

# Dancing with Serverless

December 2022

Peter Sbarski | @sbarski



# About Me

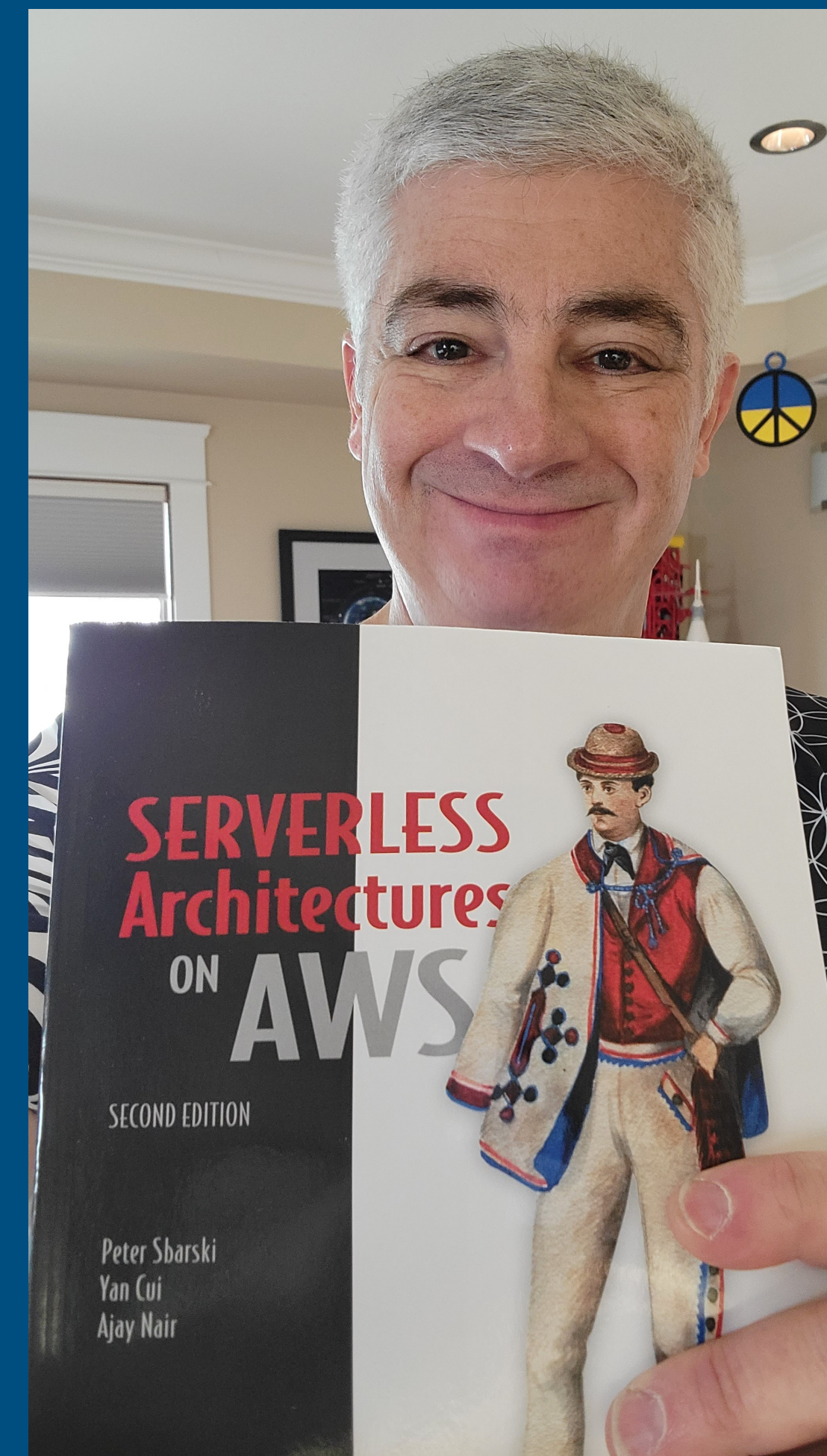
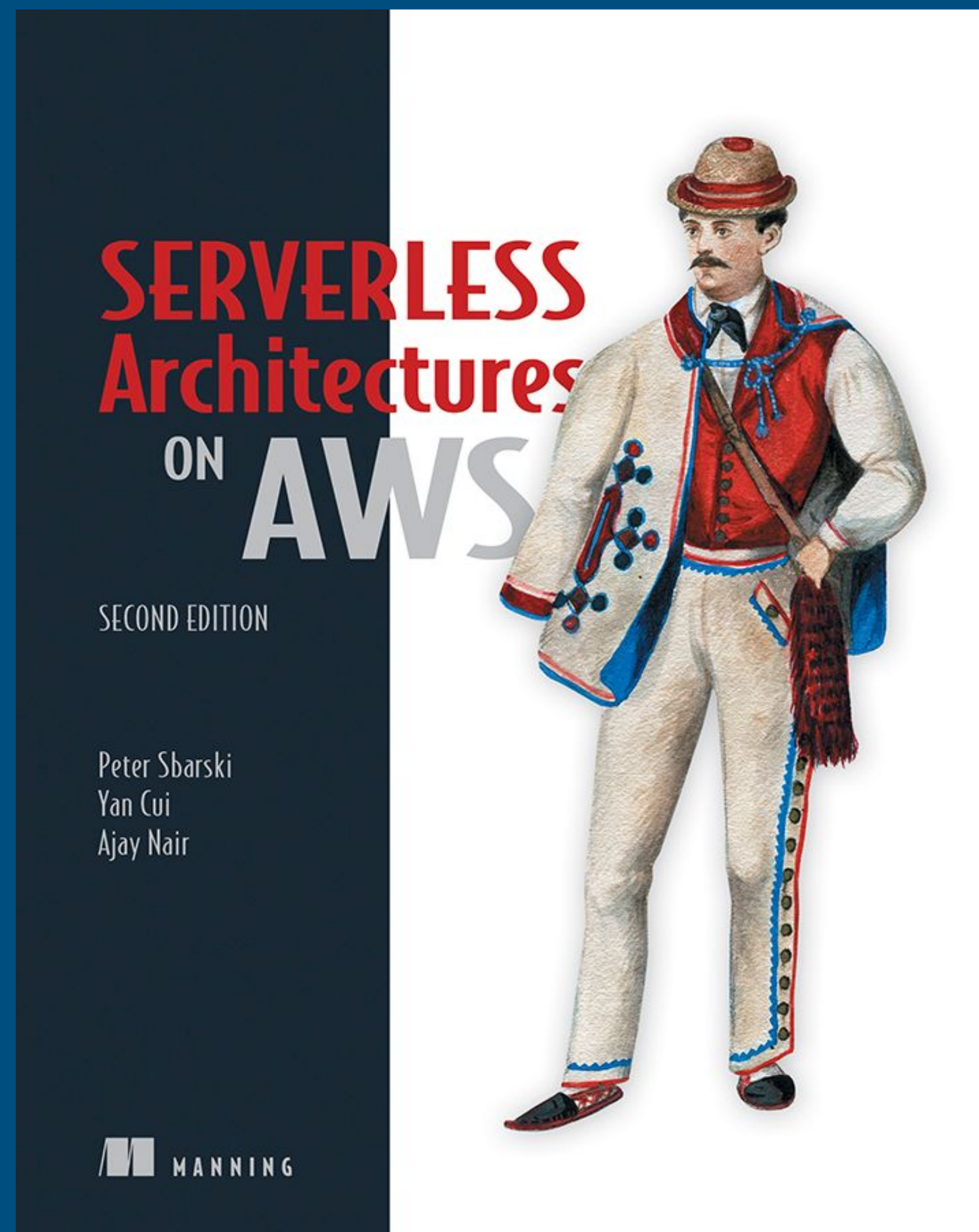


- AWS Serverless Hero
- Author of Serverless Architectures on AWS
- Organiser of the Melbourne Serverless Meetup
- Former VP Education & Research at A Cloud Guru
- Former head of Serverlessconf





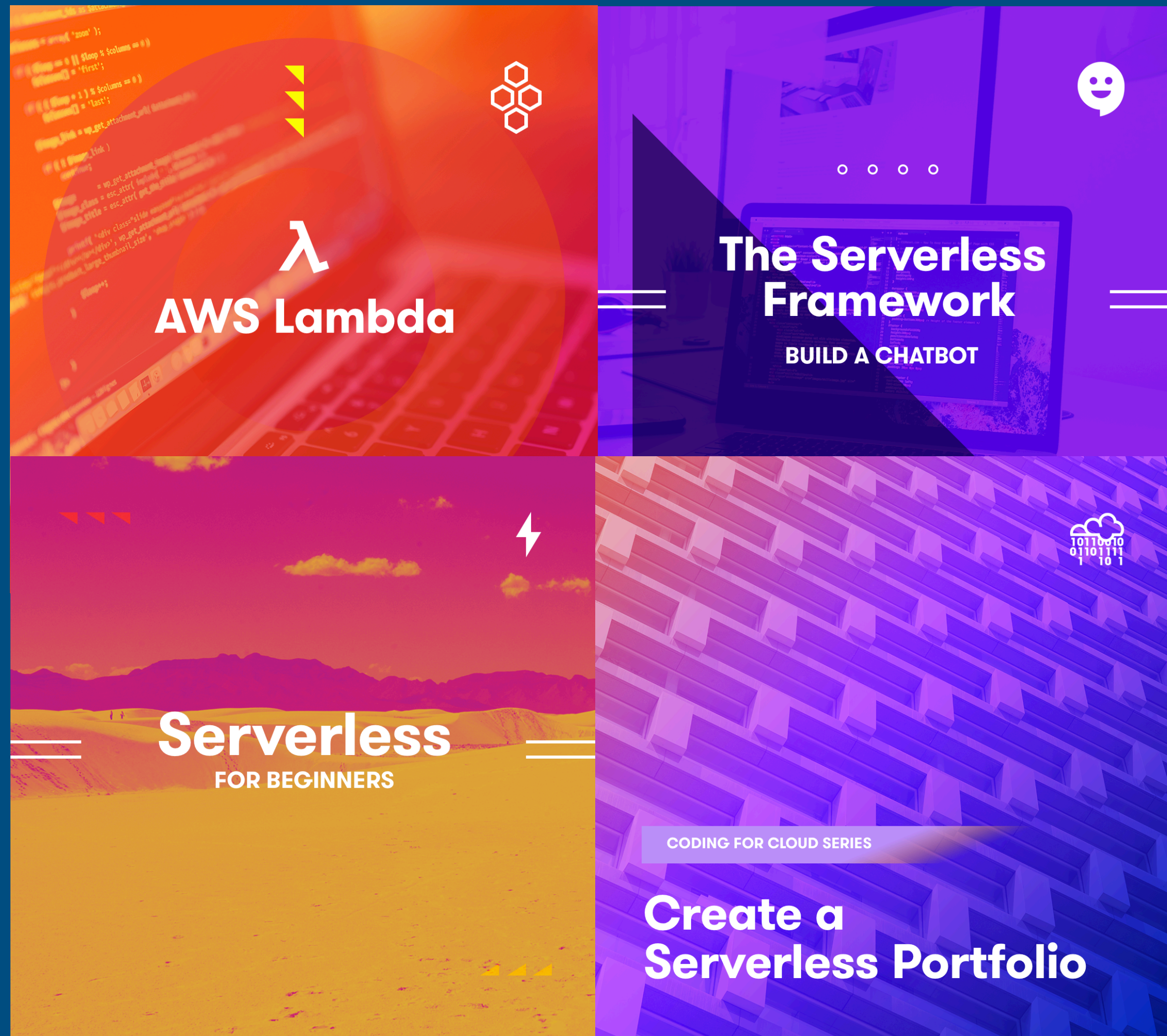
# Serverless Architectures on AWS (2nd Edition)





# Early Days at ACG

## Teaching the world to cloud

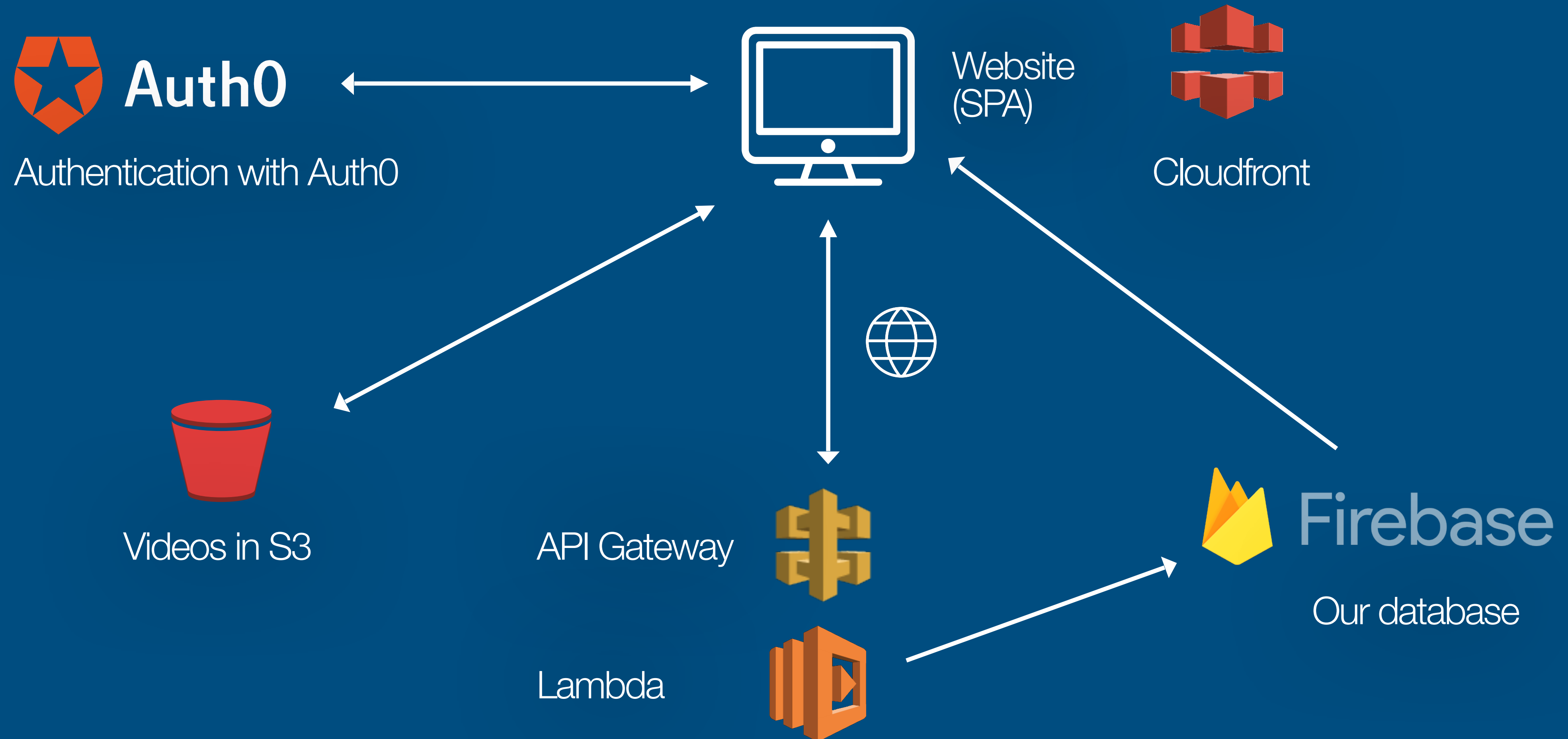


Video Lessons  
Quiz Engine  
Online Store  
Sign Up / Login  
Scale Effortlessly  
Low Operational Overhead



