AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

ARC302

Evolutionary AWS Lambda functions with hexagonal architecture

Luca Mezzalira (he/him)
Principal Solutions Architect
AWS

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How many of you find it challenging to evolve an application in the cloud?

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Luca Mezzalira

AWS Principal Solutions Architect

International speaker

O'Reilly author

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Problem space

The anatomy of hexagonal architecture

AWS Lambda with hexagonal architecture

Additional use cases

Wrap-up

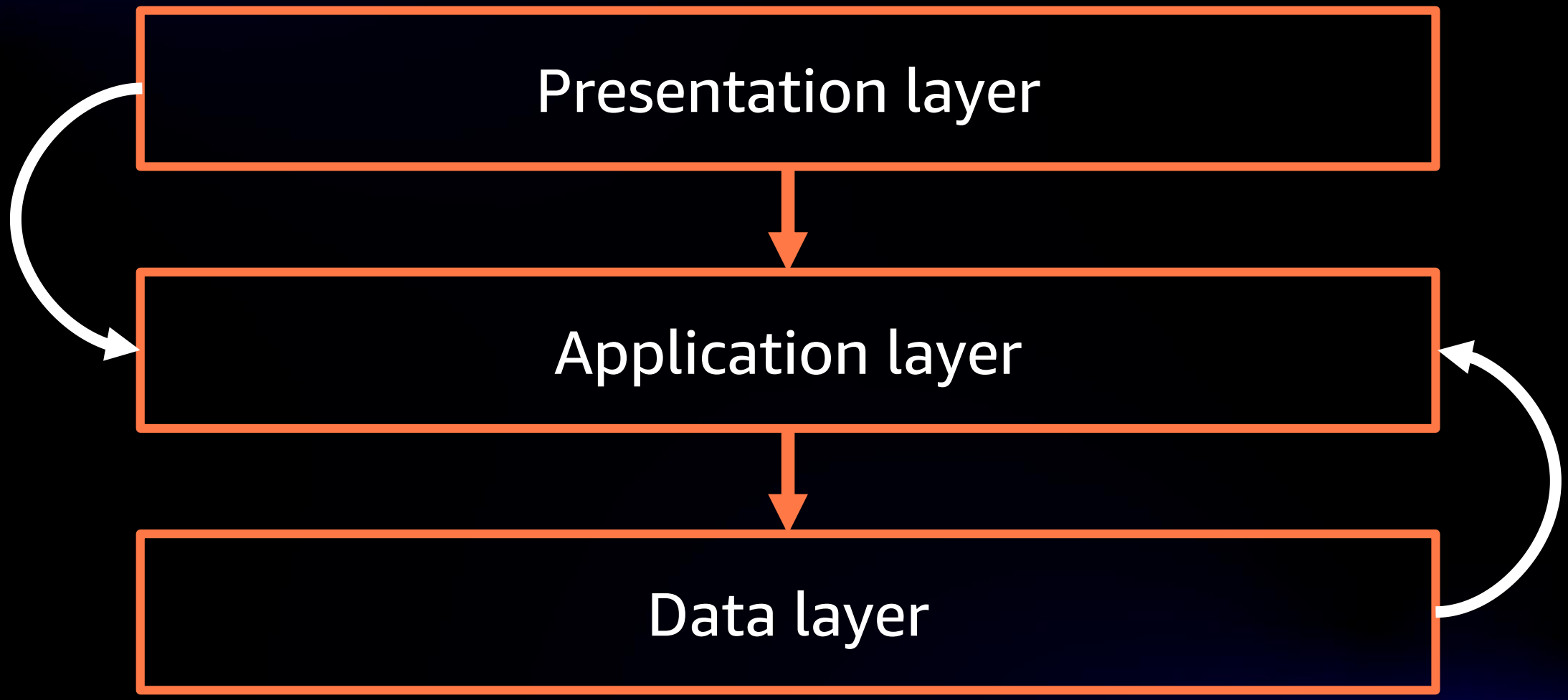
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

"Almost from day one, we knew that the software we were building would not be the software that would be running a year later."

