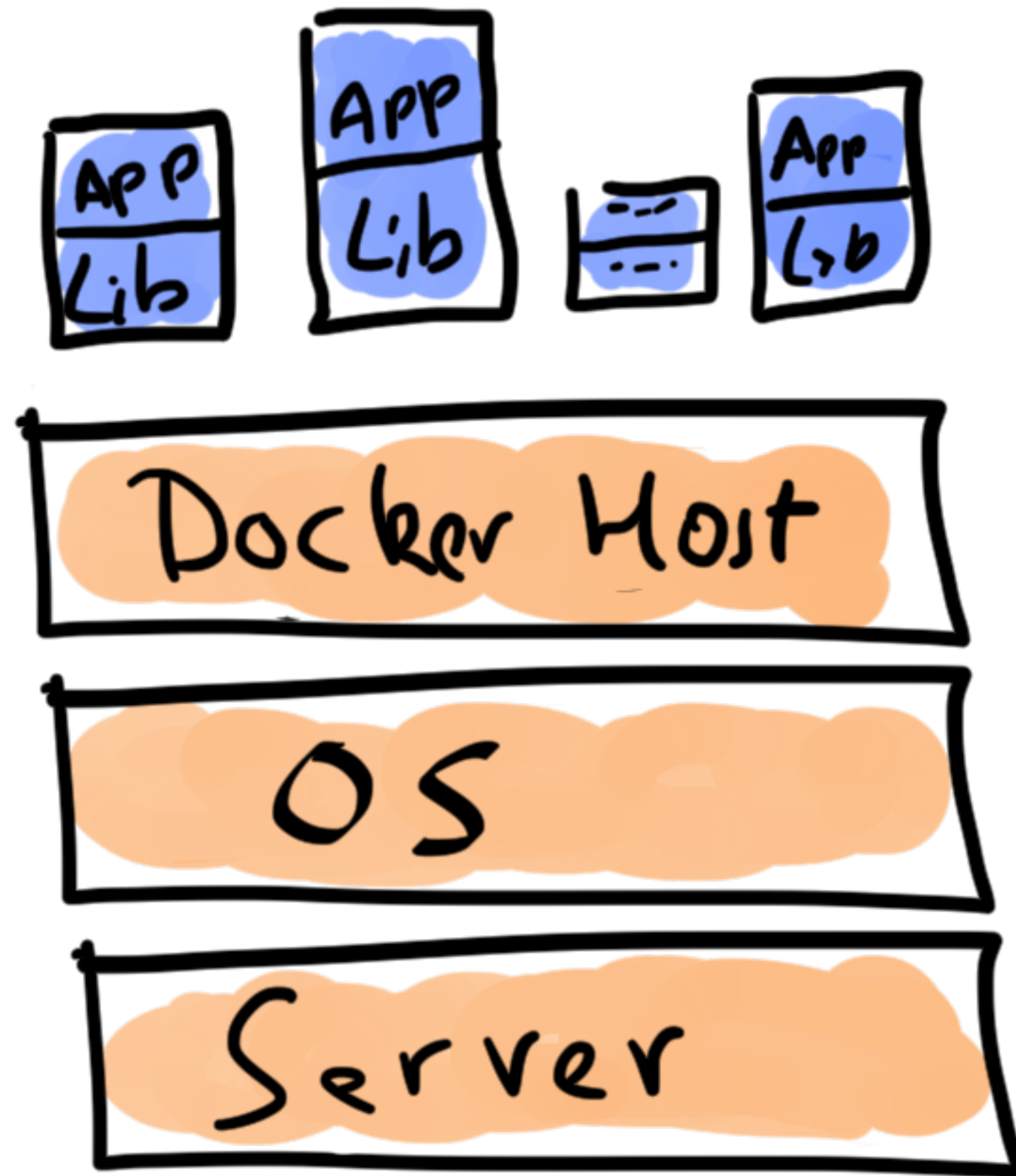
INFRASTRUCTURE AS A SERVICE



PLATFORM AS A SERVICE

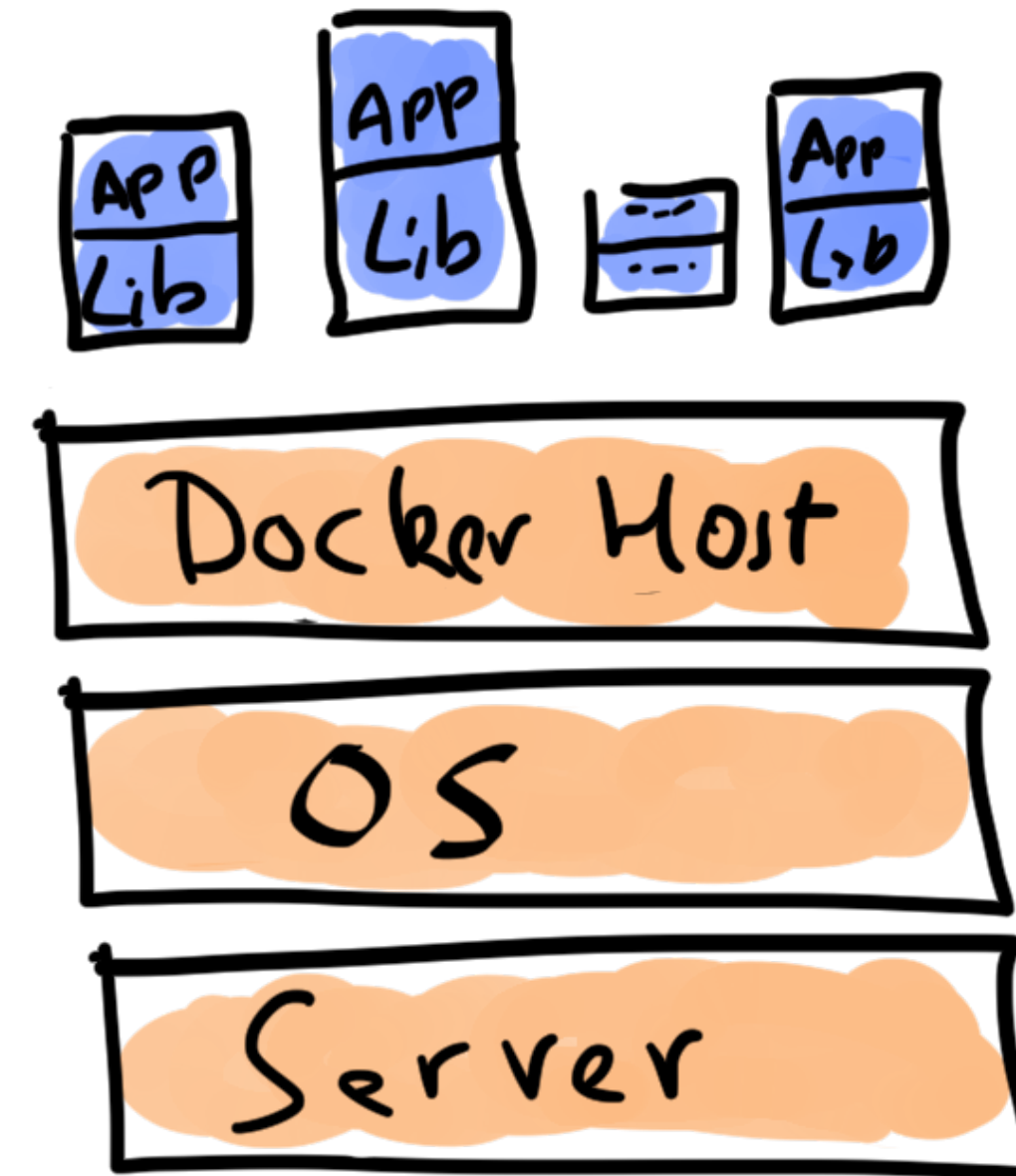


CONTAINERS

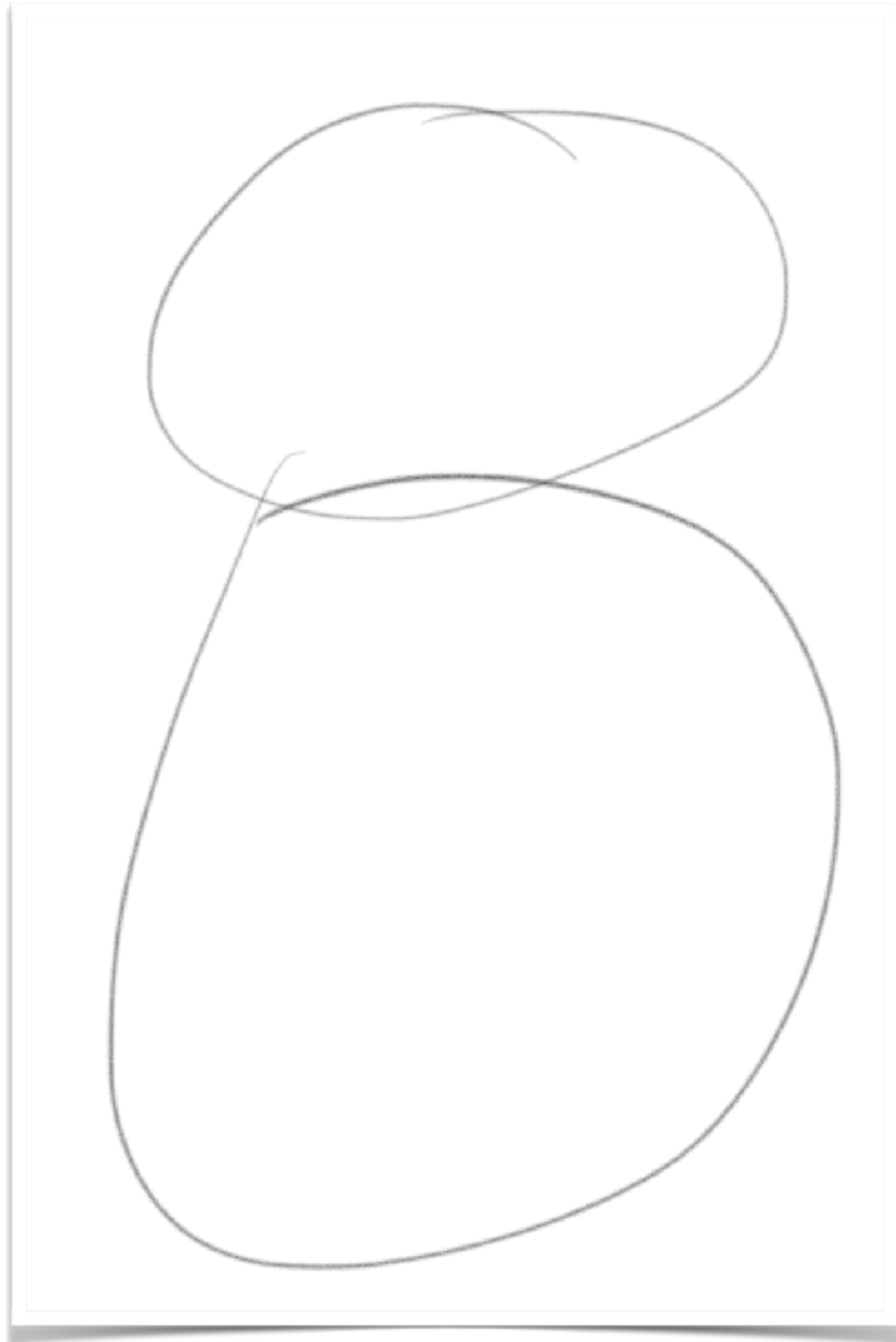


Provisioning
Scaling
Monitoring
State
Security
Deploying

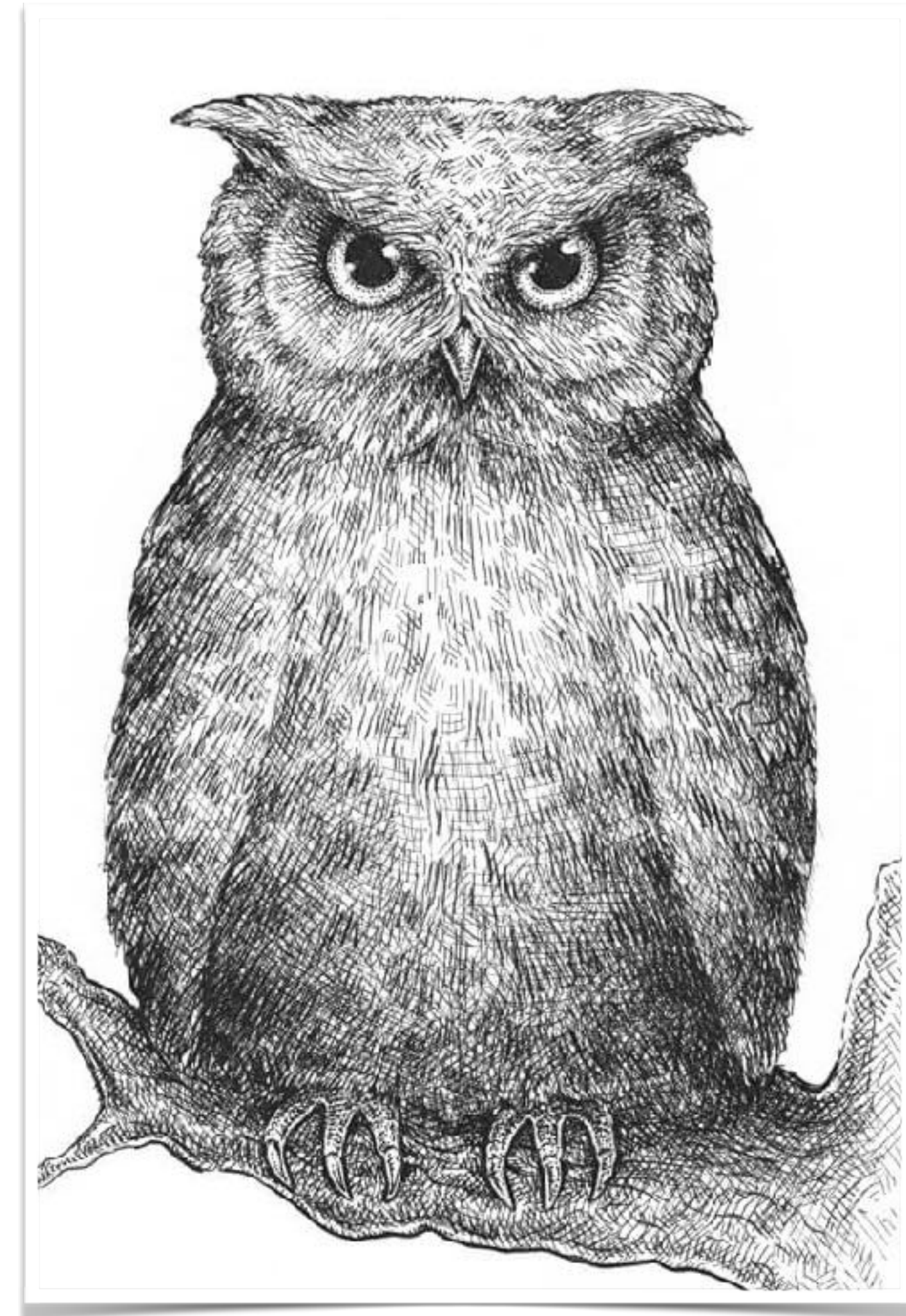
...



HOW TO DRAW AN OWL?



1. Draw some circles



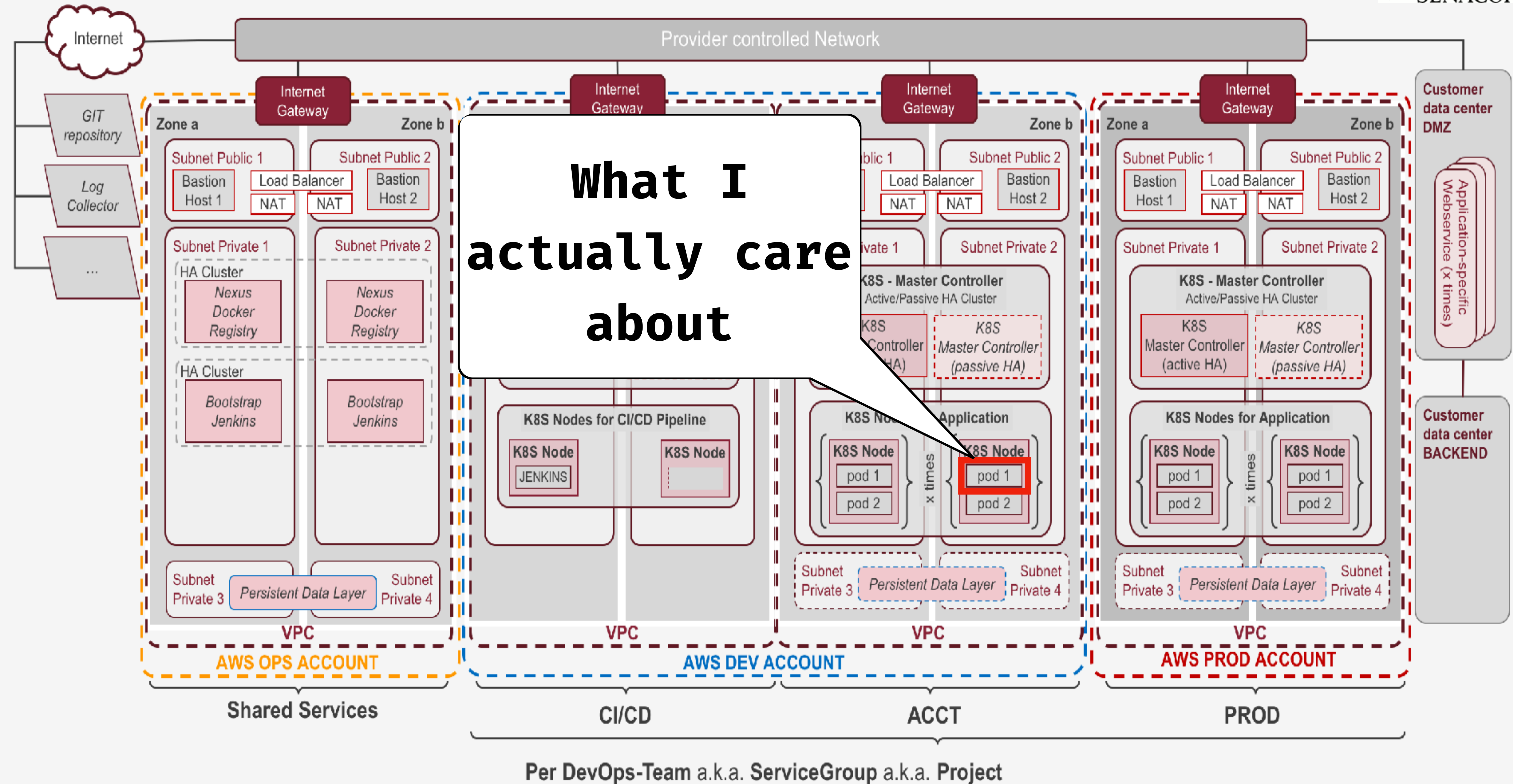
2. Draw the rest of the damn owl

**DESIGNING DISTRIBUTED
SYSTEMS IS DIFFICULT**



kubernetes







ENTER SERVERLESS

SO **WHAT** IS SERVERLESS?

Serverless computing allows you to build and run applications and services without thinking about servers. Serverless applications don't require you to provision, scale, and manage any servers..

Building serverless applications means that your developers can focus on their core product instead of worrying about managing and operating servers or runtimes..

Serverless computing allows you to build and run applications and services **without thinking about servers**. Serverless applications **don't require you to provision, scale, and manage any servers...**

Building serverless applications means that **your developers can focus on their core product** instead of worrying about managing and operating servers or runtimes...

FOCUS ON WHAT MATTERS

**...AT LEAST THAT IS
WHAT THEY SELL ;)**

aws





AWS CHEAT SHEET

LAMBDA

—

SERVERLESS COMPUTE

S3

—

DROPBOX

IAM

—

SECURITY

CLOUDFORMATION — PROVISIONING

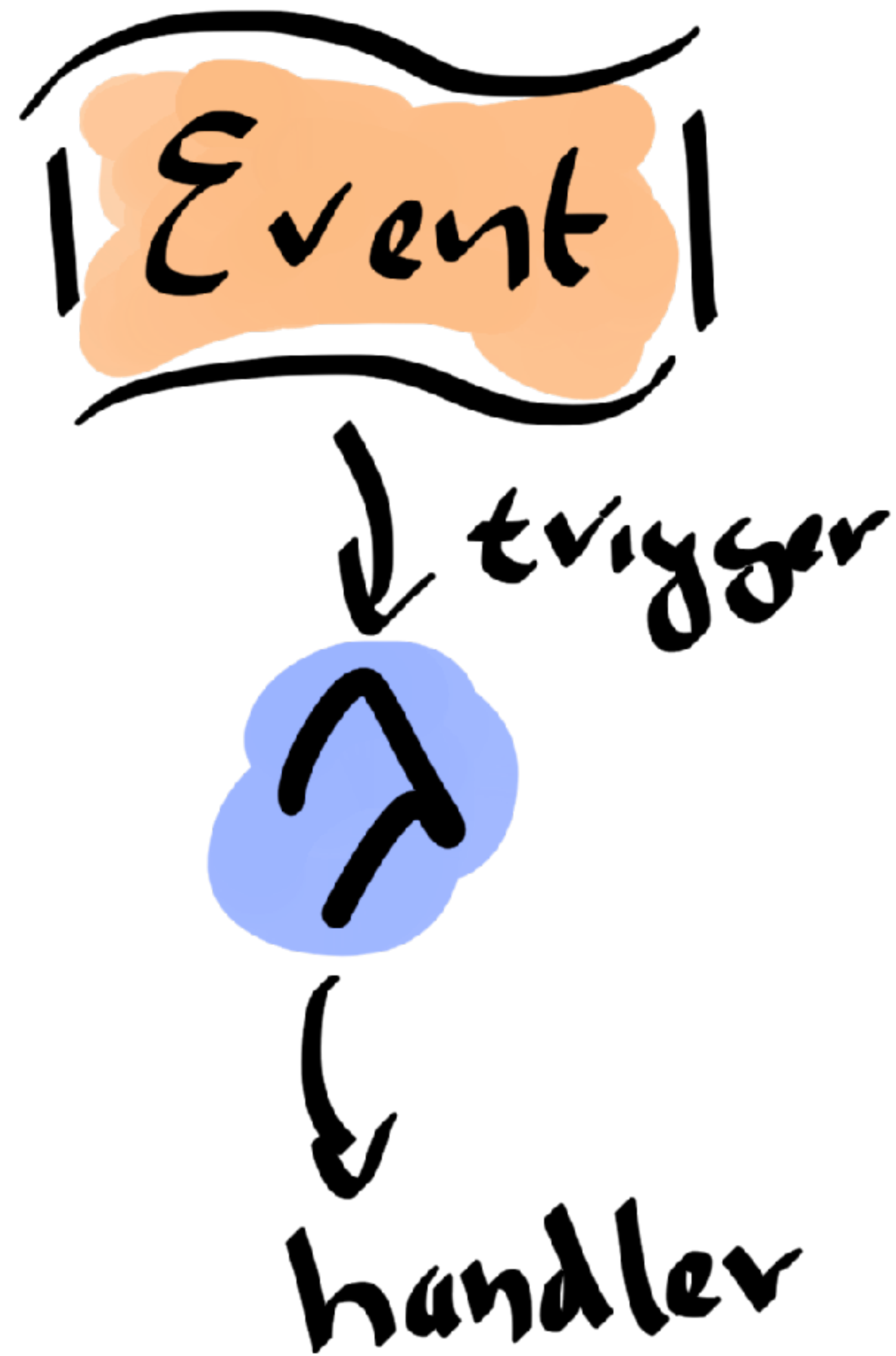
WHAT IS A “LAMBDA FUNCTION”?

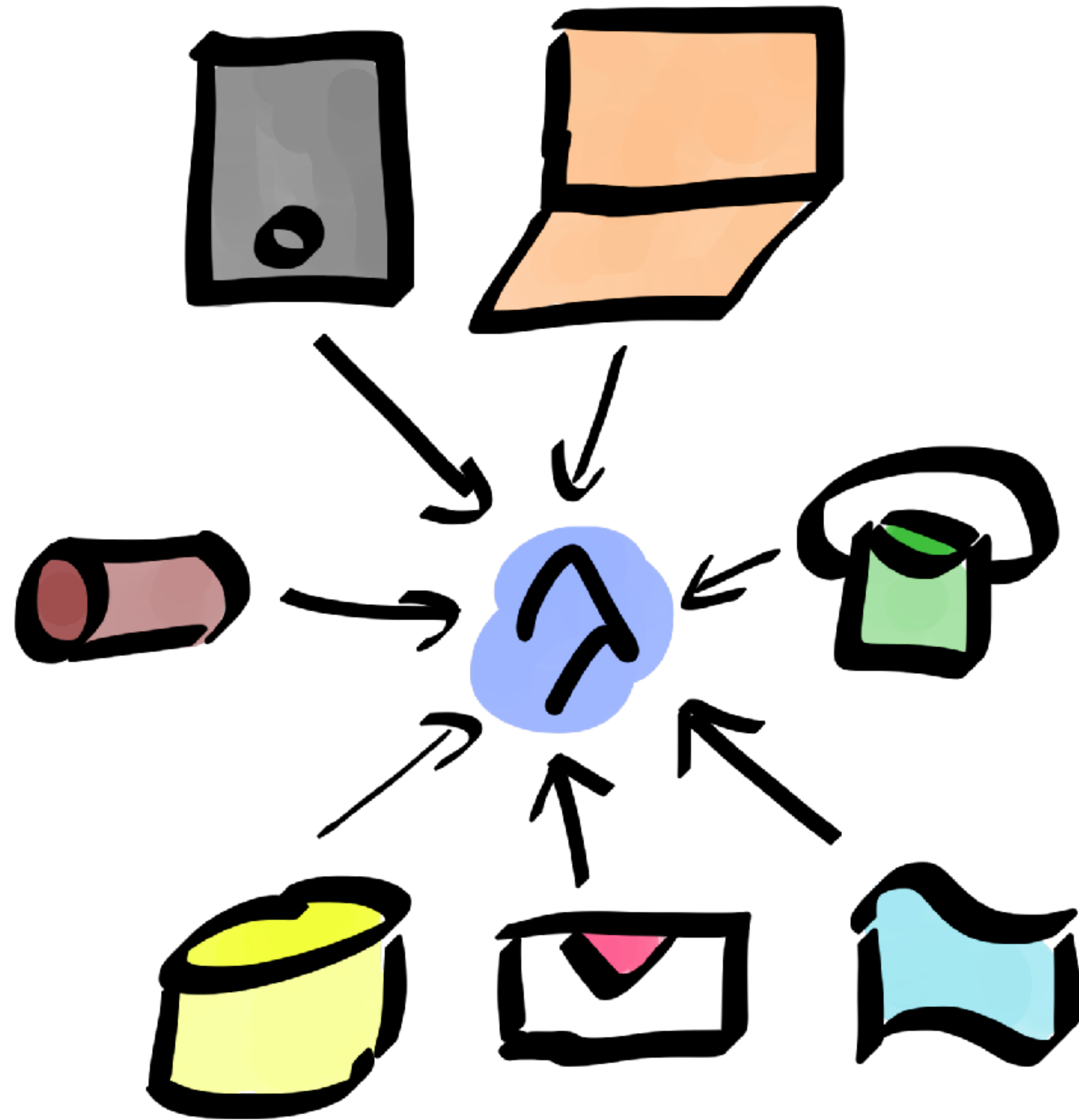
Event driven

Stateless  **Single purpose**

Asynchronous

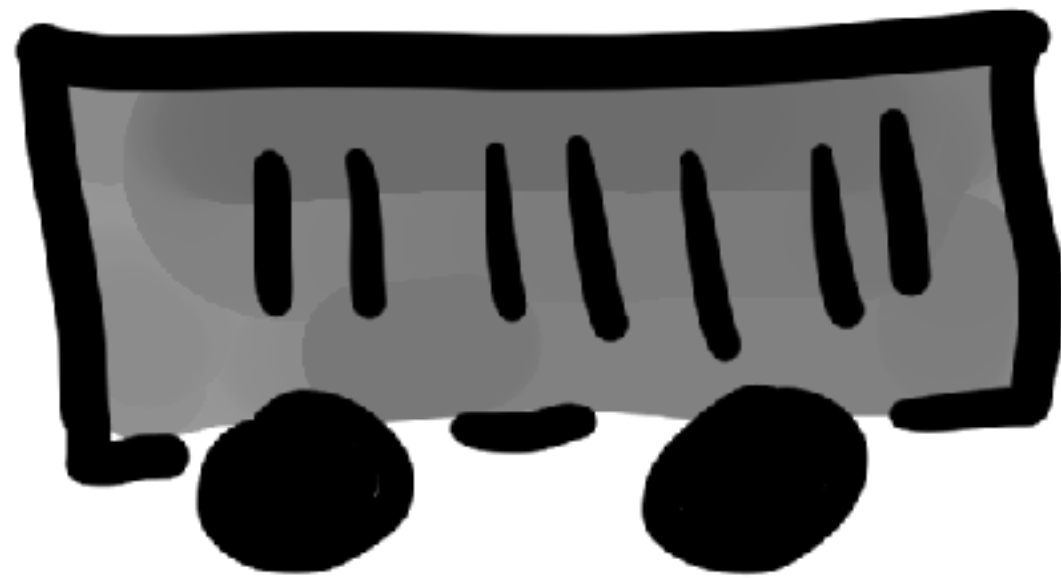
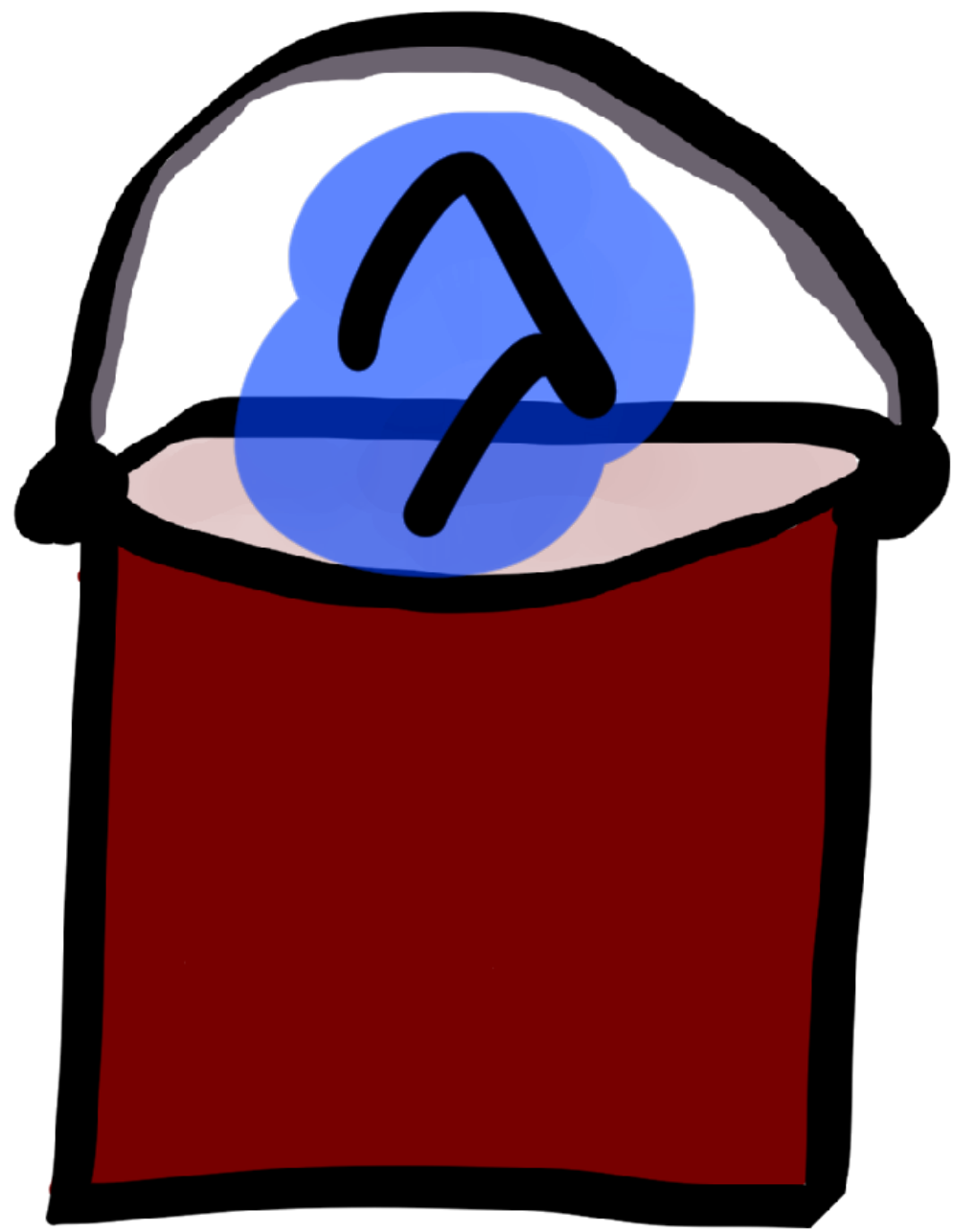
EXECUTION MODEL