# Starting up in 2015

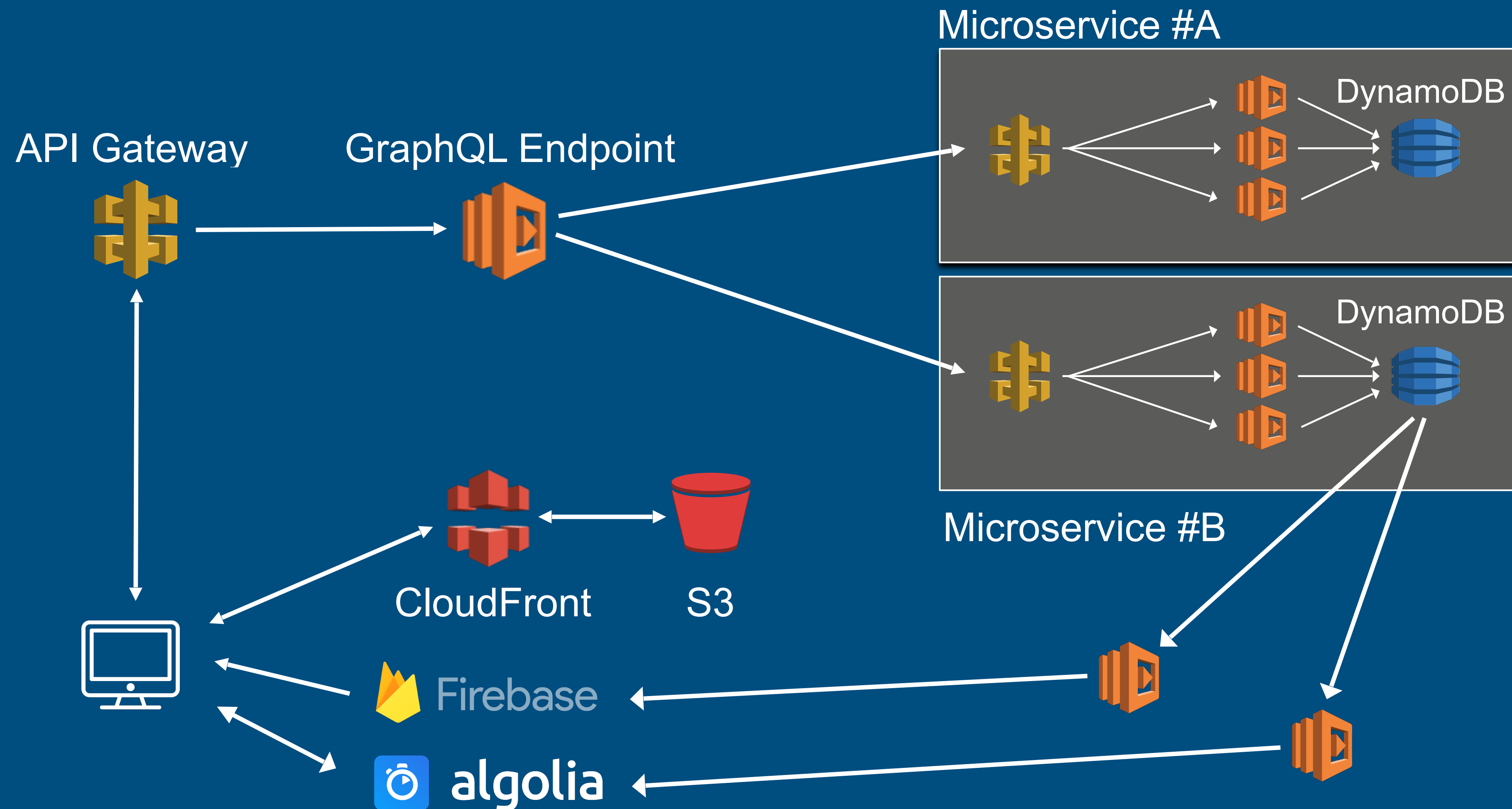
## Serverless Monolith





# Evolution

Teams of developers working in parallel





# We evolved our architecture

And the costs weren't too bad either

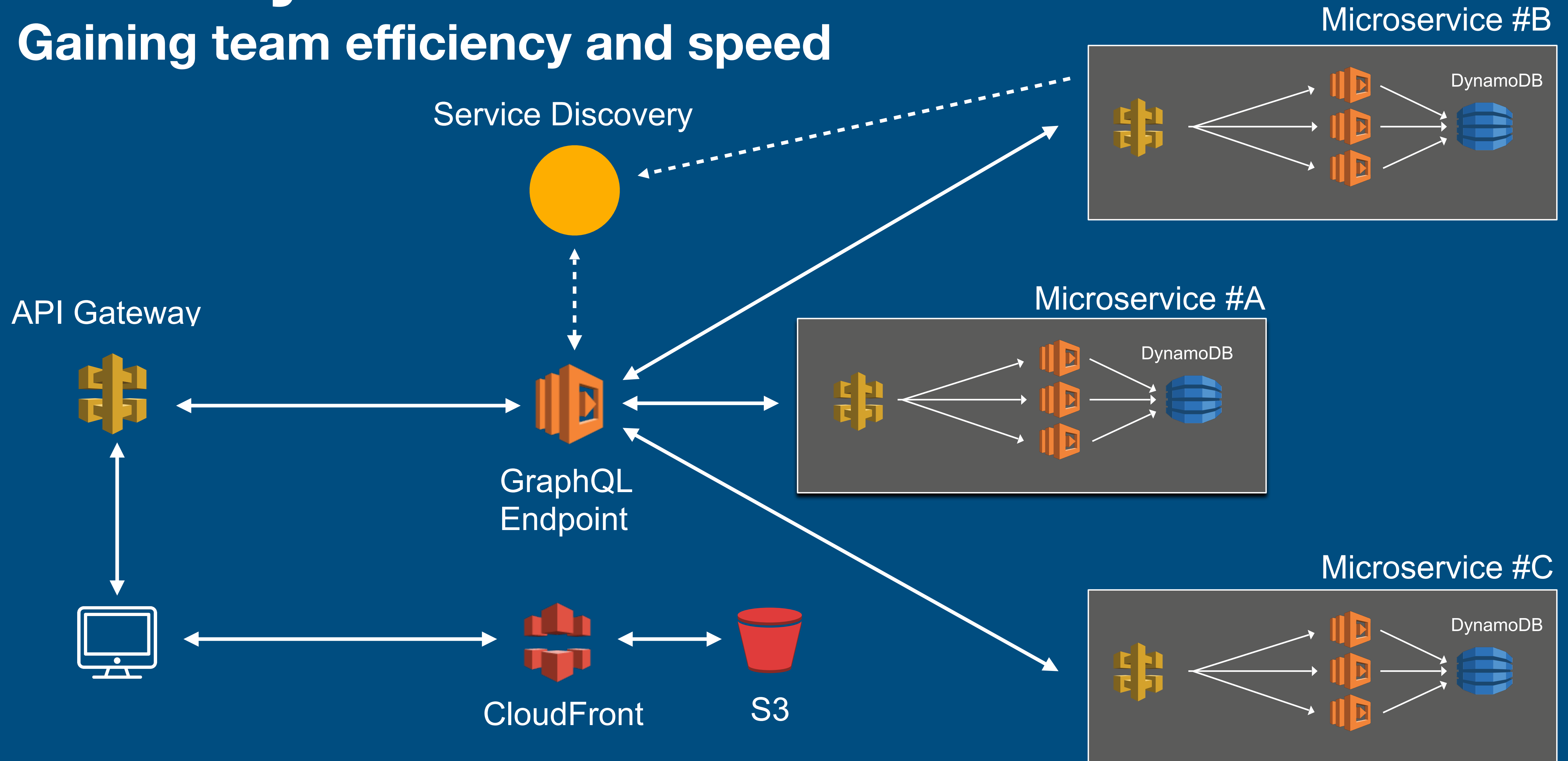
- 289 Lambda Functions
- 19 Micro-services
- 3.68TB of data in S3
- 107m Lambda Invocations / month
- 45m API Requests / month
- 3.8+ TB of data via CloudFront per day
- 650K+ users

Service	Cost
Key Management Service	\$25.26
Simple Storage Service (S3)	\$108.23
Config	\$109.84
Elastic Transcoder	\$154.17
<b>API Gateway</b>	<b>\$192.14</b>
Developer Support	\$314.59
Redshift	\$371.50
DynamoDB	\$373.54
<b>Lambda</b>	<b>\$706.49</b>
CloudWatch	\$3,142.73
CloudFront	\$5,099.85



# Maturity

Gaining team efficiency and speed





# Common Benefits

## When things go right

- It's fast to build (shortened time to market)
- Massive scale and initially can be cheap or free
- It's operationally efficient
- It's not Kubernetes
- Large architectural pivots/changes are possible
- It's fun - developers love it

# Common Complaints

Why can't things just be easy

- Hard to dev locally
- Hard to debug
- Hard to observe and monitor
- Hard or impossible to do certain things (e.g. long-running tasks)
- Lock-in is a problem. Maybe?



# How I build today

## Back to working on my own

- Back to building on my own
- Serverless first approach makes technical decisions easier
- Have to be fast and reduce operational overhead to zero
- Could potentially need a lot of compute down the road
- Leverage as many AWS services as possible
- Solve local dev & debugging

# fatfireapp.com

## Learning & Optimising Finance

fatfire

Dashboard

YOUR FINANCIALS

Bank Accounts

Shares & ETFs

Property

NET WORTH

Net Worth

FIRE

EXTRAS

Transactions

Insurance

Receipts

Taxation

Portfolio Test 2

Welcome Peter Sbarski, everything seems great!

\$1,265,594.4

ASSETS

\$-241,560.68

LIABILITIES

\$1,024,033.72

NET WORTH

ASSET	CATEGORY
Plaid Checking	Bank Accounts
Online ING	Bank Accounts
Orange One Low Rate Platinum	Bank Accounts
Superannuation	Bank Accounts

fatfire

Dashboard

YOUR FINANCIALS

Bank Accounts

Shares & ETFs

Property

NET WORTH

Net Worth

AUD

Portfolio Test 2

Peter Sbarski

Link to Bank

Refresh Data

Rename

Currency

Remove

Asset		Value
Checking Account	# transaction	AUD 90,892.32
Credit Card	# credit-card	AUD 6,972.71
Credit Card 13000	# credit-card	AUD 3,537.64
Hooli Access	# transaction	AUD -25,194.11
Hooli Awards Card	# credit-card	AUD -4,246.62
Hooli Saver	# savings	AUD -20,510.14
Loan for vacation 2017	# loan	AUD 5,521.92
Main	# transaction	AUD 22,397.61

