

# Serverless for Pythonistas



# Jonatas Baldin

Developer @ Maecenas

Creator of Serverless Brasil

@jonatasbaldin

**servers  
to  
serverless**



**application**

**application**

**bare metal**

**application**

**OS**

**bare metal**

**application**

**virtual machine**

**OS**

**bare metal**



**application**

**container**

**virtual machine**

**OS**

**bare metal**

**application**

**container**

**virtual machine**

**OS**

**bare metal**

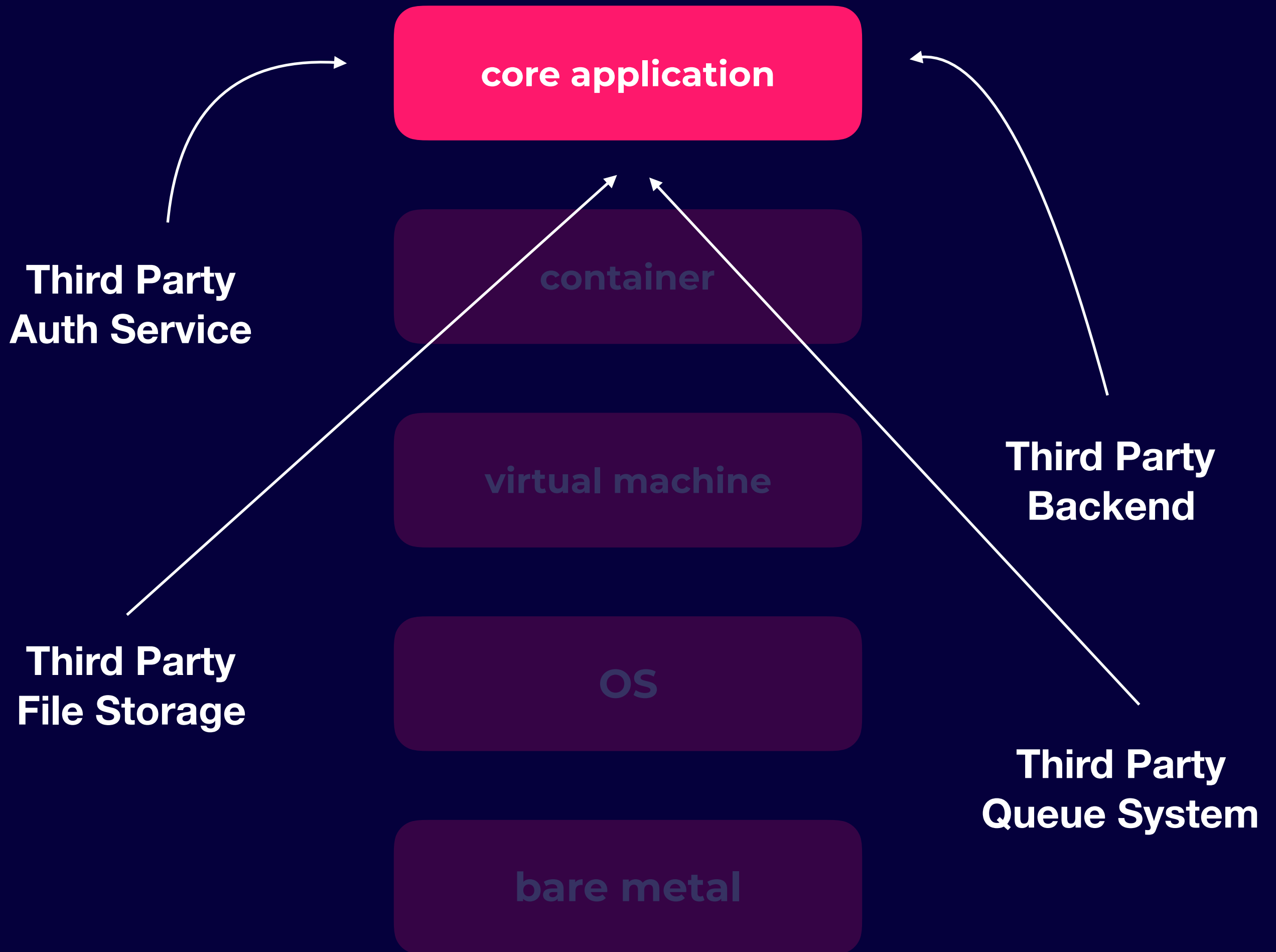
**core application**

**container**

**virtual machine**

**OS**

**bare metal**



**less servers**  
**more services**

why should *you* care?



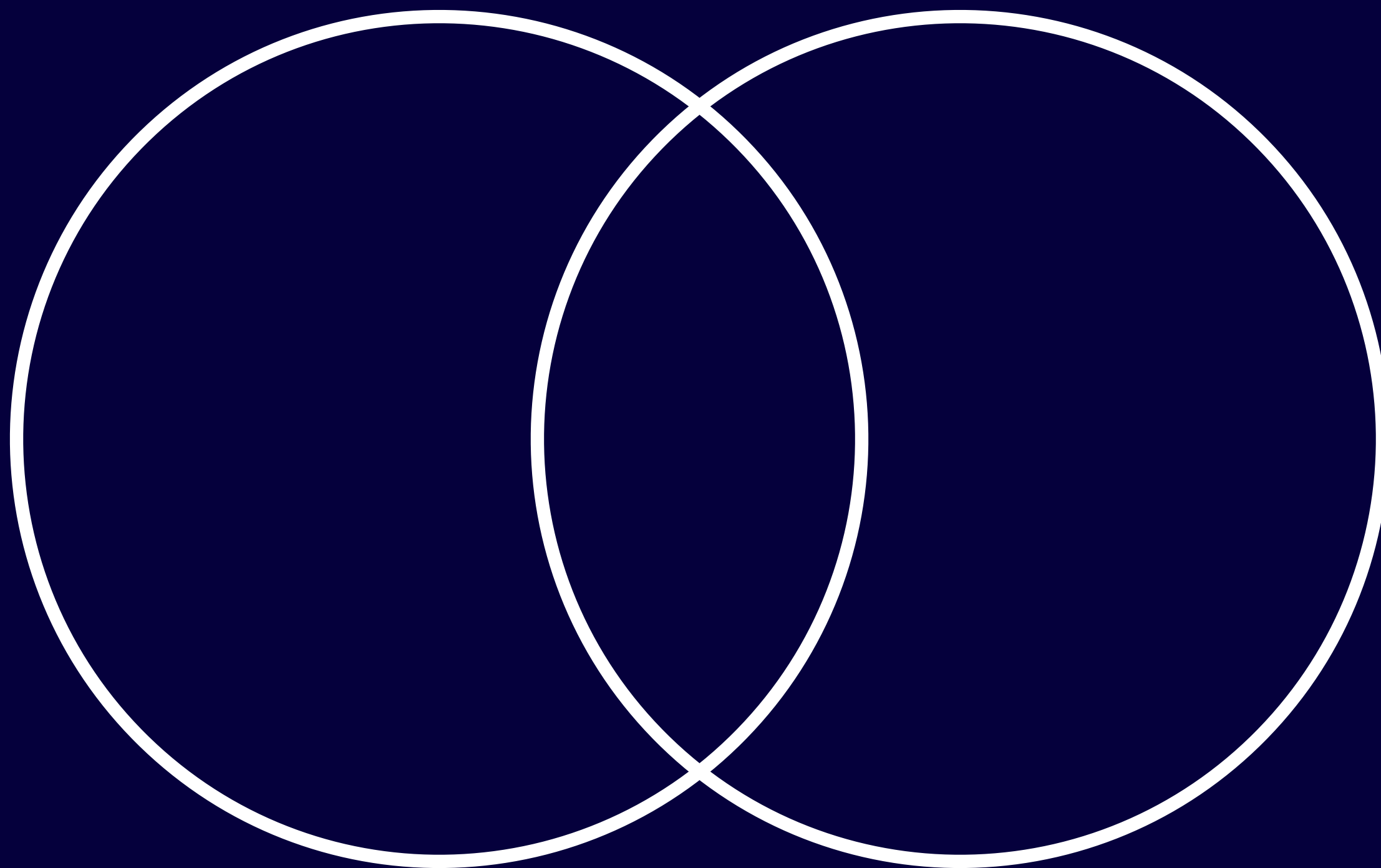
**focus on problem  
solving**

**we are in a constant  
search for abstraction  
and single purpose**



**focus on the core  
business**

**so, what is  
Serverless?**





**BaaS**



A Venn diagram consisting of two overlapping circles. The left circle is labeled 'BaaS' and the right circle is labeled 'FaaS'. The intersection of the two circles is empty.