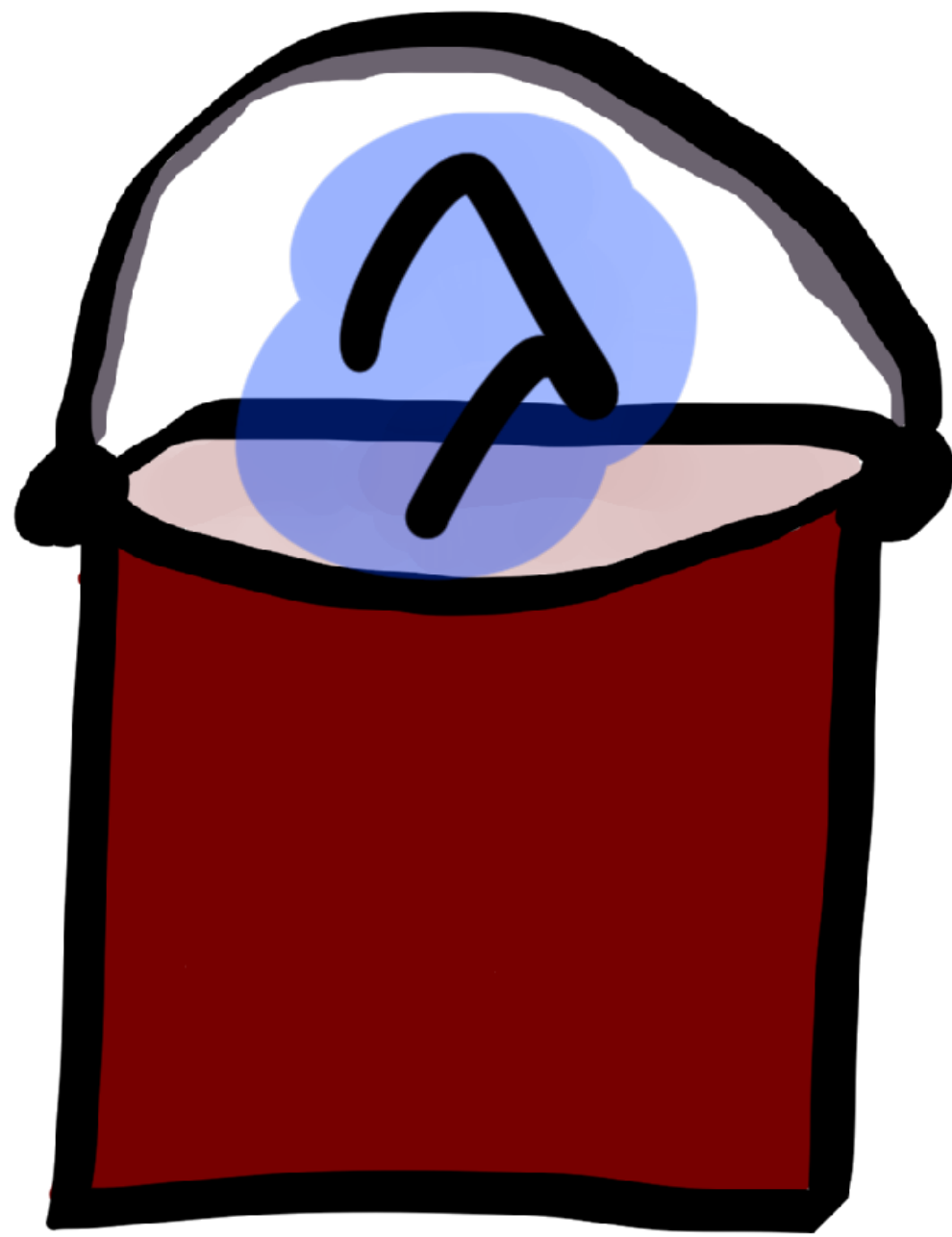






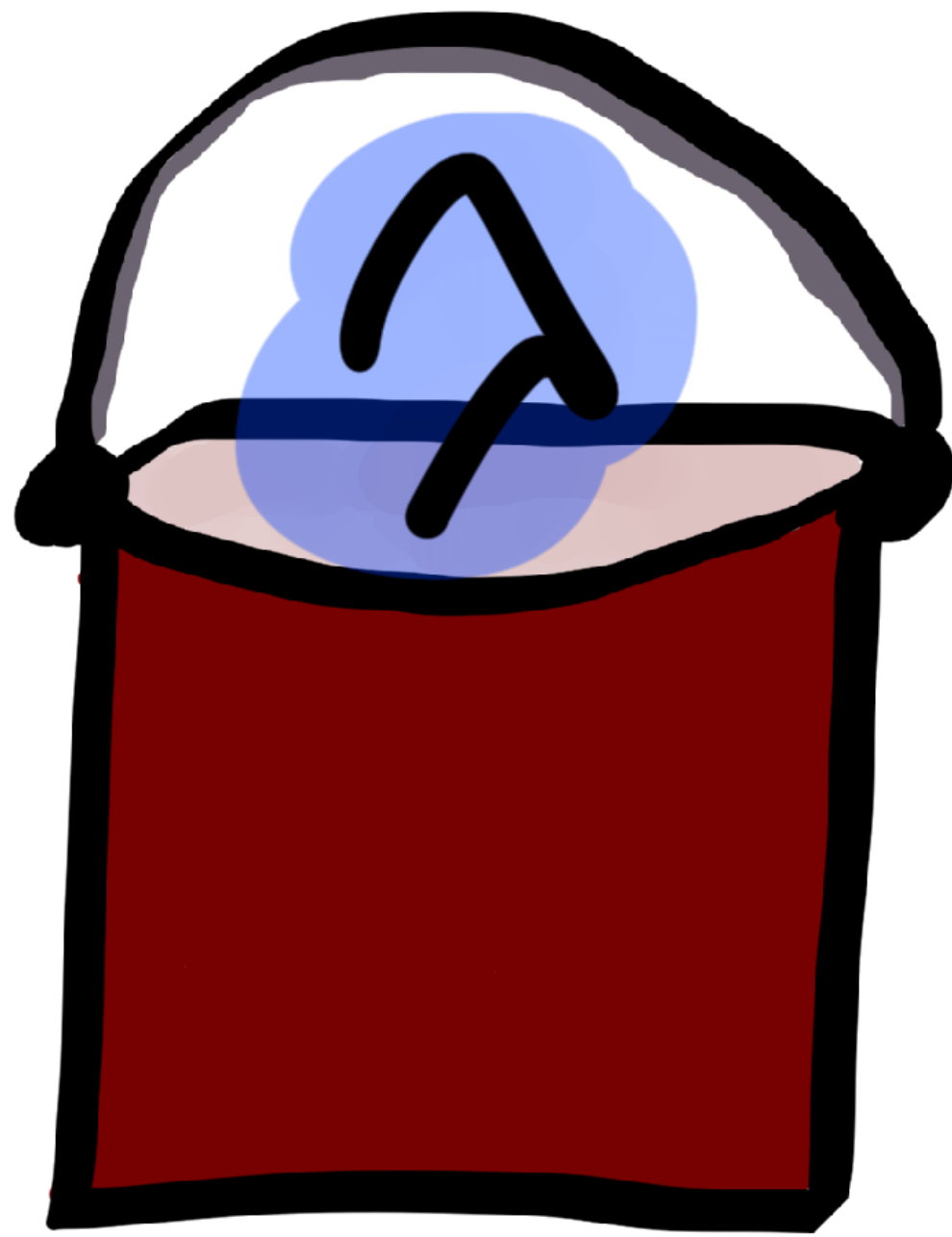




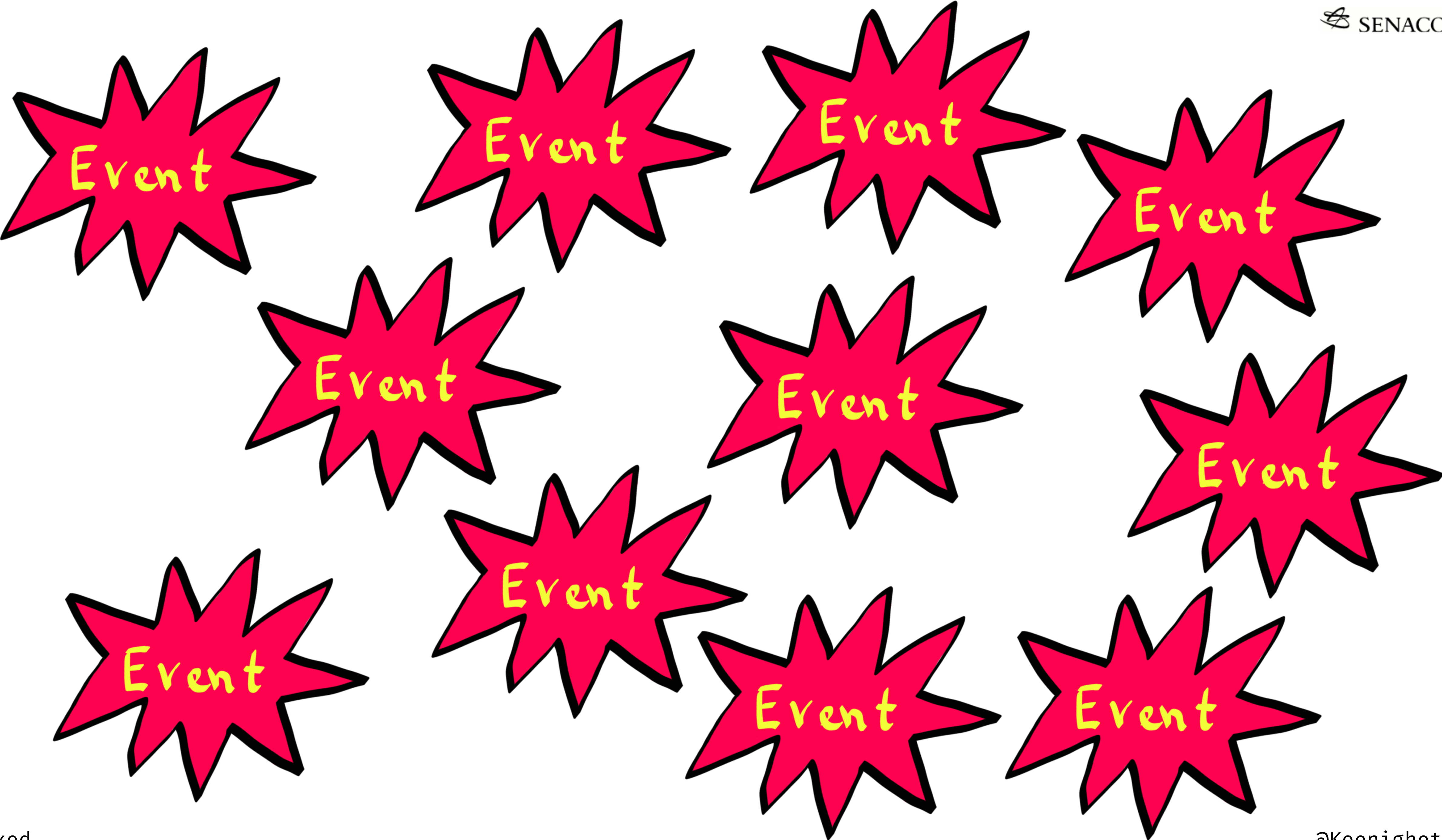


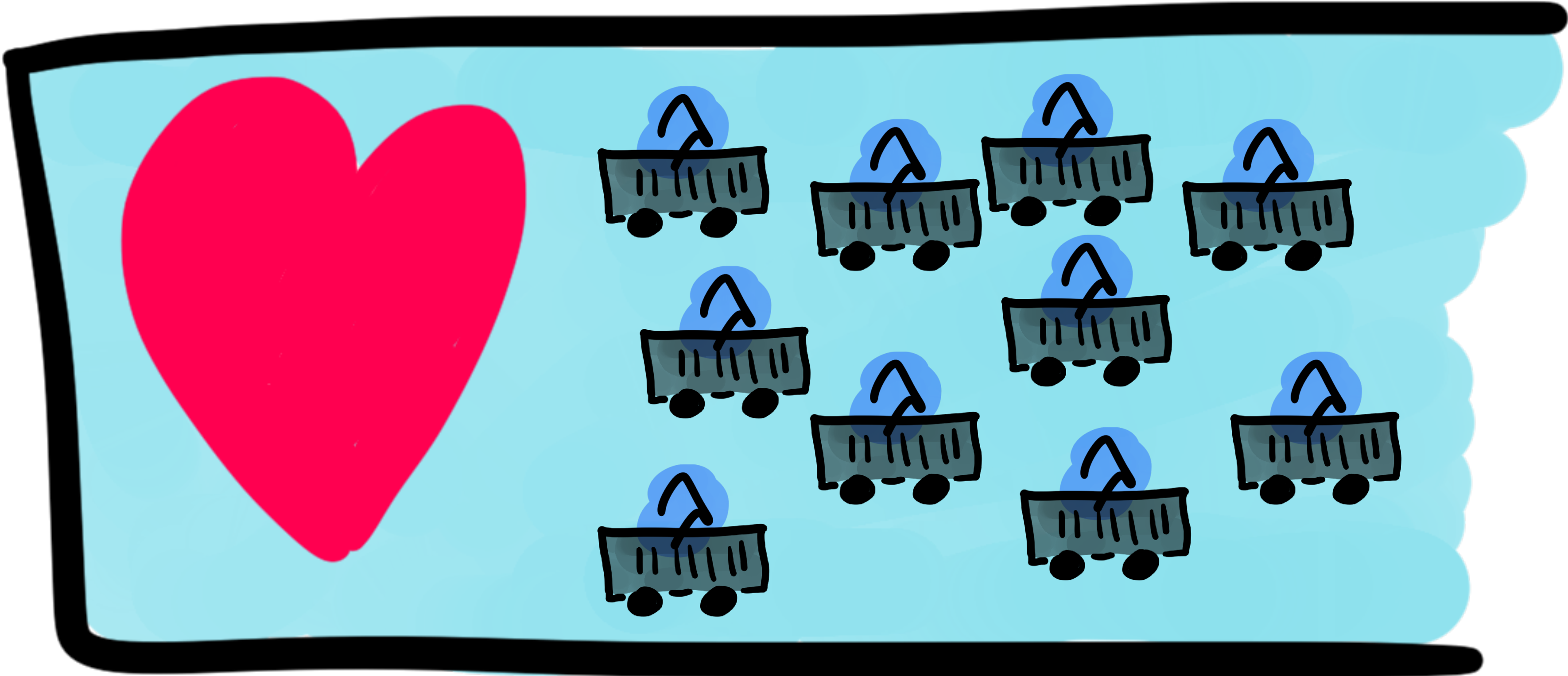


Some time later..



SCALE BY REQUEST



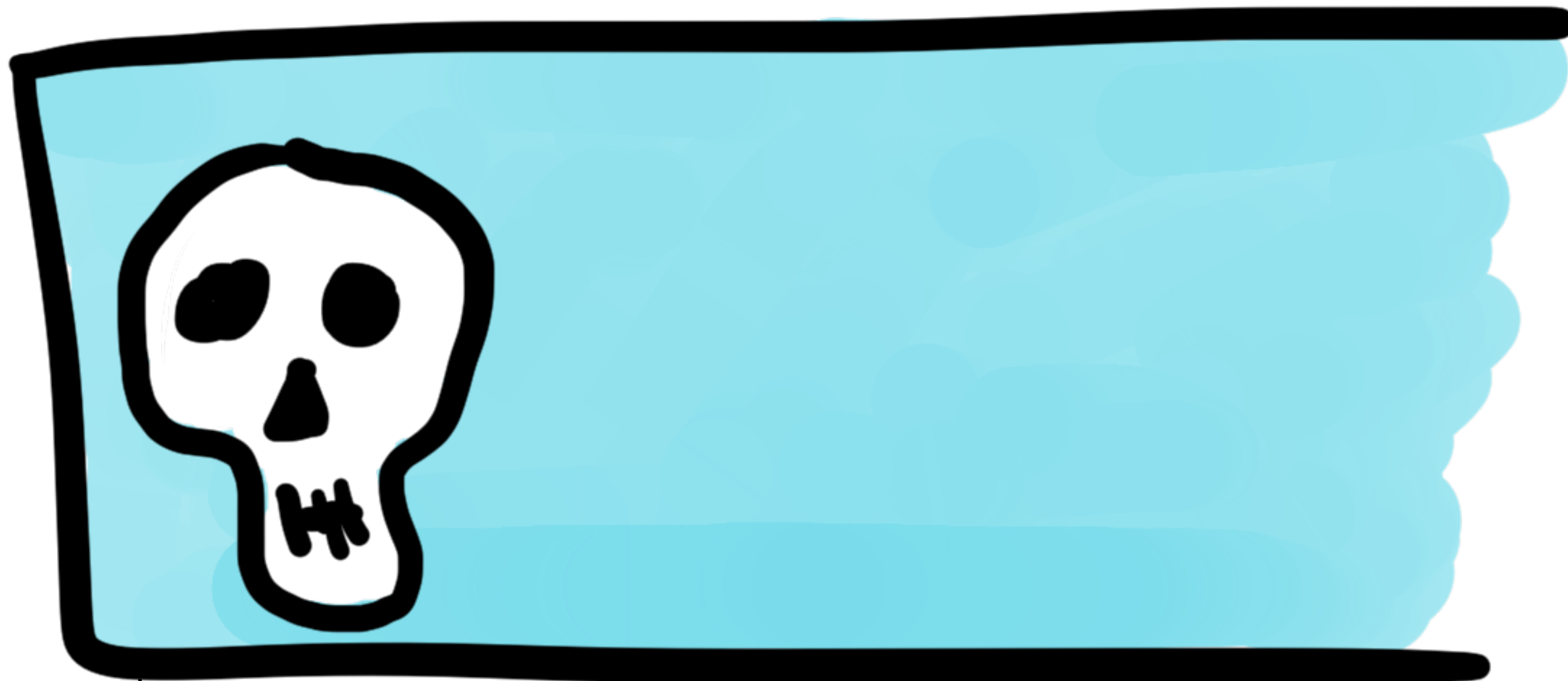


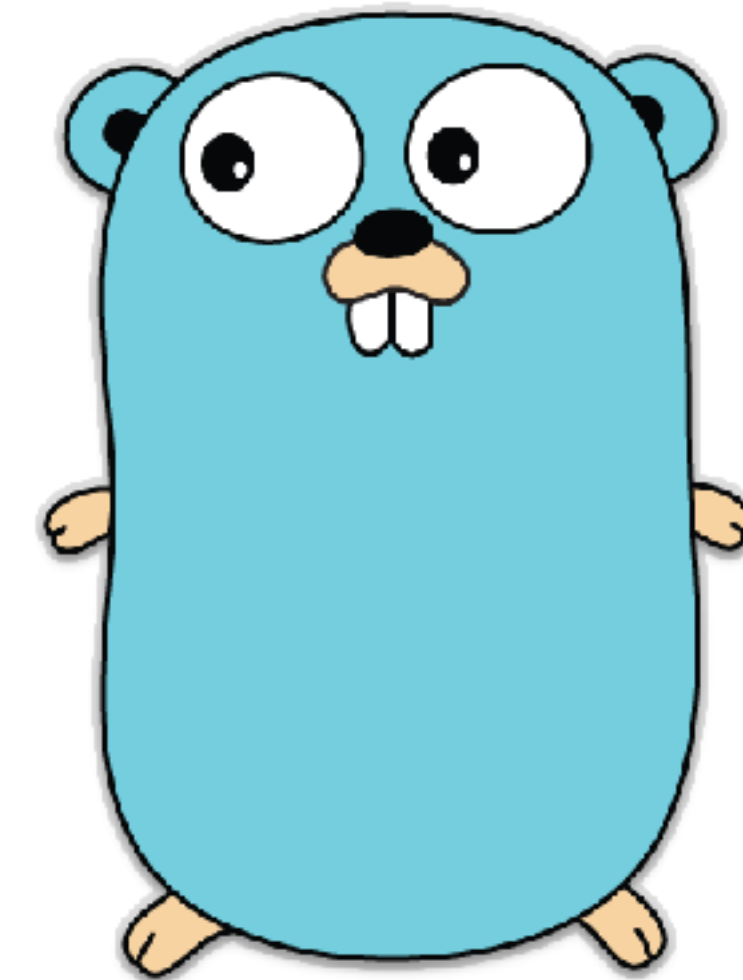
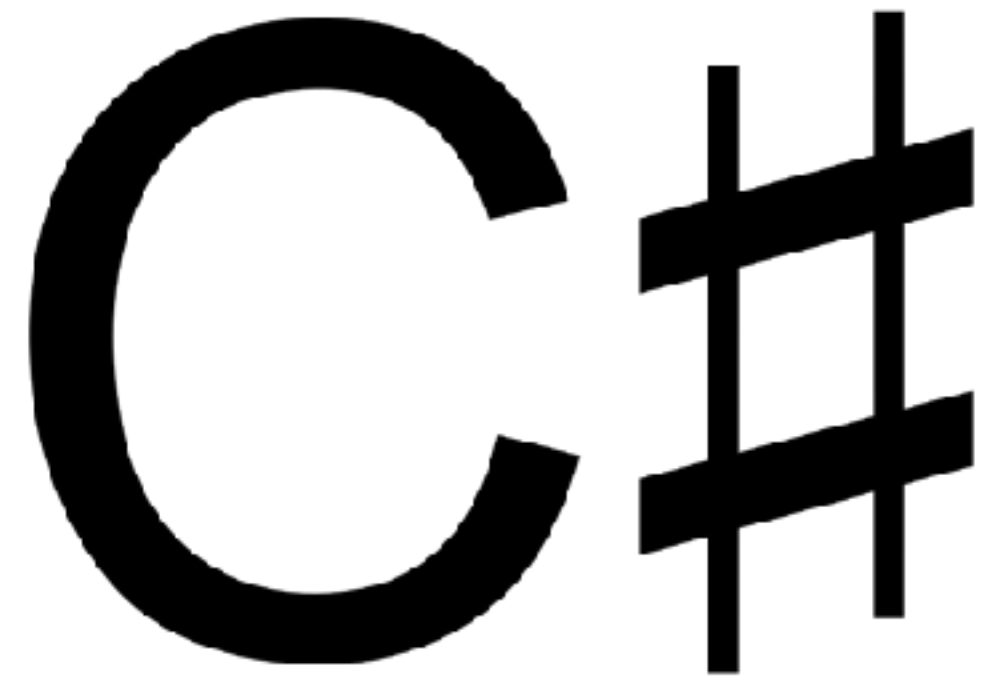
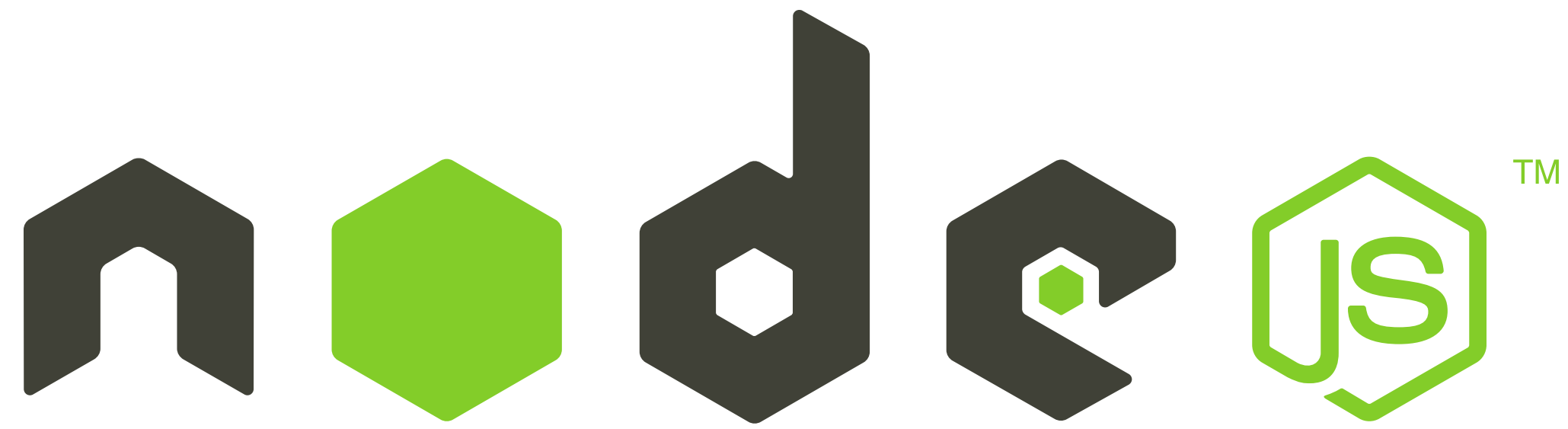
NO OVERPROVISION

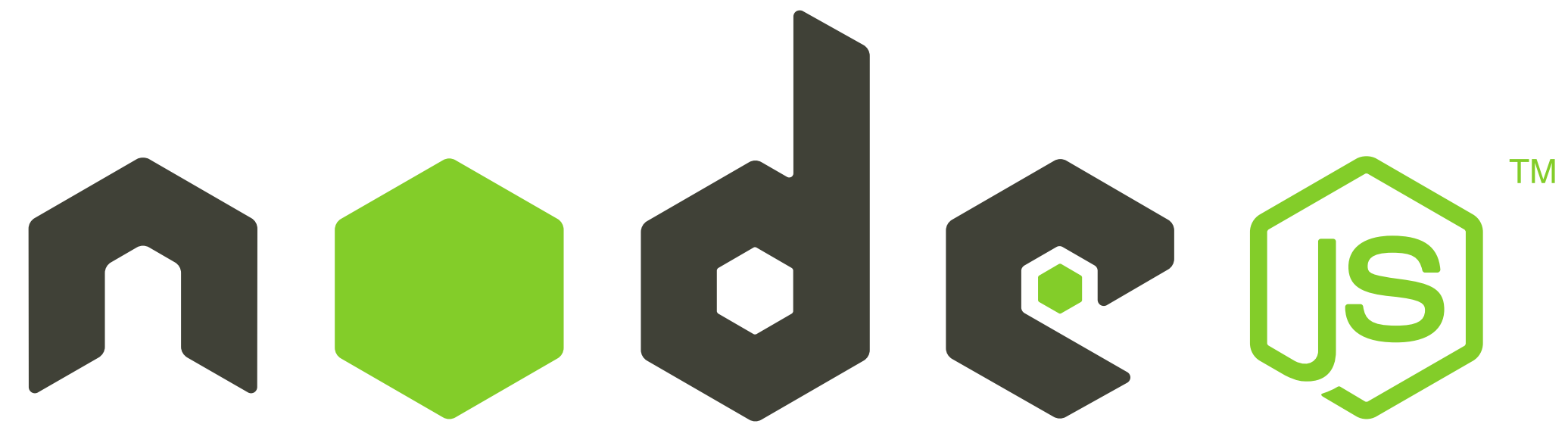
NO UNDERPROVISION



CHARGE BY EXECUTION





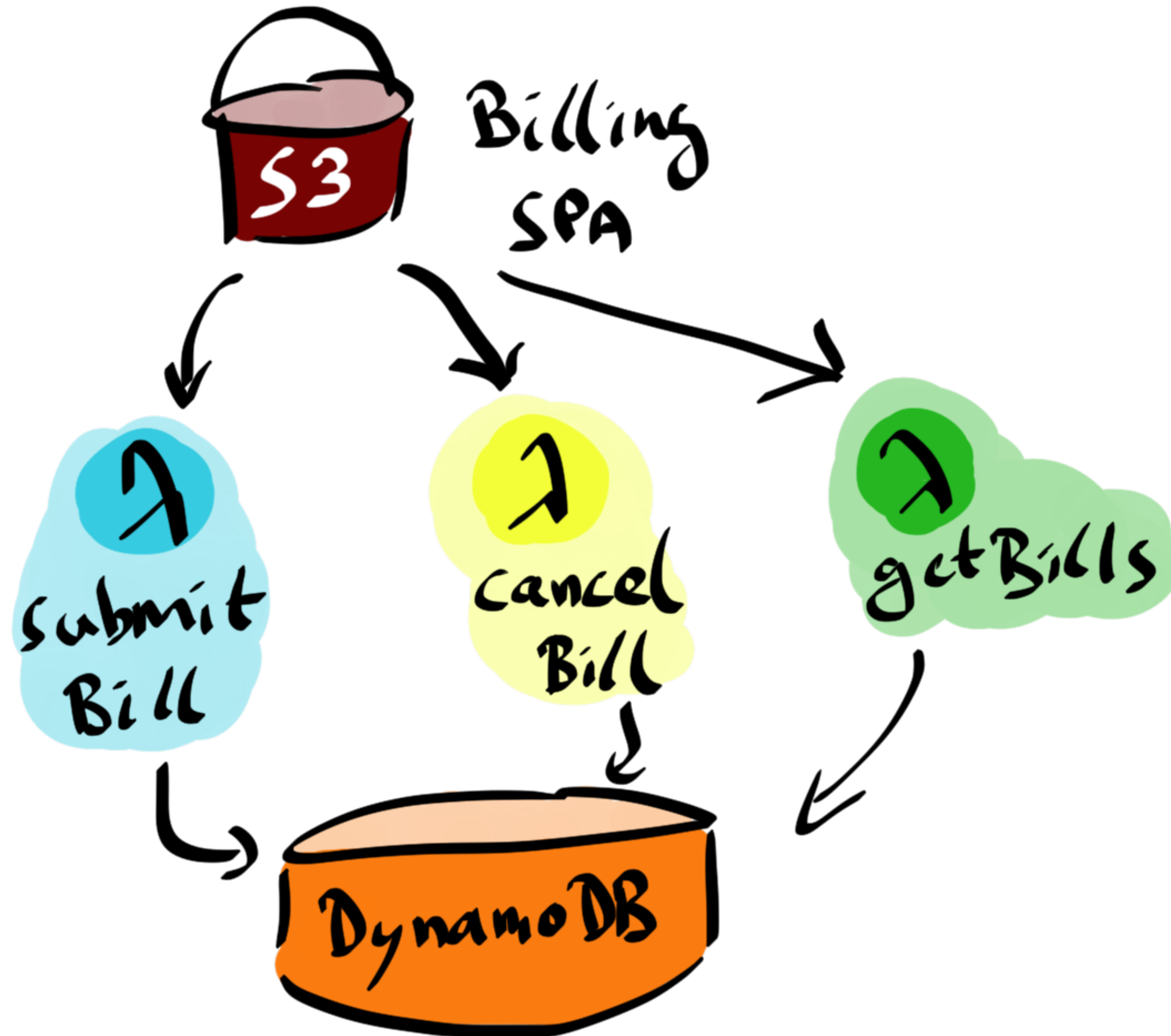


A close-up photograph of a spider web, likely a orb-weaver web, with its intricate spiral and radial patterns clearly visible. The web is set against a blurred background of green foliage and brown branches. A semi-transparent dark horizontal banner is overlaid across the center of the image, containing the text "NEW ARCHITECTURES" and "NEW TRAPS" in a bold, white, sans-serif font.

NEW ARCHITECTURES

NEW TRAPS

A SERVERLESS APPLICATION



SERVERLESS WEBSITE

SERVERLESS COMPUTE

SERVERLESS DATA

NO API GATEWAY?

NO EVENT DRIVEN

EXTRAVAGANZA?

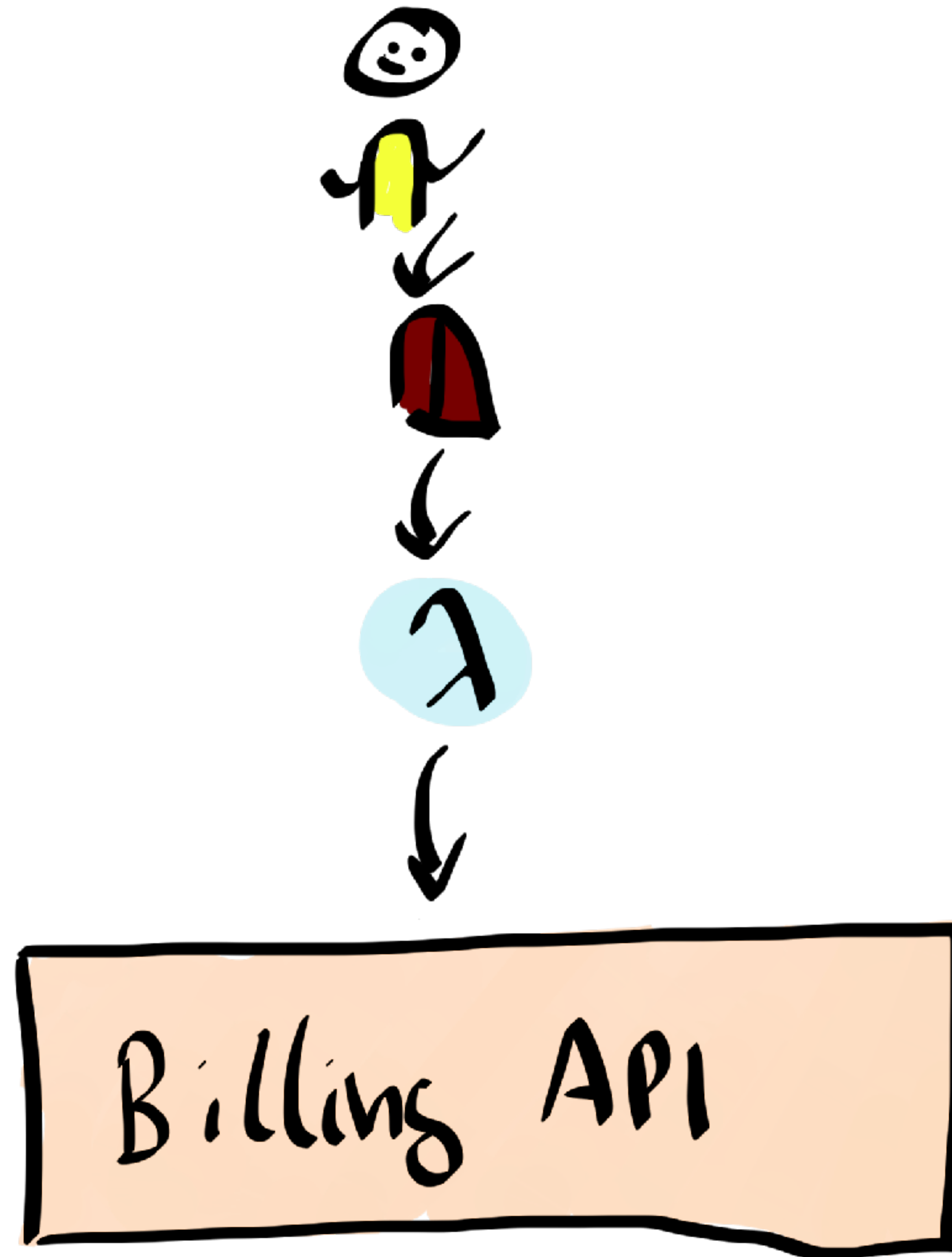


SCALE(LAMBDA) >>> REST

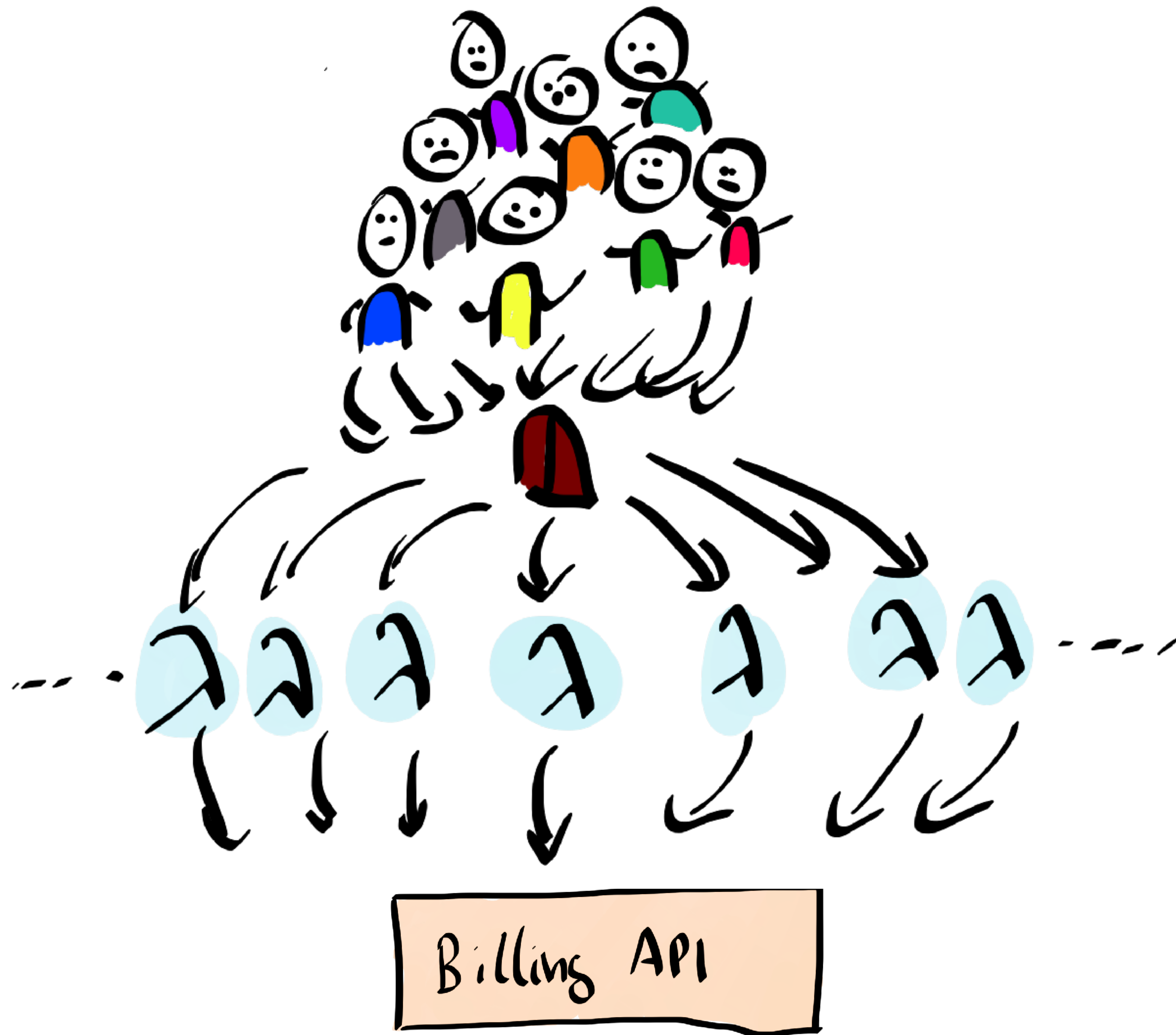
LAMBDA SCALES BY REQUEST

**...YOUR BACKEND
PROBABLY DOES NOT**

**“Billings are
submitted using
our legacy API”**



**...AND AT THE END OF A
MONTH...**



**Please stop
this lambda
non-sense**



But my CV!