fatfire

Dashboard

YOUR FINANCIALS

Bank Accounts

Shares & ETFs

Property

NET WORTH

Net Worth

FIRE

EXTRAS

Transactions

Insurance

Receipts

Taxation

AUD

Portfolio Test 2

Peter Sbarski

\$1,265,594.4

ASSETS

-0.12%

\$-241,560.68

LIABILITIES

0.60%

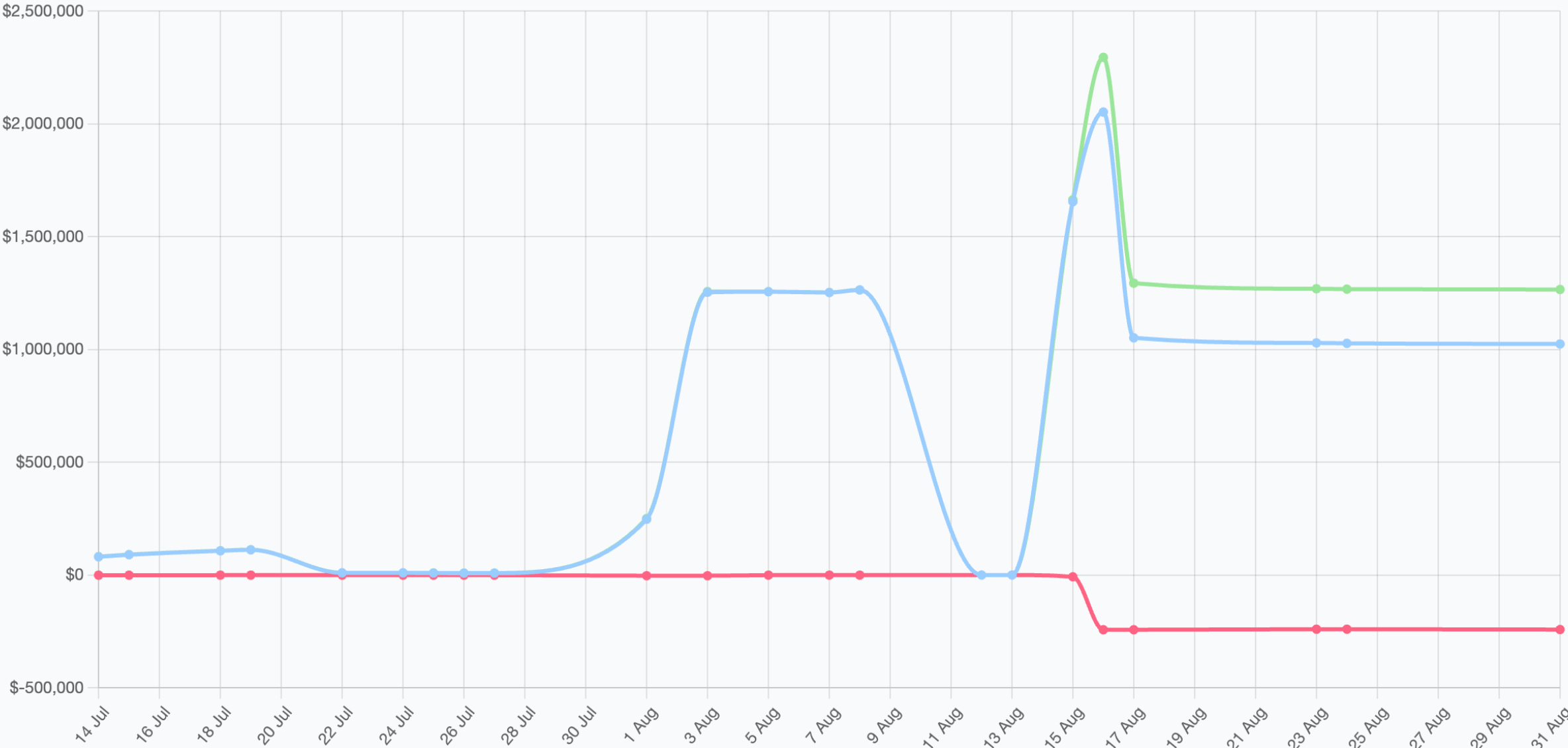
\$1,024,033.72

NET WORTH

-0.28%

Portfolio History

Portfolio Assets Debts



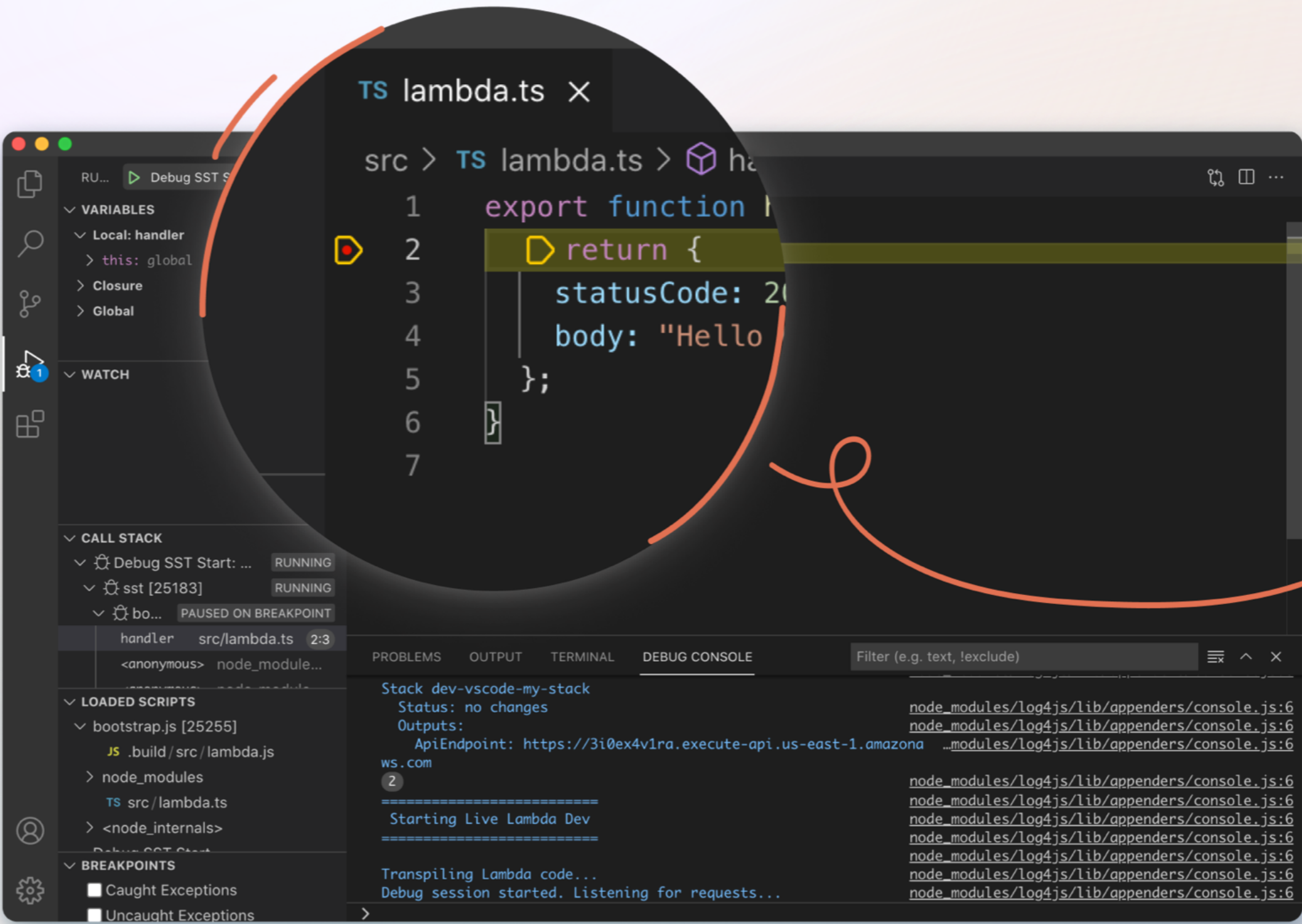


# Serverless Stack (SST)

<https://sst.dev>

Test **your apps live**

Set breakpoints in your Lambda functions and test your apps live. [Learn more >](#)



Set  
breakpoints  
in your functions

sst

Local

Stacks

Functions

API

DynamoDB

RDS

Buckets

GraphQL

Cognito

Invocations

Success	GET /integration/transactions
06:57:37.368	▶ "Request" : {...} 8 items
06:57:37.738	Missing transactions for [] getting cached response false
06:57:38.216	▶ "Response" : {...} 3 items
Success	GET /integration/transactions
06:57:37.306	▶ "Request" : {...} 8 items
06:57:37.736	Missing transactions for [] getting cached response false
06:57:38.200	▶ "Response" : {...} 3 items
Success	GET /integration/transactions
06:57:37.303	▶ "Request" : {...} 8 items
06:57:37.738	Missing transactions for [] getting cached response false
06:57:38.215	▶ "Response" : {...} 3 items
Success	GET /integration/transactions
06:57:37.300	▶ "Request" : {...} 8 items
06:57:37.737	Missing transactions for [] getting cached response false
06:57:38.216	▶ "Response" : {...} 3 items

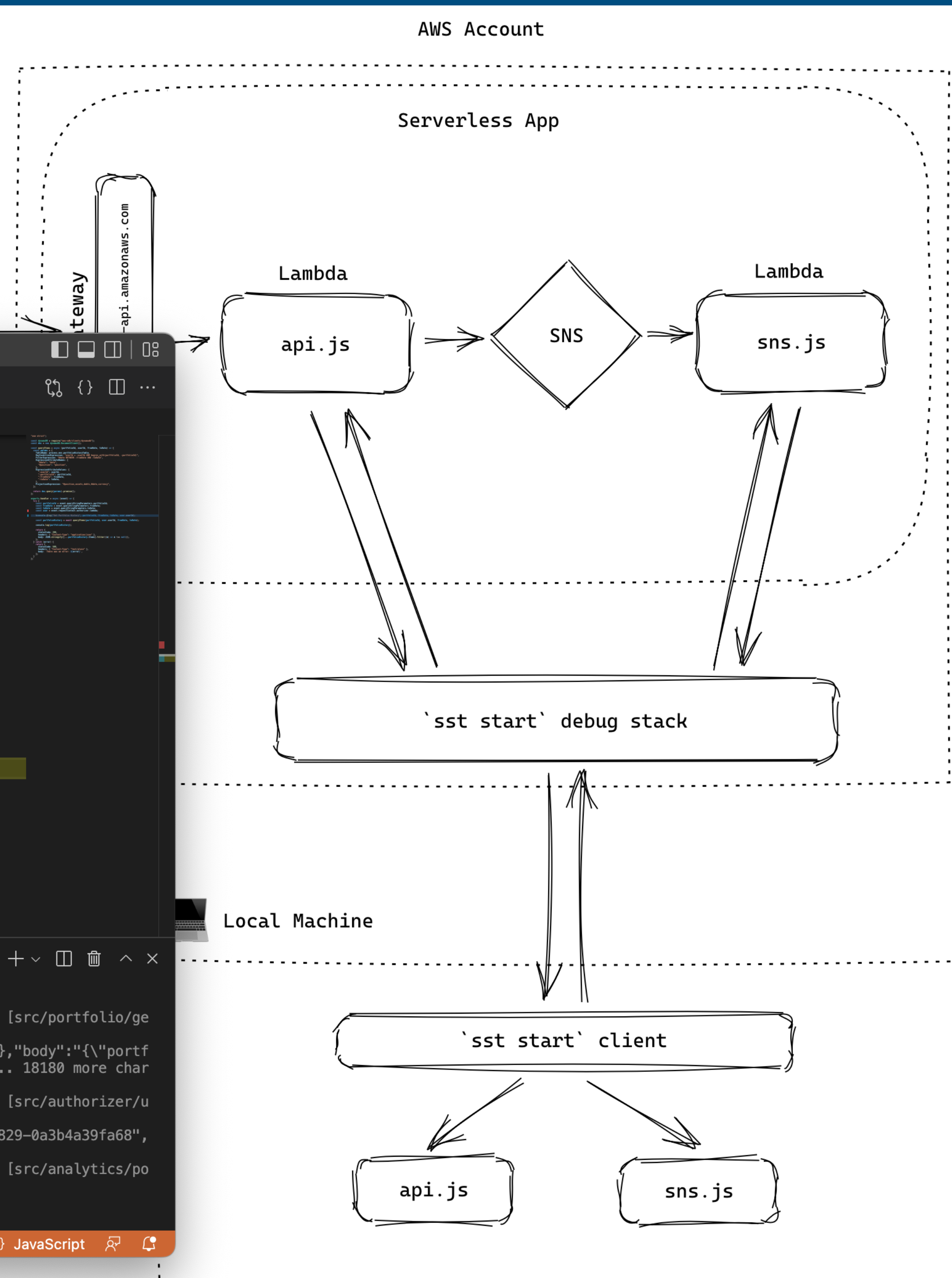
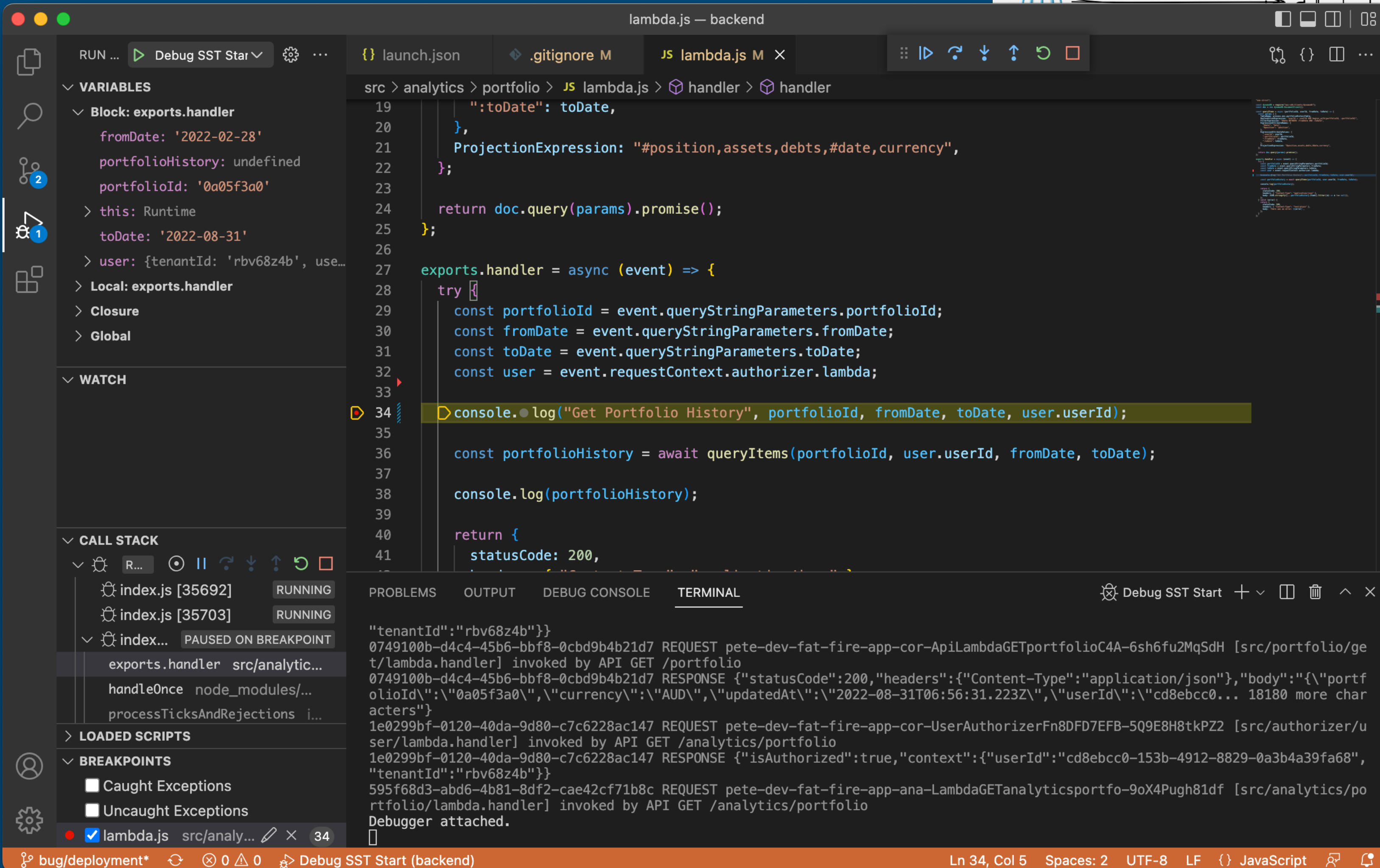
pete-dev