Dr. Werner Vogels
CTO, Amazon

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Three-tier software architecture



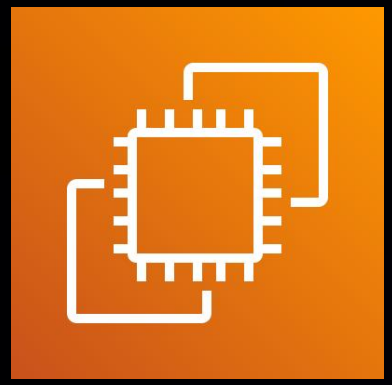
aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Moving to AWS Lambda



Amazon ECS



Amazon EC2





AWS Lambda

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lack of boundaries

```
//import libraries...

module.exports.submit = (event, context, callback) => {
  const requestBody = JSON.parse(event.body);
  const fullname = requestBody.fullname;
  const email = requestBody.email;
  const experience = requestBody.experience;

  // request body validations...

  submitCandidateP(candidateInfo(fullname, email, experience))
    .then(res => {
      callback(null, {
        statusCode: 200,
        body: JSON.stringify({
          message: `Sucessfully submitted candidate with email ${email}`,
          candidateId: res.id
        })
      });
    })
    .catch(err => {
      // error handling...
    });
};
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

"Hexagonal architecture allows an application to equally be driven by users, programs, automated tests, or batch scripts, and to be developed and tested in isolation. . ."

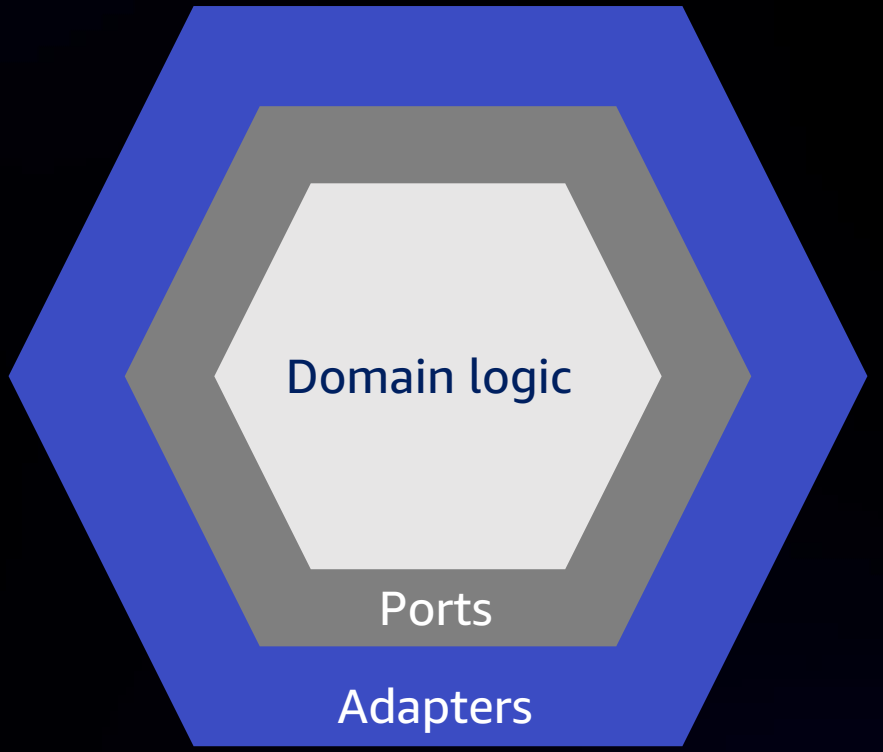
Dr. Alistair Cockburn

Hexagonal architecture creator

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Hexagonal architecture anatomy



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Ports

```
const getStockValue = require("../adapters/StocksDB");

const getStockData = async (stockID) => {
    try{
        const data = await getStockValue(stockID);
        return data.Item;
    } catch(err) {
        return err
    }
}

module.exports = {
    getStockData
}
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Ports

```java
package com.aws.articles.domain.ports;

import com.aws.articles.domain.model.Article;

public interface SocialMediaPublisher {

    void publish(Article article);

}
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Adapters

