

WORLD SOFTWARE ARCHITECTURE CHAMPIONSHIP

ThoughtWorks®

# MICROSERVICES VS SERVERLESS

μ

FEBRUARY 28TH

LIVE ON PAY-PER-VIEW @OReillySACon

PRESENTED BY  
@javatarz  
@vvgomess

WSAC

# THE CHAMPION

---

*Microservices: The Software Architecture Superstar*

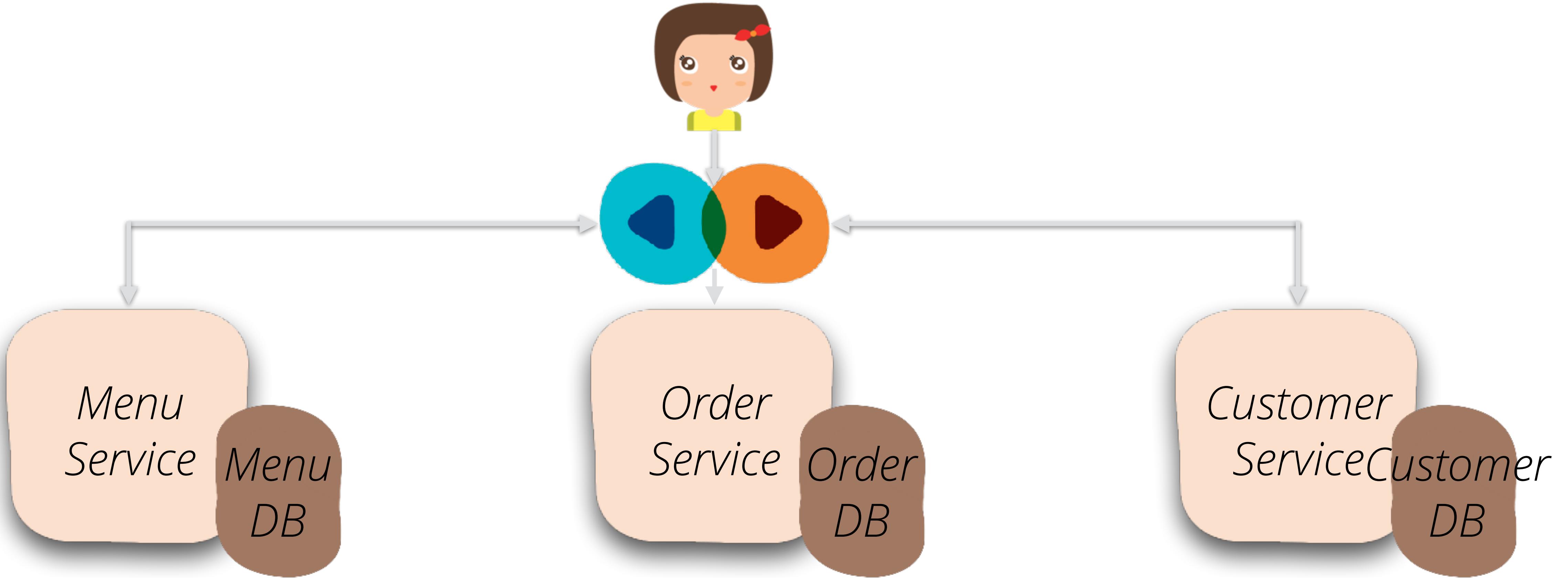


# REIGNING CHAMPION

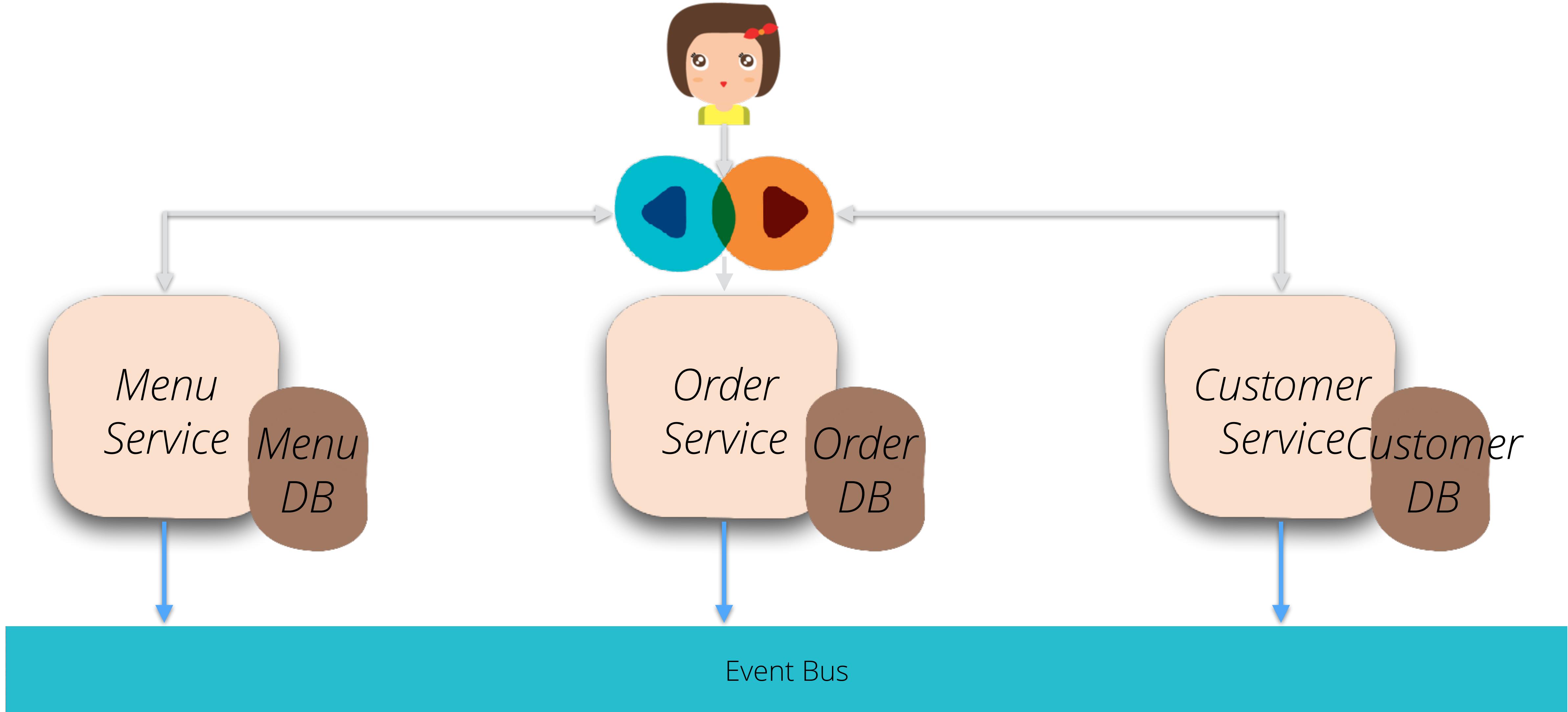
---



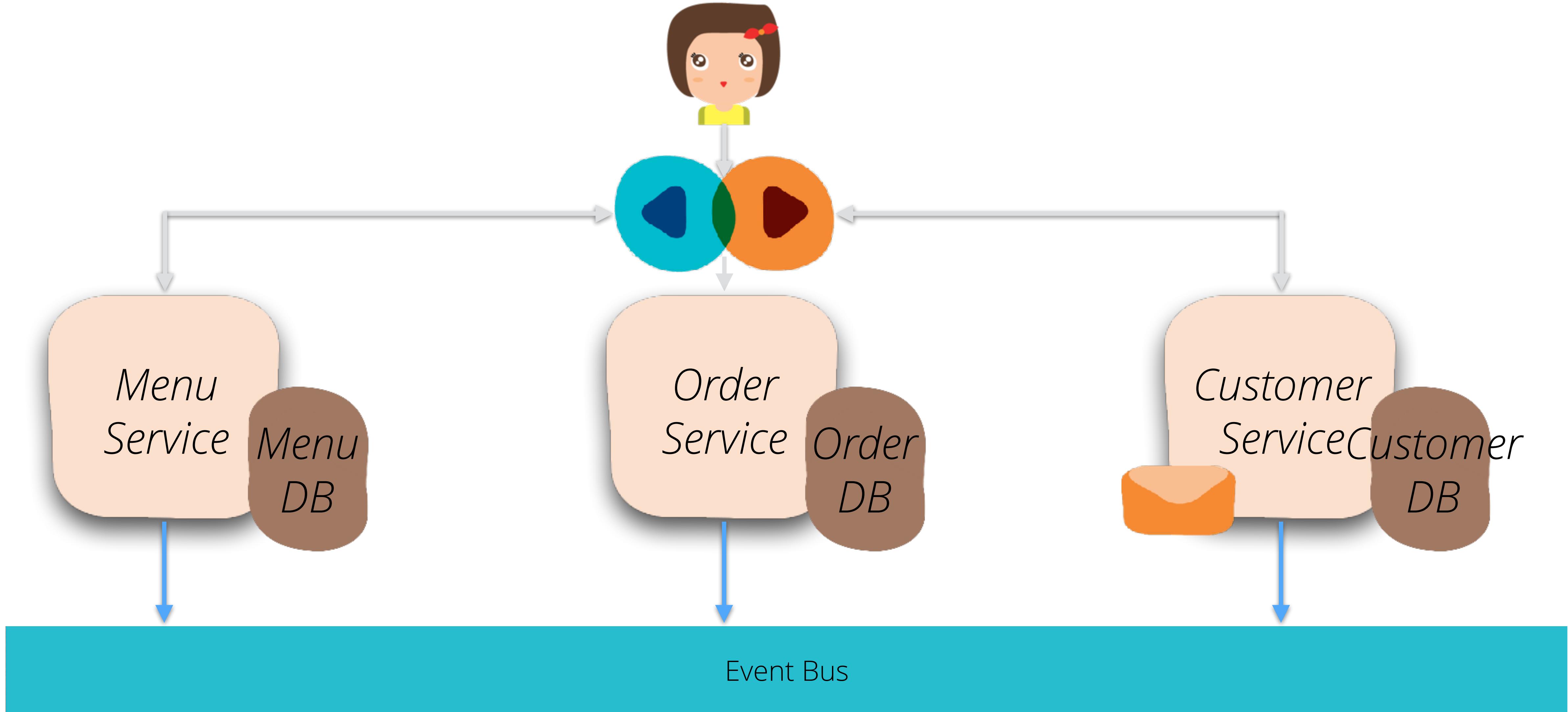
# REIGNING CHAMPION



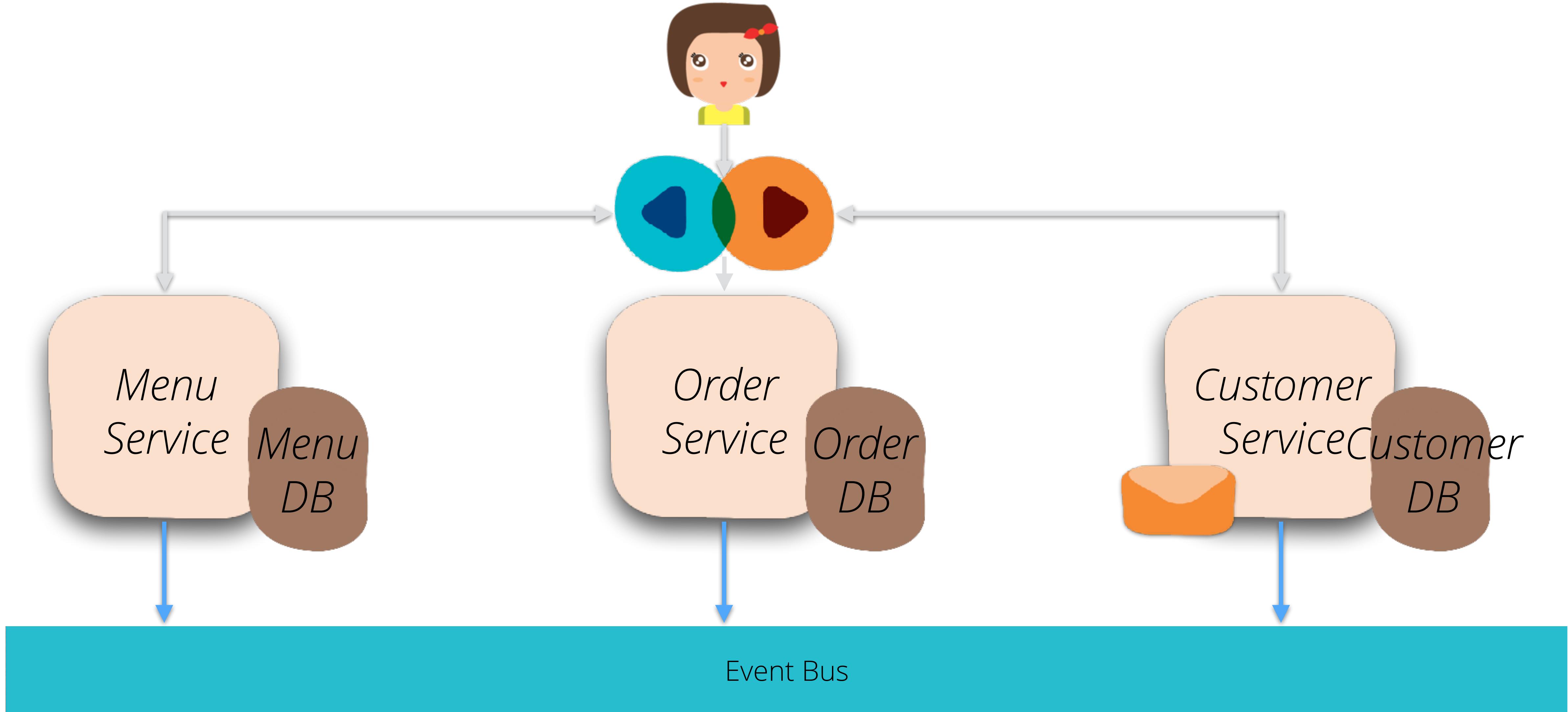
# REIGNING CHAMPION



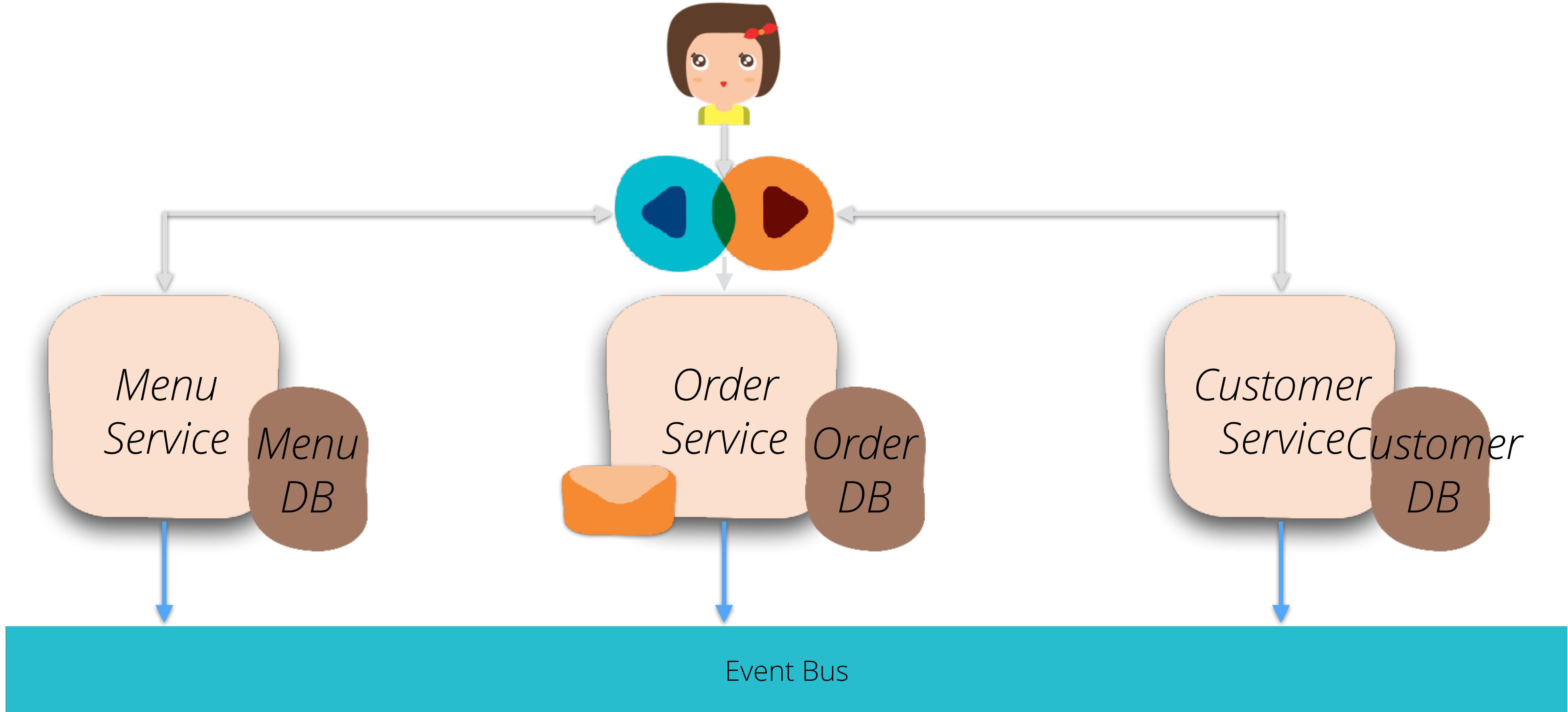
# REIGNING CHAMPION



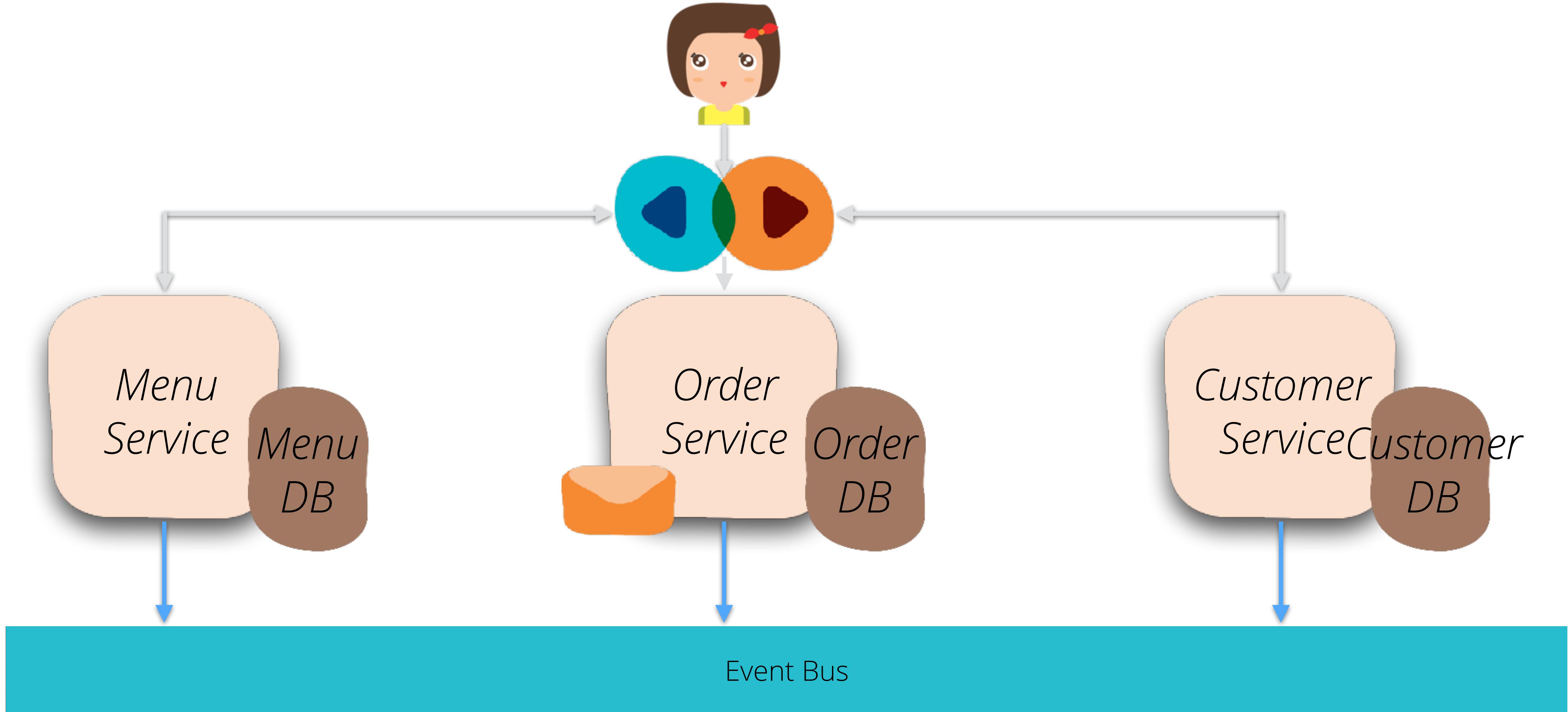
# REIGNING CHAMPION



# REIGNING CHAMPION



# REIGNING CHAMPION



# REIGNING CHAMPION

---



*Order  
Service*

# WHAT'S IN A MICROSERVICE APPLICATION

---



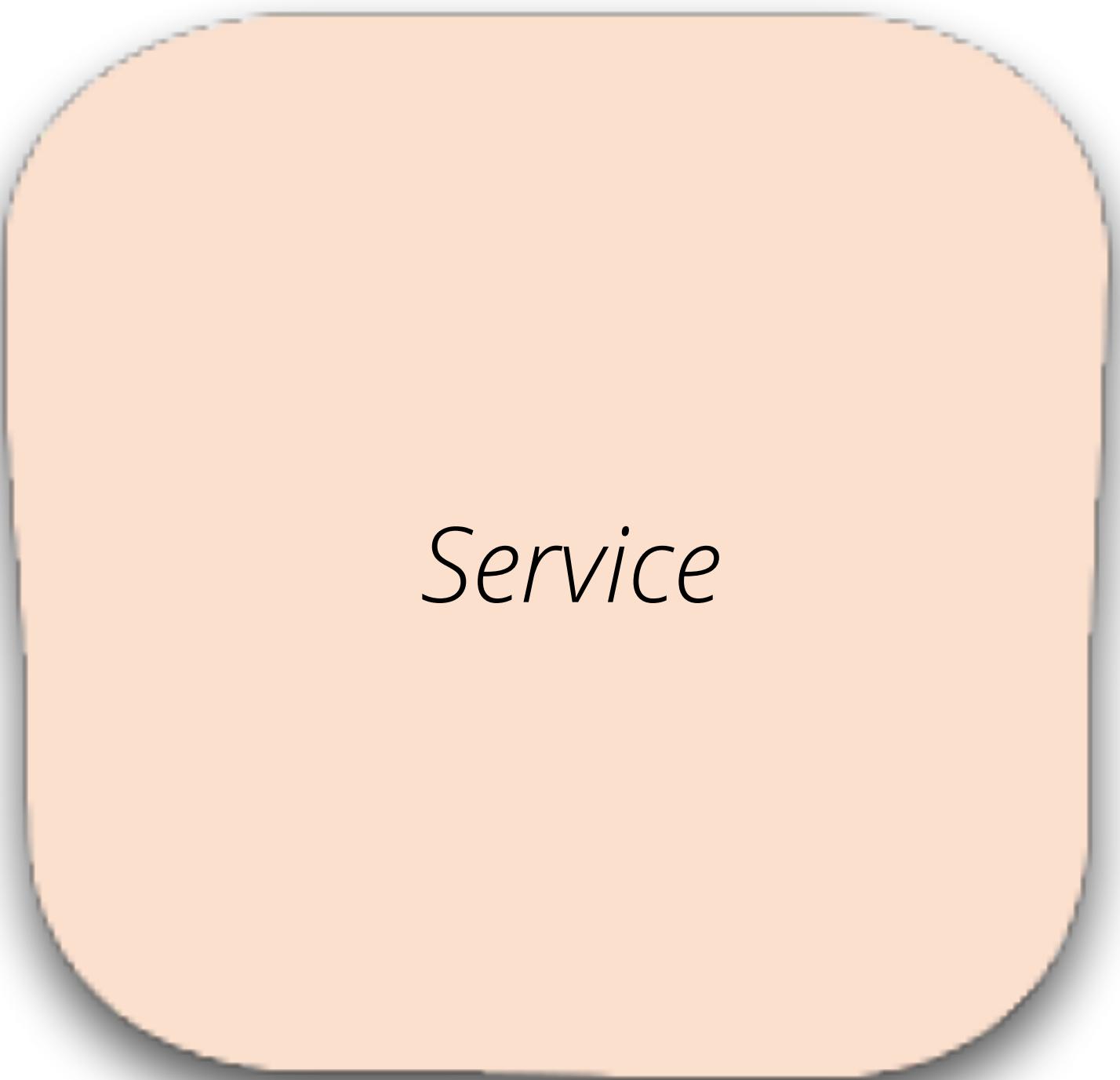
# WHAT'S IN A MICROSERVICE APPLICATION

---



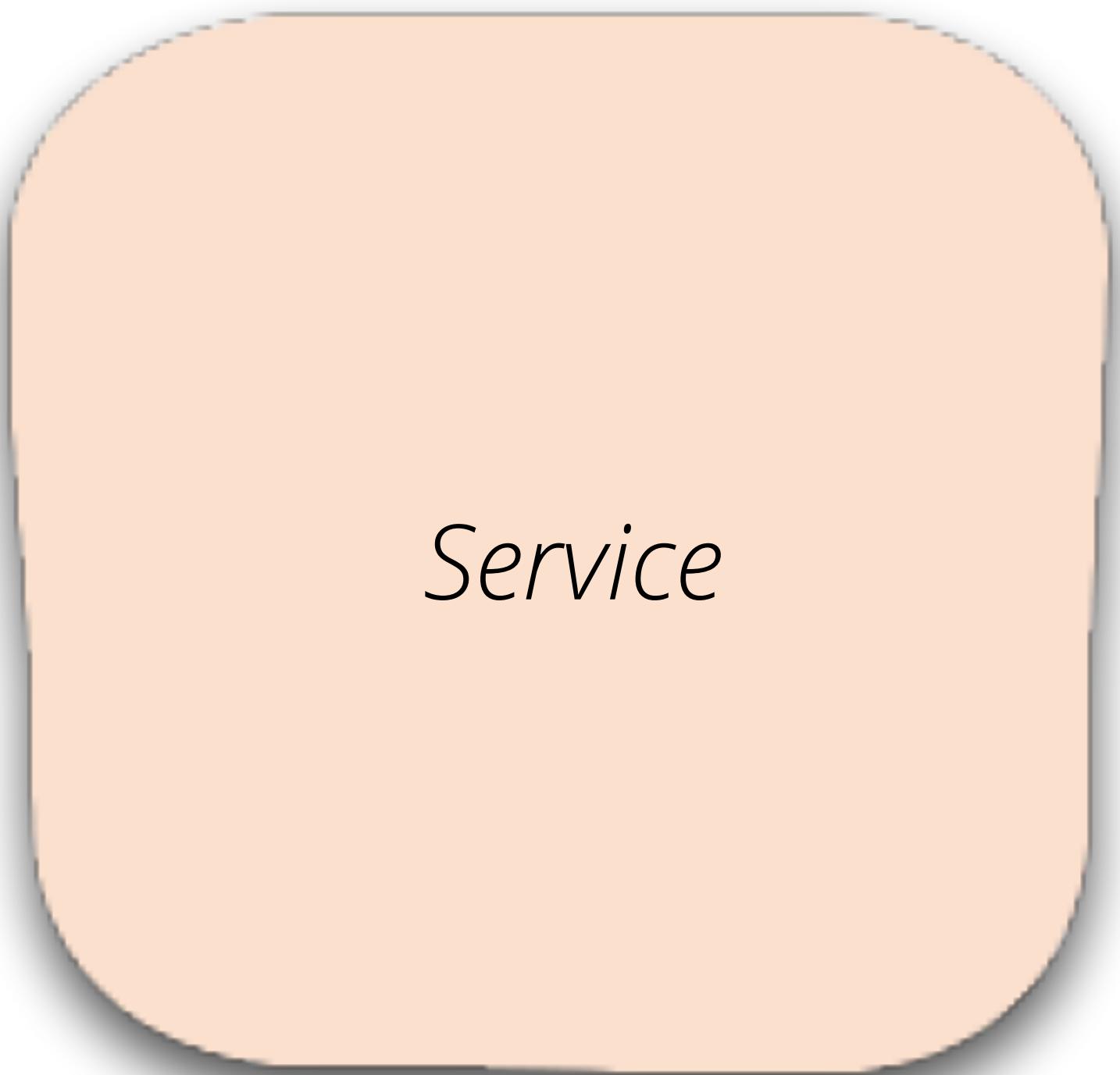
# WHAT'S IN A MICROSERVICE APPLICATION

---



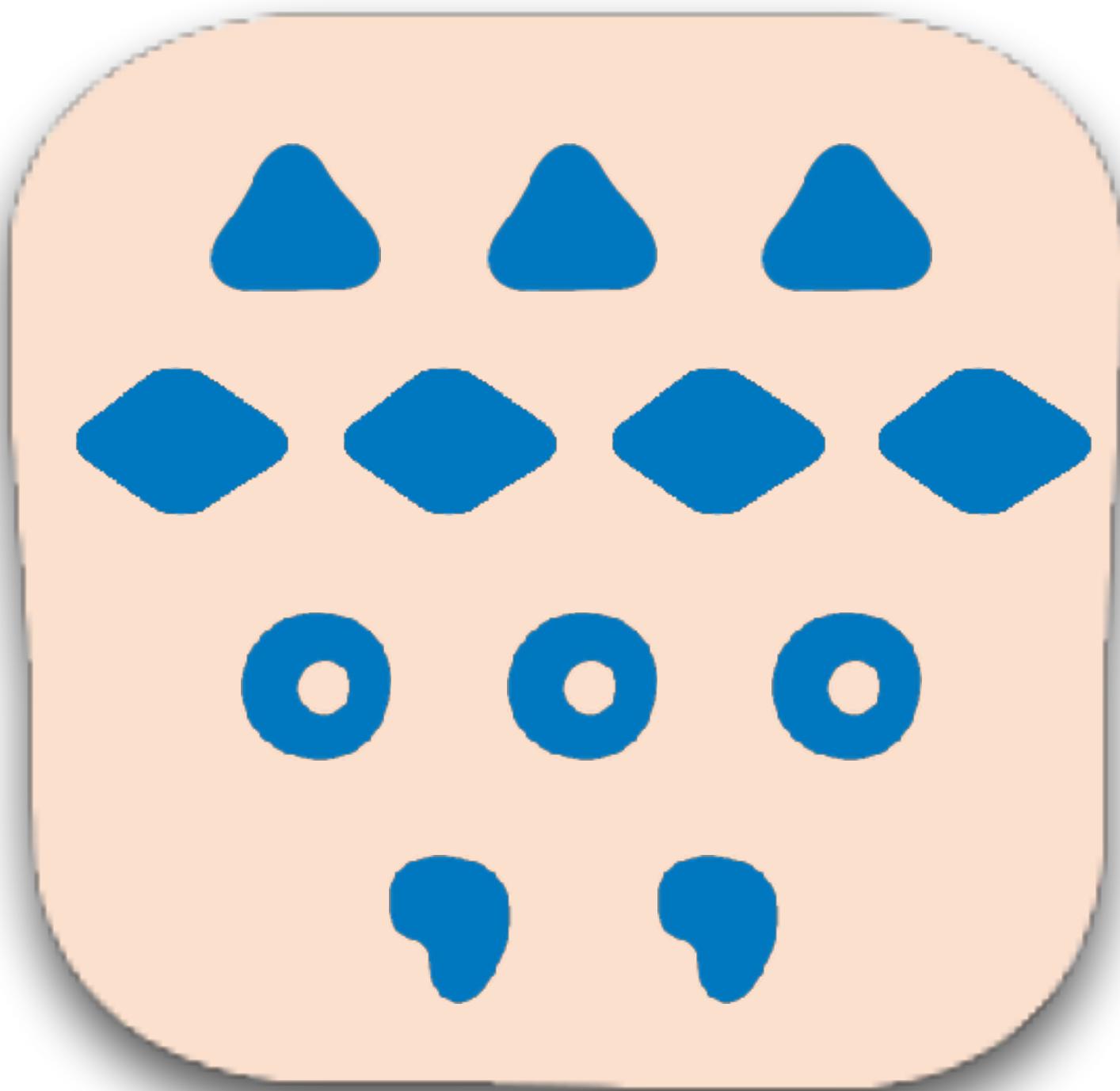
# WHAT'S IN A MICROSERVICE APPLICATION

---



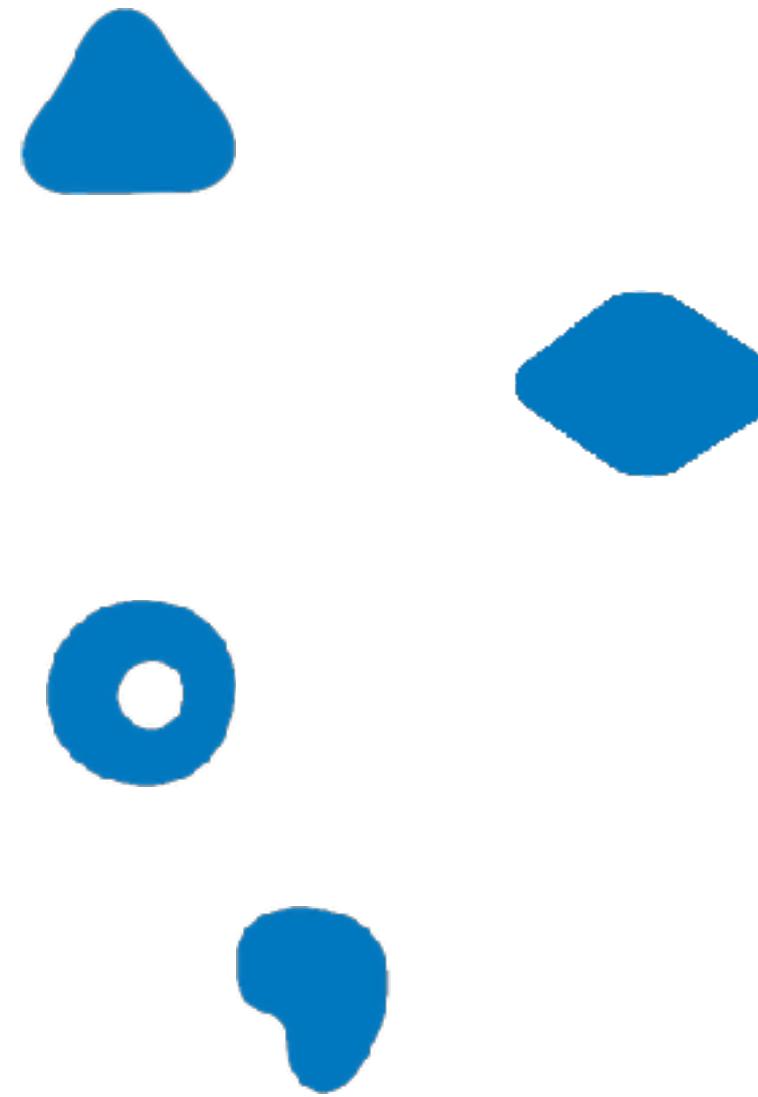
# WHAT'S IN A MICROSERVICE APPLICATION

---



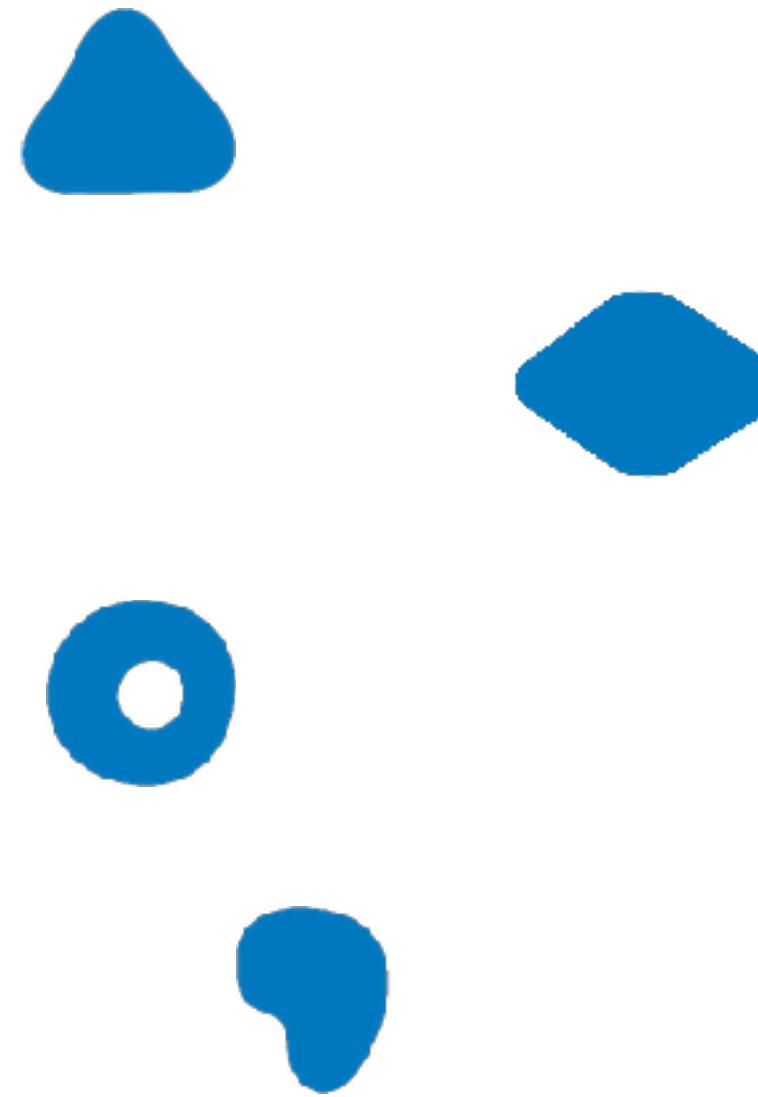
# WHAT'S IN A MICROSERVICE APPLICATION

---



# WHAT'S IN A MICROSERVICE APPLICATION

---



# WHAT'S IN A MICROSERVICE APPLICATION

---



*HTTP Controllers*



*Command Handlers*



*Event Handler*



*Aggregate*

# WHAT'S IN A MICROSERVICE APPLICATION

---



*HTTP Controllers*



*Command Handlers*



*Event Handlers*



*Aggregates*

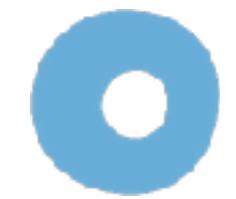
# WHAT'S IN A MICROSERVICE APPLICATION



*HTTP Controllers*



*Command Handlers*



*Event Handlers*



*Aggregates*

```
@PostMapping("/add")
public Future<?> addMenuItem(@RequestBody MenuItemRequestBody body) {
    String id = randomUUID().toString();
    String description = body.getDescription();
    BigDecimal price = body.getPrice();

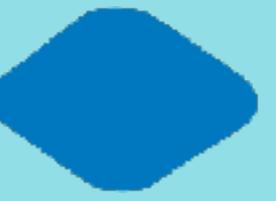
    FutureCallback<AddMenuItemCommand, Object> callback = new FutureCallback<>();
    commander.send(new AddMenuItemCommand(id, description, price), callback);

    return callback
        .thenApply(v → links.linkForSingleResource(Item.class, id)
            .withSelfRel()
            .getHref())
        .toCompletableFuture();
}
```

# WHAT'S IN A MICROSERVICE APPLICATION



*HTTP Controllers*



*Command Handlers*



*Event Handlers*



*Aggregates*

```
@CommandHandler
public MenuItem(AddMenuItemCommand command) {
    Assert.isFalse(
        command.getDescription().isEmpty(),
        () -> "Description must be present.");
    apply(new MenuItemAddedEvent(command.getId(),
        command.getDescription(), command.getPrice()));
}
```

# WHAT'S IN A MICROSERVICE APPLICATION

---



*HTTP Controllers*



*Command Handlers*



*Event Handlers*

```
@EventHandler  
public void on(MenuItemAddedEvent event) {  
    items.save(new Item(event.getId(), event.getDescription(), event.getPrice()));  
}
```



*Aggregates*

# WHAT'S IN A MICROSERVICE APPLICATION

---



*HTTP Controllers*



*Command Handlers*



*Event Handlers*



*Aggregates*

```
@Aggregate  
@FieldDefaults(level = PRIVATE)  
public class MenuItem {  
    @AggregateIdentifier  
    String id;  
    Boolean active;  
}
```

# WHAT'S IN A MICROSERVICE APPLICATION

---



*HTTP Controllers*



*Command Handlers*



*Event Handlers*



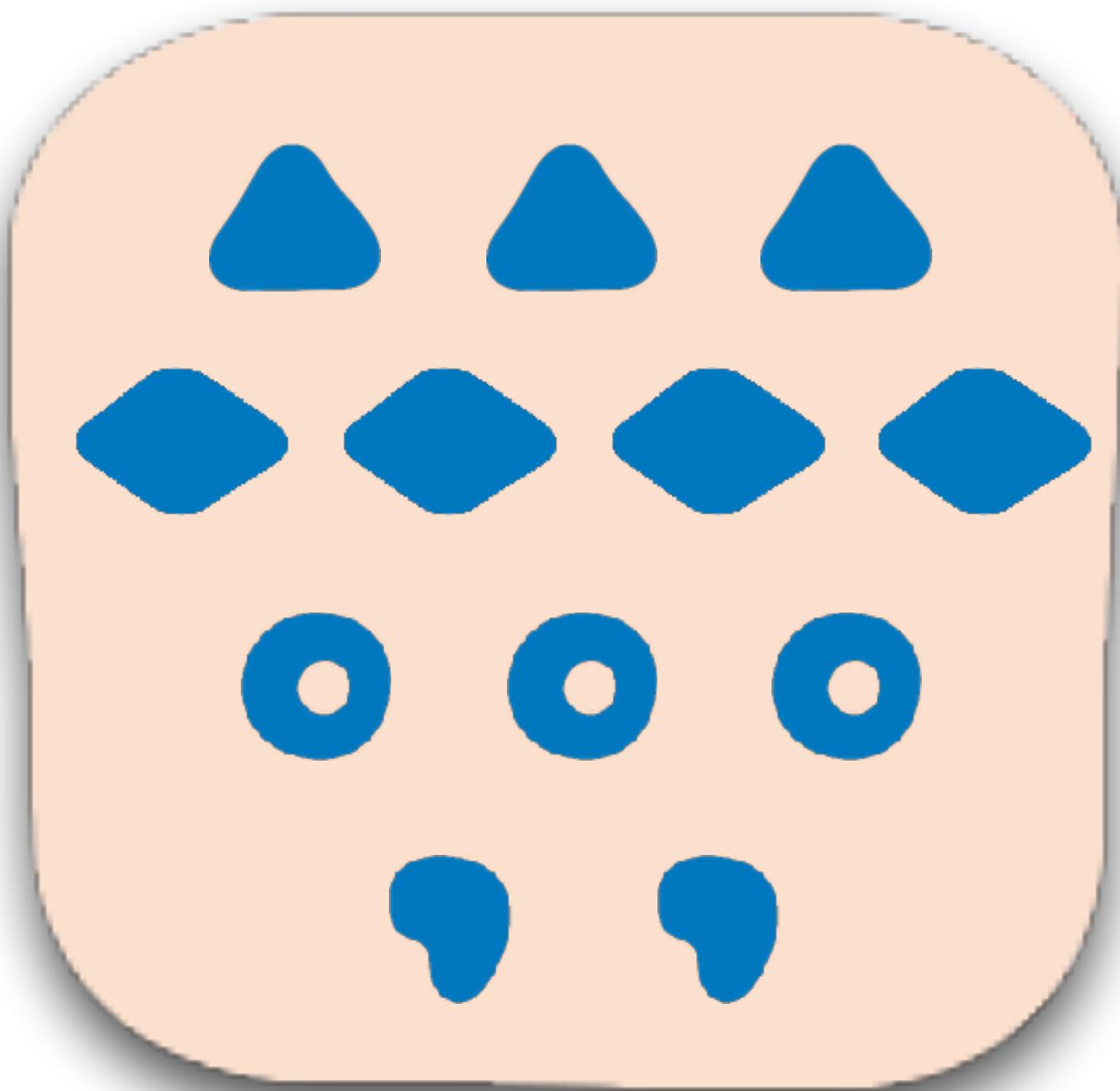
*Aggregates*

```
@Aggregate  
@FieldDefaults(level = PRIVATE)  
public class MenuItem {  
    @AggregateIdentifier  
    String id;  
    Boolean active;  
}
```

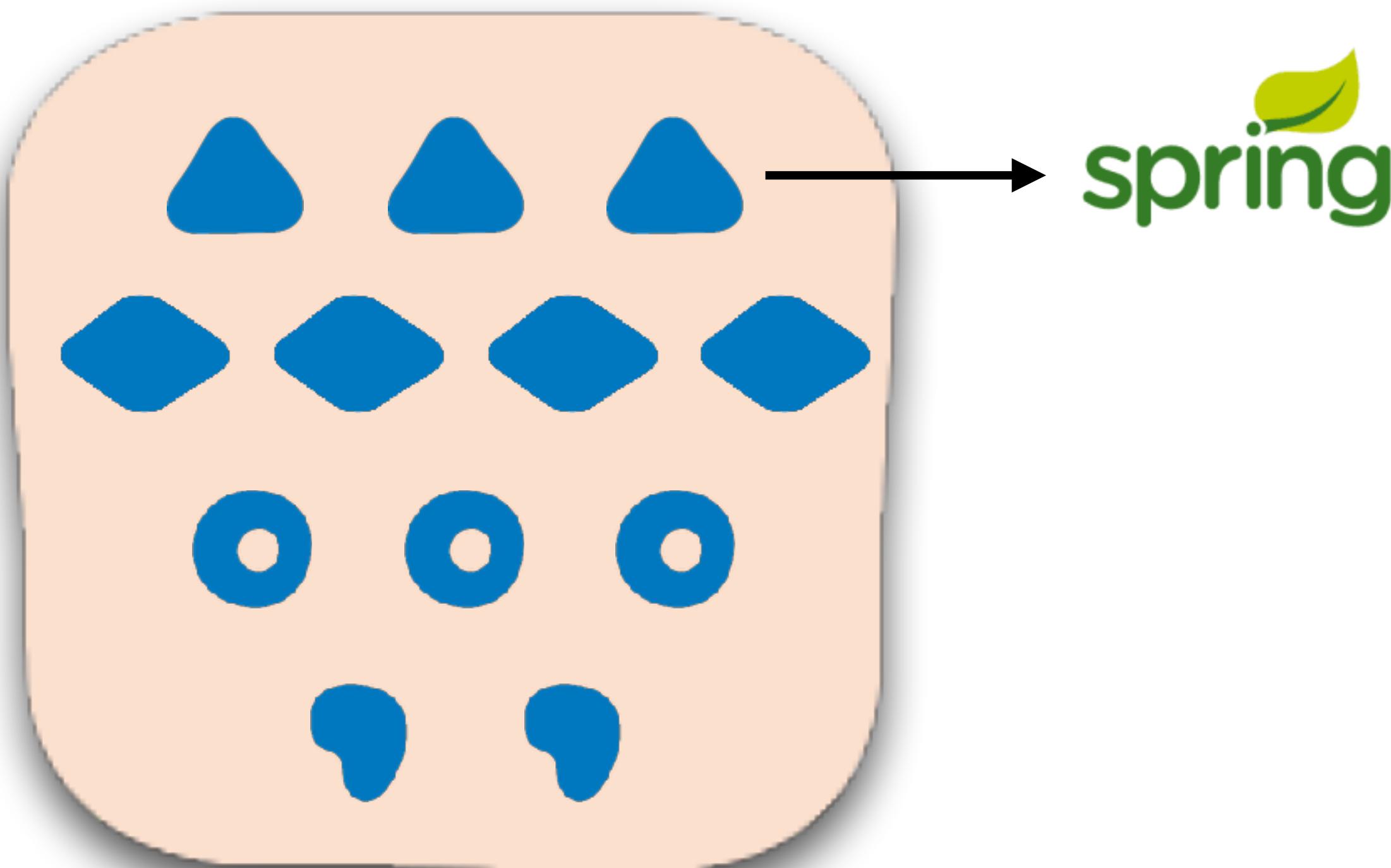
Source Code: [bit.ly/ms-ed-restaurant](https://bit.ly/ms-ed-restaurant)