**BaaS**

**FaaS**

**BaaS**



**FaaS**

✨ FaaS ✨

# ⚡ FaaS Principles ⚡

**Event Driven**  
**Ephemeral Environments**  
**Built in Scalability**  
**Pay by Execution**  
**Lack of Management**



# ⚡ FaaS Principles ⚡

**Event Driven**

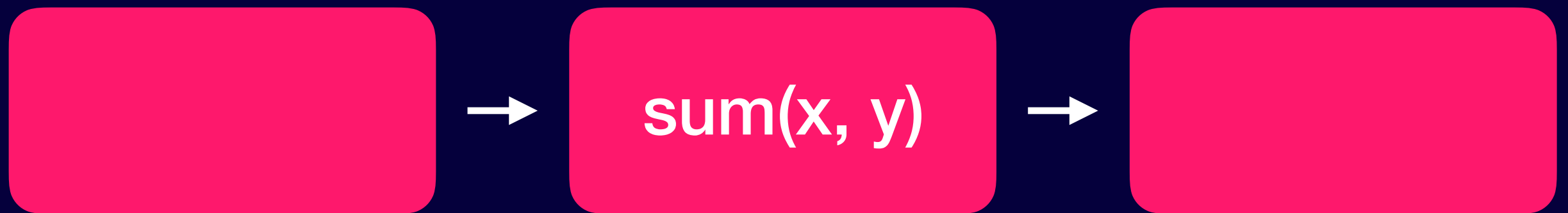
**Ephemeral Environments**

**Built in Scalability**

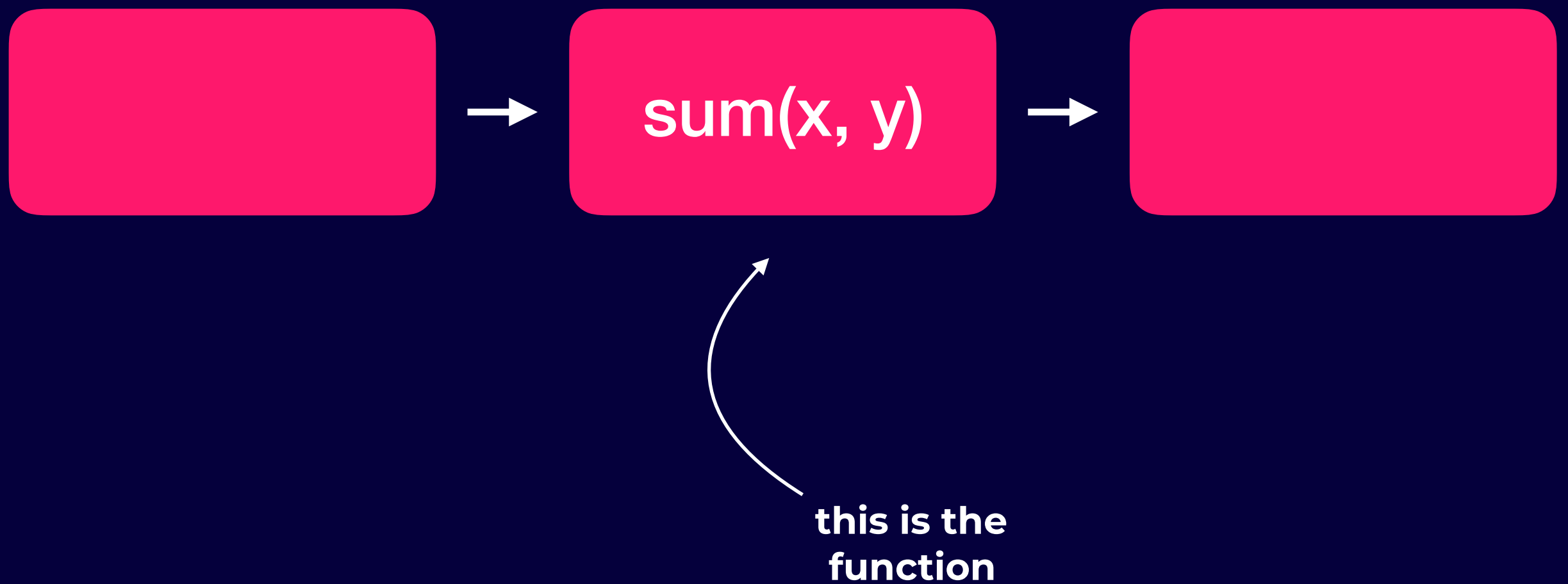