ALWAYS USE
CONCURRENCY LIMITS

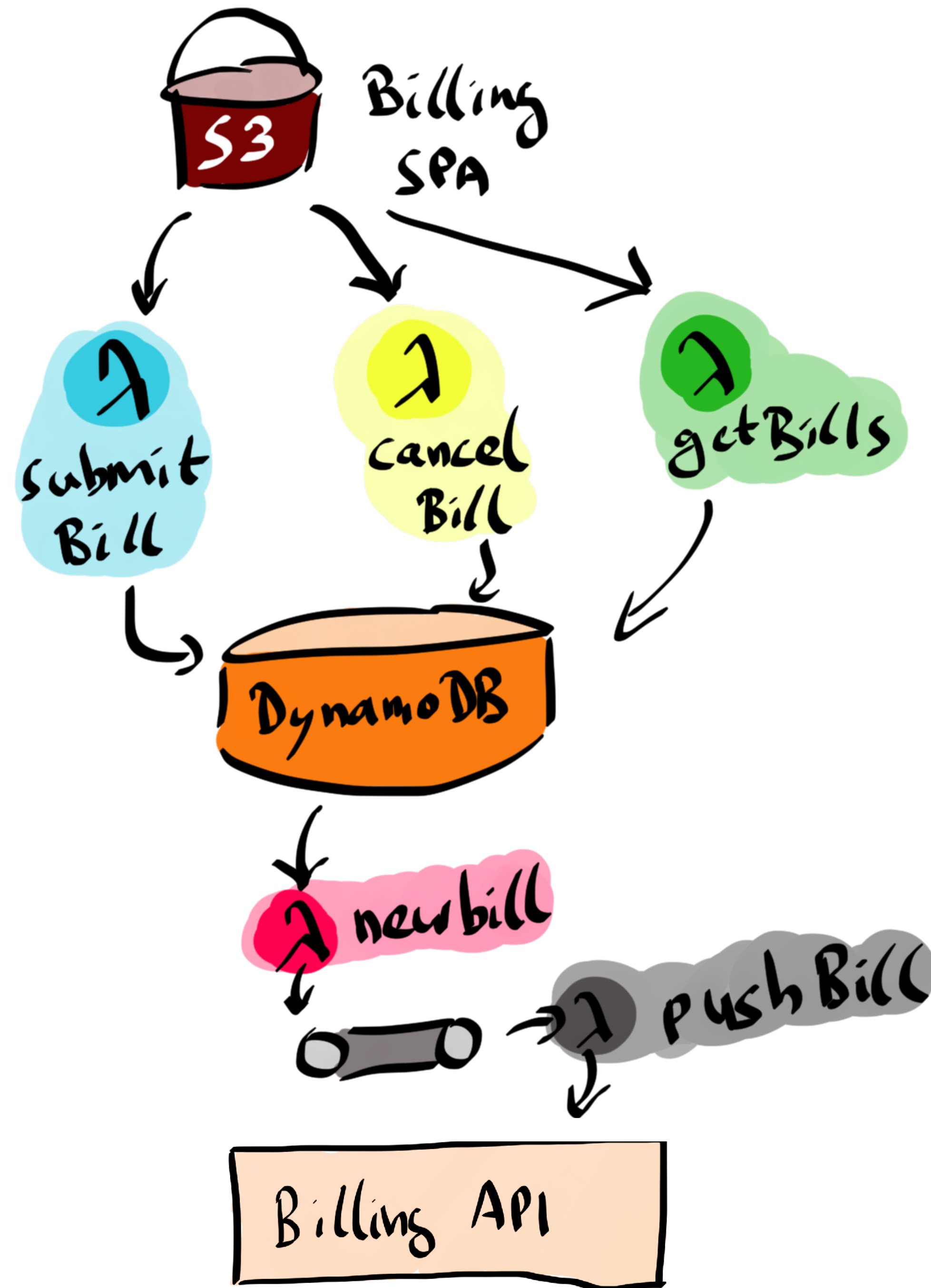
```
aws lambda put-function-concurrency  
--function-name <NAME>  
--reserved-concurrent-executions <N>
```

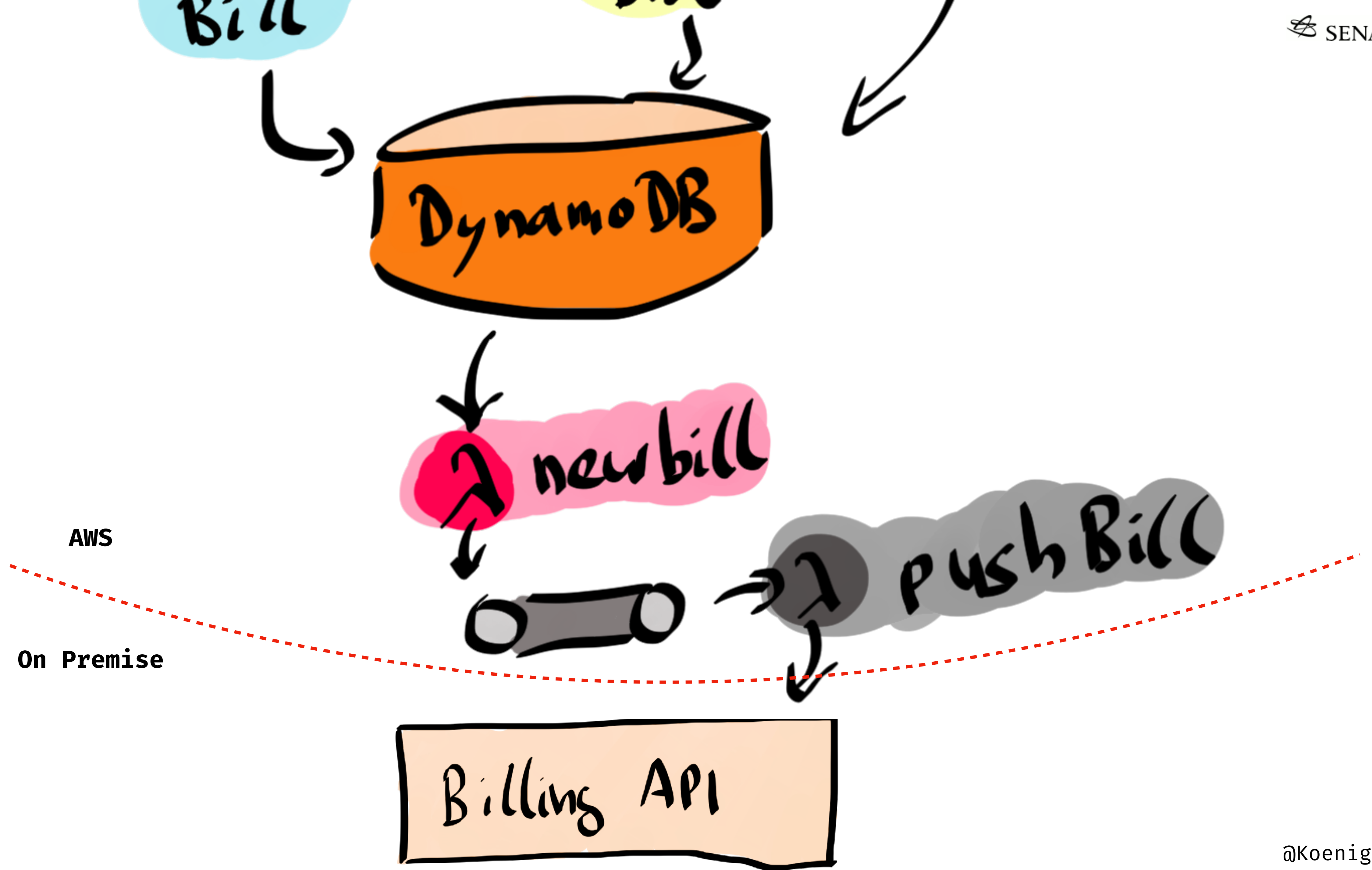


```
aws lambda put-function-concurrency  
--function-name <NAME>  
--reserved-concurrent-executions <N>
```

BOTTLENECKS TEND TO
SPREAD

PREFER ASYNCHRONOUS INTEGRATION





AWS

On Premise



ATTACK OF SELF DENIAL

CONCURRENCY IS LIMITED

KNOW YOUR PER-ACCOUNT LIMITS

AWS Lambda Resource Limits per Invocation

Resource	Limits
Memory allocation range	Minimum = 128 MB / Maximum = 3008 MB (with 64 MB increments). If the maximum memory use is exceeded, function invocation will be terminated.
Ephemeral disk capacity ("/tmp" space)	512 MB
Number of file descriptors	1,024
Number of processes and threads (combined total)	1,024
Maximum execution duration per request	300 seconds
Invoke request body payload size (RequestResponse/synchronous invocation)	6 MB
Invoke request body payload size (Event/asynchronous invocation)	128 K

AWS Lambda Account Limits Per Region

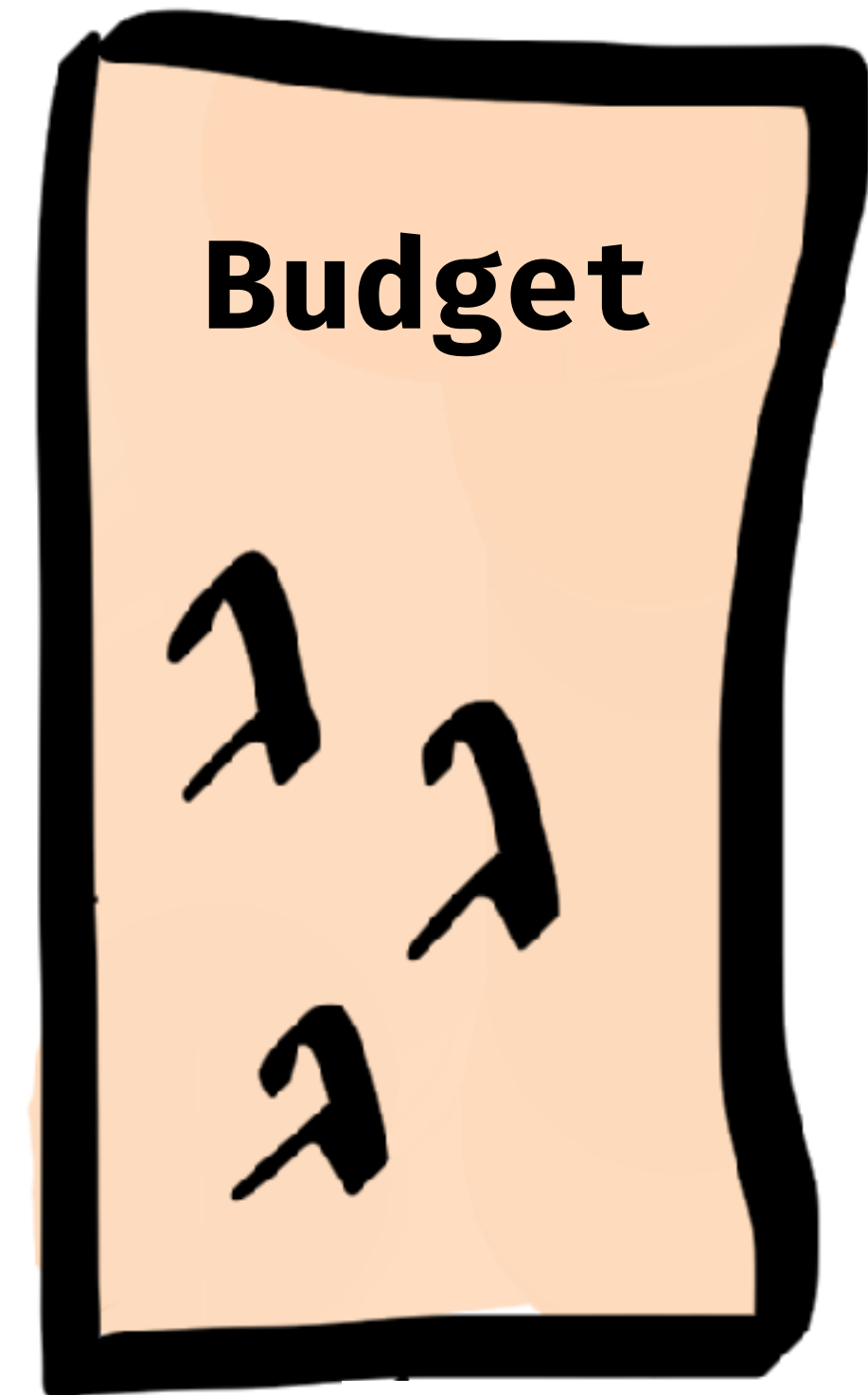
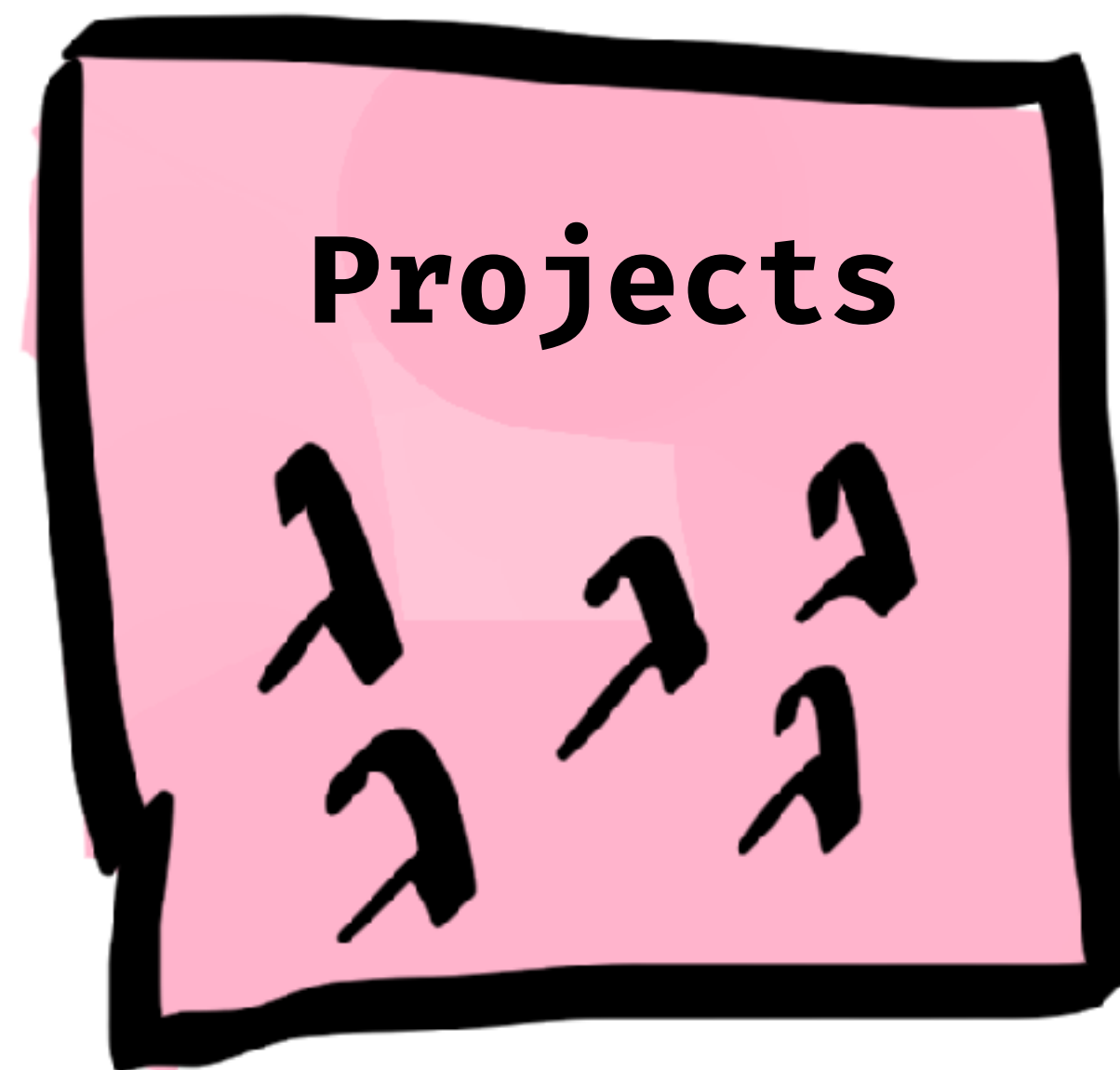
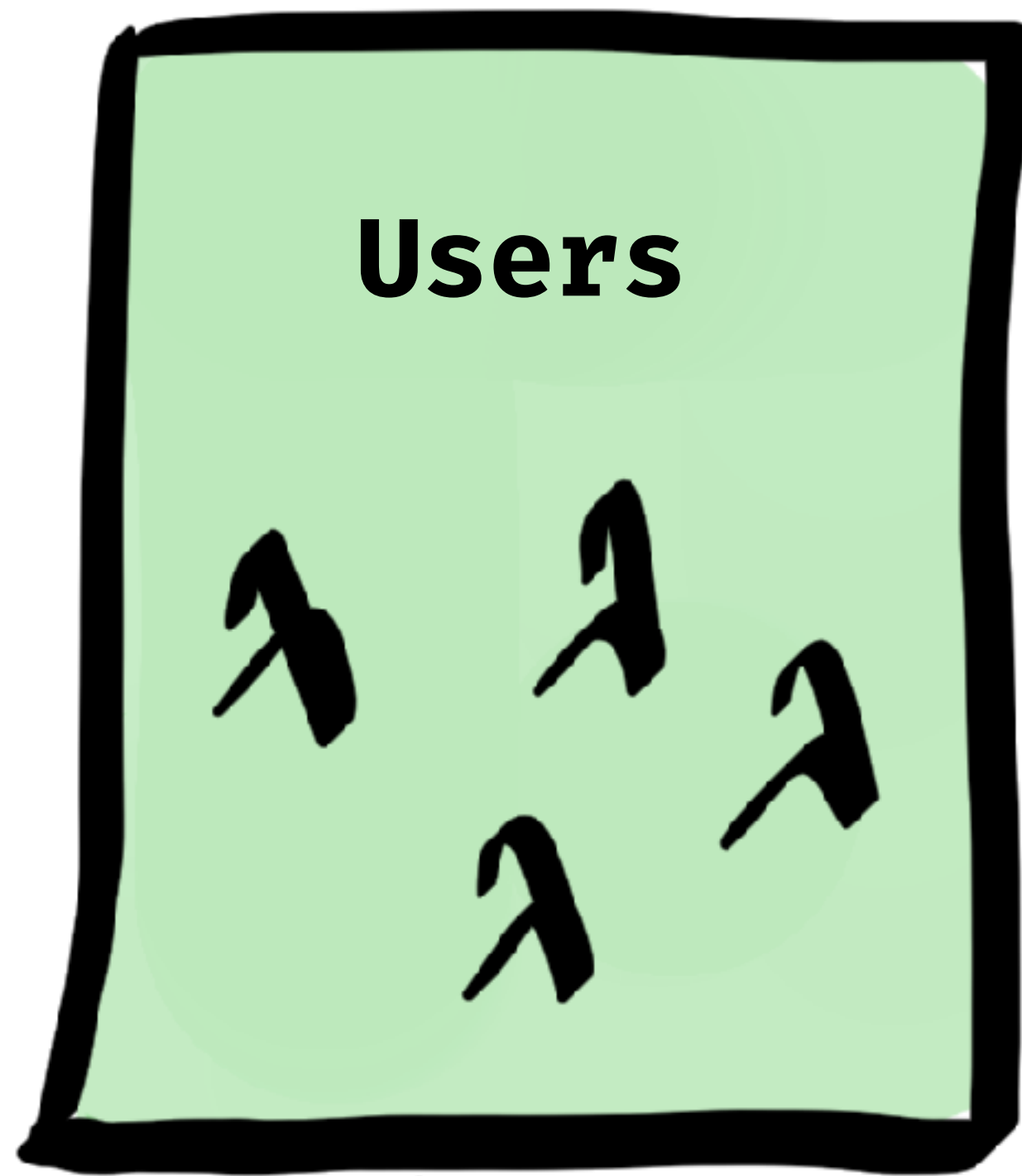
Resource	Default Limit
Concurrent executions (see Managing Concurrency)	1000

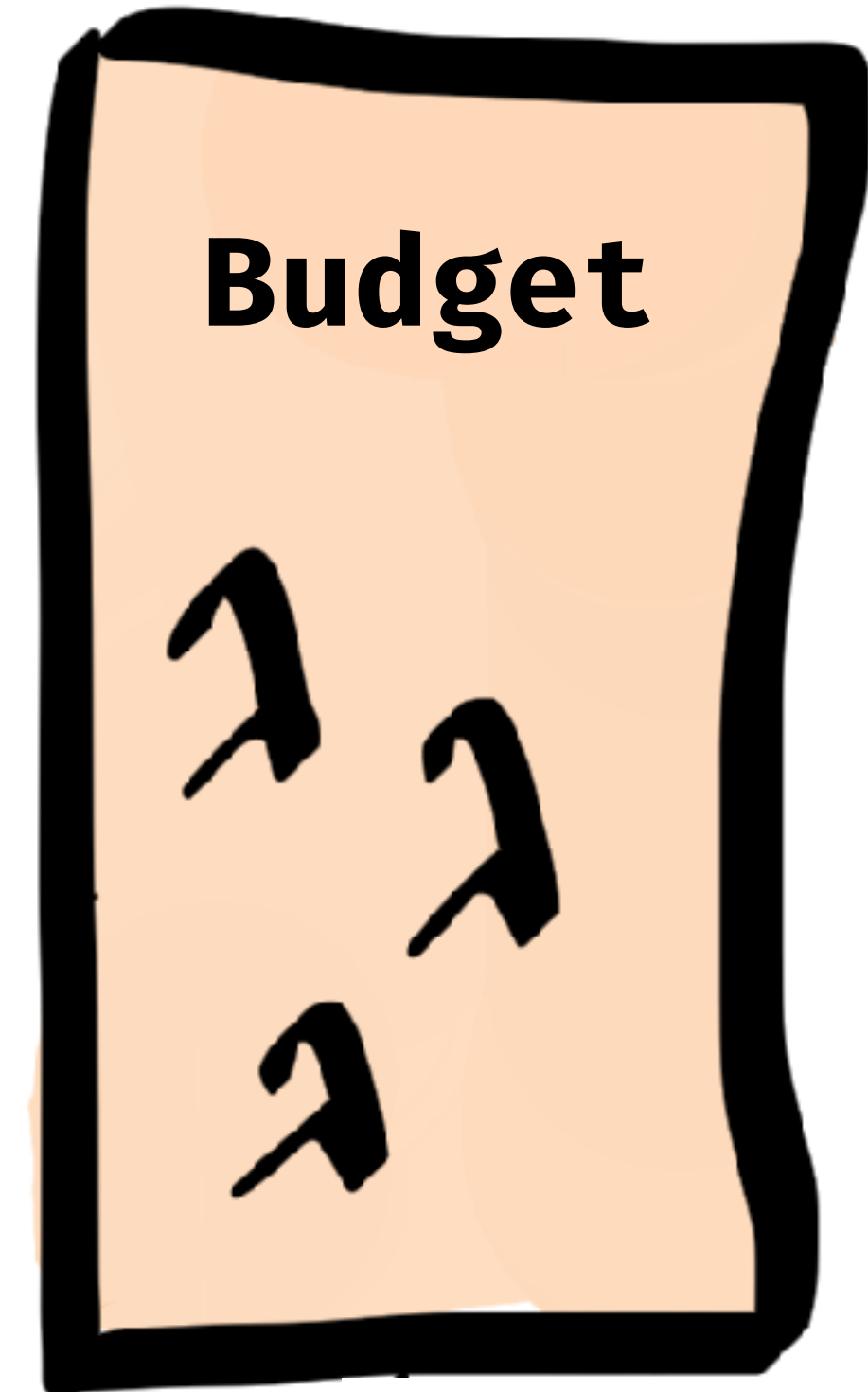
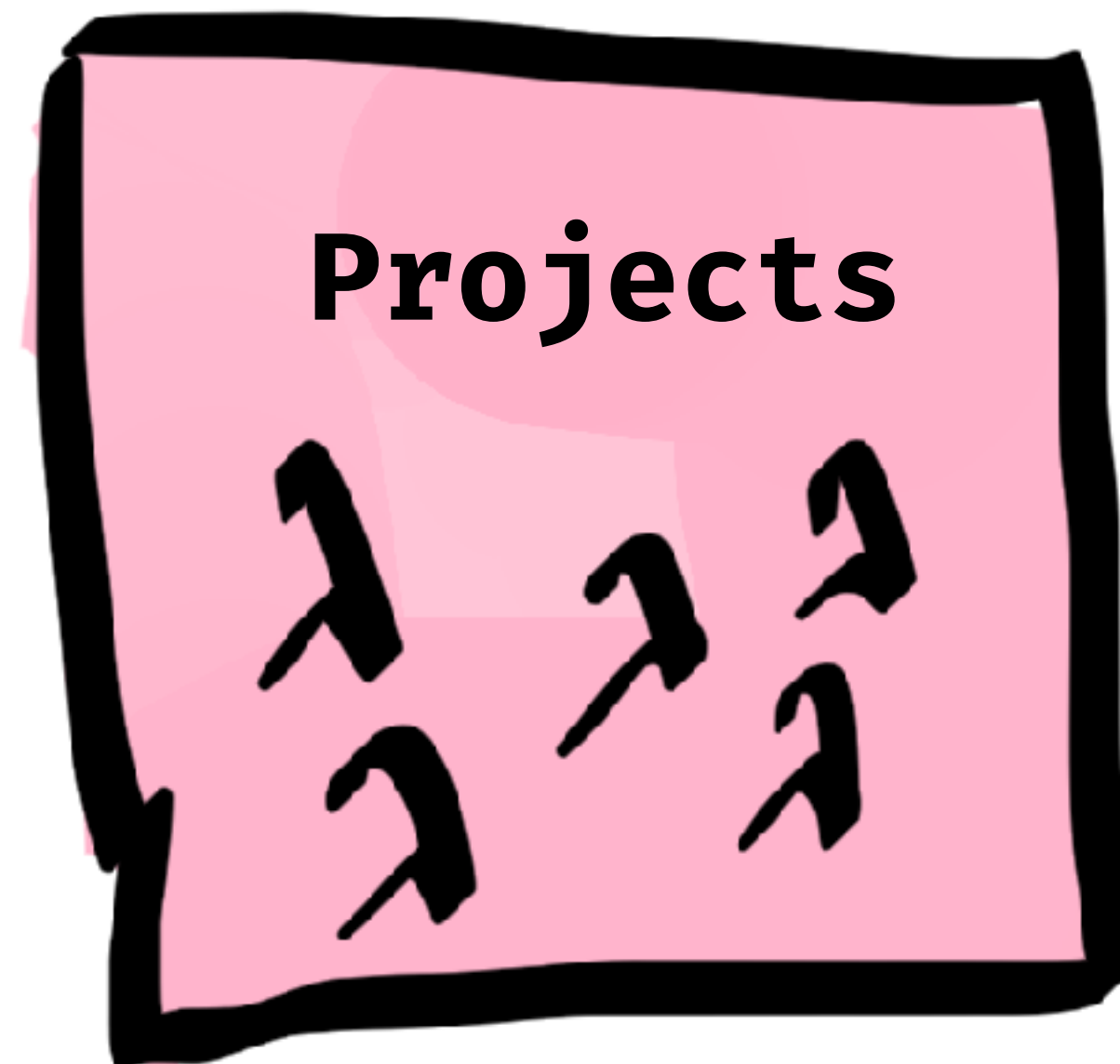
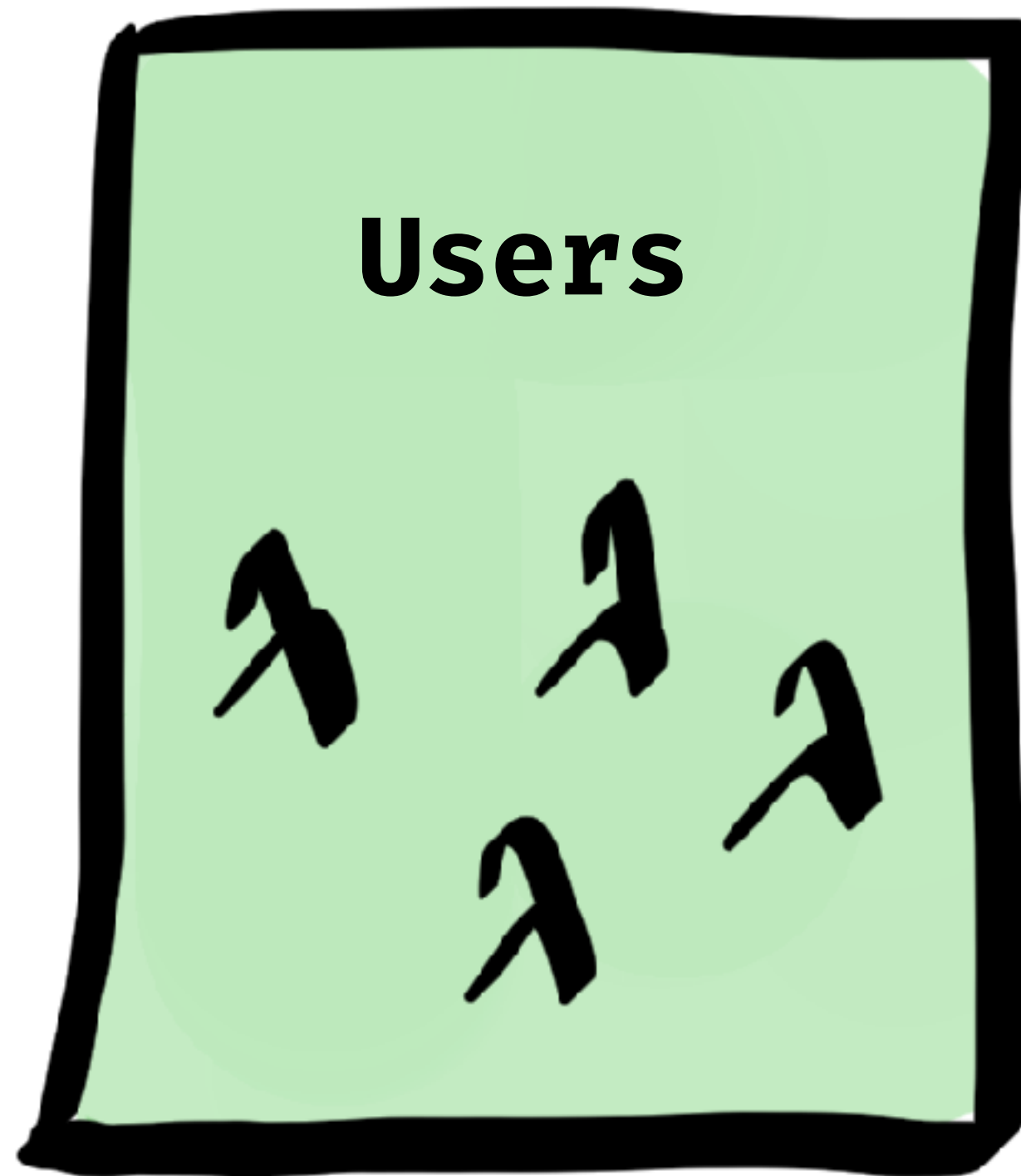
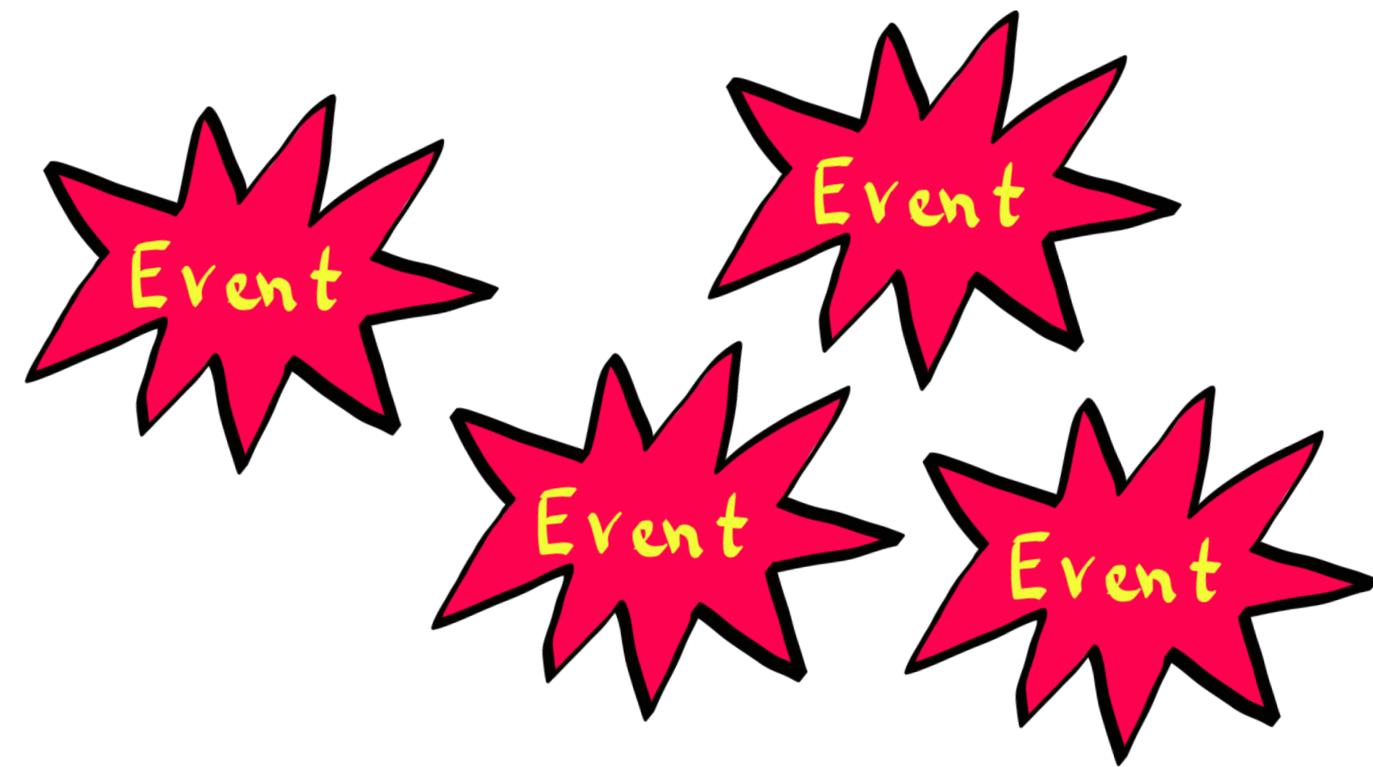
AWS Lambda Resource Limits per Invocation