# WHAT'S IN A MICROSERVICE APPLICATION?

---

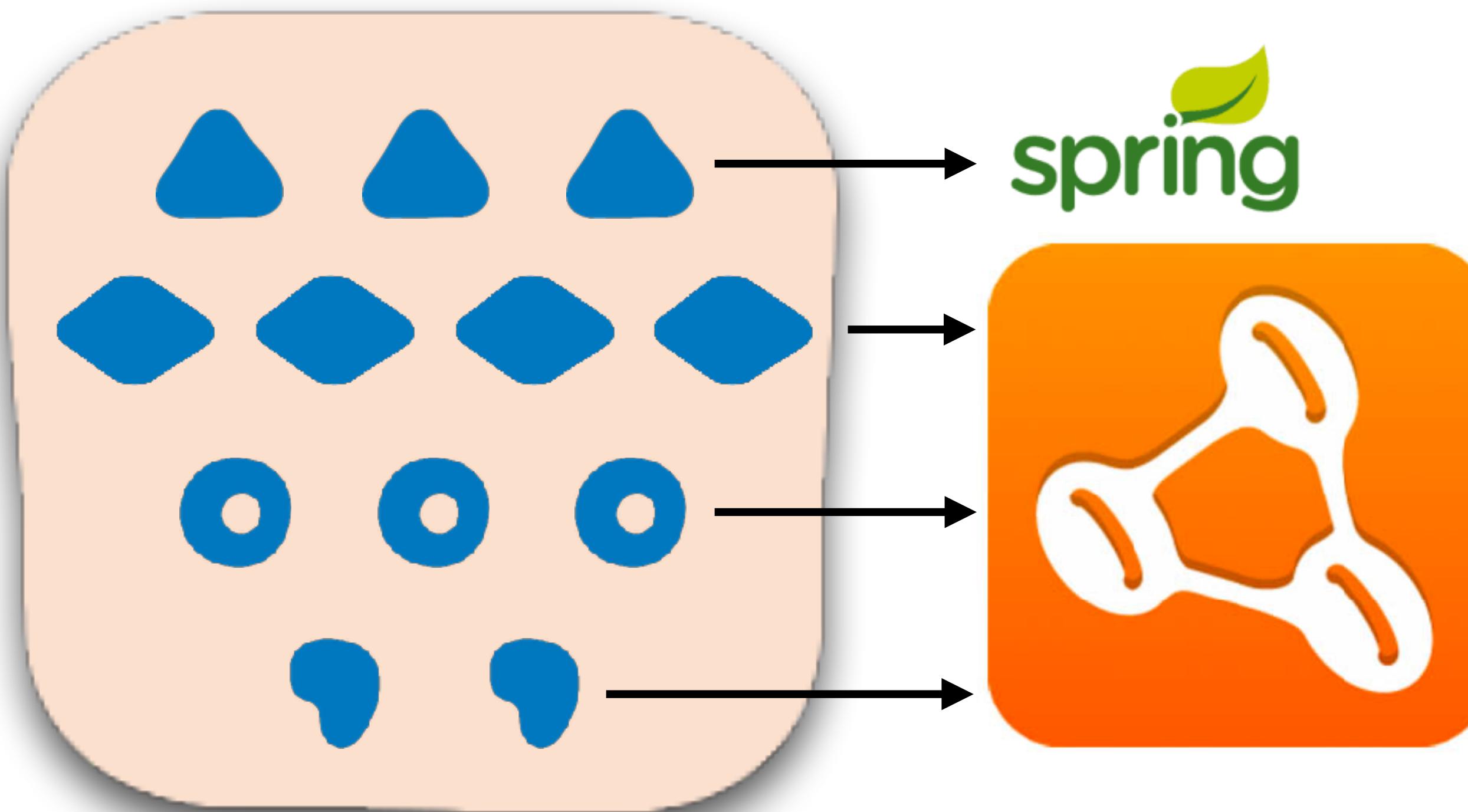


# WHAT'S IN A MICROSERVICE APPLICATION?



spring

# WHAT'S IN A MICROSERVICE APPLICATION?



# THE CHALLENGER

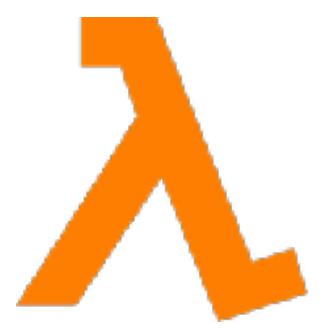
---

*Serverless: The New Kid on the block*



# WHAT IS SERVERLESS?

---



# WHAT IS SERVERLESS?

---



# WHAT IS SERVERLESS?

---



*Backend as a Service (BaaS)*

# WHAT IS SERVERLESS?

---



*Backend as a Service (BaaS)*

+

*Functions as a Service (FaaS)*

# WHAT IS SERVERLESS?

---



*Backend as a Service (BaaS)*

+

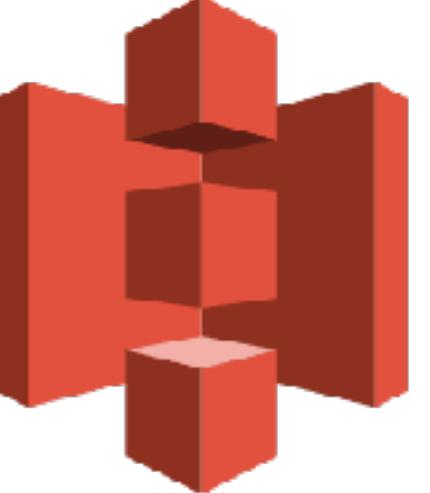
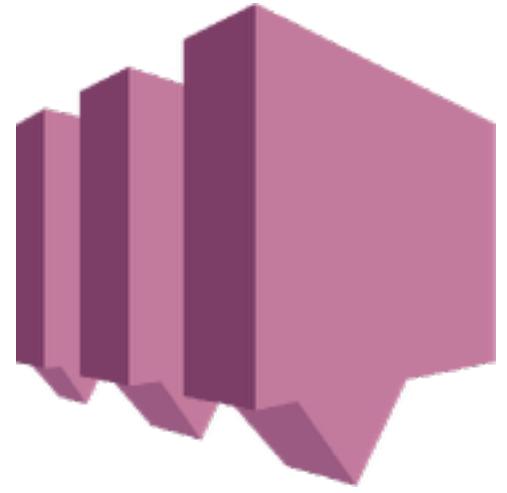
*Functions as a Service (FaaS)*

# WHAT IS SERVERLESS?

---



*Backend as a Service (BaaS)*



+

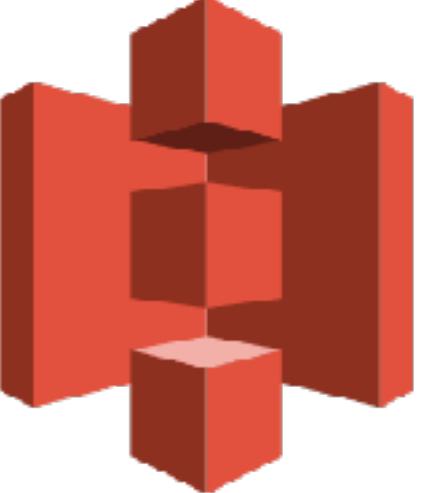
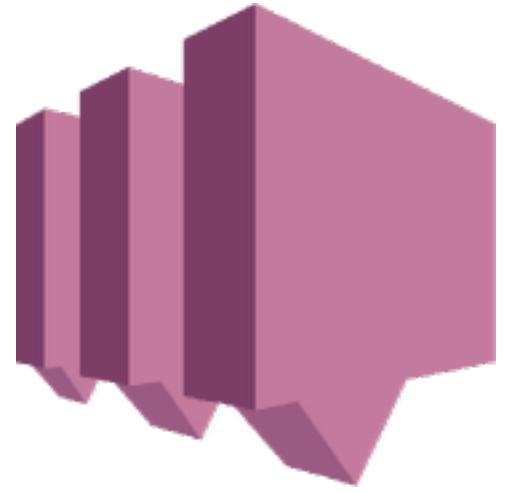
*Functions as a Service (FaaS)*

# WHAT IS SERVERLESS?

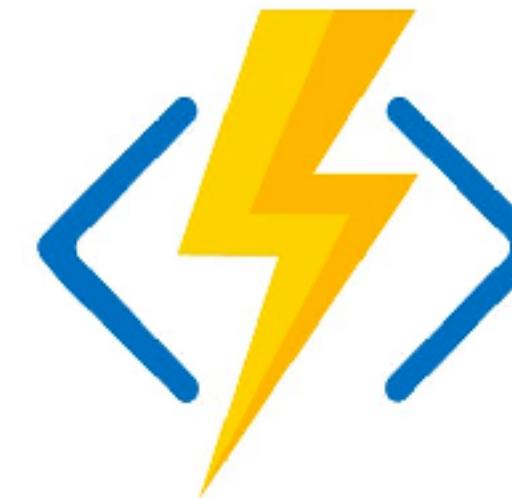
---



*Backend as a Service (BaaS)*



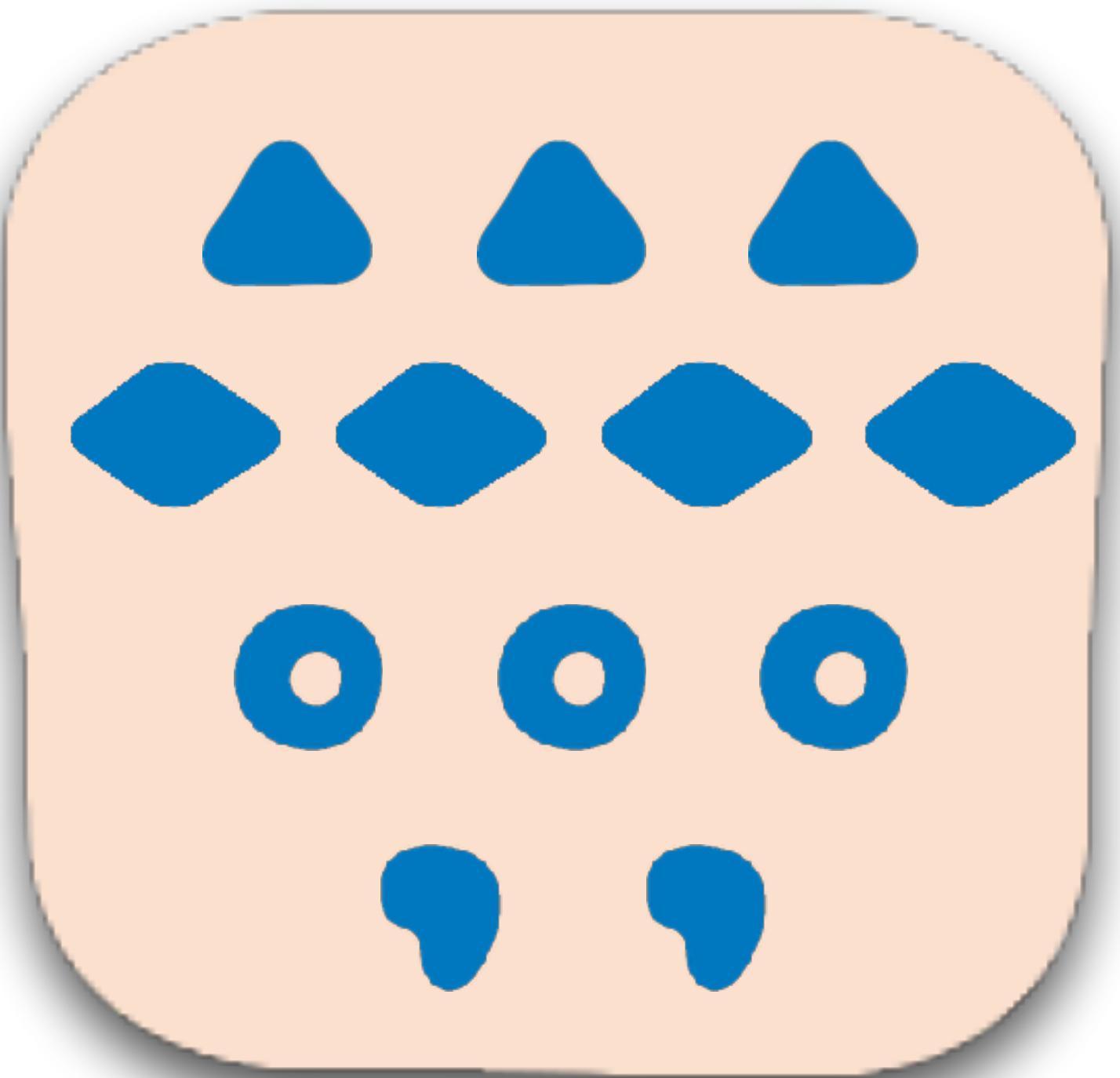
+



*Functions as a Service (FaaS)*

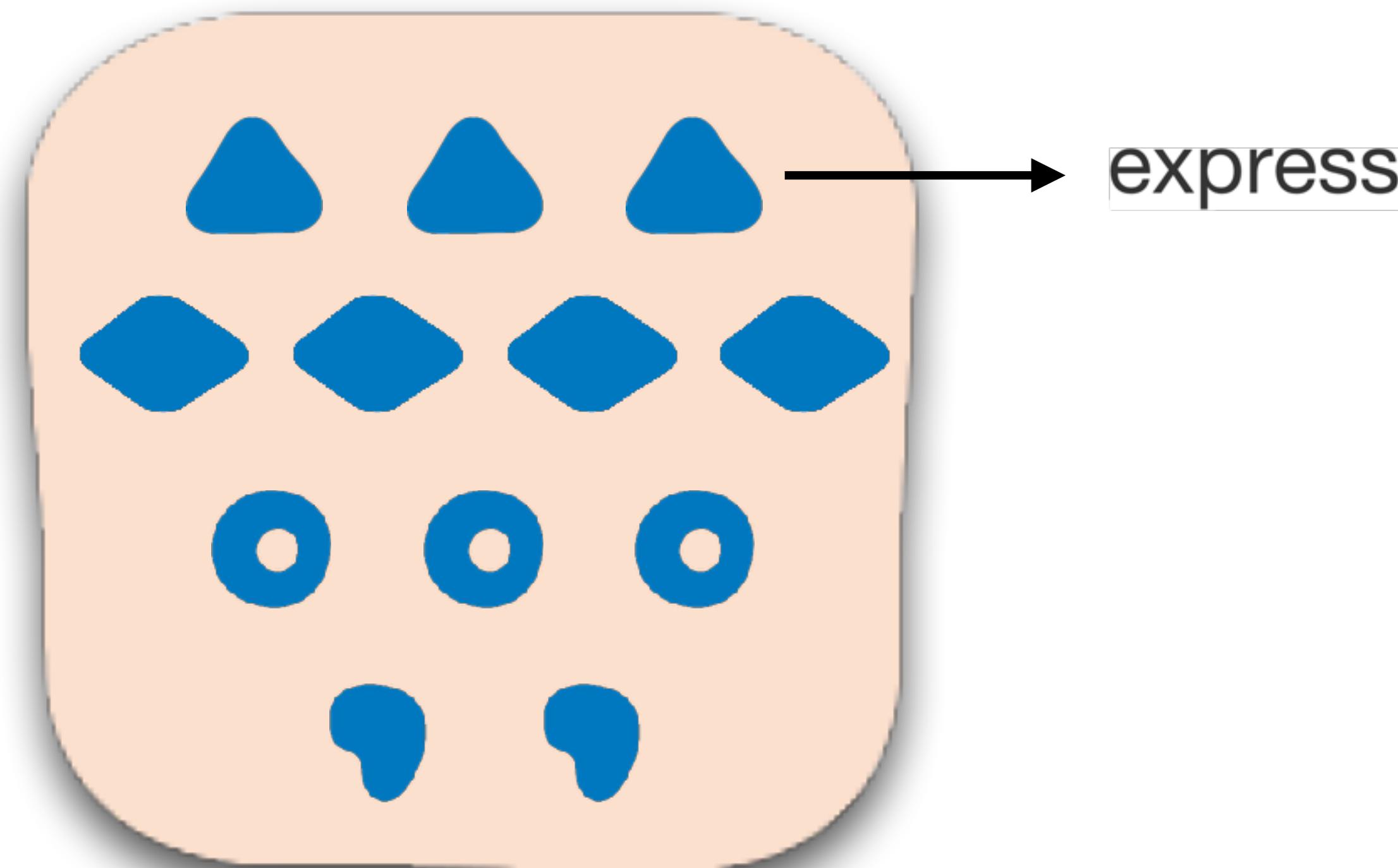
# WHAT'S IN A SERVERLESS APPLICATION?

---



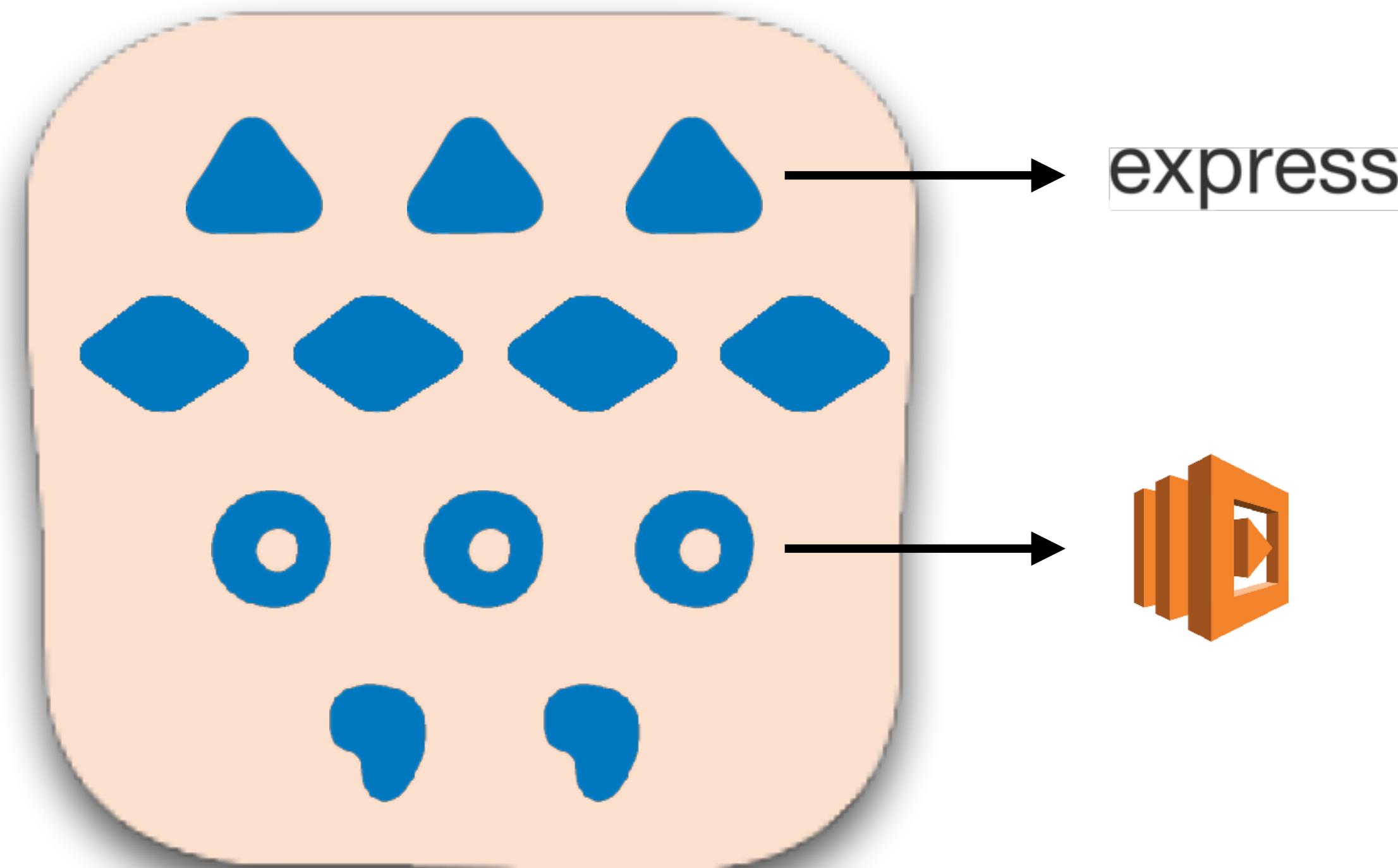
# WHAT'S IN A SERVERLESS APPLICATION?

---

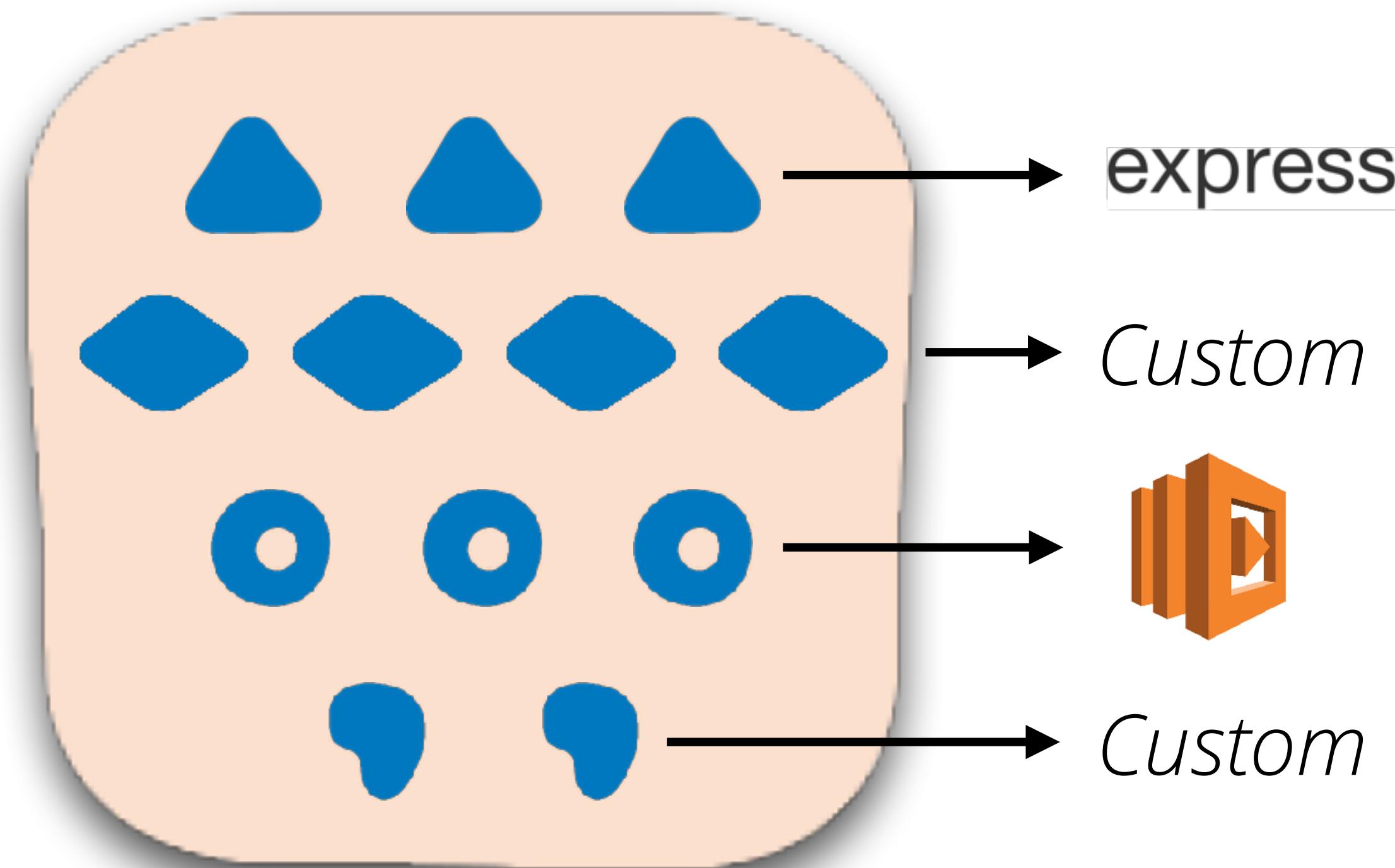
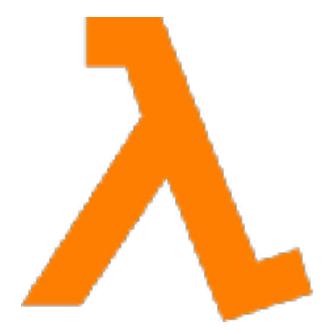


# WHAT'S IN A SERVERLESS APPLICATION?

---

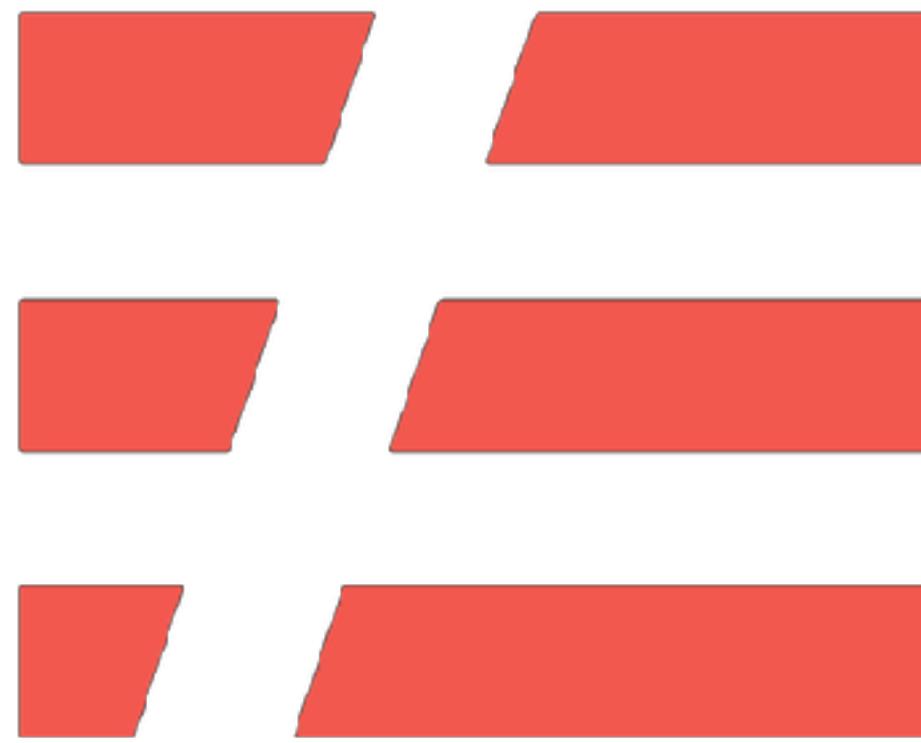


# WHAT'S IN A SERVERLESS APPLICATION?



# SERVERLESS FRAMEWORK

---



serverless



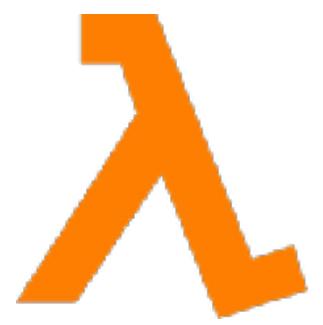
## *Vendor Agnostic FaaS Configuration*

```
functions:
  addMenuItem:
    handler: api.handler
    events:
      - http: 'POST /menu/items/commands/add'
  getMenuItem:
    handler: api.handler
    events:
      - http: 'GET /menu/items/{proxy+}'
  getMenuItems:
    handler: api.handler
    events:
      - http: 'GET /menu/items/'
  persistEvent:
    handler: listeners.persistEvent
    events:
      - sns:
          topicName: menu-events
          displayName: "Menu domain events"
  persistMenuItem:
    handler: listeners.persistMenuItem
    events:
      - sns:
          topicName: menu-events
          displayName: "Menu domain events"
```



## *Vendor Specific Deployment Configuration*

```
provider:
  name: aws
  runtime: nodejs6.10
  stage: dev
  region: us-east-1
  iamRoleStatements:
    - Effect: Allow
      Action:
        - dynamodb:Query
        - dynamodb:Scan
        - dynamodb:GetItem
        - dynamodb:PutItem
        - dynamodb:UpdateItem
        - dynamodb:DeleteItem
        - "sns:*"
      Resource:
        - { "Fn :: GetAtt": [ "MenuEventsDynamoDBTable", "Arn" ] }
        - { "Fn :: GetAtt": [ "MenuMenuItemsDynamoDBTable", "Arn" ] }
        - "arn:aws:sns:us-east-1:${ssm:accountId}:menu-events"
  environment:
    TOPIC_NAME: 'menu-events'
    AWS_ACCOUNT_ID: '${ssm:accountId}'
    EVENTS_TABLE: 'menu-events-table-${self:provider.stage}'
    MENU_ITEMS_TABLE: 'menu-menu-items-table-${self:provider.stage}'
```



## *Vendor Specific Deployment Configuration*

```
provider:  
  name: aws  
  runtime: nodejs6.10  
  stage: dev  
  region: us-east-1  
  iamRoleStatements:  
    - Effect: Allow  
      Action:  
        - dynamodb:Query  
        - dynamodb:Scan  
        - dynamodb:GetItem  
        - dynamodb:PutItem  
        - dynamodb:UpdateItem  
        - dynamodb:DeleteItem  
        - "sns:*"  
      Resource:  
        - { "Fn :: GetAtt": [ "MenuEventsDynamoDBTable", "Arn" ] }  
        - { "Fn :: GetAtt": [ "MenuMenuItemsDynamoDBTable", "Arn" ] }  
        - "arn:aws:sns:us-east-1:${ssm:accountId}:menu-events"  
environment:  
  TOPIC_NAME: 'menu-events'  
  AWS_ACCOUNT_ID: '${ssm:accountId}'  
  EVENTS_TABLE: 'menu-events-table-${self:provider.stage}'  
  MENU_ITEMS_TABLE: 'menu-menu-items-table-${self:provider.stage}'
```

## *Vendor Specific Resource Configuration*

```
resources:  
  Resources:  
    MenuEventsDynamoDBTable:  
      Type: 'AWS :: DynamoDB :: Table'  
      Properties:  
        TableName: 'menu-events-table-${self:provider.stage}'  
      AttributeDefinitions:  
        -  
          AttributeName: id  
          AttributeType: S  
      KeySchema:  
        -  
          AttributeName: id  
          KeyType: HASH  
      ProvisionedThroughput:  
        ReadCapacityUnits: 1  
        WriteCapacityUnits: 1
```



## *Plugins for local development*

```
service: serverless-cqrs-menu

plugins:
  - serverless-offline-sns
  - serverless-dynamodb-local
  - serverless-offline #serverless-offline needs to be last in the list

custom:
  dynamodb:
    start:
      migrate: true
  serverless-offline-sns:
    port: 4002
    debug: true
  serverless-offline:
    babelOptions:
      presets: ["env"]
```

A photograph of a boxing ring. In the center, a referee in a black shirt and khaki pants stands with arms crossed. Around him are several people, including two men in white shirts and dark trousers who appear to be judges. The ring floor is blue with white ropes. The ropes are decorated with the word "BELLATOR" in large letters and the Corona logo. In the background, spectators are visible in the stands.