**Pay by Execution**

**Lack of Management**

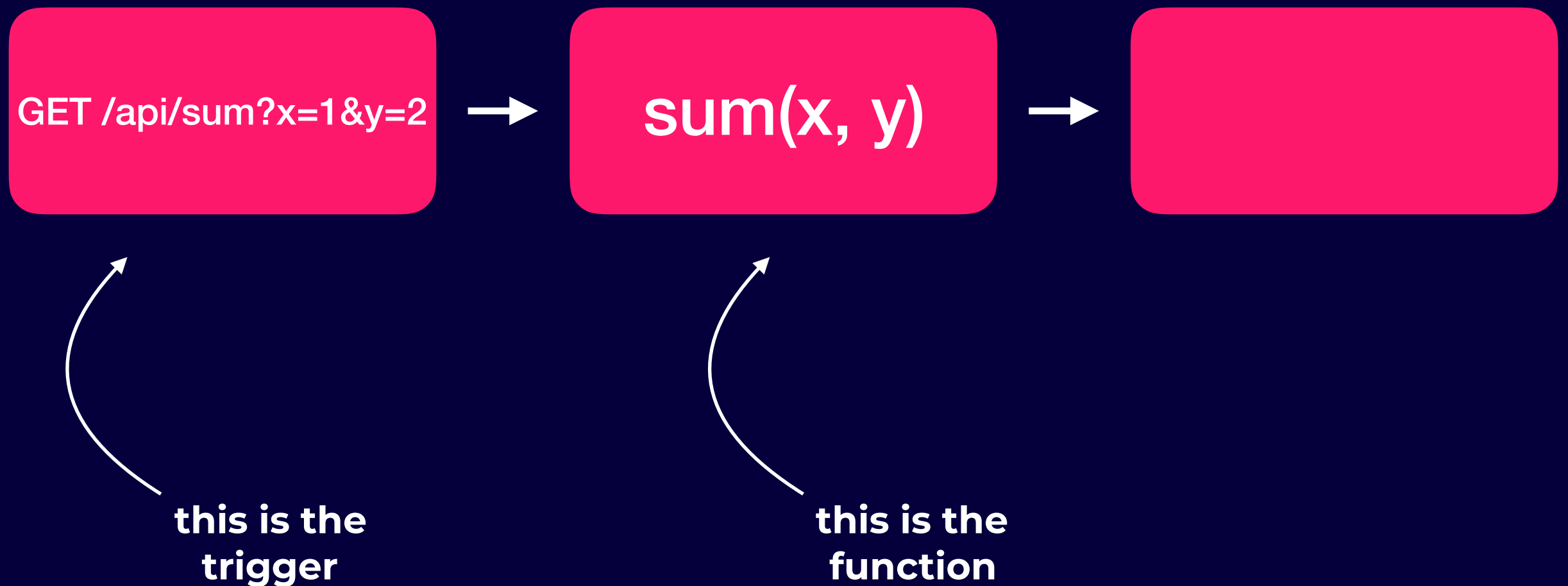
# Event Driven



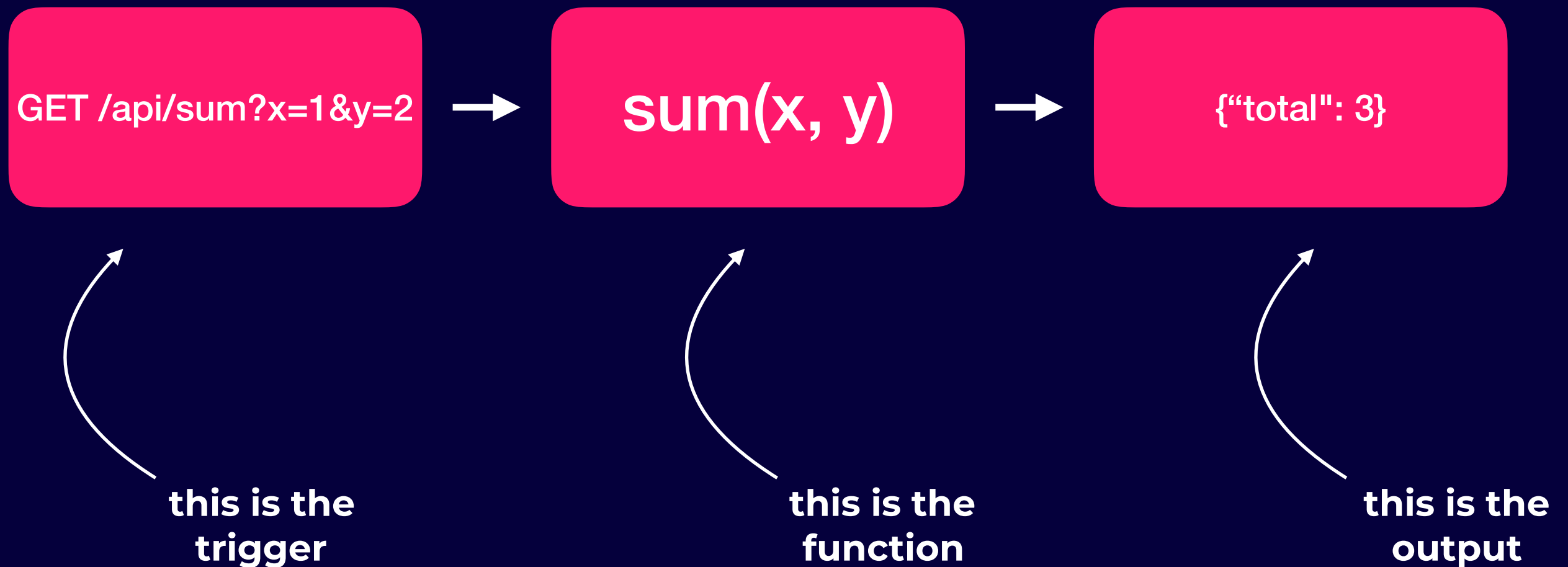
# Event Driven



# Event Driven



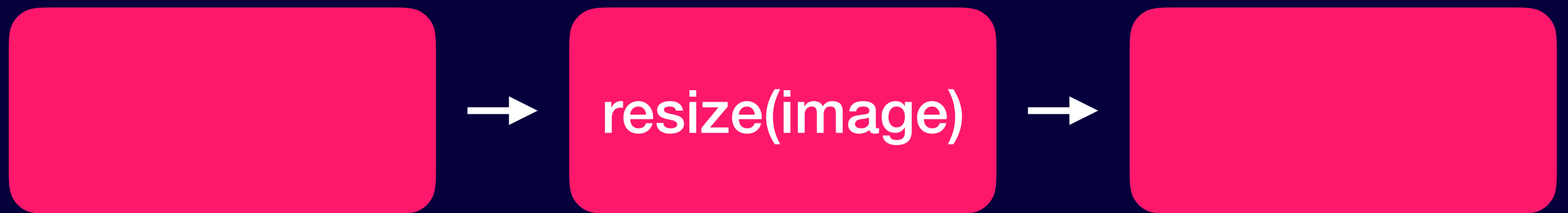
# Event Driven



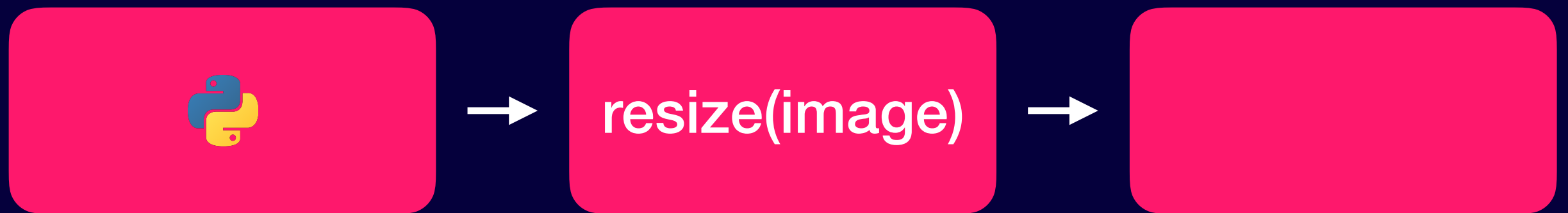
# Event Driven



# Event Driven

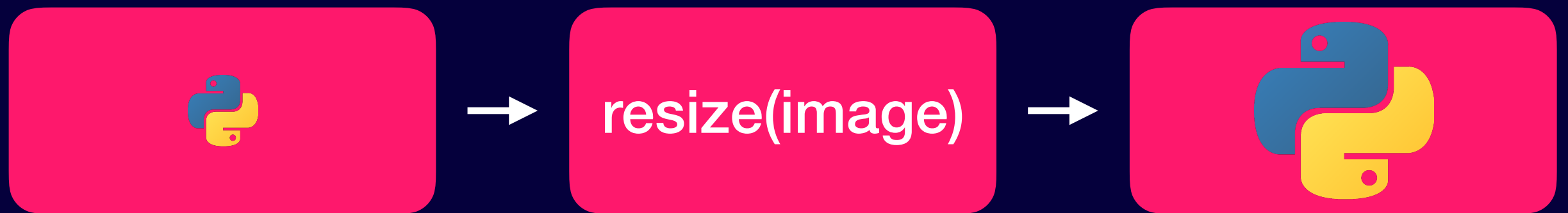


# Event Driven





# Event Driven



# ⚡ FaaS Principles ⚡

Event Driven