```
const documentClient = new AWS.DynamoDB.DocumentClient();

const DB_TABLE = process.env.DB_TABLE;

const getStockValue = async (stockID) => {
    let params = {
        TableName : DB_TABLE,
        Key:{
            'STOCK_ID': stockID
        }
    }

    try {
        const stockData = await documentClient.get(params).promise()
        return stockData
    }
    catch (err) {
        console.log(err)
        return err
    }
}

module.exports = getStockValue;
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Adapters

```
class TwitterPublisher implements SocialMediaPublisher {

    private final TwitterClient twitterClient;

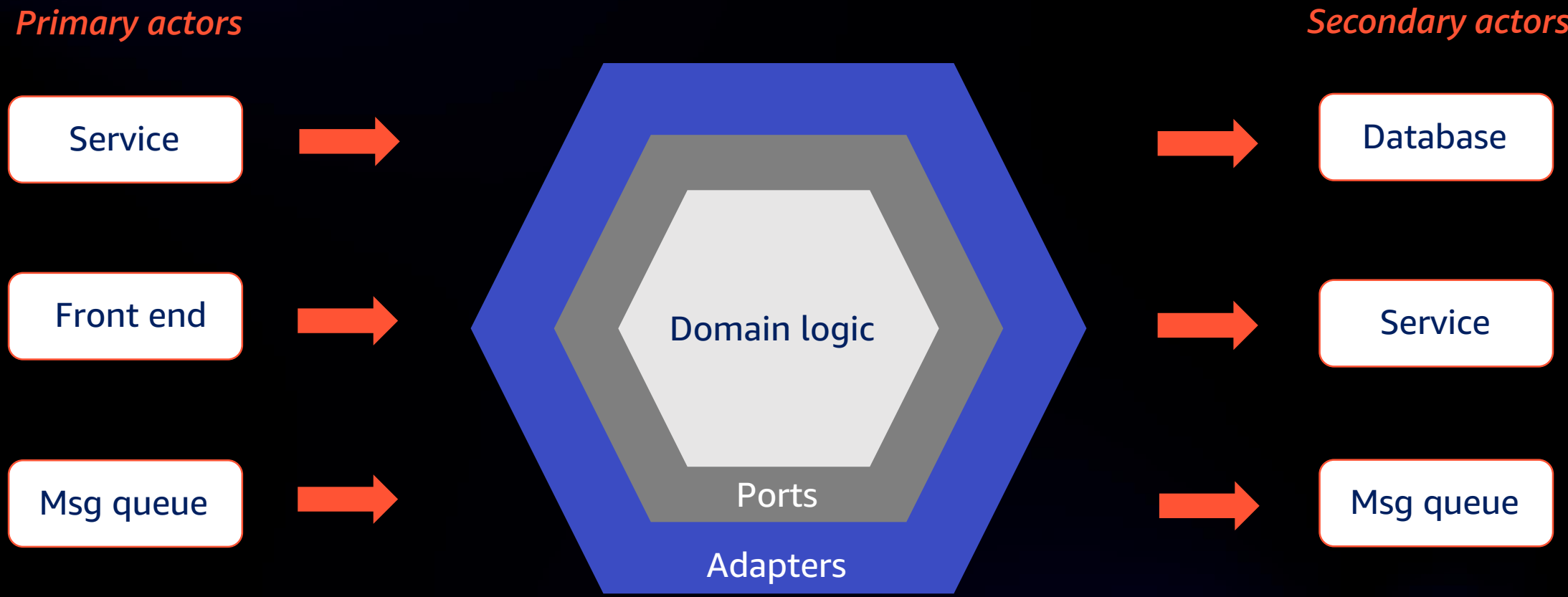
    TwitterPublisher(final TwitterClient twitterClient) {
        this.twitterClient = twitterClient;
    }

    @Override
    public void publish(final Article article) {
        //code to publish a tweet
    }
}
```

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Hexagonal architecture anatomy



aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

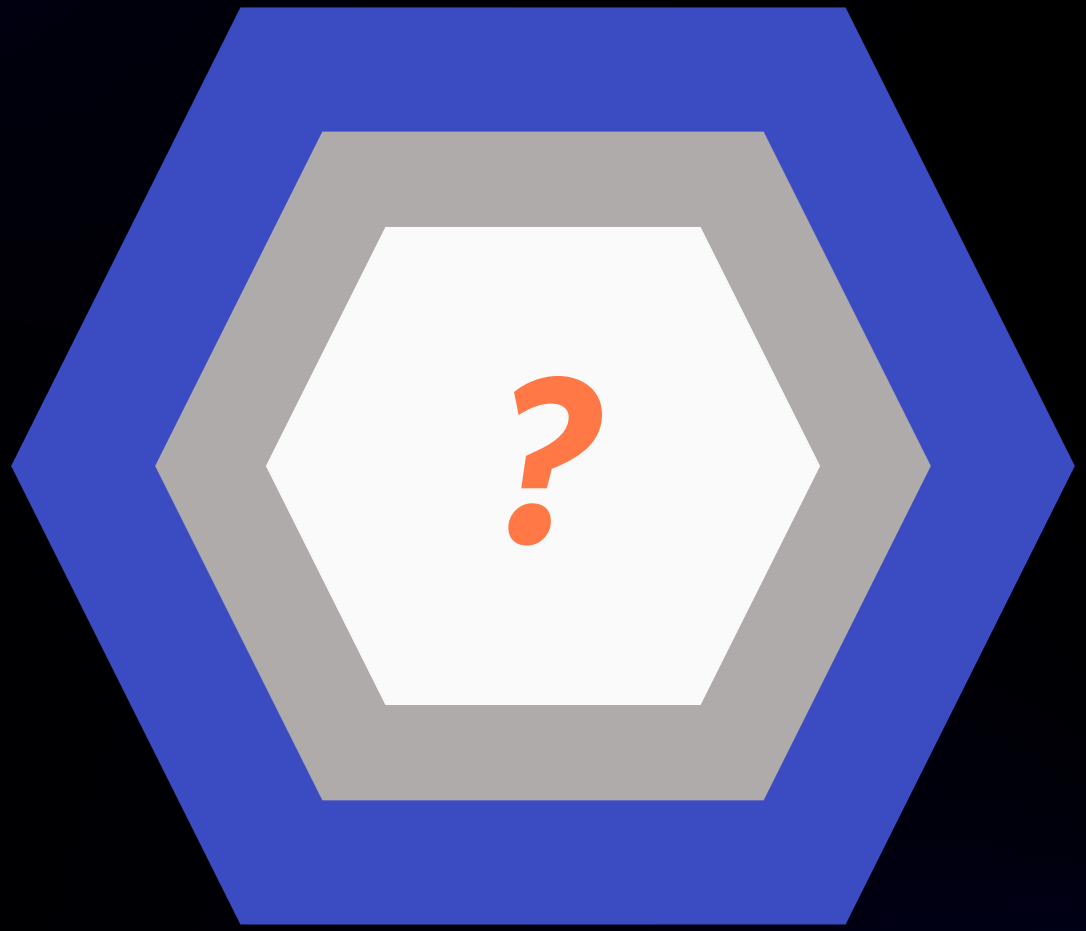
Benefits 

- Independent business logic from the outside world
- Inside-out programming
- Easier to test in isolation
- Low technical debt

Drawbacks 

- Building more layers up front
- Loose implementation details for the business logic

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

"The hexagon … allow[s] the people doing the drawing to have room to insert ports and adapters as they need. . . The term 'hexagonal architecture' comes from this visual effect."