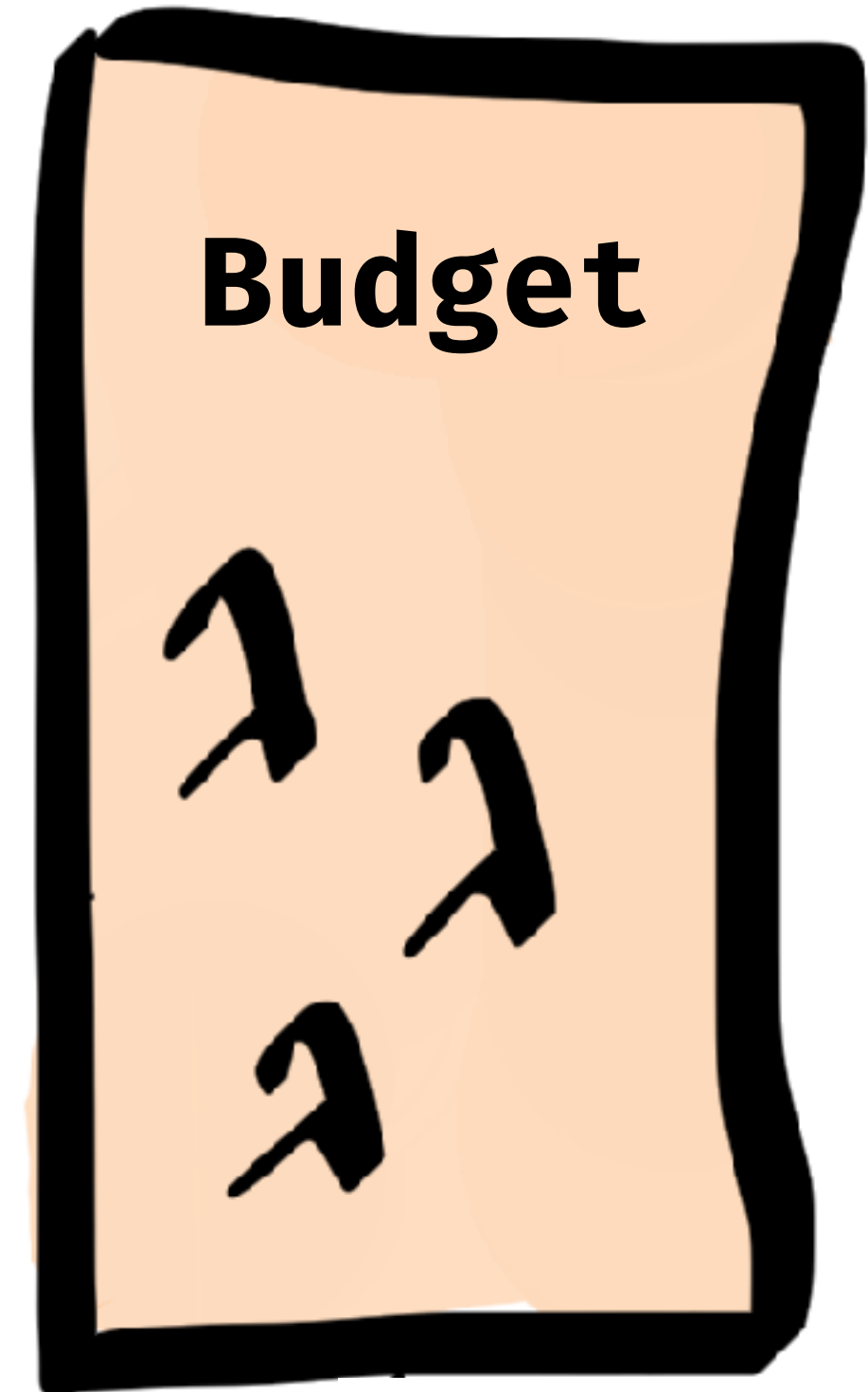
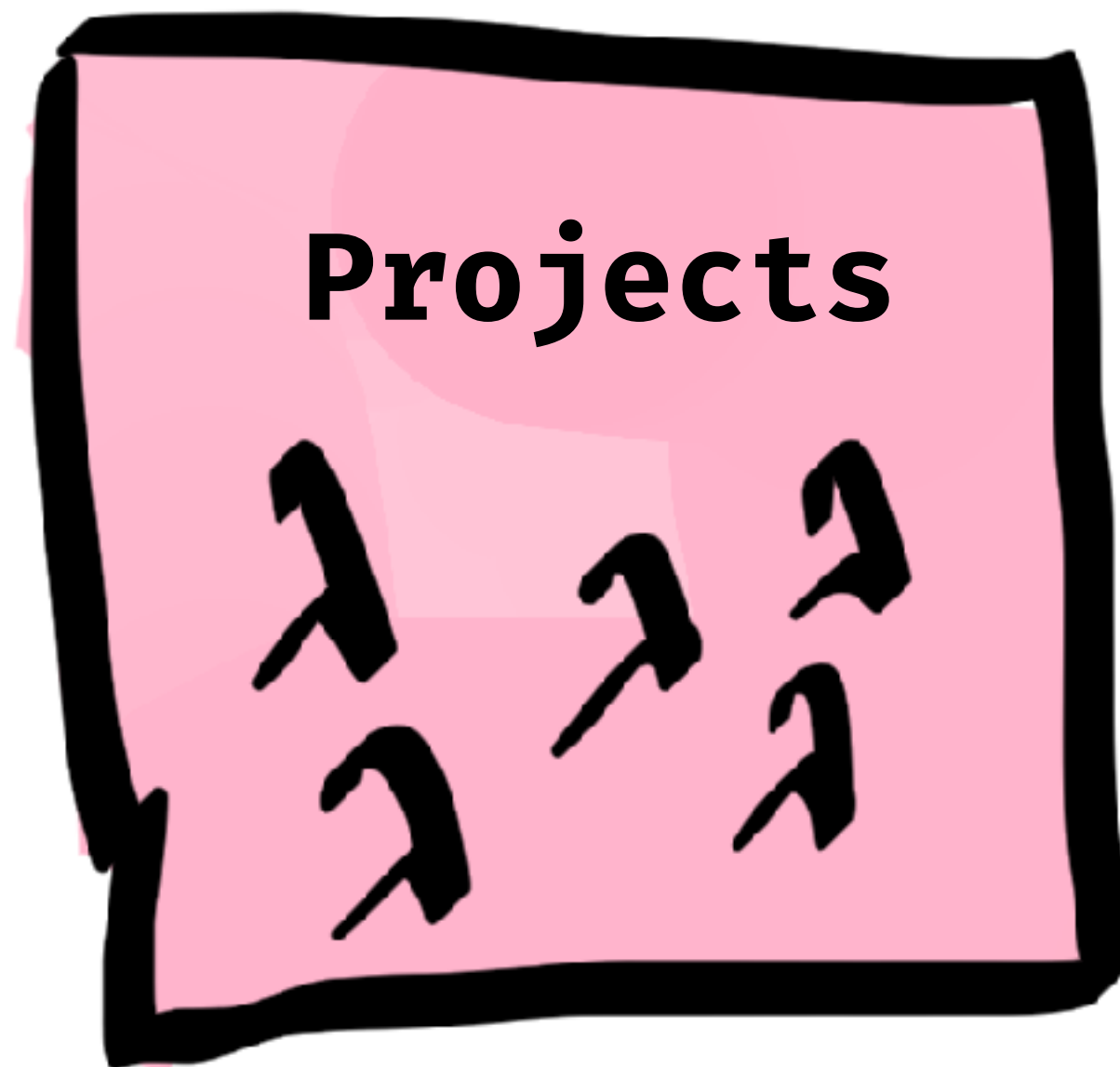
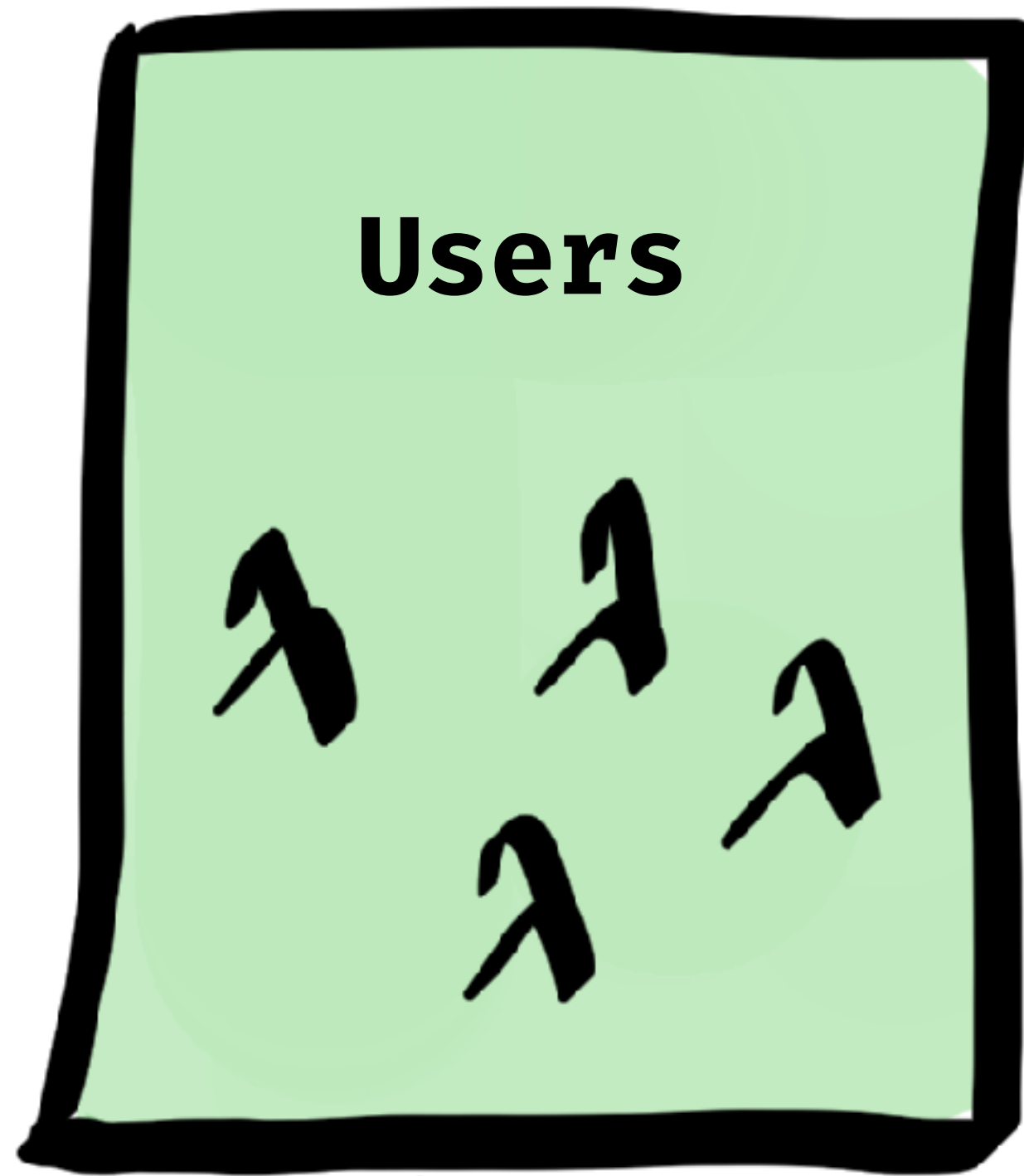
Resource	Limits
Memory allocation range	Minimum = 128 MB / Maximum = 3008 MB (with 64 MB increments). If the maximum memory use is exceeded, function invocation will be terminated.
Ephemeral disk capacity ("/tmp" space)	512 MB
Number of file descriptors	1,024
Number of processes and threads (combined total)	1,024
Maximum execution duration per request	300 seconds
Invoke request body payload size (RequestResponse/synchronous invocation)	6 MB
Invoke request body payload size (Event/asynchronous invocation)	128 K

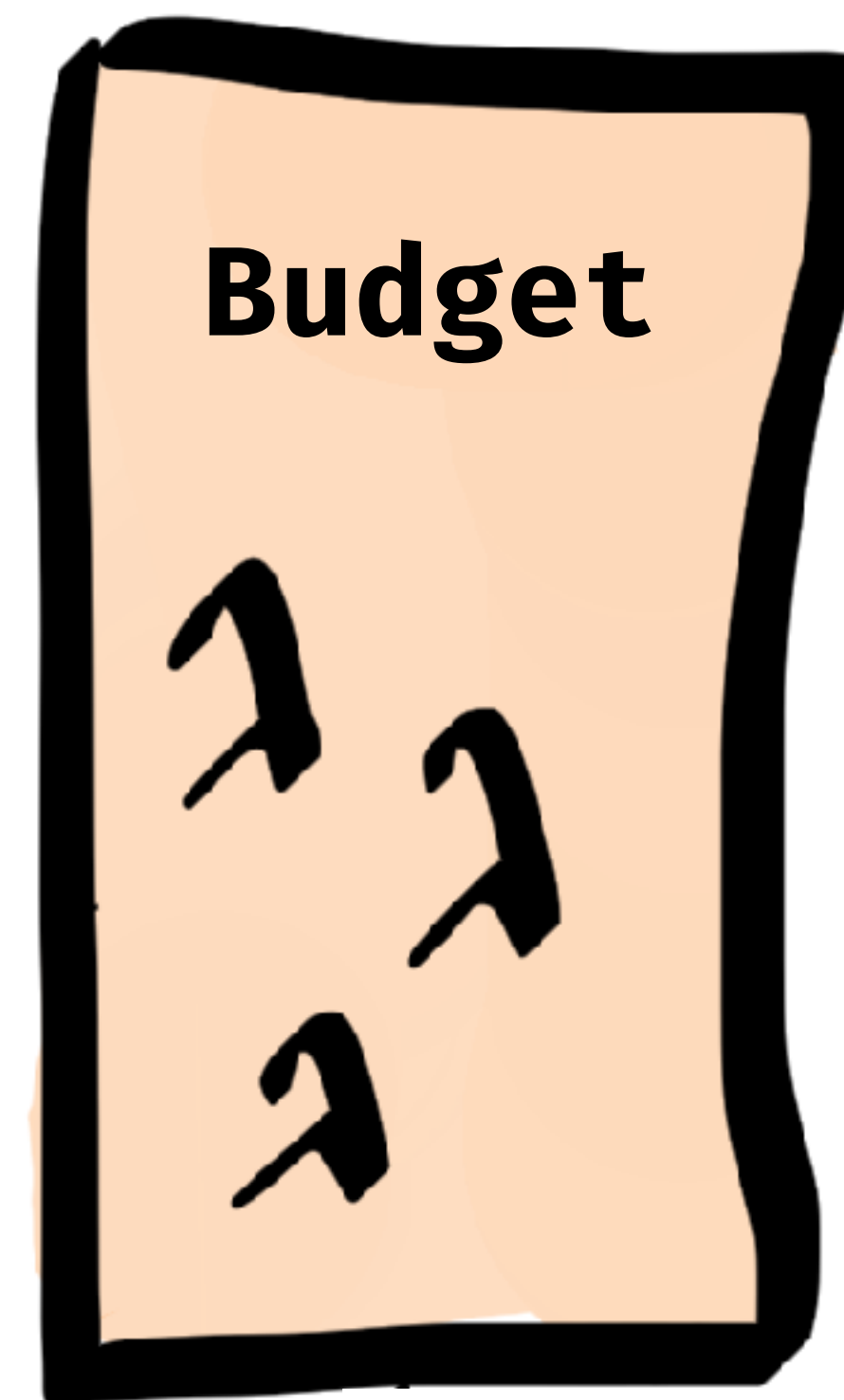
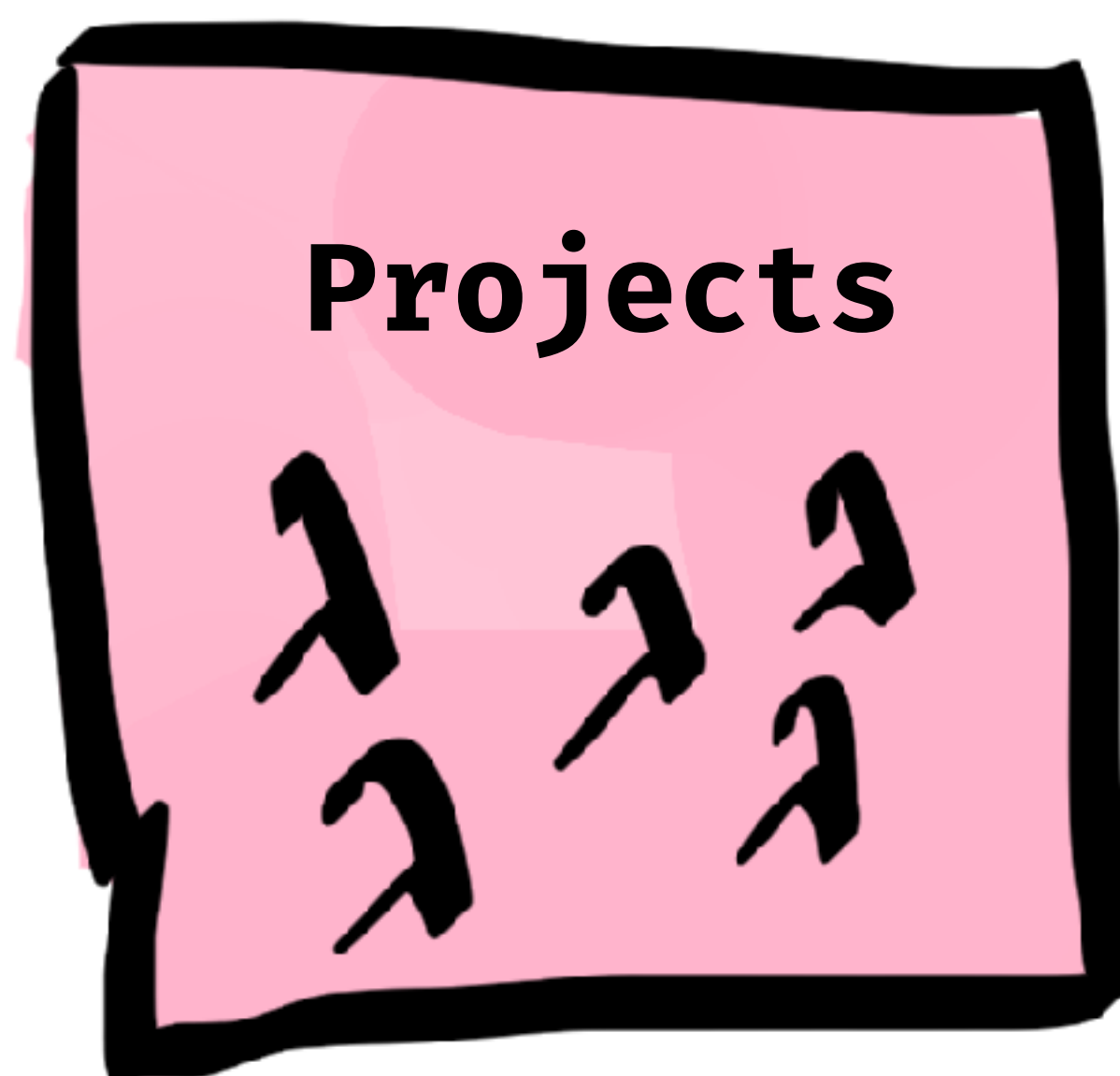
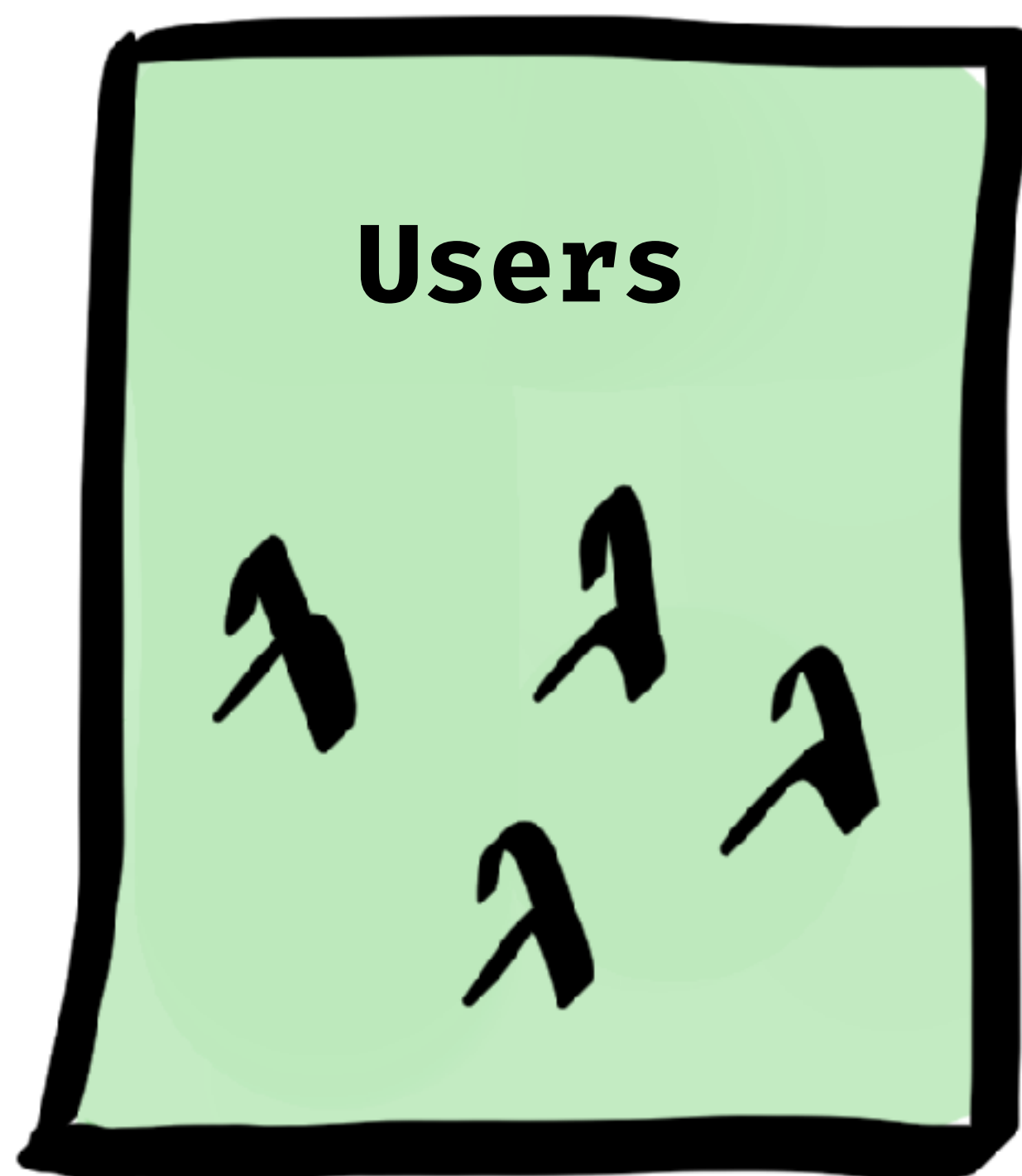
AV	Resource	Default Limit
R	Concurrent executions (see Managing Concurrency)	1000
C		