**Ephemeral Environments**

Built in Scalability

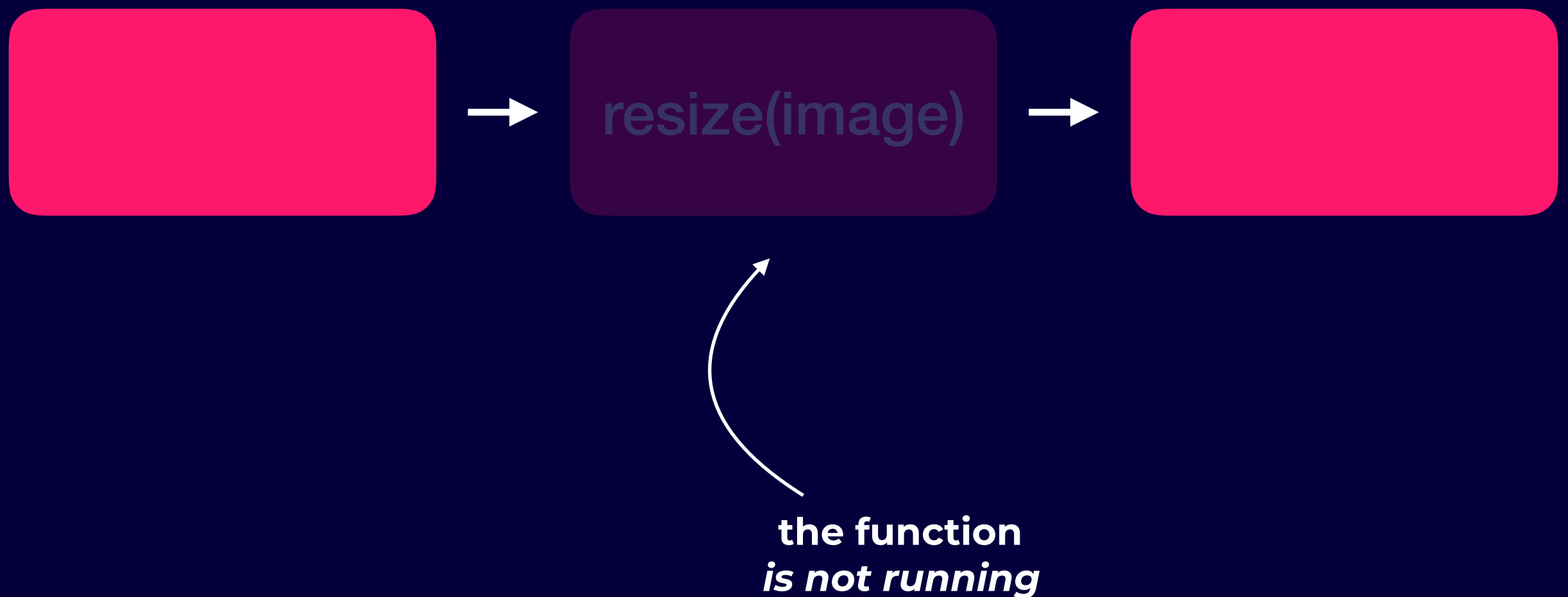
Pay by Execution

Lack of Management

# Ephemeral Environment



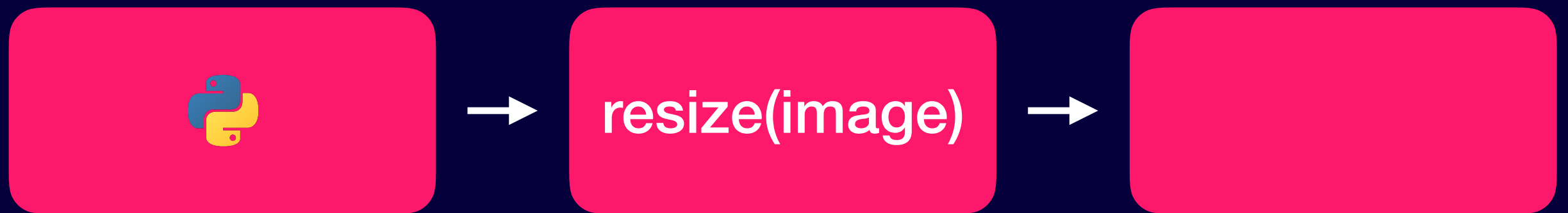
# Ephemeral Environment



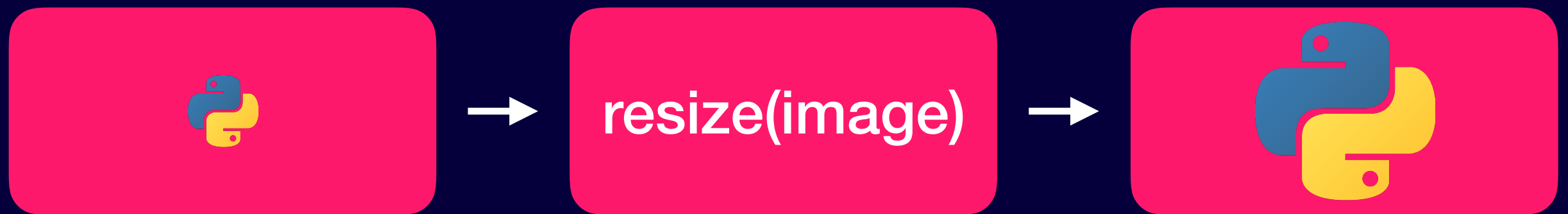
# Ephemeral Environment



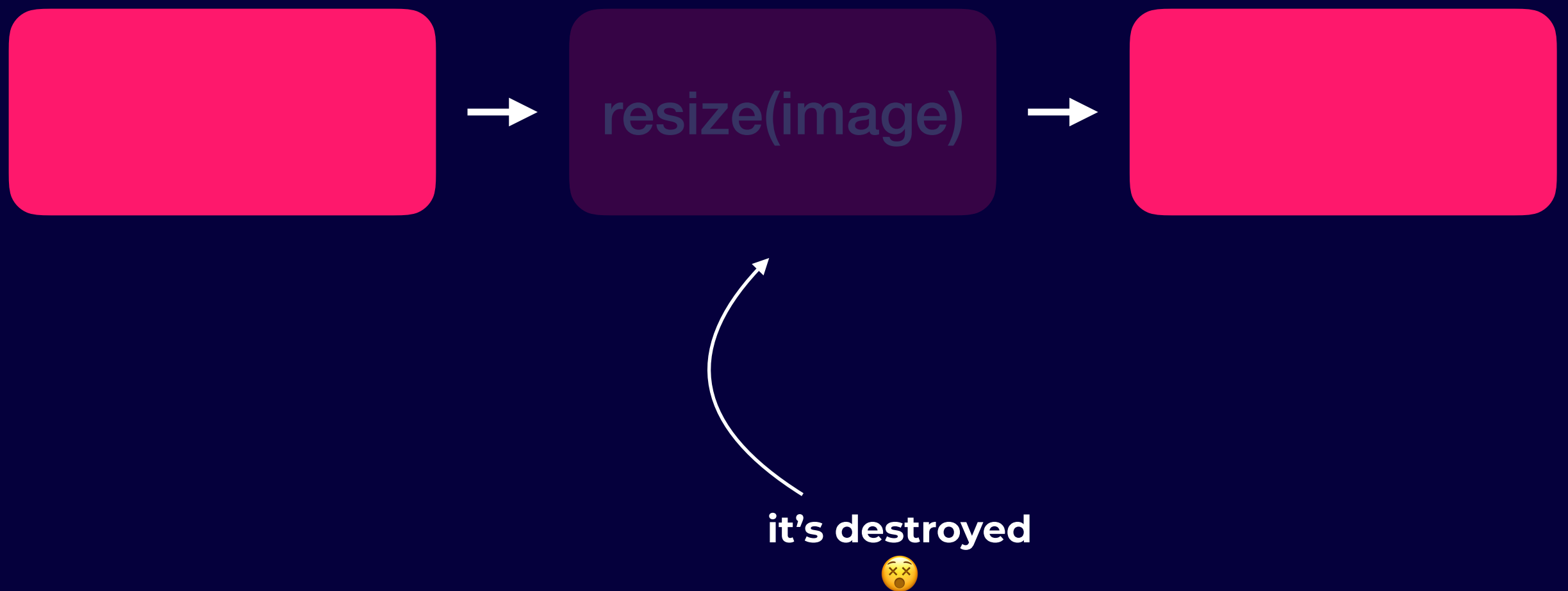