# THE JUDGES

---

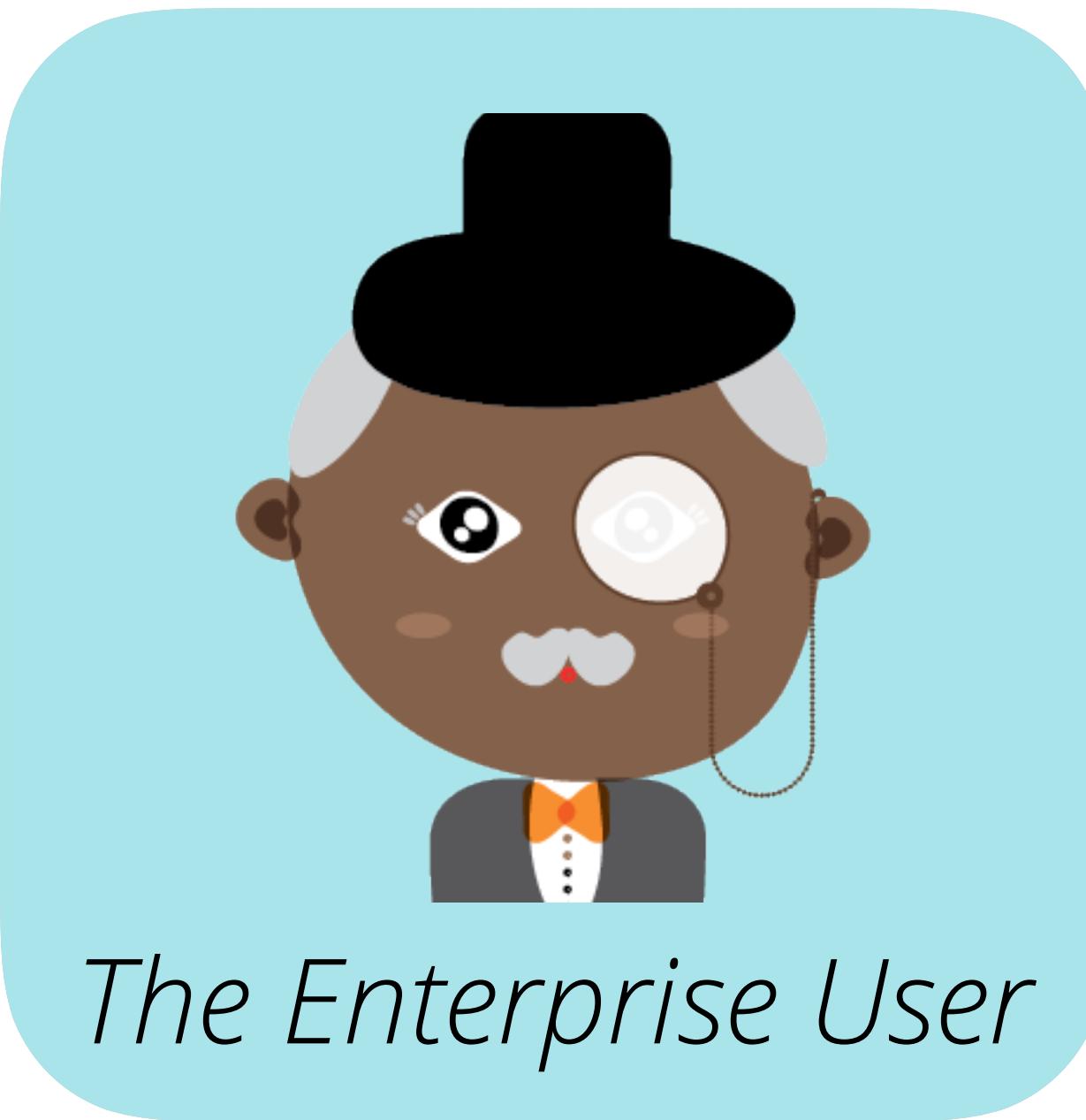
*How do we evaluate these frameworks?*

# THE JUDGES

---

# THE JUDGES

---

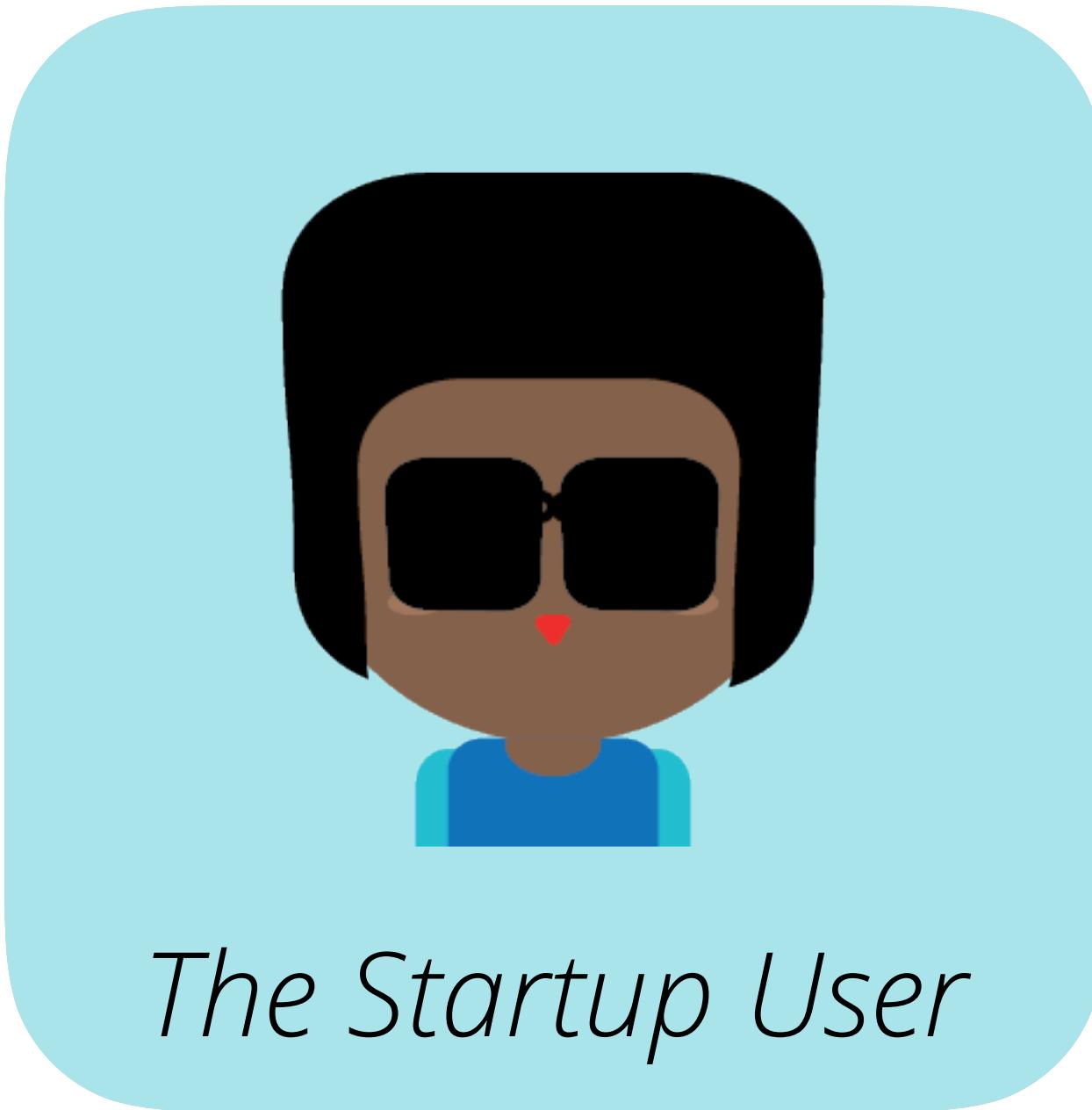


## THE JUDGES

---



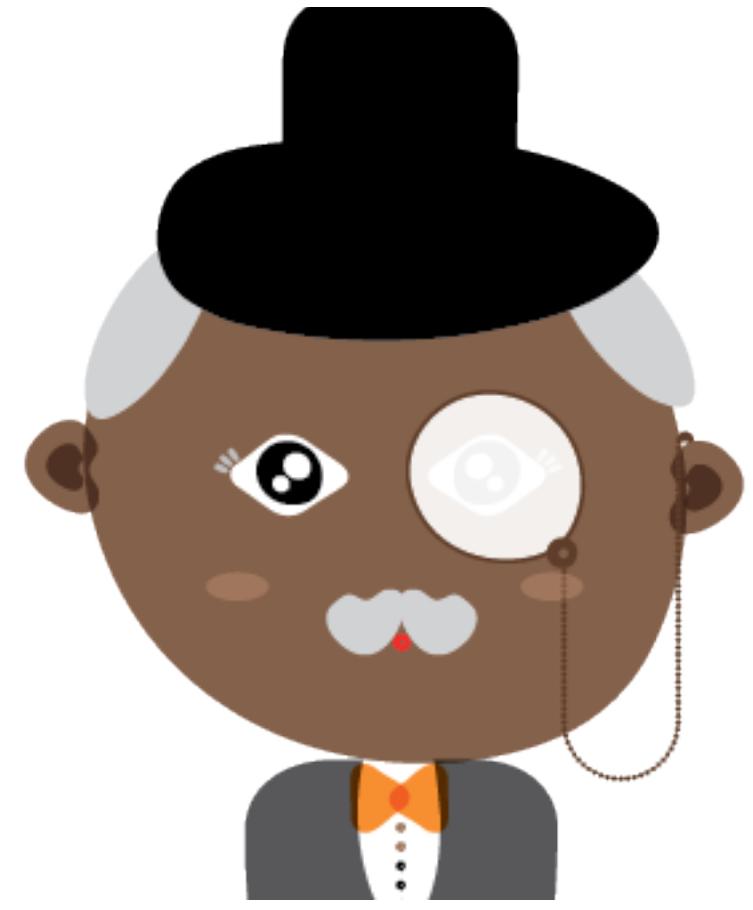
*The Enterprise User*



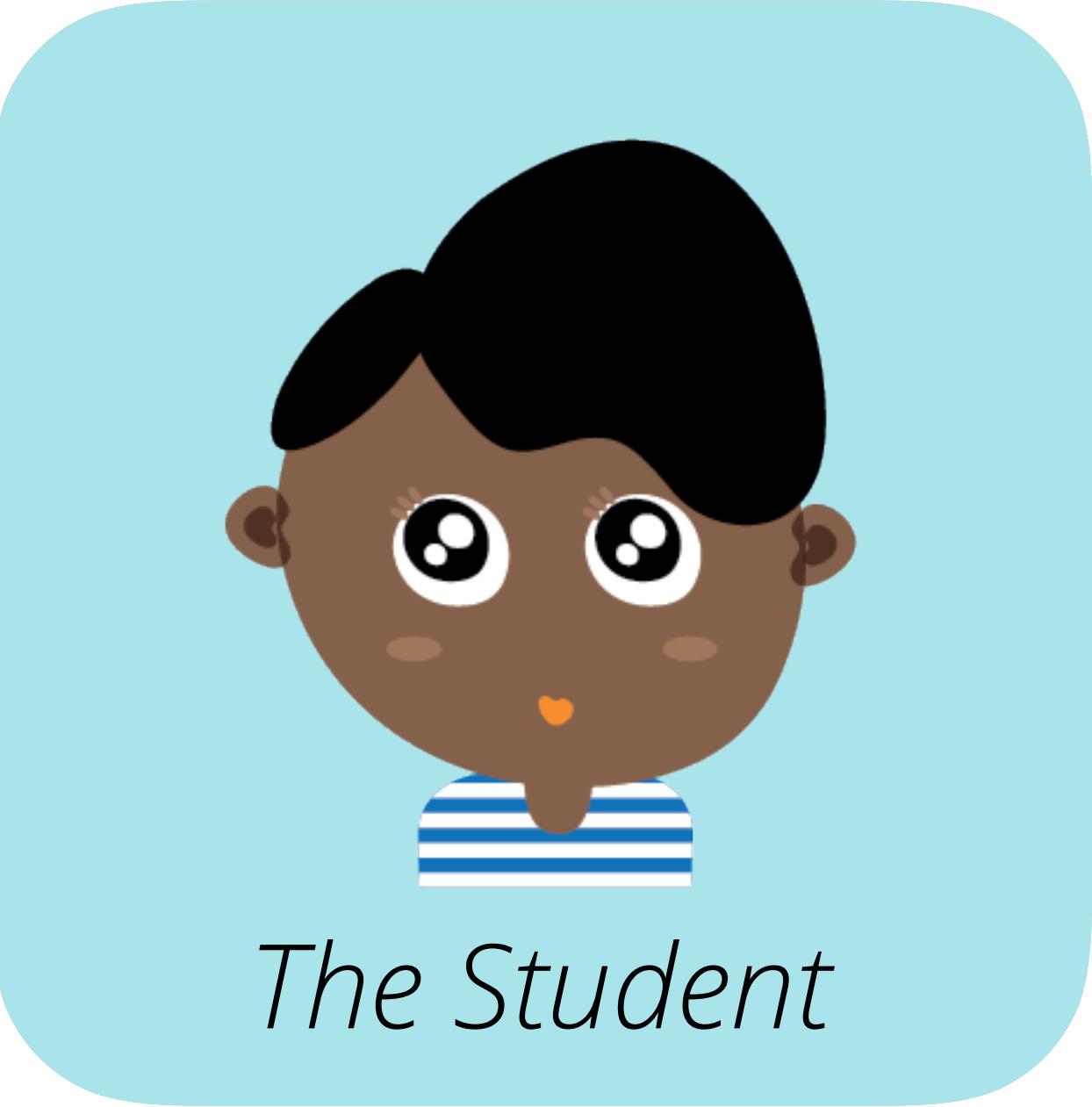
*The Startup User*

# THE JUDGES

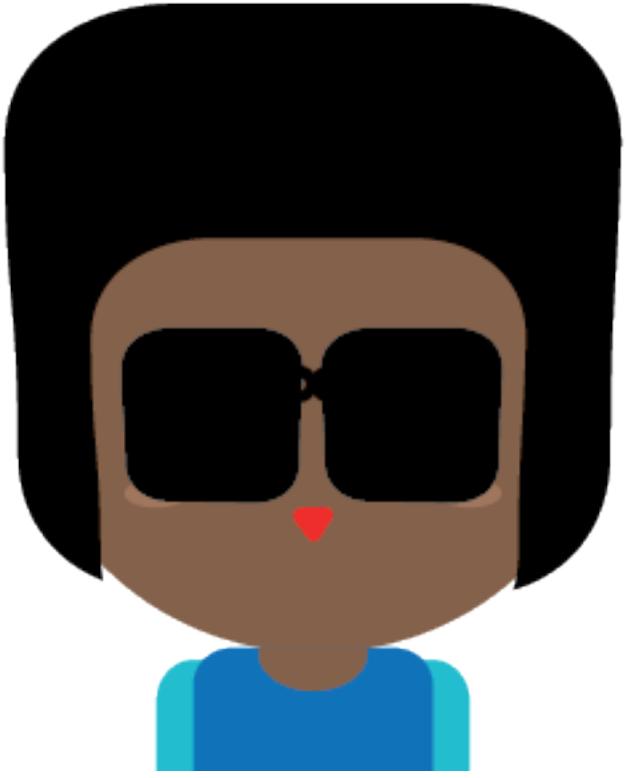
---



*The Enterprise User*



*The Student*



*The Startup User*



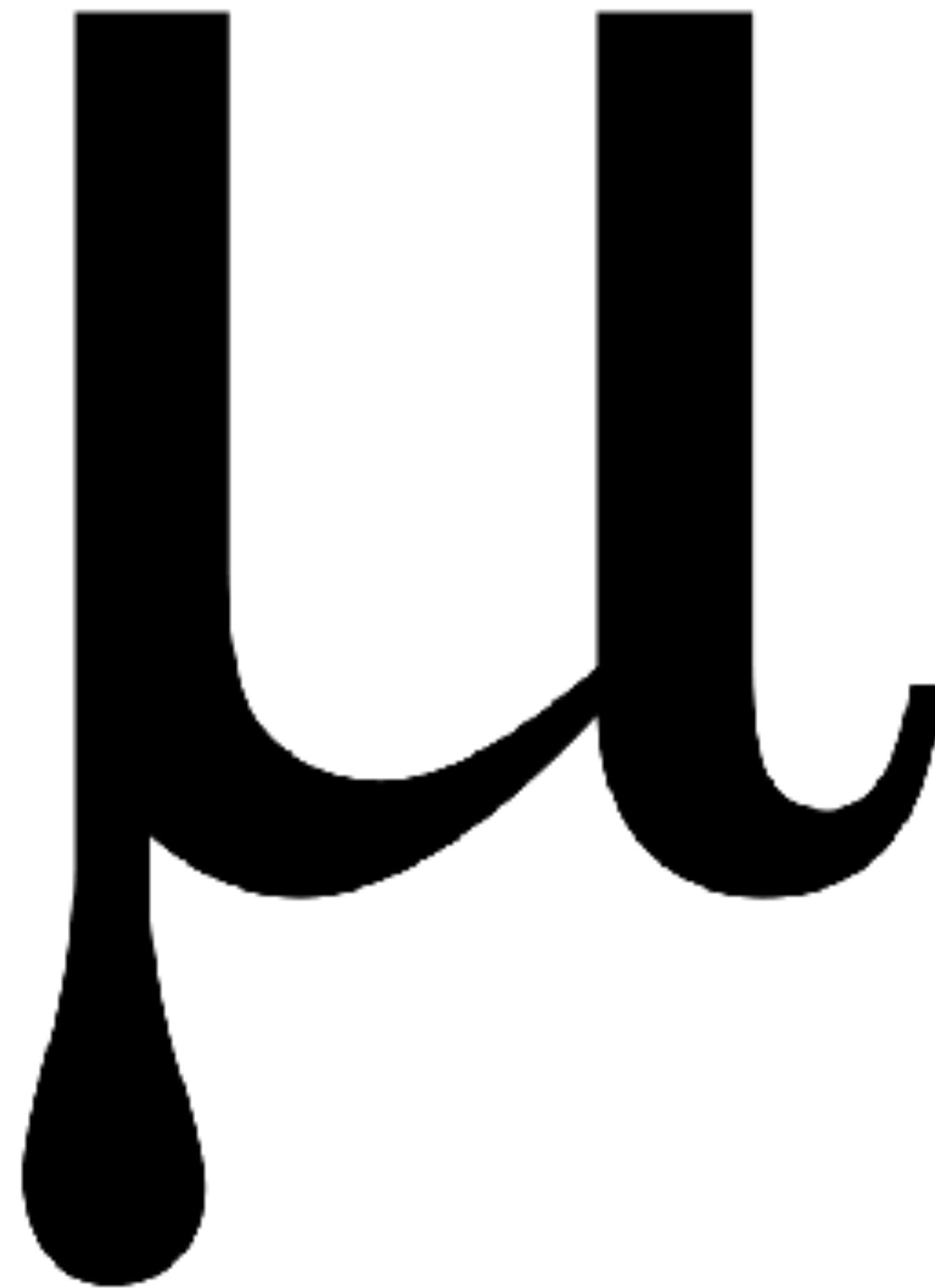
# THE TITLE FIGHT

---

*The bell has rung.. Here we go!*

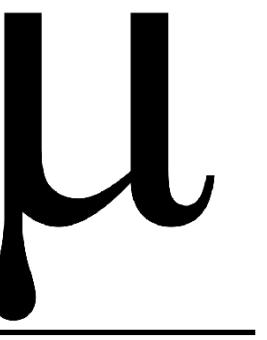
## ROUND 1: DEPLOYMENT - MICROSERVICES

---



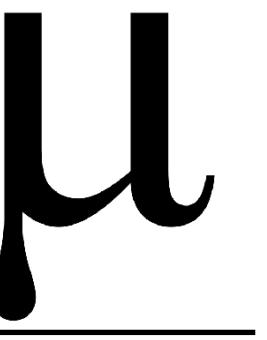
# ROUND 1: DEPLOYMENT - MICROSERVICES

---



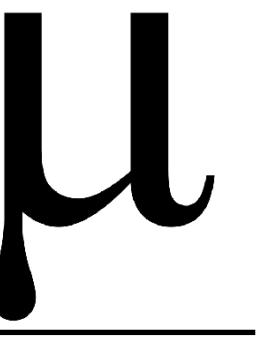
# ROUND 1: DEPLOYMENT - MICROSERVICES

---



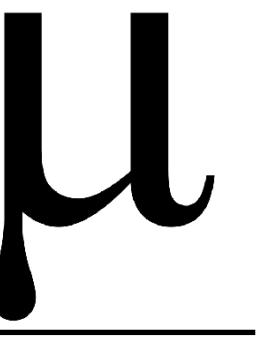
# ROUND 1: DEPLOYMENT - MICROSERVICES

---



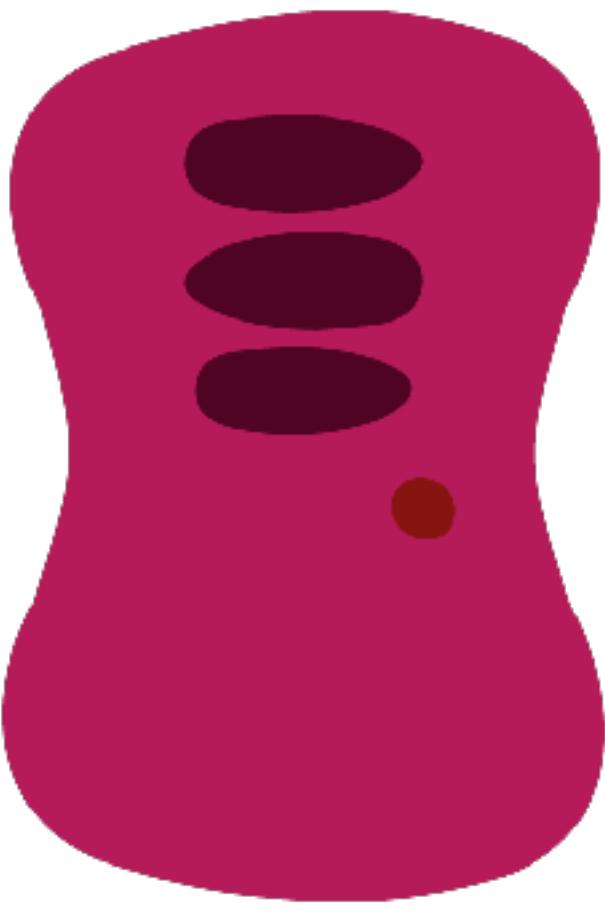
# ROUND 1: DEPLOYMENT - MICROSERVICES

---



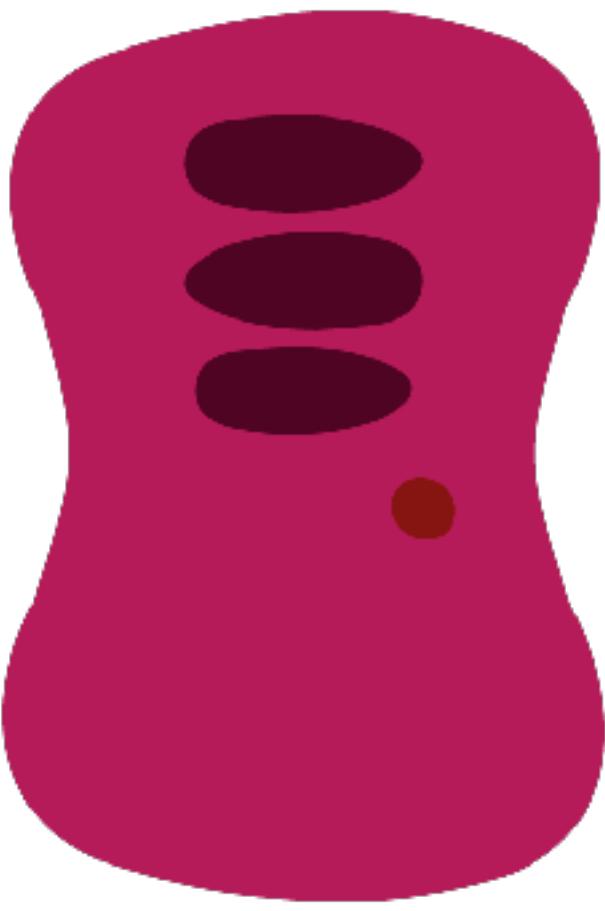
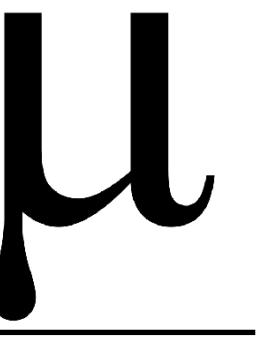
# ROUND 1: DEPLOYMENT - MICROSERVICES

---



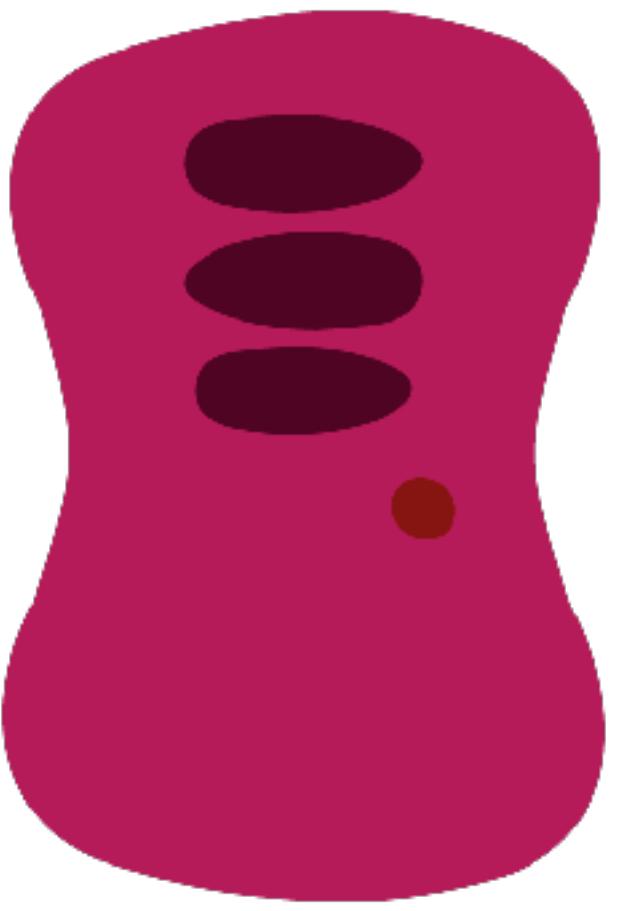
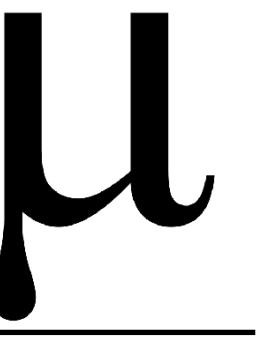
# ROUND 1: DEPLOYMENT - MICROSERVICES

---



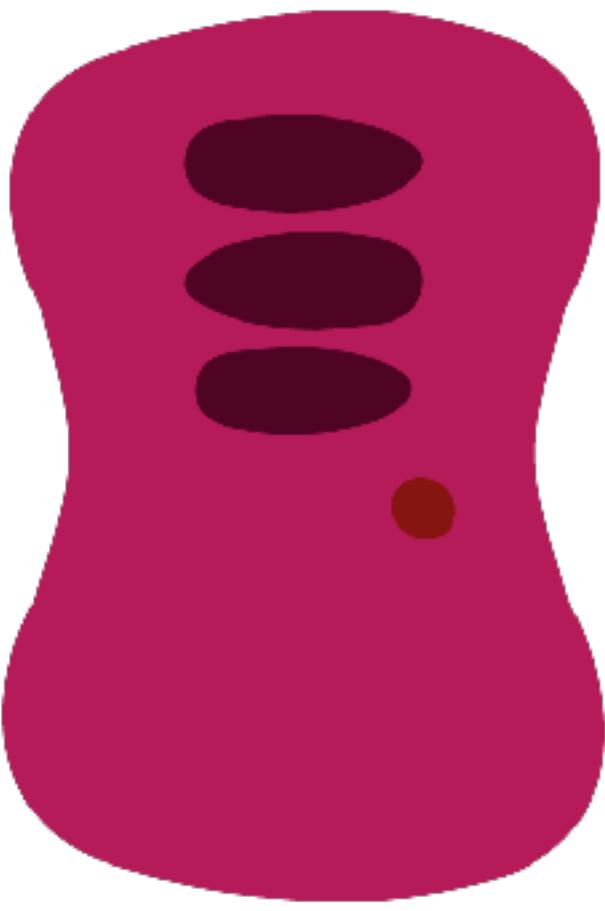
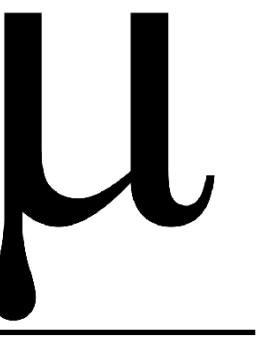
# ROUND 1: DEPLOYMENT - MICROSERVICES

---



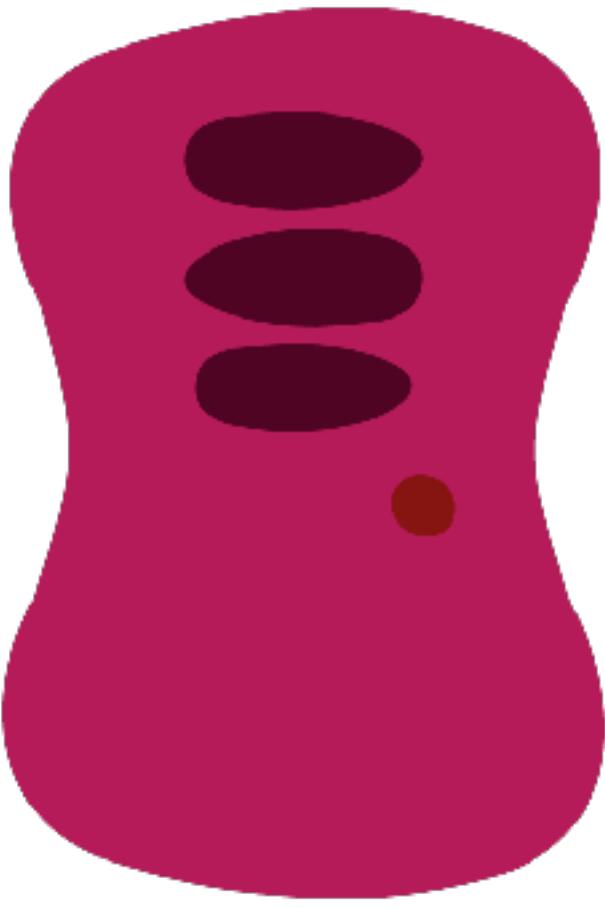
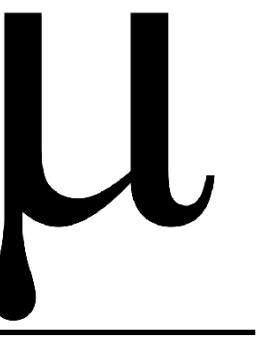
# ROUND 1: DEPLOYMENT - MICROSERVICES

---



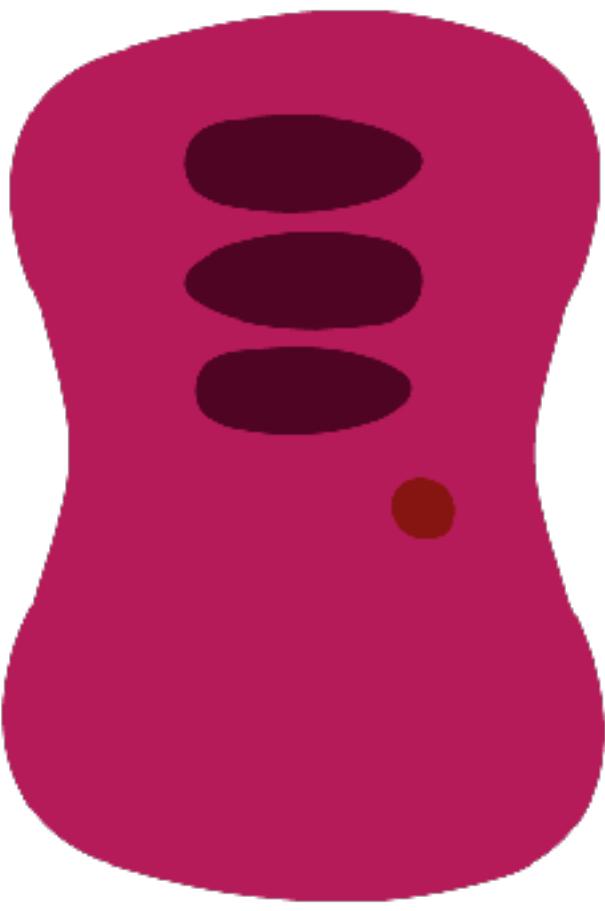
# ROUND 1: DEPLOYMENT - MICROSERVICES

---



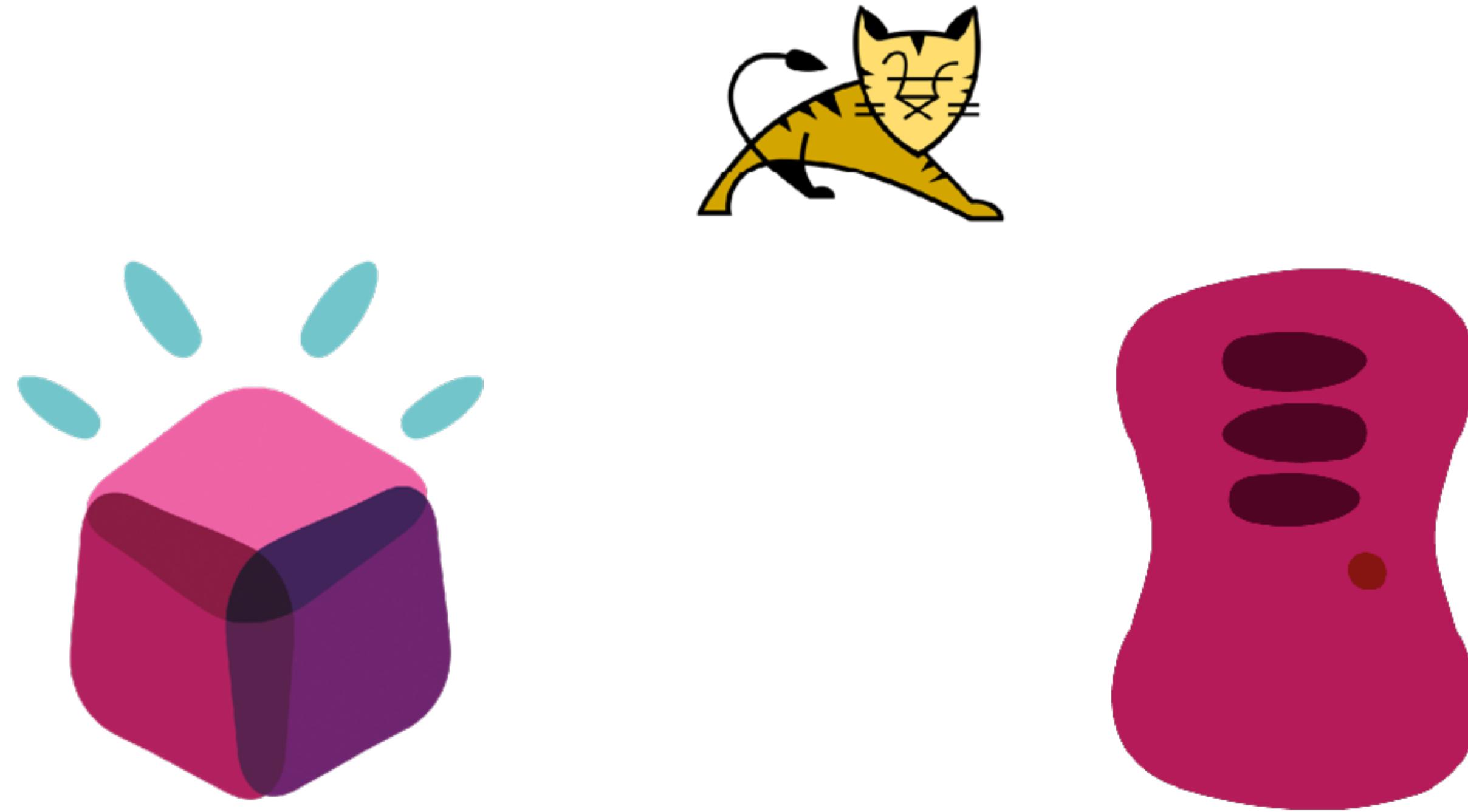
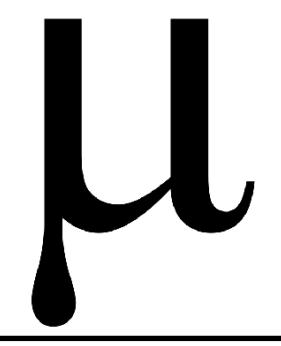
# ROUND 1: DEPLOYMENT - MICROSERVICES

---



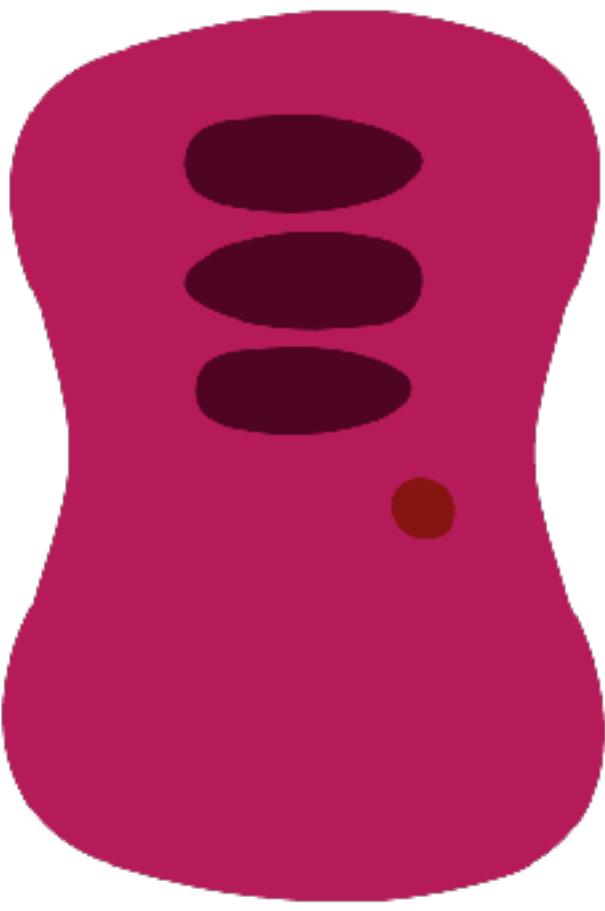
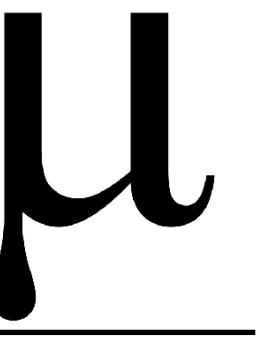
# ROUND 1: DEPLOYMENT - MICROSERVICES

---



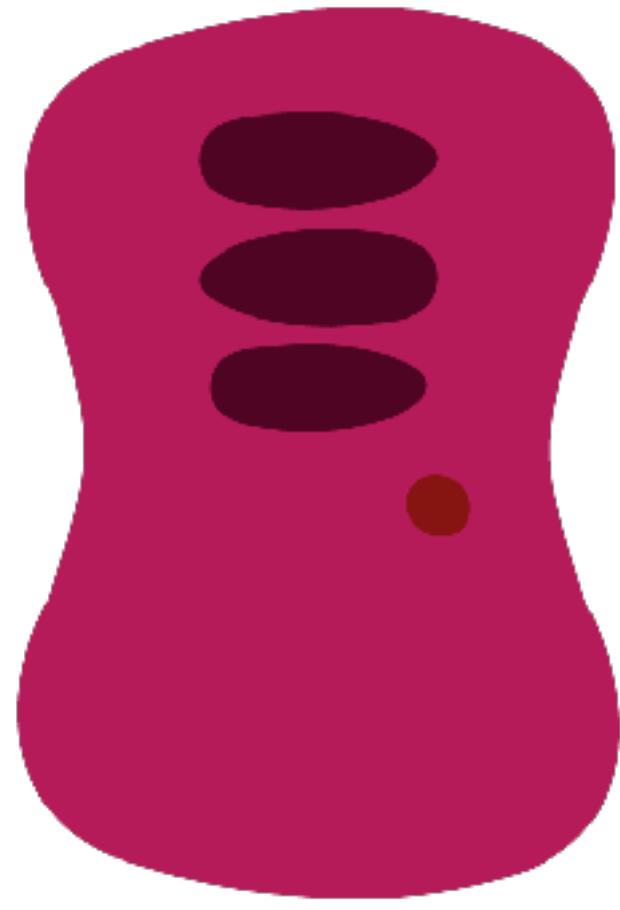
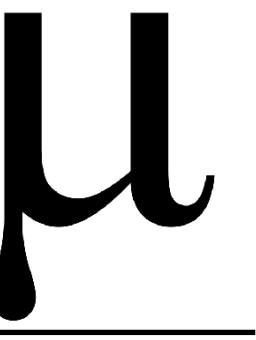
# ROUND 1: DEPLOYMENT - MICROSERVICES

---



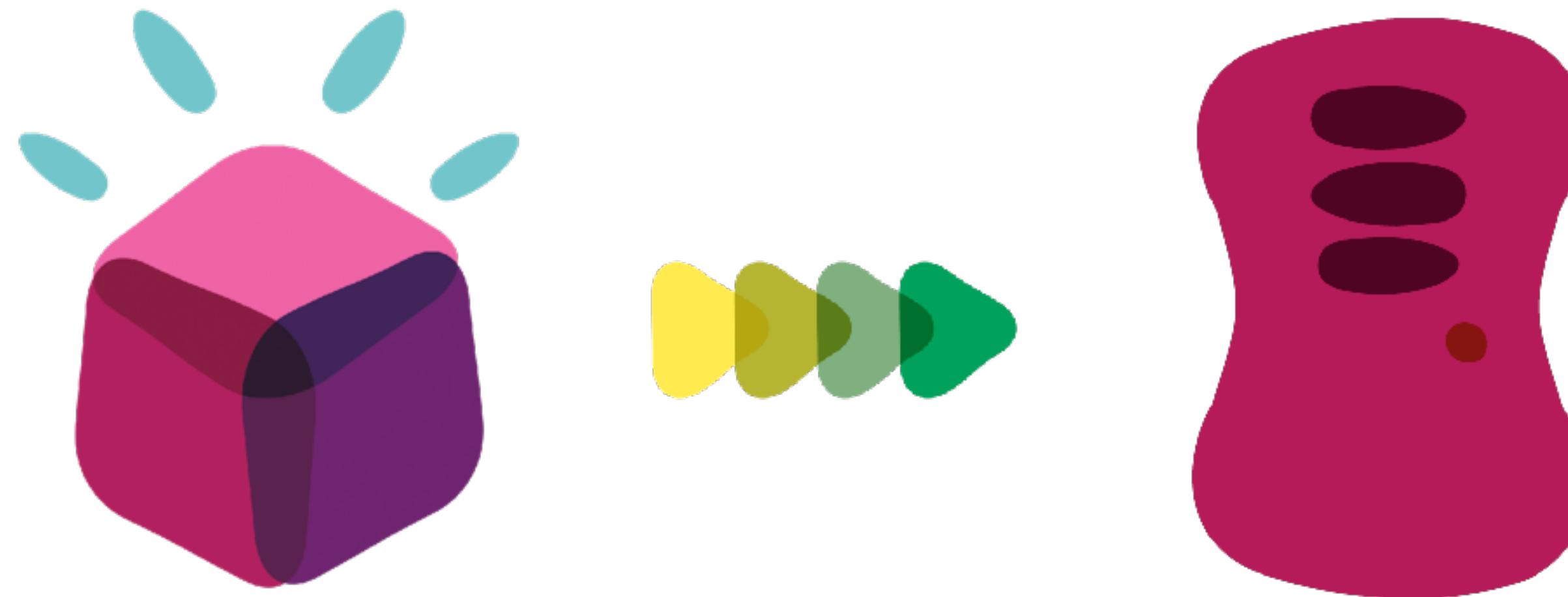
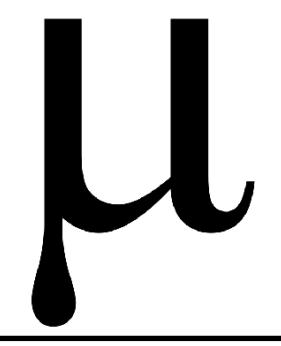
# ROUND 1: DEPLOYMENT - MICROSERVICES

---



## ROUND 1: DEPLOYMENT - MICROSERVICES

---



```
$ java -jar ./app.jar
```

## ROUND 1: DEPLOYMENT - SERVERLESS

---



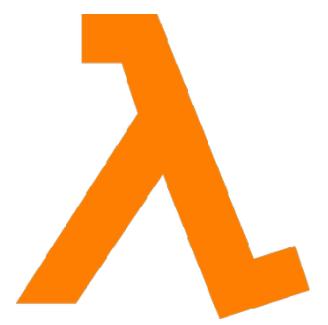
# ROUND 1: DEPLOYMENT - SERVERLESS

---



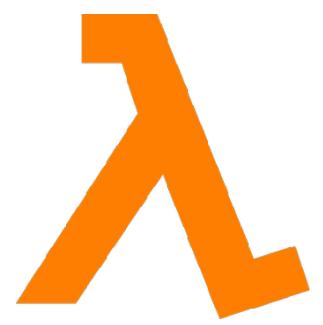
# ROUND 1: DEPLOYMENT - SERVERLESS

---



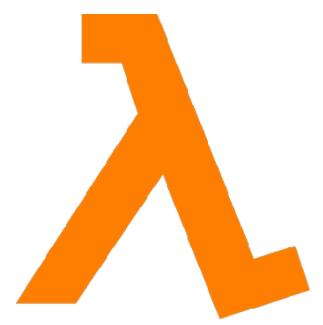
# ROUND 1: DEPLOYMENT - SERVERLESS

---



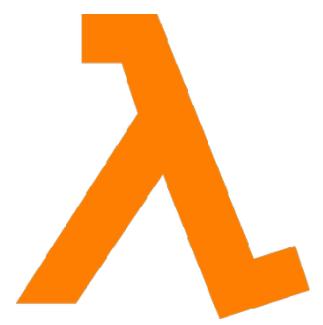
# ROUND 1: DEPLOYMENT - SERVERLESS

---



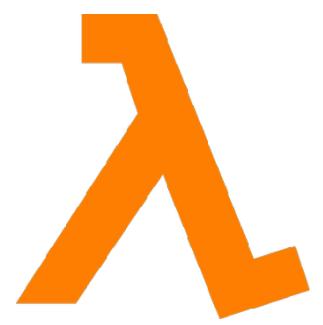
## ROUND 1: DEPLOYMENT - SERVERLESS

---



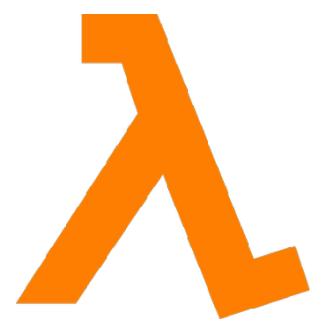
# ROUND 1: DEPLOYMENT - SERVERLESS

---



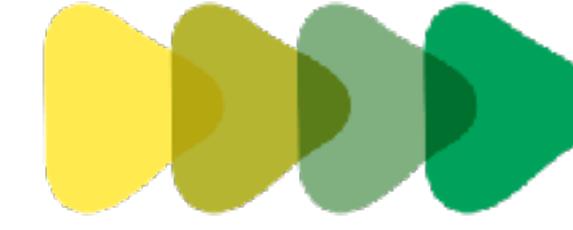
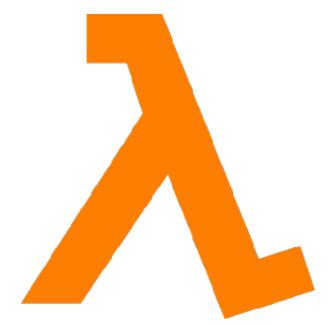
# ROUND 1: DEPLOYMENT - SERVERLESS

---



## ROUND 1: DEPLOYMENT - SERVERLESS

---



\$ serverless deploy

## ROUND 1 - SCORE CARD

---



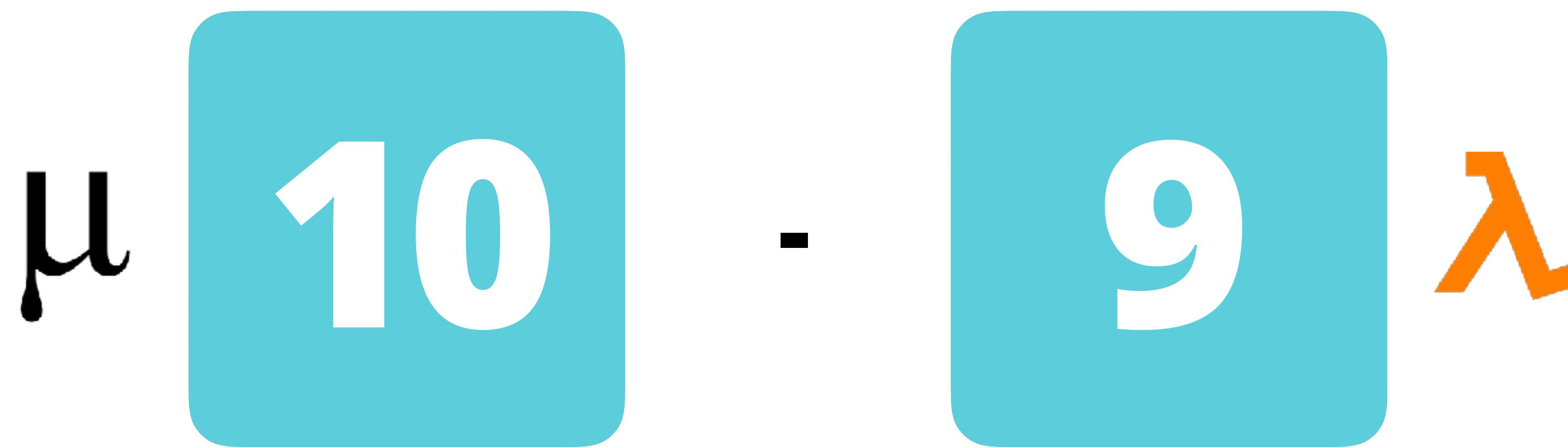
*The Enterprise User*

## ROUND 1 - SCORE CARD

---

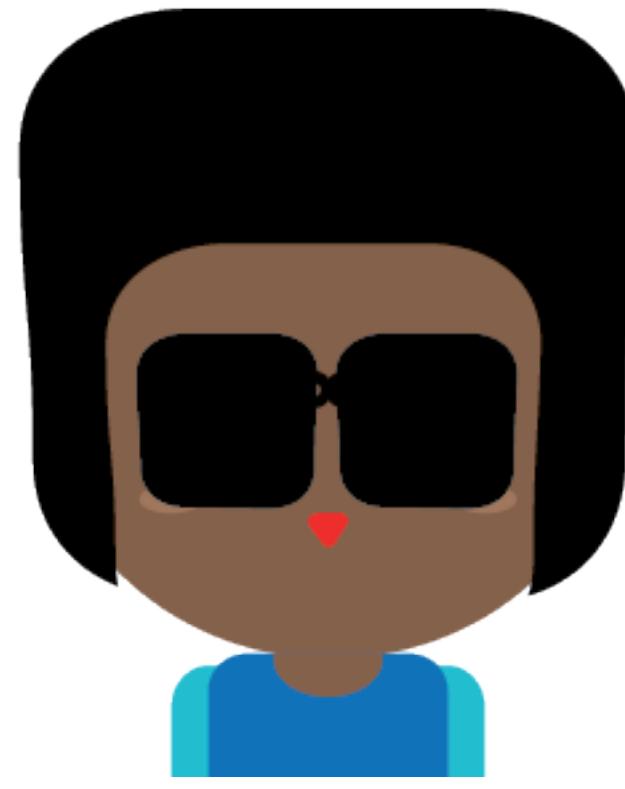


*The Enterprise User*



## ROUND 1 - SCORE CARD

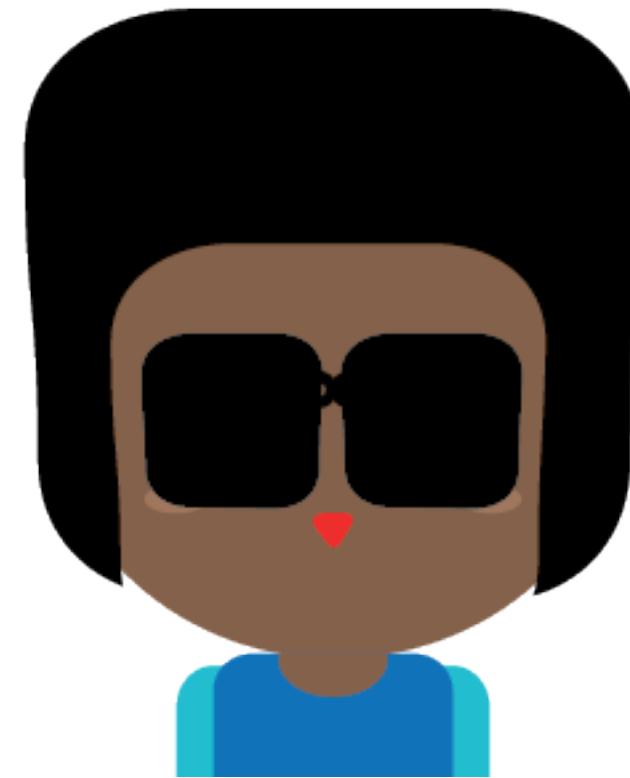
---



*The Startup User*

## ROUND 1 - SCORE CARD

---



*The Startup User*

$\mu$

8

.

10

$\lambda$

# ROUND 1 - SCORE CARD

---



*The Student*

## ROUND 1 - SCORE CARD

---



*The Student*

$\mu$

7

-

10

$\lambda$

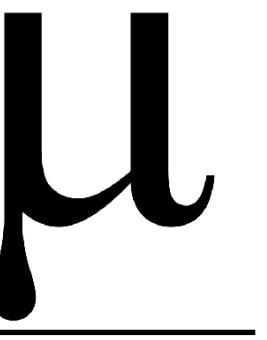
## ROUND 2: SCALABILITY - MICROSERVICES

---



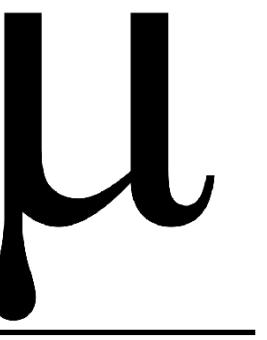
## ROUND 2: SCALABILITY - MICROSERVICES

---



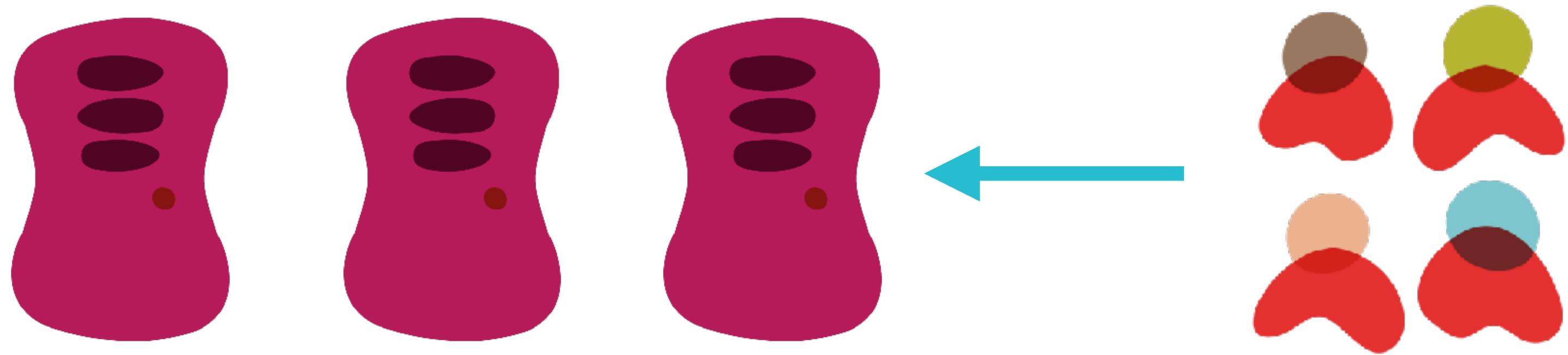
## ROUND 2: SCALABILITY - MICROSERVICES

---



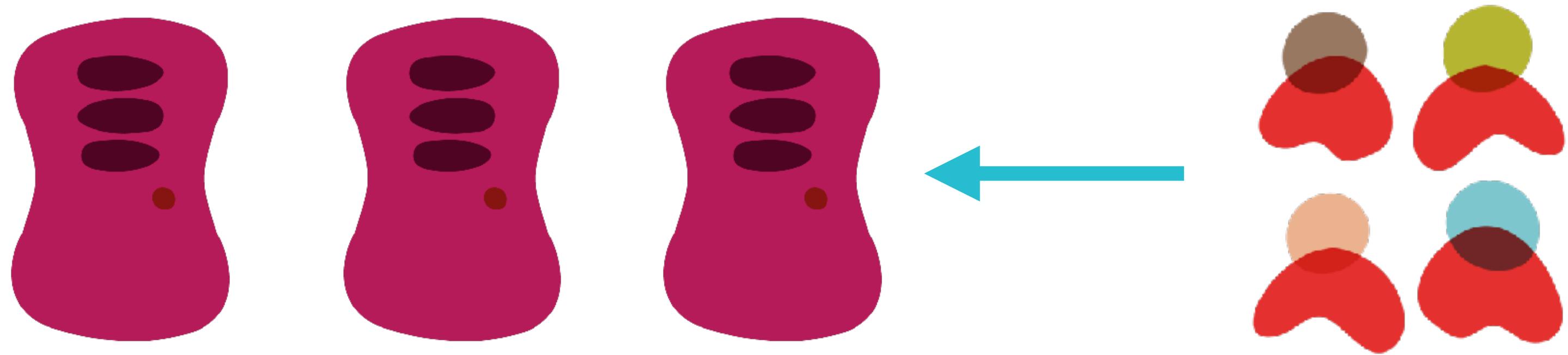
## ROUND 2: SCALABILITY - MICROSERVICES

---

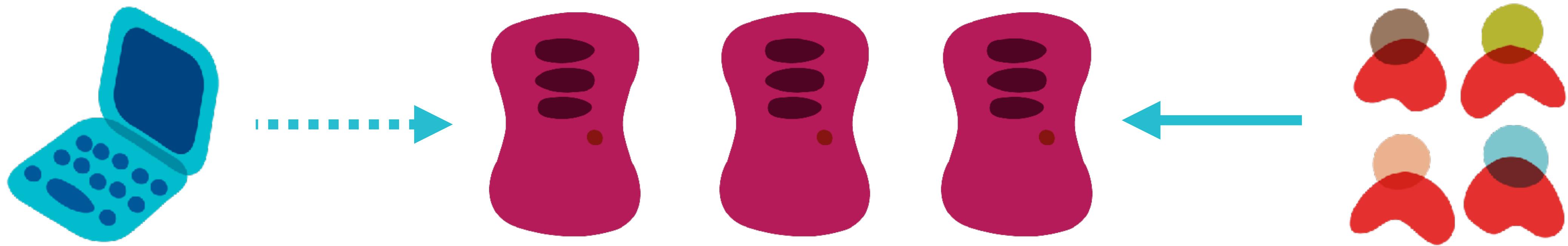


## ROUND 2: SCALABILITY - MICROSERVICES

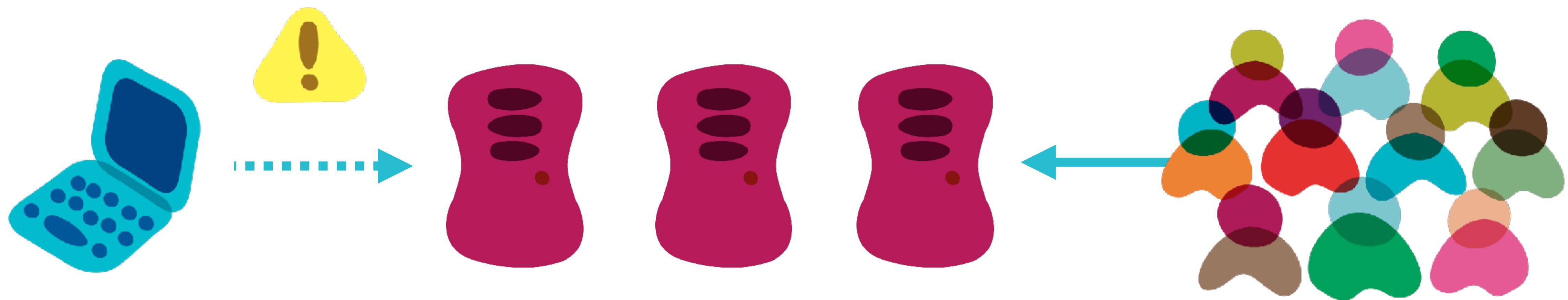
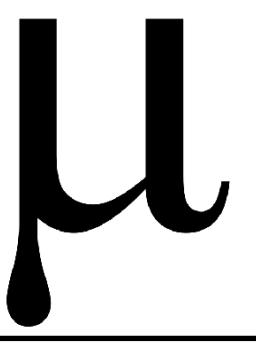
---



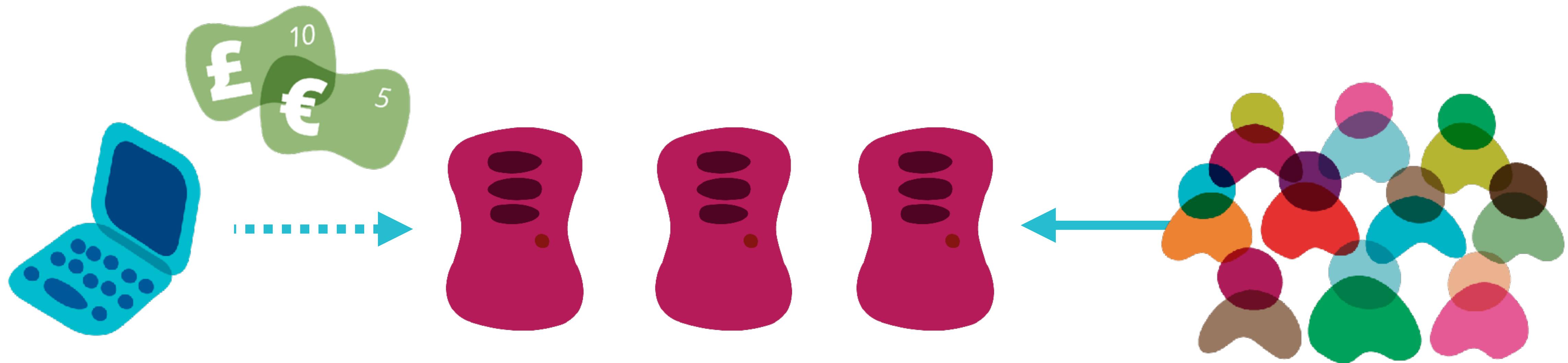
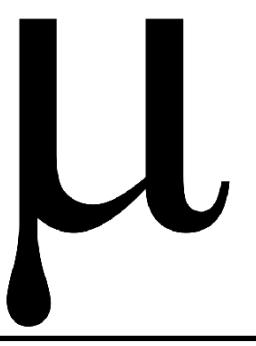
## ROUND 2: SCALABILITY - MICROSERVICES