Dr. Alistair Cockburn

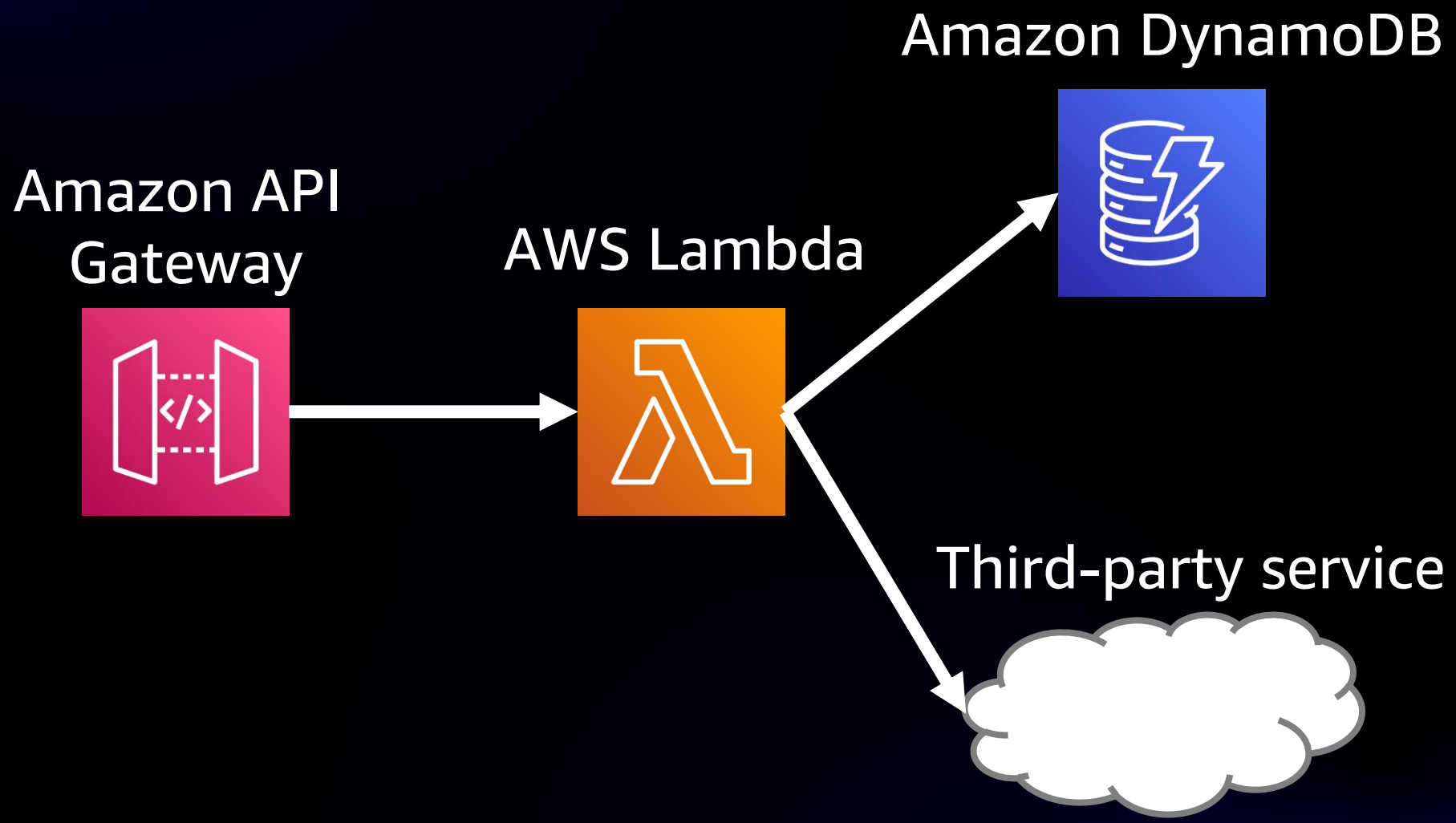
Hexagonal architecture creator

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

It’s demo time!

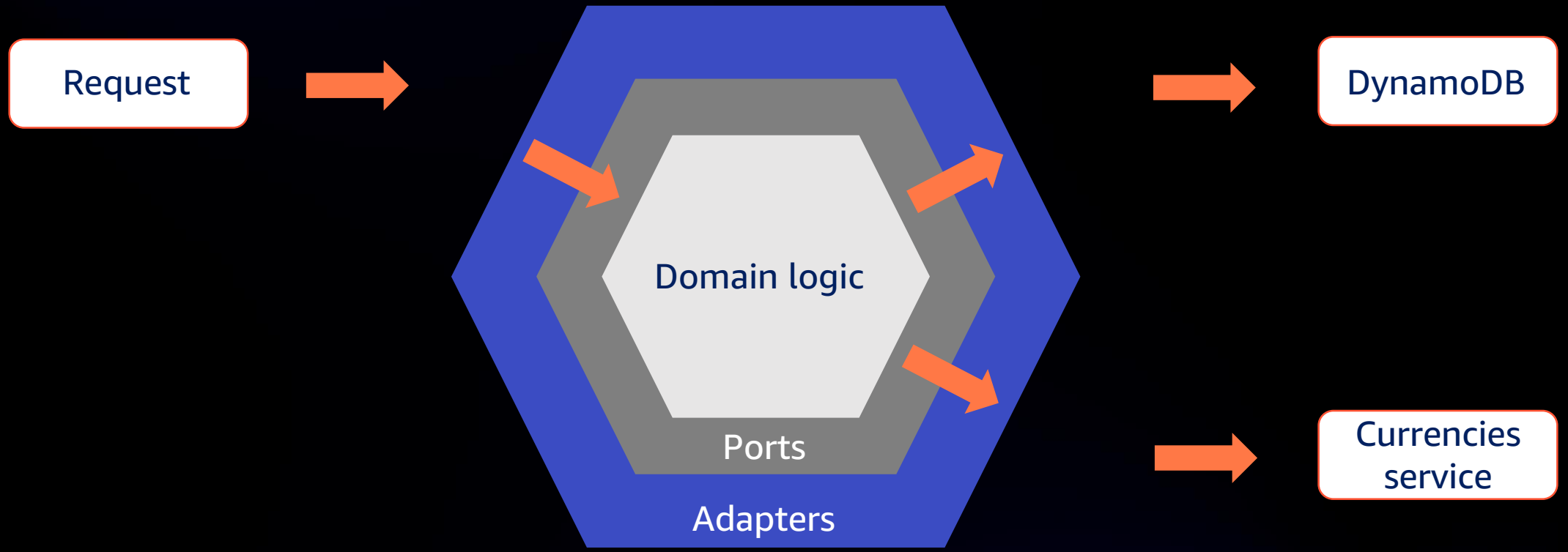
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Stocks API