# Ephemeral Environment



# Ephemeral Environment



# Ephemeral Environment





# ⚡ FaaS Principles ⚡

Event Driven

Ephemeral Environments

**Built in Scalability**

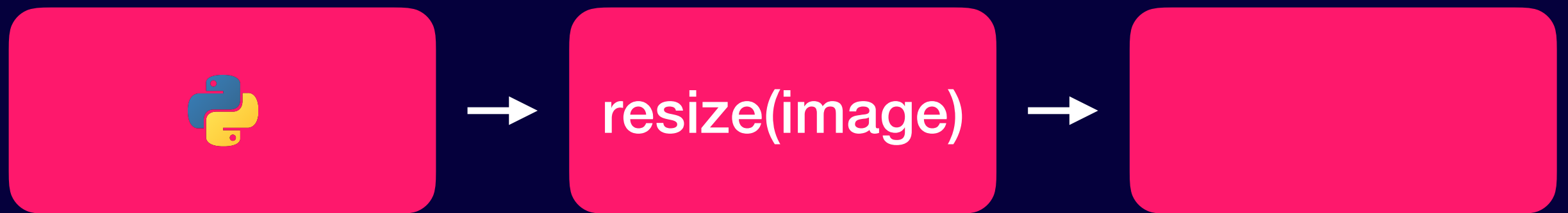
Pay by Execution

Lack of Management

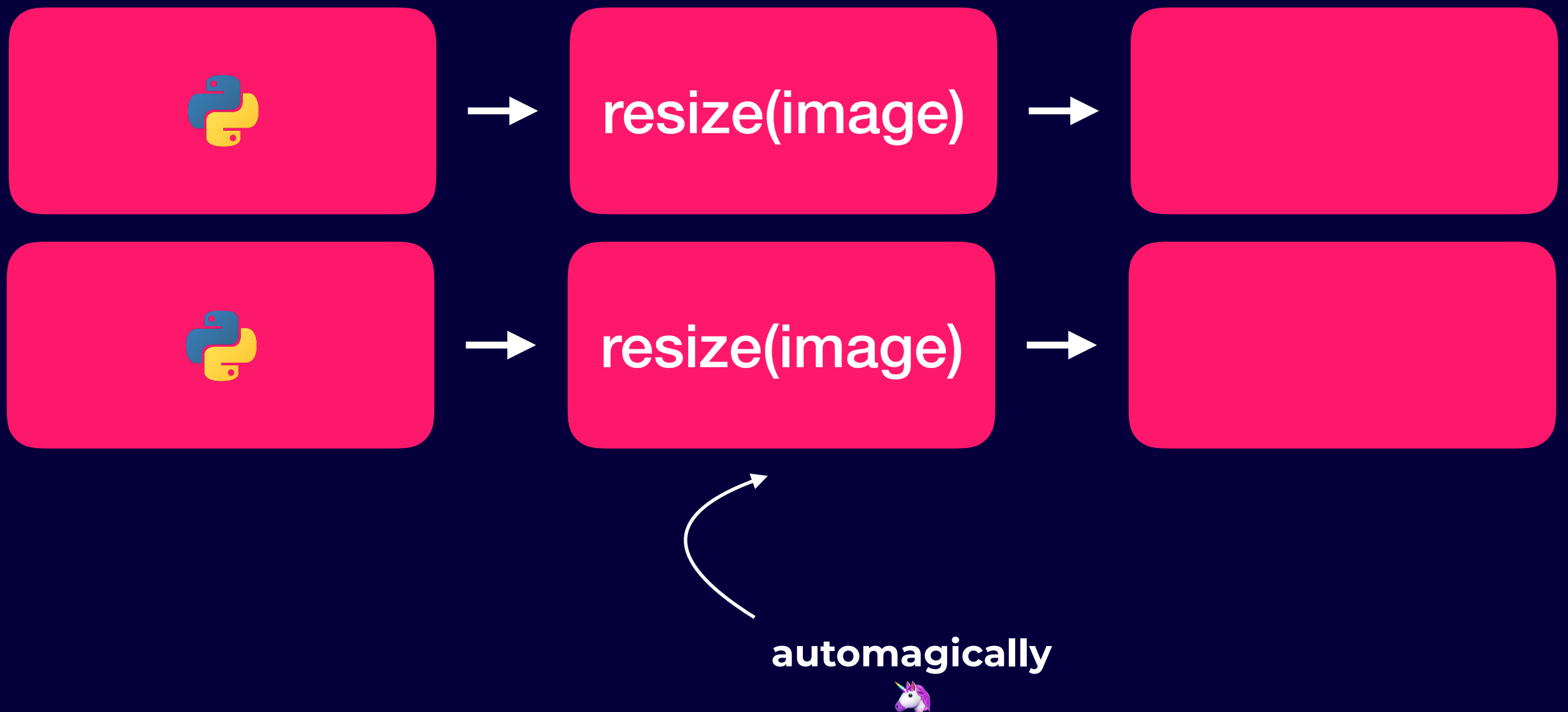
# Built in Scalability



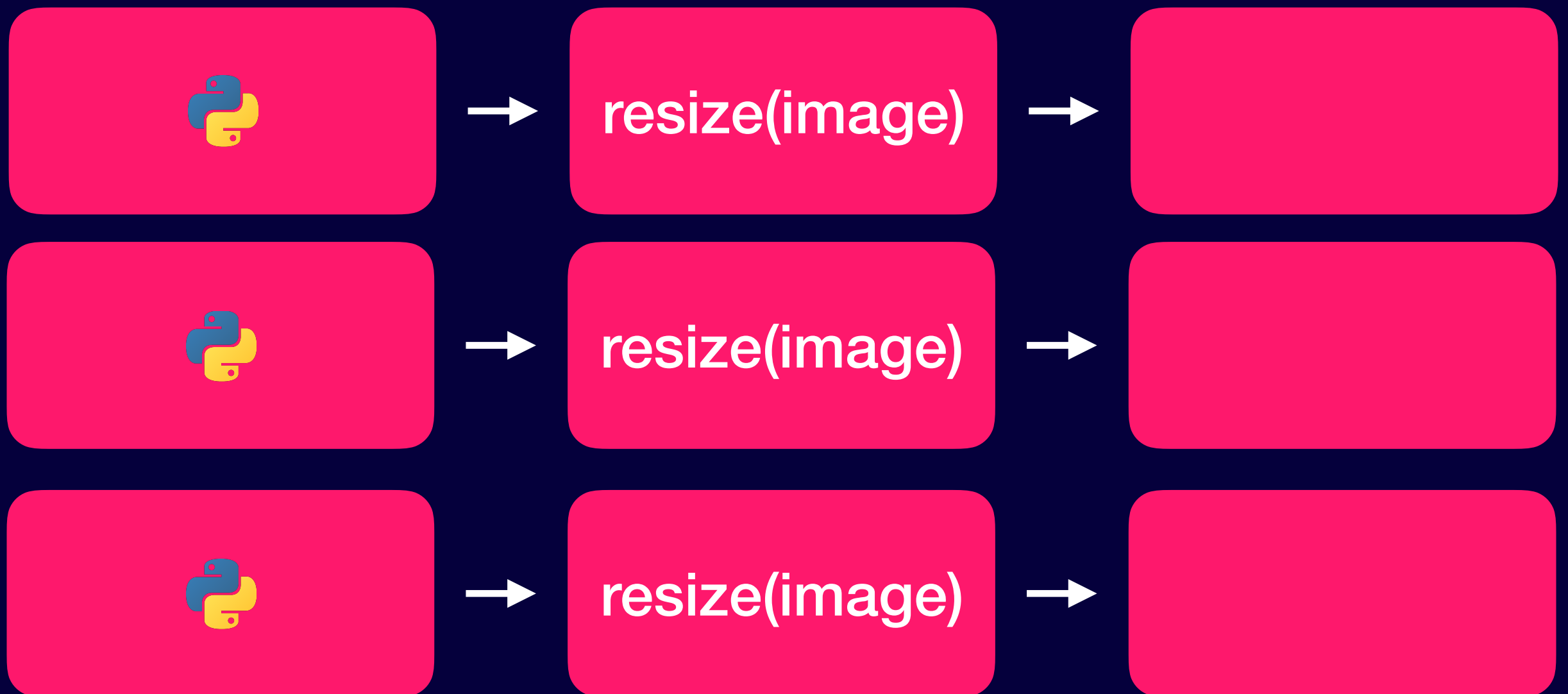
# Built in Scalability



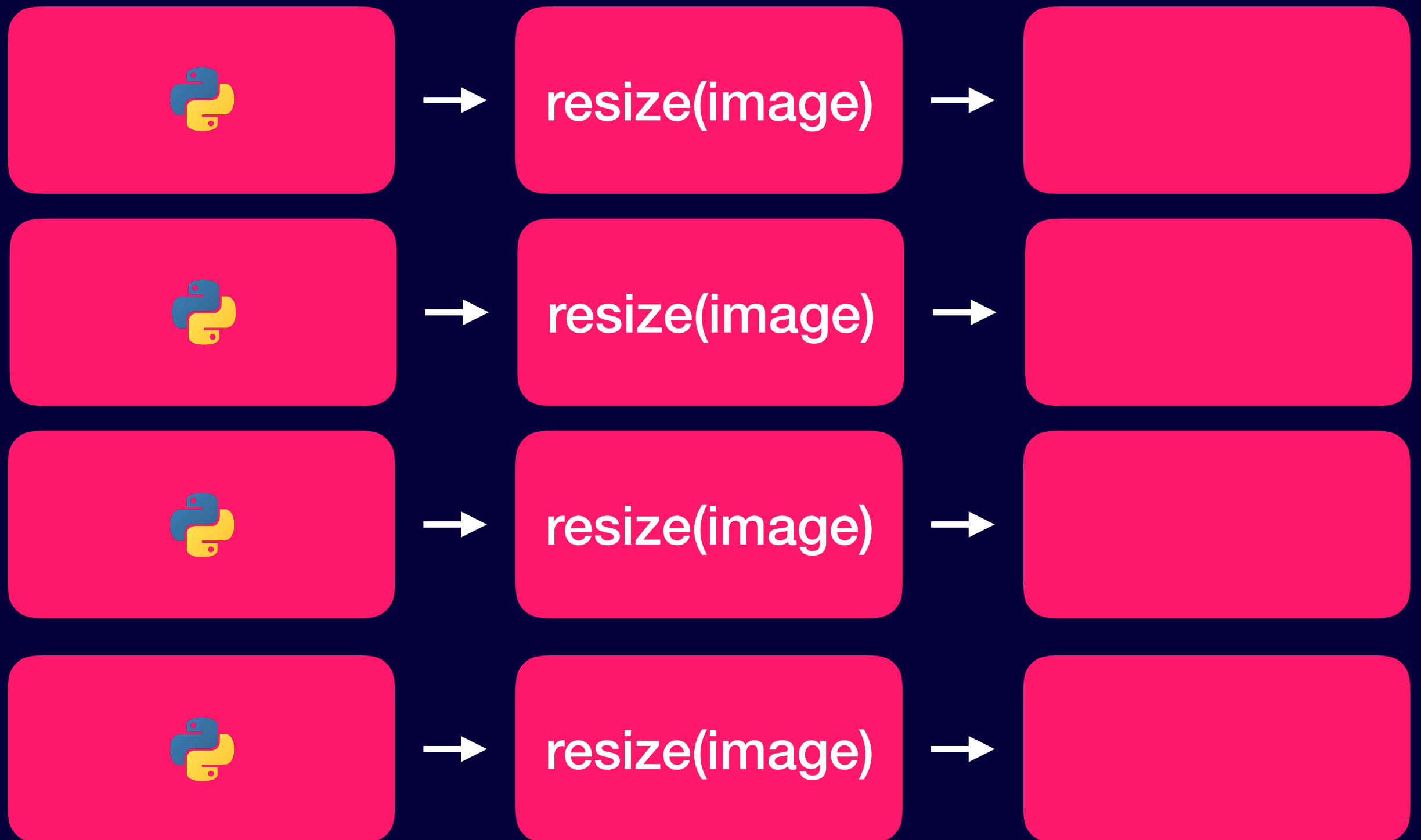