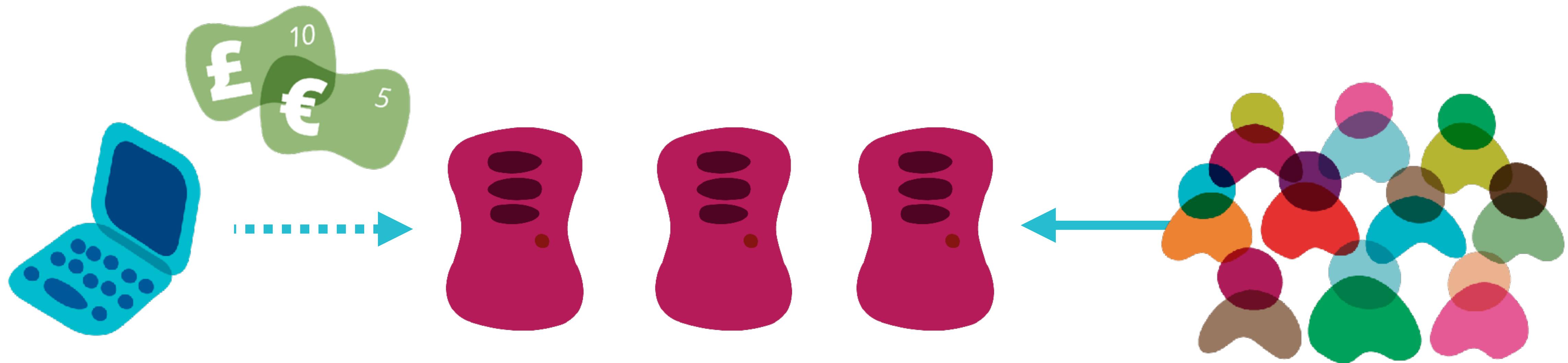
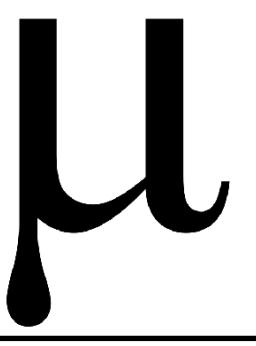
## ROUND 2: SCALABILITY - MICROSERVICES



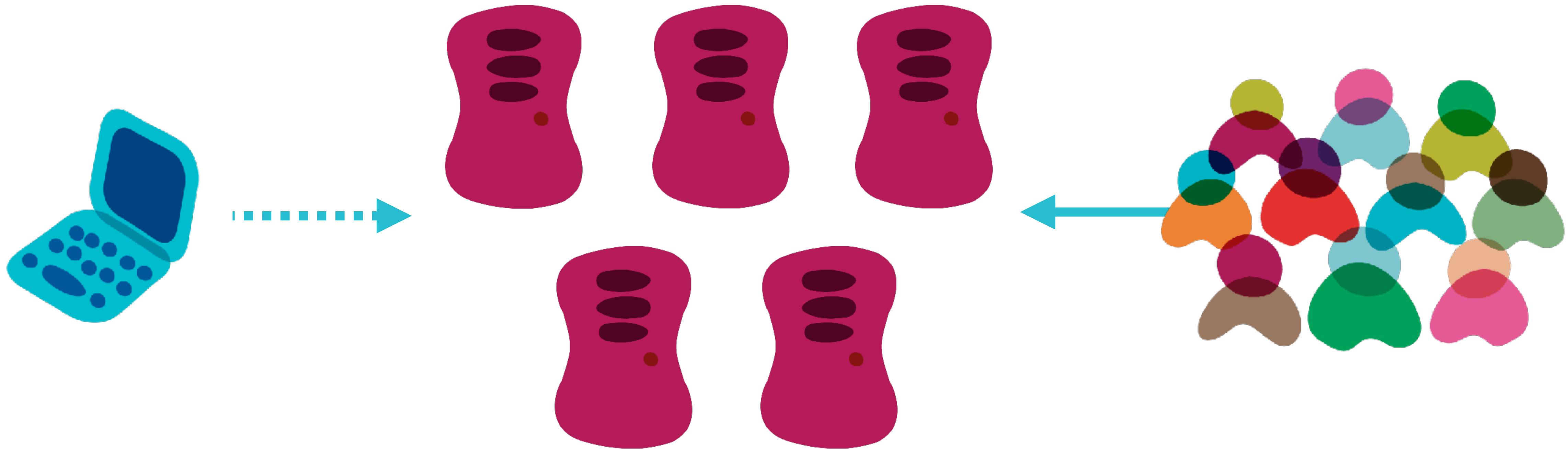
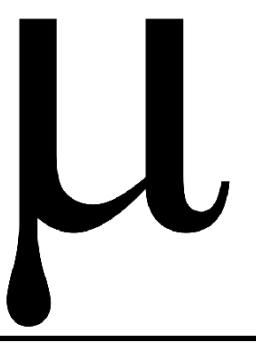
## ROUND 2: SCALABILITY - MICROSERVICES



## ROUND 2: SCALABILITY - MICROSERVICES

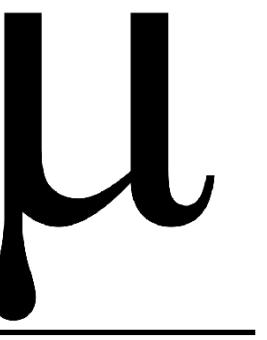


## ROUND 2: SCALABILITY - MICROSERVICES



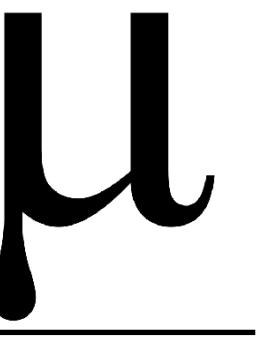
## ROUND 2: SCALABILITY - MICROSERVICES

---



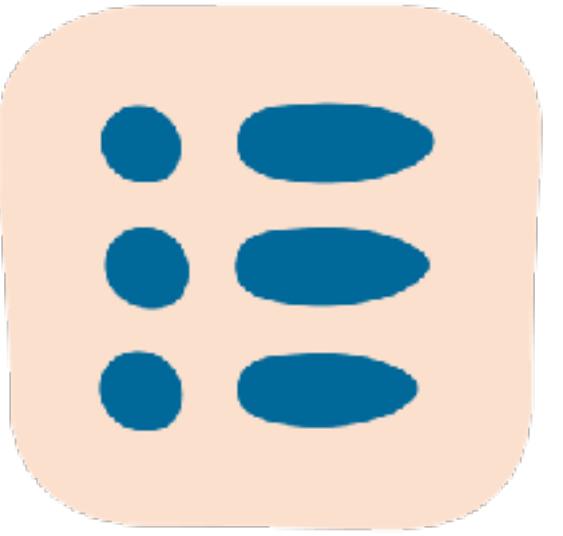
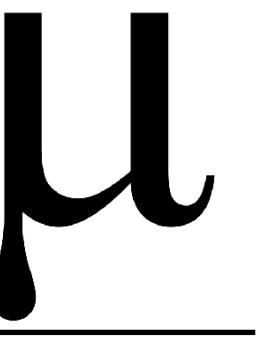
## ROUND 2: SCALABILITY - MICROSERVICES

---



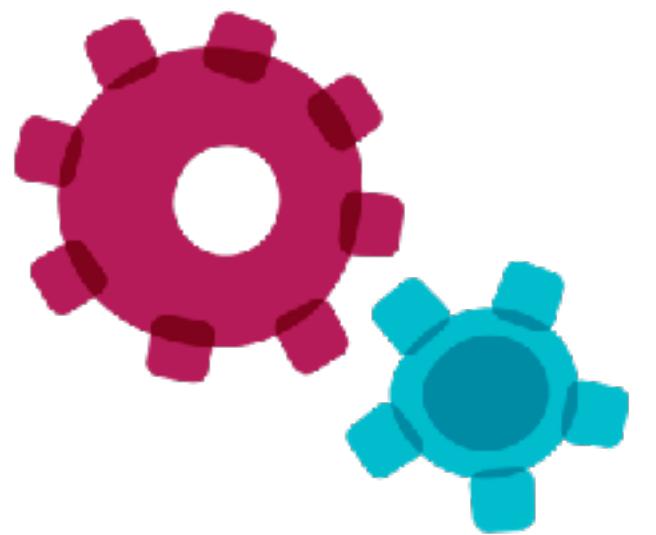
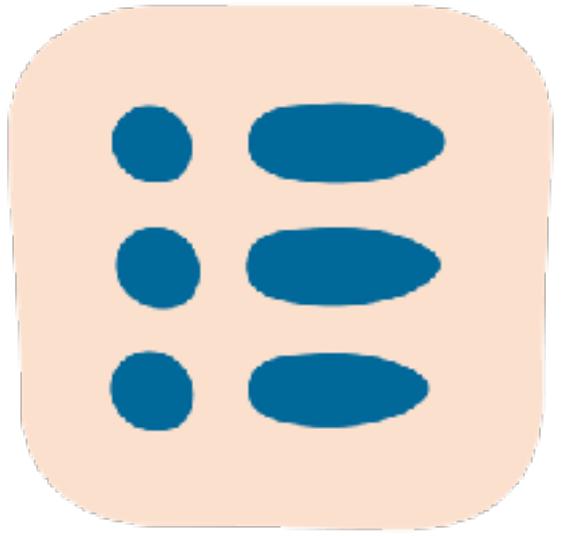
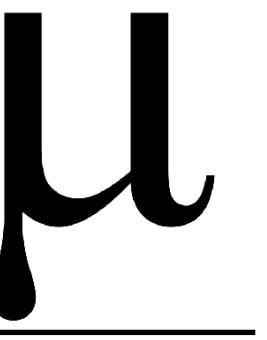
## ROUND 2: SCALABILITY - MICROSERVICES

---



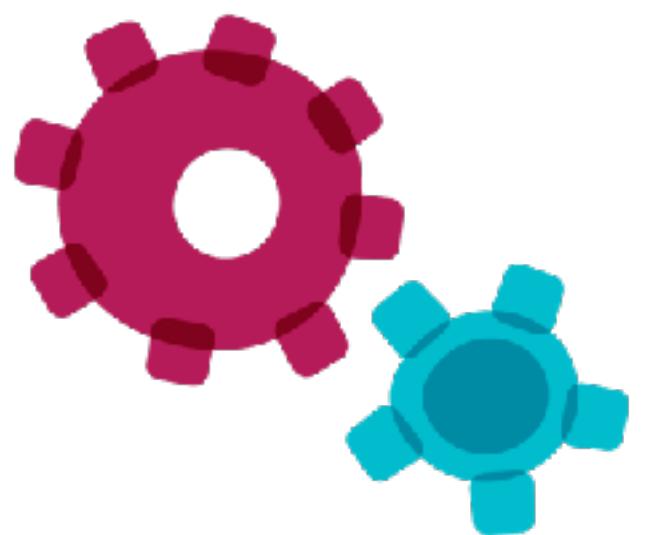
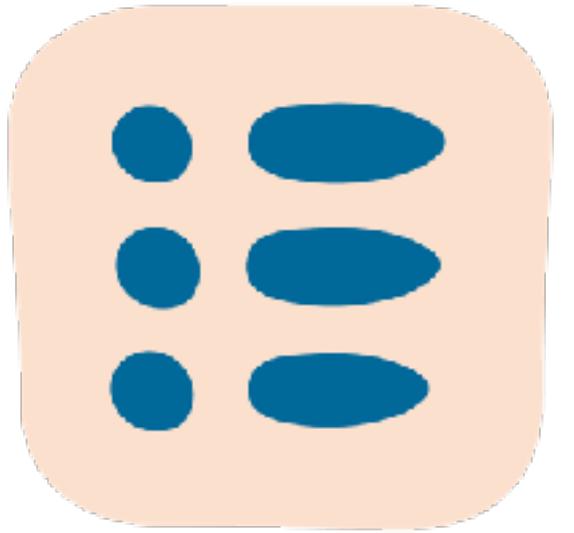
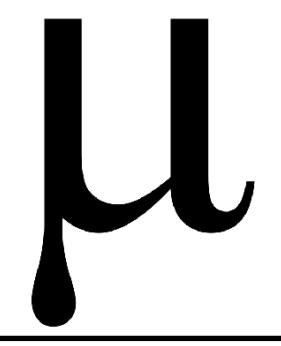
## ROUND 2: SCALABILITY - MICROSERVICES

---



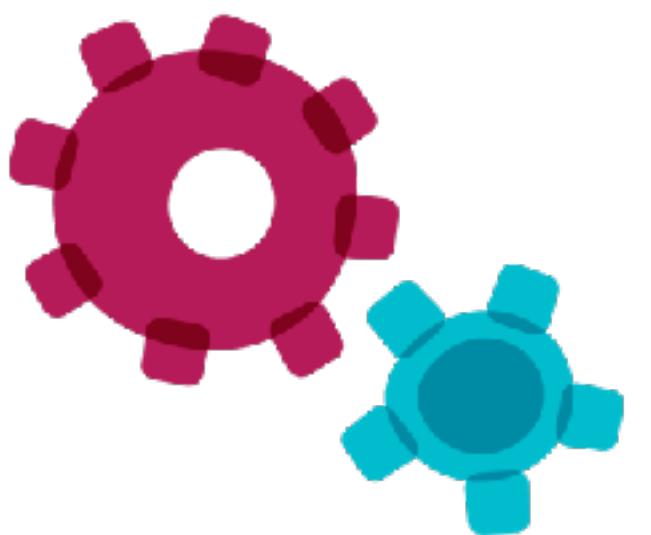
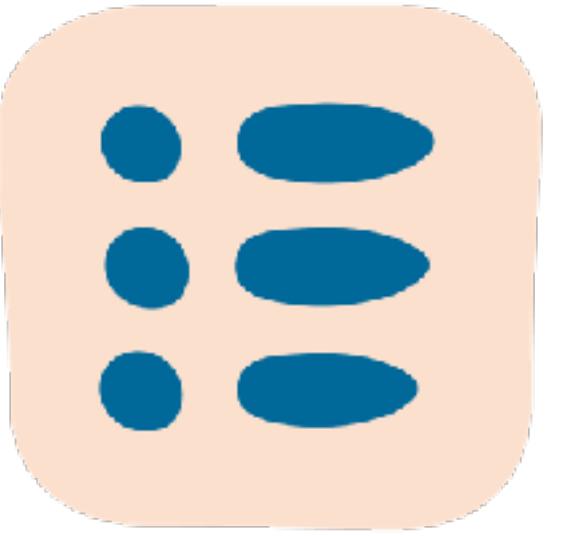
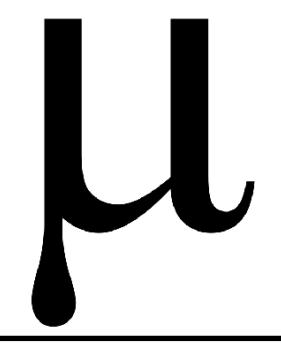
## ROUND 2: SCALABILITY - MICROSERVICES

---



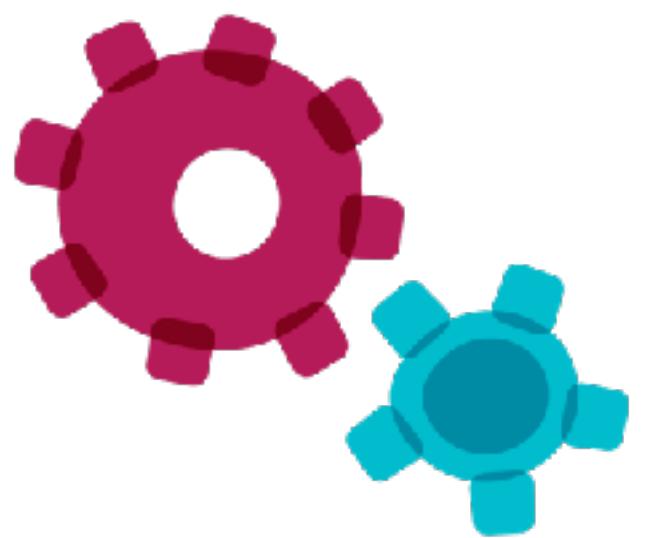
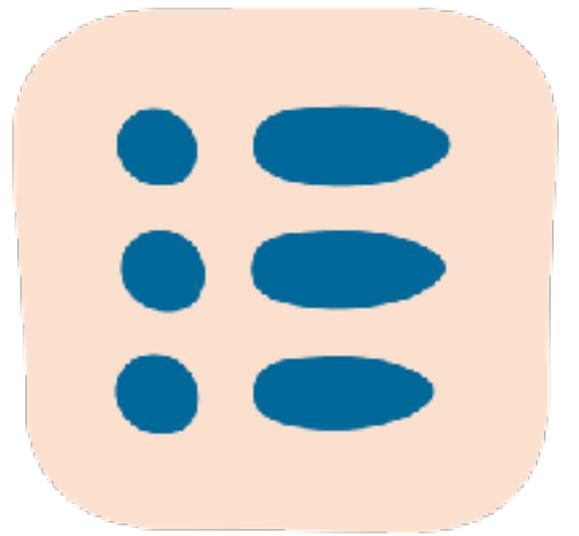
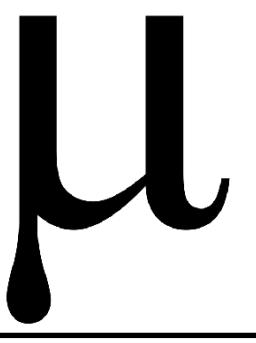
## ROUND 2: SCALABILITY - SERVERLESS

---

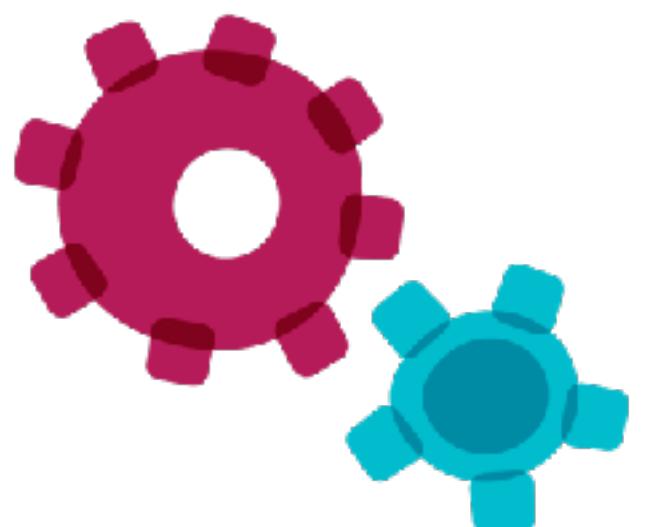
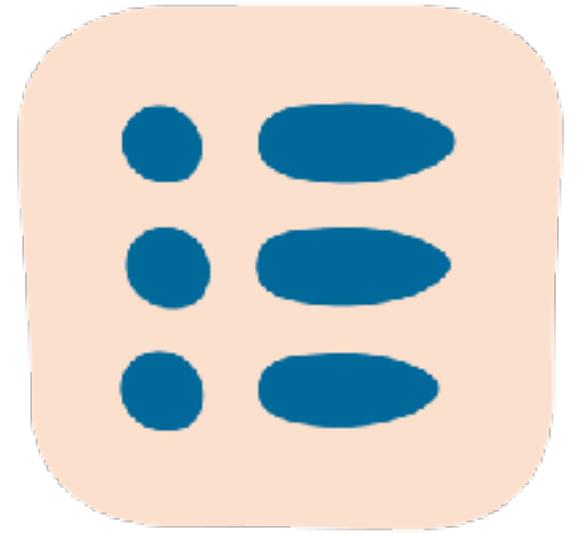
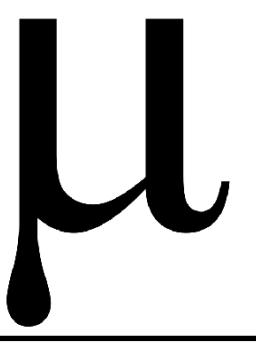


## ROUND 2: SCALABILITY - SERVERLESS

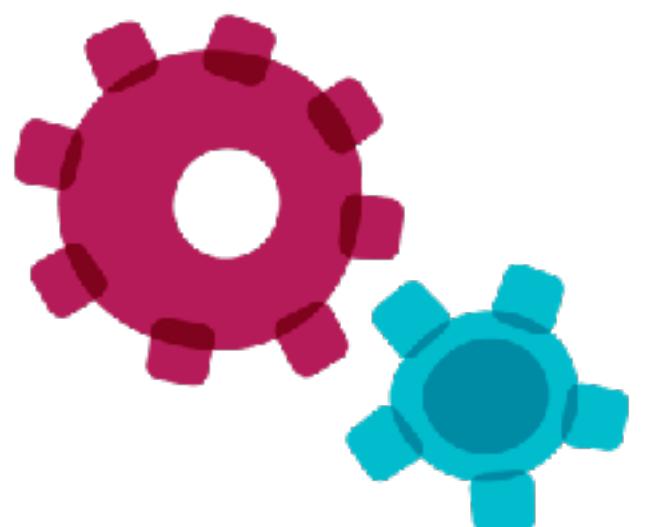
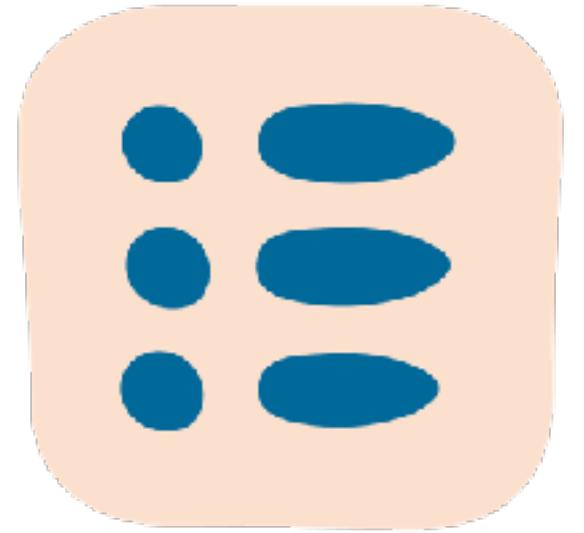
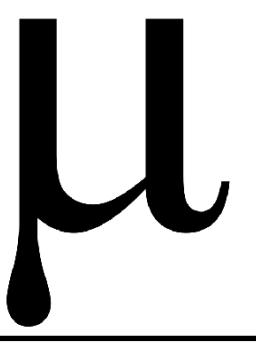
---



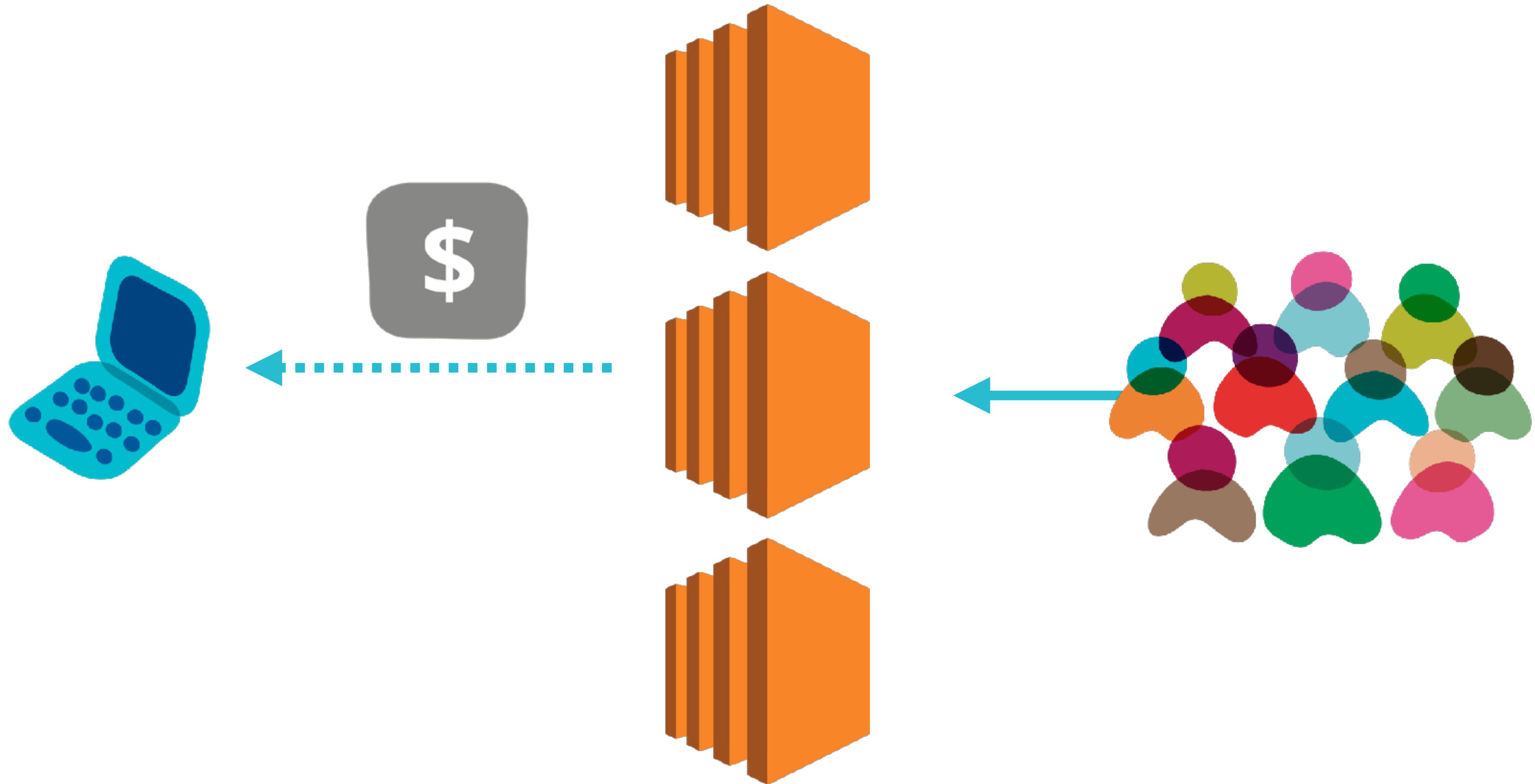
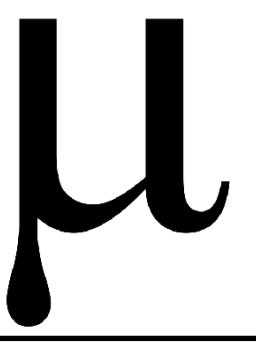
## ROUND 2: SCALABILITY - SERVERLESS



## ROUND 2: SCALABILITY - SERVERLESS



## ROUND 2: SCALABILITY - SERVERLESS



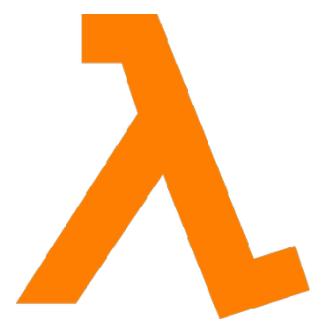
## ROUND 2: SCALABILITY - SERVERLESS

---



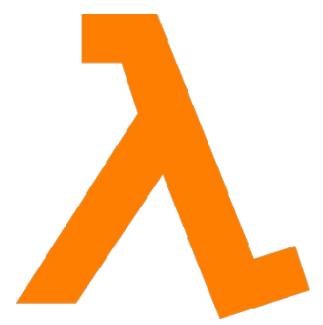
## ROUND 2: SCALABILITY - SERVERLESS

---



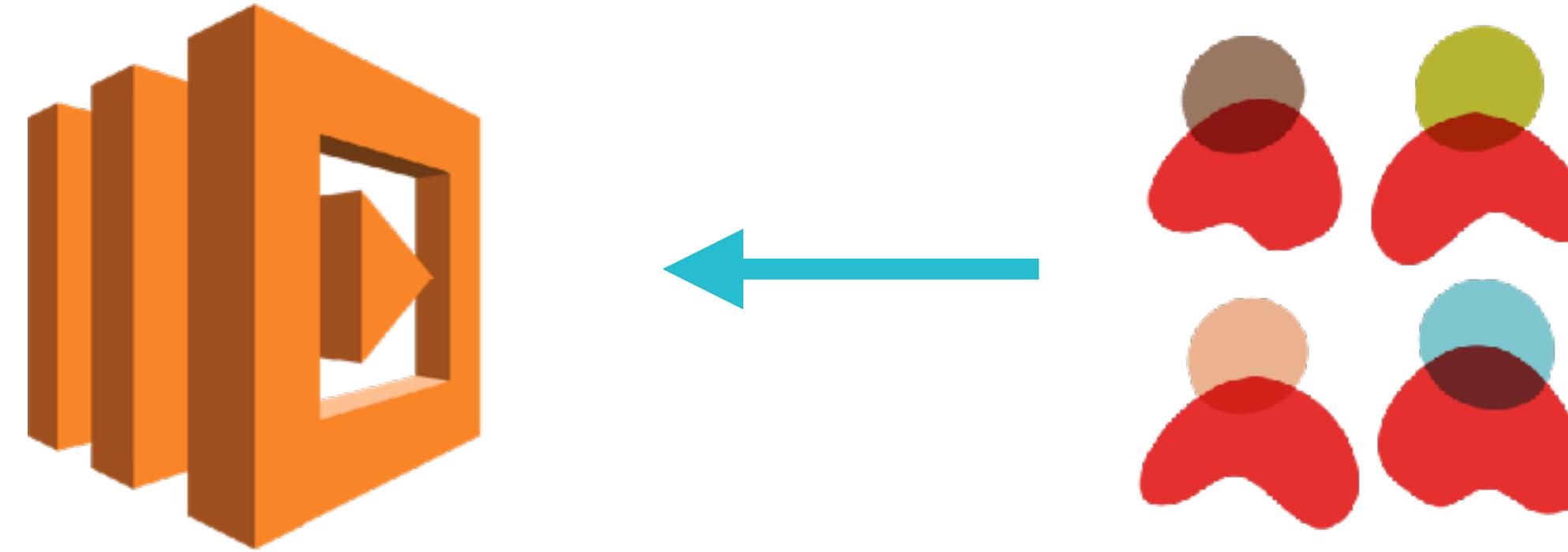
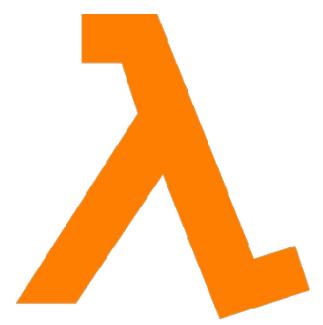
## ROUND 2: SCALABILITY - SERVERLESS

---



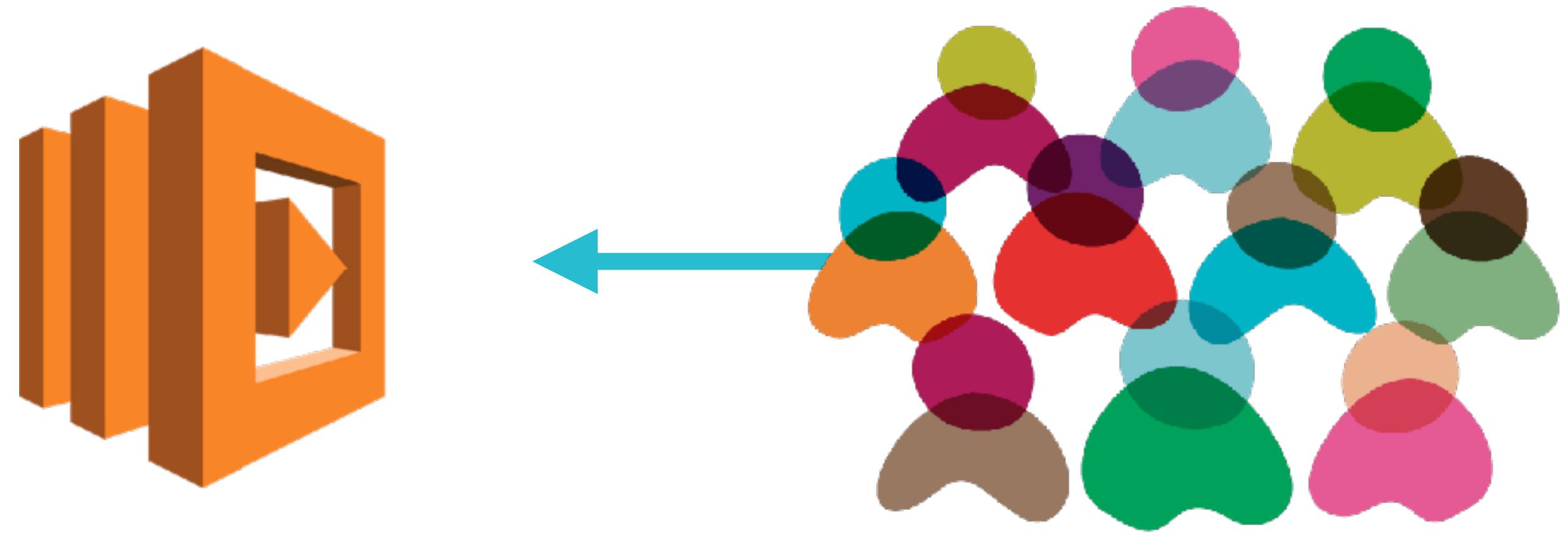
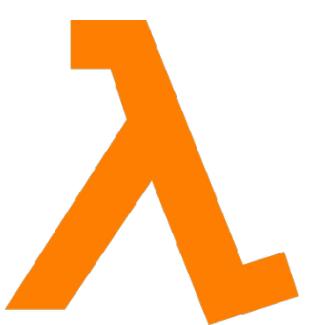
## ROUND 2: SCALABILITY - SERVERLESS

---



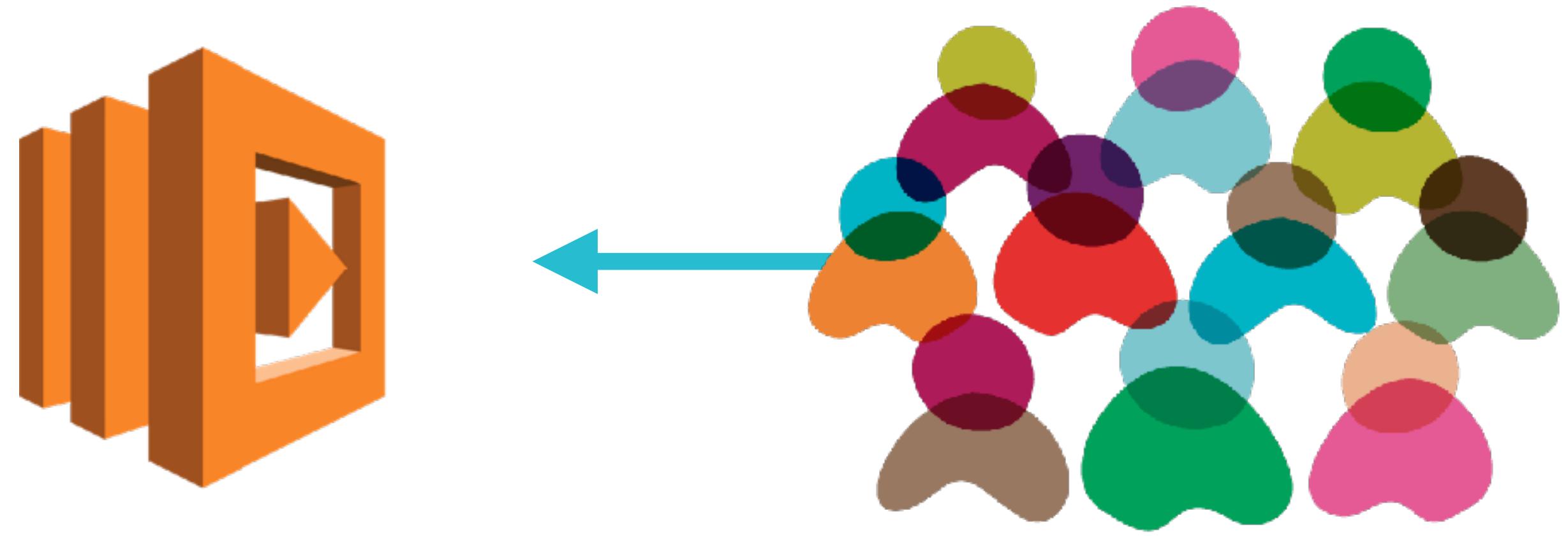
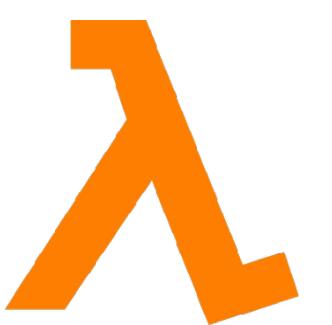
## ROUND 2: SCALABILITY - SERVERLESS

---



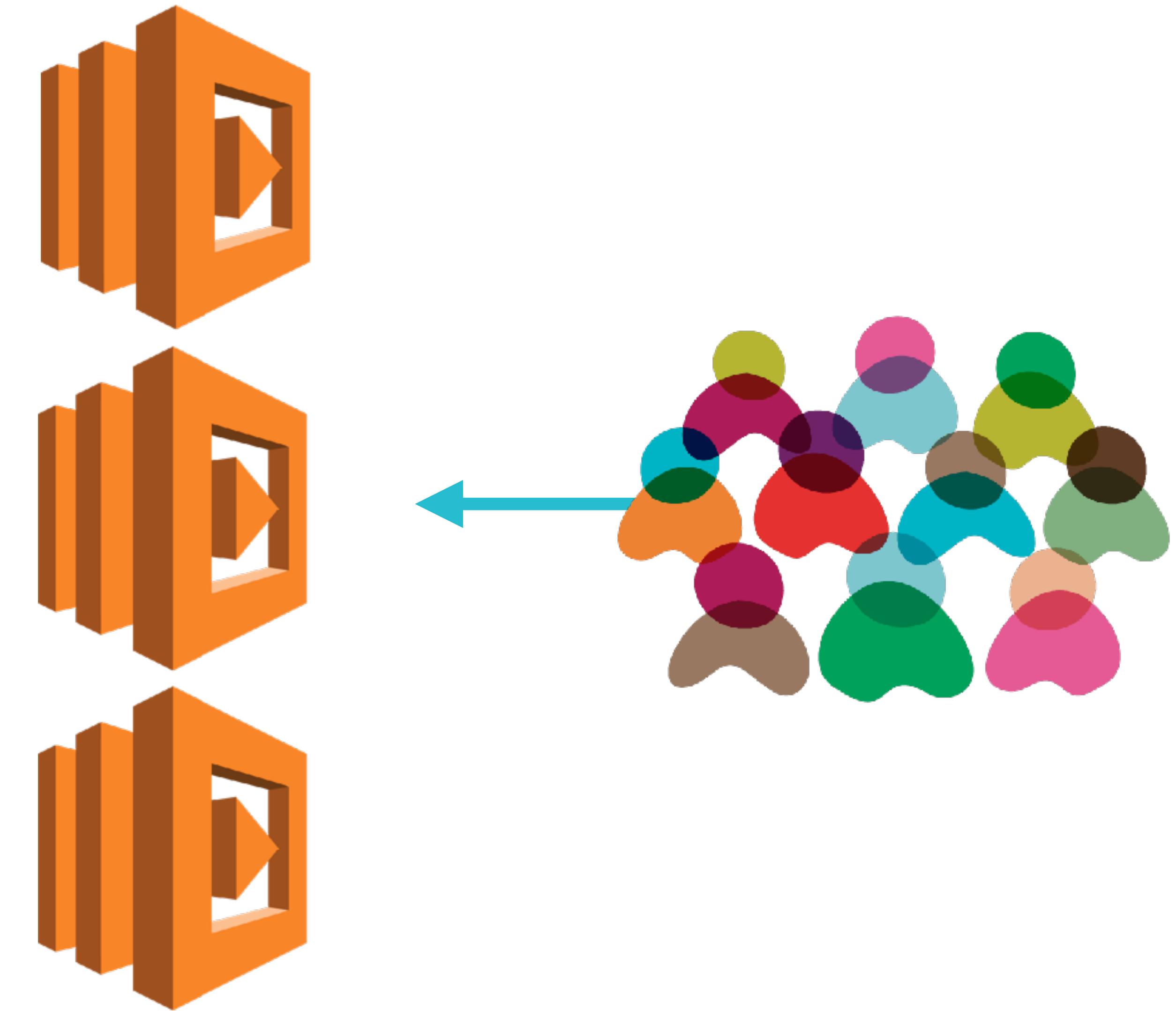
## ROUND 2: SCALABILITY - SERVERLESS

---

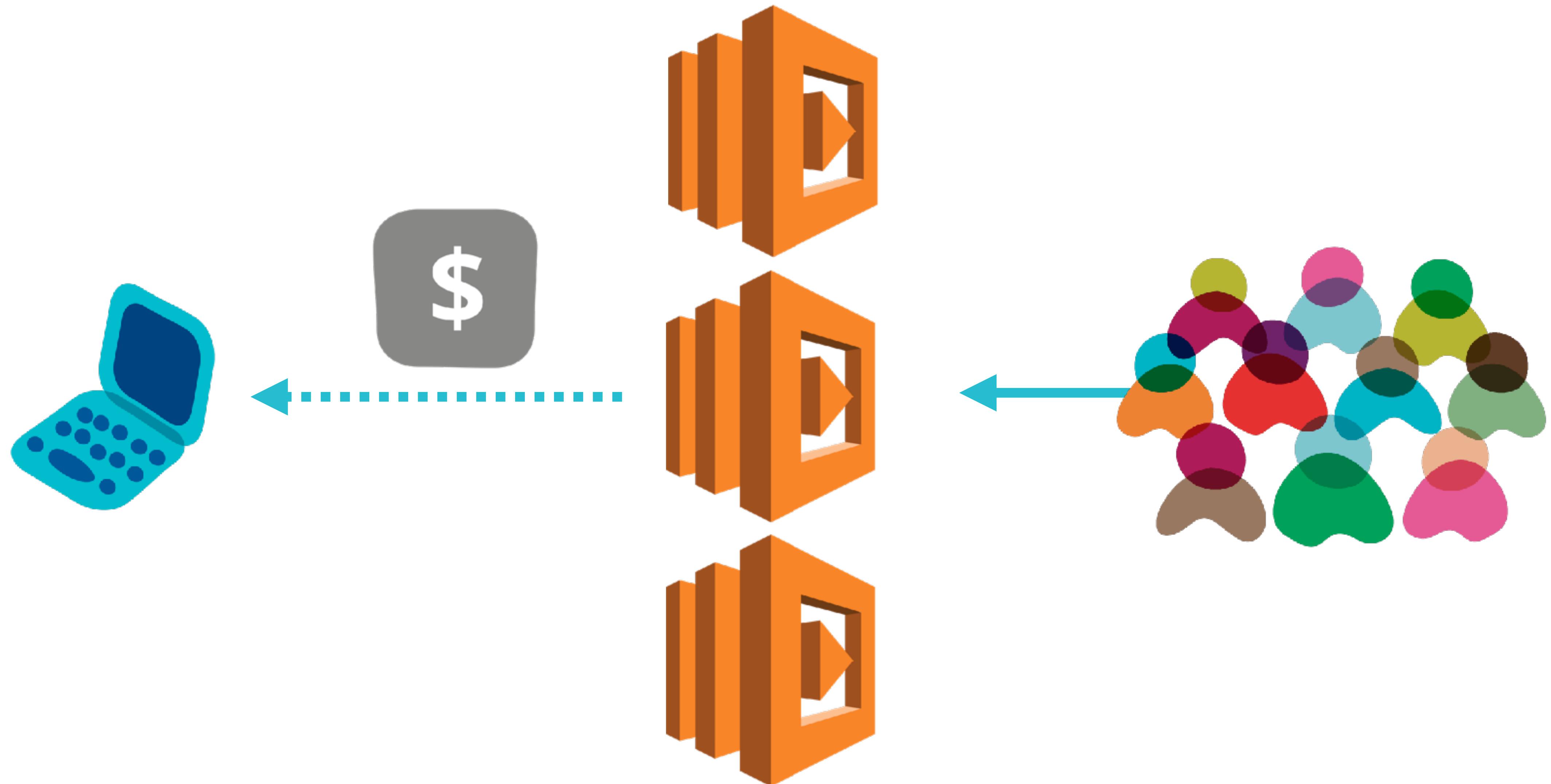


## ROUND 2: SCALABILITY - SERVERLESS

---



## ROUND 2: SCALABILITY - SERVERLESS



## ROUND 2 - SCORE CARD

---



*The Enterprise User*

## ROUND 2 - SCORE CARD

---



*The Enterprise User*

$\mu$

10

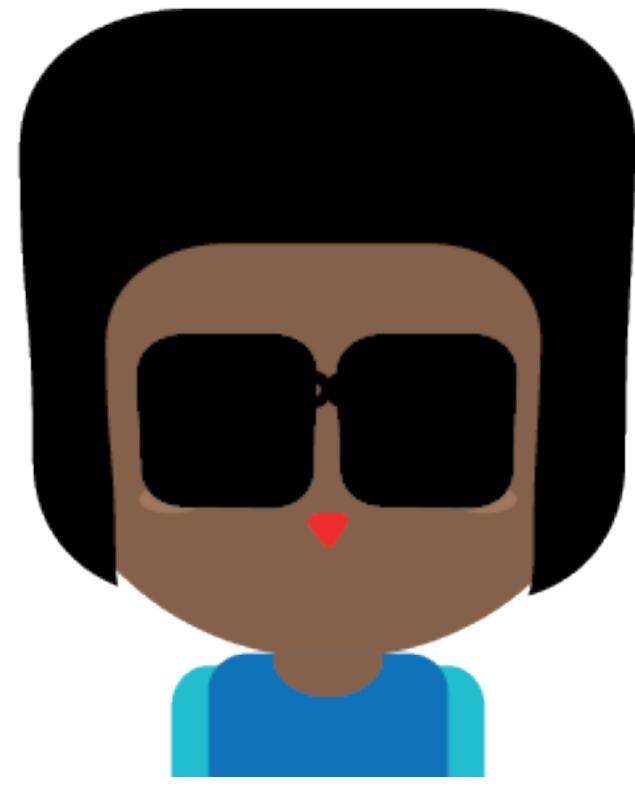
-

8

$\lambda$

## ROUND 2 - SCORE CARD

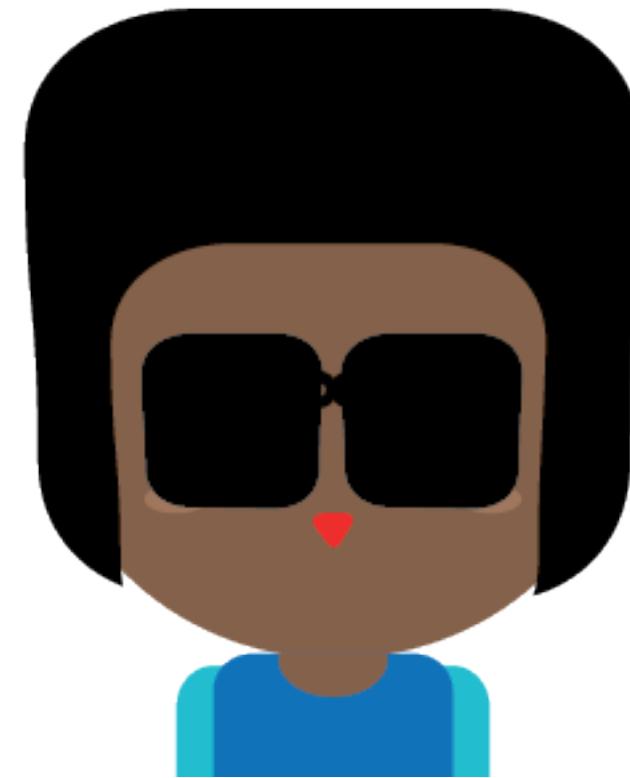
---



*The Startup User*

## ROUND 2 - SCORE CARD

---



*The Startup User*

$\mu$

8

.

10

$\lambda$

## ROUND 2 - SCORE CARD

---



*The Student*

## ROUND 2 - SCORE CARD

---



*The Student*

$\mu$

8

.