# Serverless Stack (SS)

## Local Dev. Absolutely Brilliant.





# Observability

https://lumigo.io

Dashboard

Issues

Functions

ECS

Transactions

Live Tail

Explore

Resources

System Map

Alerts

Settings

What's New

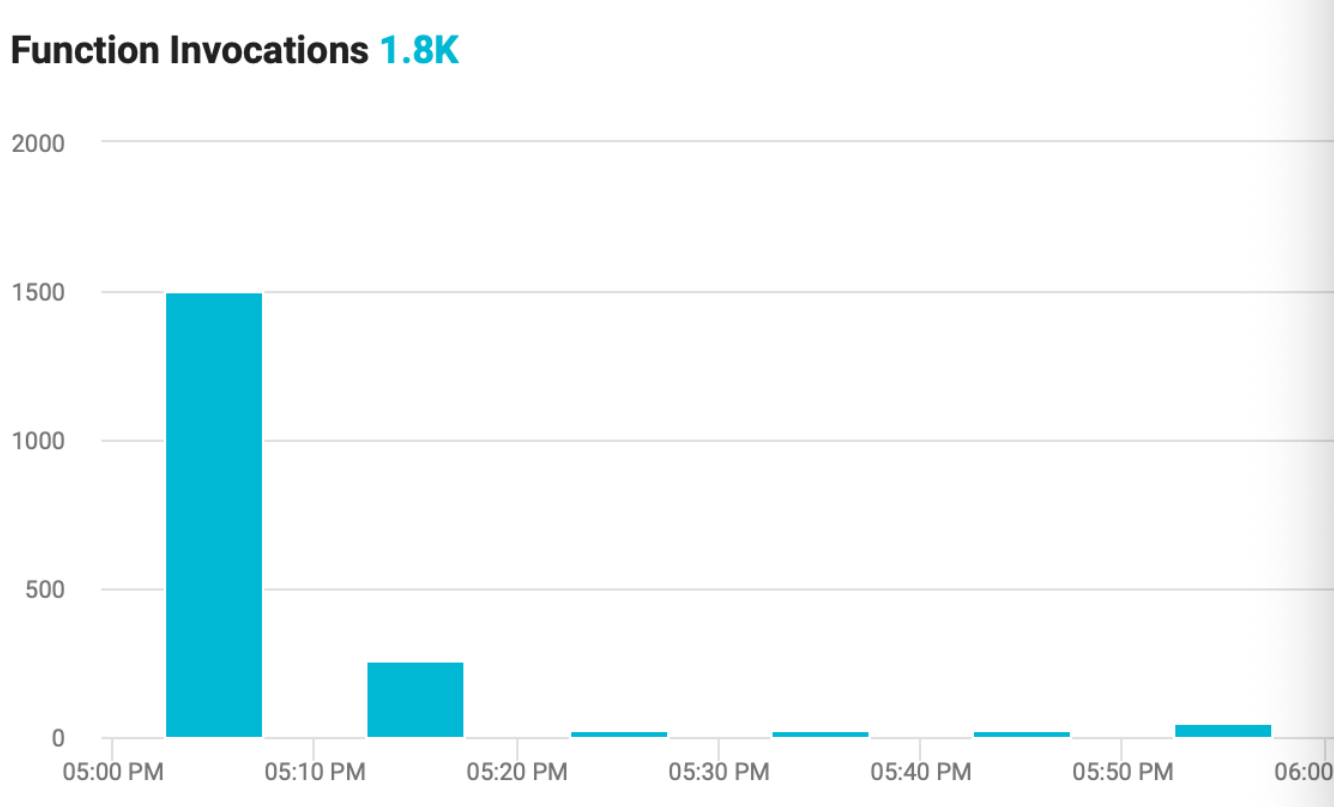
Help

Dark Mode

Demo

Select Lumigo Tag

Function Invocations 1.8K



Most Failed Functions

Function Name	Lumigo ...	% Failur...	# Failures	
pete-dev-fat-fire-app-cor-... us-east-1	Not Set	4.12%	8	
pete-dev-fat-fire-app-int-B... us-east-1	Not Set	100%	1	
pete-dev-fat-fire-app-his-... us-east-1	Not Set	100%	1	

Transactions

Last Hour31 Aug, 05:00pm to 31 Aug, 06:10pmfatfire-pete-dev

Search by entry pointTransactions with issues only

Transactions	Result	Start Time	Duration	Cost	Issues
pete-dev-fat-fire-app-cor-ApiLambdaGETportfolioC4A-6sh6fu... us-east-1	Success	5:55:43 PM	1,886 ms	< 1¢	
pete-dev-fat-fire-app-cor-ApiLambdaGETportfolioC4A-6sh6fu... us-east-1	Success	5:54:38 PM	937 ms	< 1¢	
pete-dev-fat-fire-app-cor-ApiLambdaGETportfolioC4A-6sh6fu... us-east-1	Success	5:54:12 PM	2,568 ms	< 1¢	
pete-dev-fat-fire-app-his-ProcessUsersTimezoneC1DD-NAujh... us-east-1	Failure	5:51:13 PM	1,993 ms	< 1¢	Runtime.UnhandledPromiseRejection
pete-dev-fat-fire-app-int-BasiqTokenRotation825450-QGbWW... us-east-1	Failure	5:30:17 PM	2,199 ms	< 1¢	Runtime.UnhandledPromiseRejection
pete-dev-fat-fire-app-cor-UserAuthorizerFn8DFD7EFB-5Q9E8... us-east-1	Failure	5:06:40 PM	4 ms	< 1¢	Error
pete-dev-fat-fire-app-cor-UserAuthorizerFn8DFD7EFB-5Q9E8... us-east-1	Failure	5:06:40 PM	1 ms	< 1¢	Error
pete-dev-fat-fire-app-cor-UserAuthorizerFn8DFD7EFB-5Q9E8... us-east-1	Failure	5:06:40 PM	3 ms	< 1¢	Error
pete-dev-fat-fire-app-cor-UserAuthorizerFn8DFD7EFB-5Q9E8... us-east-1	Failure	5:06:40 PM	1 ms	< 1¢	Error

1 to 14 of 14Page 1 of 1

### Most Invoked Functions

Function Name	Lumigo Tag	# Invocations	
pete-dev-fat-fire-app-debug-stac... us-east-1	Not Set	772	
pete-dev-fat-fire-app-debug-sta-D... us-east-1	Not Set	507	
pete-dev-fat-fire-app-cor-UserAut... us-east-1	Not Set	194	
pete-dev-fat-fire-app-int-Lambda... us-east-1	Not Set	172	
pete-dev-fat-fire-app-debug-stac... us-east-1	Not Set	157	