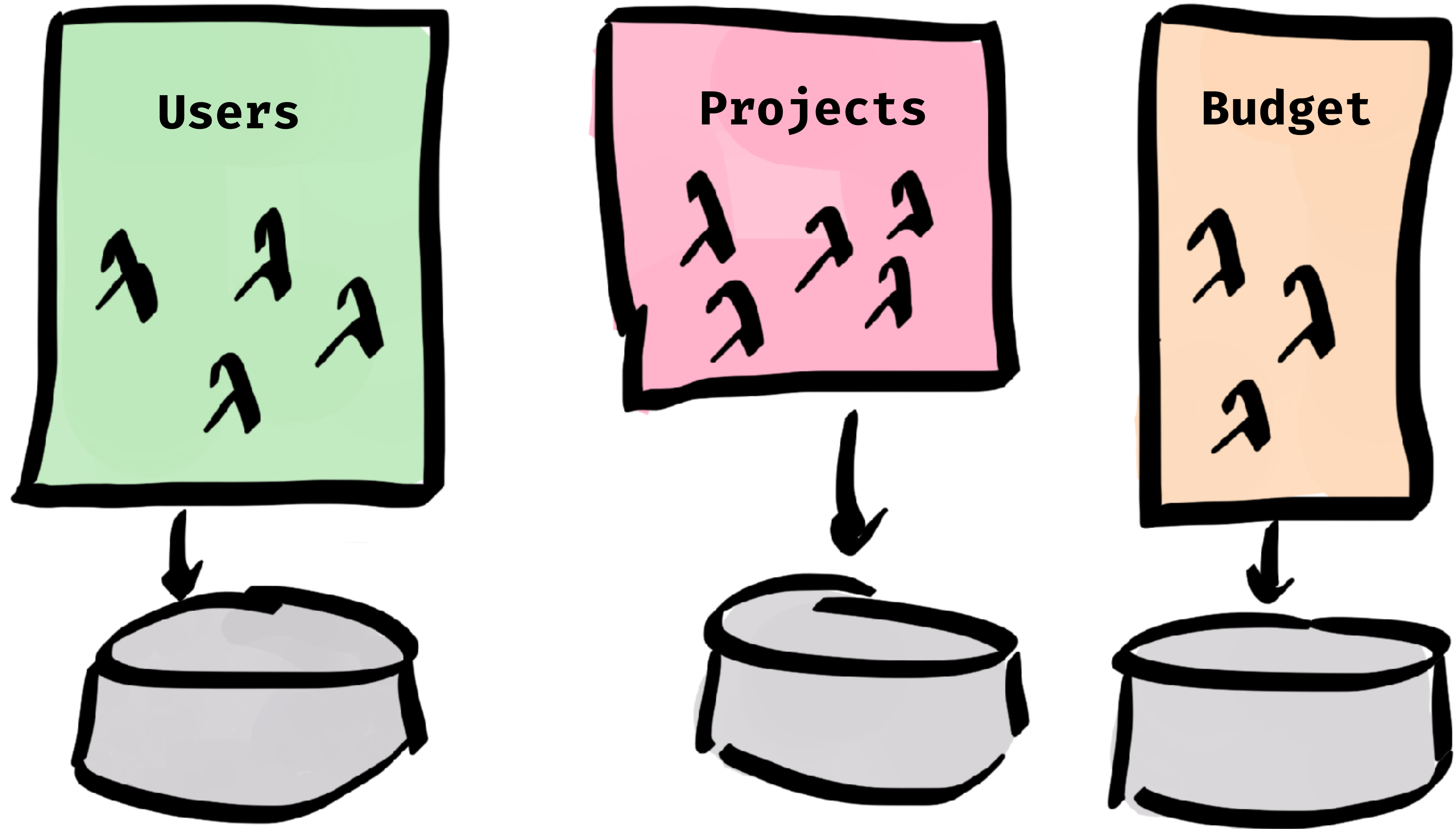




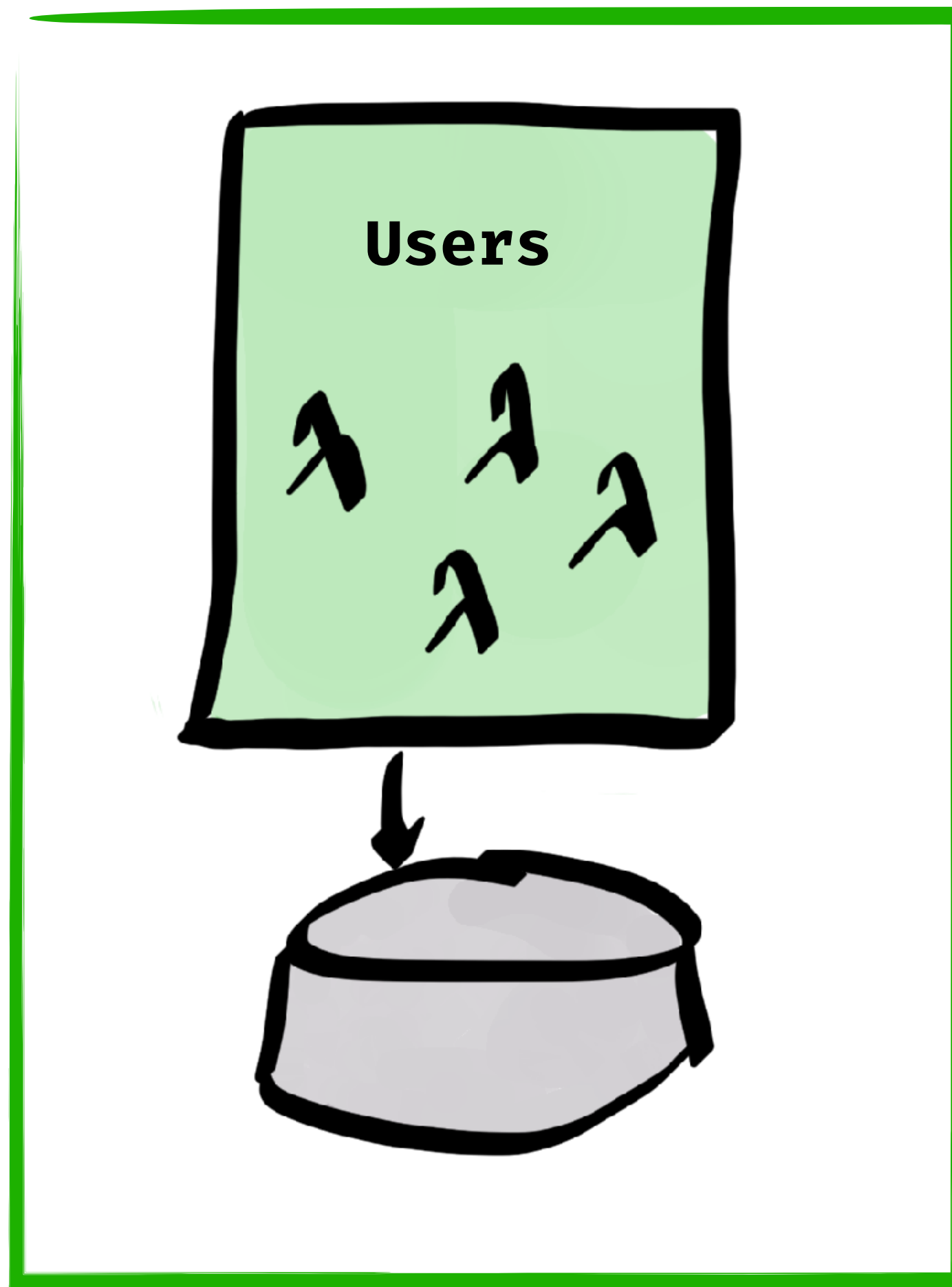
SHARE NOTHING

LAMBDA EDITION

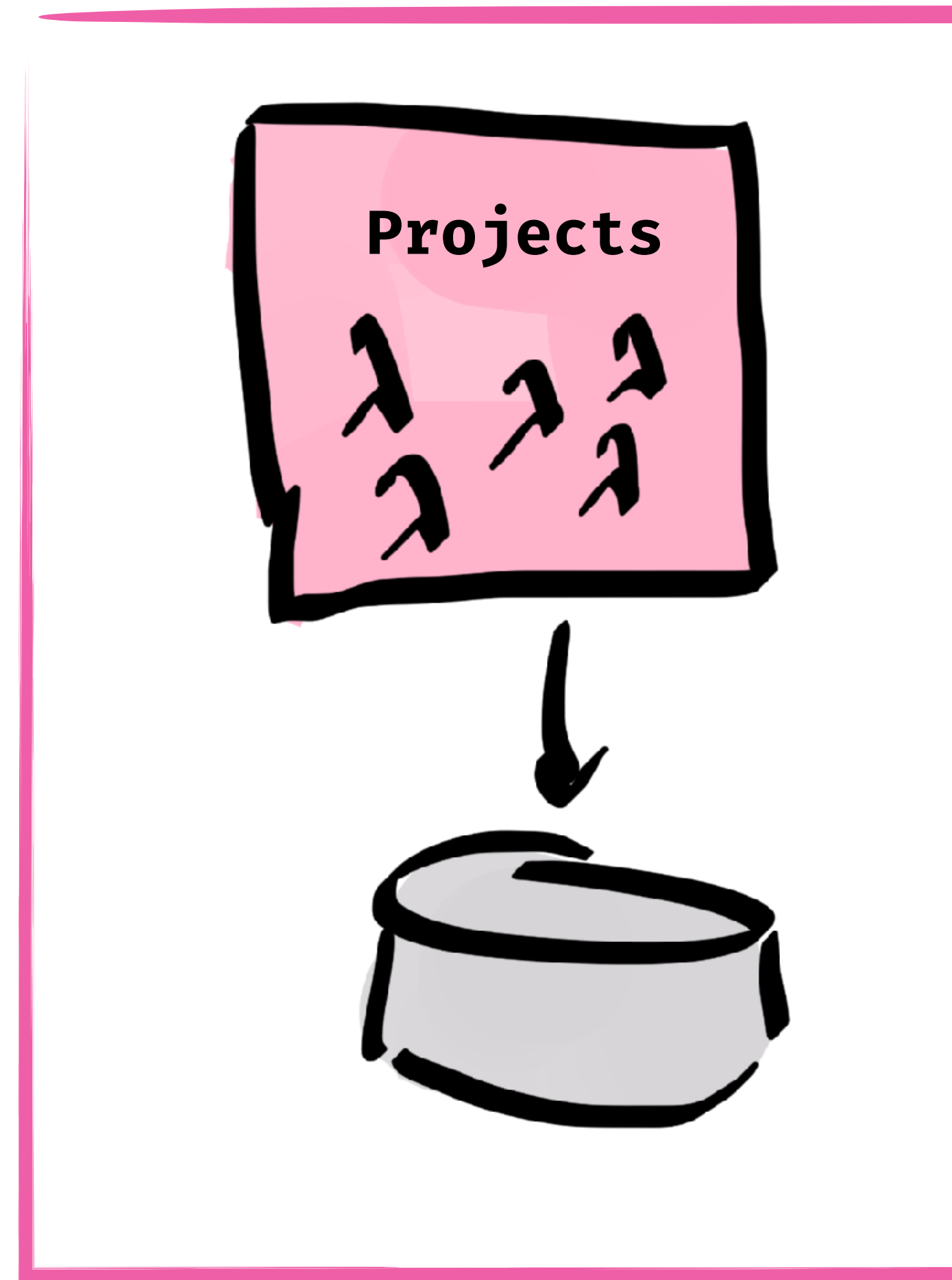
SPLIT PER BOUNDED CONTEXT



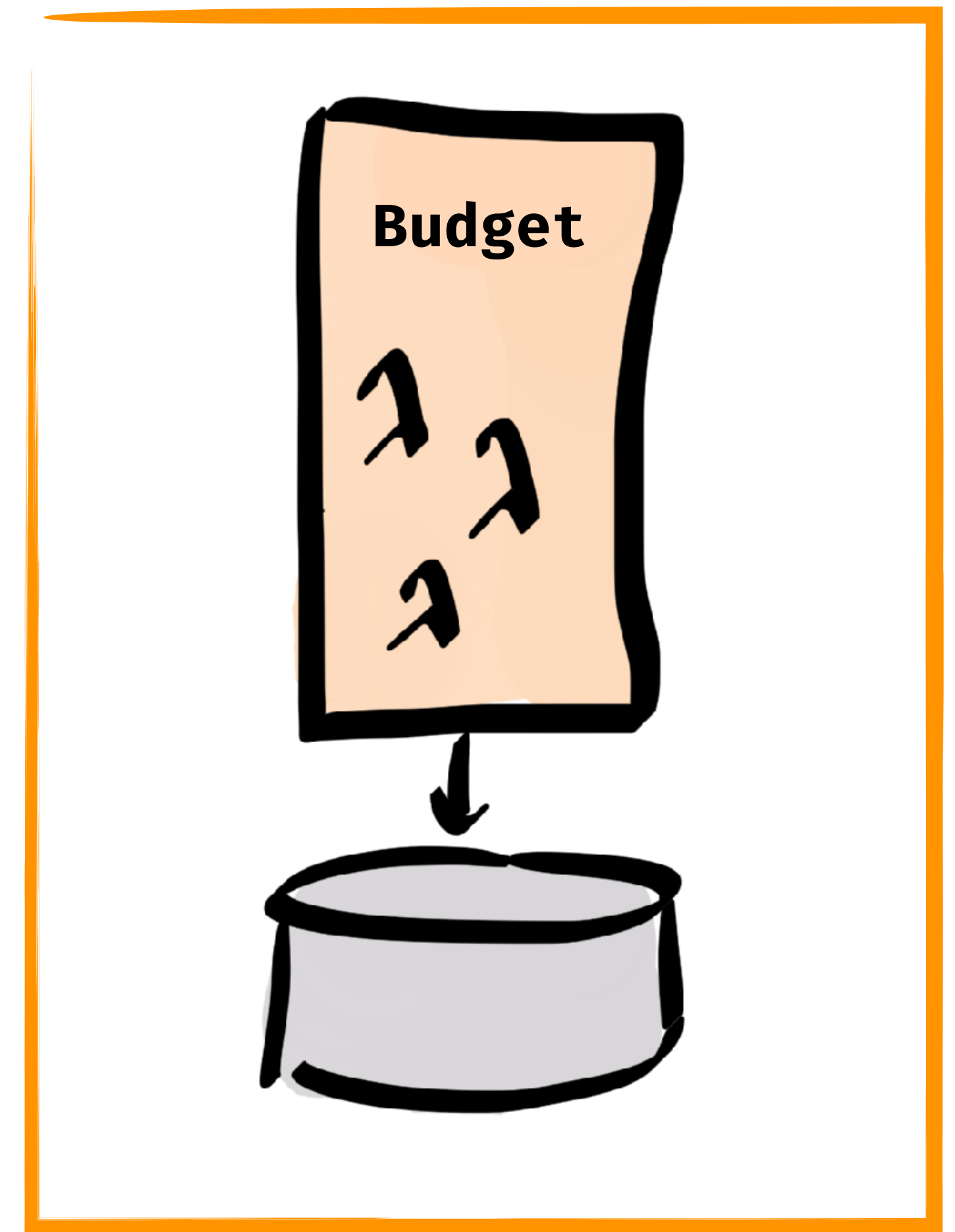
Users Account



Project Account



Budget Account





LAMBDA-TIMTOWTDY

ONE LAMBDA WITH DISPATCH

MULTI LAMBDA – SHARED ZIP

MULTI LAMBDA – SHARED NOTHING

ONE LAMBDA WITH DISPATCH

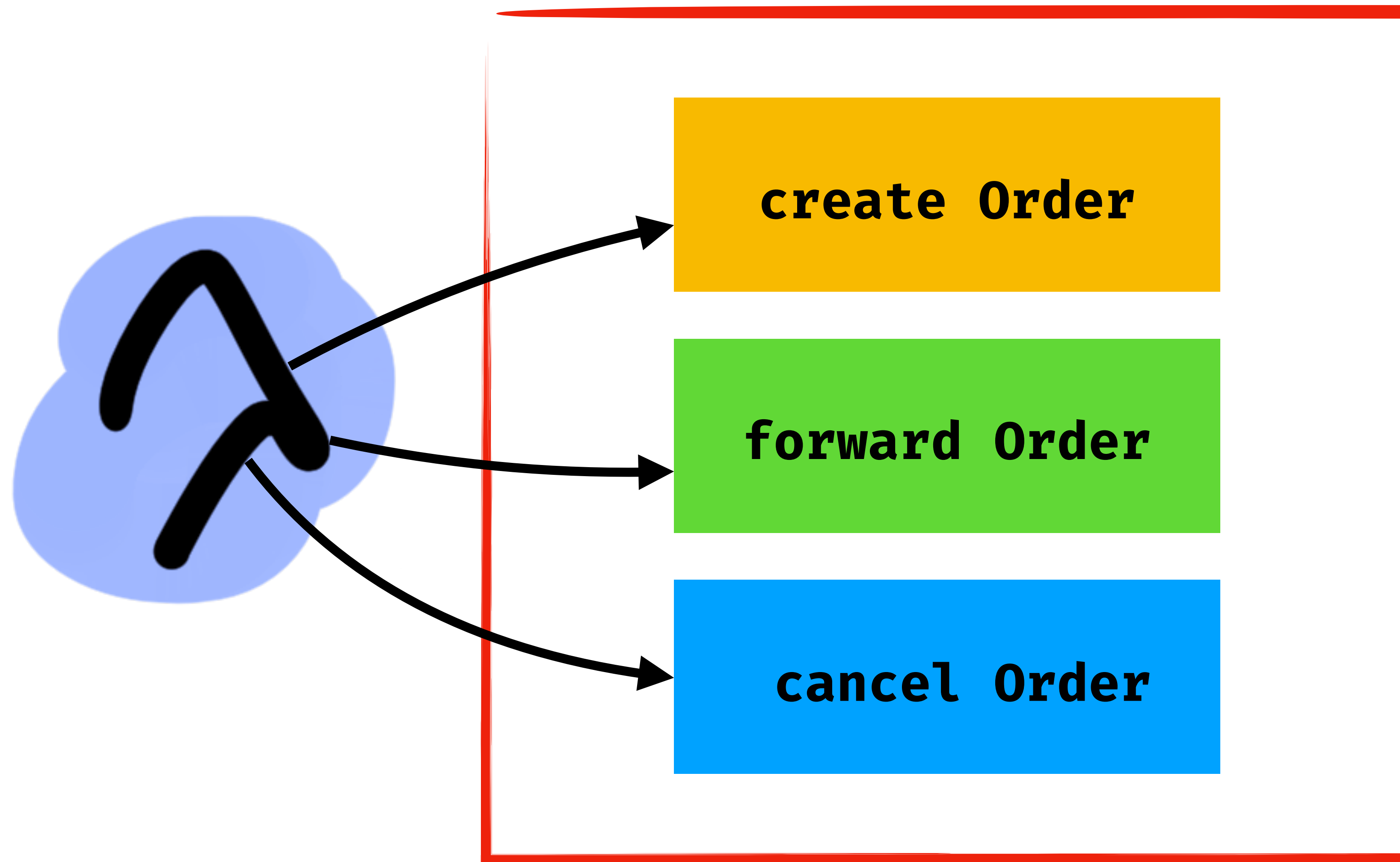
Order ZIP

create Order

forward Order

cancel Order

Order ZIP



EASE OF TRANSITION	✓
FLEXIBILITY	✗
SECURITY	✗
SIZING	✗
DEPLOYMENT	✓
COMPLEXITY	✓

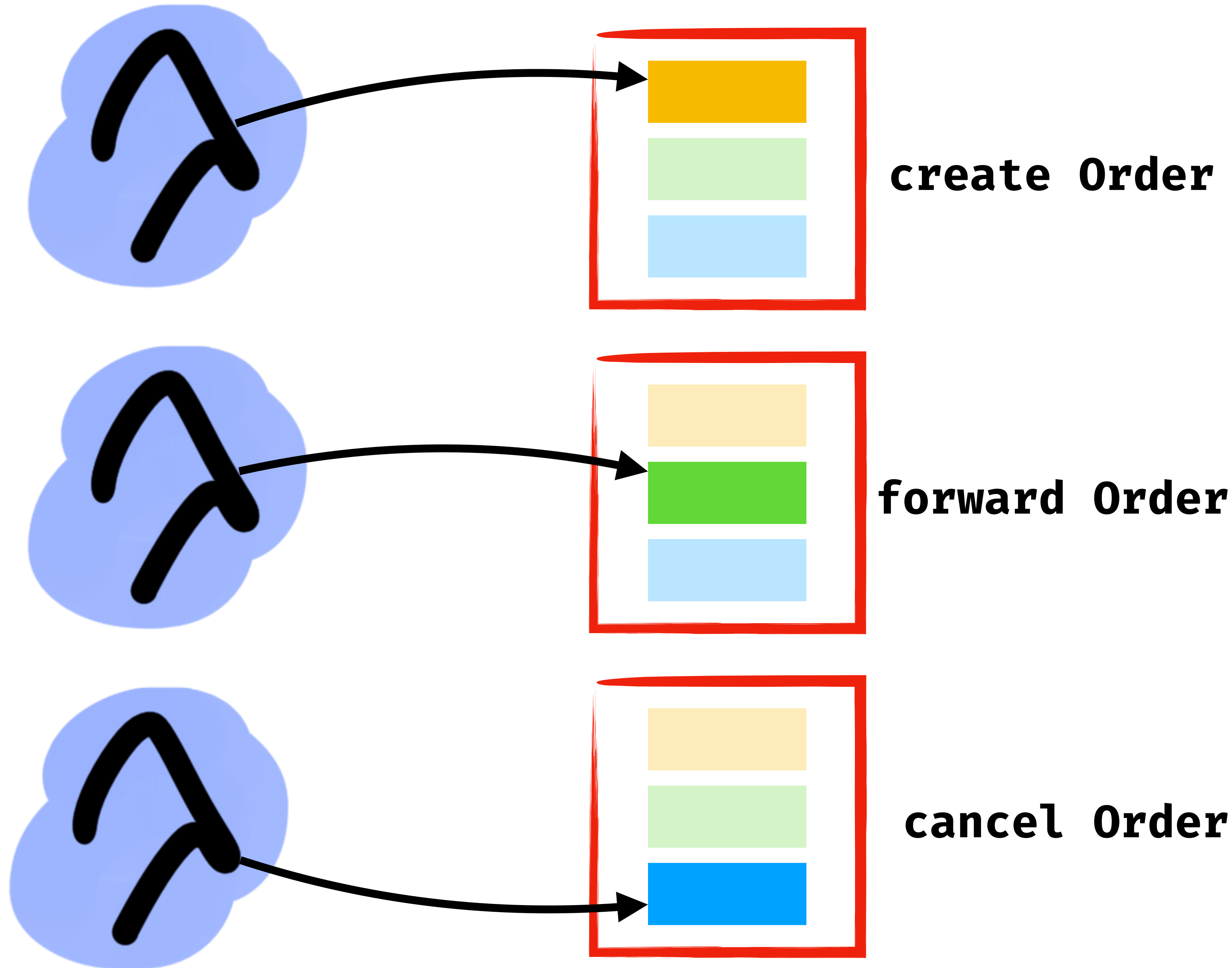
MULTI LAMBDA — SHARED ZIP

Order ZIP

create Order

forward Order

cancel Order



EASE OF TRANSITION	
FLEXIBILITY	
SECURITY	
SIZING	
DEPLOYMENT	
COMPLEXITY	

MULTI LAMBDA

—

SHARED NOTHING

Order ZIP

create Order

forward Order

cancel Order

Create Order ZIP

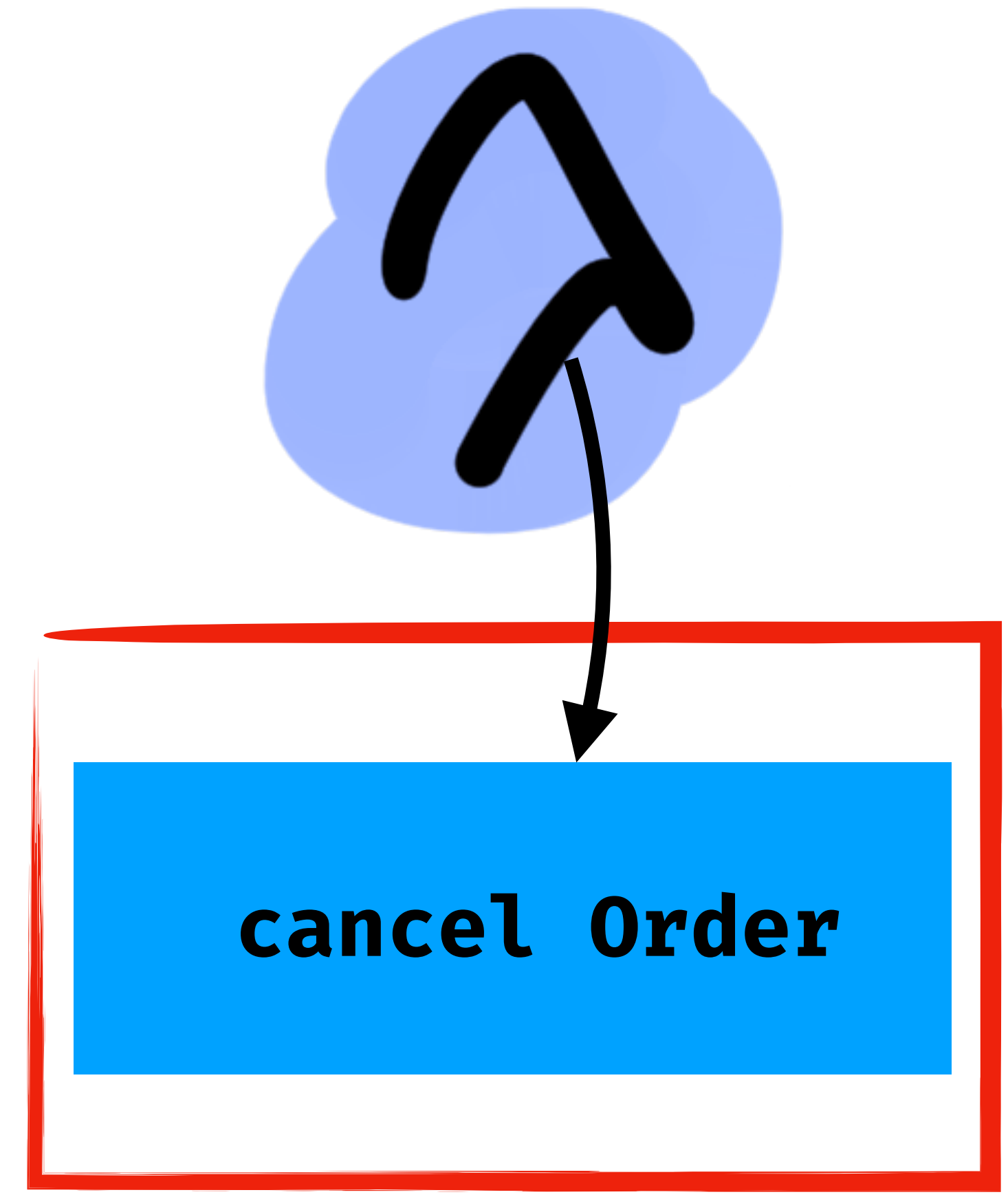
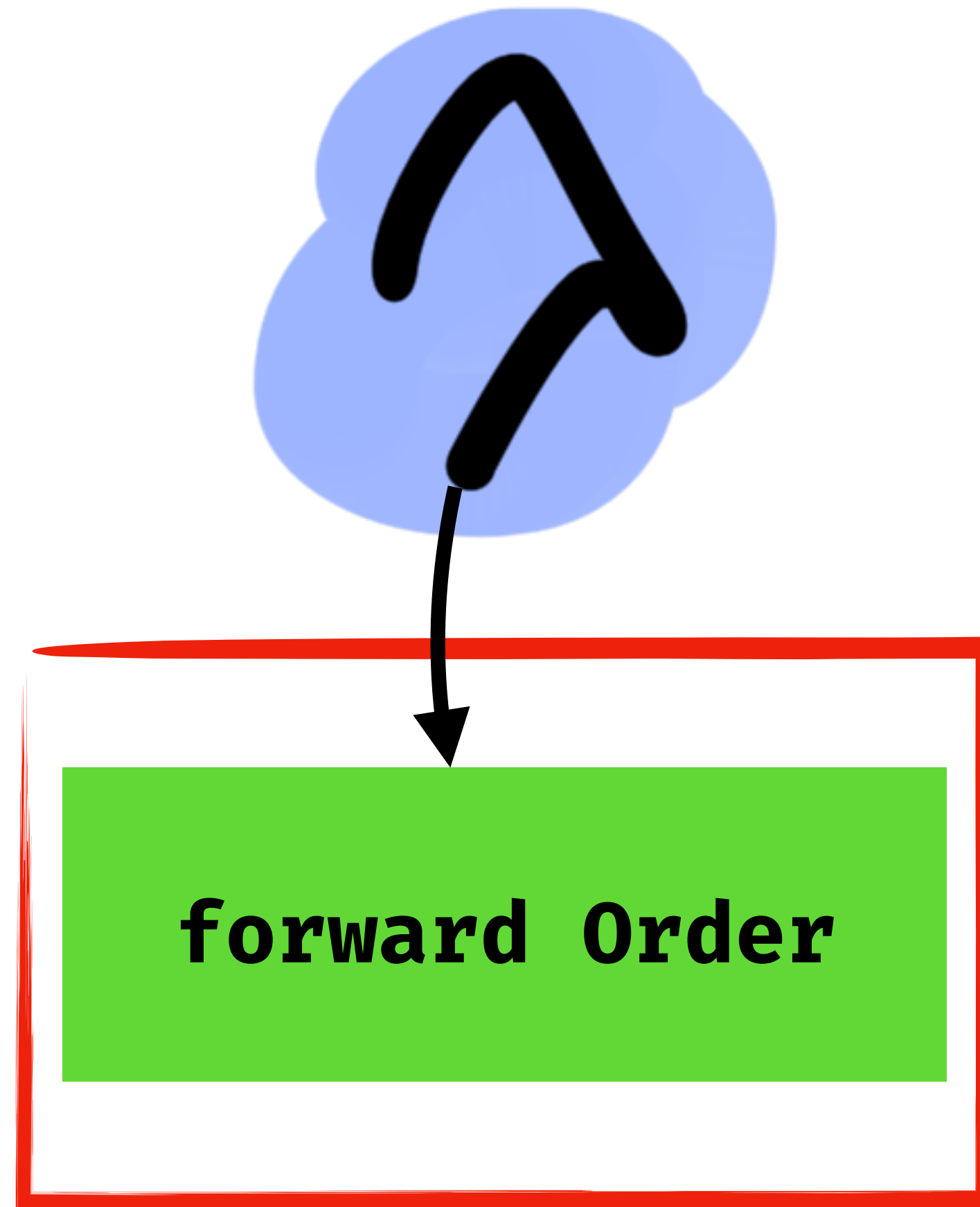
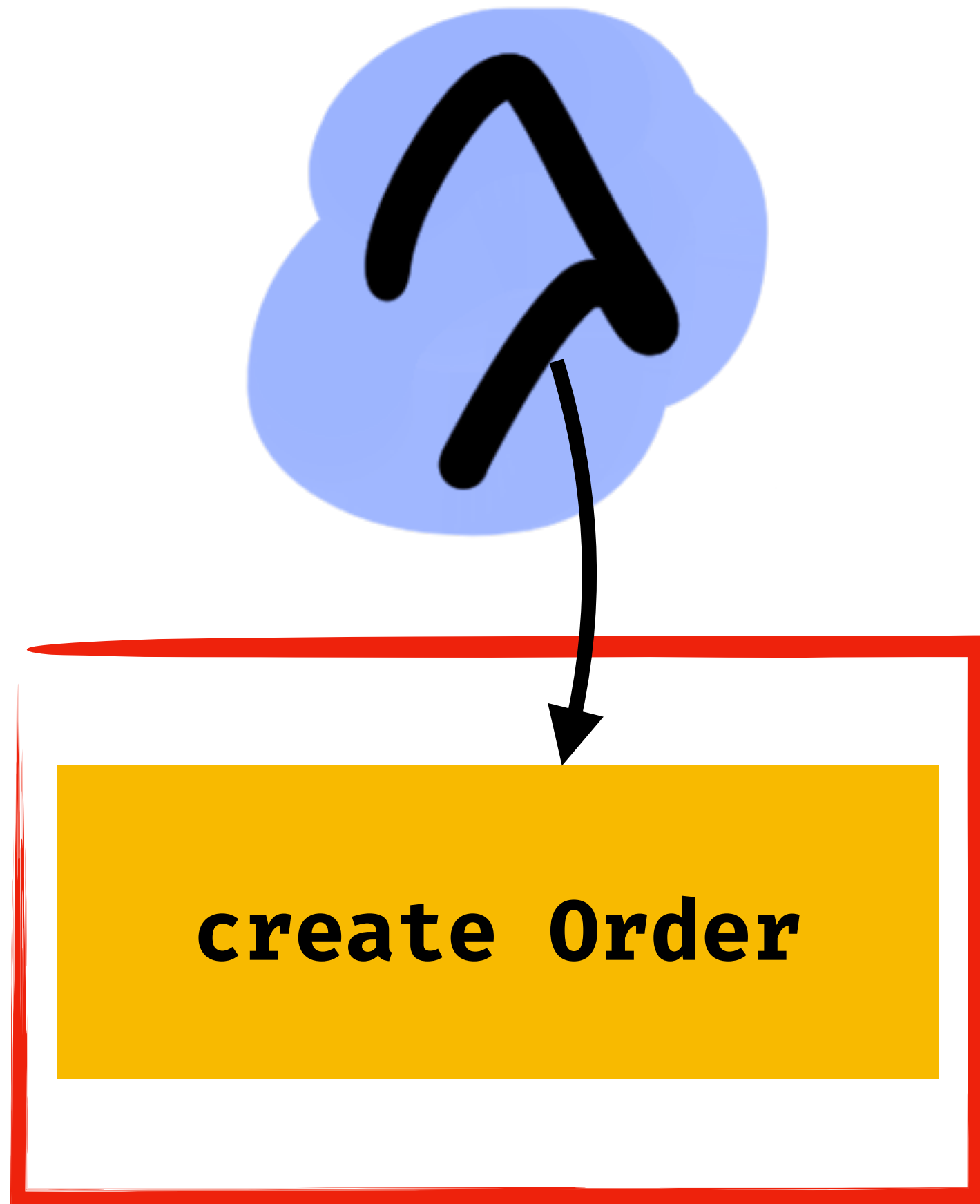
create Order

Forward Order ZIP

forward Order

Cancel Order ZIP

cancel Order



EASE OF TRANSITION	
FLEXIBILITY	
SECURITY	
SIZING	
DEPLOYMENT	
COMPLEXITY	



**DEPLOYMENT STILL
SUCKS**

APEX

Apex



Chalice



Claudia.js

Claudia.js



Serverless Express



Serverless
Framework



Serverless Java
Container



Sparta

ZAPPA

Zappa

AWS LAMBDA MAY NEED
NO SPECIAL FRAMEWORK

VENDOR LOCK-IN

LOG4J

VS.

COMMONS-LOGGING

**YOU ABSOLUTELY NEED
TOOLING FOR
DEPLOYMENT**

AVOID UI DEPLOYMENTS

 When you save your function with active tracing enabled, Lambda will automatically add permissions: "xray:PutTraceSegments", "xray:PutTelemetryRecords" to the function's current role if it does not have necessary permissions.

⚠ When you save your function with active tracing enabled, Lambda will automatically add permissions: "xray:PutTraceSegments", "xray:PutTelemetryRecords" to the function's current role if it does not have necessary permissions.

Policy name ▼[AWSLambdaBasicExecutionRole-ef7...](#)[AWSLambdaBasicExecutionRole-9ffe...](#)[AWSLambdaBasicExecutionRole-7a4...](#)[AWSLambdaBasicExecutionRole-76e...](#)[AWSLambdaBasicExecutionRole-61a...](#)[AWSLambdaBasicExecutionRole-19a...](#)

MINIMAL LIST OF THINGS NEEDED TO RUN A LAMBDA

CODE AND DEPENDENCIES

HANDLER

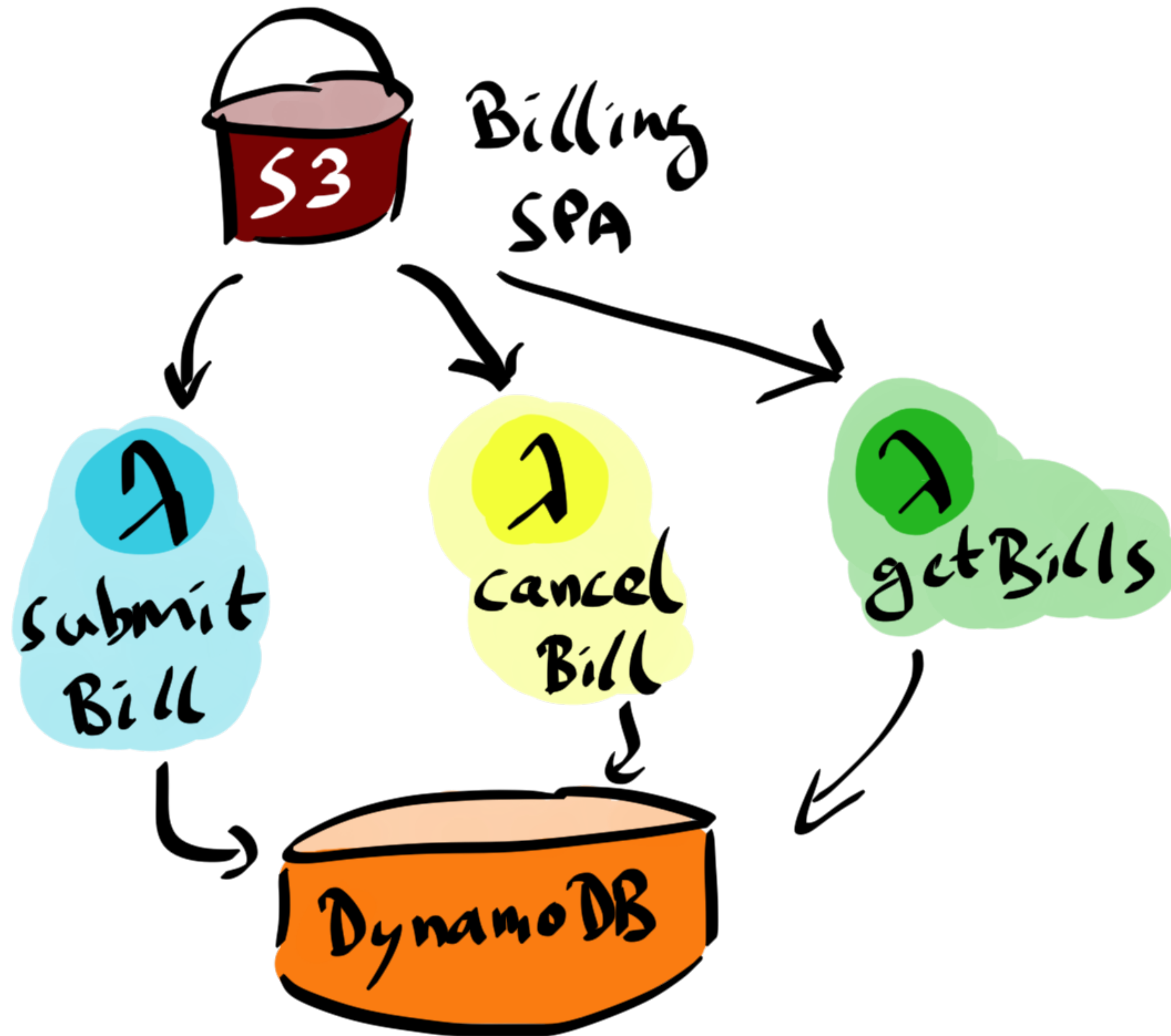
NAME

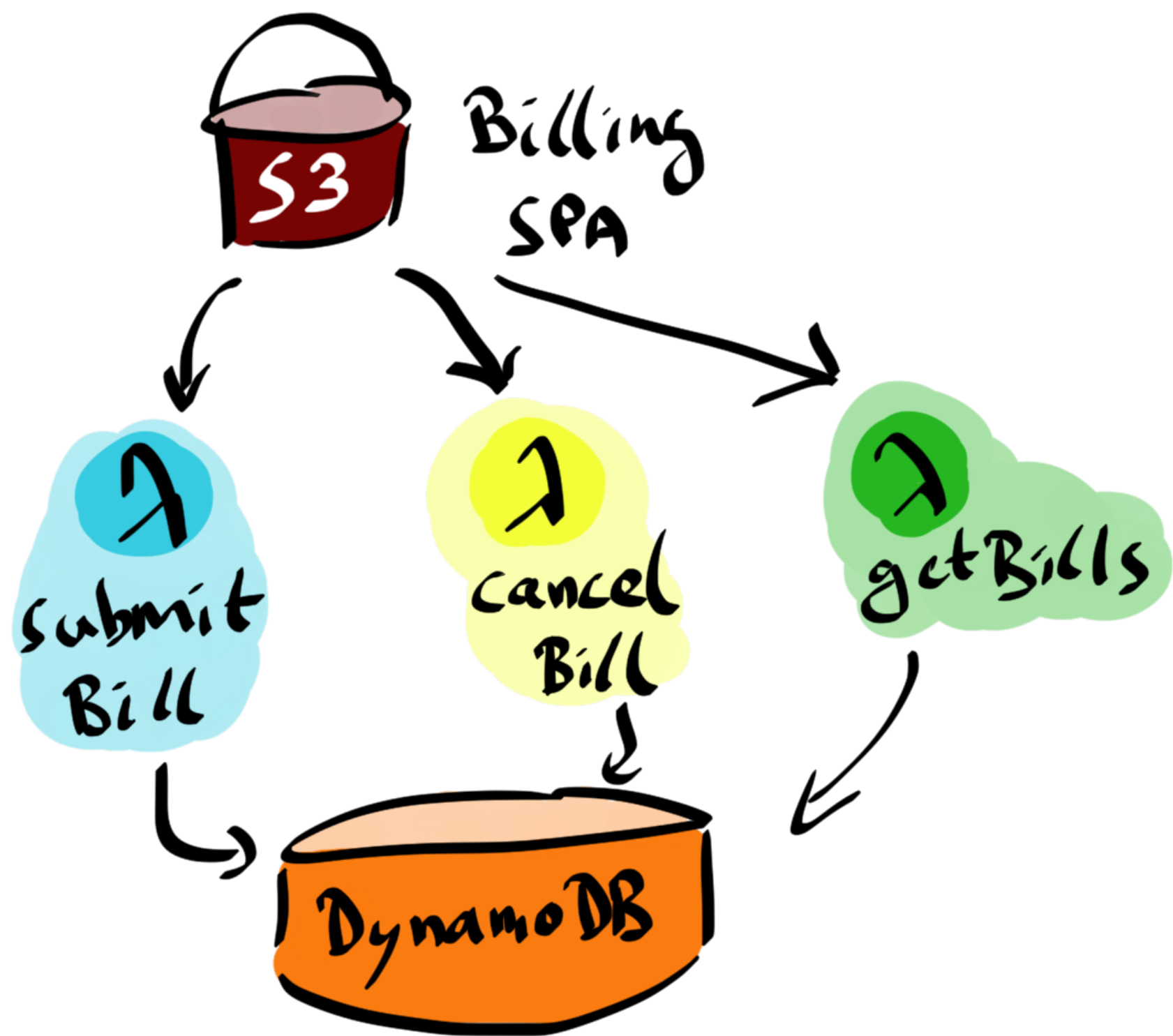
RUNTIME

ROLE AND POLICY


```
$ aws lambda create-function \  
    --function-name HelloWorld \  
    --runtime nodejs6.10 \  
    --role arn:aws:iam::...:... \  
    --handler index.handler \  
    --zip-file fileb://index.zip
```

```
$ aws lambda create-function \  
    --function-name HelloWorld \  
    --runtime nodejs6.10 \  
    --role arn:aws:iam::...:... \  
    --handler index.handler \  
    --zip-file fileb://index.zip
```





**S3 BUCKET WITH
WEBSITE HOSTING
3 LAMBDA + ALIAS
3 ROLES, POLICIES
DYNAMODB TABLE**

REAL AUTOMATION IS KEY



Terraform



```
$ claudia create \  
  --region eu-west-1 \  
  --handler index.handler \  
  --name HelloWorld
```