10

$\lambda$

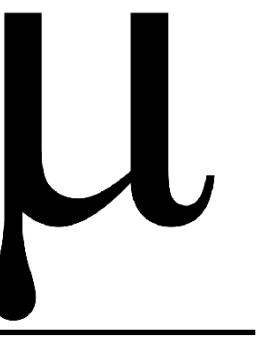
## ROUND 3: TESTABILITY - MICROSERVICES

---



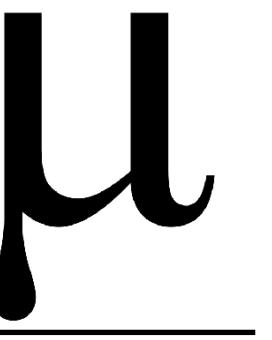
## ROUND 3: TESTABILITY - MICROSERVICES

---



## ROUND 3: TESTABILITY - MICROSERVICES

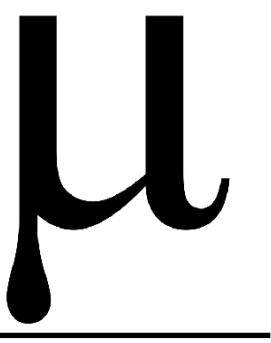
---



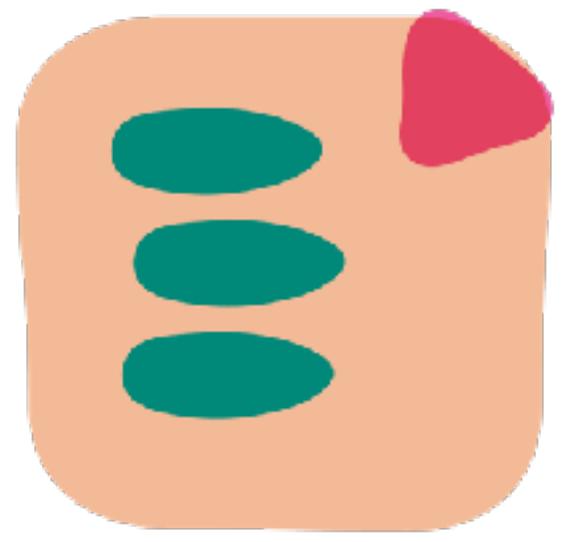
*Does it work in isolation?*

## ROUND 3: TESTABILITY - MICROSERVICES

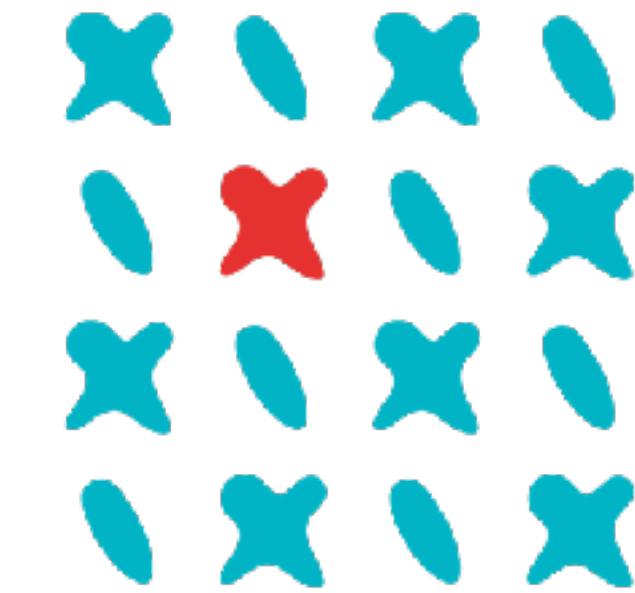
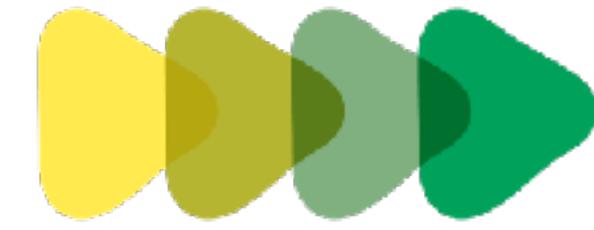
---



*Does it work in isolation?*

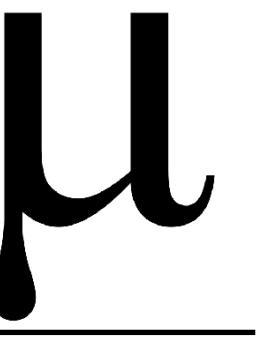


*App code*



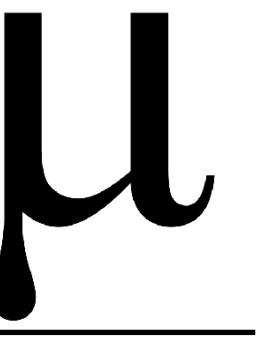
## ROUND 3: TESTABILITY - MICROSERVICES

---



## ROUND 3: TESTABILITY - MICROSERVICES

---

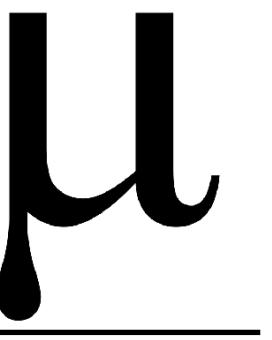


*Does it work on my machine?*

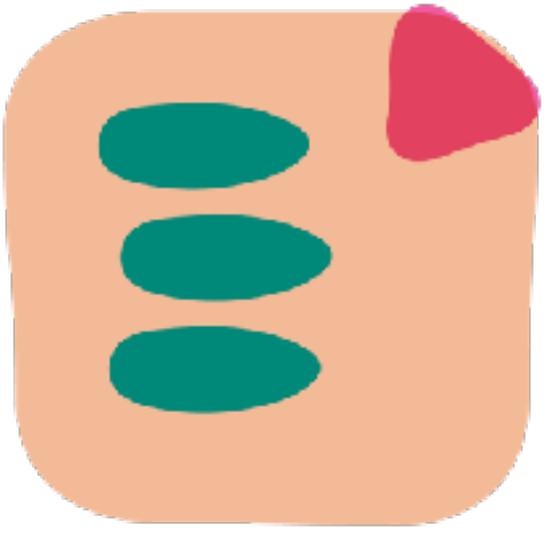


## ROUND 3: TESTABILITY - MICROSERVICES

---

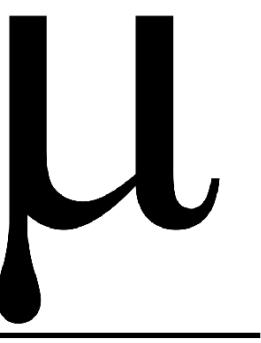


*Does it work on my machine?*

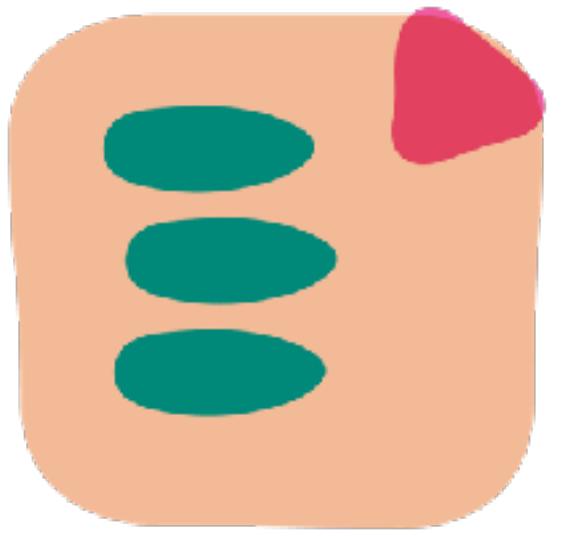


*App code*

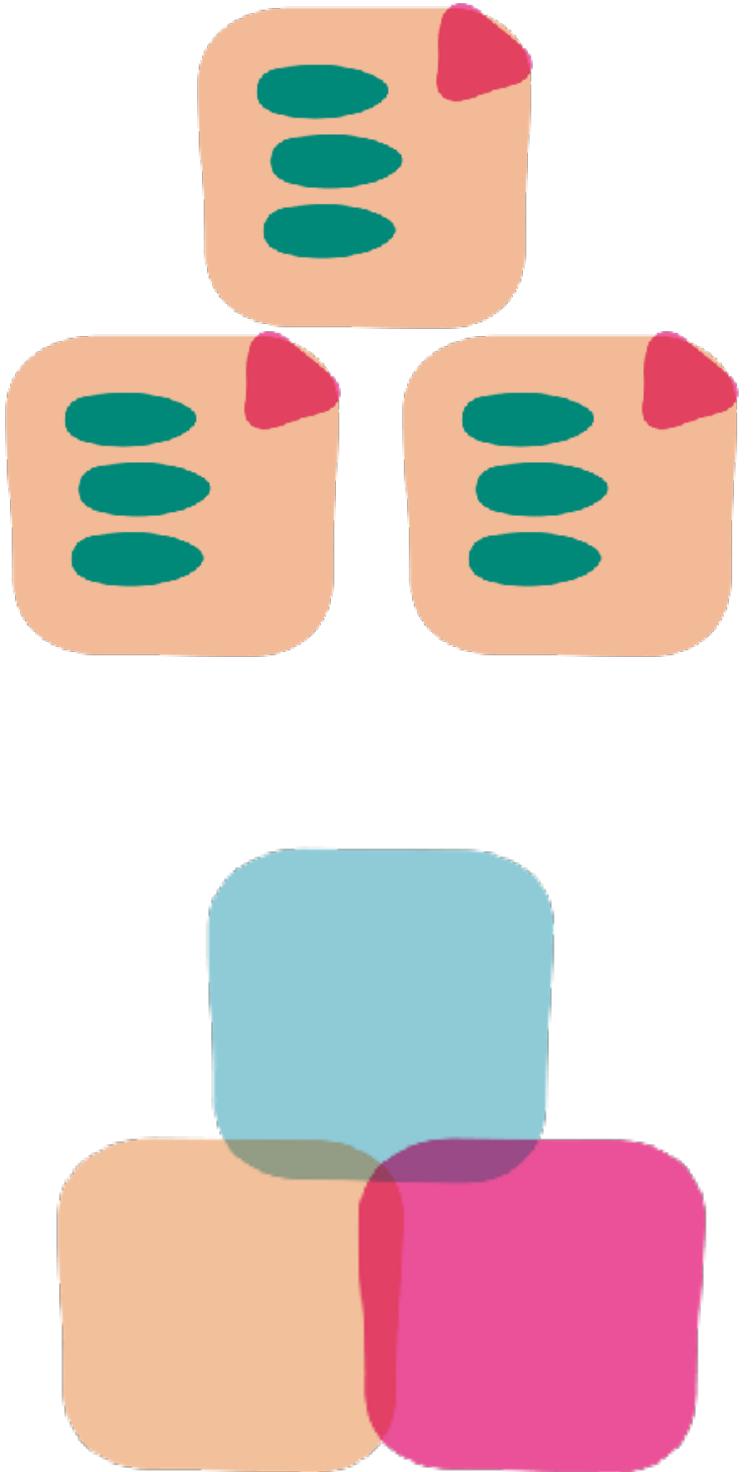
## ROUND 3: TESTABILITY - MICROSERVICES



*Does it work on my machine?*

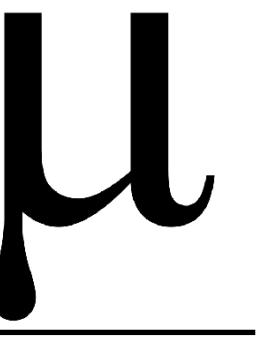


*App code*



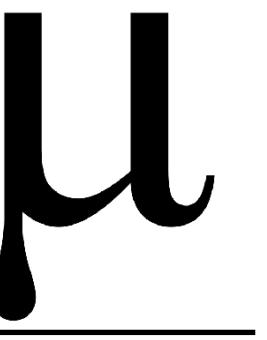
## ROUND 3: TESTABILITY - MICROSERVICES

---



## ROUND 3: TESTABILITY - MICROSERVICES

---

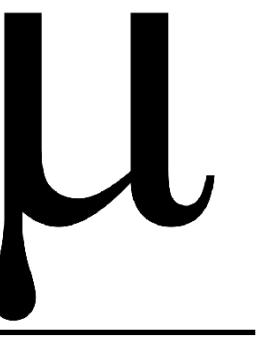


*Does it integrate?*



## ROUND 3: TESTABILITY - MICROSERVICES

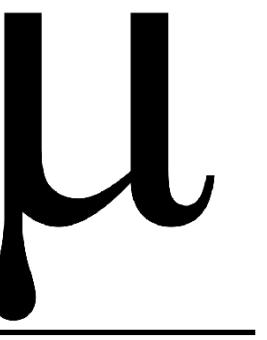
---



*Does it integrate?*

## ROUND 3: TESTABILITY - MICROSERVICES

---

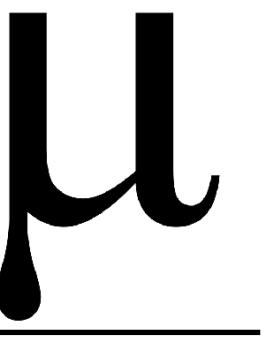


*Does it integrate?*

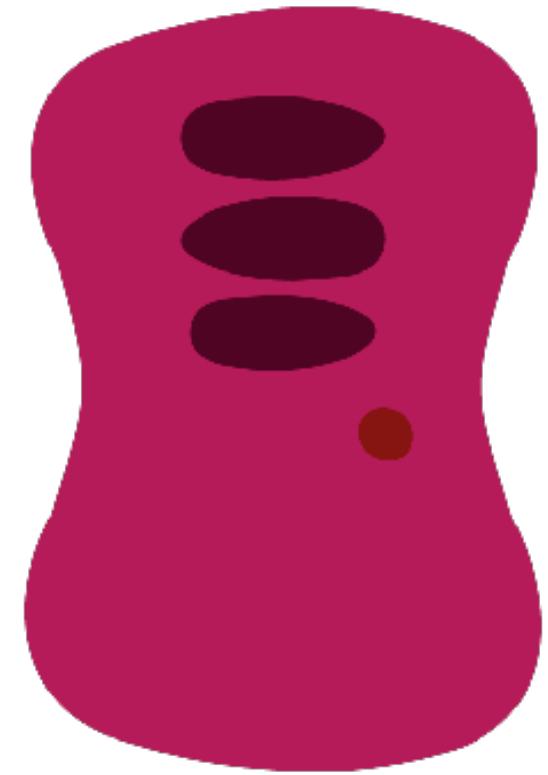


## ROUND 3: TESTABILITY - MICROSERVICES

---

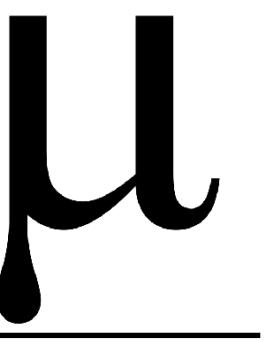


*Does it integrate?*

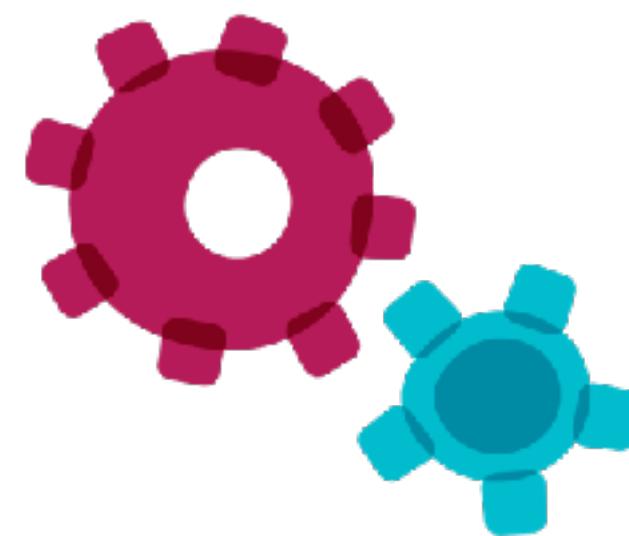
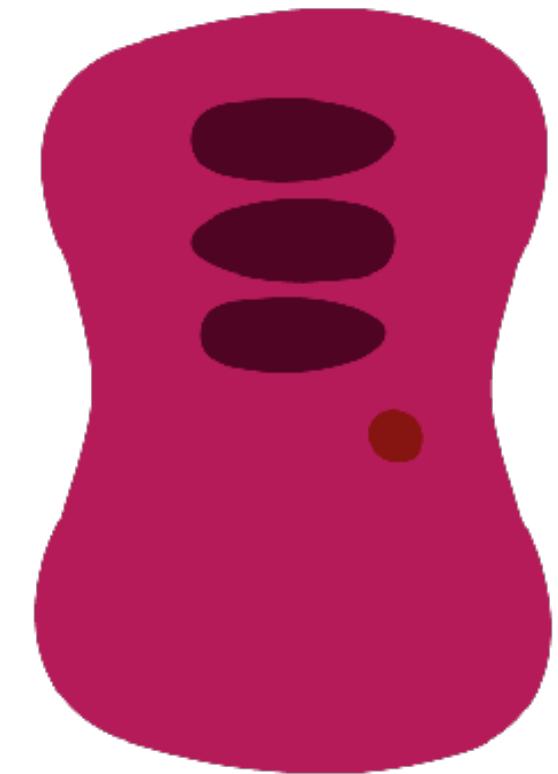


## ROUND 3: TESTABILITY - MICROSERVICES

---



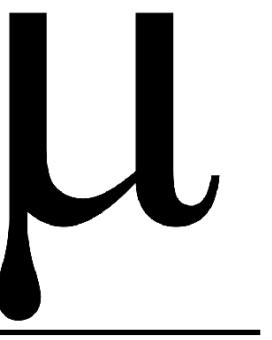
*Does it integrate?*



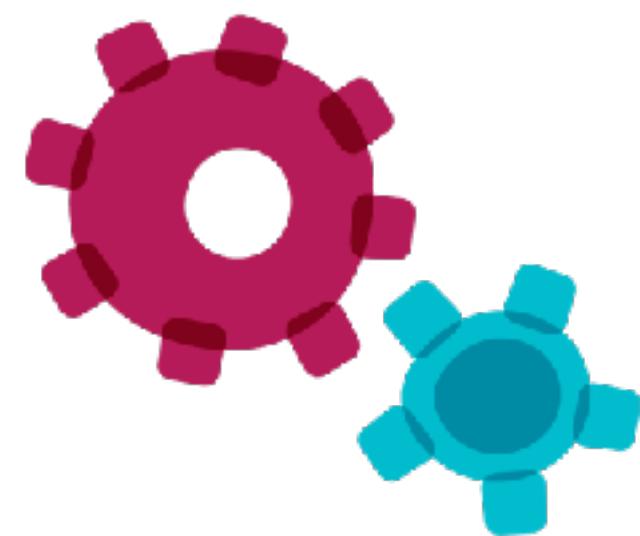
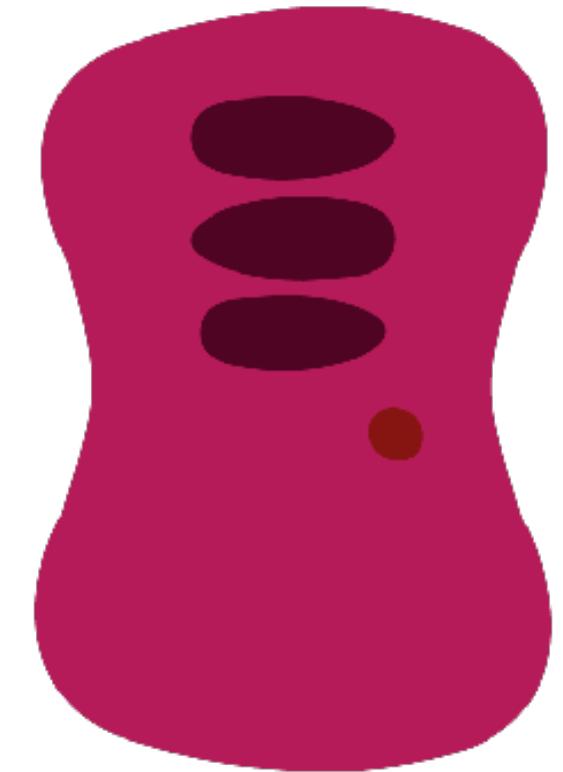
*Provisioning*

## ROUND 3: TESTABILITY - MICROSERVICES

---



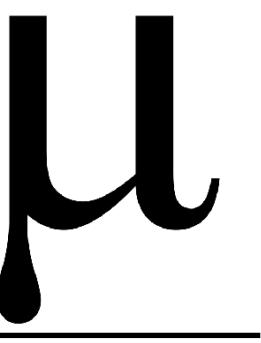
*Does it integrate?*



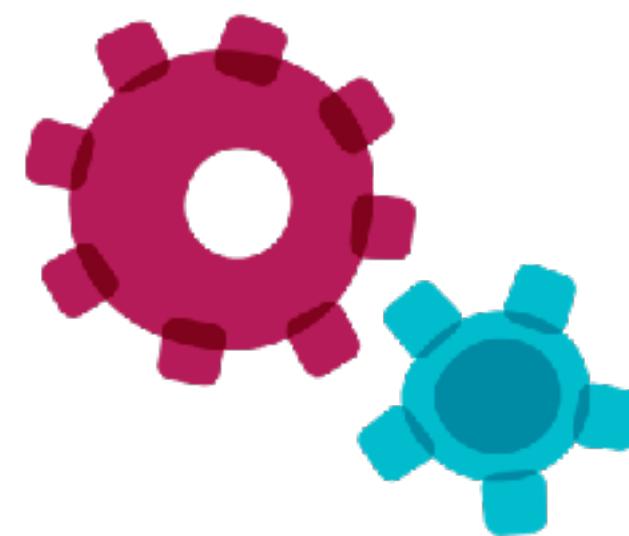
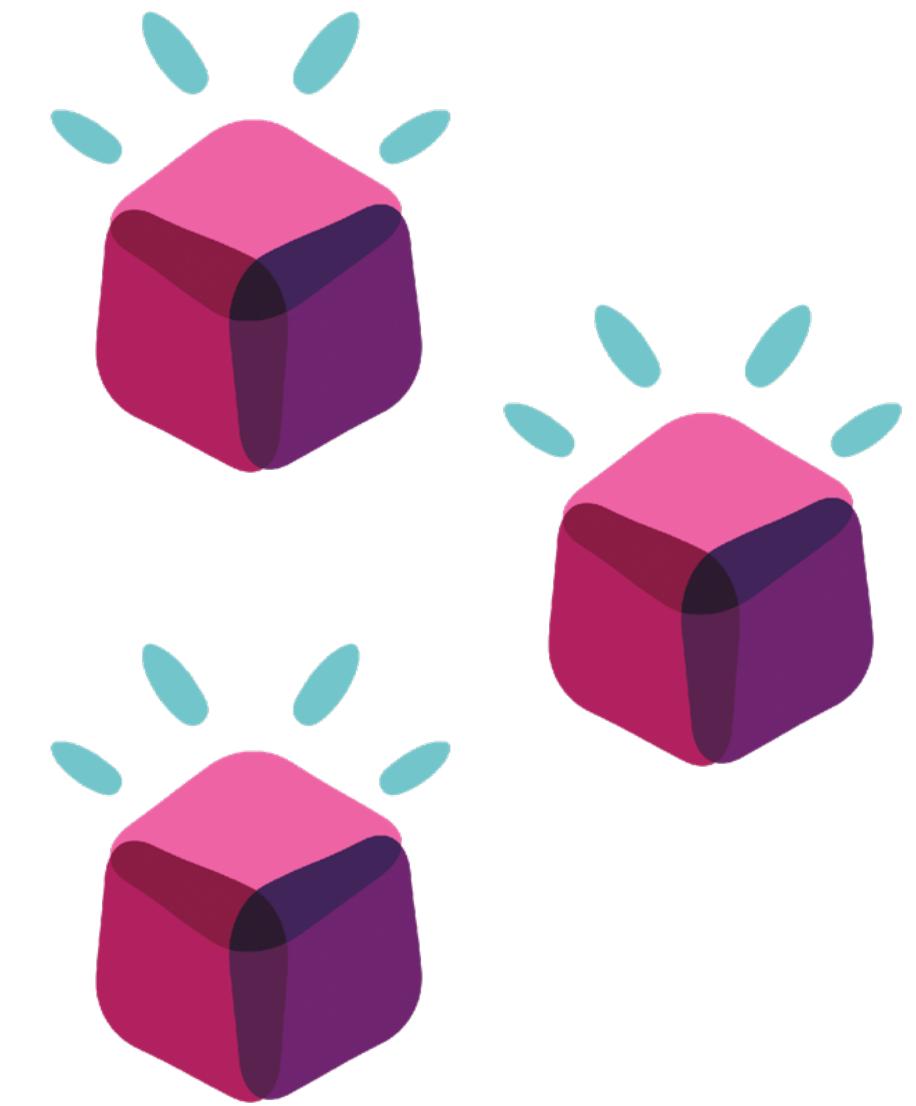
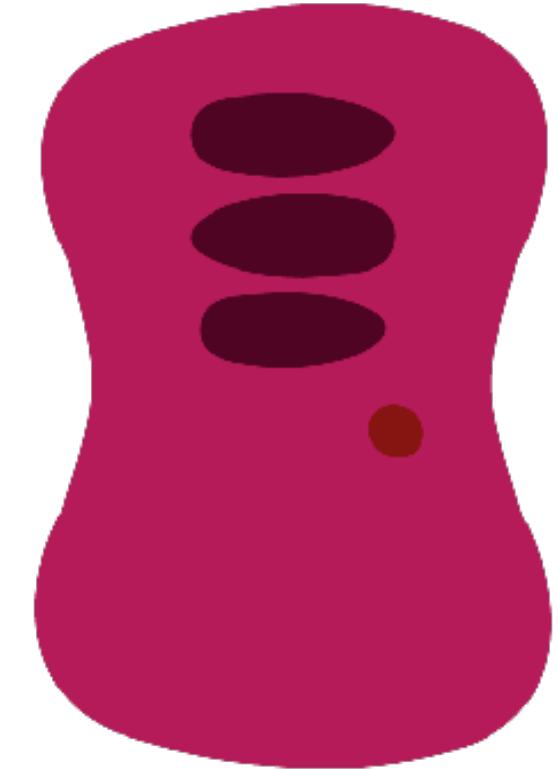
*Provisioning*

## ROUND 3: TESTABILITY - MICROSERVICES

---

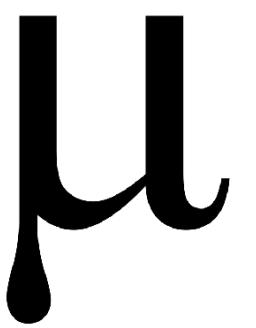


*Does it integrate?*

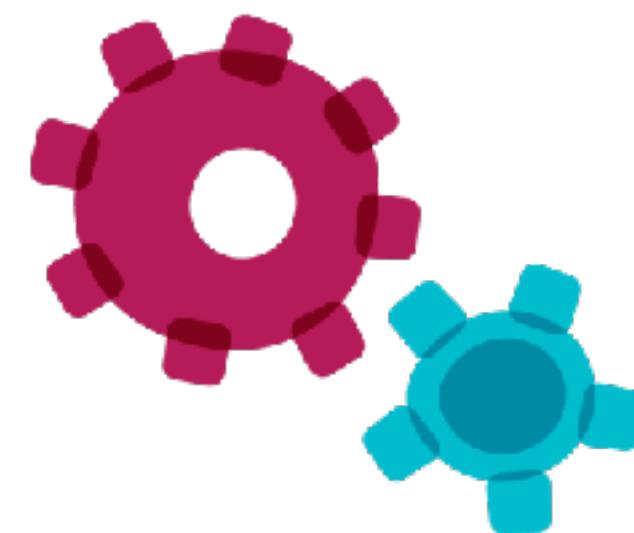
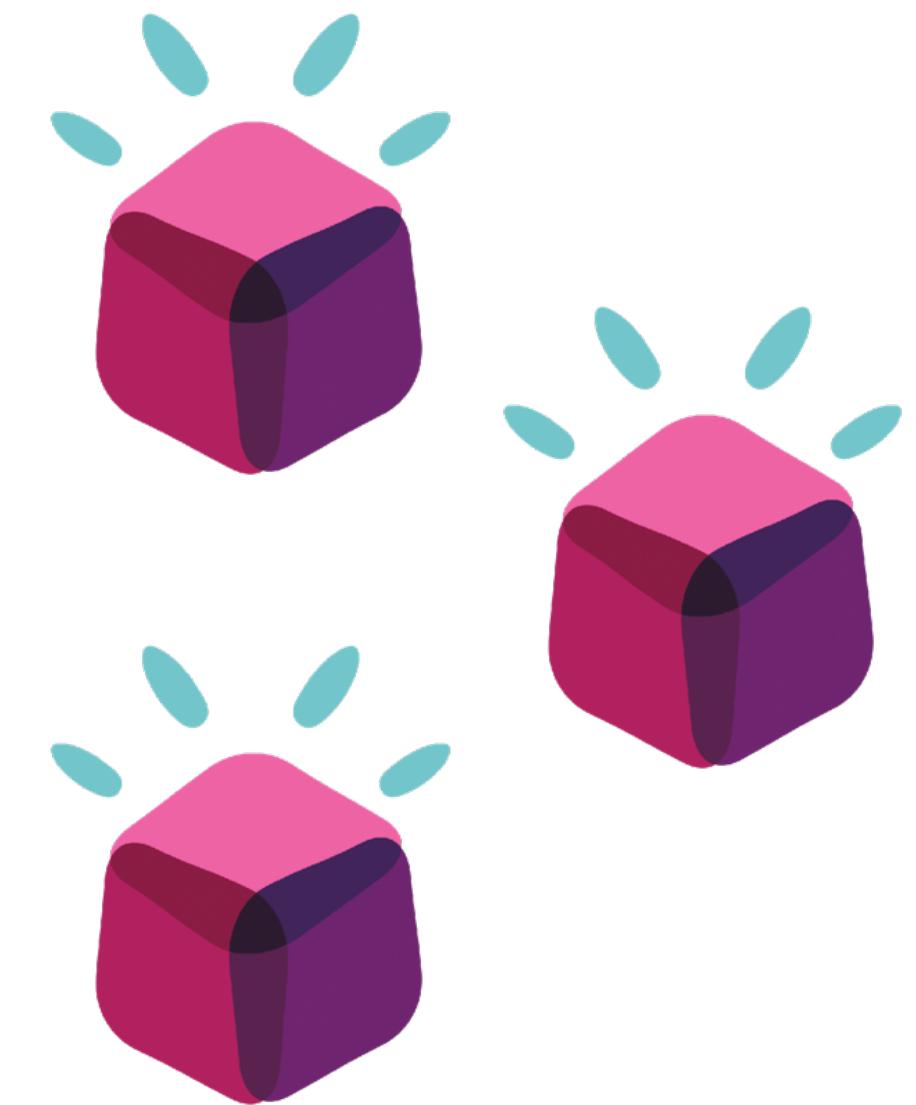
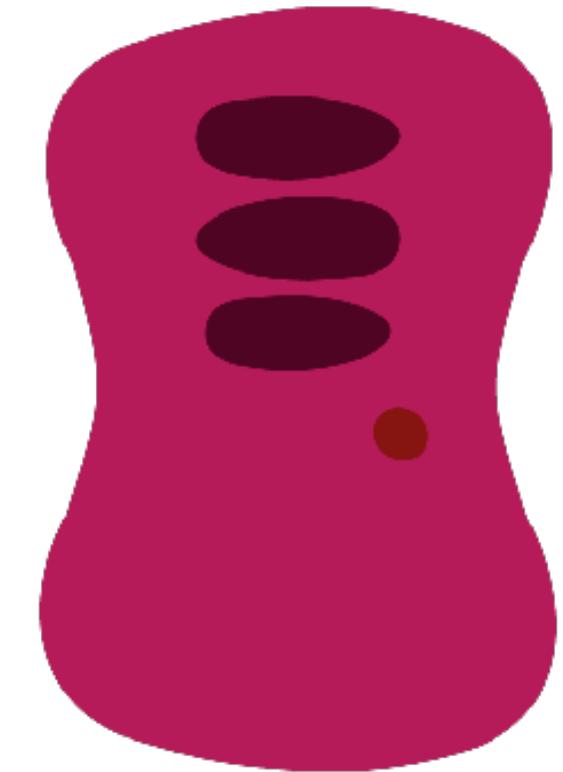


*Provisioning*

## ROUND 3: TESTABILITY - MICROSERVICES

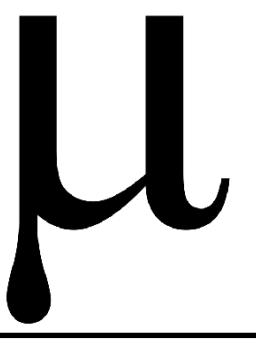


*Does it integrate?*

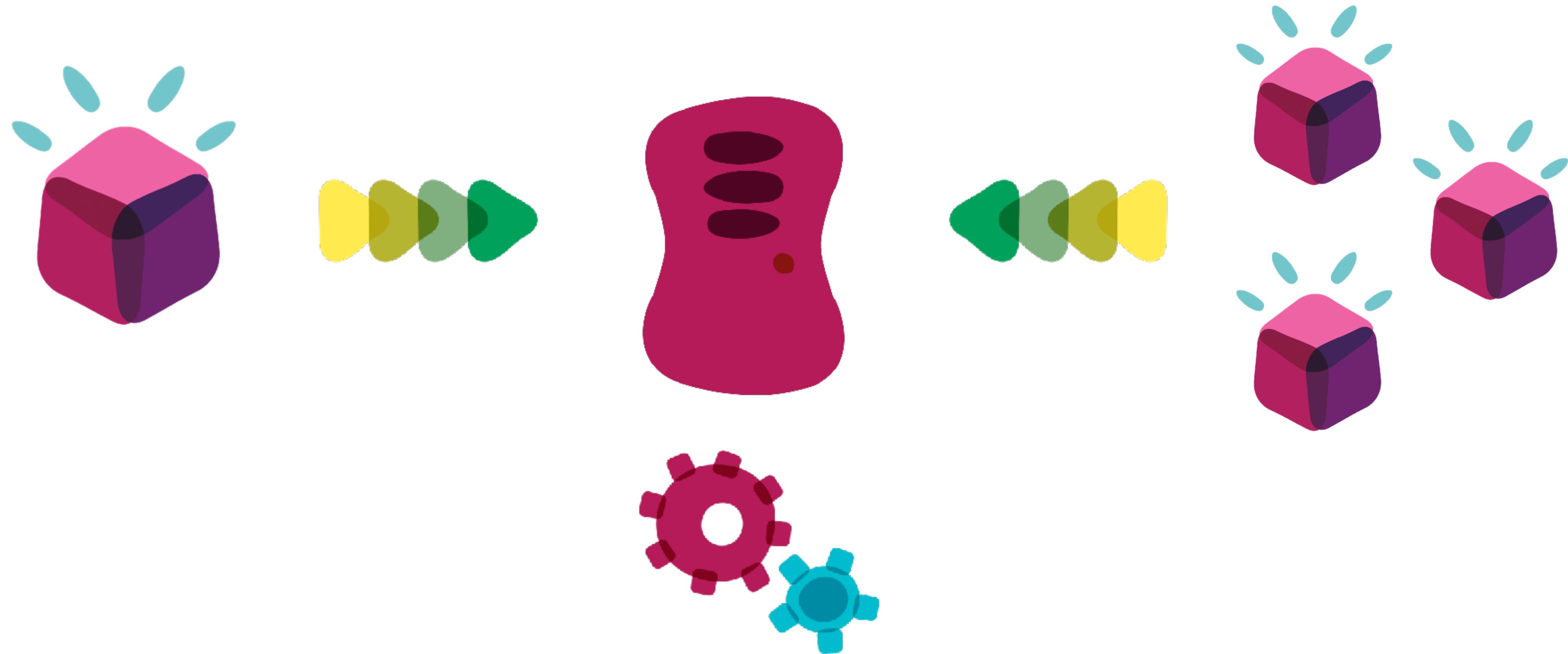


*Provisioning*

## ROUND 3: TESTABILITY - MICROSERVICES



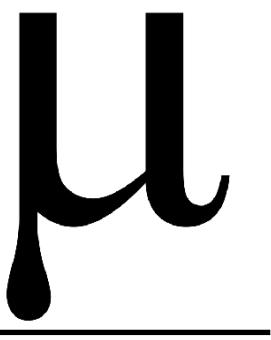
*Does it integrate?*



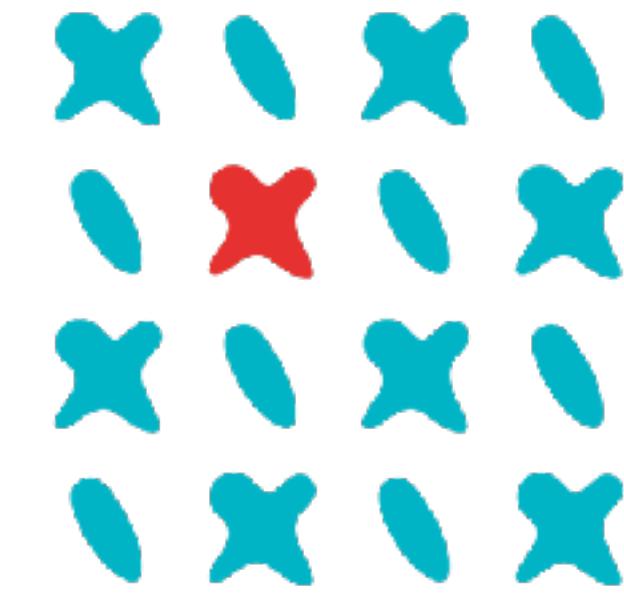
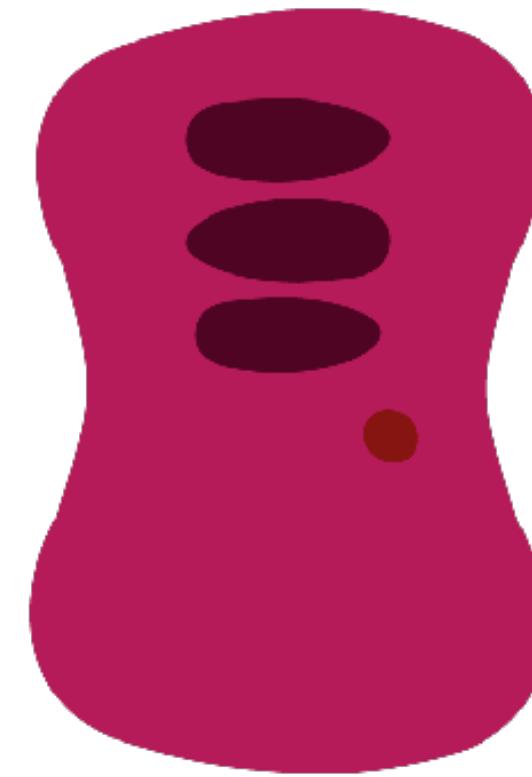
*Provisioning*

## ROUND 3: TESTABILITY - MICROSERVICES

---



*Does it integrate?*



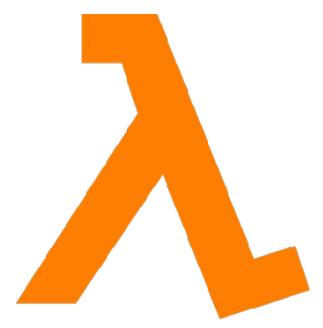
## ROUND 3: TESTABILITY - SERVERLESS

---



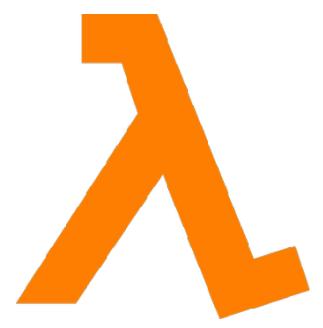
## ROUND 3: TESTABILITY - SERVERLESS

---



## ROUND 3: TESTABILITY - SERVERLESS

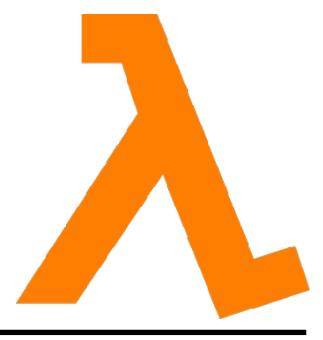
---



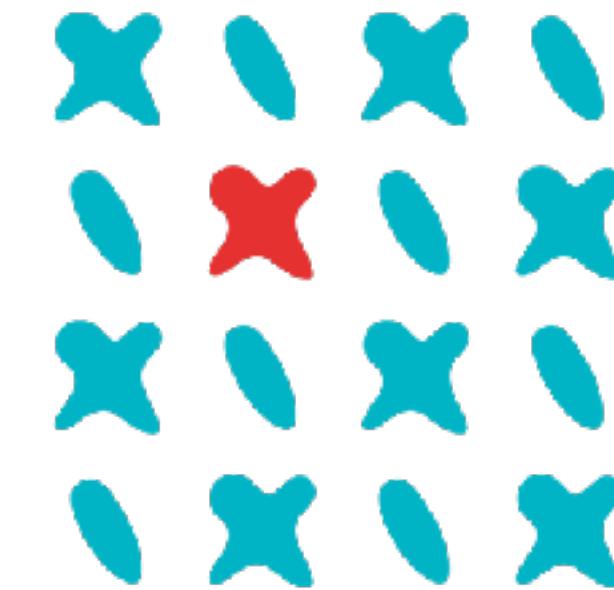
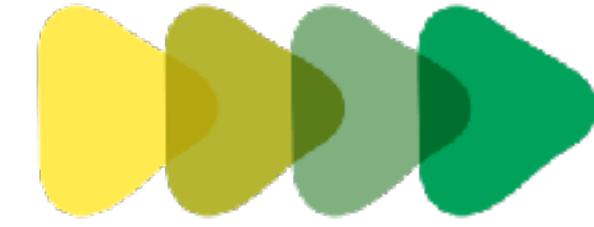
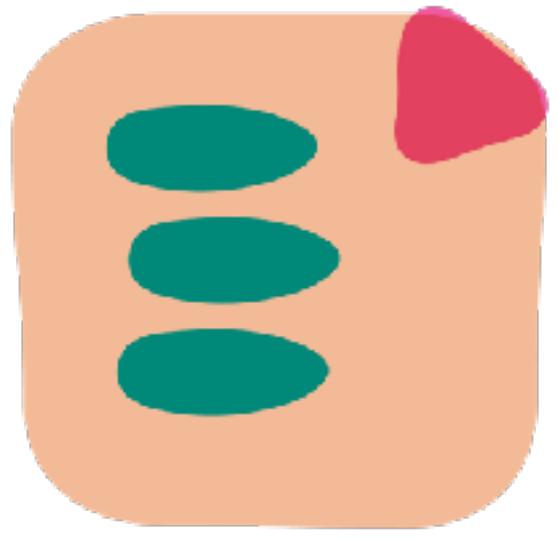
*Does it work in isolation?*