# Fatfire Architecture



Core Service



Exchange Service



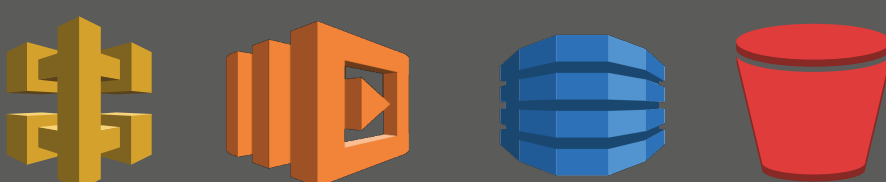
History Service



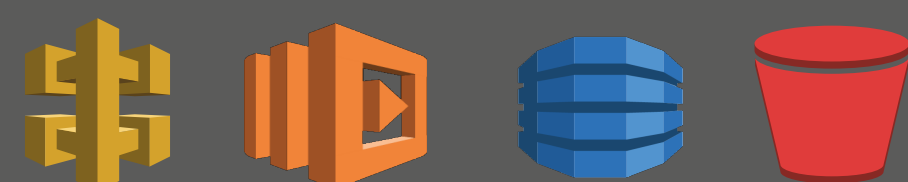
Cleanup Service



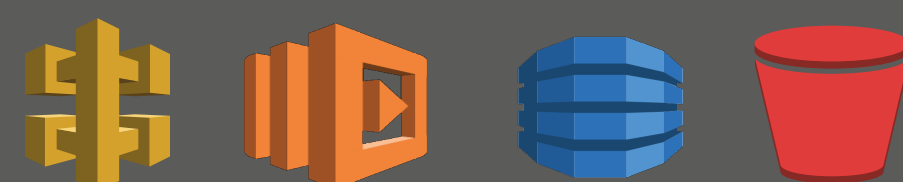
Analytics Service



Integration Service



Insurance Service



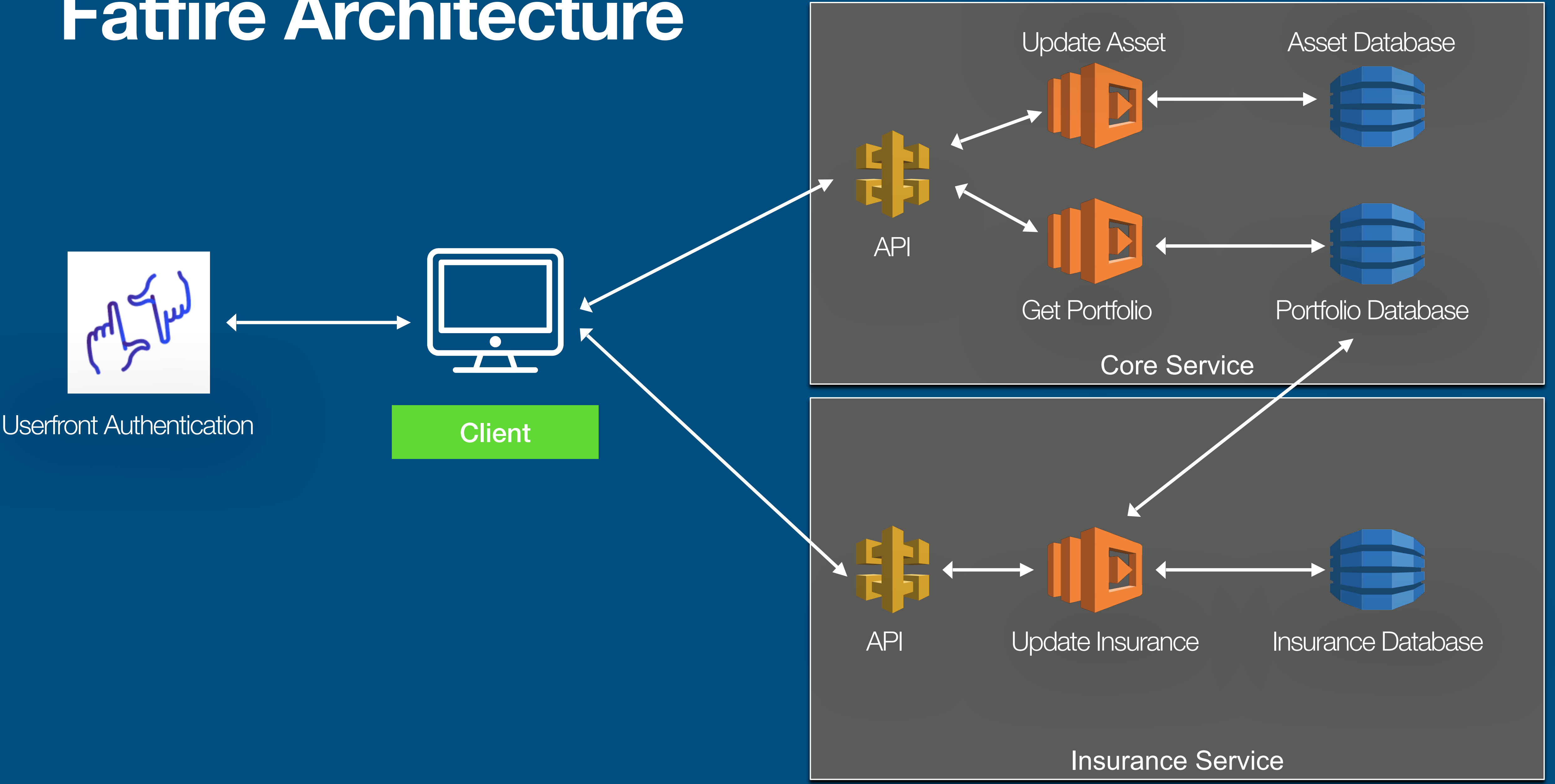
Receipts Service

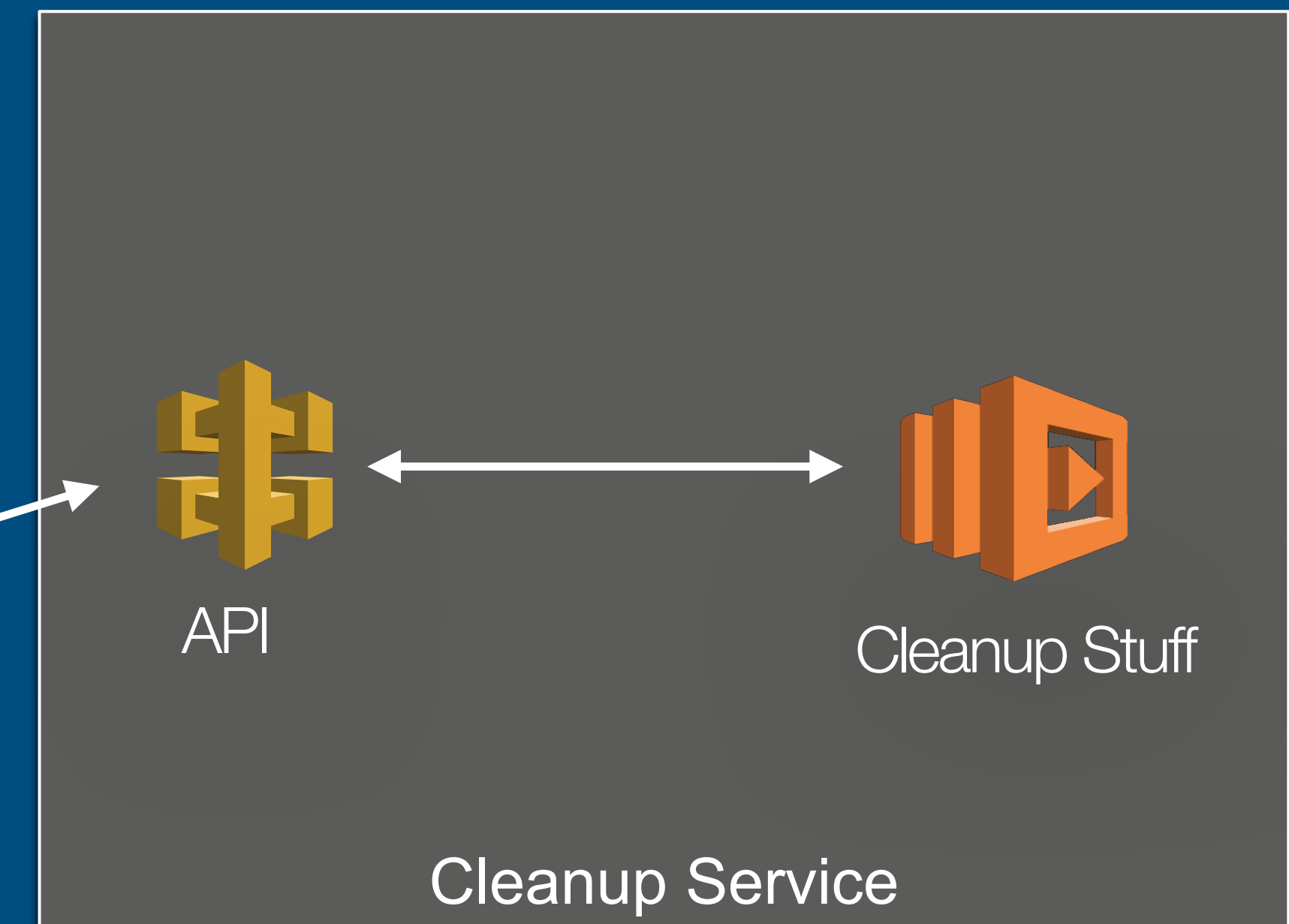
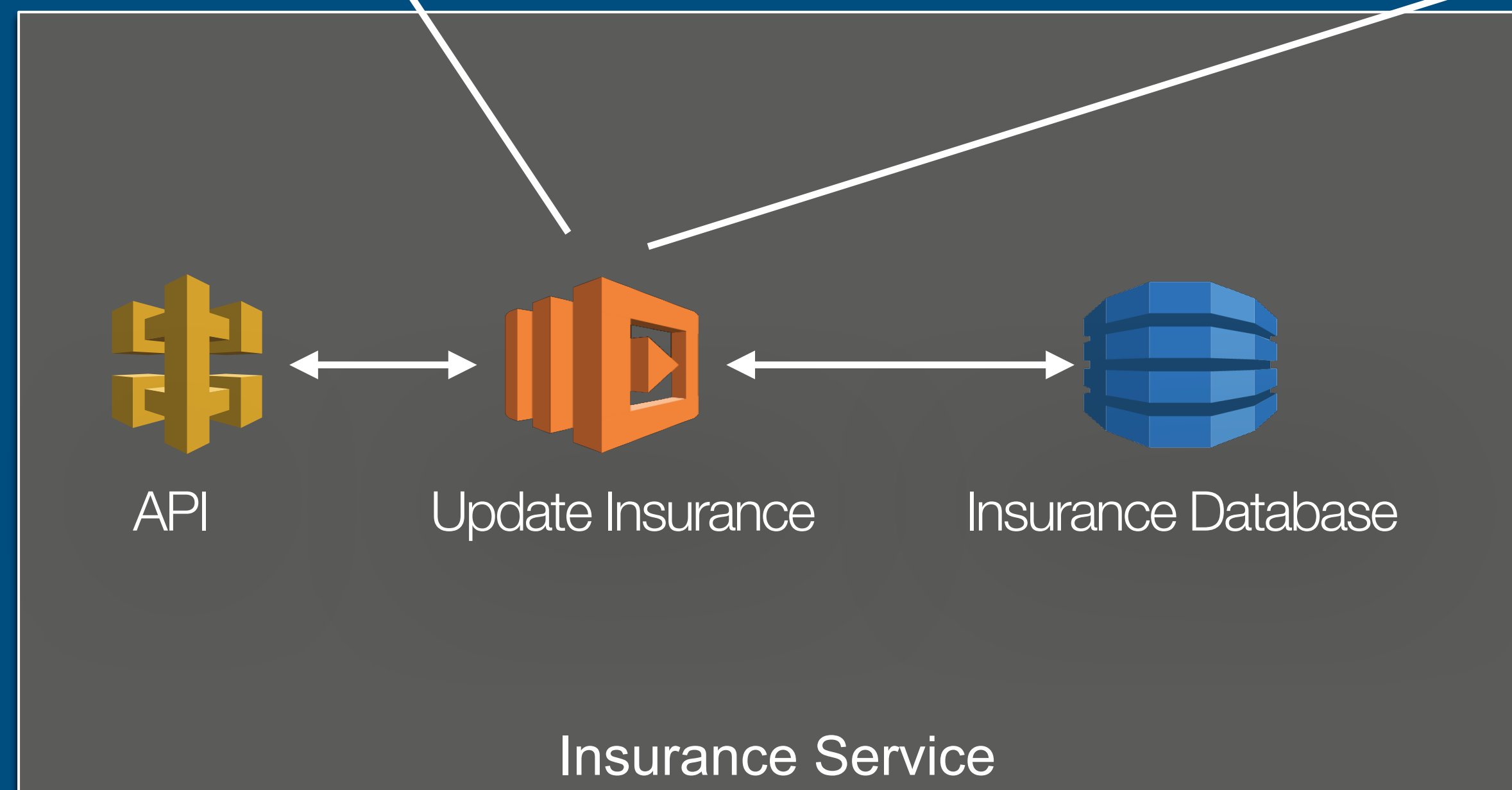
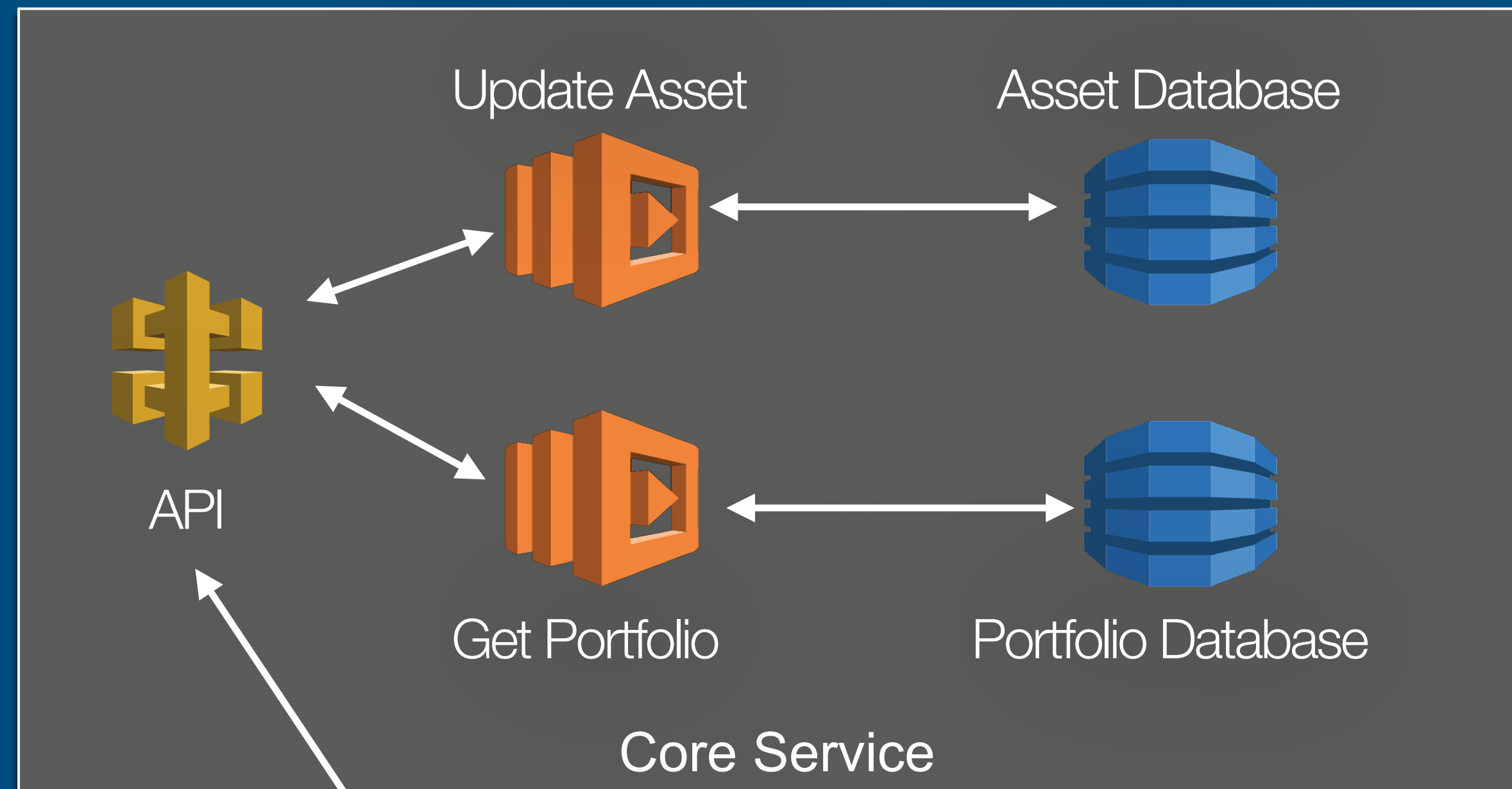


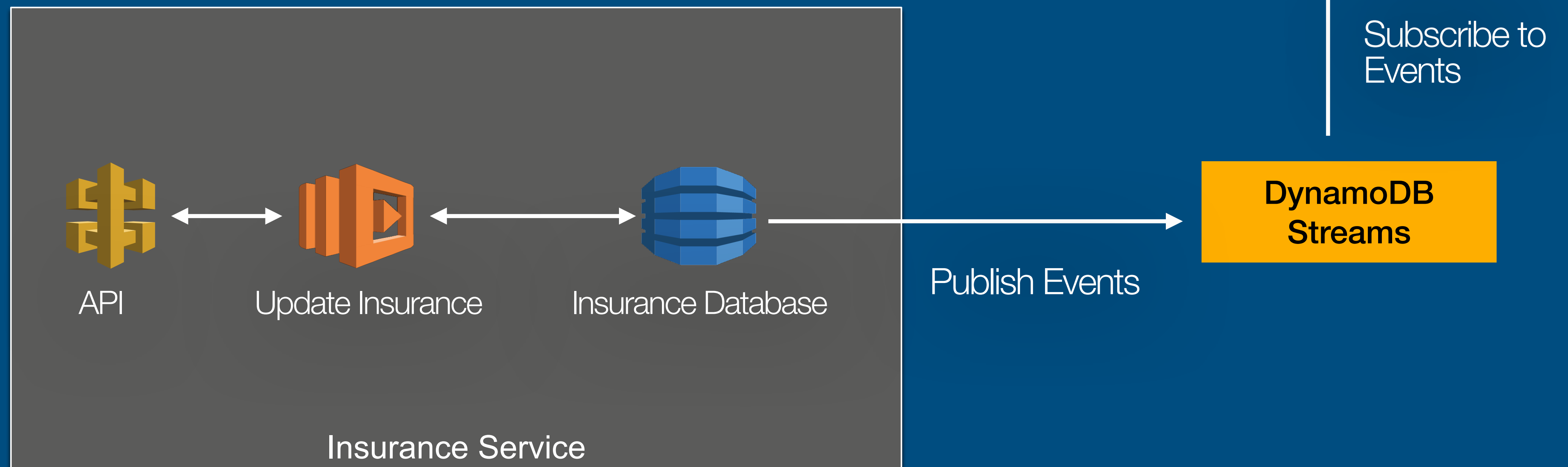
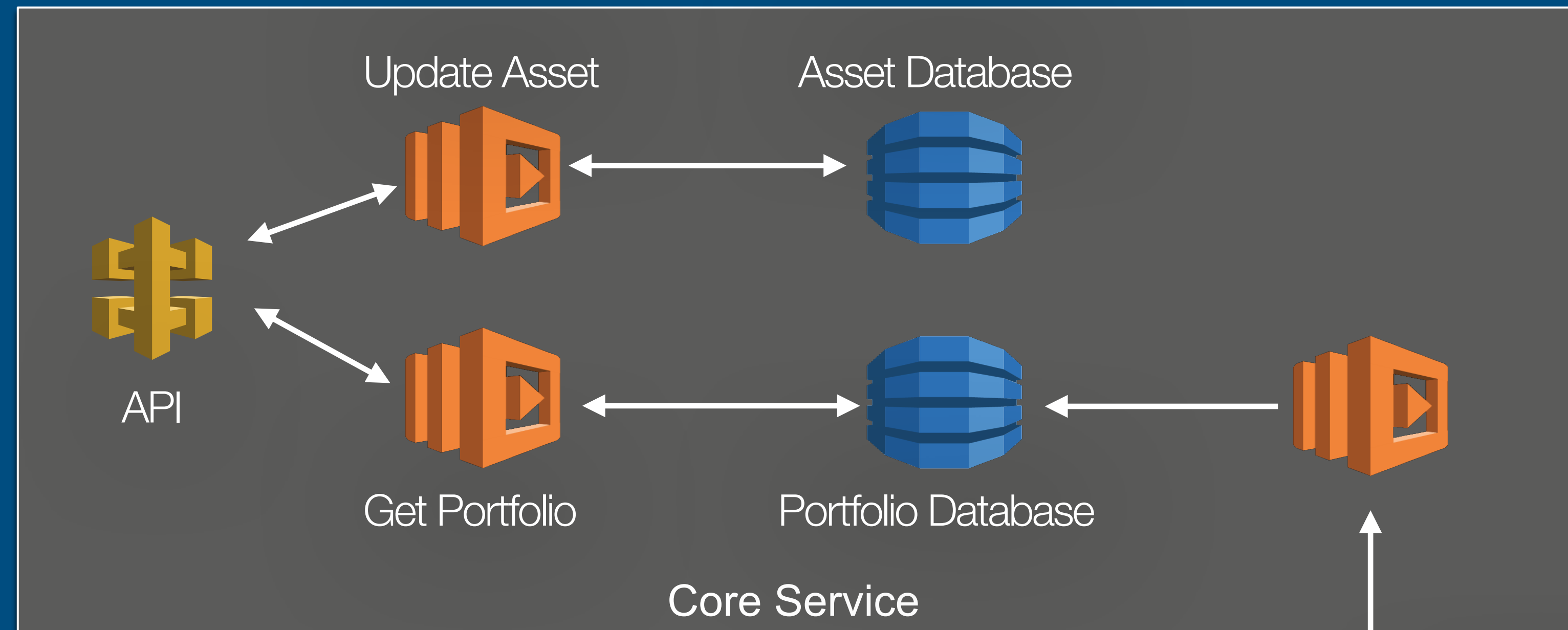
Tax Service