# Built in Scalability

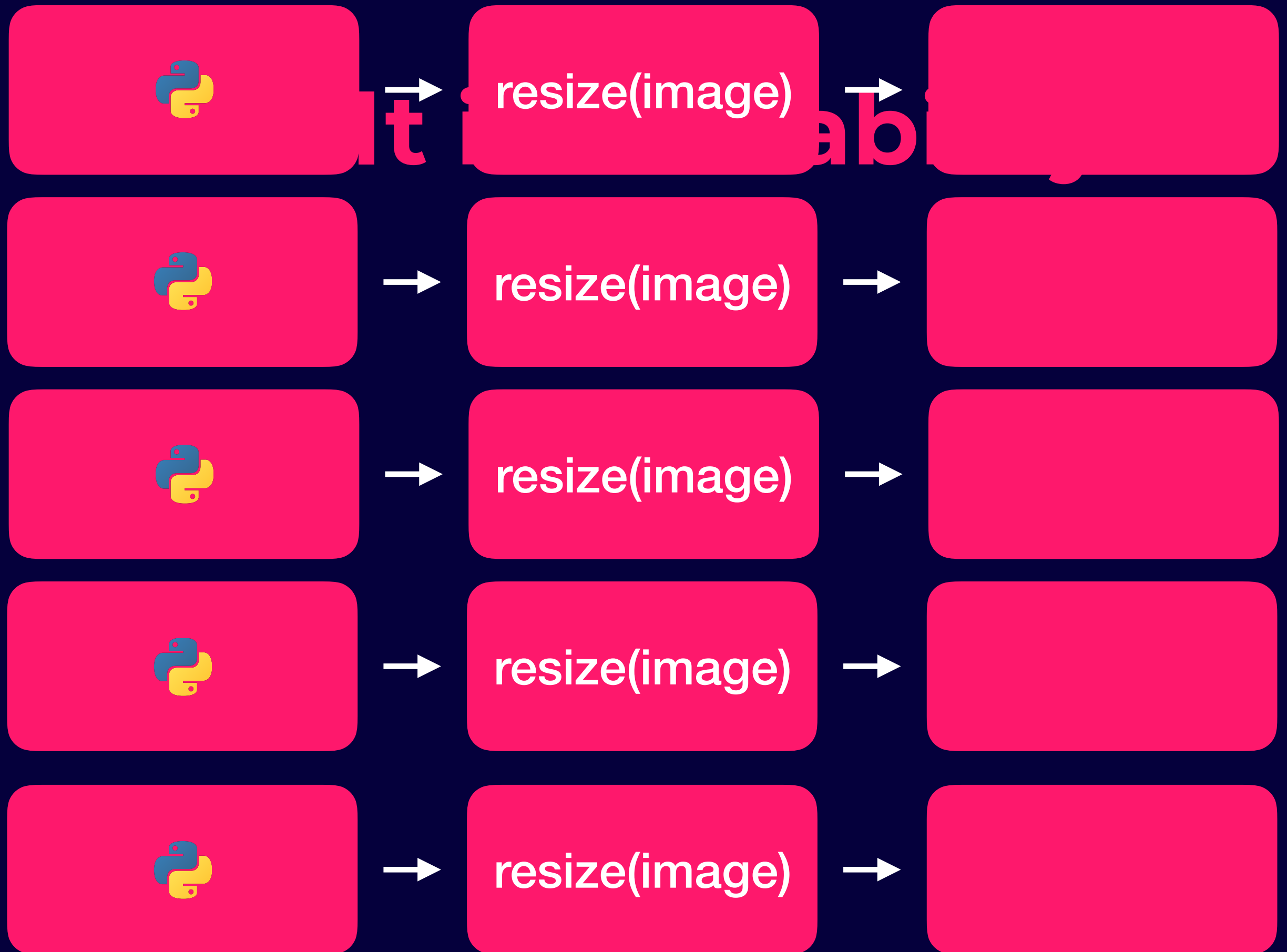


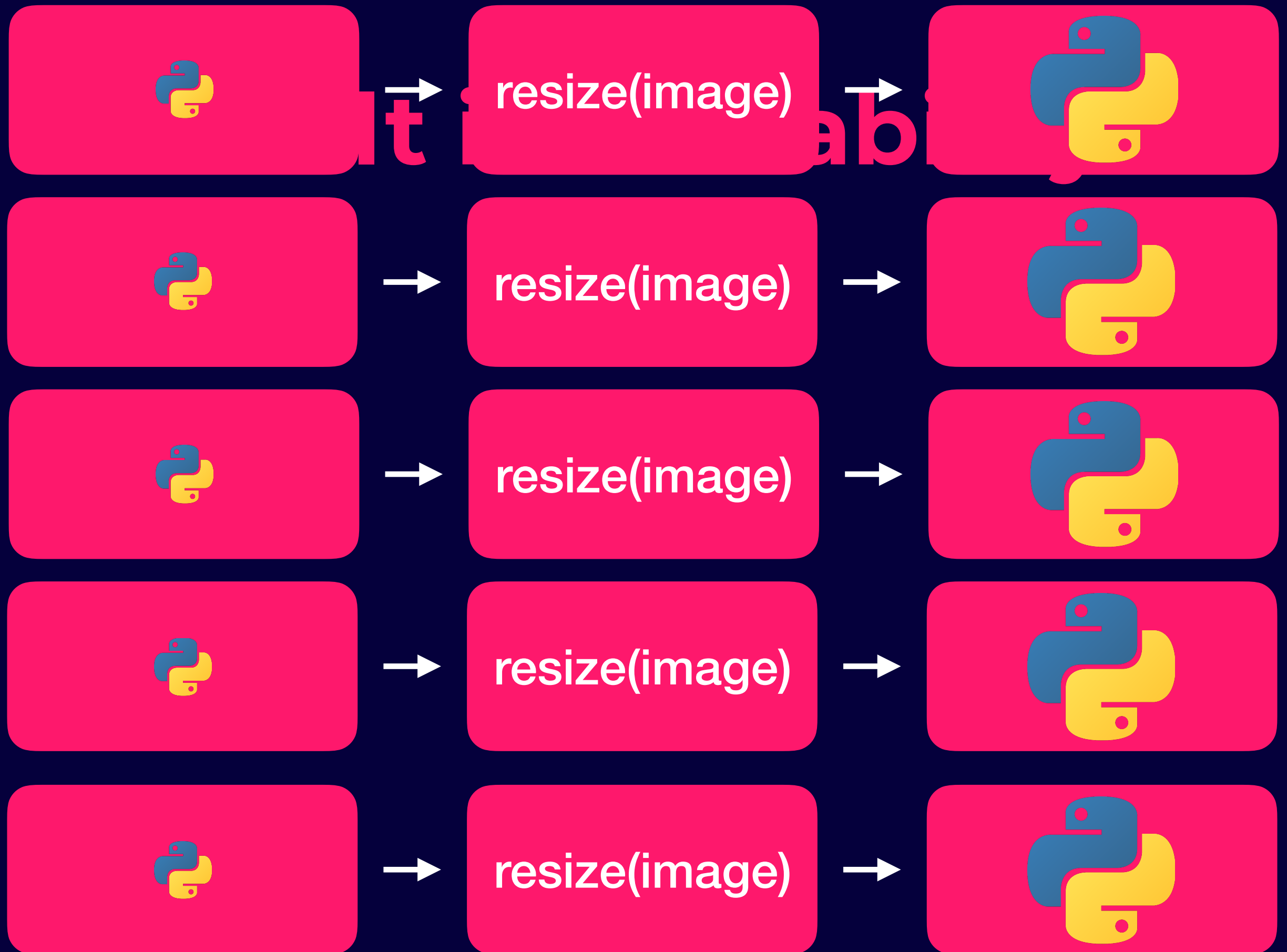
# Built in Scalability



# Built in Scalability









# Built in Scalability



# ⚡ FaaS Principles ⚡

Event Driven

Ephemeral Environments

Built in Scalability

**Pay by Execution**

Lack of Management

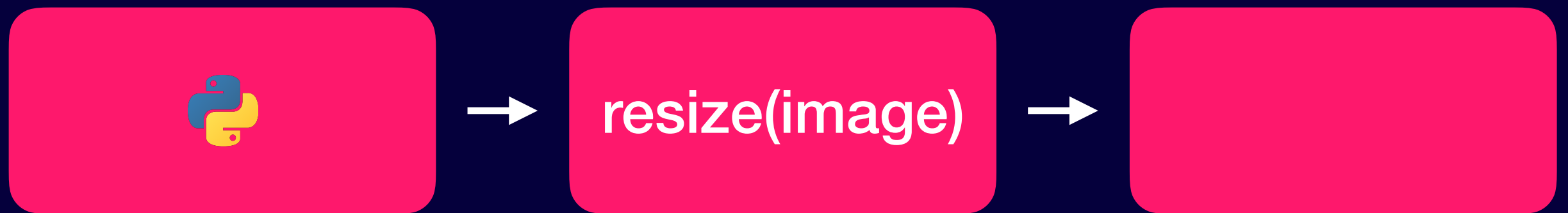