## ROUND 3: TESTABILITY - SERVERLESS

---



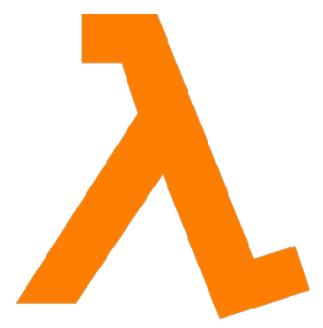
*Does it work in isolation?*



*Function code*

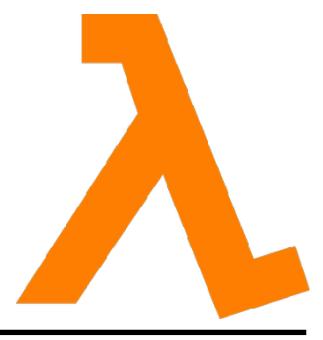
## ROUND 3: TESTABILITY - SERVERLESS

---



## ROUND 3: TESTABILITY - SERVERLESS

---

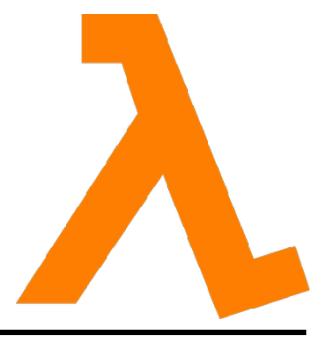


*Does it work on my machine?*

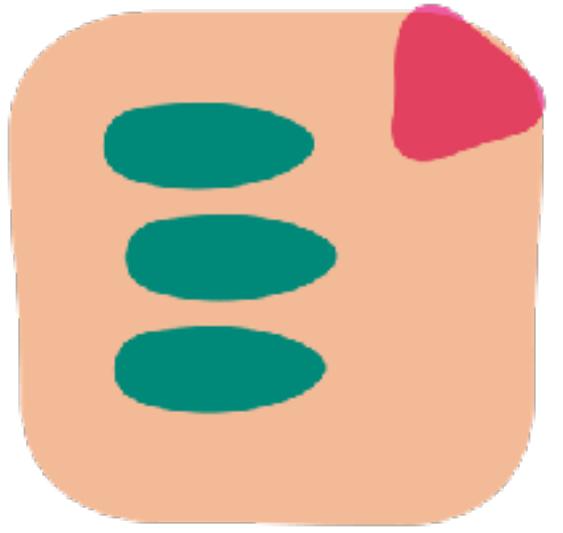


## ROUND 3: TESTABILITY - SERVERLESS

---

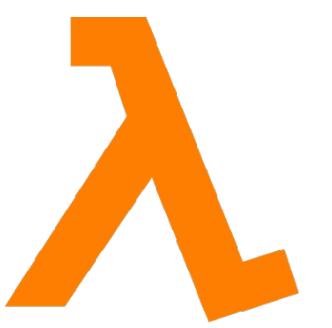


*Does it work on my machine?*

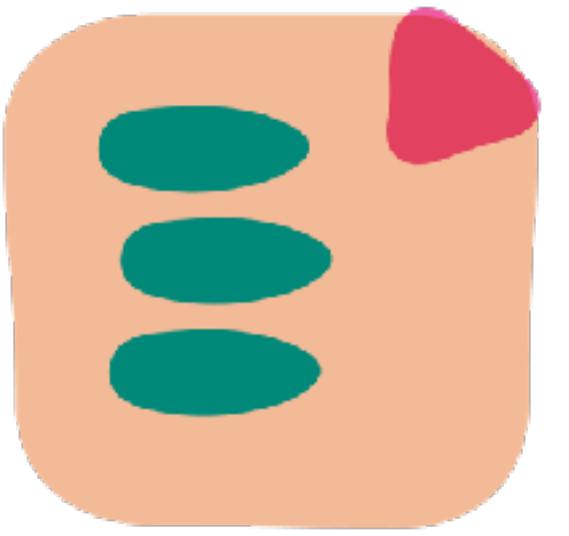


*Function code*

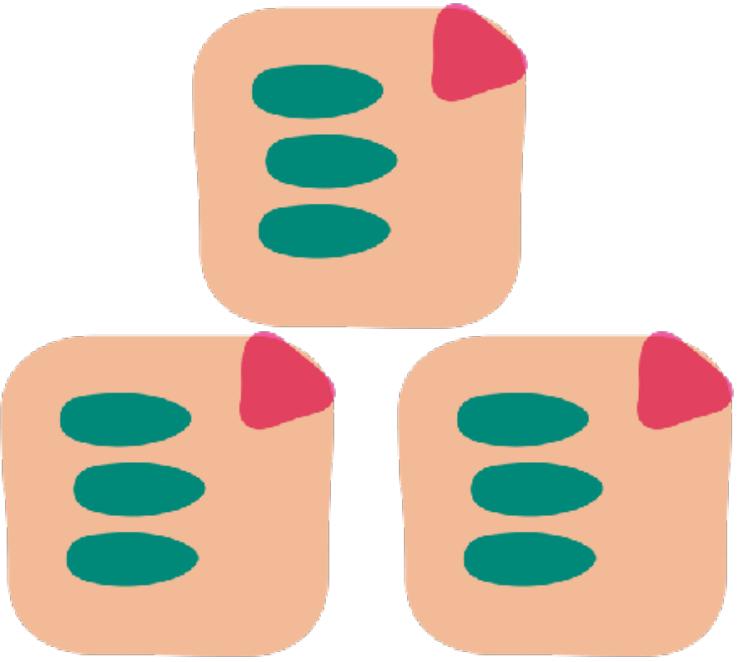
## ROUND 3: TESTABILITY - SERVERLESS



*Does it work on my machine?*

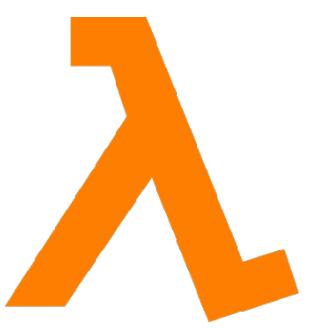


*Function code*

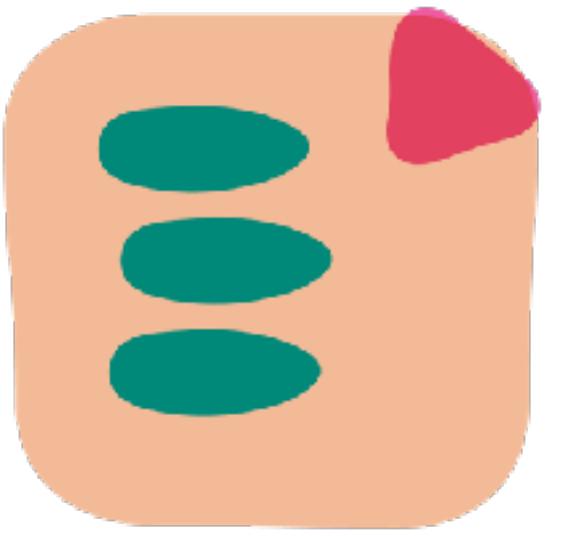


*Emulators*

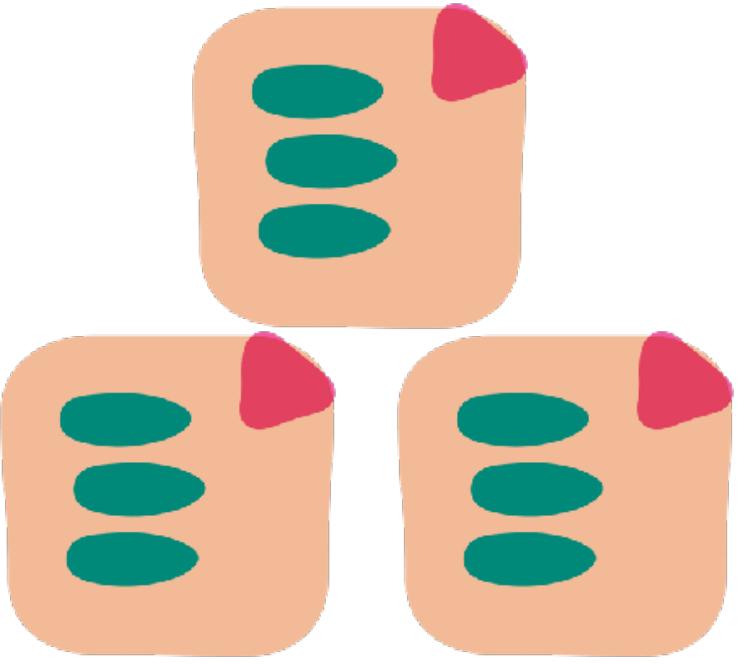
## ROUND 3: TESTABILITY - SERVERLESS



*Does it work on my machine?*



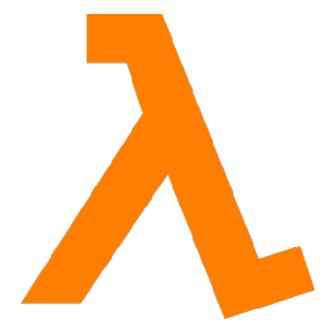
*Function code*



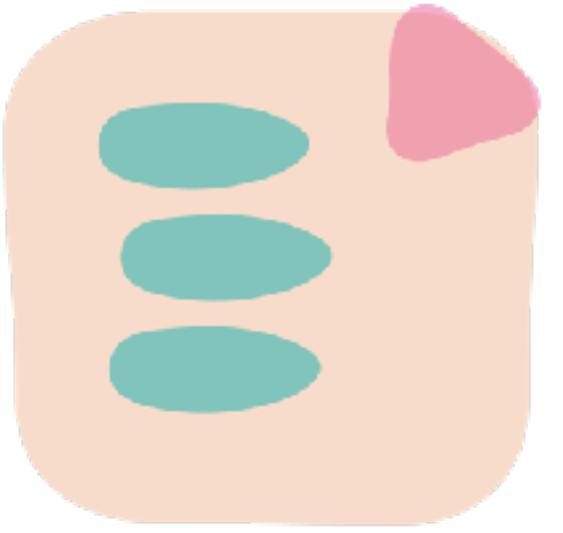
*Emulators*

## ROUND 3: TESTABILITY - SERVERLESS

---

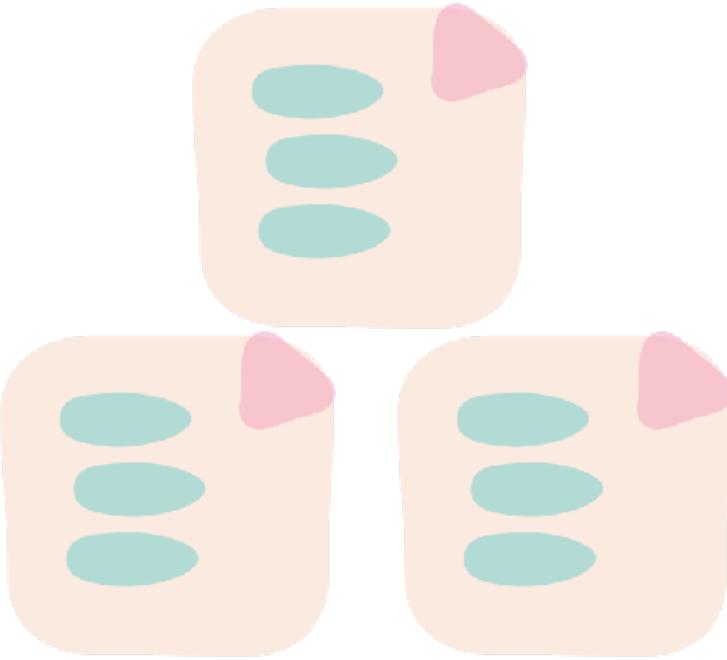


*Does it work on my machine?*



*Function code*

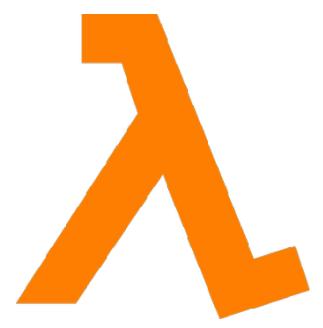
```
plugins:  
- serverless-offline-sns  
- serverless-dynamodb-local  
- serverless-offline
```



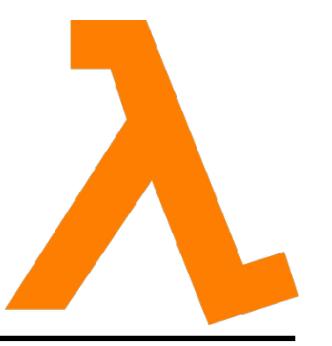
*Emulators*

## ROUND 3: TESTABILITY - SERVERLESS

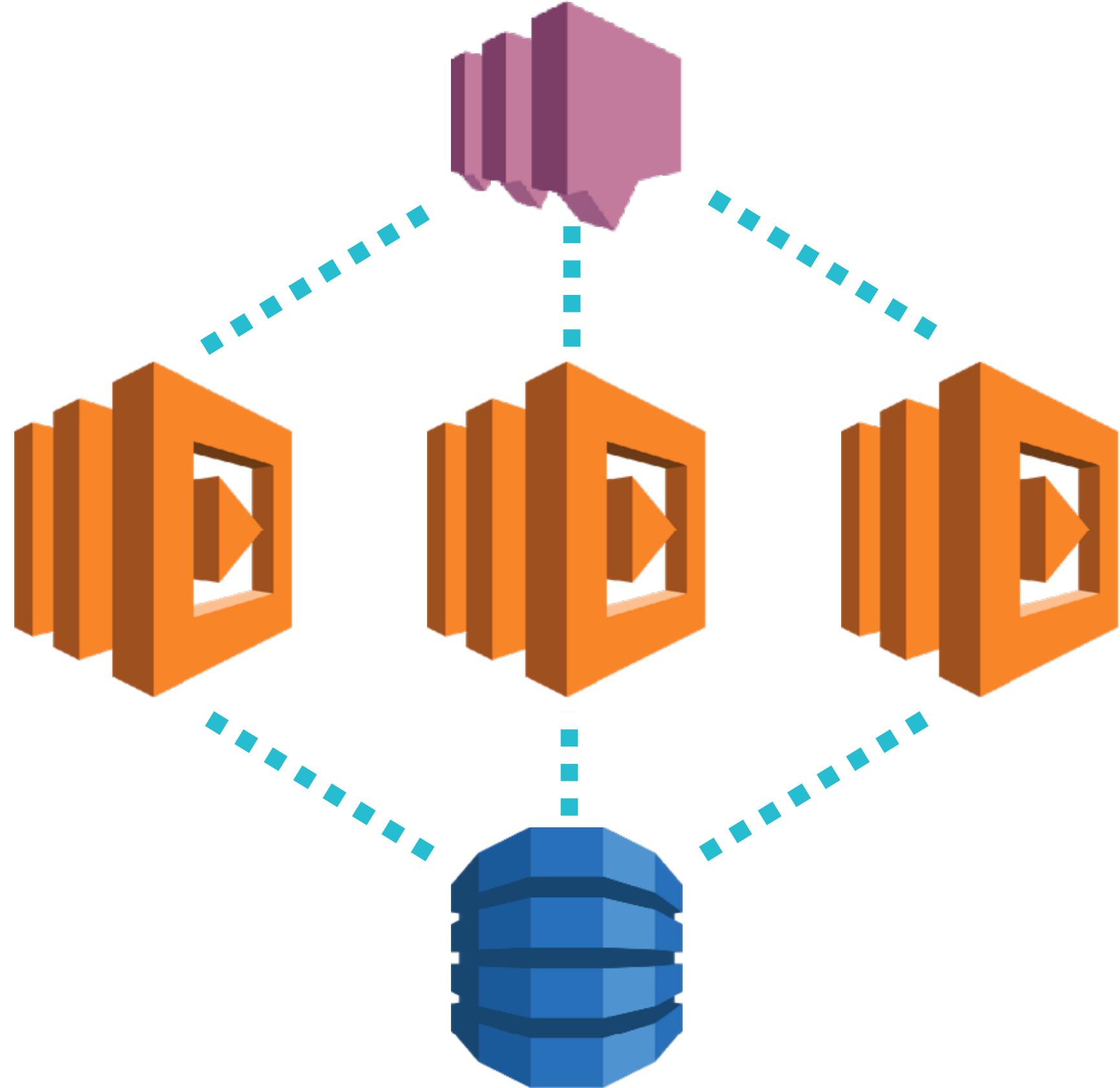
---



## ROUND 3: TESTABILITY - SERVERLESS

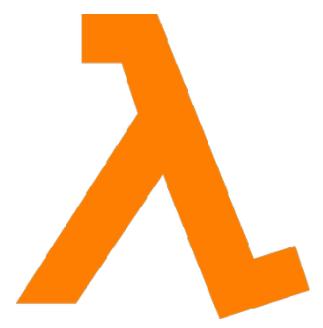


*Does it integrate?*



## ROUND 3: TESTABILITY - SERVERLESS

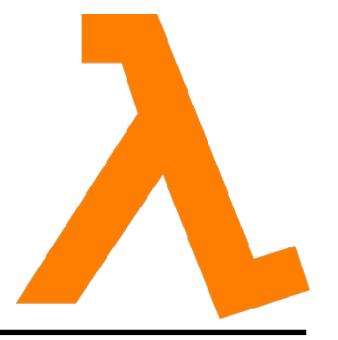
---



*Does it integrate?*

## ROUND 3: TESTABILITY - SERVERLESS

---

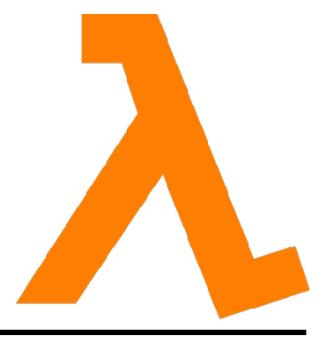


*Does it integrate?*



## ROUND 3: TESTABILITY - SERVERLESS

---

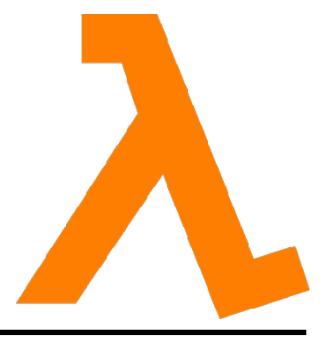


*Does it integrate?*

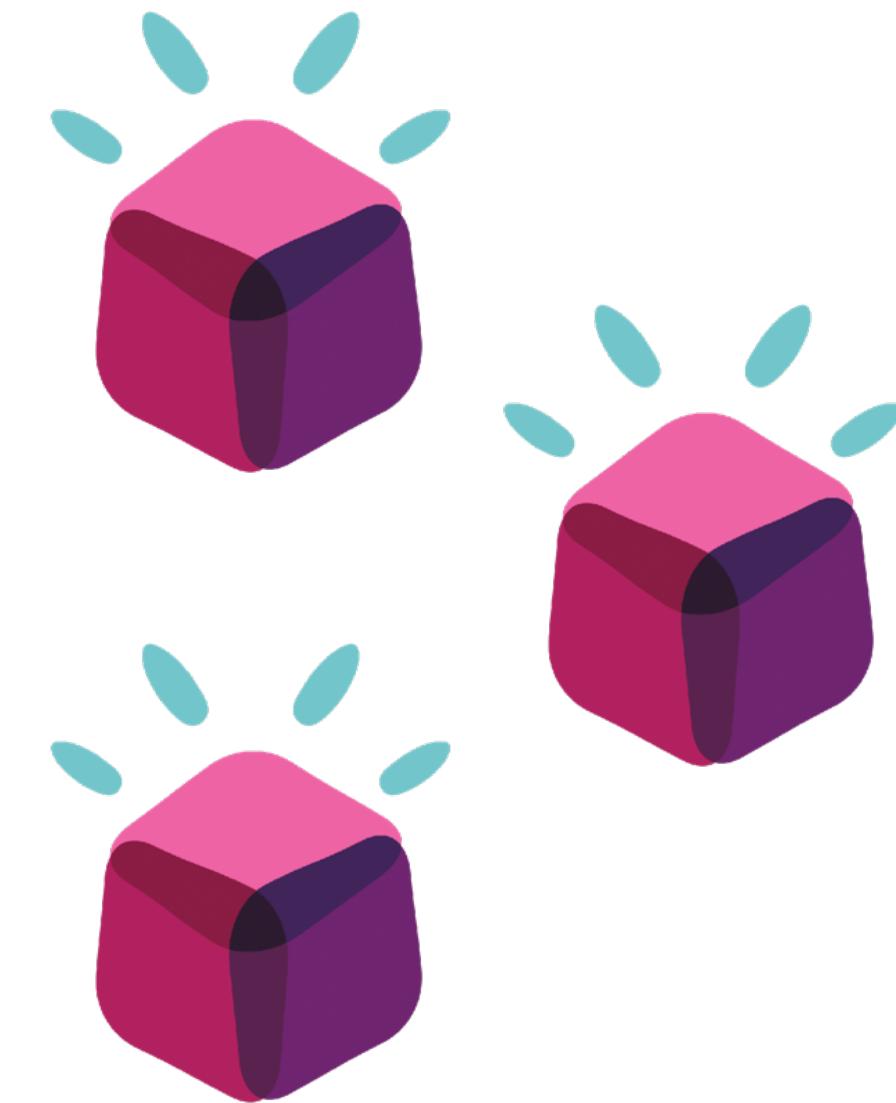


## ROUND 3: TESTABILITY - SERVERLESS

---



*Does it integrate?*



## ROUND 3 - SCORE CARD

---



*The Enterprise User*

## ROUND 3 - SCORE CARD

---



*The Enterprise User*

$\mu$

10

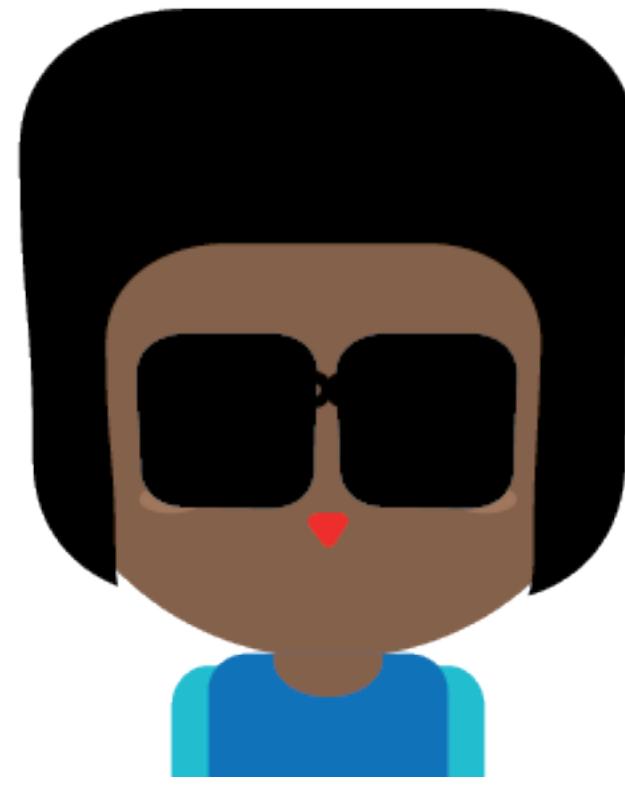
-

8

$\lambda$

## ROUND 3 - SCORE CARD

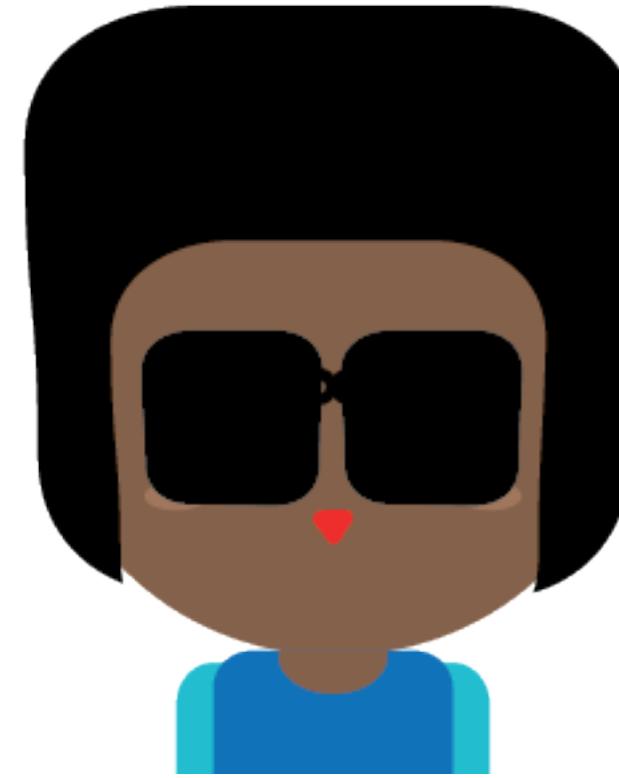
---



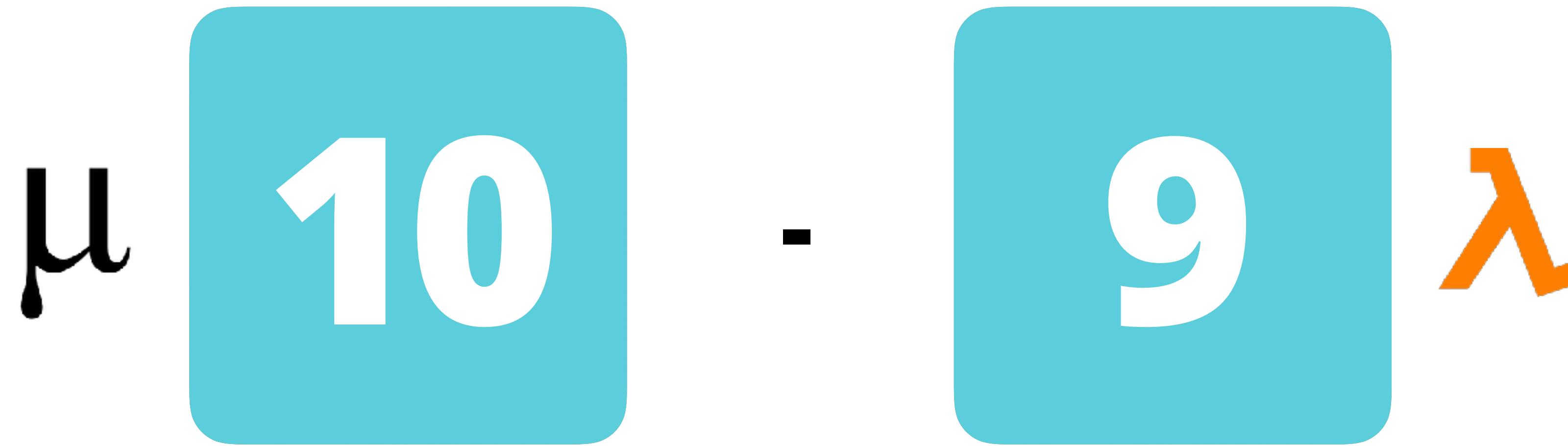
*The Startup User*

## ROUND 3 - SCORE CARD

---



*The Startup User*



## ROUND 3 - SCORE CARD

---



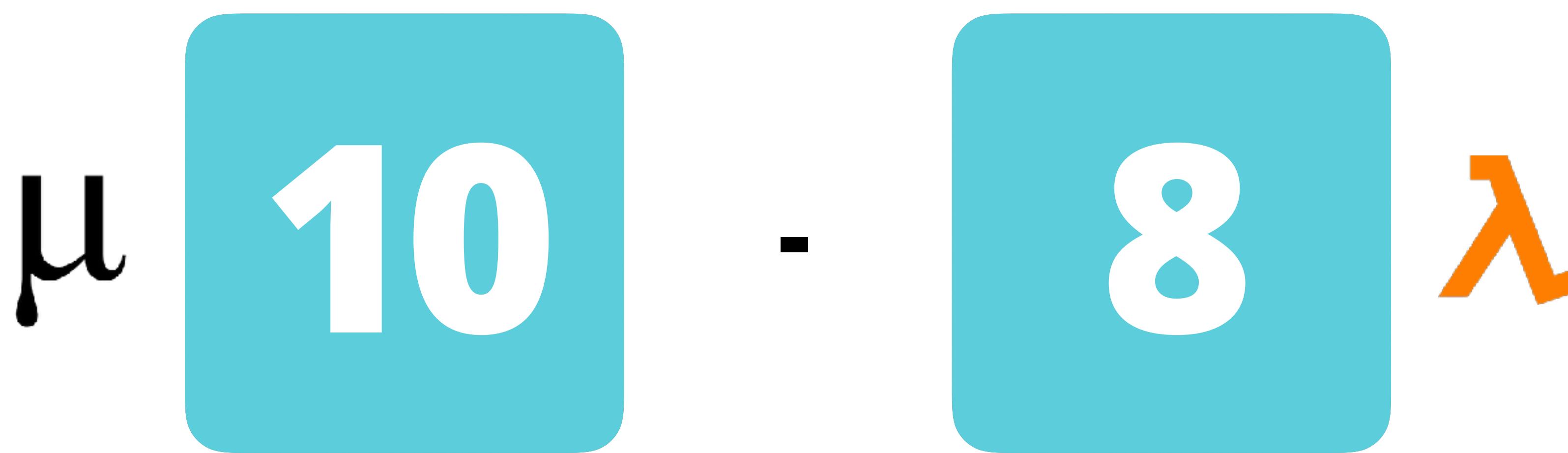
*The Student*

## ROUND 3 - SCORE CARD

---

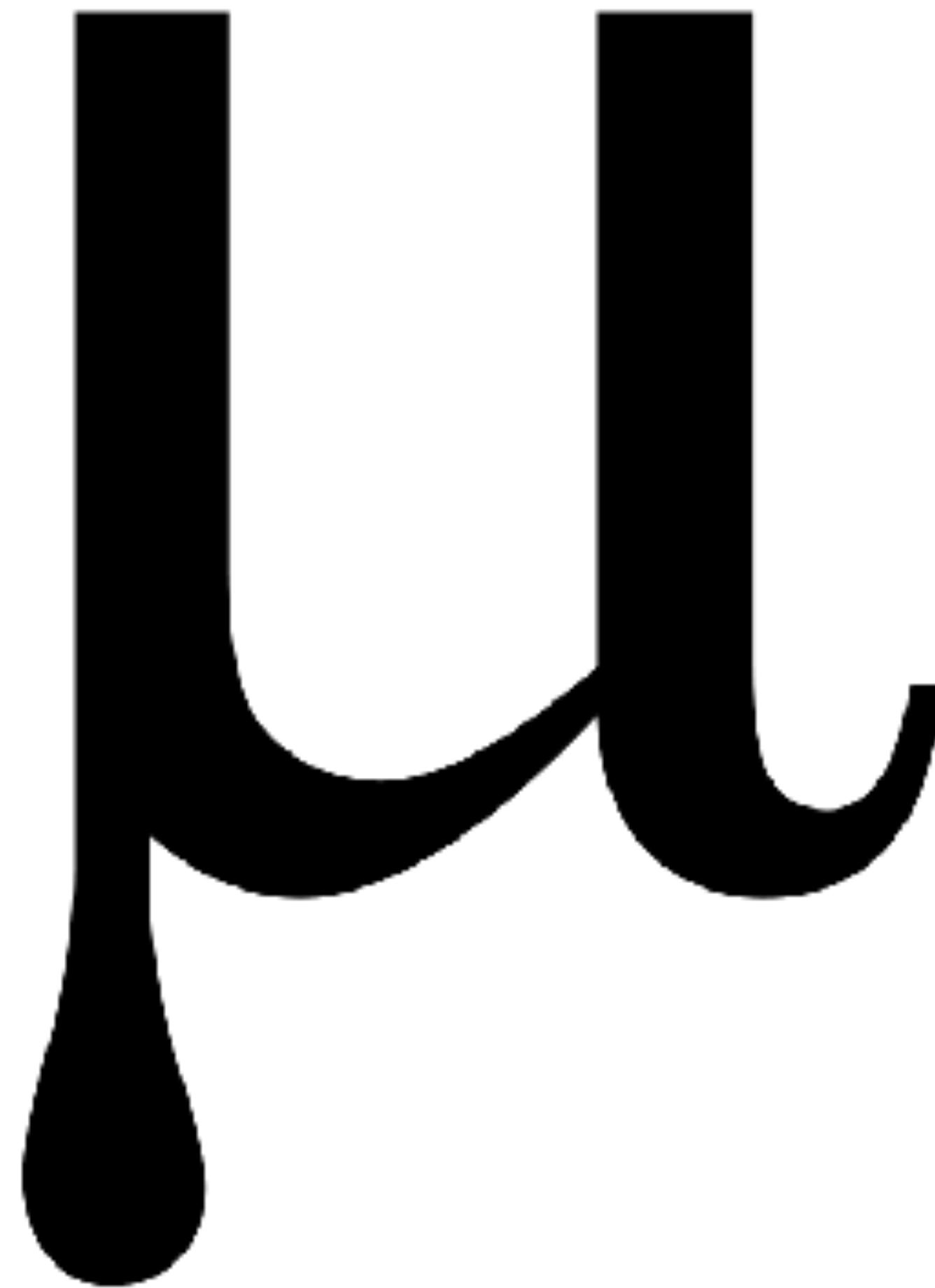


*The Student*



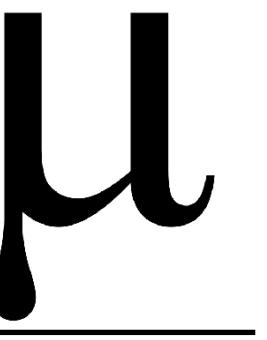
## ROUND 4: MONITORING - MICROSERVICES

---



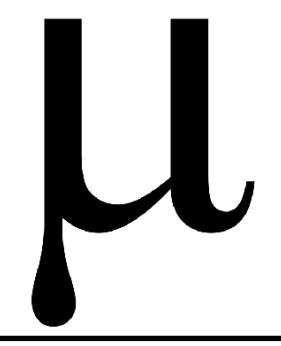
## ROUND 4: MONITORING - MICROSERVICES

---



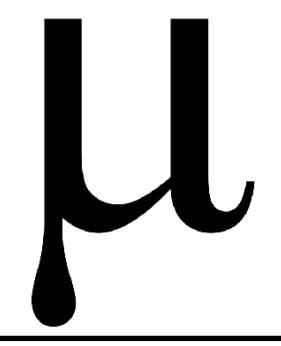
## ROUND 4: MONITORING - MICROSERVICES

---



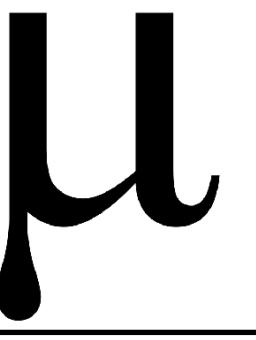
## ROUND 4: MONITORING - MICROSERVICES

---



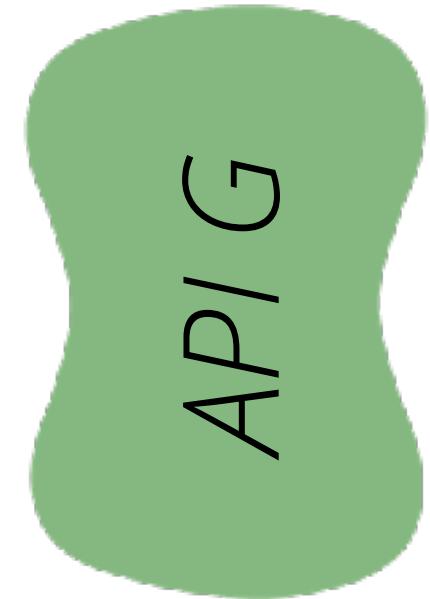
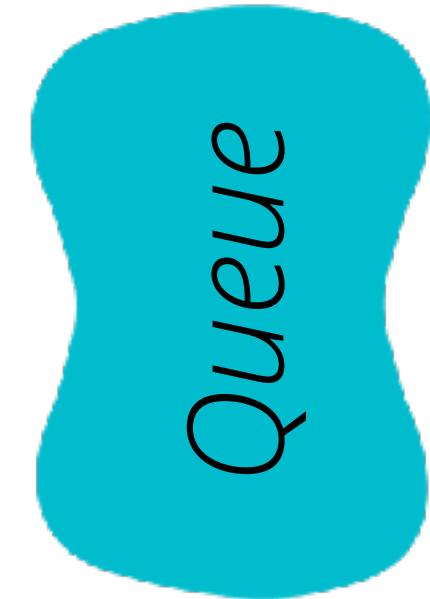
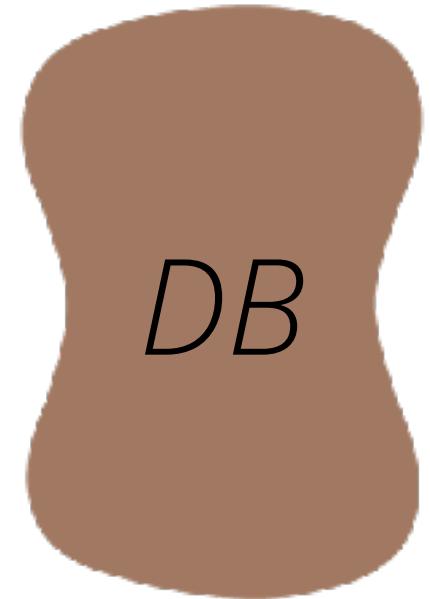
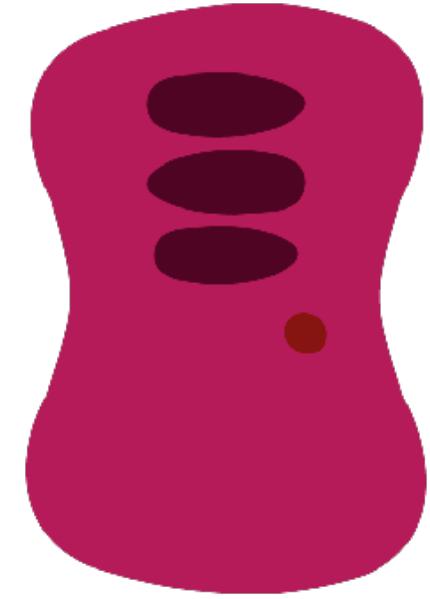
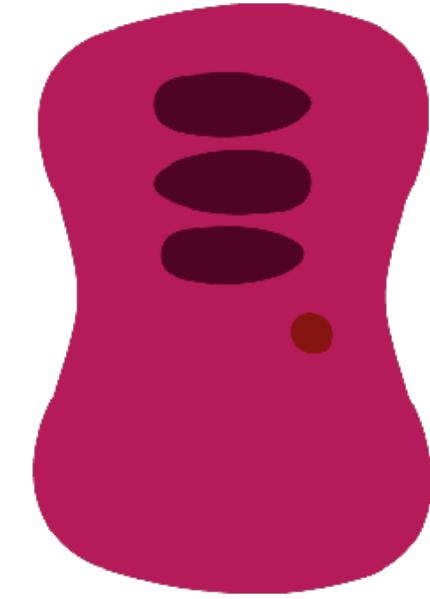
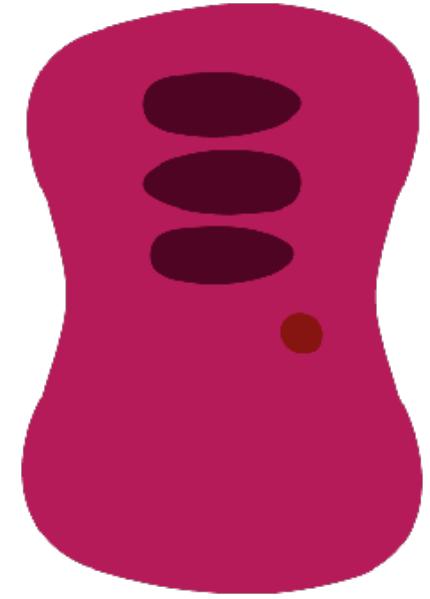
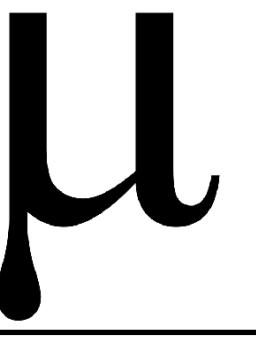
## ROUND 4: MONITORING - MICROSERVICES

---



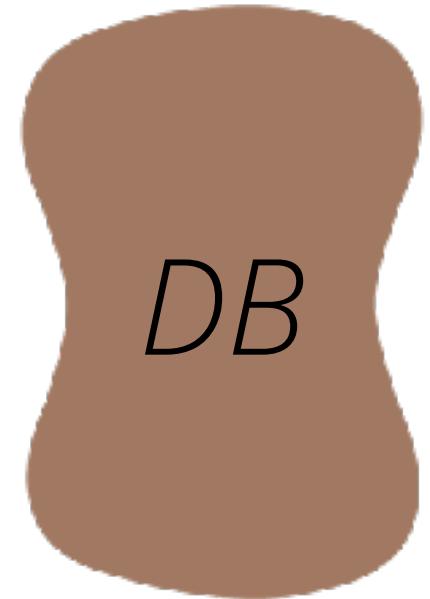
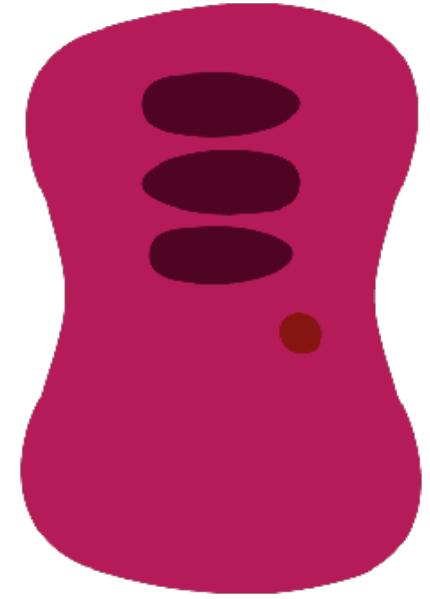
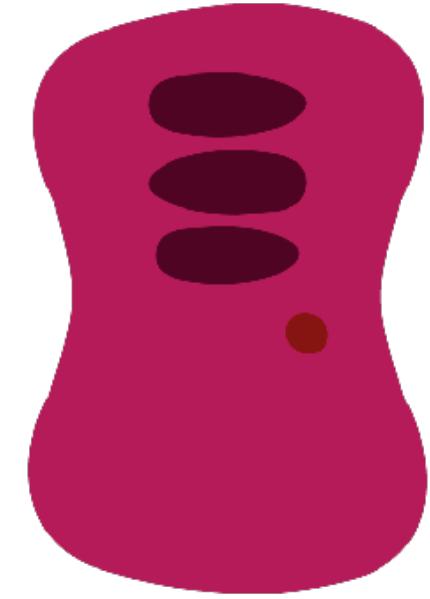
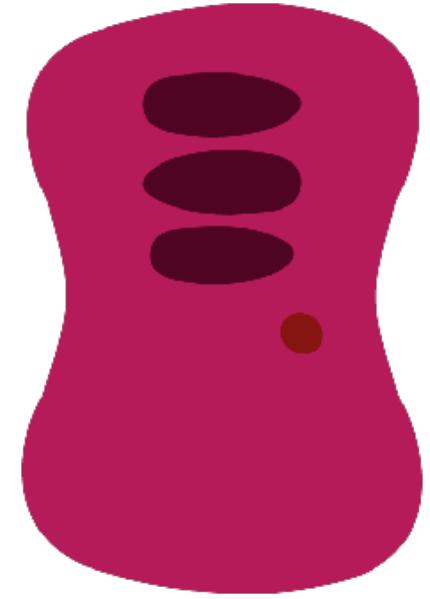
## ROUND 4: MONITORING - MICROSERVICES

---



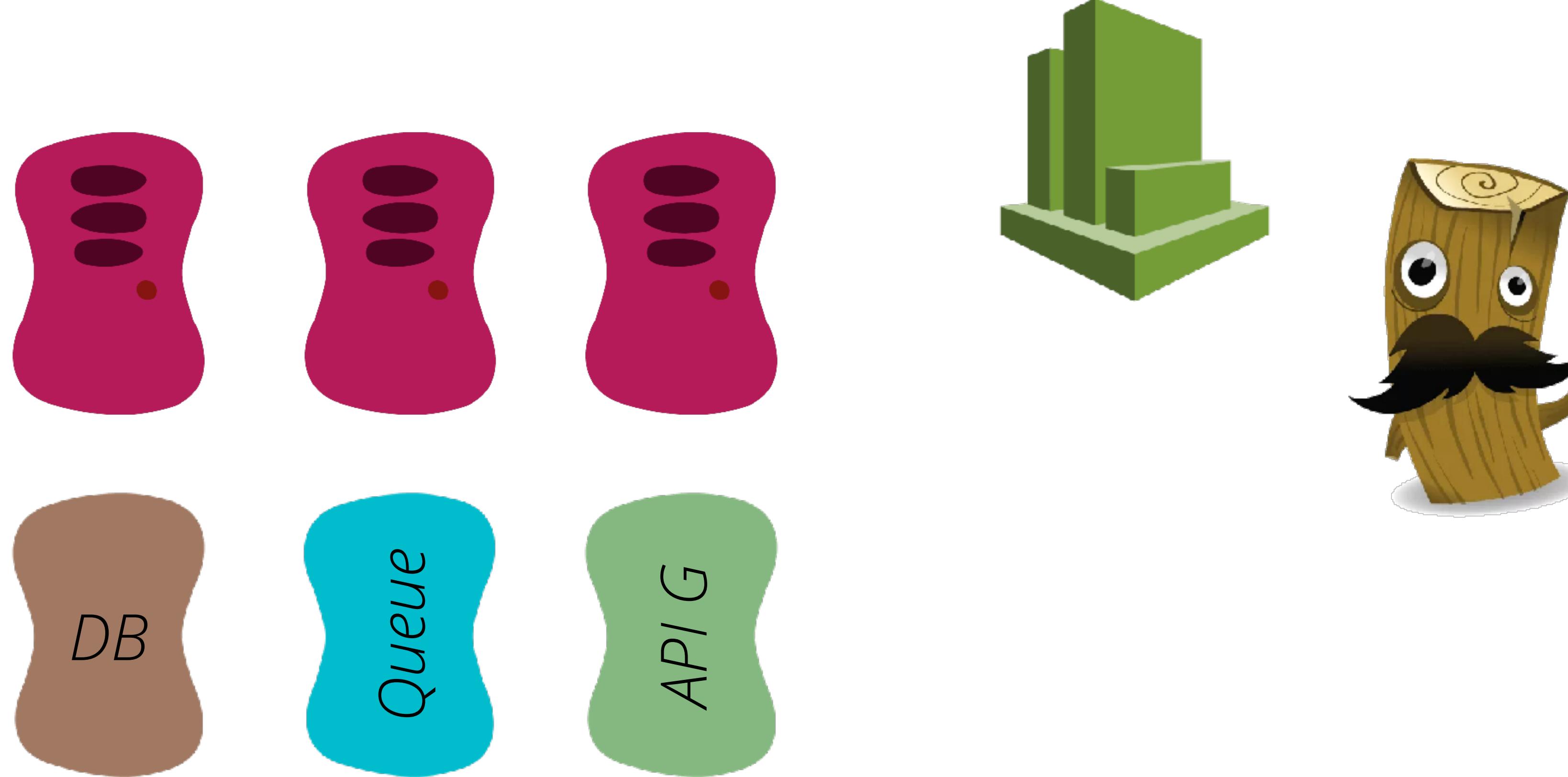
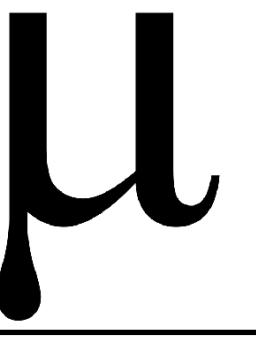
## ROUND 4: MONITORING - MICROSERVICES

---



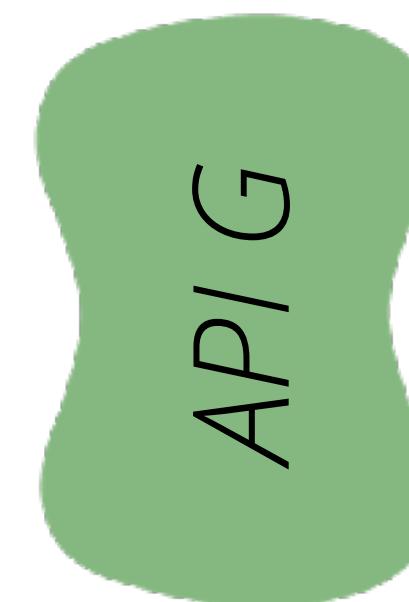
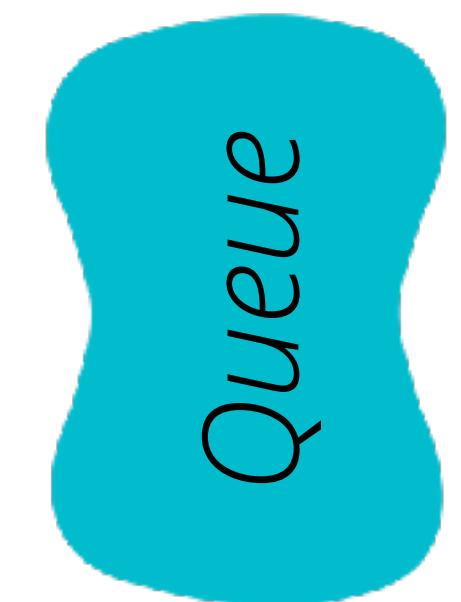
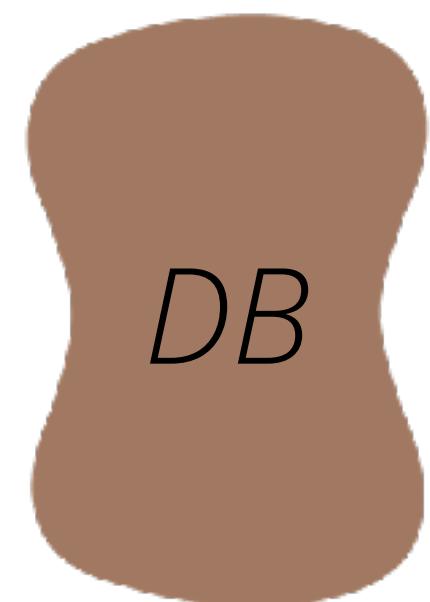
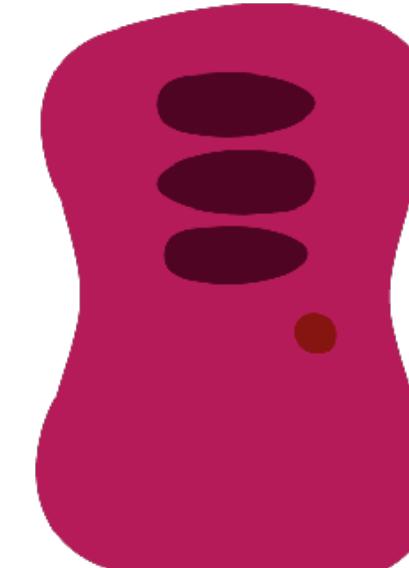
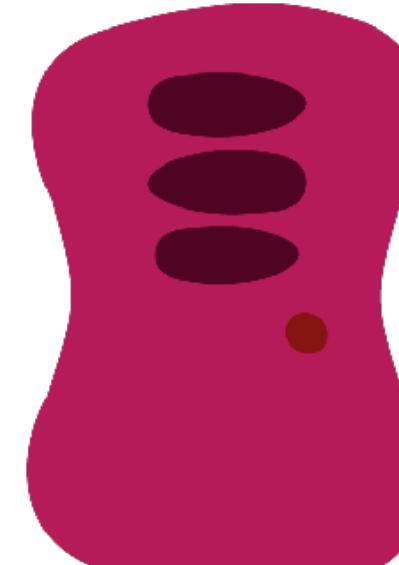
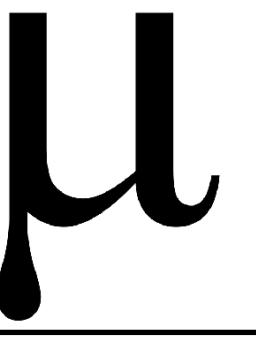
## ROUND 4: MONITORING - MICROSERVICES