# Fatfire Architecture

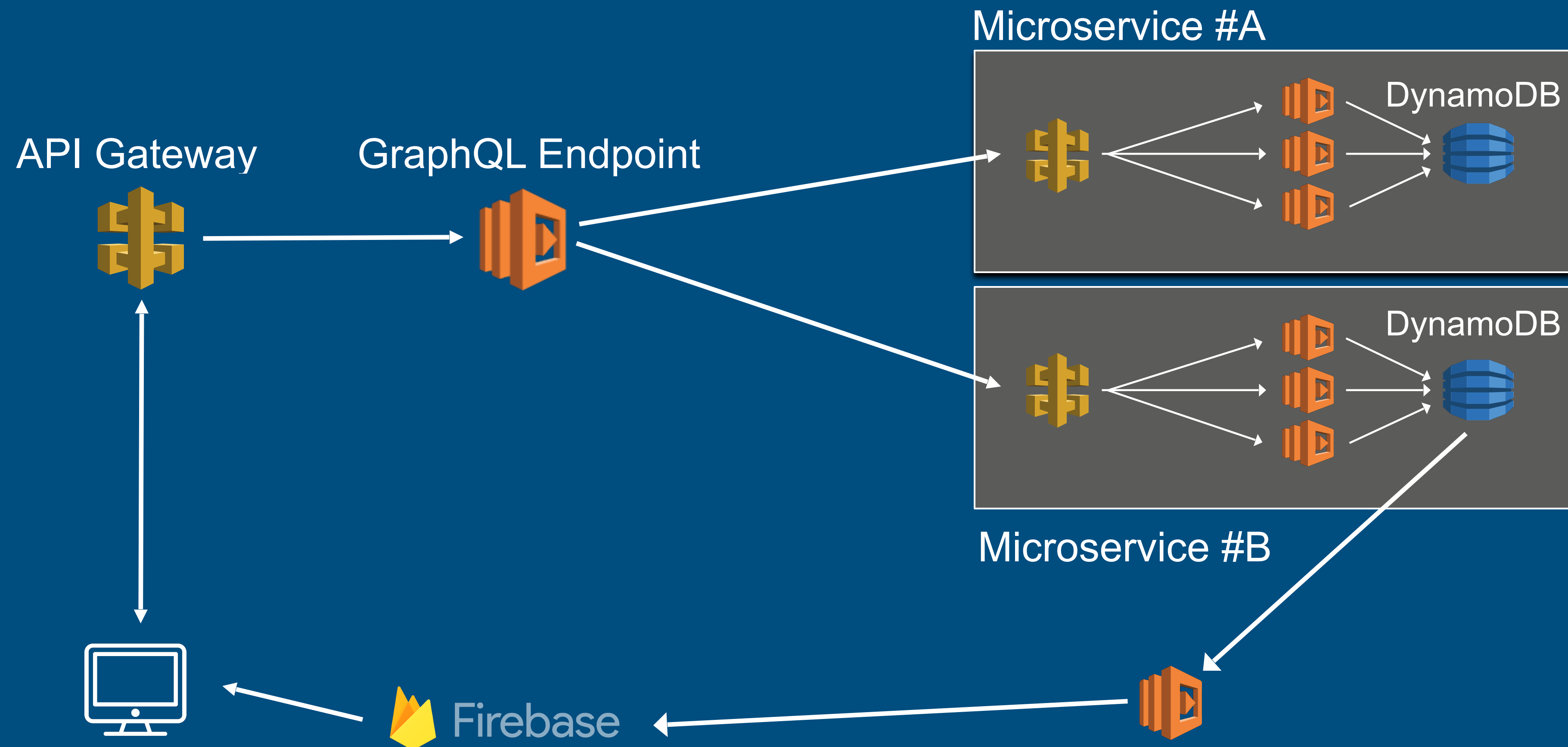


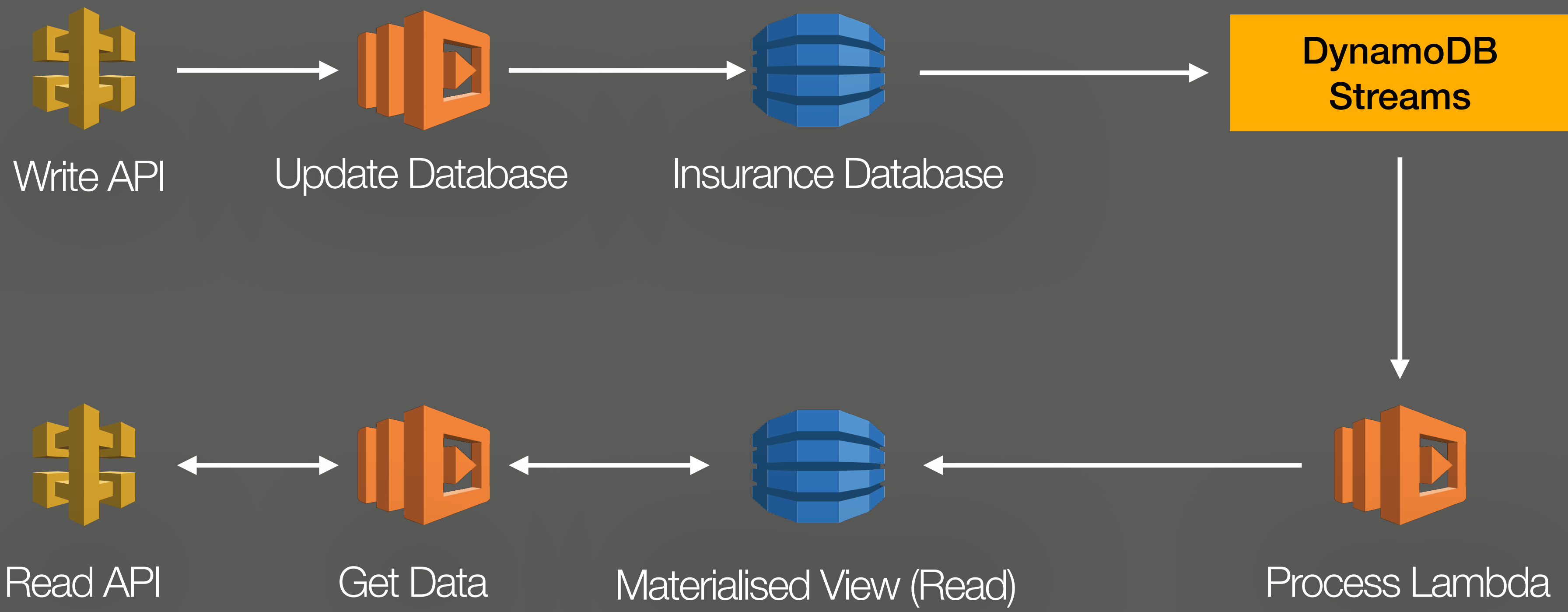






# Remember this architecture?

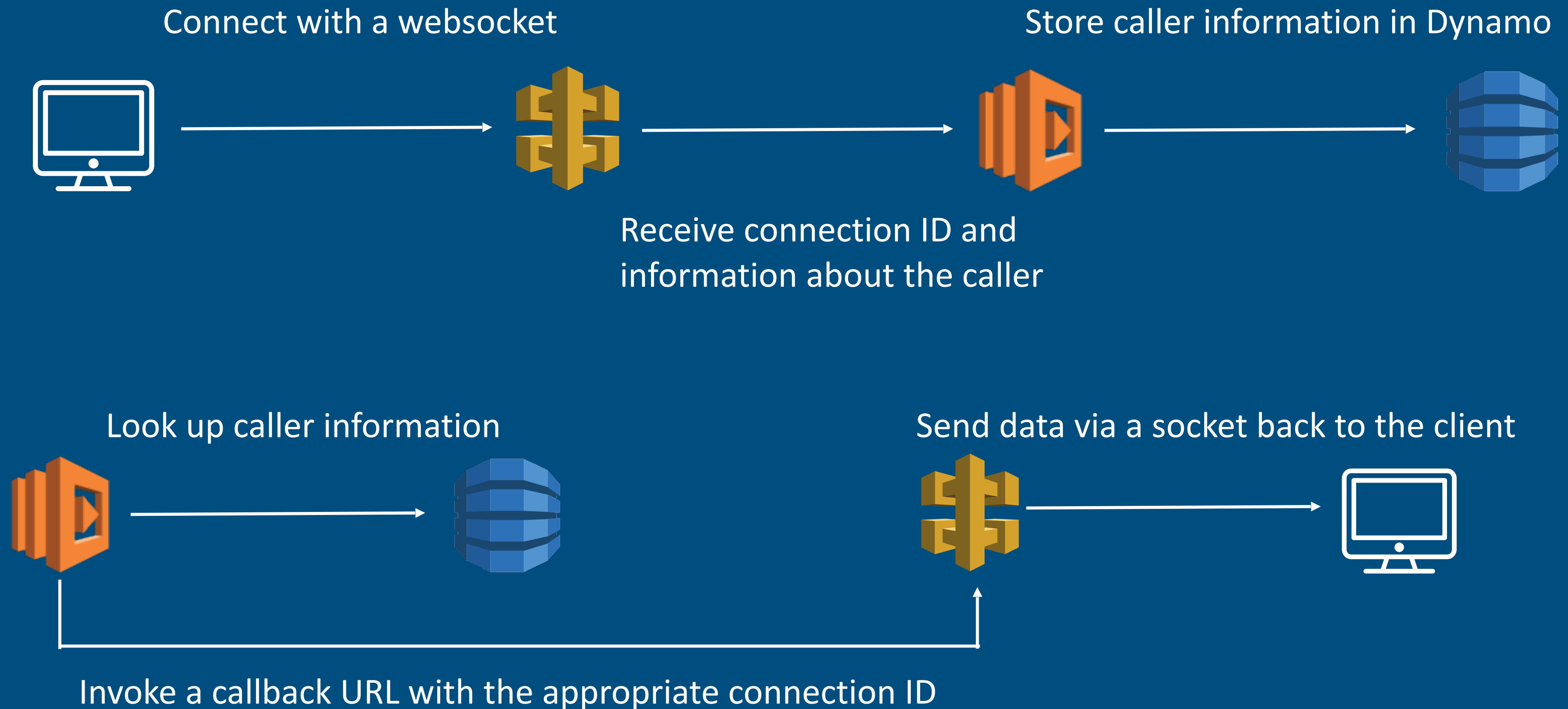


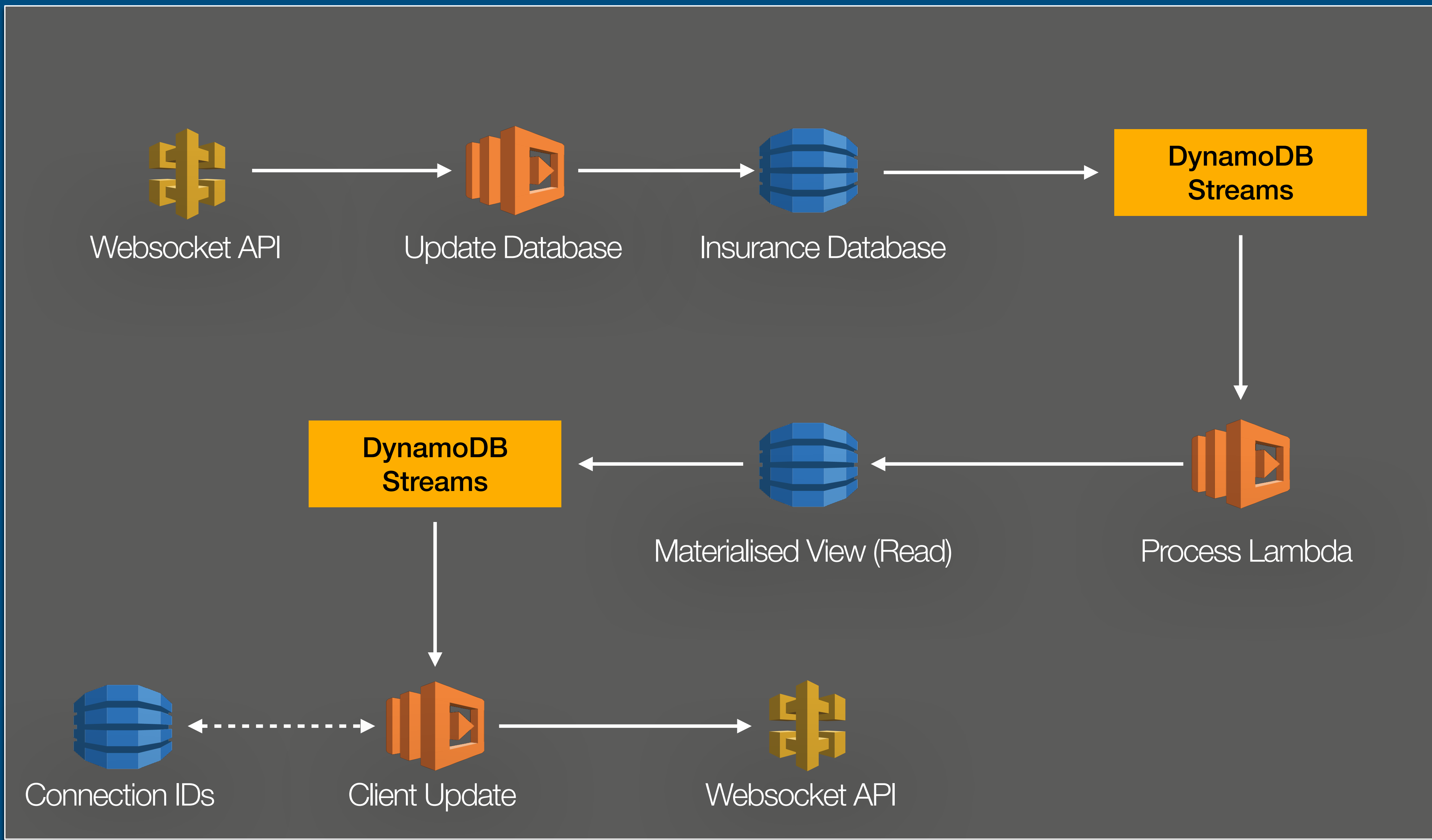




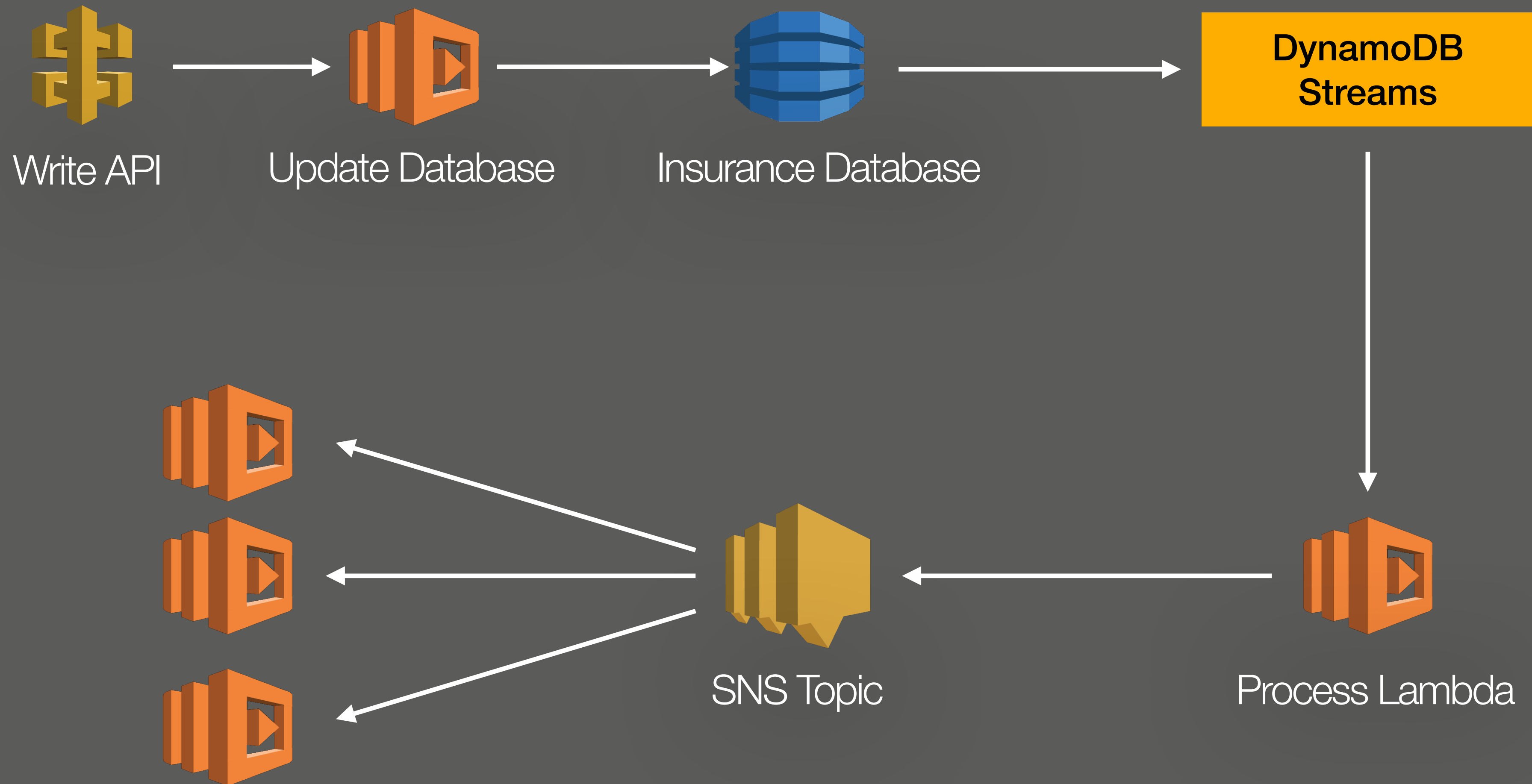
# Web Sockets

## With API Gateway







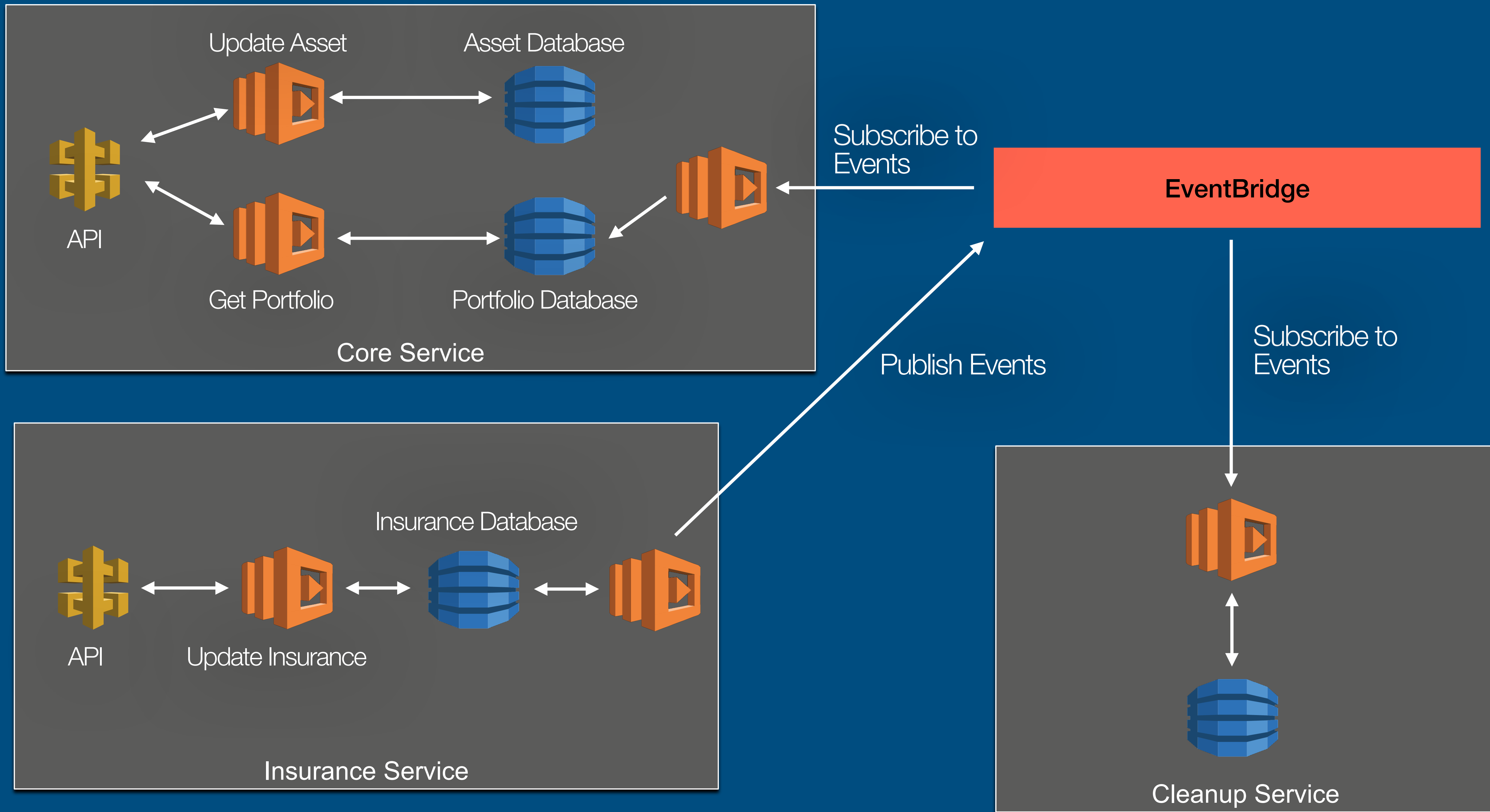


# Mullet Architecture

With thanks to Tim Wagner