# Pay by Execution



# Pay by Execution



# Pay by Execution



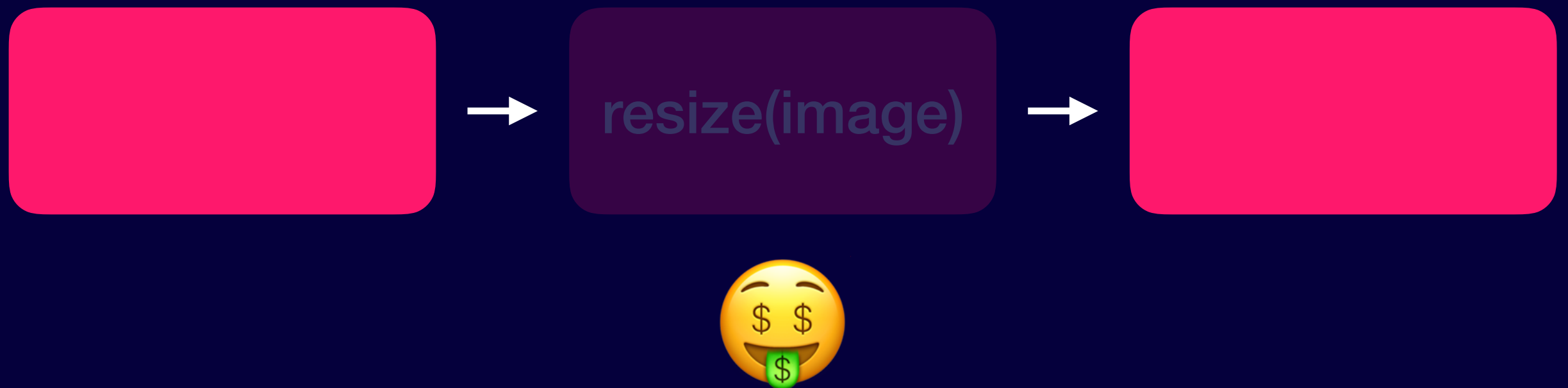
# Pay by Execution



# Pay by Execution



# Pay by Execution





# ⚡ FaaS Principles ⚡

Event Driven

Ephemeral Environments

Built in Scalability

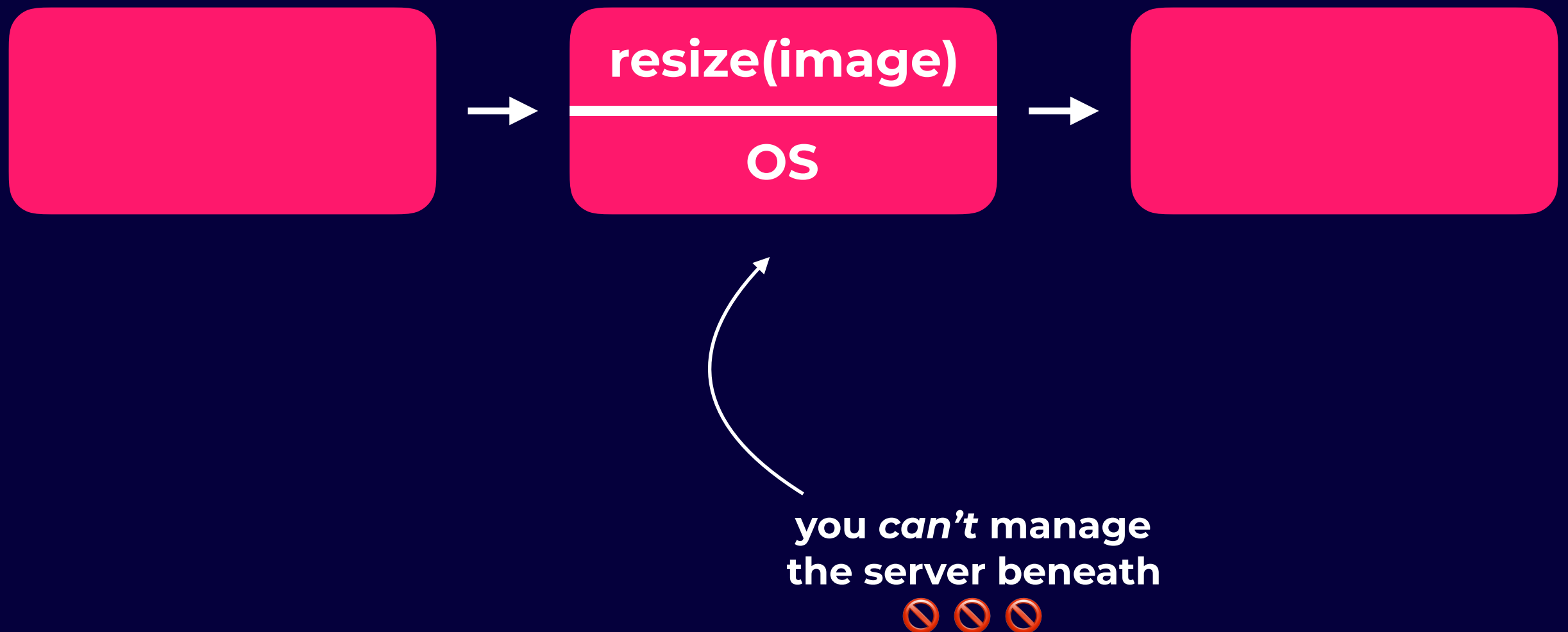
Pay by Execution

**Lack of Management**

# Lack of Management

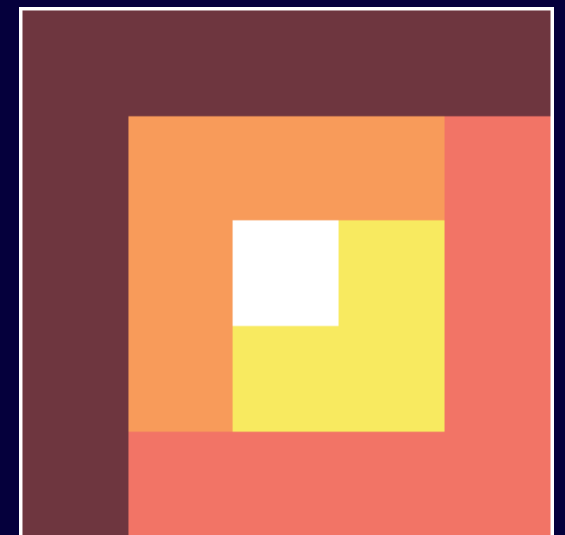
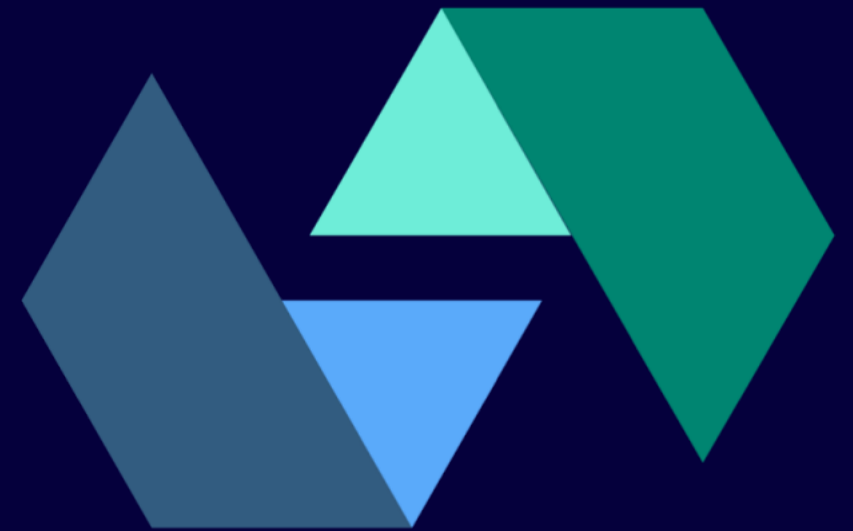