---



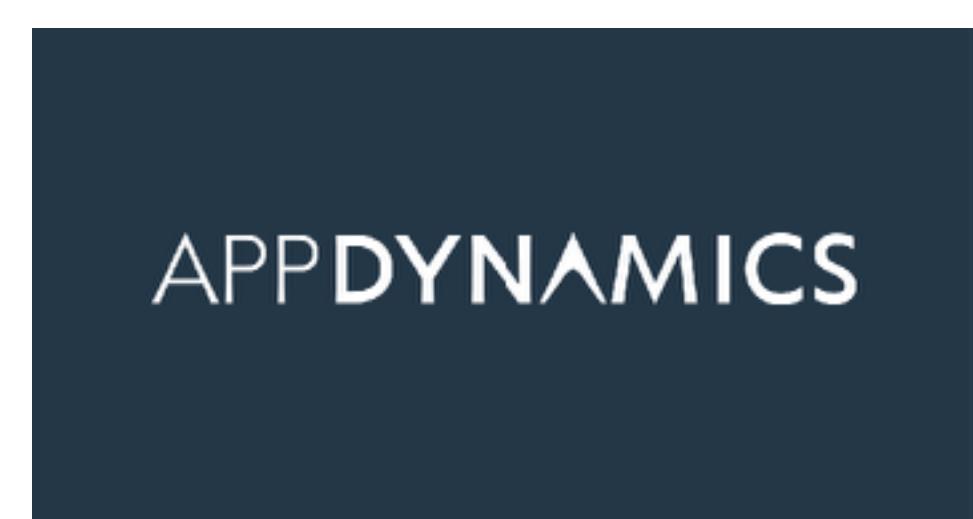
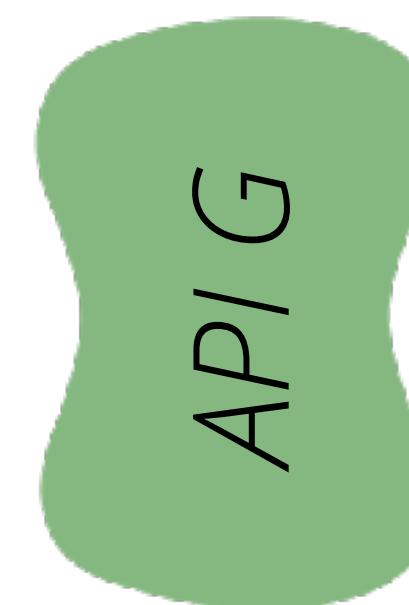
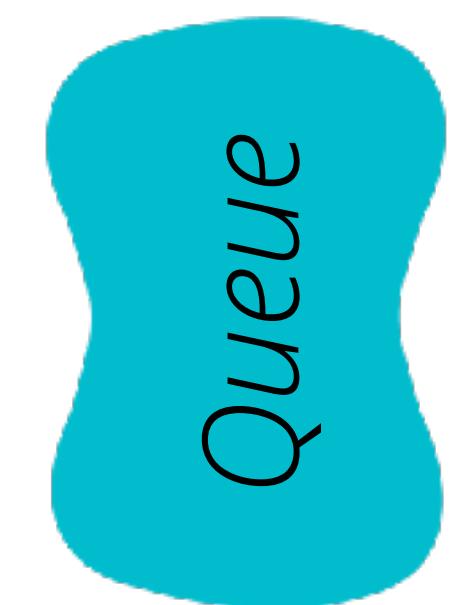
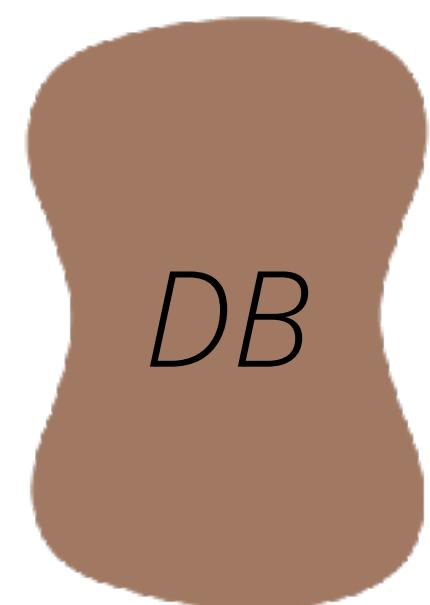
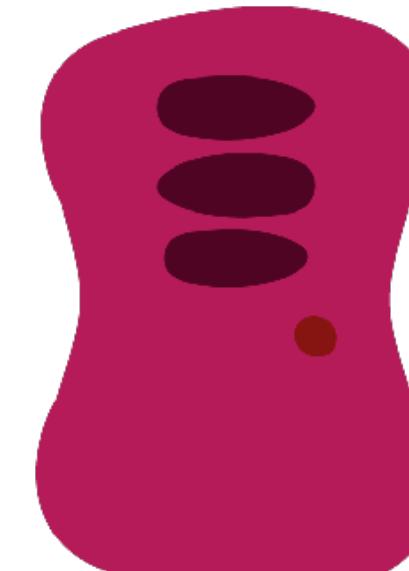
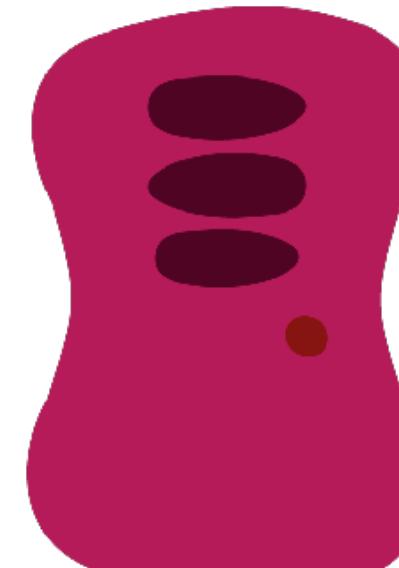
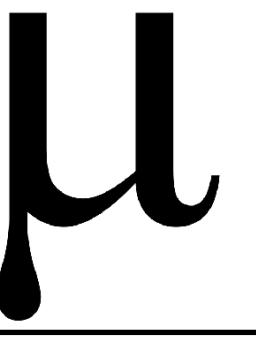
## ROUND 4: MONITORING - MICROSERVICES

---

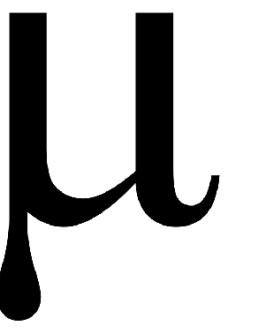


## ROUND 4: MONITORING - MICROSERVICES

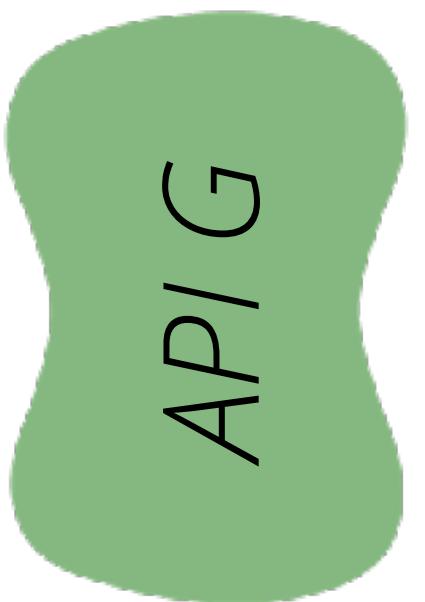
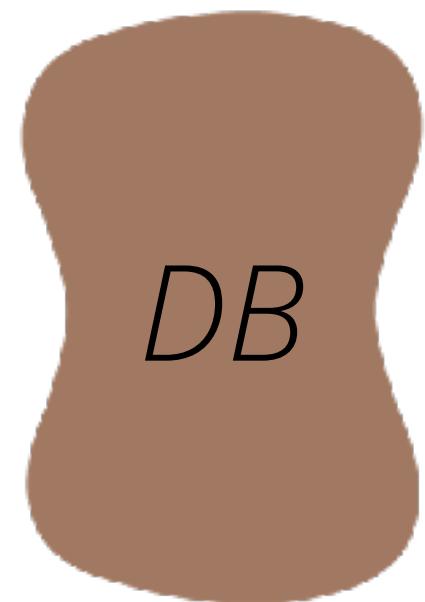
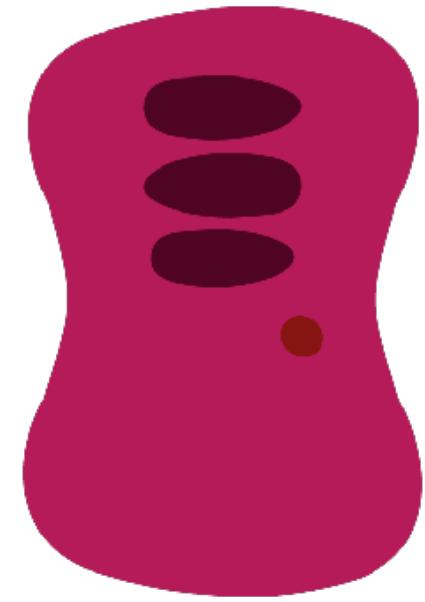
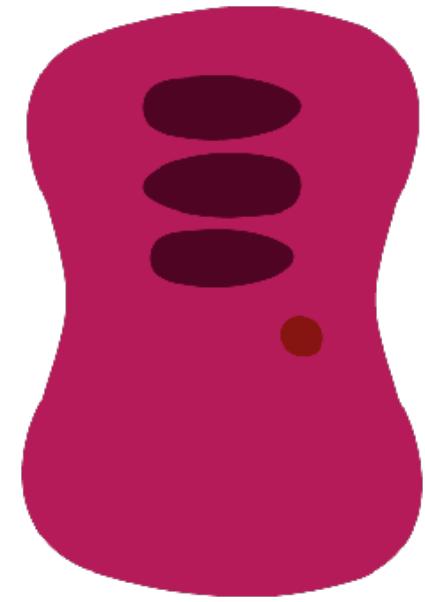
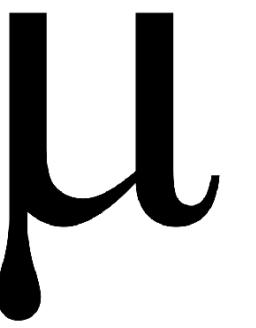
---



## ROUND 4: MONITORING - MICROSERVICES



## ROUND 4: MONITORING - MICROSERVICES



ZABBIX

Nagios®

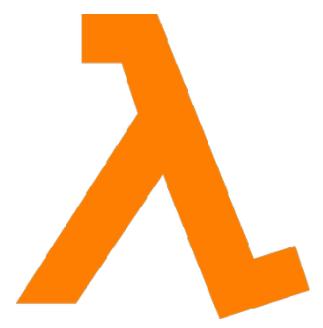
## ROUND 4: MONITORING - SERVERLESS

---

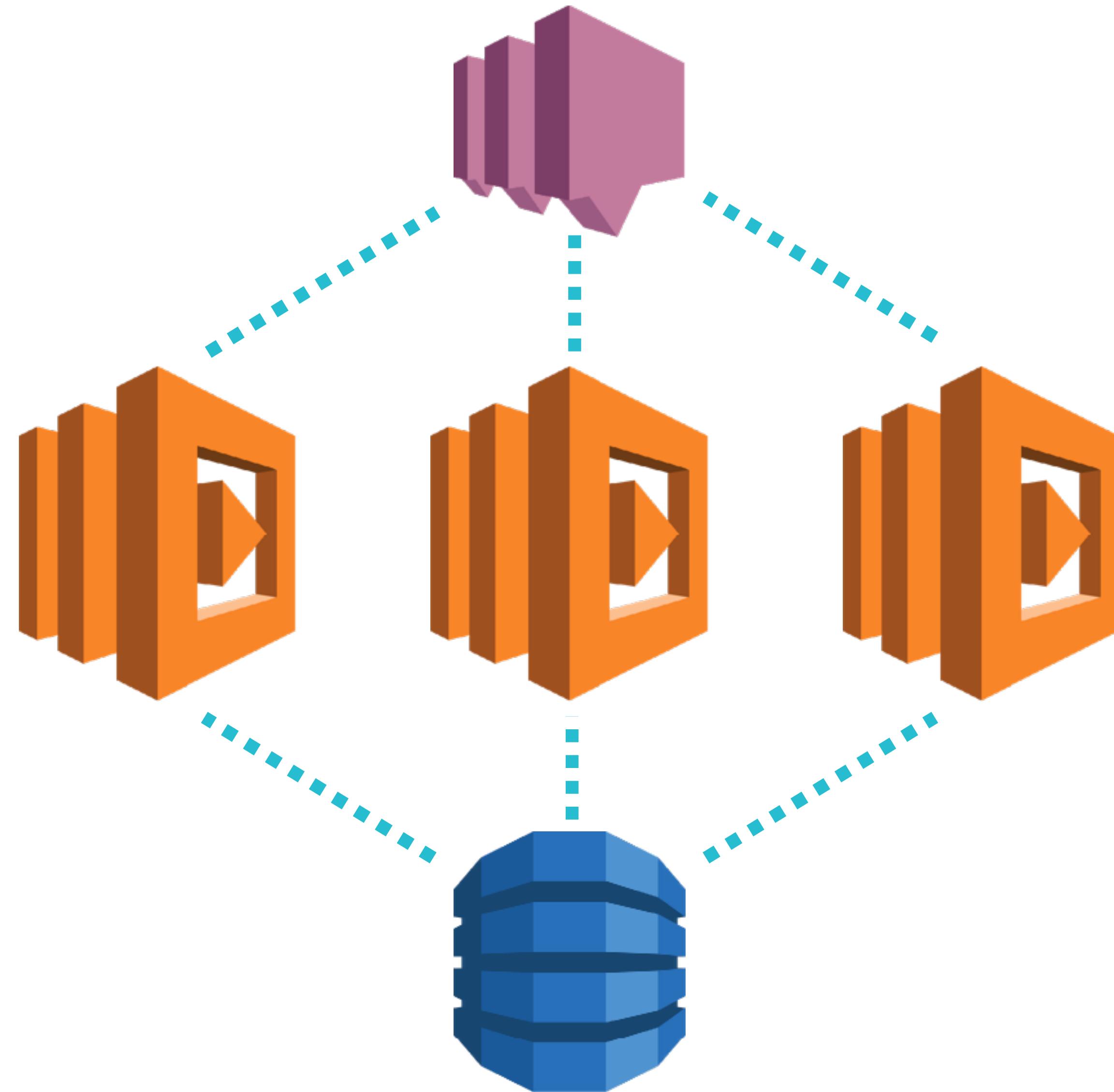
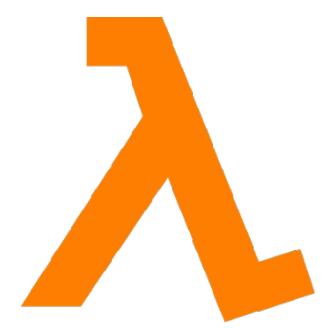


## ROUND 4: MONITORING - SERVERLESS

---

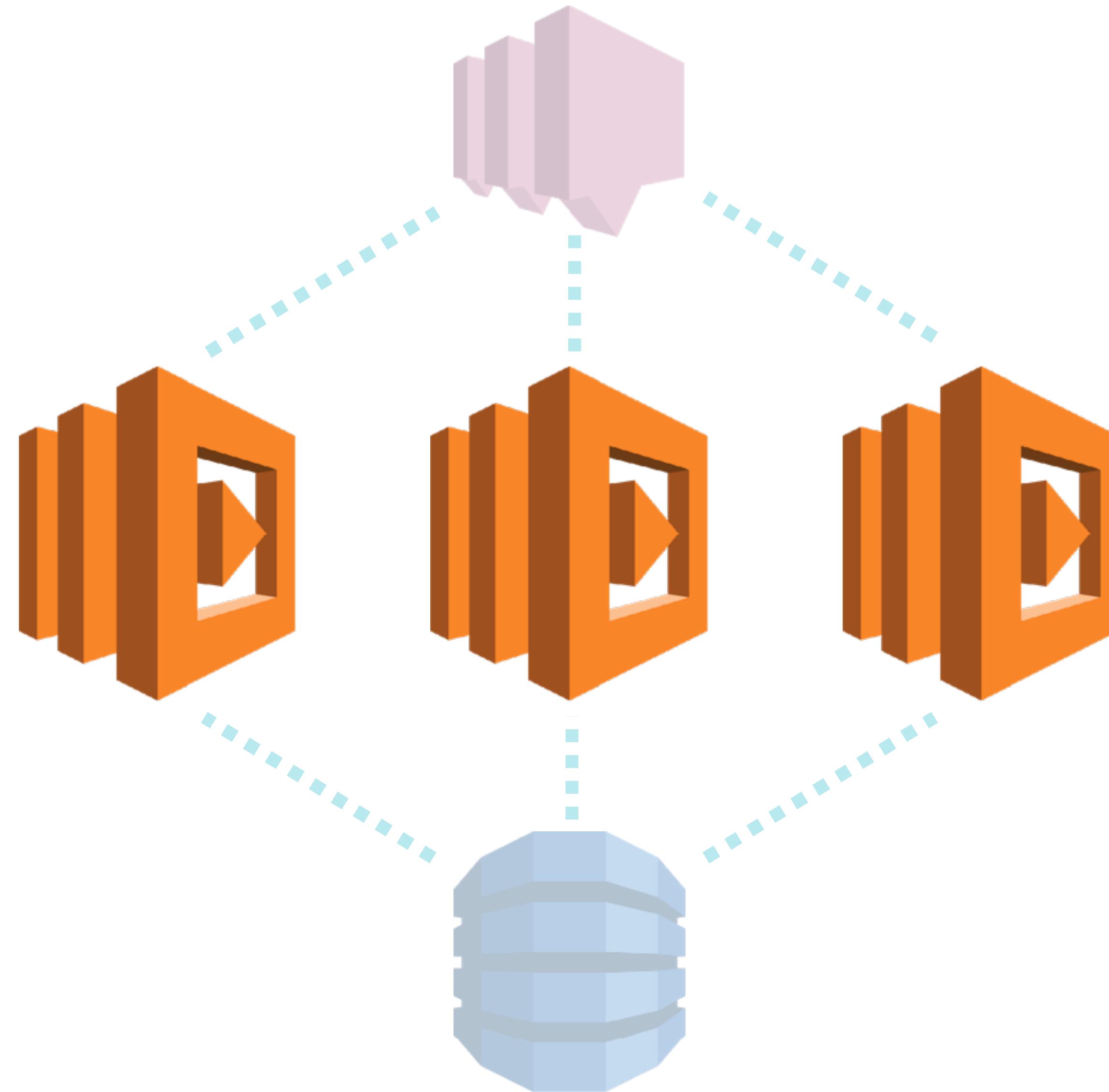
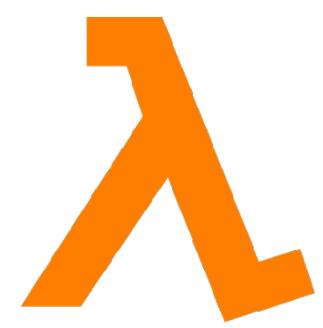


## ROUND 4: MONITORING - SERVERLESS

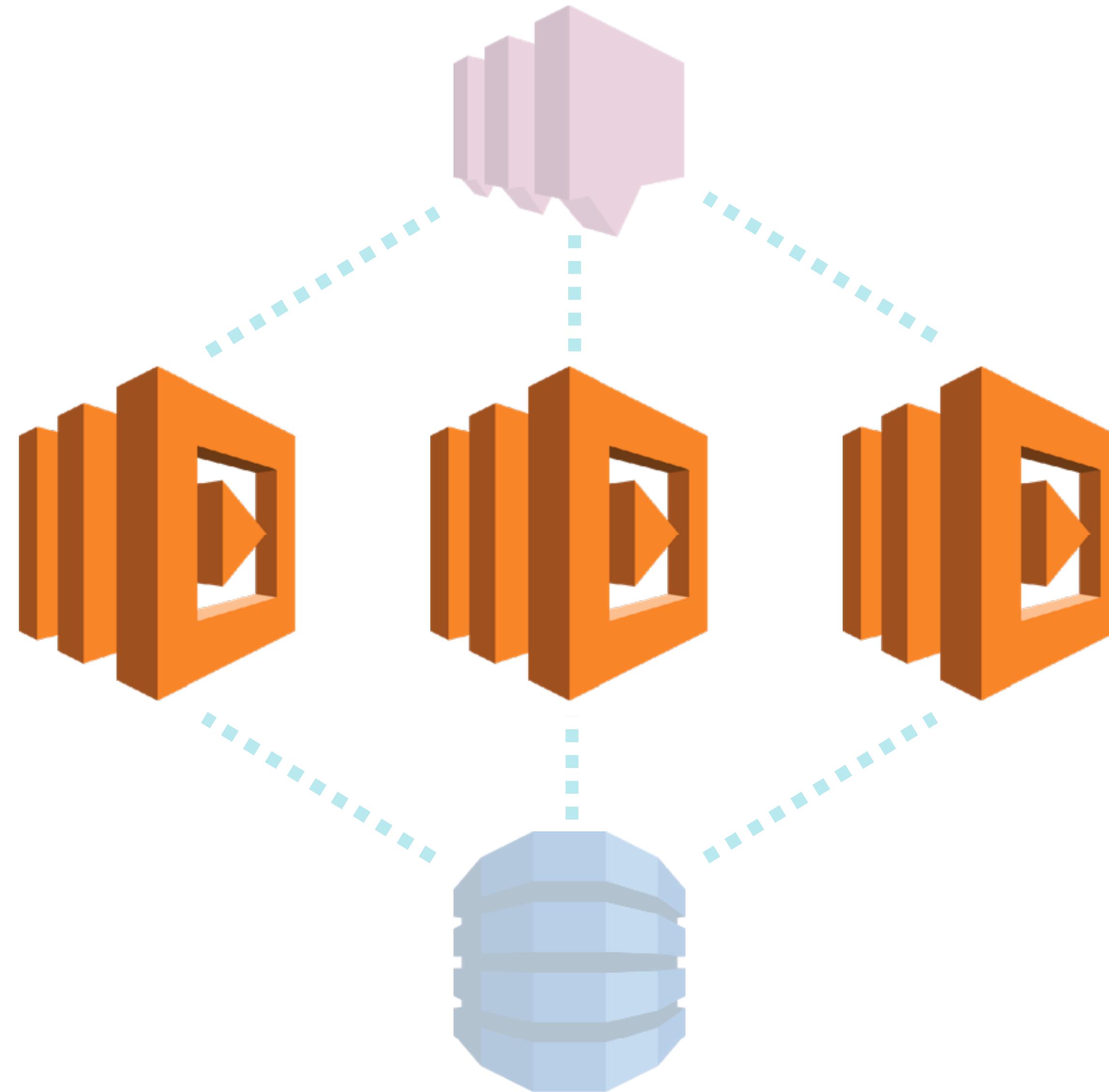
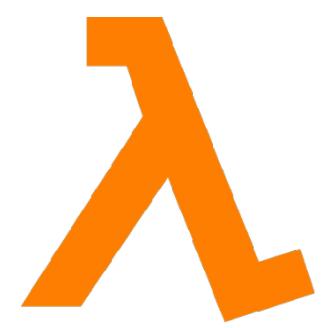


## ROUND 4: MONITORING - SERVERLESS

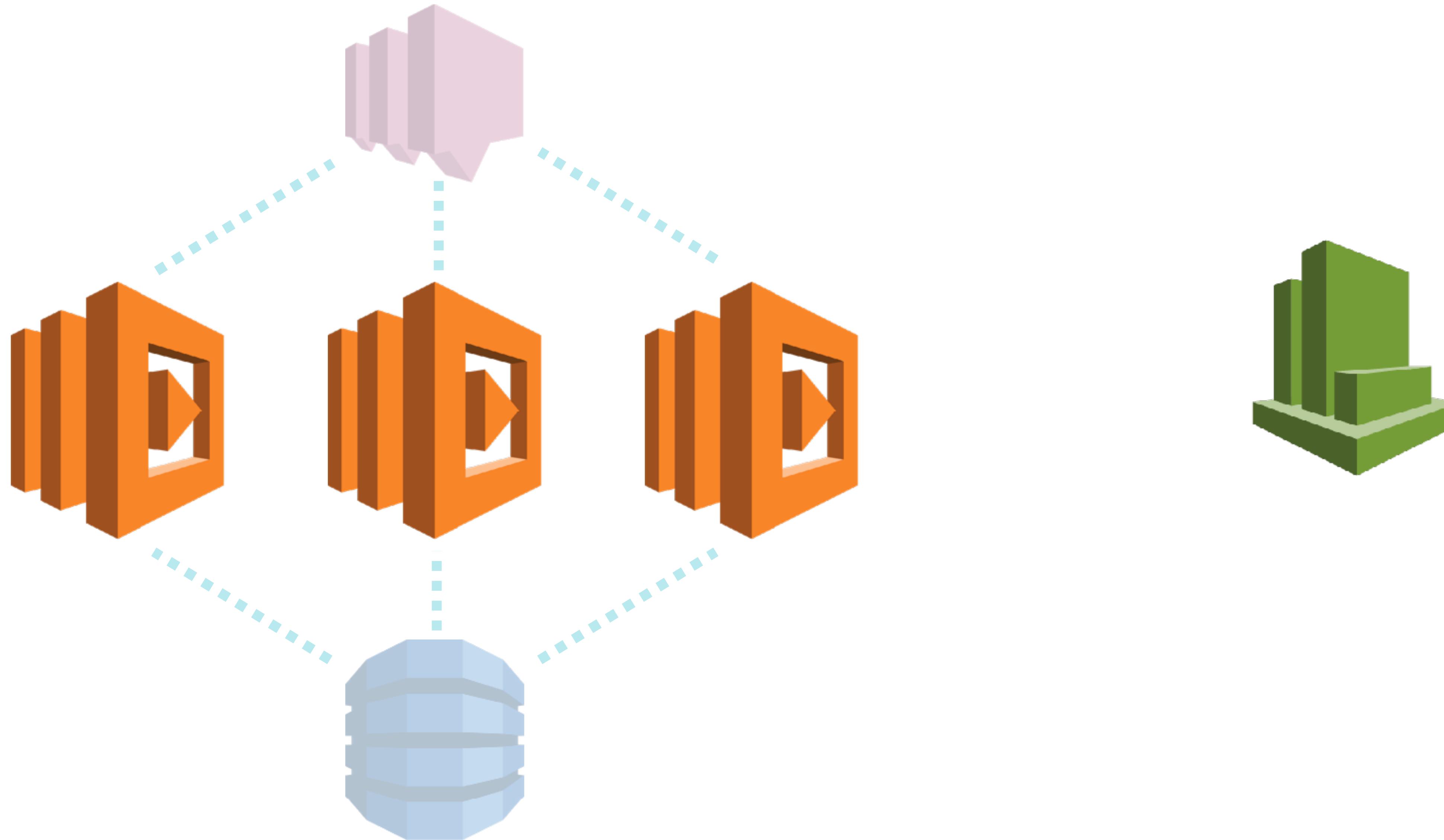
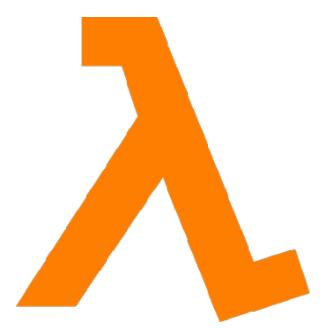
---



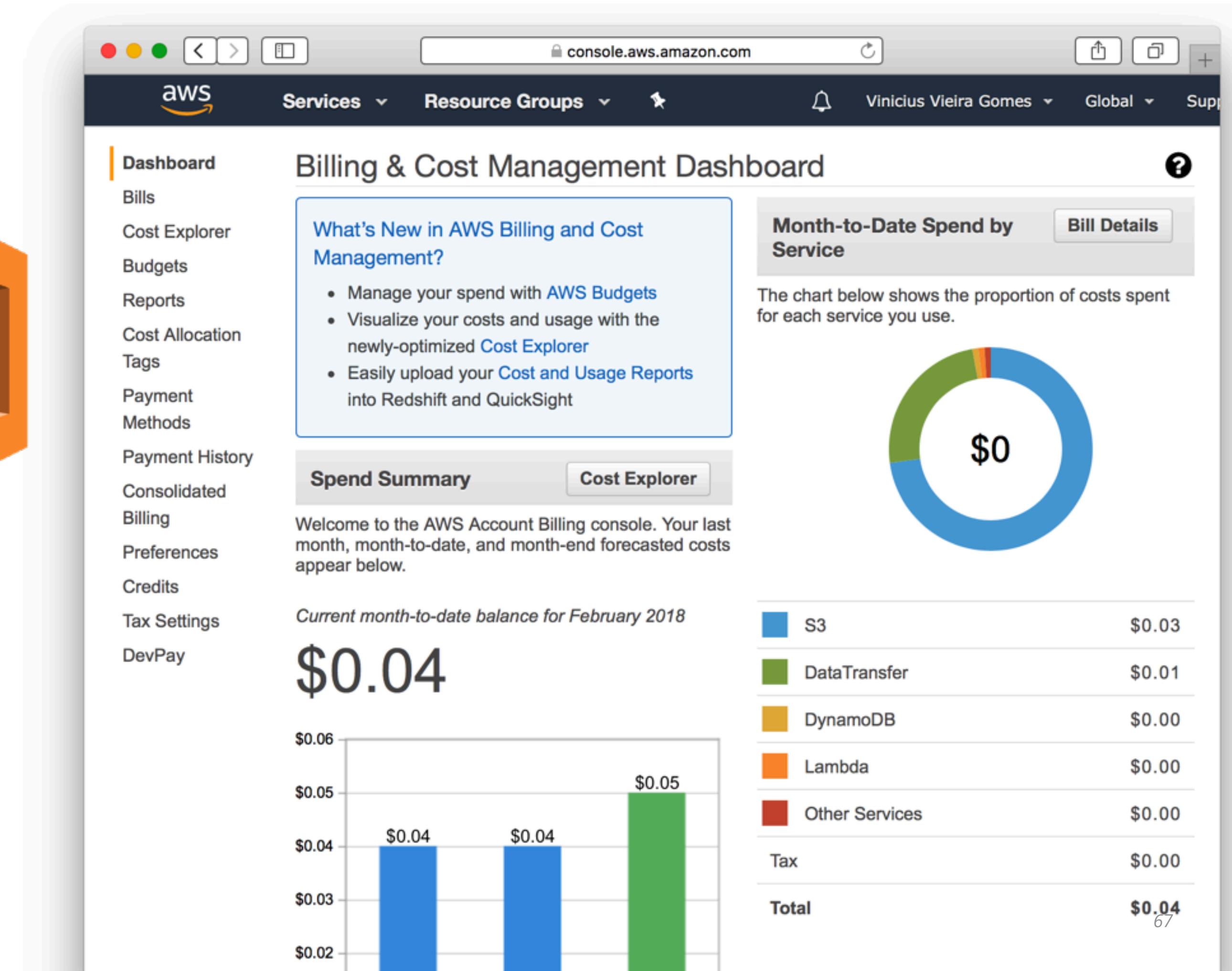
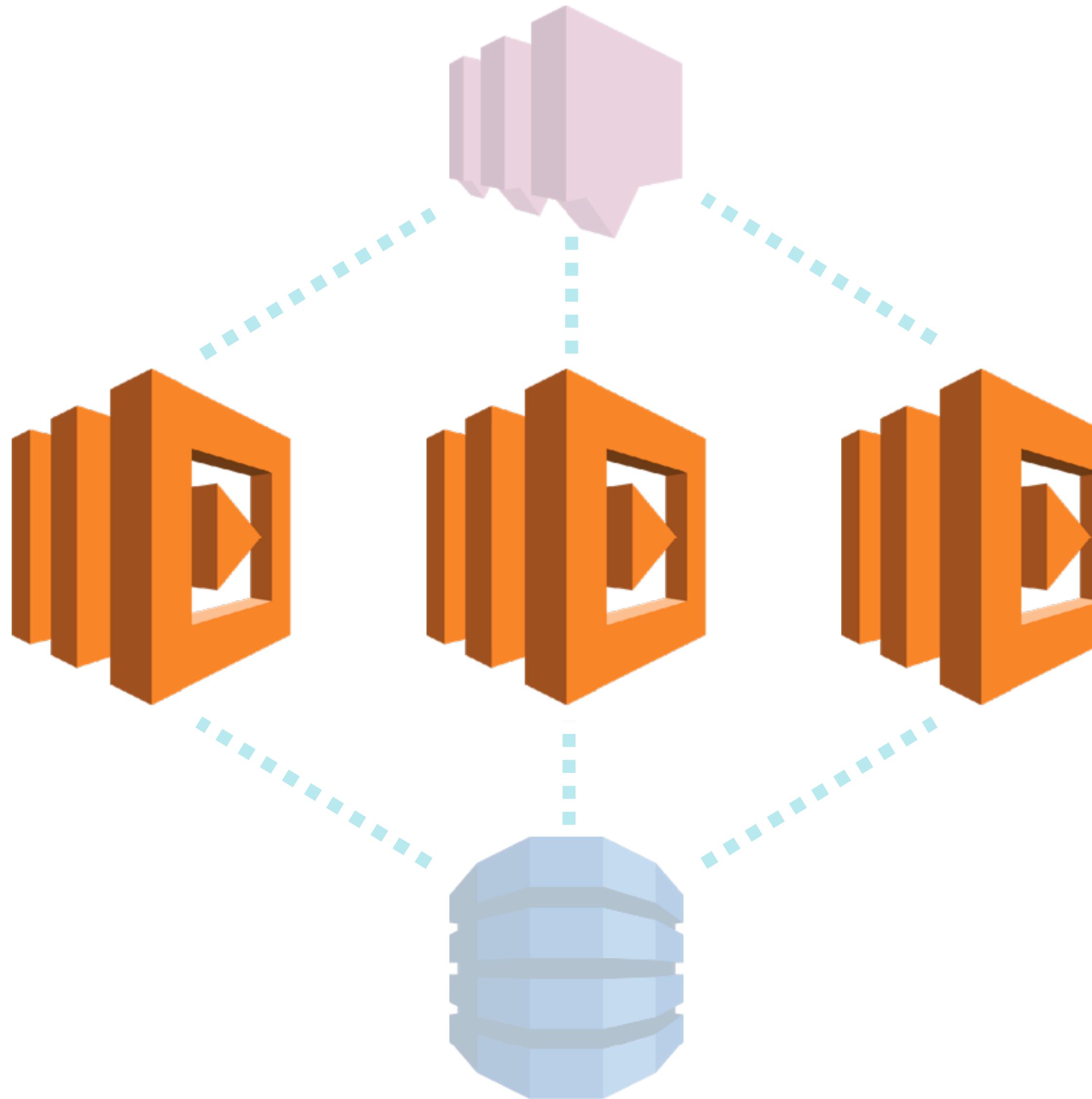
## ROUND 4: MONITORING - SERVERLESS



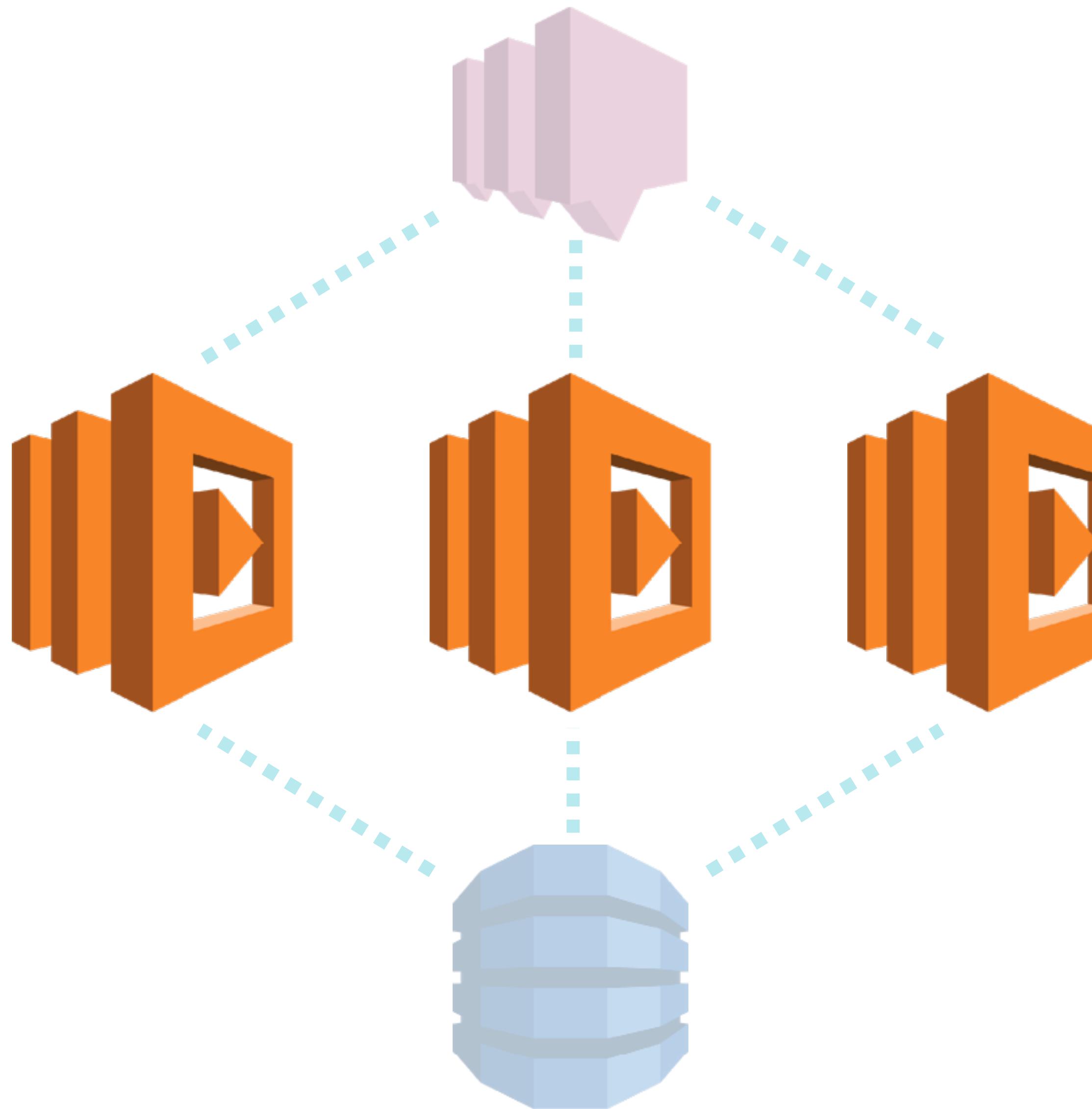
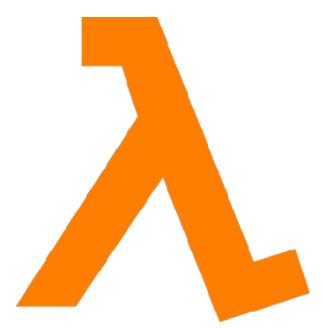
## ROUND 4: MONITORING - SERVERLESS



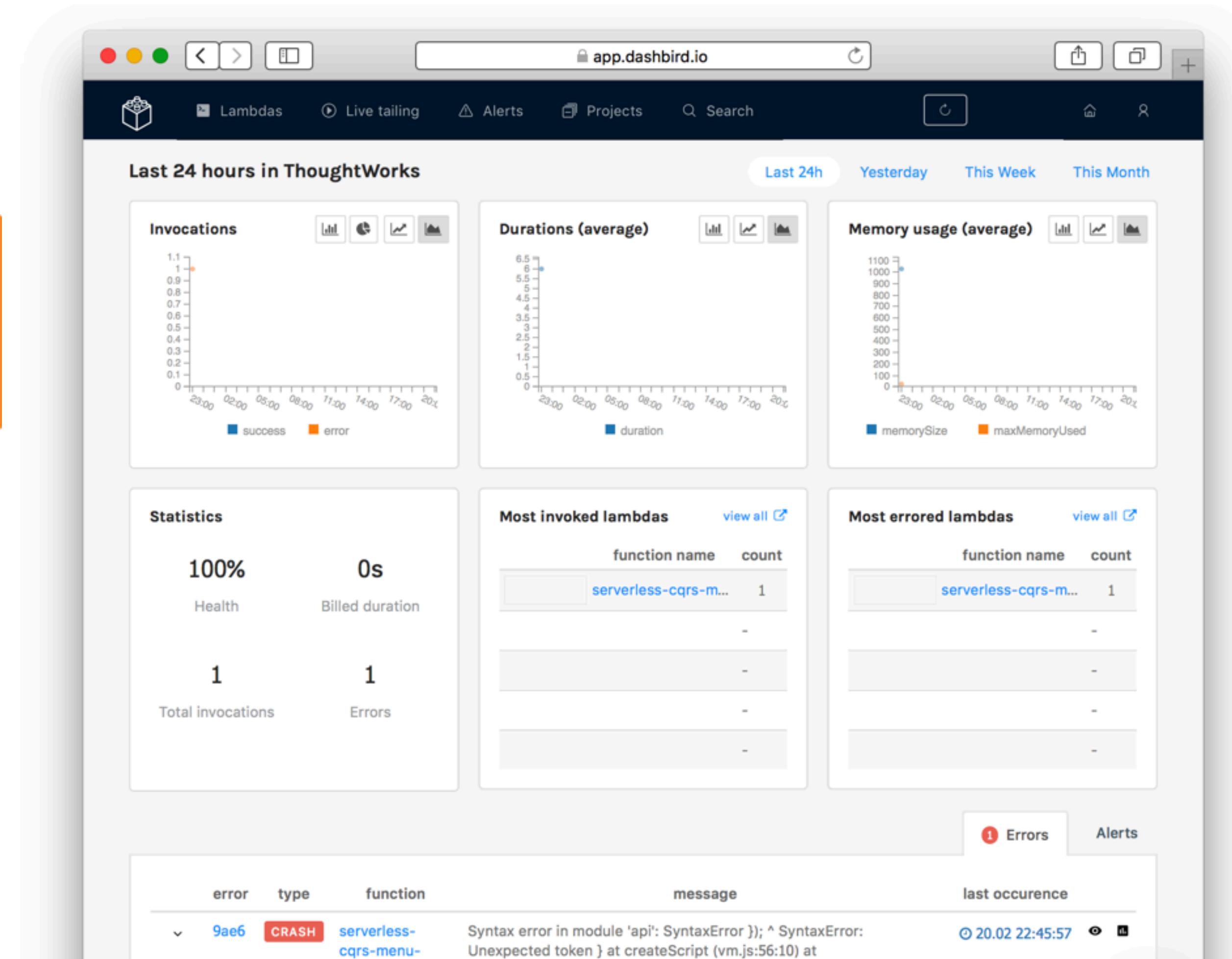
# ROUND 4: MONITORING - SERVERLESS



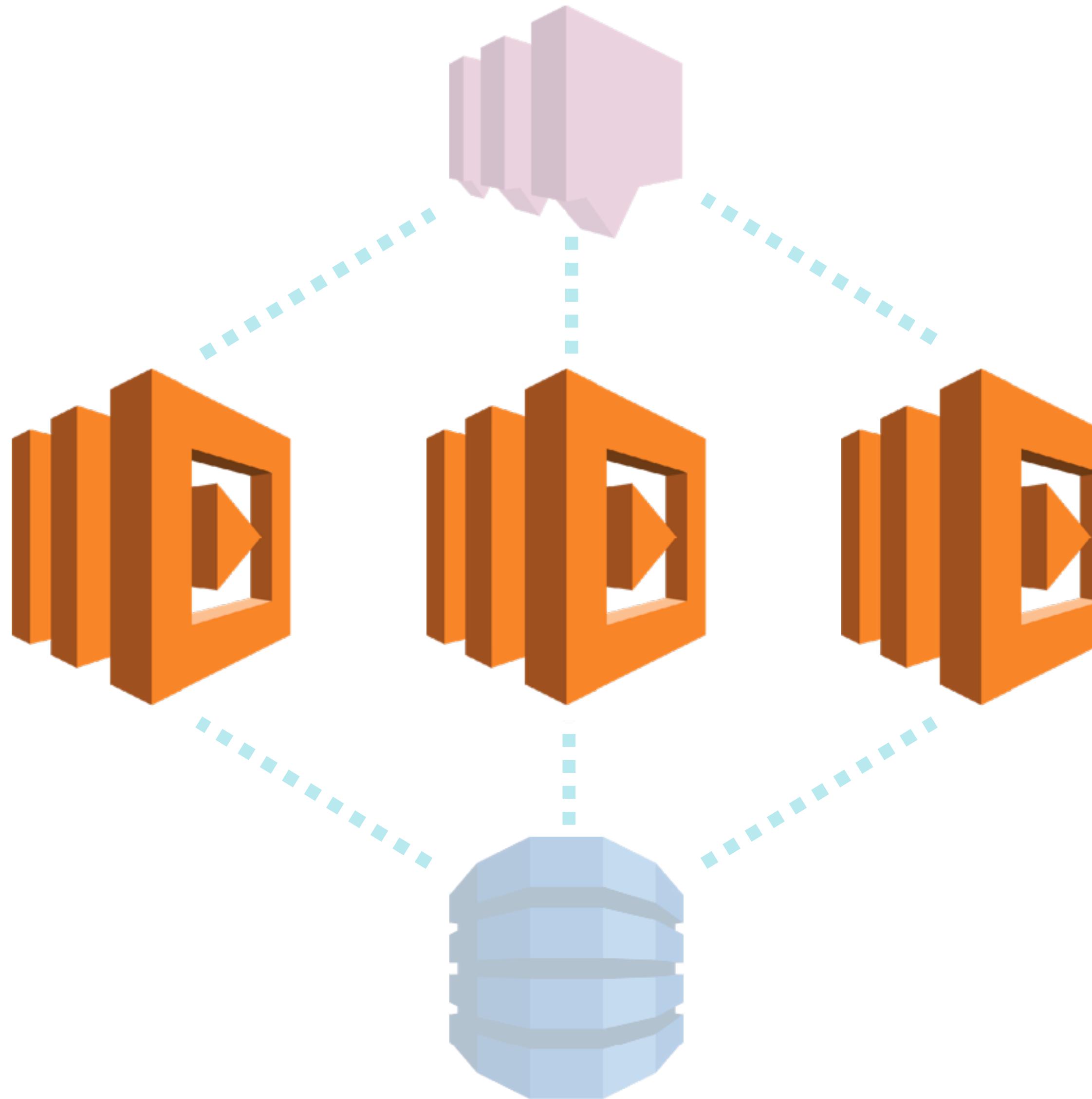
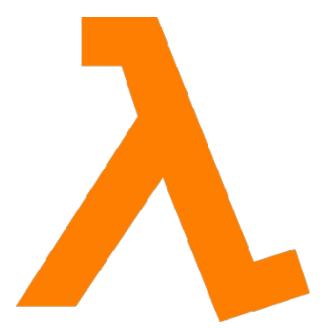
# ROUND 4: MONITORING - SERVERLESS