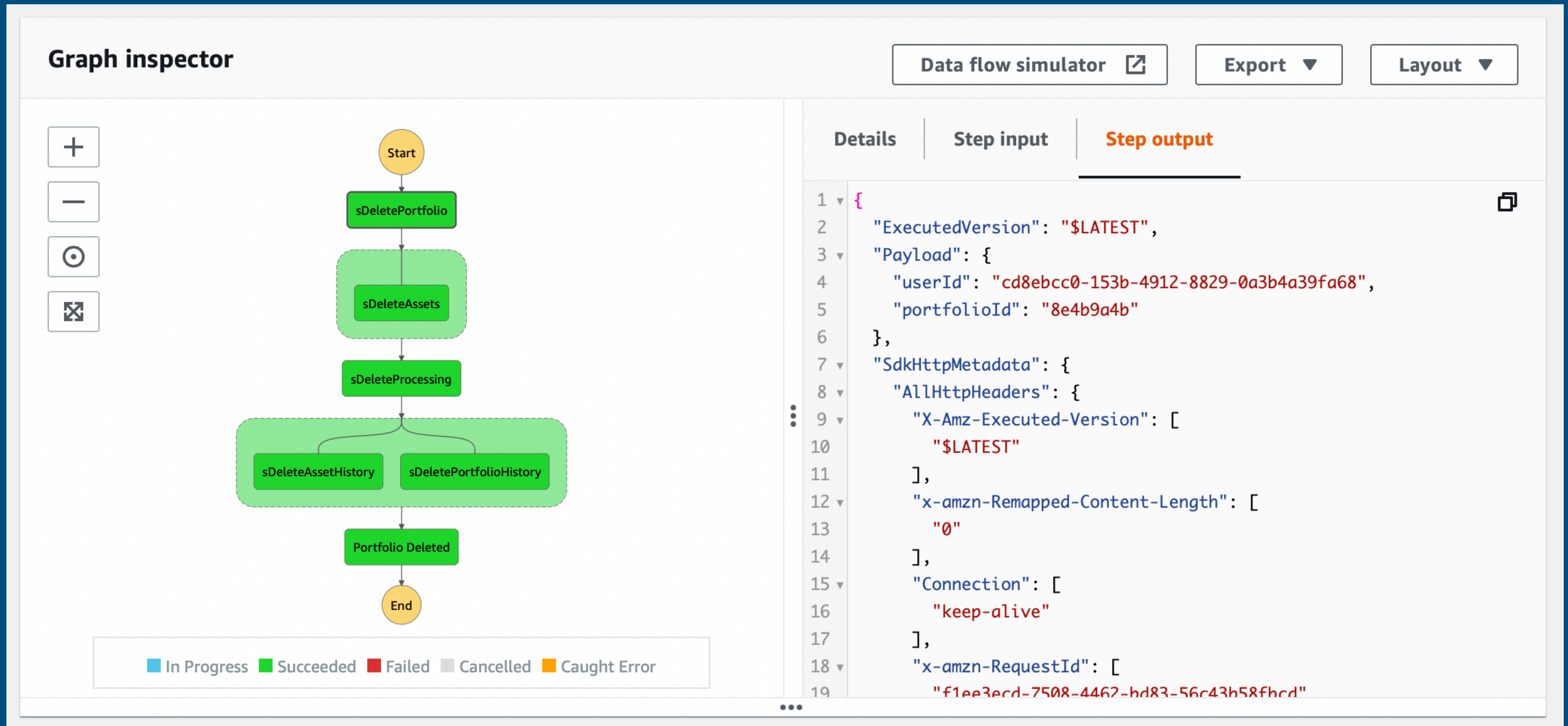






# Step Functions

## Coordinate Distributed Applications





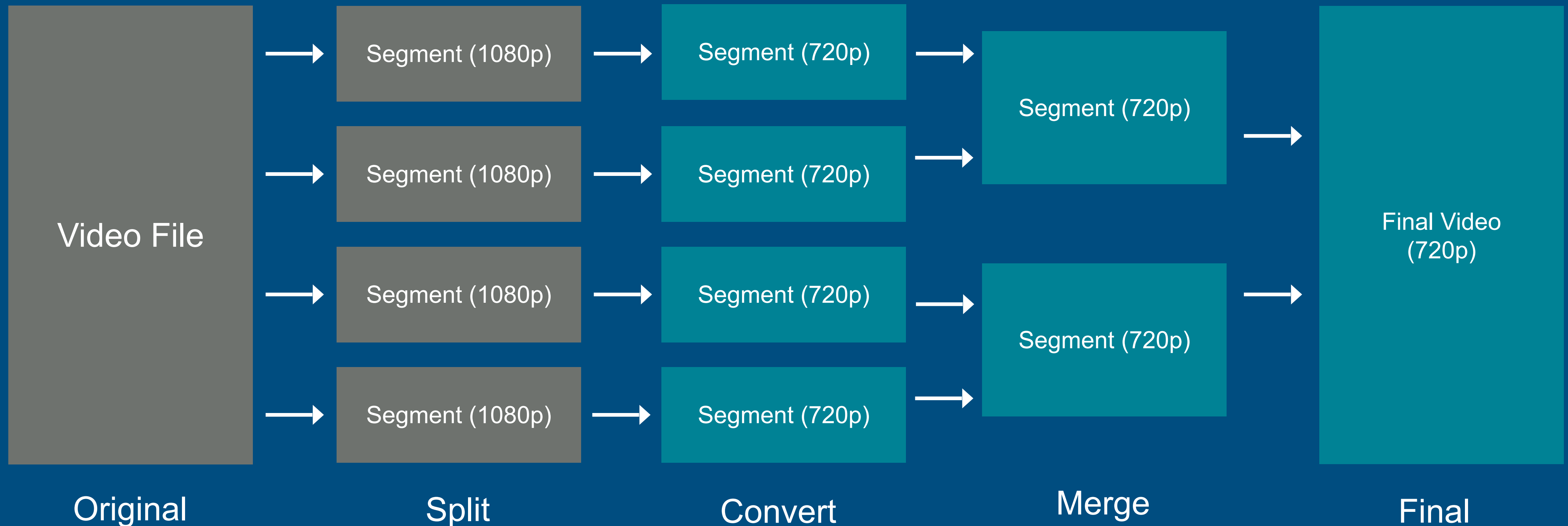
# Parallel Architecture

## Dealing with unexpected problems

- Take a complex problem and solve it with Lambda by applying techniques like MapReduce & Parallelisation
- Can you transcode (i.e. encode) a large video file with a Serverless-only approach?