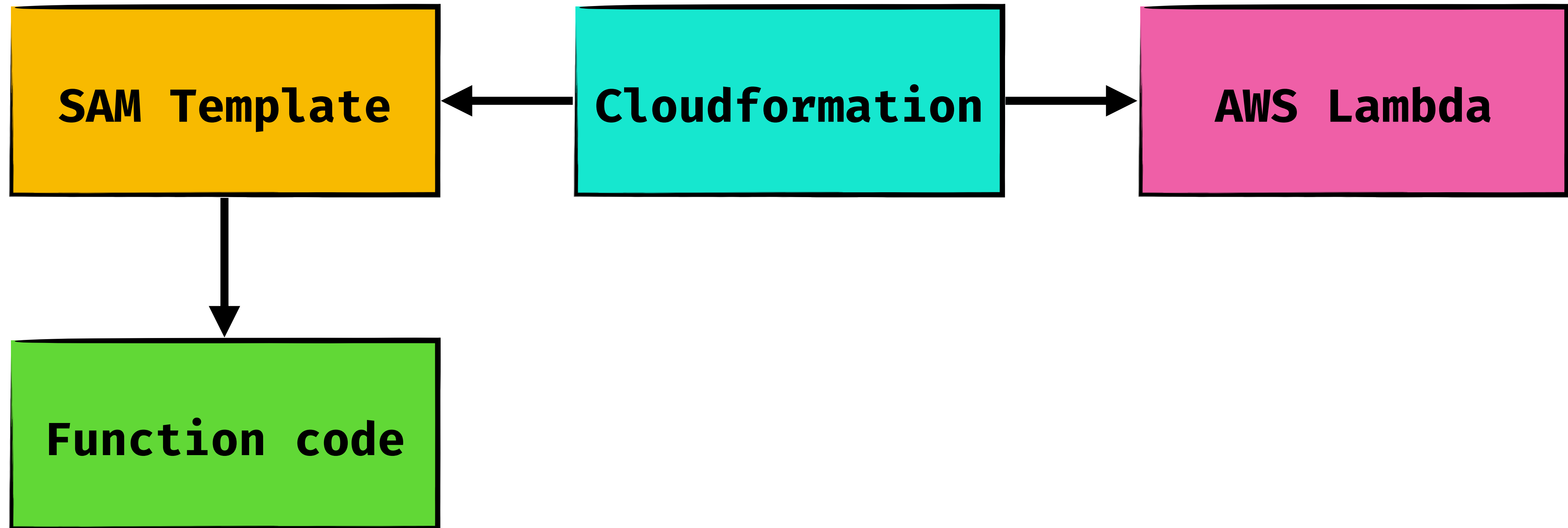
```
$ claudia create \  
  --region eu-west-1 \  
  --handler index.handler \  
  --name HelloWorld
```



MEET AWS SAM

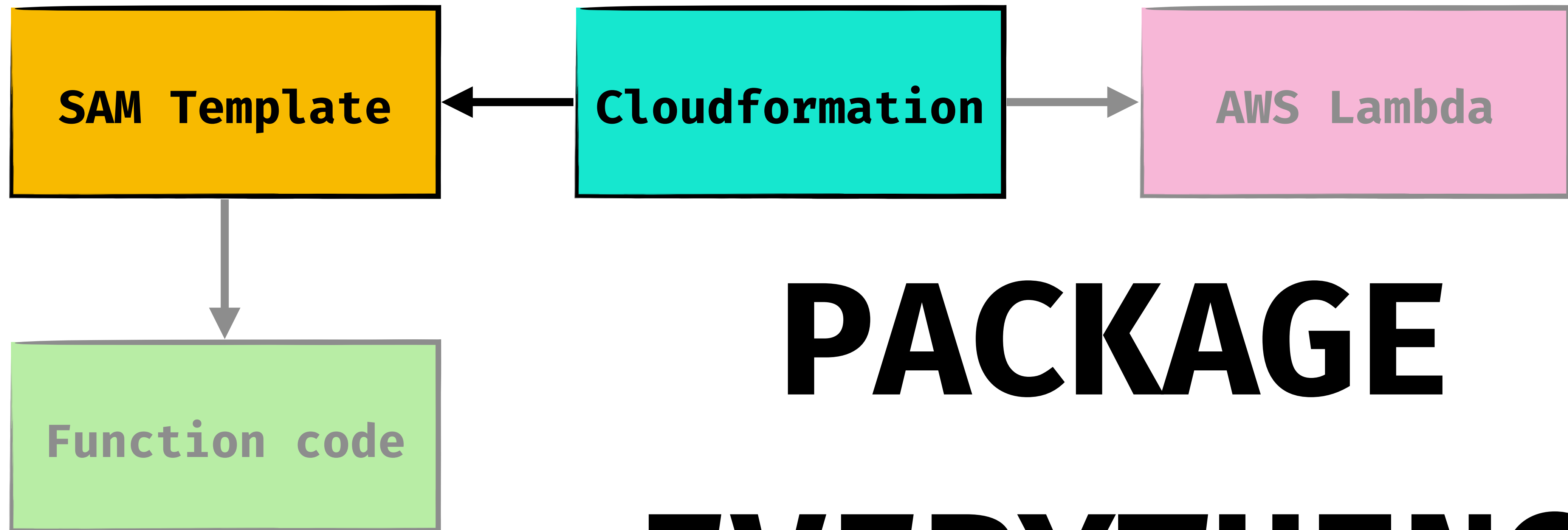


SERVERLESS APPLICATION MODEL

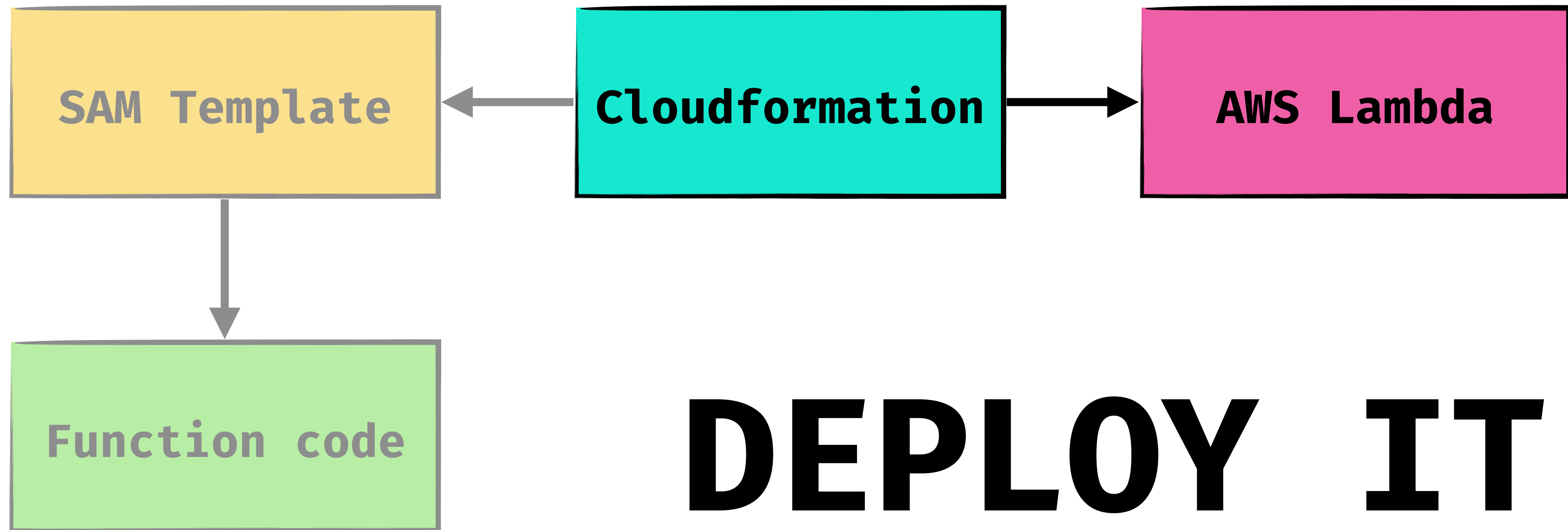




**DESCRIBE YOUR
FUNCTION**



**PACKAGE
EVERYTHING**



AWSTemplateFormatVersion: '2010-09-09'

Transform: AWS::Serverless-2016-10-31

Description: Hello World

Resources:

HelloWorld:

Type: AWS::Serverless::Function

Properties:

Timeout: 5

Runtime: nodejs6.10

Handler: index.handler

CodeUri: .


```
AWSTemplateFormatVersion: '2010-09-09'  
Transform: AWS::Serverless-2016-10-31
```

```
Description: Hello World
```

```
Resources:
```

```
    HelloWorld:
```

```
        Type: AWS::Serverless::Function
```

```
        Properties:
```

```
            Timeout: 5
```

```
            Runtime: nodejs6.10
```

```
            Handler: index.handler
```

```
            CodeUri: .
```



```
AWSTemplateFormatVersion: '2010-09-09'  
Transform: AWS::Serverless-2016-10-31
```

```
Description: Hello World
```

```
Resources:
```

```
    HelloWorld:
```

```
        Type: AWS::Serverless::Function
```

```
        Properties:
```

```
            Timeout: 5
```

```
            Runtime: nodejs6.10
```

```
            Handler: index.handler
```

```
            CodeUri: .
```

ALL TOOLS HAVE DIFFERENT DRAWBACKS

DEBUGGING ROLLBACK STATE FEATURE COMPLETENESS

A yellow sponge is shown with a patch of green moss growing on its top surface. The sponge is positioned in the lower half of the frame, and the moss is a vibrant green color. The background is a light gray gradient.

**CLEANLINESS IS NEXT
TO GODLINESS**

INTEGRATION TEST

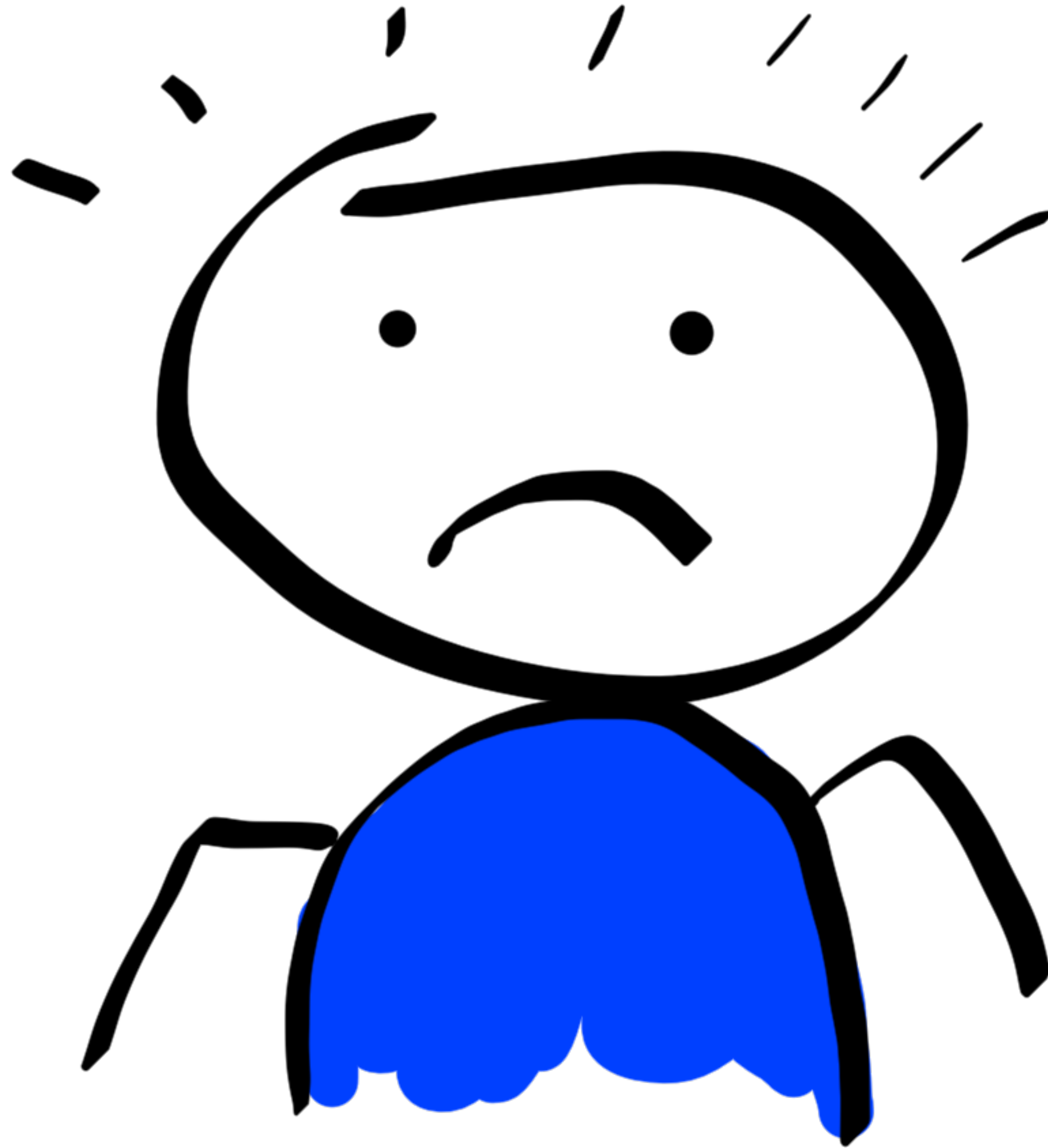
RIGHT FROM THE START



SAM

LOCAL

```
$ sam -version  
sam version 0.2.6
```

YOUR TESTING TOOLS
SHOULD BE AS
MATURE AS YOUR
PRODUCTION TOOLS

**TESTING LAMBDA FOR
REAL IS JUST TOO EASY**

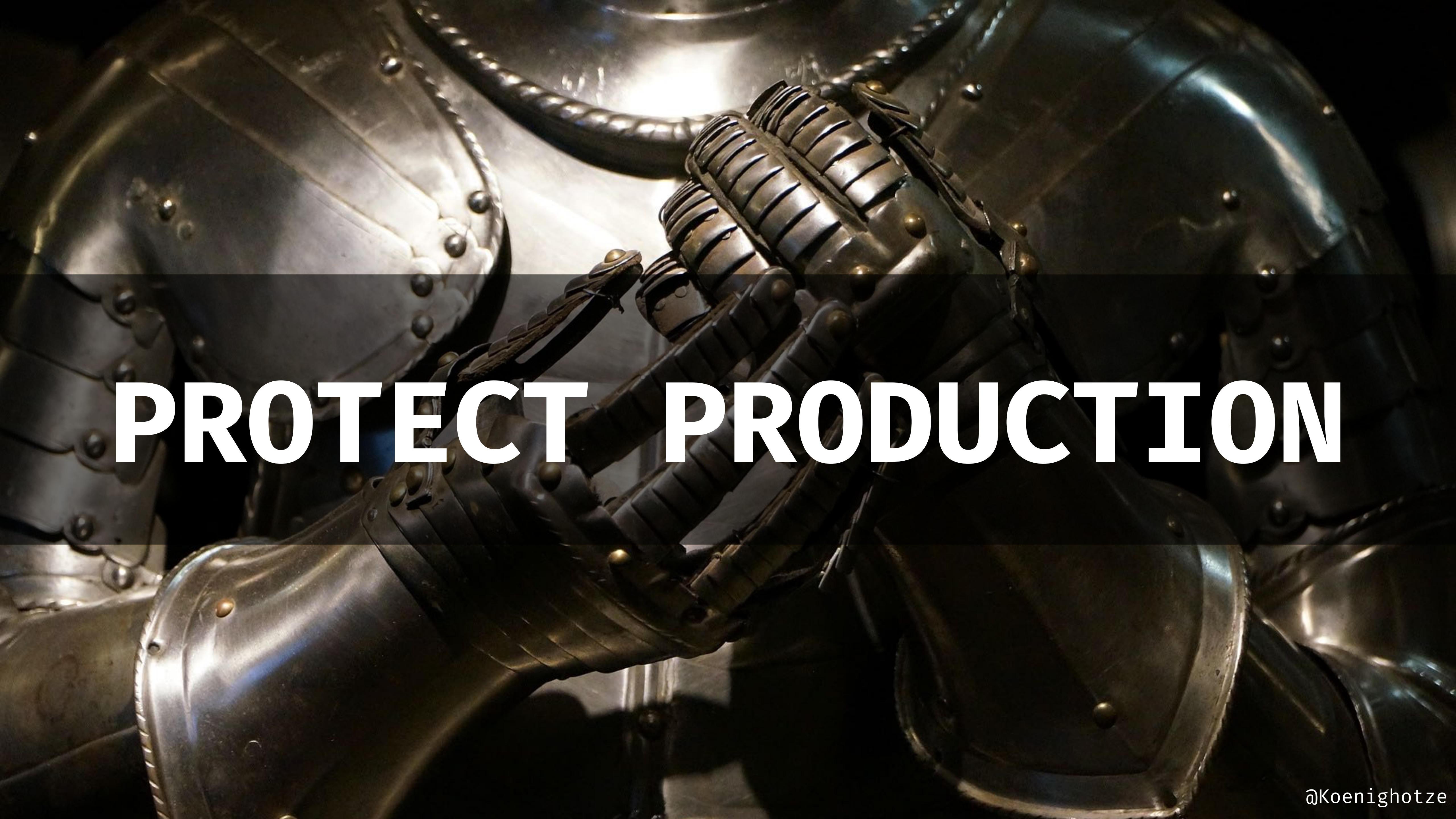
```
aws lambda invoke \  
  --function-name ReceiptUpload \  
  --region eu-central-1 \  
  --payload file://expensive_receipt.json \  
  out.json
```



```
aws lambda invoke \  
  --function-name ReceiptUpload \  
  --region eu-central-1 \  
  --payload file://expensive_receipt.json \  
  out.json
```

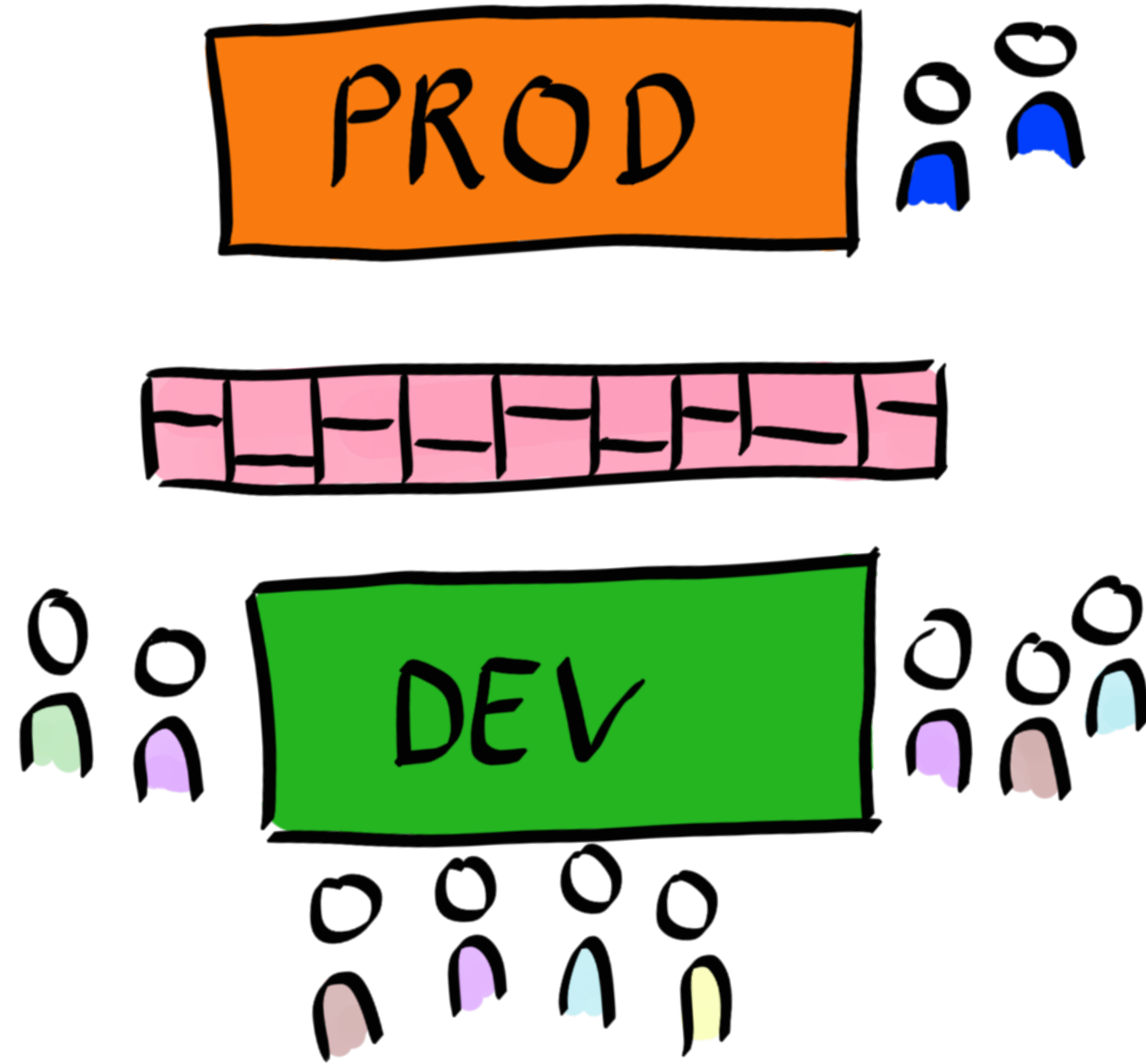


```
aws lambda invoke \  
  --function-name ReceiptUpload \  
  --region eu-central-1 \  
  --payload file://expensive_receipt.json \  
  out.json
```

PROTECT PRODUCTION

STAGING IS A THING



Parameters:

...

Stage:

Type: String

Default: DEV

ForwardBilling:

Type: AWS::Serverless::Function

Properties:

FunctionName: !Sub ForwardBilling-\${Stage}

Parameters:

...

Stage:

Type: String

Default: DEV

ForwardBilling:

Type: AWS::Serverless::Function

Properties:

FunctionName: !Sub ForwardBilling-\${Stage}

```
aws cloudformation deploy  
  --template-file ...  
  --stack-name ...  
  --parameter-overrides Stage=PROD
```



```
aws cloudformation deploy  
  --template-file ...  
  --stack-name ...  
  --parameter-overrides Stage=PROD
```


**YES ALIASES HELP,
TOO**

CLEANUP **UNUSED** **LAMBDA**



HOW DO YOU **REASON** ABOUT CLOUD ARCHITECTURES?



HOW TO KNOW WHAT TO CLEANUP?

<https://github.com/epsagon/list-lambdas>

**PYTHON LIST_LAMBDAS.PY **
—INACTIVE-DAYS-FILTER 10

Region	Function	Last Modified	Last Invocation
eu-central-1	digital_logistics_03_fb_webhook	54 days ago	42 days ago
eu-central-1	fis-projects-dev-newProject	53 days ago	54 days ago
eu-central-1	marlon_login_handler	54 days ago	54 days ago
eu-central-1	copy-picture-lambda-dev-hello	144 days ago	144 days ago
eu-central-1	Fis2_0-Contract-DismissContract	52 days ago	52 days ago
eu-central-1	marlon_account_linking	54 days ago	54 days ago
eu-central-1	test	238 days ago	238 days ago
eu-central-1	digital_logistics_03_user_login	55 days ago	42 days ago
eu-central-1	fis-projects-dev-getProjects	53 days ago	52 days ago
eu-central-1	Fis2_0-Employees-GetEmployees	53 days ago	27 days ago
eu-central-1	digital_logistics_03_dialogflow_webhook	55 days ago	39 days ago
eu-central-1	transactions-api-dev-getpictures	144 days ago	143 days ago
eu-central-1	digital_logistics_03_update_parcel_status	55 days ago	42 days ago
eu-central-1	LambdaHelloWorldTest	144 days ago	N/A (no invocations?)
eu-central-1	AwsWatchDog	7 days ago	7 days ago
eu-central-1	Fis2_0-Contract-GetContractByEmployee	52 days ago	27 days ago
eu-central-1	fis-projects-dev-deleteProject	53 days ago	54 days ago
eu-central-1	HelloWorld	88 days ago	88 days ago
eu-central-1	Fis2_0-Employees-GetEmployee	53 days ago	53 days ago
eu-central-1	Fis2_0-Employees-GetContractByEmployee	53 days ago	53 days ago
eu-central-1	fis-projects-dev-getProject	53 days ago	54 days ago
eu-west-1	digital_logistics_alexahandler	55 days ago	55 days ago
eu-west-1	theMotivator	111 days ago	72 days ago
eu-west-1	DevconScheduleToSpeech	97 days ago	94 days ago
eu-west-1	wasserstandinfo	36 days ago	8 days ago

Region	Function	Last Modified	Last Invocation
eu-central-1	digital_logistics_03_dev_newhook	54 days ago	42 days ago
eu-central-1	fis-projects-dev-newProject	53 days ago	54 days ago
eu-central-1	marlon_login_handler	54 days ago	54 days ago
eu-central-1	copy-picture-lambda-dev-hello	144 days ago	144 days ago
eu-central-1	Fis2_0-Contract-DismissContract	52 days ago	52 days ago
eu-central-1	marlon_account_linking	54 days ago	54 days ago
eu-central-1	test	238 days ago	238 days ago
eu-central-1	digital_logistics_03_user_login	55 days ago	42 days ago
eu-central-1	fis-projects-dev-getProjects	53 days ago	52 days ago
eu-central-1	Fis2_0-Employees-GetEmployees	53 days ago	27 days ago
eu-central-1	digital_logistics_03_dialogflow_webhook	55 days ago	39 days ago
eu-central-1	transactions-api-dev-getpictures	144 days ago	143 days ago
eu-central-1	digital_logistics_03_update_parcel_status	55 days ago	42 days ago
eu-central-1	LambdaHelloWorldTest	144 days ago	N/A (no invocations?)
eu-central-1	AwsWatchDog	7 days ago	7 days ago
eu-central-1	Fis2_0-Contract-GetContractByEmployee	52 days ago	27 days ago
eu-central-1	fis-projects-dev-deleteProject	53 days ago	54 days ago
eu-central-1	HelloWorld	88 days ago	88 days ago
eu-central-1	Fis2_0-Employees-GetEmployee	53 days ago	53 days ago
eu-central-1	Fis2_0-Employees-GetContractByEmployee	53 days ago	53 days ago
eu-central-1	fis-projects-dev-getProject	53 days ago	54 days ago
eu-west-1	digital_logistics_alex_a_handler	55 days ago	55 days ago
eu-west-1	theMotivator	111 days ago	72 days ago
eu-west-1	DevconScheduleToSpeech	97 days ago	94 days ago
eu-west-1	wasserstandinfo	36 days ago	8 days ago

Region	Function	Last Modified	Last Invocation
eu-central-1	digital_logistics_03_fb_webhook	54 days ago	42 days ago
eu-central-1	fis-projects-dev-newProject	53 days ago	54 days ago
eu-central-1	marlon_login_handler	54 days ago	54 days ago
eu-central-1	copy-picture-lambda-dev-hello	144 days ago	144 days ago
eu-central-1	Fis2_0-Contract-DismissContract	52 days ago	52 days ago
eu-central-1	marlon_login_handler	54 days ago	54 days ago
eu-central-1	test	238 days ago	238 days ago
eu-central-1	digital_logistics_03_user_login	55 days ago	42 days ago
eu-central-1	fis-projects-dev-getProjects	53 days ago	52 days ago
eu-central-1	Fis2_0-Employees-GetEmployees	53 days ago	27 days ago
eu-central-1	digital_logistics_03_dialogflow_webhook	55 days ago	39 days ago
eu-central-1	transactions-api-dev-getpictures	144 days ago	143 days ago
eu-central-1	digital_logistics_03_update_parcel_status	55 days ago	42 days ago
eu-central-1	LambdaHelloWorldTest	144 days ago	N/A (no invocations?)
eu-central-1	AwsWatchDog	7 days ago	7 days ago
eu-central-1	Fis2_0-Contract-GetContractByEmployee	52 days ago	27 days ago
eu-central-1	fis-projects-dev-deleteProject	53 days ago	54 days ago
eu-central-1	HelloWorld	88 days ago	88 days ago
eu-central-1	Fis2_0-Employees-GetEmployee	53 days ago	53 days ago
eu-central-1	Fis2_0-Employees-GetContractByEmployee	53 days ago	53 days ago
eu-central-1	fis-projects-dev-getProject	53 days ago	54 days ago
eu-west-1	digital_logistics_alexander_handler	55 days ago	55 days ago
eu-west-1	theMotivator	111 days ago	72 days ago
eu-west-1	DevconScheduleToSpeech	97 days ago	94 days ago
eu-west-1	wasserstandinfo	36 days ago	8 days ago

REDUCE ATTACK SURFACE

REDUCE COGNITIVE

CLUTTER

STAY IN CONTROL



SECURITY IS A THING

SERVERLESS IS
GOOD FOR SECURITY

NO LONG LIVED SERVERS
OR MANUAL PATCH
SCHEDULES

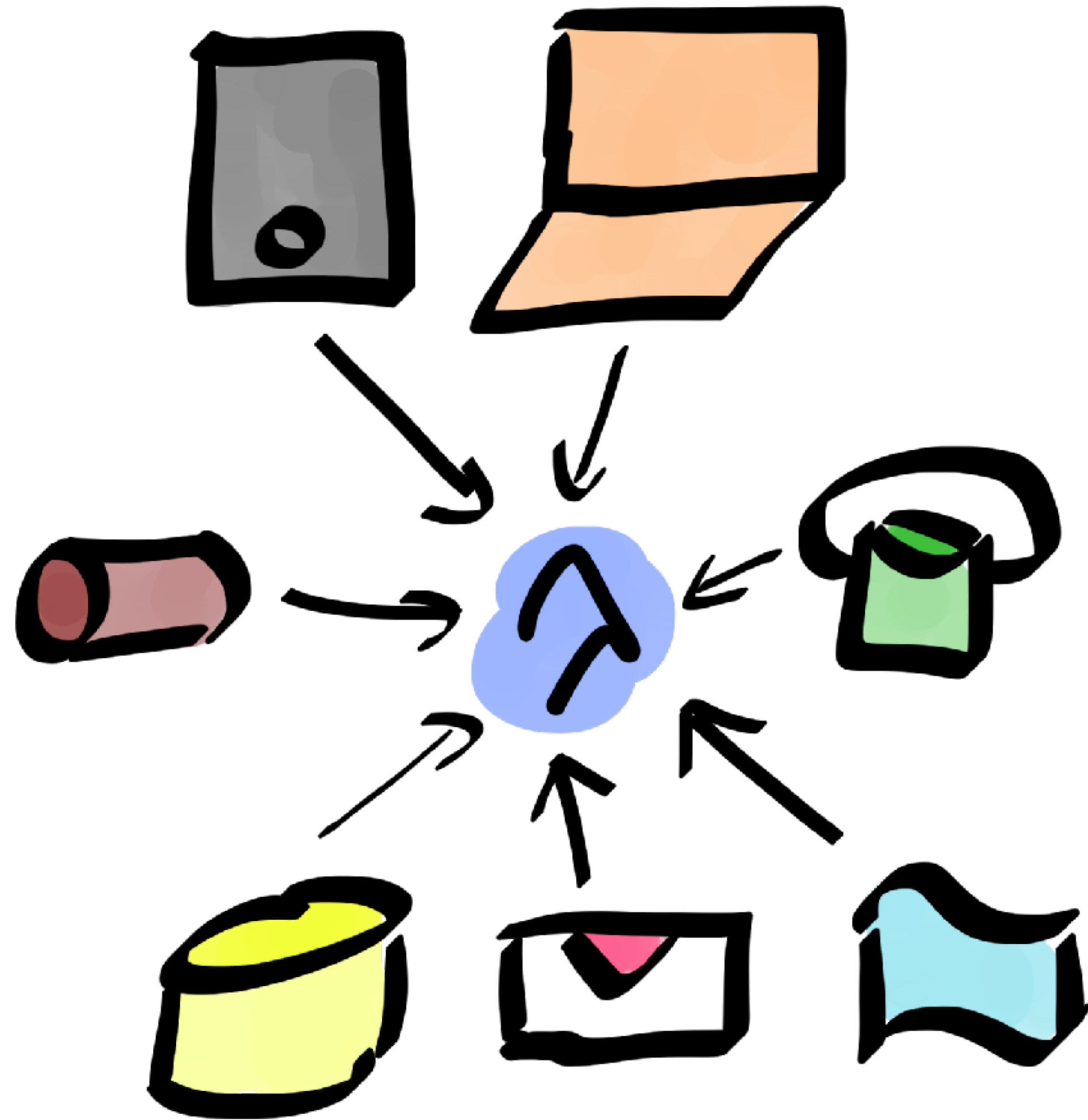
**THAT YOU NEED TO
HANDLE**

...BUT AMAZON DOES

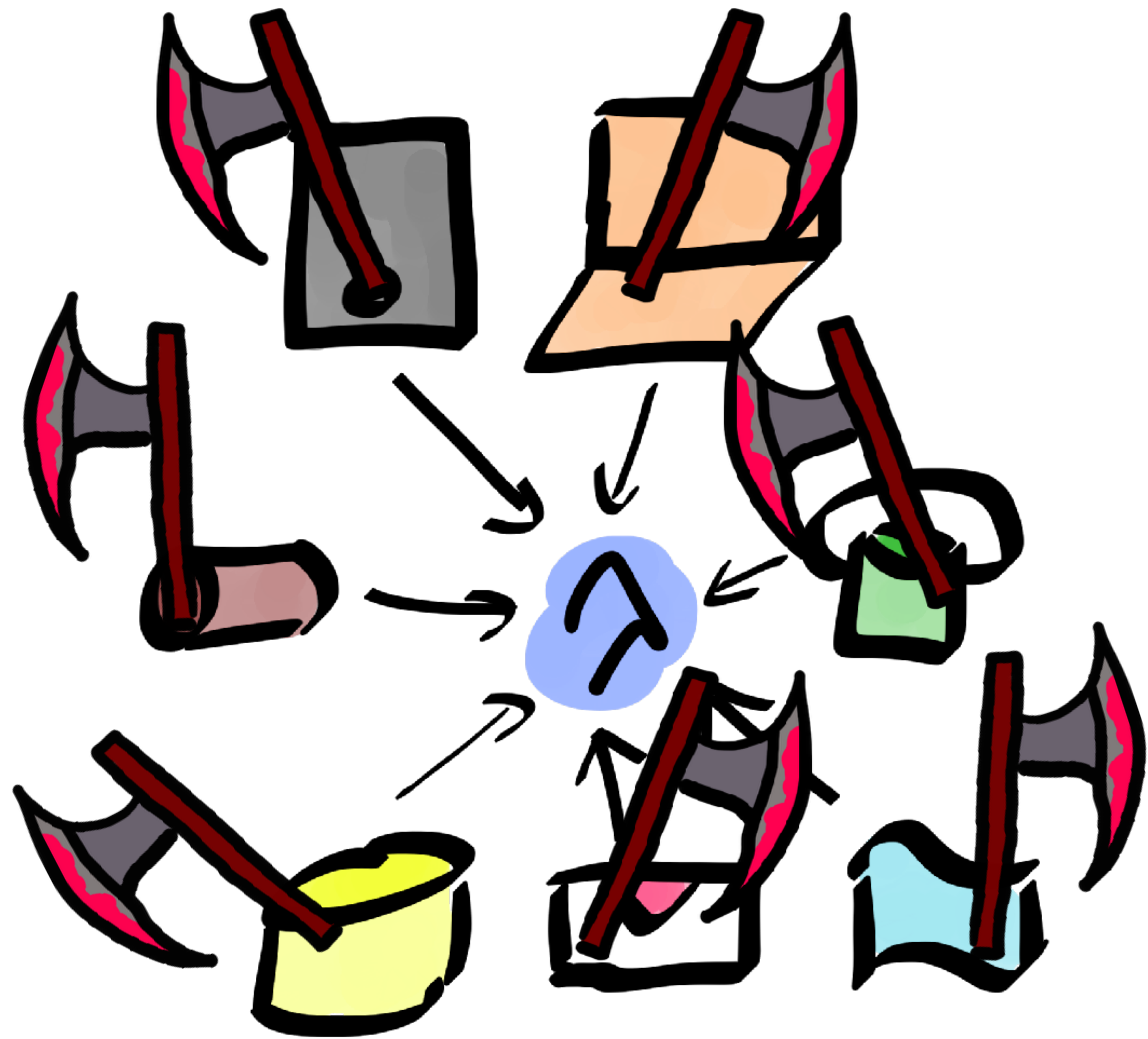
**HOW DO YOU DEPLOY A WEB
APPLICATION FIREWALL...**
...WITHOUT A WEB
...WITHOUT AN APPLICATION
...WITHOUT A SERVER