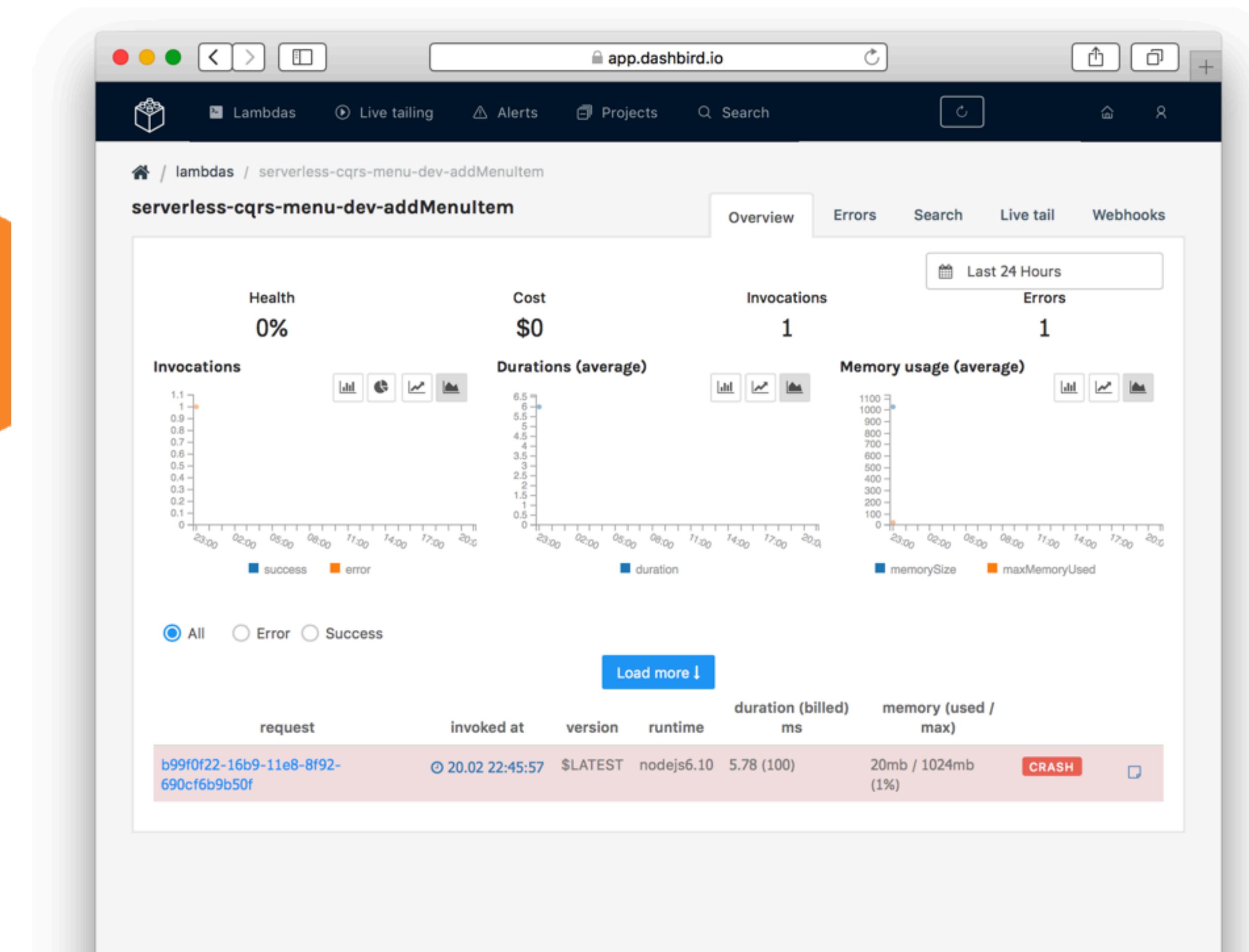
*DashBird.io*



# ROUND 4: MONITORING - SERVERLESS



*DashBird.io*



## ROUND 4 - SCORE CARD

---



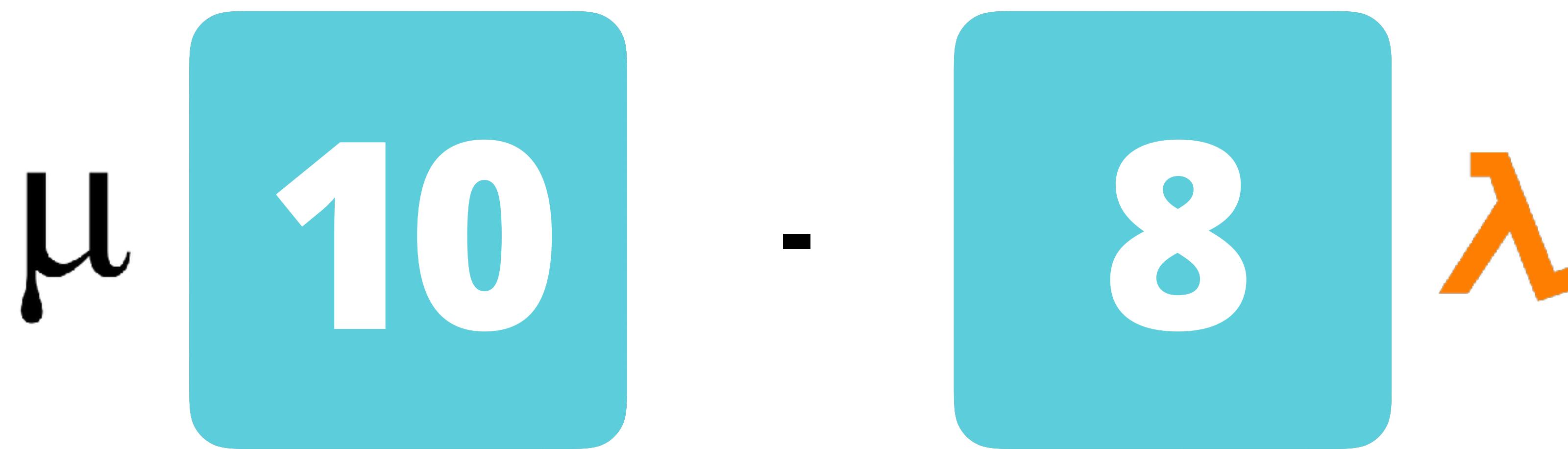
*The Enterprise User*

## ROUND 4 - SCORE CARD

---

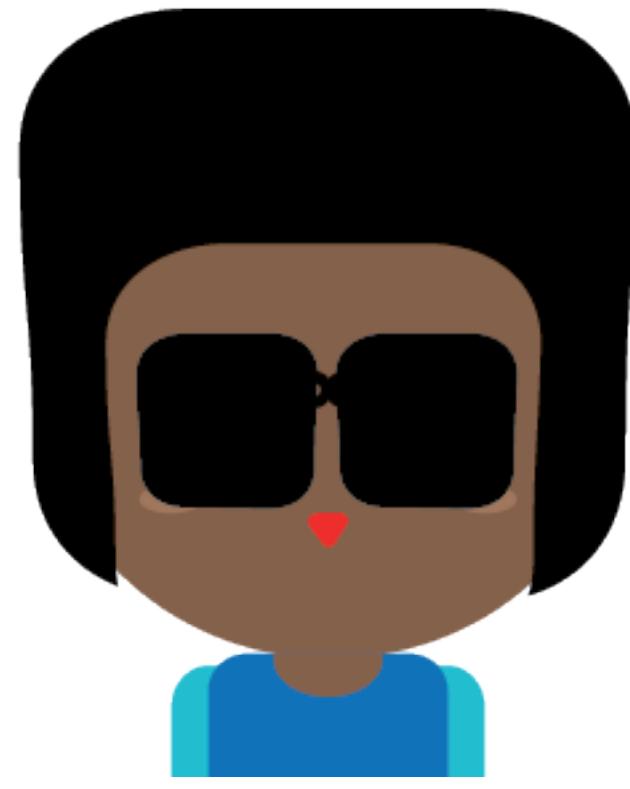


*The Enterprise User*



## ROUND 4 - SCORE CARD

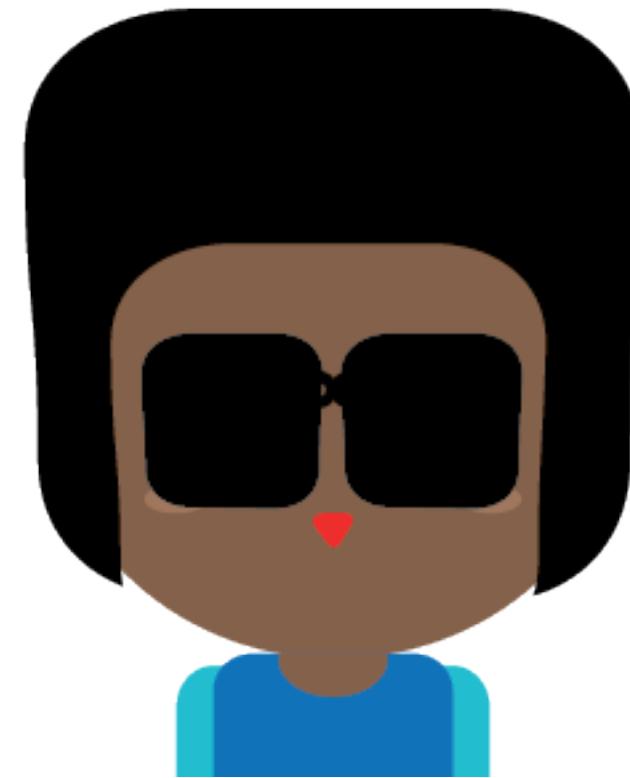
---



*The Startup User*

## ROUND 4 - SCORE CARD

---



*The Startup User*

$\mu$

9

.

10

$\lambda$

## ROUND 4 - SCORE CARD

---



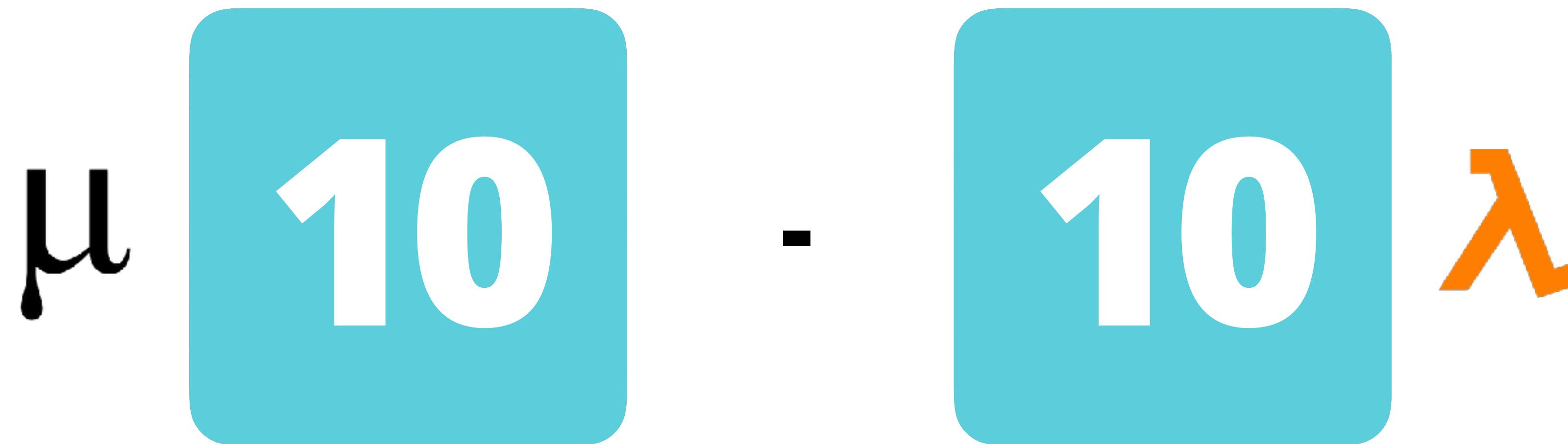
*The Student*

## ROUND 4 - SCORE CARD

---



*The Student*



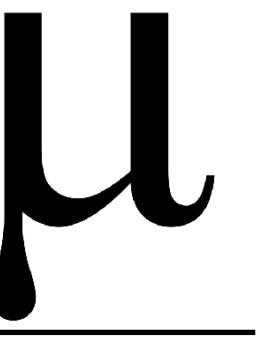
## ROUND 5: COST - MICROSERVICES

---



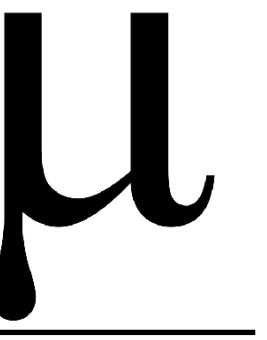
## ROUND 5: COST - MICROSERVICES

---



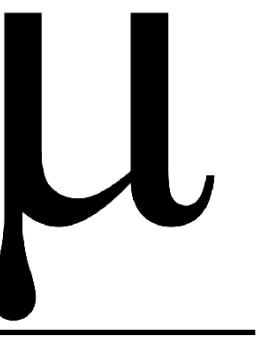
## ROUND 5: COST - MICROSERVICES

---



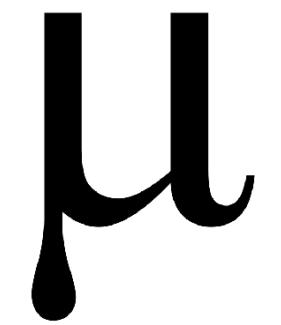
## ROUND 5: COST - MICROSERVICES

---



## ROUND 5: COST - MICROSERVICES

---



t2.micro

1GB memory

**\$0.0116** per hour

**\$8.352** per month

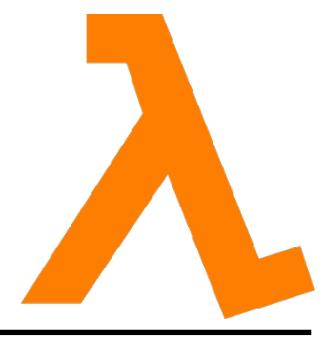
## ROUND 5: COST - SERVERLESS

---



## ROUND 5: COST - SERVERLESS

---



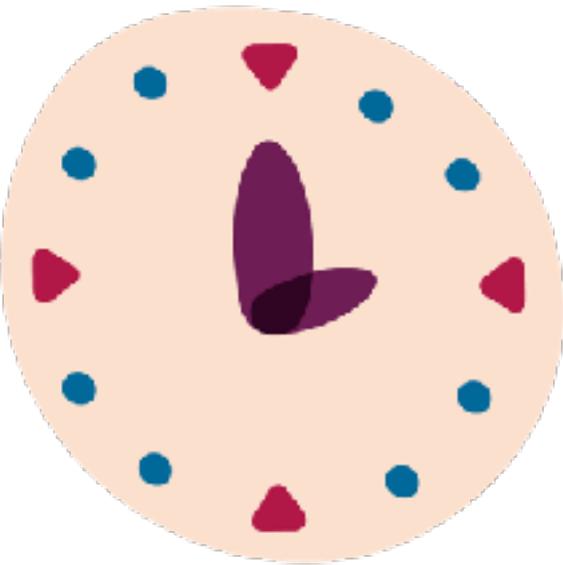
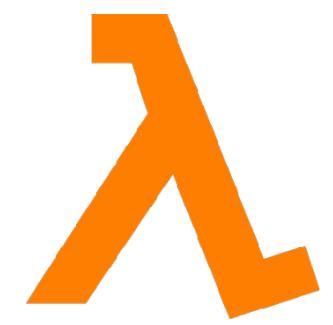
## ROUND 5: COST - SERVERLESS

---



## ROUND 5: COST - SERVERLESS

---

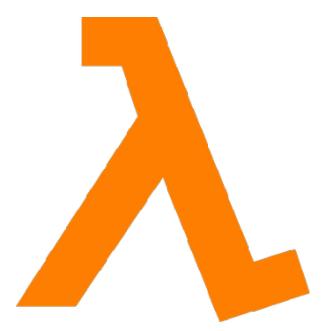


**\$0.00001667**  
per GB-sec

**400,000**  
free tier GB-sec  
per month

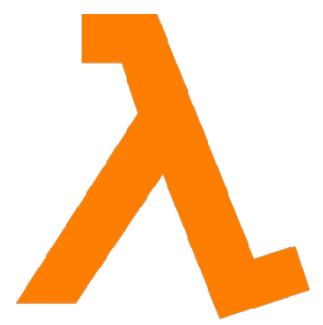
## ROUND 5: COST - SERVERLESS

---



## ROUND 5: COST - SERVERLESS

---

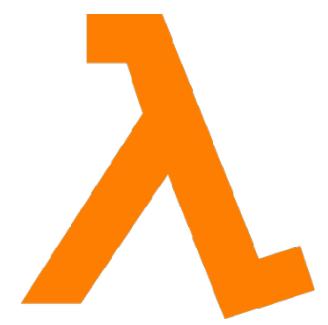


**\$0.000002**  
per request

**1,000,000**  
free tier requests  
per month

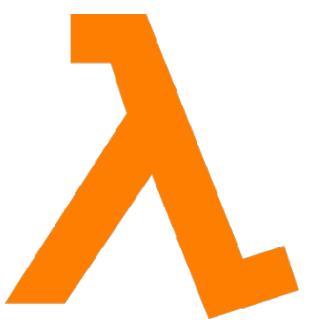
## ROUND 5: COST - SERVERLESS

---



## ROUND 5: COST - SERVERLESS

---

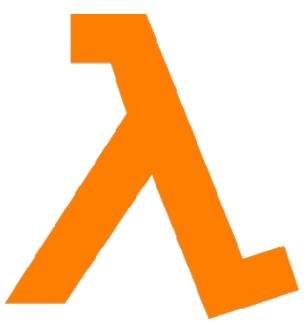


### ***Scenario***

1 request takes 20ms

1GB of memory allocated

# ROUND 5: COST - SERVERLESS



## Scenario

1 request takes 20ms

1GB of memory allocated

Concurrency Level:	10
Time taken for tests:	20.869 seconds
Complete requests:	1000
Failed requests:	0
Total transferred:	588000 bytes
Total body sent:	244000
HTML transferred:	48000 bytes
Requests per second:	47.92 [#/sec] (mean)
Time per request:	208.685 [ms] (mean)
Time per request:	20.869 [ms] (mean, across all concurrent requests)
Transfer rate:	27.52 [Kbytes/sec] received
	11.42 kb/s sent
	38.93 kb/s total

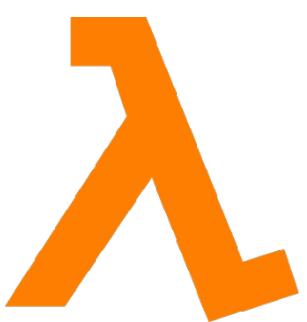
### Connection Times (ms)

	min	mean[+/-sd]	median	max
Connect:	47	71 23.2	66	275
Processing:	62	128 180.8	101	2767
Waiting:	62	127 180.7	101	2767
Total:	118	200 182.2	170	2834

### Percentage of the requests served within a certain time (ms)

50%	170
66%	183
75%	192
80%	200
90%	228
95%	274
98%	437
99%	1299
100%	2834 (longest request)

## ROUND 5: COST - SERVERLESS



### *Scenario*

1 request takes 20ms

1GB of memory allocated

Requests per second	Monthly charge
1	\$0.32
2	\$0.84
3	\$1.35
4	\$1.87
5	\$2.39
6	\$2.91
7	\$3.43
8	\$4.19
9	\$5.57
10	\$6.96
11	\$8.34
12	\$9.72

## ROUND 5: COST - SERVERLESS



### *Scenario*

1 request takes 20ms

1GB of memory allocated

**t2.micro**

1GB memory

**\$0.0116** per hour

**\$8.352** per month

Requests per second	Monthly charge
1	<b>\$0.32</b>
2	<b>\$0.84</b>
3	<b>\$1.35</b>
4	<b>\$1.87</b>
5	<b>\$2.39</b>
6	<b>\$2.91</b>
7	<b>\$3.43</b>
8	<b>\$4.19</b>
9	<b>\$5.57</b>
10	<b>\$6.96</b>
11	<b>\$8.34</b>
12	<b>\$9.72</b>

## ROUND 5 - SCORE CARD

---



*The Enterprise User*

## ROUND 5 - SCORE CARD

---



*The Enterprise User*

$\mu$

10

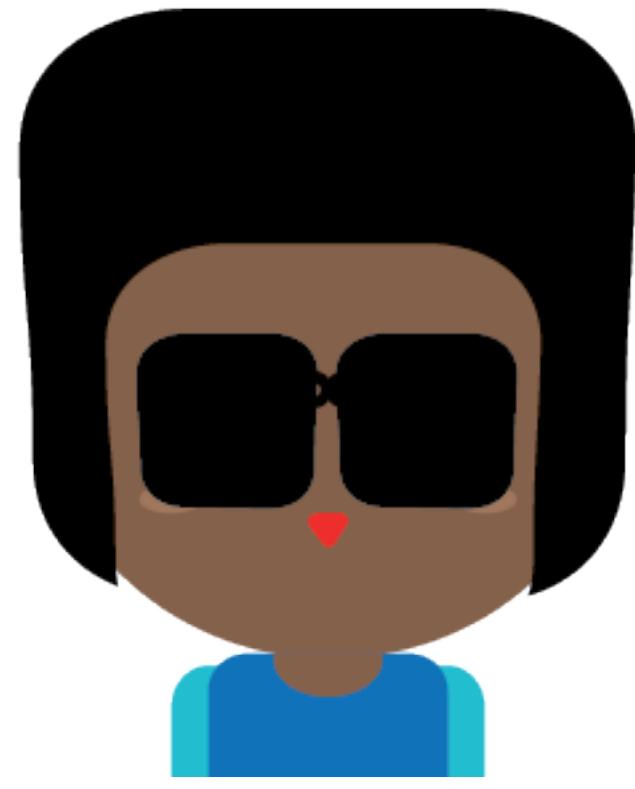
-

8

$\lambda$

## ROUND 5 - SCORE CARD

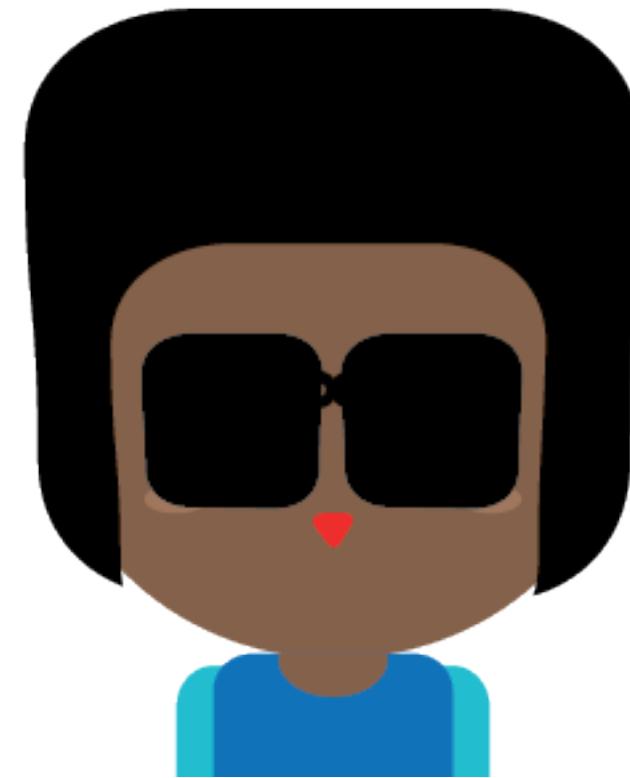
---



*The Startup User*

## ROUND 5 - SCORE CARD

---



*The Startup User*

$\mu$

9

.

10

$\lambda$

## ROUND 5 - SCORE CARD

---



*The Student*

## ROUND 5 - SCORE CARD

---



*The Student*

$\mu$

7

-

10

$\lambda$

# RESULTS

---

*What did the judges have to say?*

$\mu$

$\lambda$

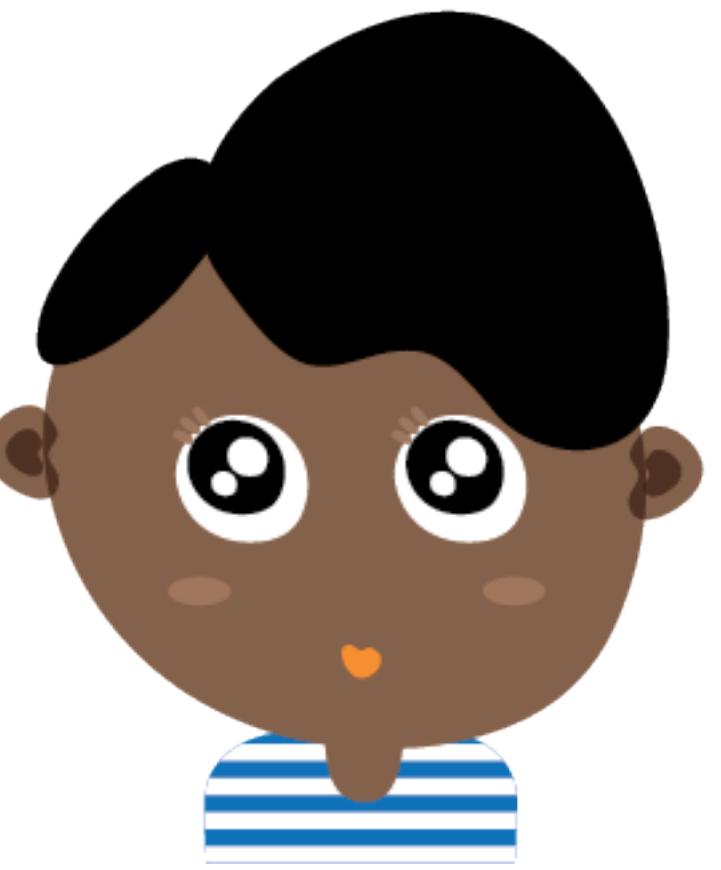


# RESULTS

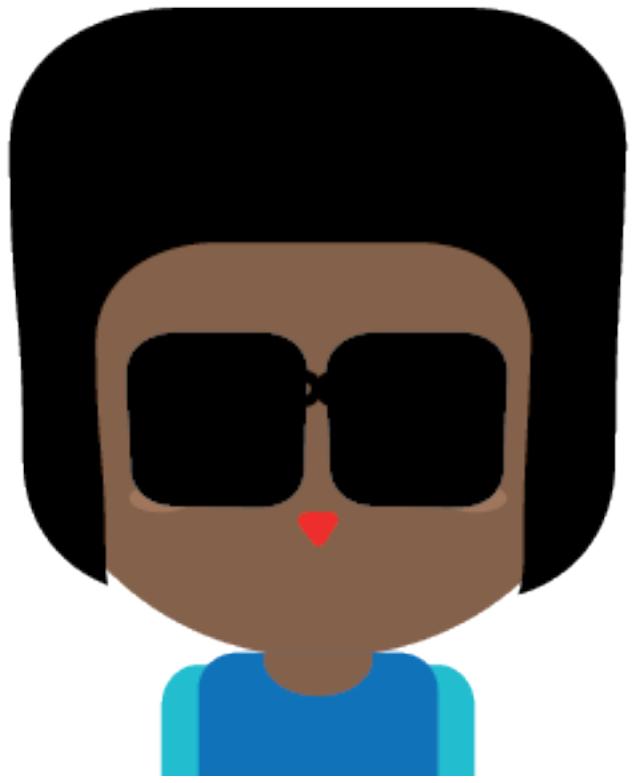
---



*The Enterprise User*



*The Student*



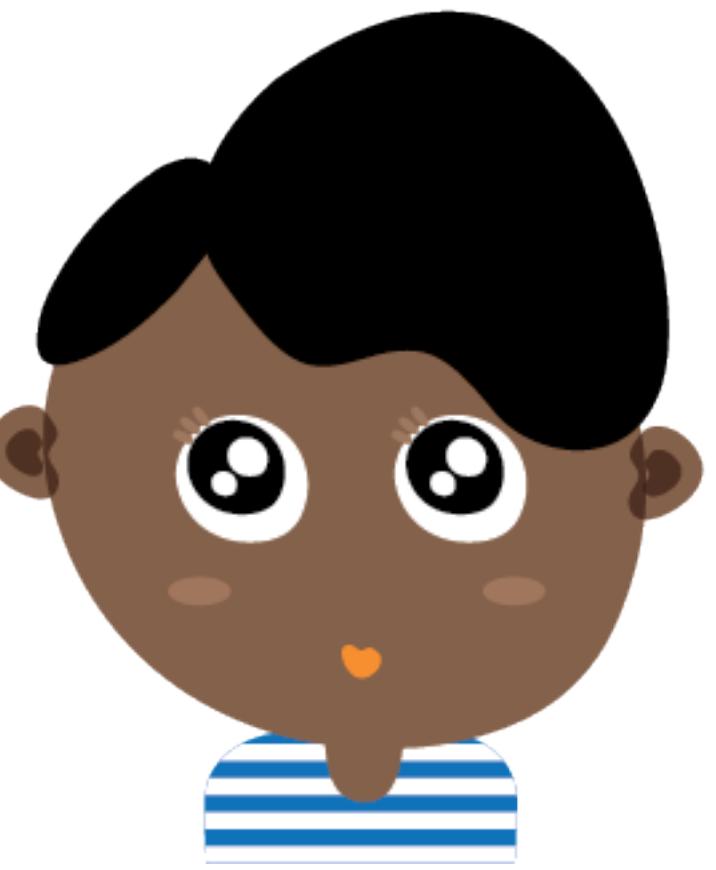
*The Startup User*

# RESULTS

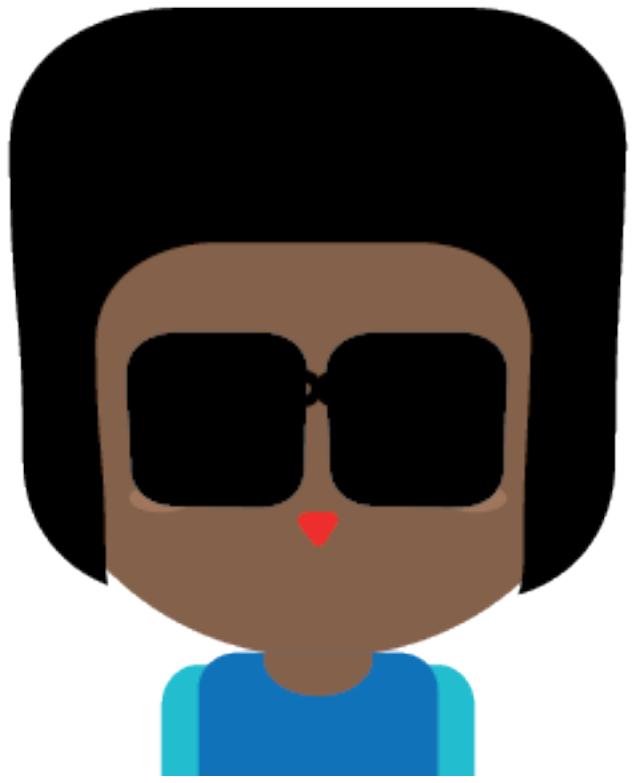
---



*The Enterprise User*



*The Student*



*The Startup User*

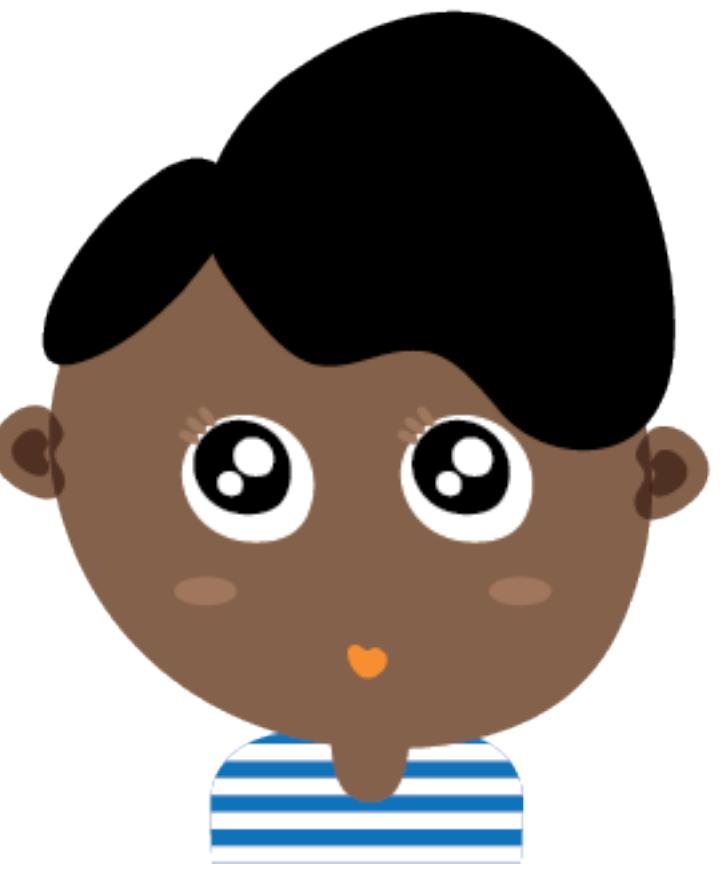
$$\mu > \lambda$$

# RESULTS

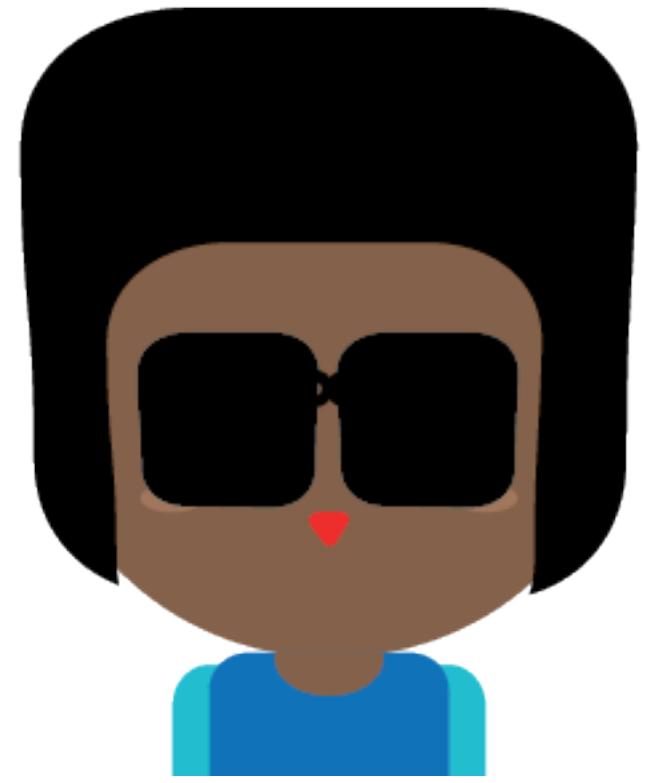
---



*The Enterprise User*



*The Student*



*The Startup User*

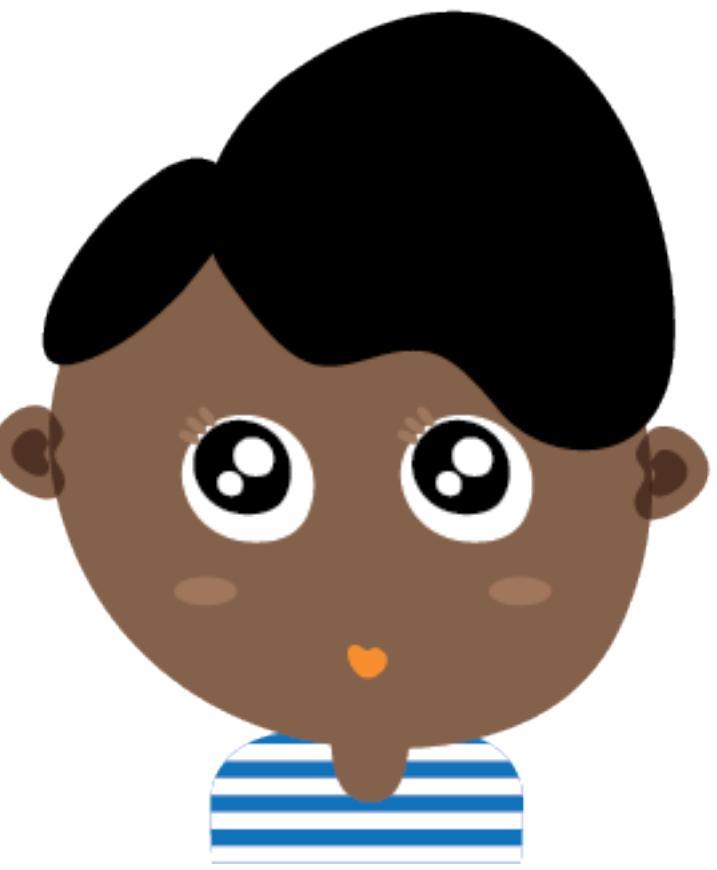
$$\mu > \lambda$$

# RESULTS

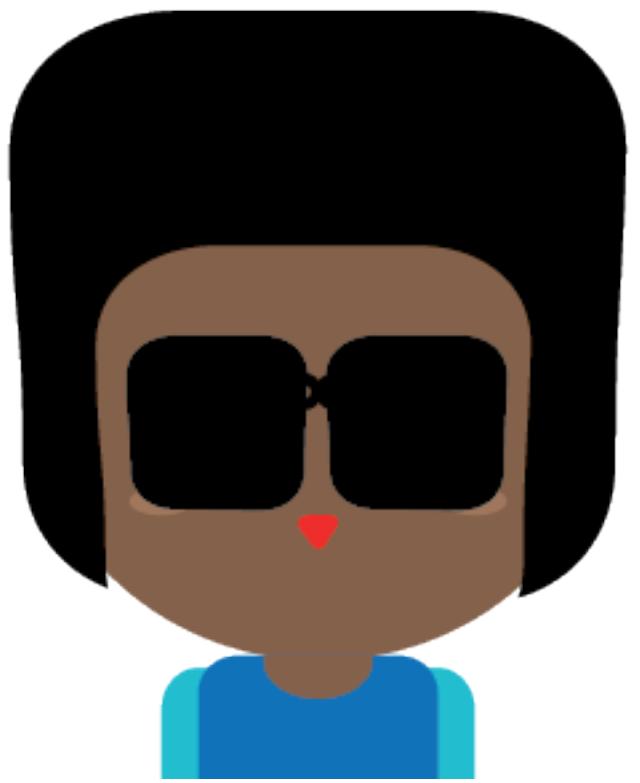
---



*The Enterprise User*



*The Student*



*The Startup User*

$$\mu > \lambda$$

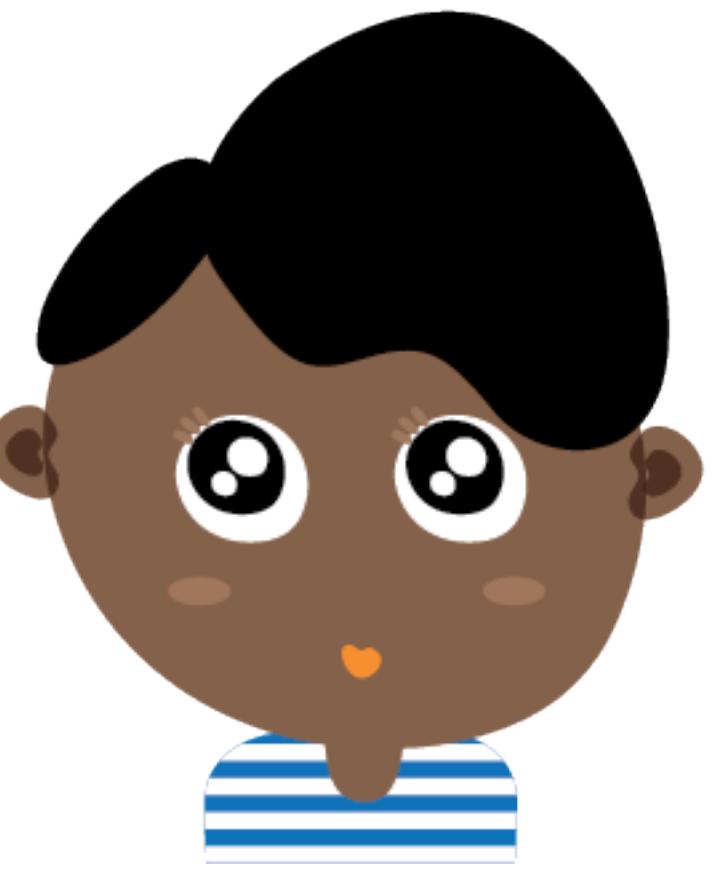
$$\mu < \lambda$$

# RESULTS

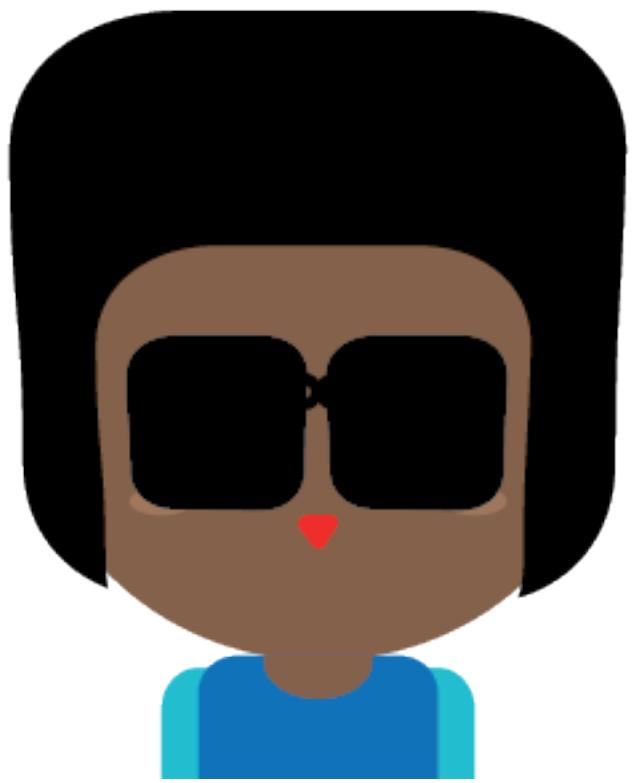
---



*The Enterprise User*



*The Student*



*The Startup User*

$$\mu > \lambda$$

$$\mu < \lambda$$

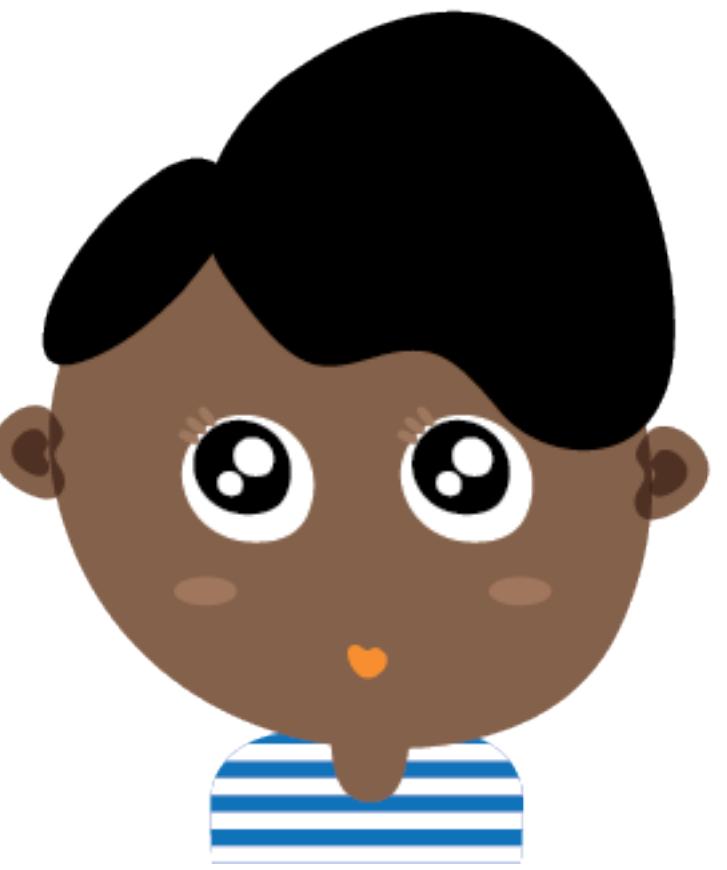
# RESULTS

---



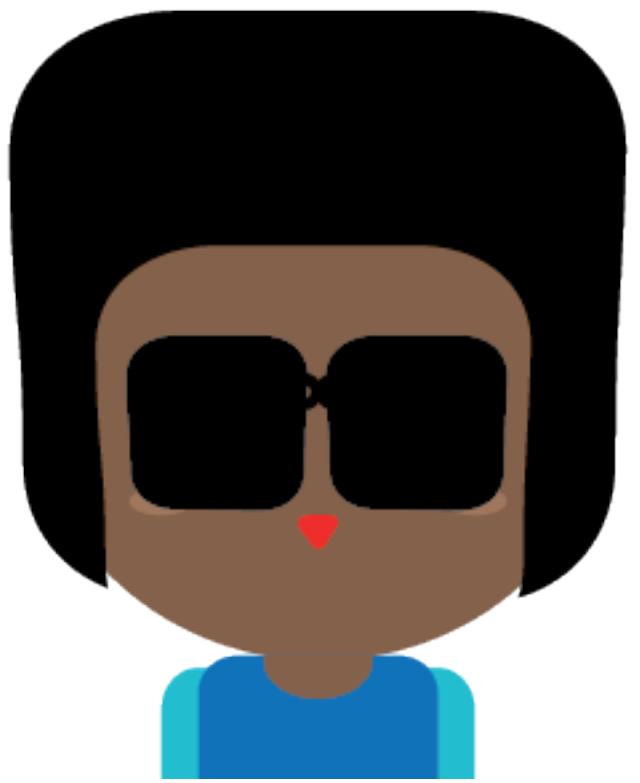
*The Enterprise User*

$$\mu > \lambda$$



*The Student*

$$\mu < \lambda$$



*The Startup User*

$$\mu < \lambda$$

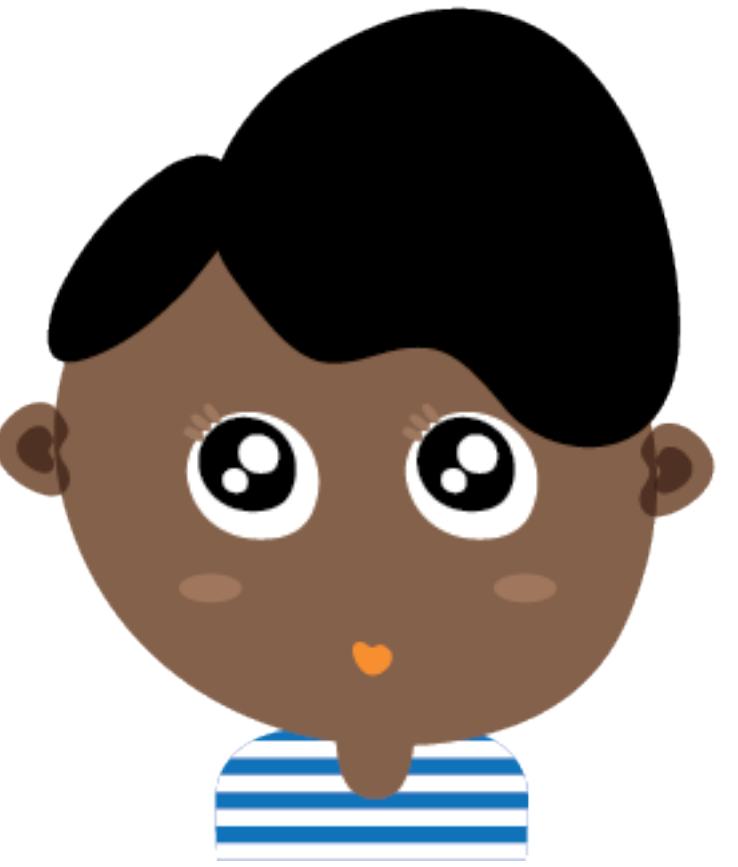
# RESULTS

---



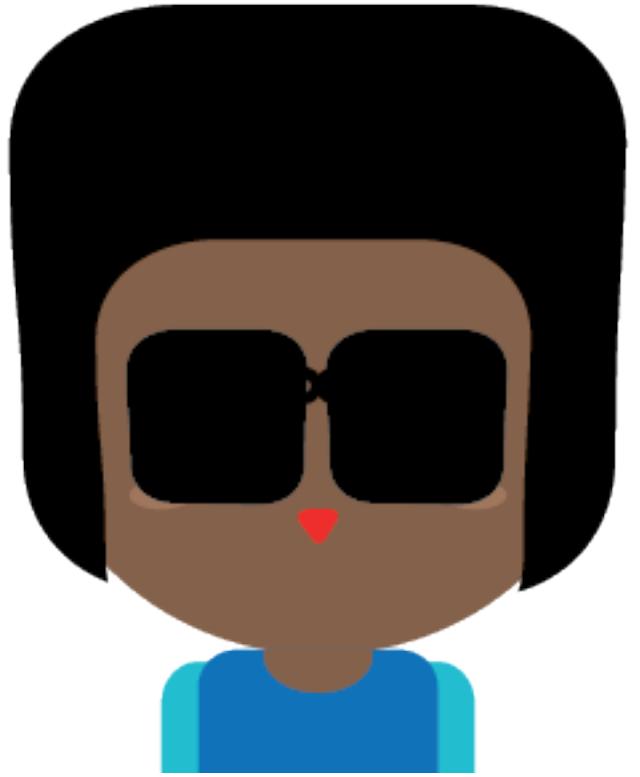
*The Enterprise User*

$$\mu > \lambda$$



*The Student*

$$\mu < \lambda$$



*The Startup User*

$$\mu < \lambda$$

# WINNER

---

*And NEW Software Architecture Champion!*

A

# THANK YOU

*Karun Japhet (@javatarz)  
Vinicius Gomes (@vvgomess)*

*Slides: <http://bit.ly/ms-vs-sls-sacon-ny-18>*

*Serverless code: [bit.ly/sls-cqrs](http://bit.ly/sls-cqrs)*

*Microservices code: [bit.ly/ms-ed-restaurant](http://bit.ly/ms-ed-restaurant)*

ThoughtWorks®