# Divide and conquer

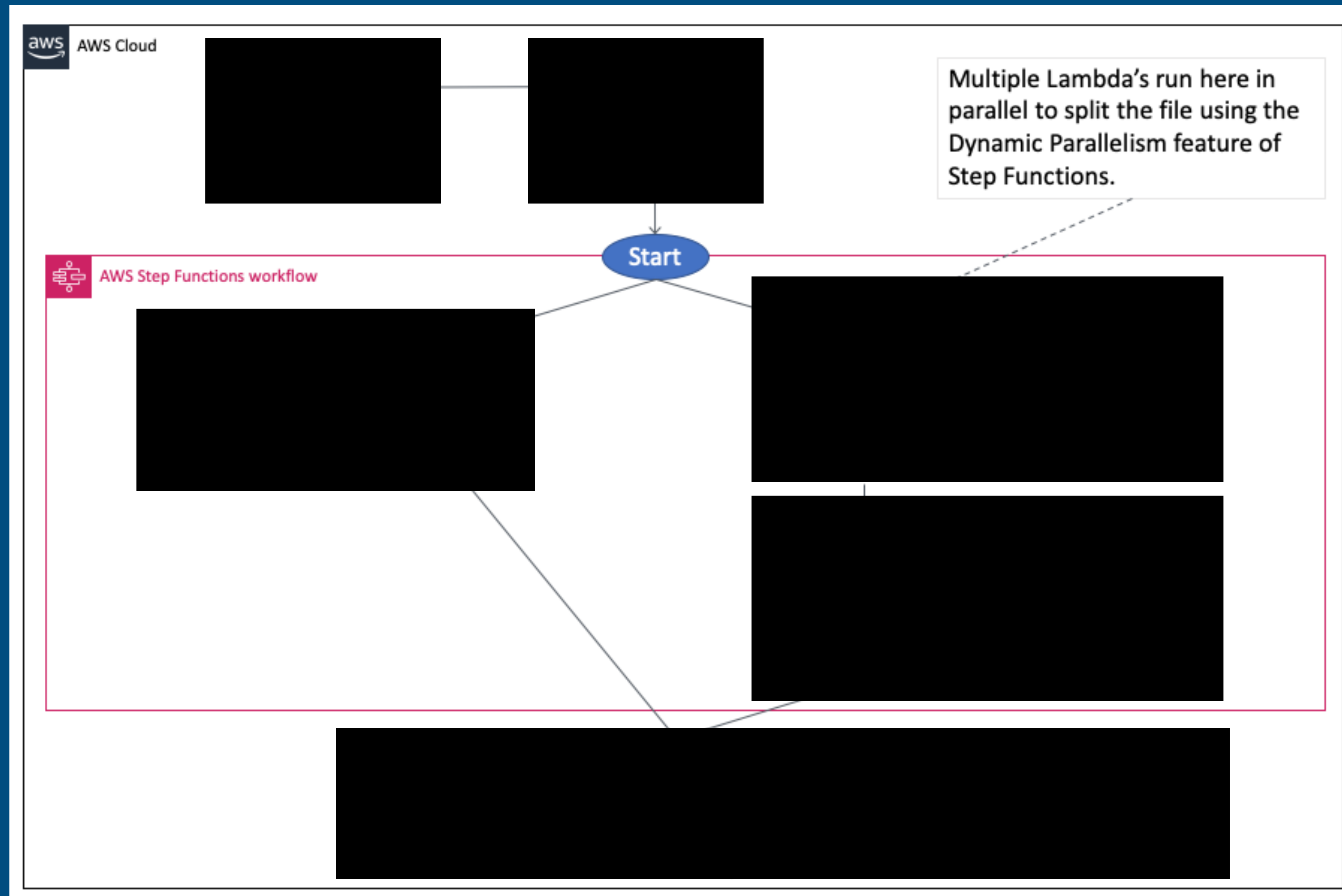
## Using the Lambda supercomputer



Read more: <https://bit.ly/3wJOdvQ>

# Parallel Computing with Lambda & Step Functions

Read more: <https://bit.ly/3wJOdvQ>





# Serverless Video Transcoder

## Parallel and conquer

	Serverless Lambda	Traditional EC2 (t2.large)	MacBook Pro 16GB 3.5GHz i7
34MB MP4 (00:43, 1920×1080)	11 seconds	32 seconds	18 seconds
77MB MP4 (6:49, 2048×1152)	26 seconds	144 seconds	78 seconds
100MB MP4 (59:56, 1280×720)	86 seconds	1073 seconds	592 seconds
350MB MP4 (07:45, 2560×1440)	35 seconds	432 seconds	224 seconds
420MB MKV (01:02, 3840 x 1606)	112 seconds	157 seconds	101 seconds
1GB MKV (57:57, 1280 x 718)	185 seconds	4320 seconds	2367 seconds

Read more: <https://bit.ly/3wJOdvQ>

# Common Complaints

Why can't things just be easy

- Hard to dev locally ✓
- Hard to debug ✓
- Hard to observe and monitor ✓
- Hard or impossible to do certain things (e.g. long-running tasks) ✓
- Lock-in is a problem ?

# Modern Applications

## Some lessons were learnt

- Security/compliance first
- Use microservices
- Serverless where possible
- CI/CD
- Monitor, monitor, monitor!
- <https://youtu.be/IPOvrK3S3gQ>
- Serverless monoliths can be OK!
- Automation is a must
- Think through your testing strategy
- Experimentation and architectural changes are easier
- Serverless (& services) > containers

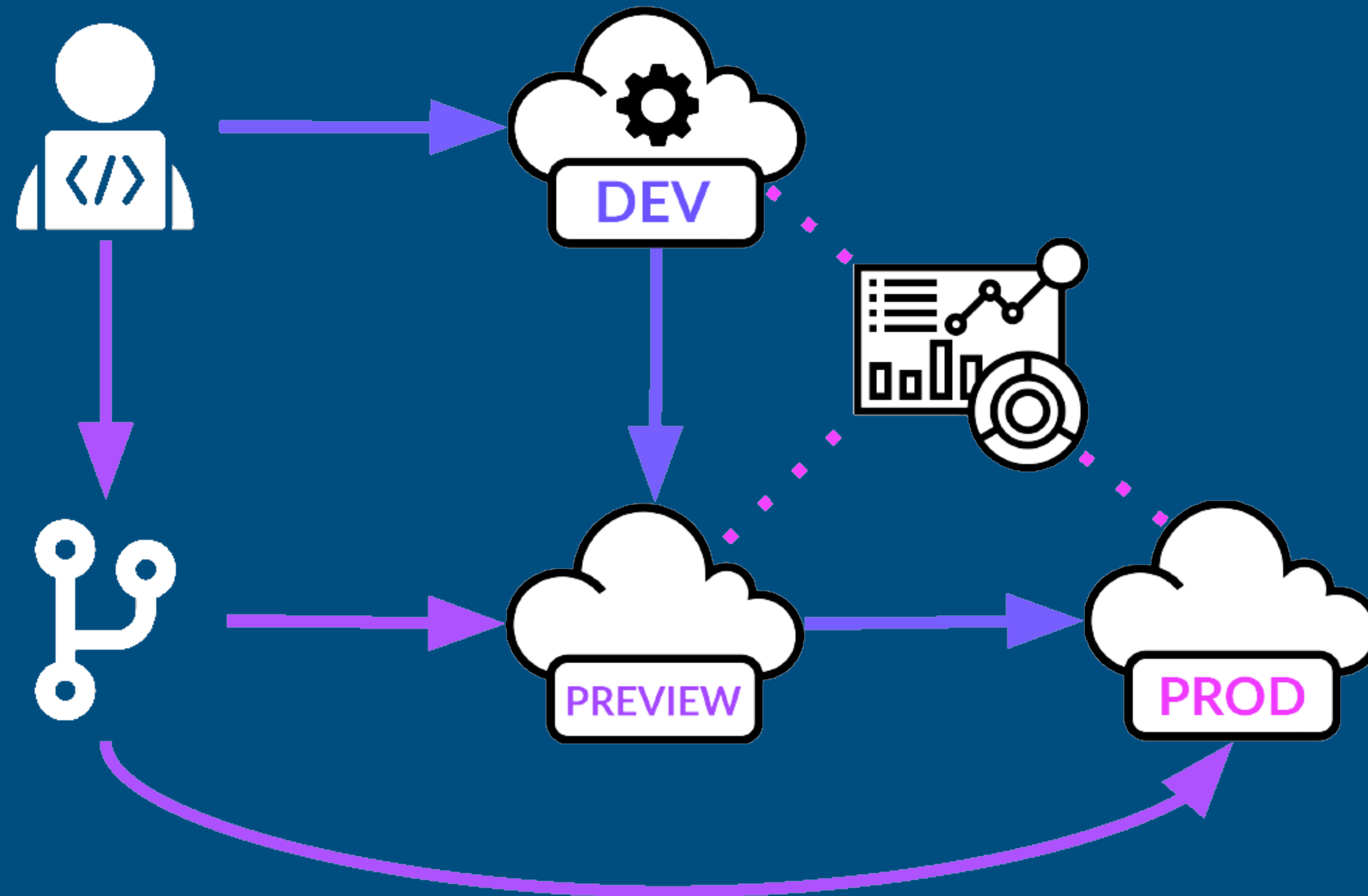


# Ampt

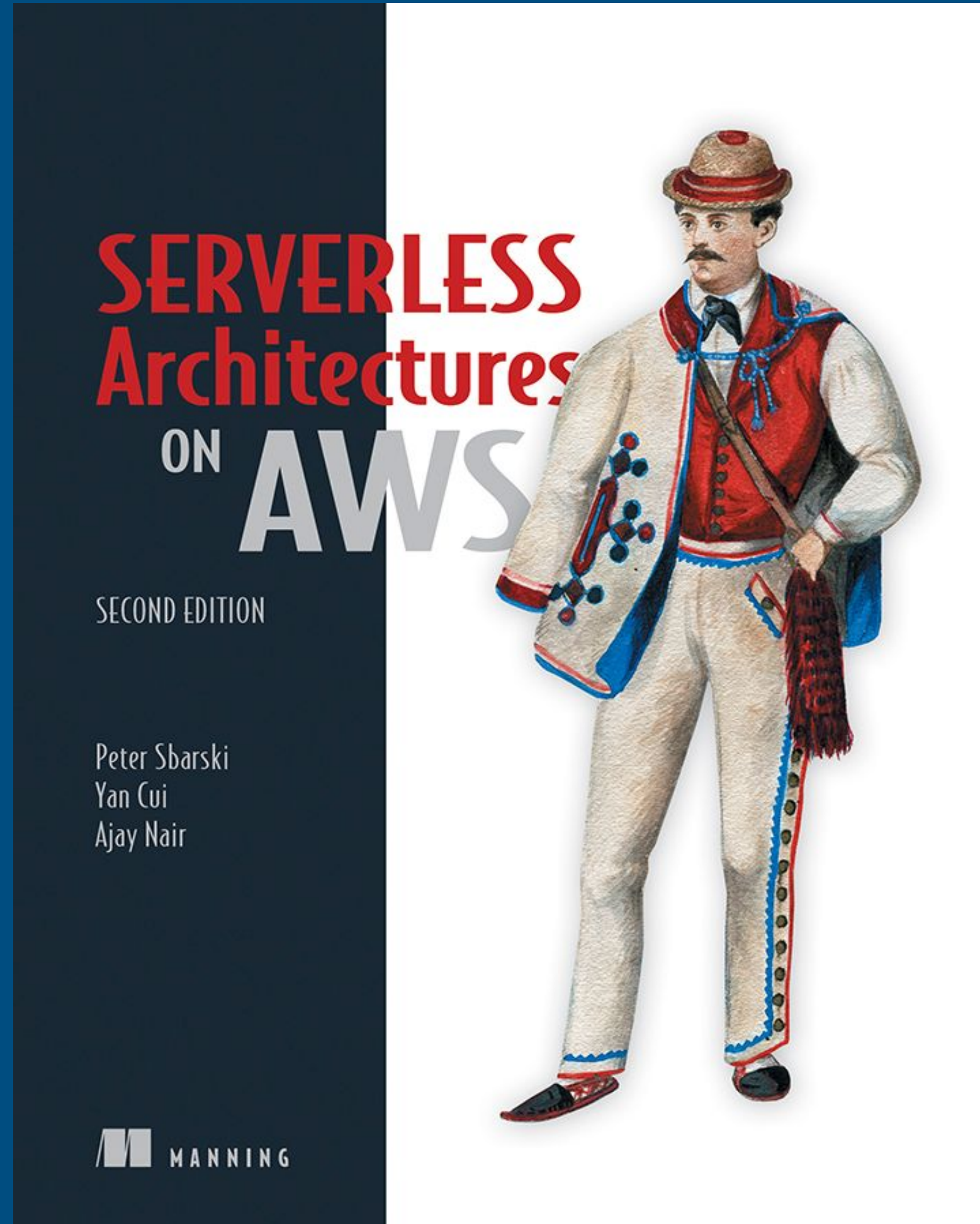
<https://getampt.com>



# ampt



# Thank you



**Fatfire:** <https://fatfireapp.com>

**Ampt:** <https://getampt.com>

**Serverless Architectures on AWS:**  
<https://www.manning.com/books/serverless-architectures-on-aws-second-edition>

**The Value Flywheel Effect:**  
<https://itrevolution.com/product/the-value-flywheel-effect/>