NEW OR ADAPTED
SECURE CODING
PATTERNS NEEDED

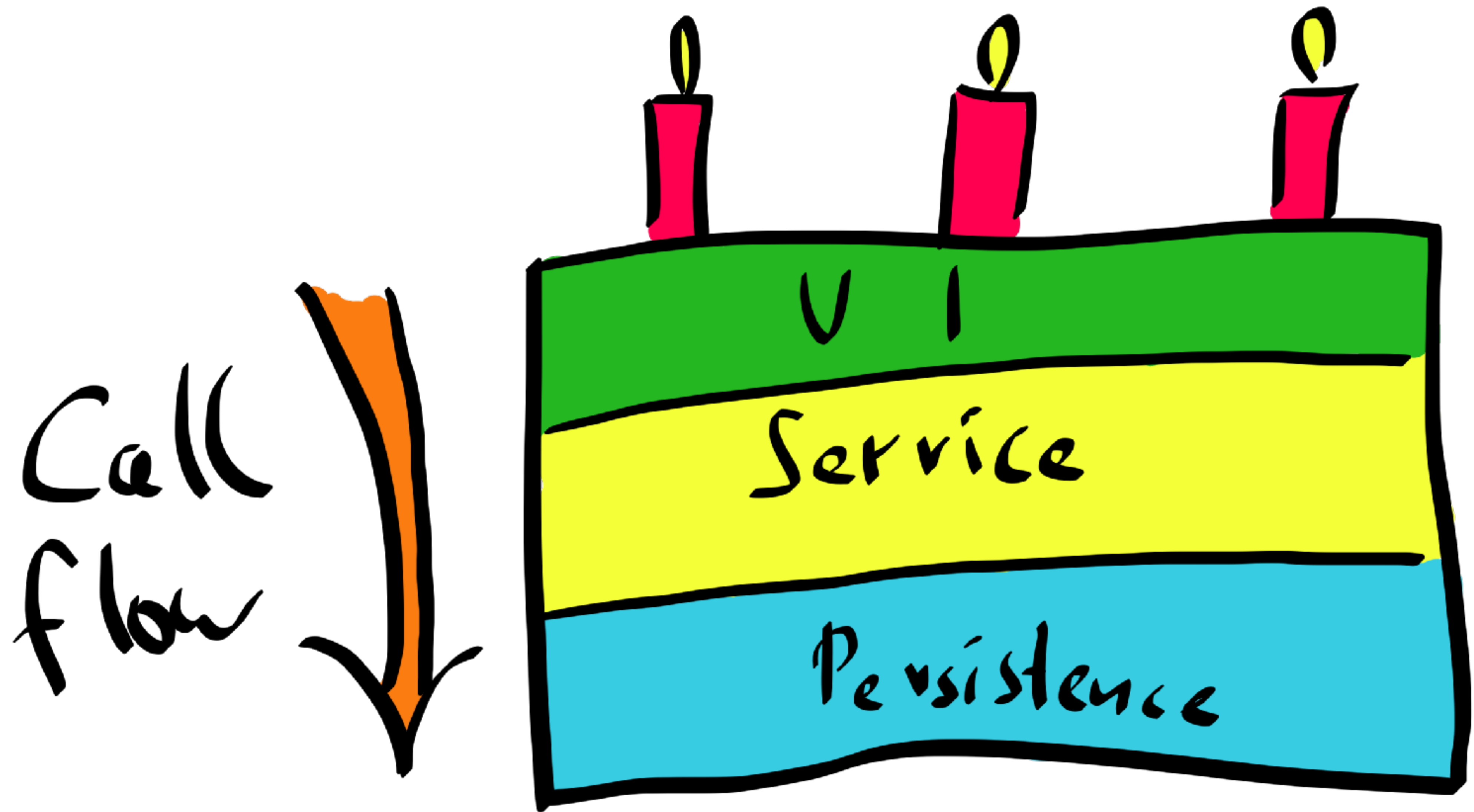
INCREASED ATTACK SURFACE



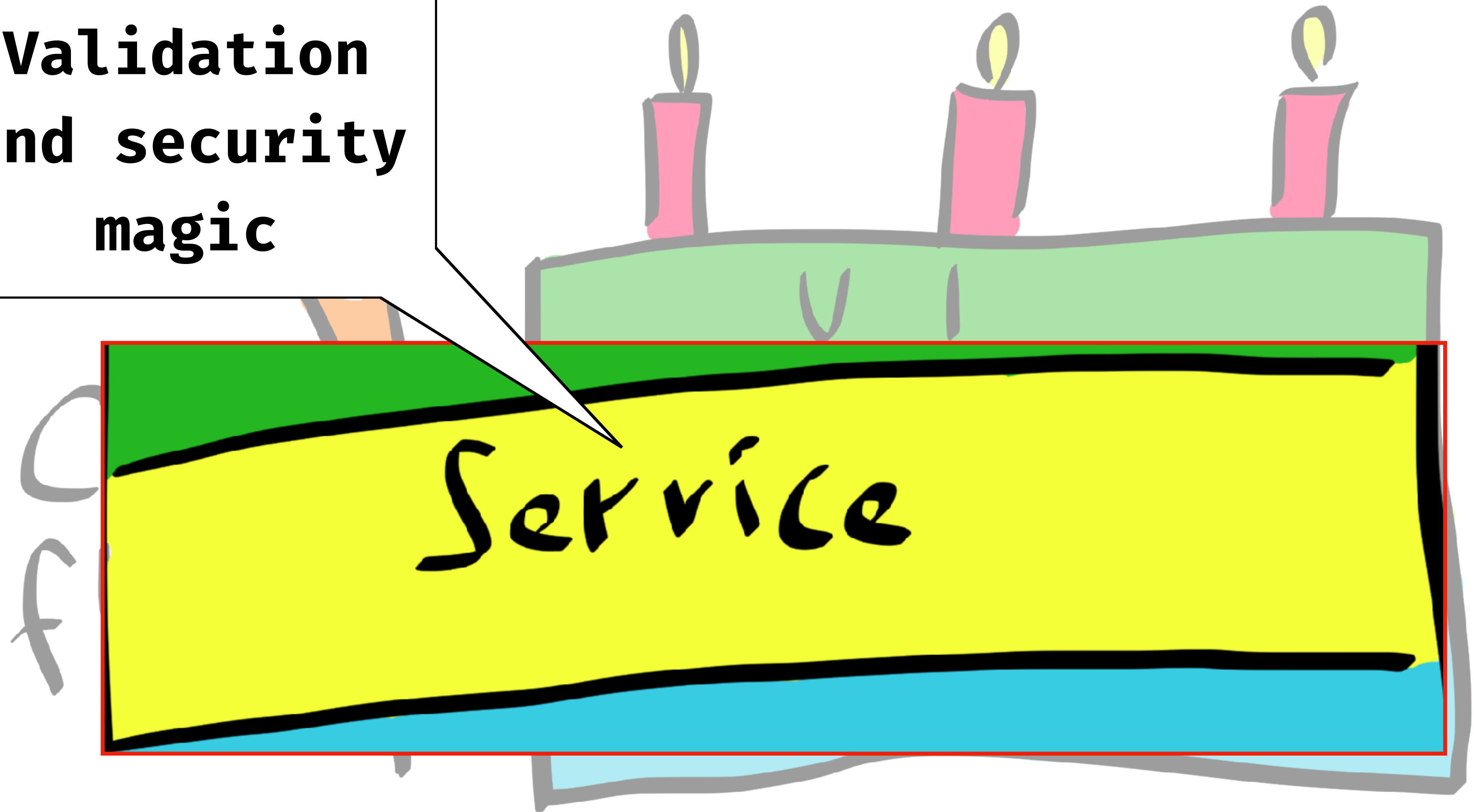
IMAGINE EVERYBODY
WANTS TO HARM YOU



EVENT INJECTION



**Validation
and security
magic**



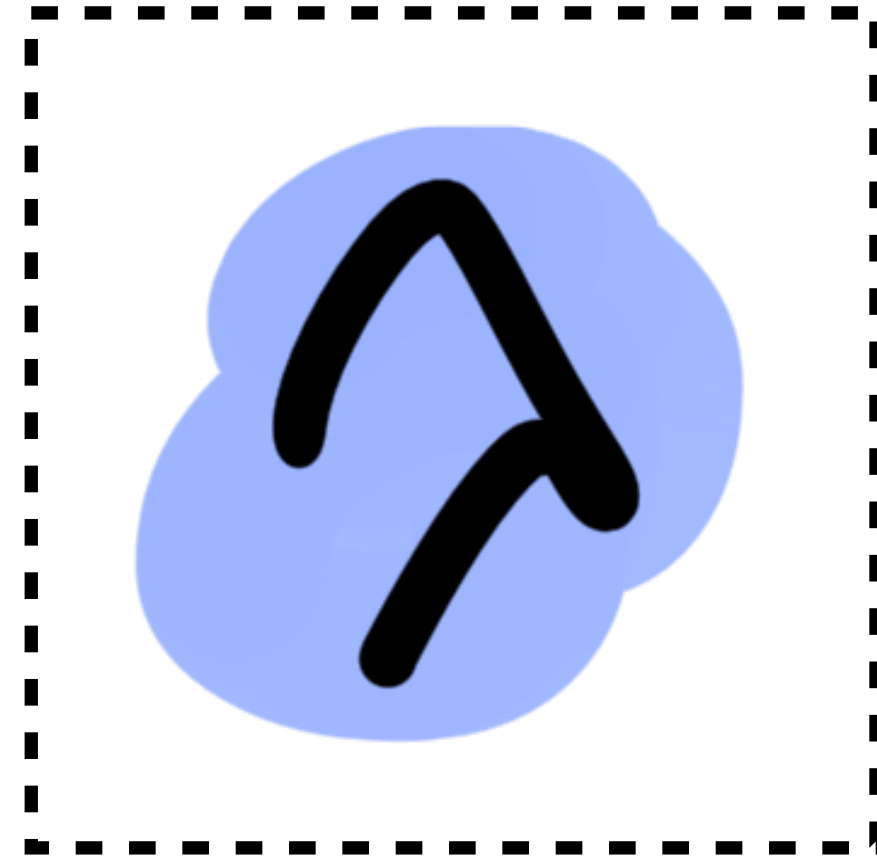
**CLASSIC LAYERED CAKE
ARCHITECTURE DOES NOT
APPLY TO SERVERLESS**

**NEVER TRUST YOUR
CALLER**

Security boundary



Security boundary



Security boundary



Security boundary



BILLING ATTACK

Billing Alarm

You can create a billing alarm to receive e-mail alerts when your AWS charges exceed a threshold you choose. Simply:

1. Enter a spending threshold
2. Provide an email address
3. Check your inbox for a confirmation email and click the link provided

When my total AWS charges for the month

exceed: \$ USD

send a notification to:

NotifyMe

[New list](#)

Reminder: for each address you add, you will receive an email from AWS with the subject "AWS Notification - Subscription Confirmation". Click the link provided in the message to confirm that AWS may deliver alerts to that address.

SECURITY IS LIKE CSS
NO ONE DELETES RULES
YOU ONLY ADD RULES

BEWARE CRAPPY
CODE EXAMPLES


```
"Statement": [{  
  "Effect": "Allow",  
  "Action": [  
    "s3:*",  
  ],  
  "Resource": [  
    "arn:aws:s3:::*"  
  ]  
}, ...]
```

```
"Statement": [{  
  "Effect": "Allow",  
  "Action": [  
    "s3:*",  
  ],  
  "Resource": [  
    "arn:aws:s3:::*"  
  ]  
}, ...]
```

SECURITY SHOTGUN

PRINCIPLE OF LEAST PRIVILEGES

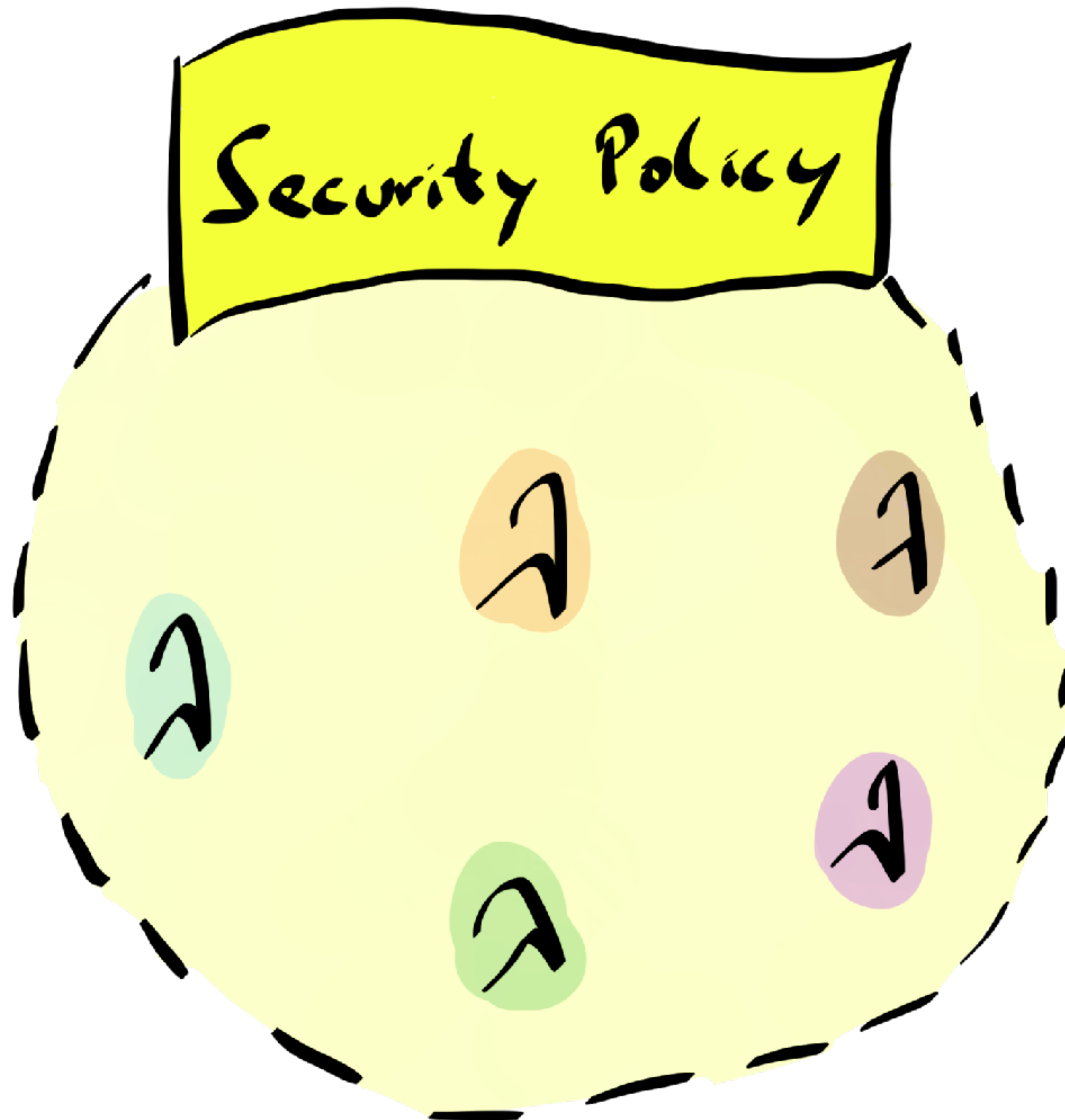

```
"Statement": [{  
  "Effect": "Allow",  
  "Action": [  
    "s3:*",  
  ],  
  "Resource": [  
    "arn:aws:s3:::senacor/fis/orders/*"  
  ]  
}, ...]
```

```
"Statement": [{  
  "Effect": "Allow",  
  "Action": [  
    "s3:GetObject"  
  ],  
  "Resource": [  
    "arn:aws:s3:::senacor/fis/orders/*"  
  ]  
}, ...]
```

```
"Statement": [{  
  "Effect": "Allow",  
  "Action": [  
    "s3:GetObject"  
  ],  
  "Resource": [  
    "arn:aws:s3:::senacor/fis/orders/*"  
  ]  
}, ...]
```


**“ALLOW ONLY READING
OBJECTS FROM A
SPECIFIC BUCKET”**

WHAT ABOUT SHARING CONFIGURATION?



DO NOT SHARE
POLICIES



**NO EXCUSES WITH
PROPER TOOLING**

Resources:

ForwardBillingRole:

Type: "AWS::IAM::Role"

Properties:

RoleName: *!Sub* "ForwardBillingRole-\${Stage}"

AssumeRolePolicyDocument: ...

ManagedPolicyArns: ...

Policies:

-

PolicyName: *!Sub* "StoreBillingInS3Policy-\${Stage}"

PolicyDocument: ...

ForwardBilling:

Type: AWS::Serverless::Function

Properties:

Role: *!GetAtt* ForwardBillingRole.Arn

Resources:

ForwardBillingRole:

Type: "AWS::IAM::Role"

Properties:

RoleName: *!Sub* "ForwardBillingRole-\${Stage}"

AssumeRolePolicyDocument: ...

ManagedPolicyArns: ...

Policies:

-

PolicyName: *!Sub* "StoreBillingInS3Policy-\${Stage}"

PolicyDocument: ...

ForwardBilling:

Type: AWS::Serverless::Function

Properties:

Role: *!GetAtt* ForwardBillingRole.Arn

Resources:

ForwardBillingRole:

Type: "AWS::IAM::Role"

Properties:

RoleName: *!Sub* "ForwardBillingRole-\${Stage}"

AssumeRolePolicyDocument: ...

ManagedPolicyArns: ...

Policies:

-

PolicyName: *!Sub* "StoreBillingInS3Policy-\${Stage}"

PolicyDocument: ...

ForwardBilling:

Type: AWS::Serverless::Function

Properties:

Role: *!GetAtt* ForwardBillingRole.Arn

Resources:

ForwardBillingRole:

Type: "AWS::IAM::Role"

Properties:

RoleName: *!Sub* "ForwardBillingRole-\${Stage}"

AssumeRolePolicyDocument: ...

ManagedPolicyArns: ...

Policies:

-

PolicyName: *!Sub* "StoreBillingInS3Policy-\${Stage}"

PolicyDocument: ...

ForwardBilling:

Type: AWS::Serverless::Function

Properties:

Role: *!GetAtt* ForwardBillingRole.Arn

USE A SECURITY WATCHDOG

**YOUR FUNCTION MAY BE
SMALL, BUT YOUR
DEPENDENCIES MAY NOT**


```
$ du -s src/
```

```
8.0K src/
```

```
$ ls -lh lambda.zip
```

```
7.6M Mar  1 07:13 lambda.zip
```



```
$ du -s src/
```

```
8.0K src/
```

```
$ ls -lh lambda.zip
```

```
7.6M Mar  1 07:13 lambda.zip
```



Snyk helps you use open source and stay secure.

Continuously find & fix vulnerabilities in your dependencies





Where is the code you want to test?

Choose from the sources connected to your organisation:



GitHub



AWS Lambda

Source not shown? [Browse available integrations](#)

18:42

snyk-bot APP

New remediation for Prototype Pollution

New upgrade to address this vulnerability in package **hoek**.
Affects project **senacor-aktuelles** in organisation koenighotze.
Fix with CLI **snyk wizard**.

Remediation

Upgrade

Project

senacor-aktuelles

Package

hoek

Version

4.2.0



serverless-puresec-cli

serverless ⚡ npm package 1.2.0

[Website](#) • [Newsletter](#) • [Twitter](#)

Serverless plugin for [PureSec CLI](#).

Features

- Saves you time - magically creates IAM roles for you
- Reduces the attack surface of your AWS Lambda based application
- Helps create least privileged roles with the minimum required permissions
- Currently supported runtimes: Node.js, Python (more runtimes coming soon...)
- Currently supported services: DynamoDB, Kinesis, KMS, Lambda, S3, SES, SNS & Step Functions
- Works with the [Serverless Framework](#)

**SECURITY MUST BE PART
OF DEVELOPMENT RIGHT
FROM THE START**

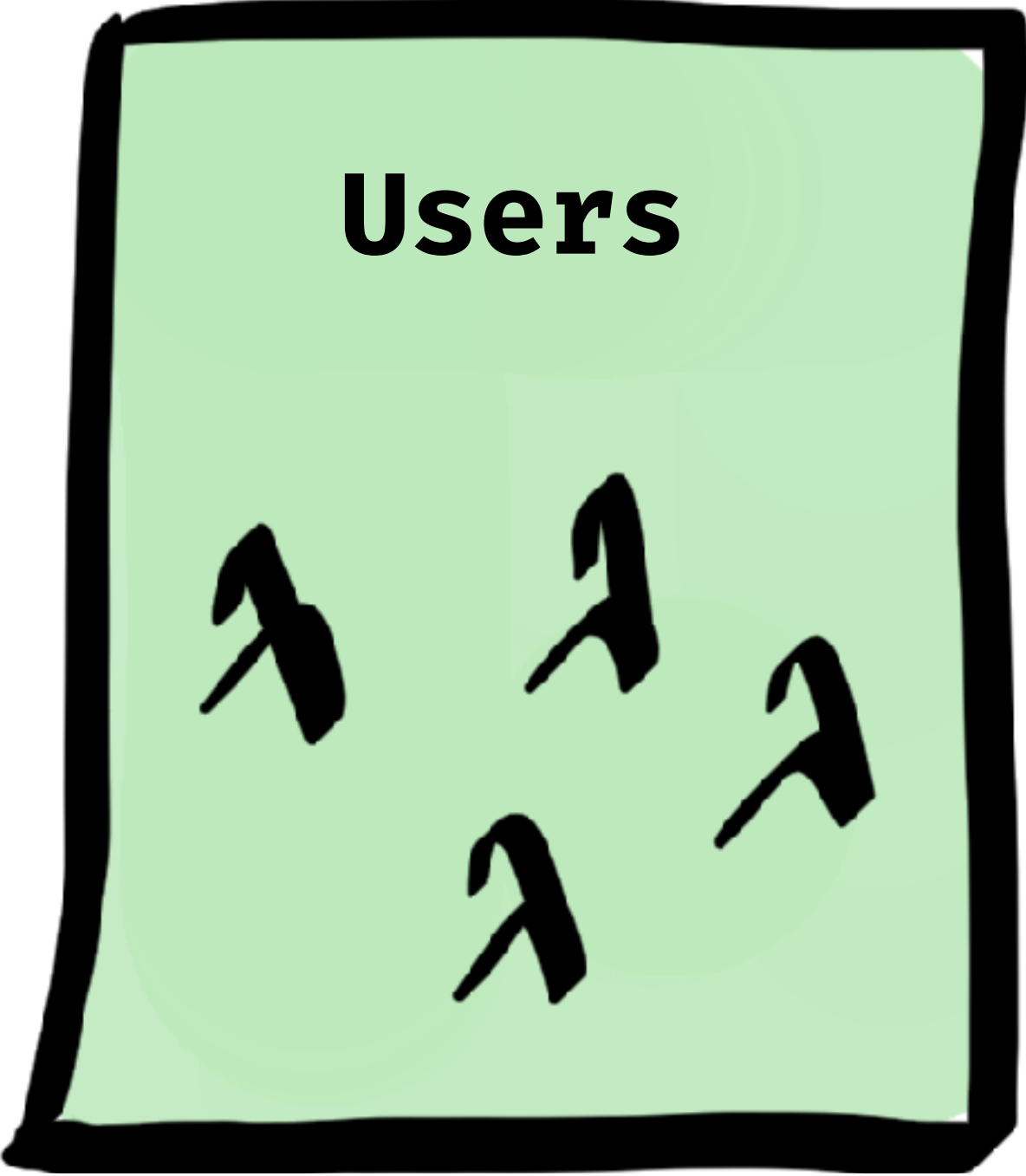
YOU BUILD IT

YOU RUN IT

YOU BUILD IT
YOU SECURE IT
AND THEN RUN IT

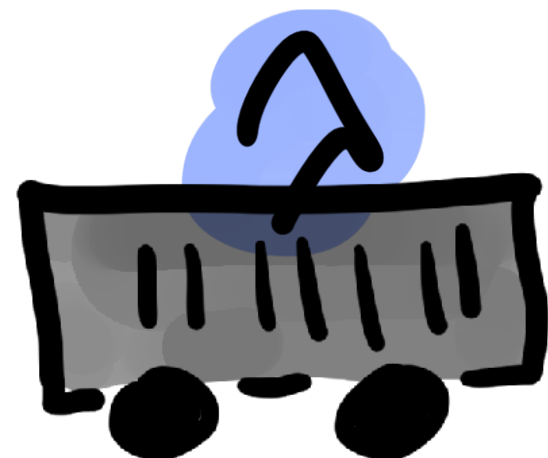
A blue car is parked in a grassy field with trees in the background. The hood of the car is open, and a person is visible working under it. The car is a dark blue color with silver wheels. The background shows a line of green trees under a clear sky.

LAMBDA WITCHCRAFT AKA. HOT AND COLD START

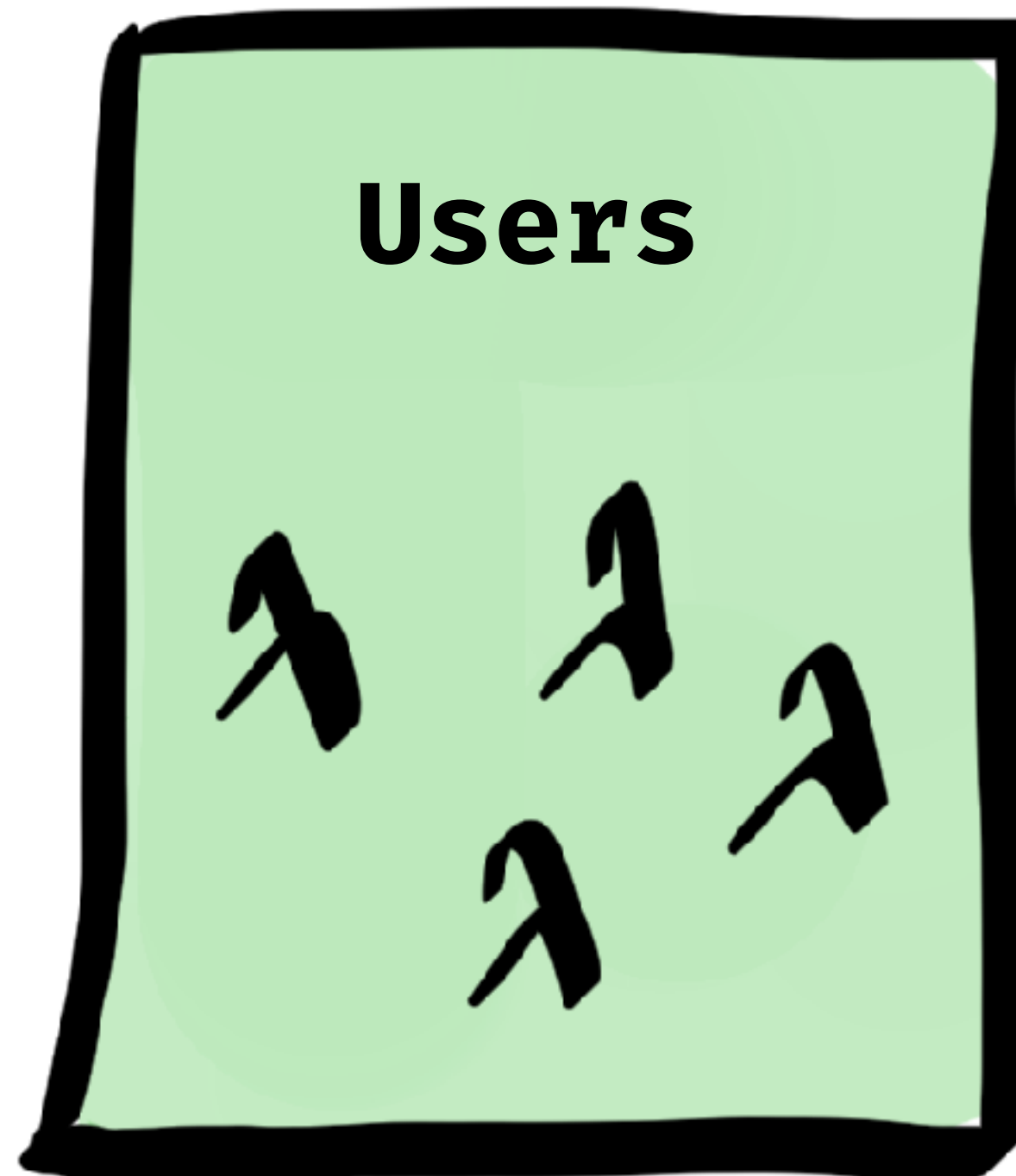




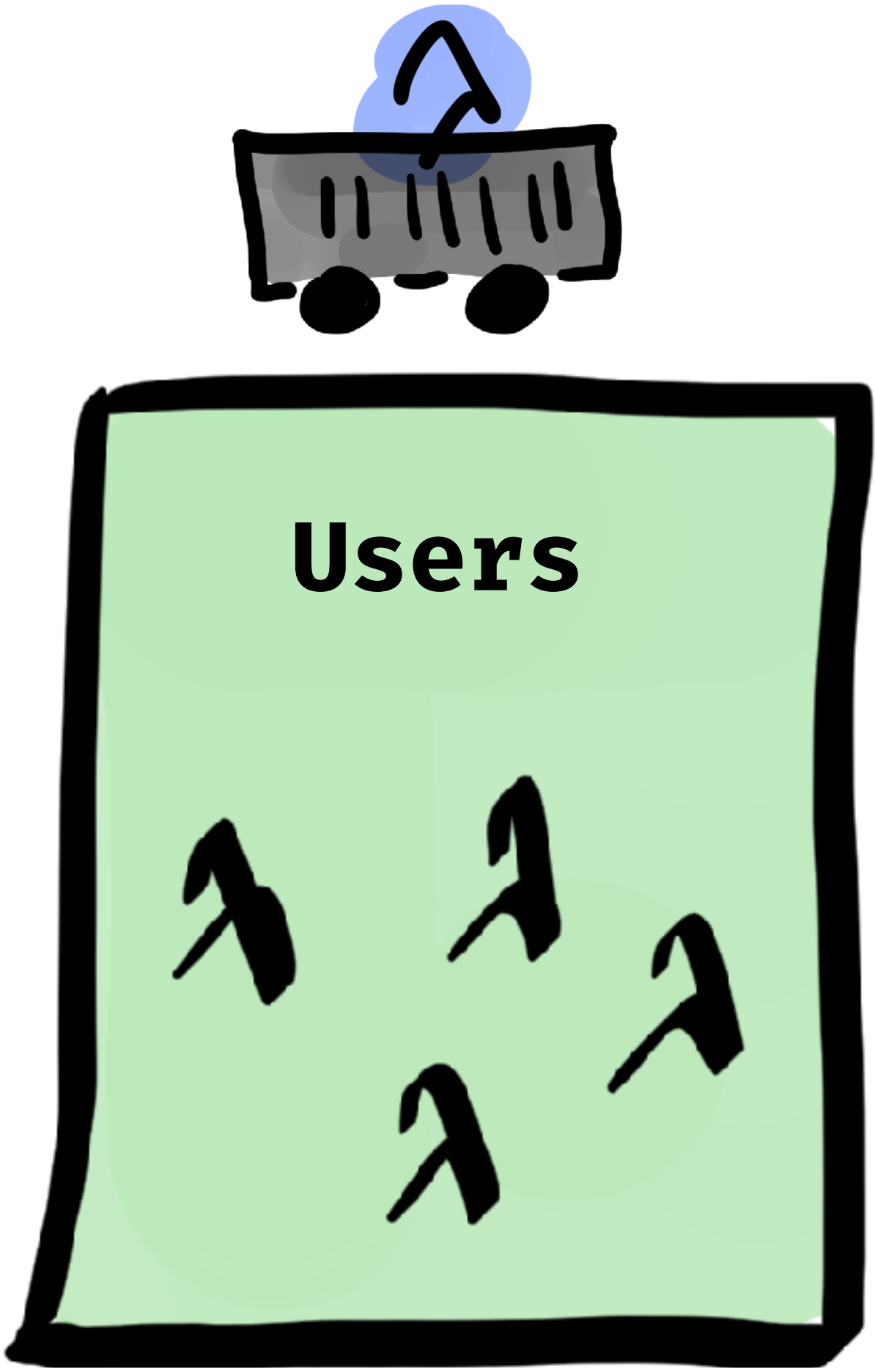
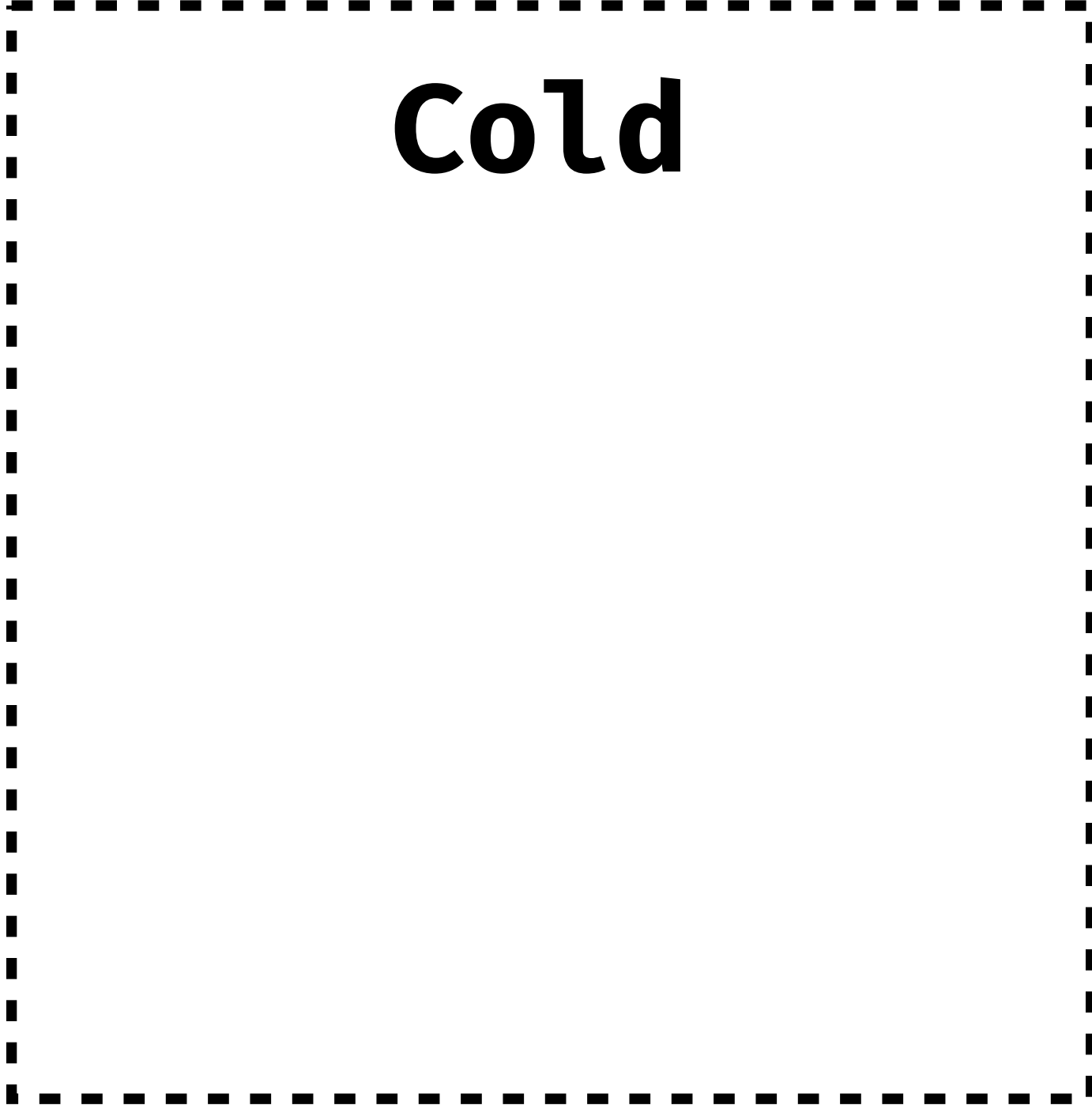
Cold