# Lack of Management

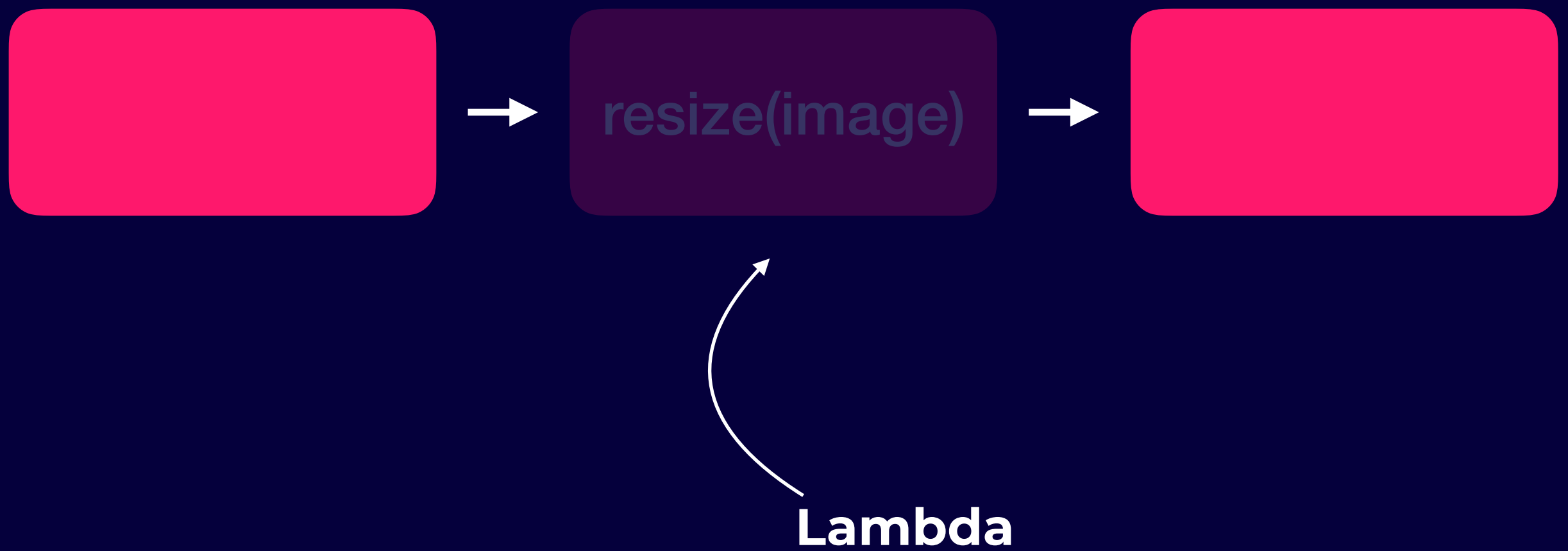


**but where does the  
code *actually run*?**

# Providers



# AWS



API Gateway  
S3  
SQS  
DynamoDB  
...

**AWS**



**Lambda**

API Gateway  
S3  
SQS  
DynamoDB  
...

**AWS**

Anything, really



**Lambda**





let's open the  
black (pink) box

# What about the Pythonistas?



# Python 2.7



**Kubeless**

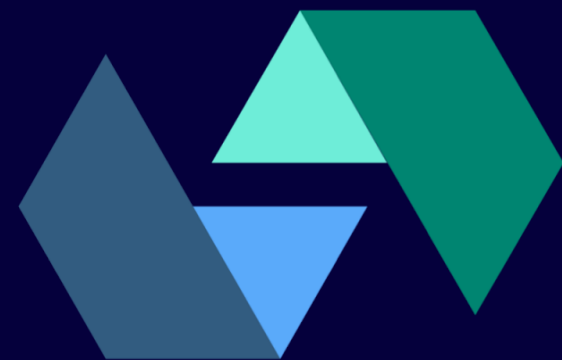


**Azure (experimental)**

# Python 2 & 3



**AWS Lambda**



**IBM Cloud Functions**

♥ Frameworks ♥



```
$ pip install chalice
$ chalice new-project sum && cd sum

# app.py
from chalice import Chalice

app = Chalice(app_name='sum')

def sum(x, y):
    return int(x) + int(y)

@app.route('/')
def index():
    x = app.current_request.query_params.get('x')
    y = app.current_request.query_params.get('y')
    total = sum(x, y)

    return {'total': total}

$ chalice deploy
```

# aws/chalice

A large crowd of people is seen from behind, looking towards a stage. The stage features a large, white, stylized word 'Zappa' on a dark background. Bright stage lights illuminate the word and the crowd. A person in the crowd is holding up a smartphone to take a photo.

**Zappa**



# Serverless Framework

The open-source CLI for building serverless architectures and event-driven applications.

**Python is great  
for Serverless!**





# Moria

## Zappa + Django

<https://moriatoken.com>



**Zappa + Flask Ask + Alexa**

<https://bondhome.io/>



Jonatas Baldin  
/start

✓✓ 17:38



Serverless Telegram Bot

17:38

Hello, human! I am an echo bot, built with Python and the Serverless Framework.

You can take a look at my source code here: <https://github.com/jonatasbaldin/serverless-telegram-bot>.

If you have any issues, please drop a tweet to my creator: <https://twitter.com/jonatsbaldin>.  
Happy botting!

GitHub

jonatasbaldin/serverless-telegram-bot

serverless-telegram-bot - This example demonstrates how to setup an echo

Telegram Bot using the Serverless Framework ⚡🤖



Jonatas Baldin  
I'm an echo bot!

✓✓ 17:38



Serverless Telegram Bot  
I'm an echo bot!

17:38

# @sls\_telegram\_bot

<https://github.com/jonatasbaldin/serverless-telegram-bot>

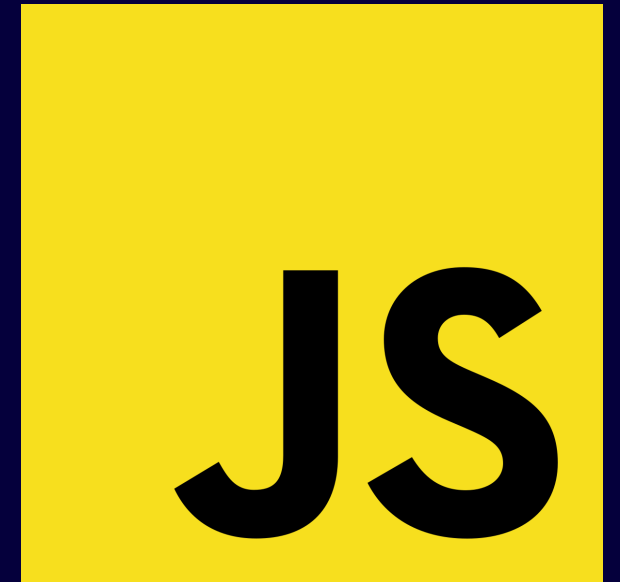
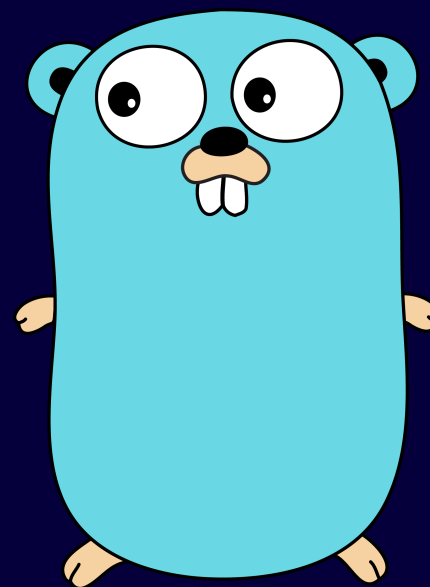


# TensorFlow Predictions

[\*\*https://github.com/jacopotagliabue/  
tensorflow\\_to\\_lambda\\_serverless\*\*](https://github.com/jacopotagliabue/tensorflow_to_lambda_serverless)



**VS**



**C#**

**Serverless Notes <sup>TM</sup>**

**Things you should  
be aware**

# Costs





# Vendor Lock-in



**No server  
optimisations**



Visibility and testing  
are *harder*



# Youngness



**Future?**

**Lots of opportunities  
for improvements!**



**@jonatasbaldin**

# Serverless for Pythonistas

@jonatasbaldin