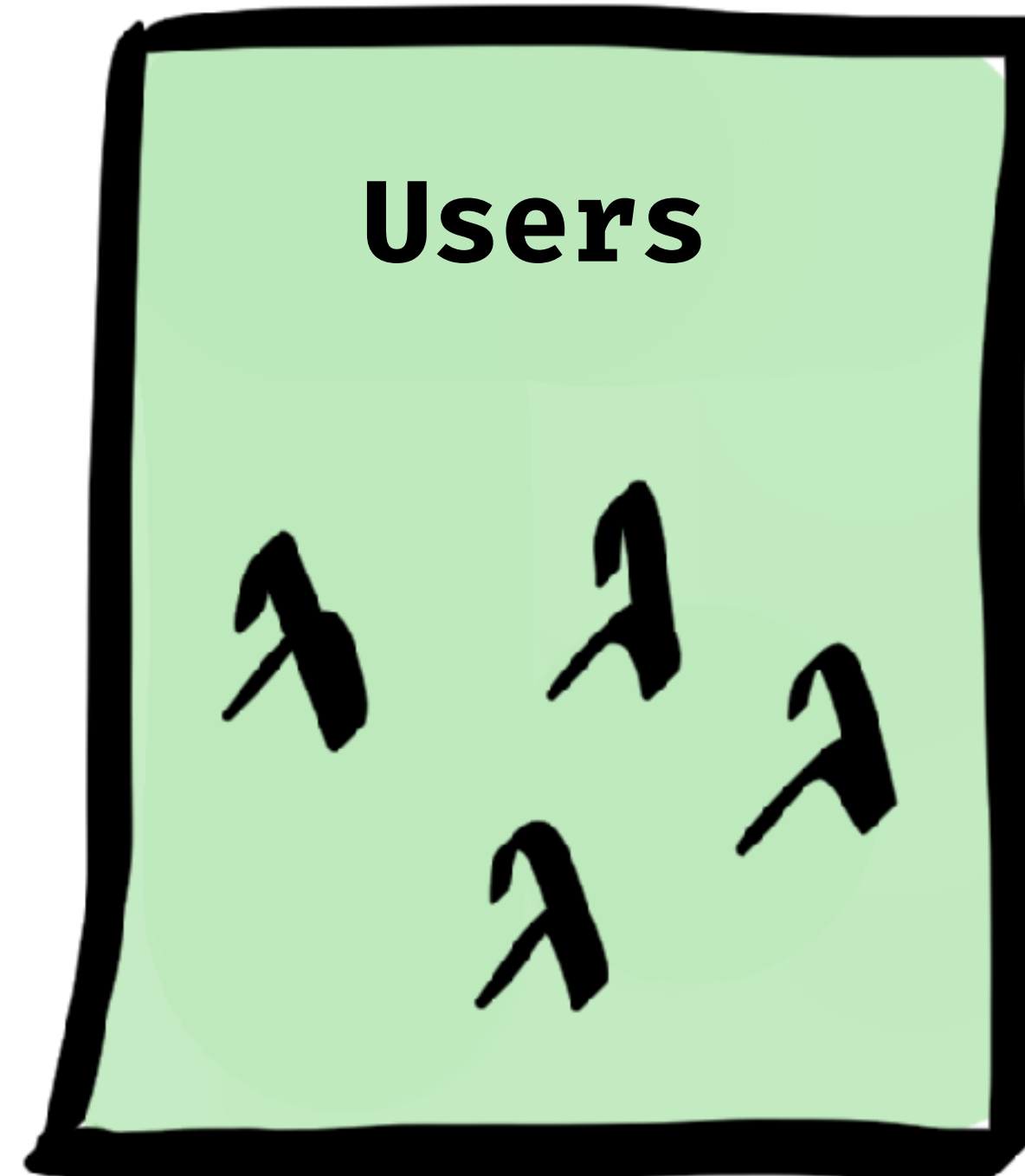


Users



COLD
START



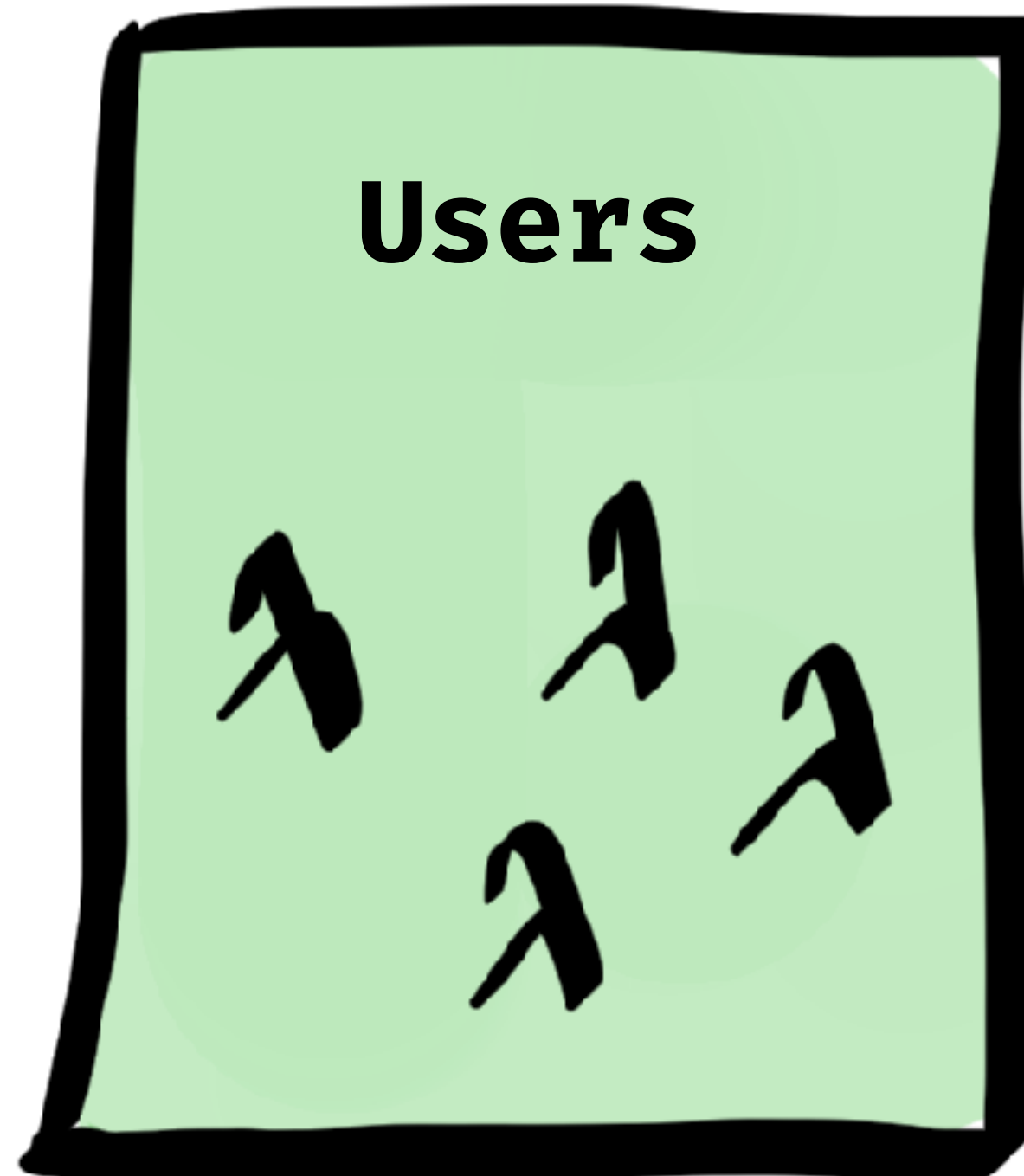
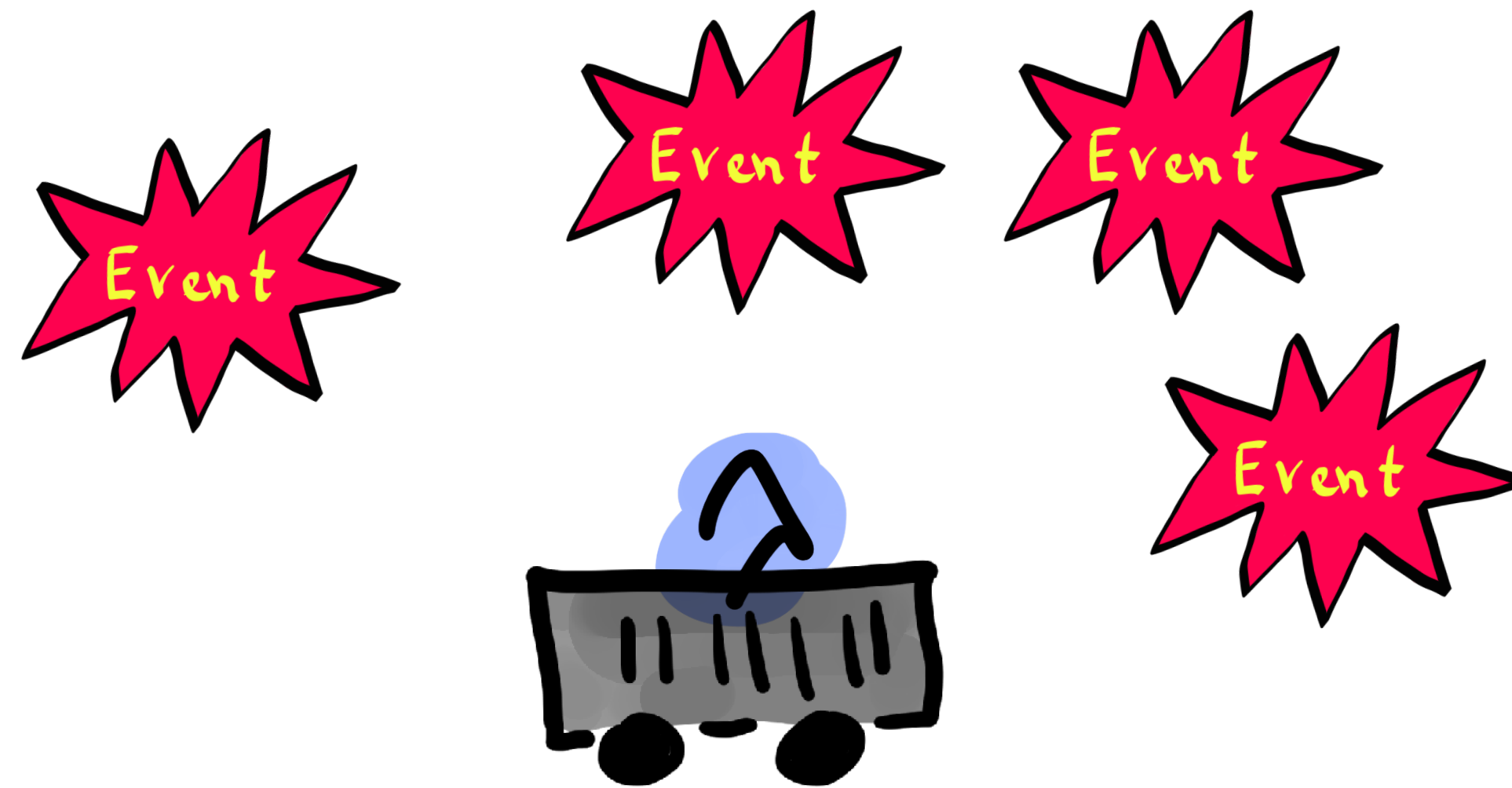
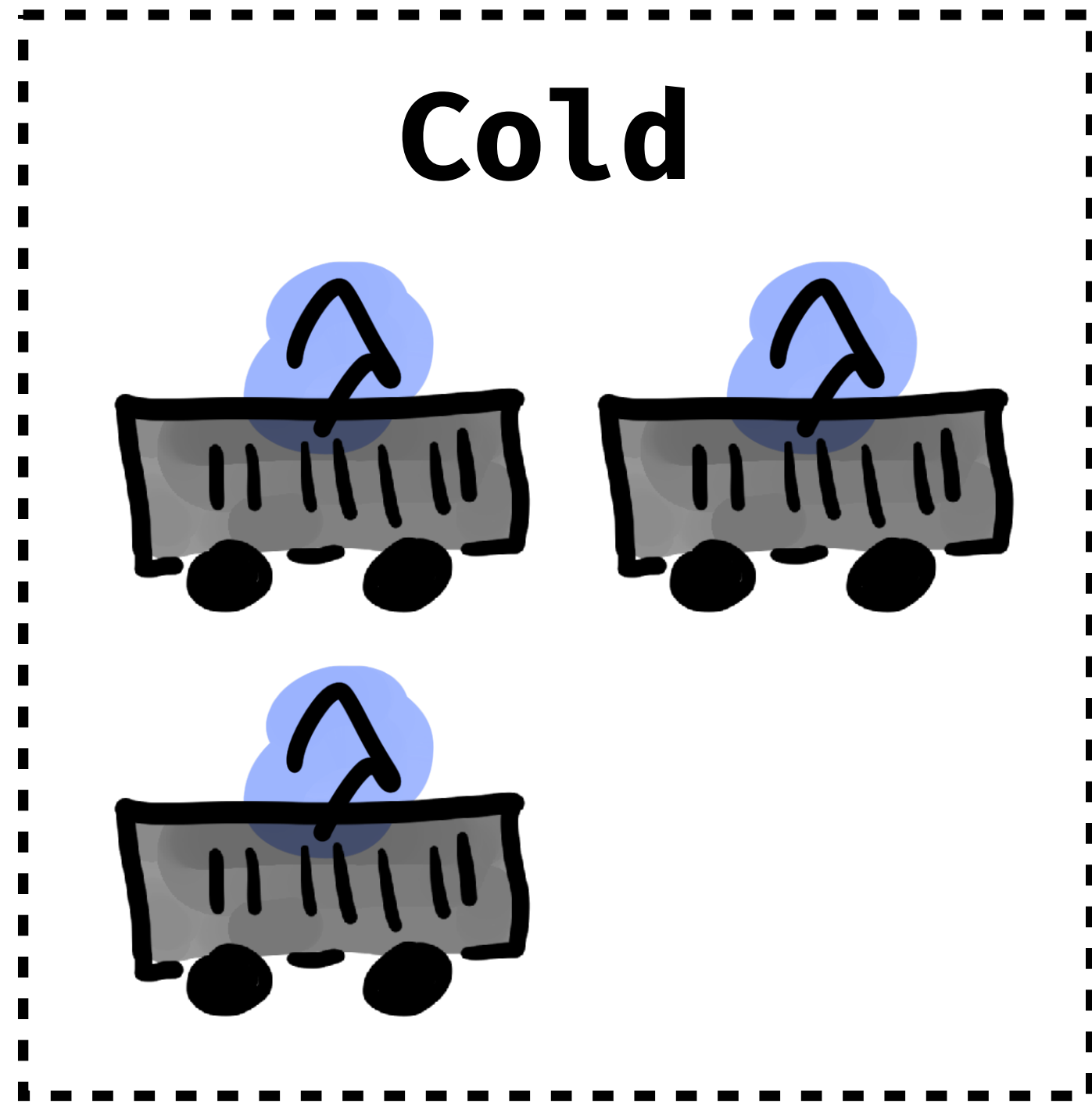


Cold

HOT
START

COLD START HAPPENS
ONCE, RIGHT?

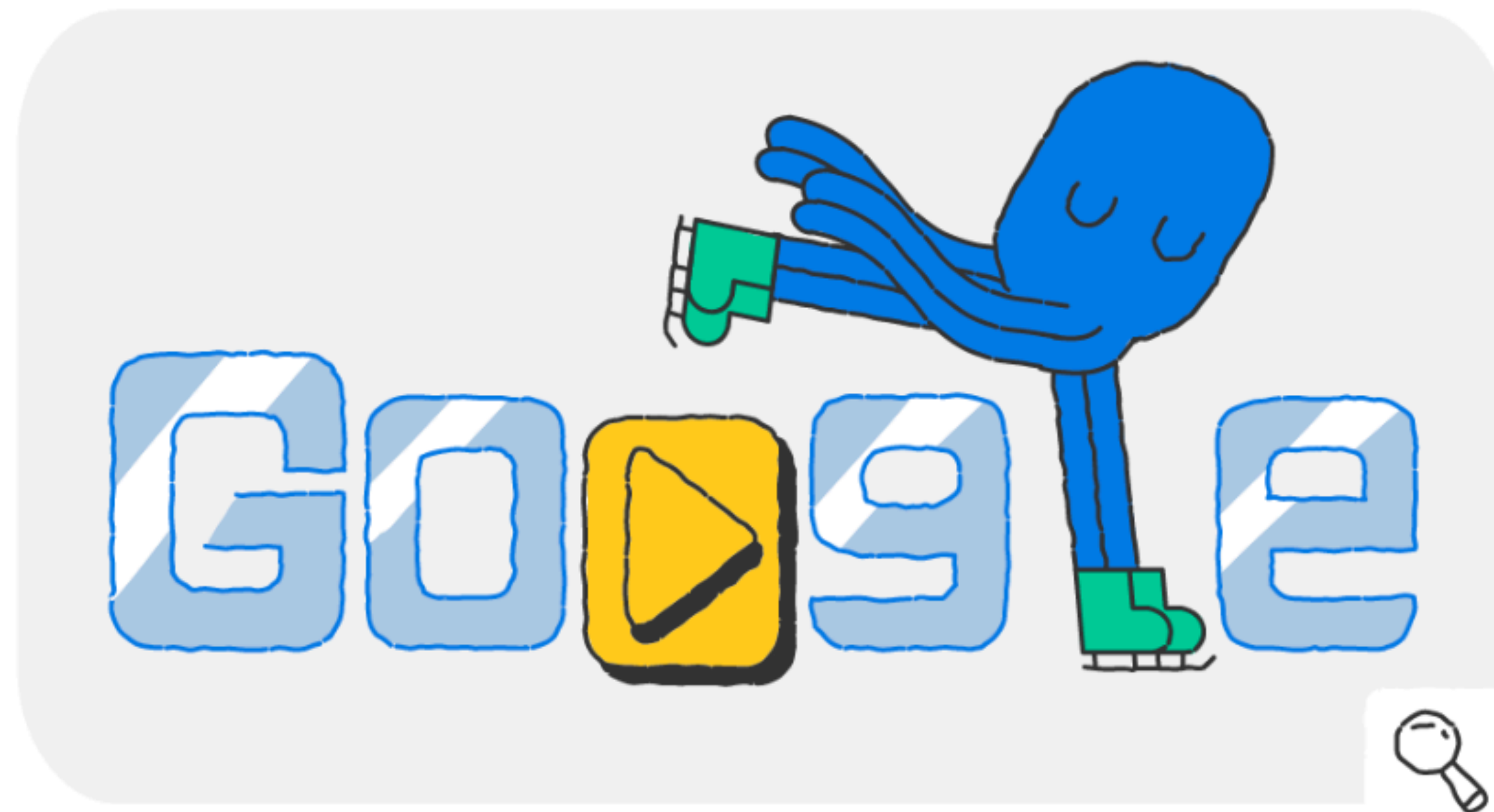
SCALE BY REQUEST





COLD START WILL HAPPEN
ONCE FOR EACH
CONCURRENT EXECUTION
OF YOUR FUNCTION

WHAT CAN YOU DO?



Please help me with aws lambda cold start

Google Search

I'm Feeling Lucky

NO SLA

NO HARD DETAILS
JUST GUIDELINES

PACKAGE SIZE?

Function name ▼	Description	Runtime ▼	Code size ▼
Fis2_0-Contract-DismissContract	Terminates a contract	Node.js 6.10	51.7 MB
Fis2_0-Contract-GetContractByEmployee	Get contract for an employee	Node.js 6.10	51.7 MB
Fis2_0-Employees-GetEmployees	Get all known employees	Node.js 6.10	23.9 MB
Fis2_0-Employees-GetEmployee	Get employee by id	Node.js 6.10	23.9 MB
Fis2_0-Employees-GetContractByEmployee	Get contract for an employee	Node.js 6.10	23.9 MB

**NO OBVIOUS
CORRELATION! ?**

MEMORY SETTINGS?

Memory (MB) [Info](#)

Your function is allocated CPU proportional to the memory configured.



128 MB

/AWS/LAMBDA/...REPORT

REQUESTID:

DURATION: 3208.99 MS

BILLED DURATION: 3300 MS

MEMORY SIZE: 128 MB

MAX MEMORY USED: 59 MB

/AWS/LAMBDA/...REPORT

REQUESTID:

DURATION: 3208.99 MS

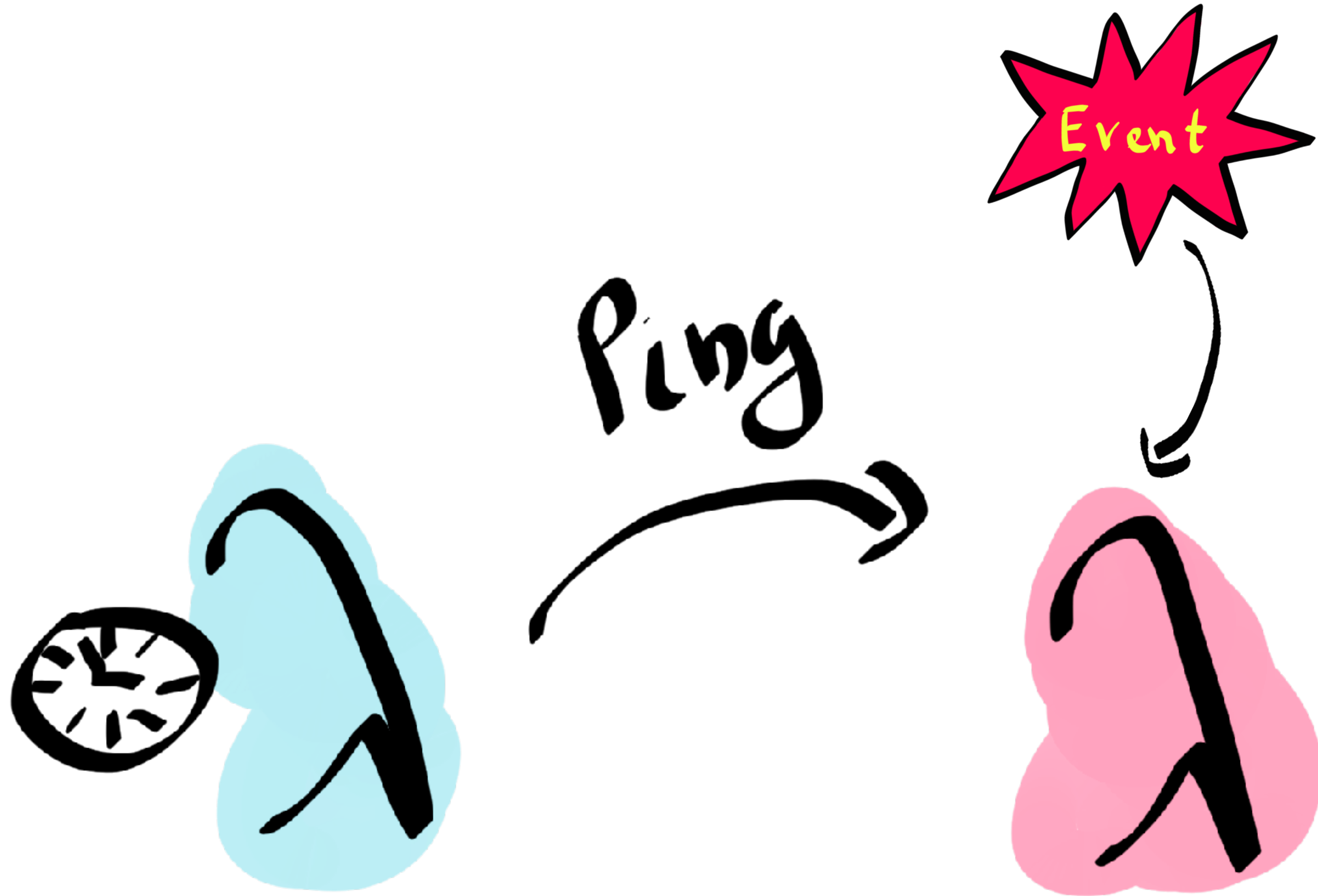
BILLED DURATION: 3300 MS

MEMORY SIZE: 128 MB

MAX MEMORY USED: 59 MB

MAYBE?

WARMUP – PING?



**INVASIVE
SCALE BY REQUEST
MONEY FOR NOTHING**

**AWS WILL KILL EVEN
WARM LAMBDA AFTER A
CERTAIN UNSPECIFIED
TIME FRAME**

MEASURE
CHANGE
MEASURE

MEASURE USER EXPERIENCE

DESIGN WITH LATENCY IN MIND



SUMMARY

SERVERLESS

HYPE TRAIN

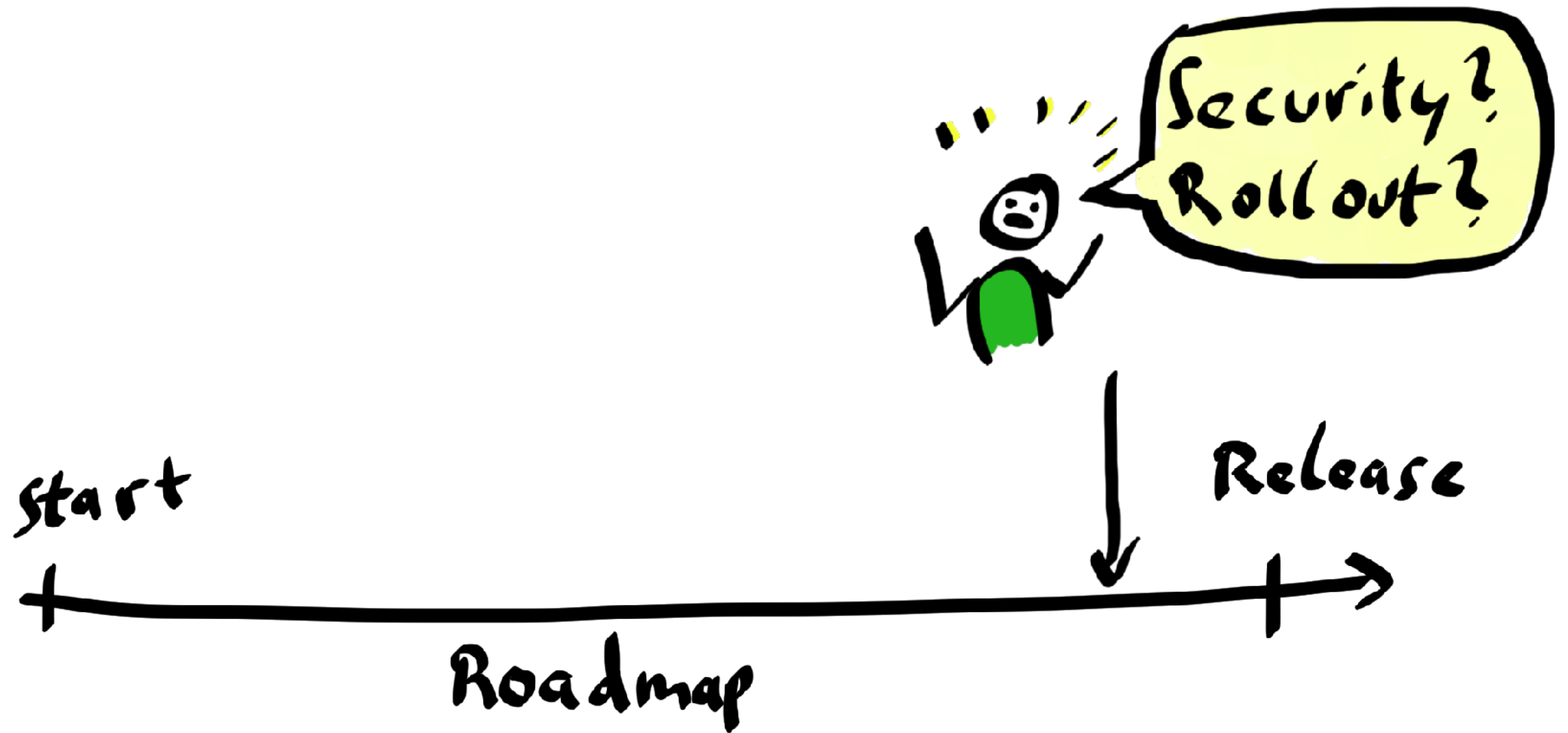
**DESIGNING DISTRIBUTED
SYSTEMS IS STILL HARD**

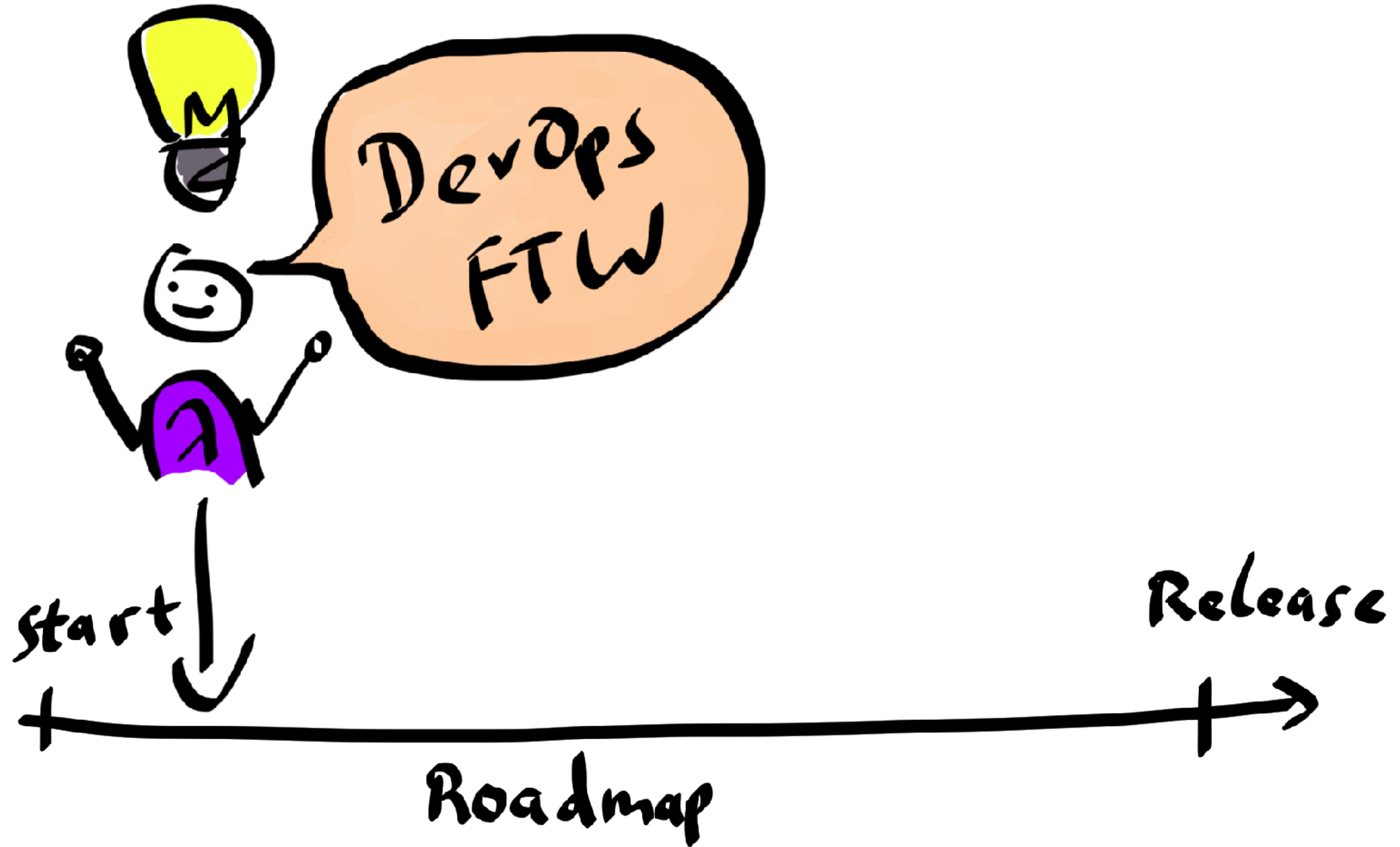
MOSTLY DIFFERENT CHALLENGES

**YOU NEED A CLOUD-SAVVY
OPERATIONS TEAM MEMBER**

DEVOPSSEC DONE RIGHT







TOOLING IS STILL EVOLVING

NOT A SILVER BULLET

SOMETIMES A SIMPLE
VM IS ENOUGH

SERVERLESS

LESS SERVER THAT WE
NEED TO TAKE CARE OF

**AND...SET BILLING
ALERTS, I MEAN IT**



THANK YOU!

**FEEDBACK? COMMENTS?
QUESTIONS?**

@KOENIGHOTZE