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The flow



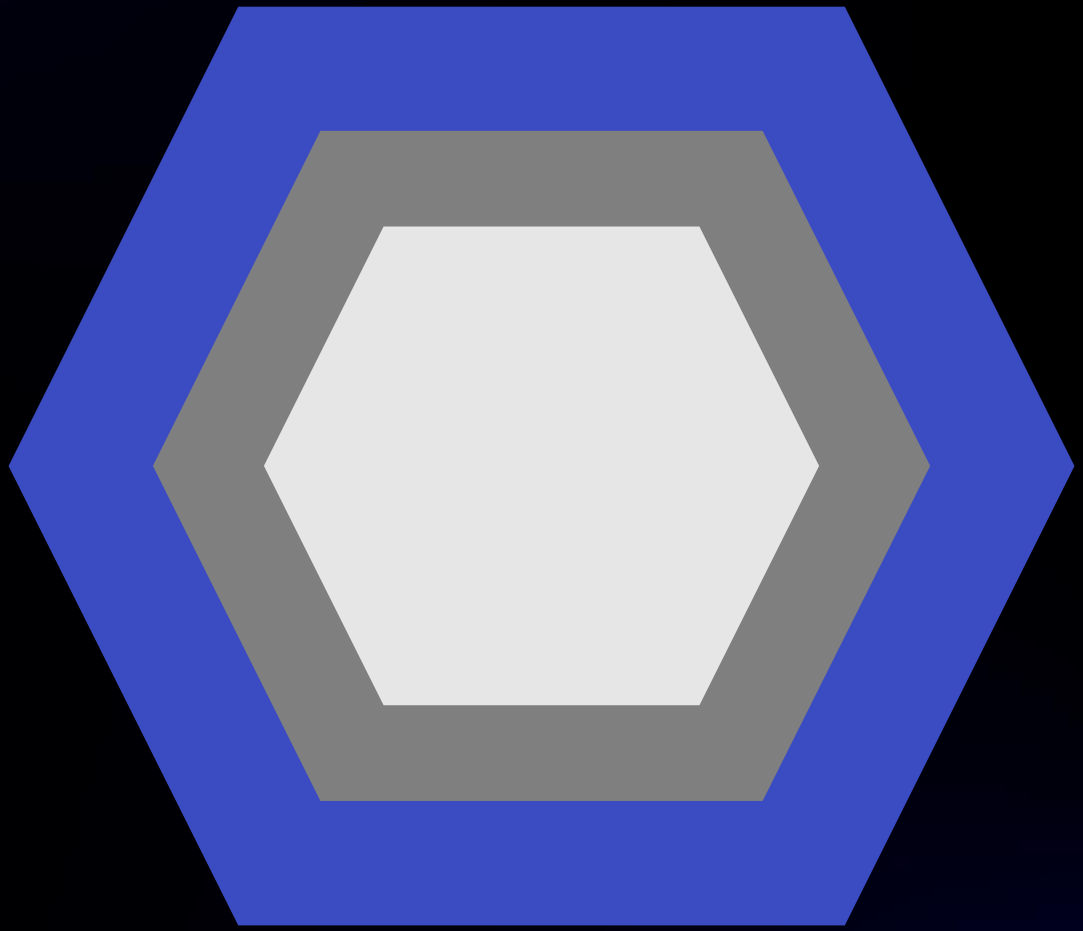
aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Additional use cases

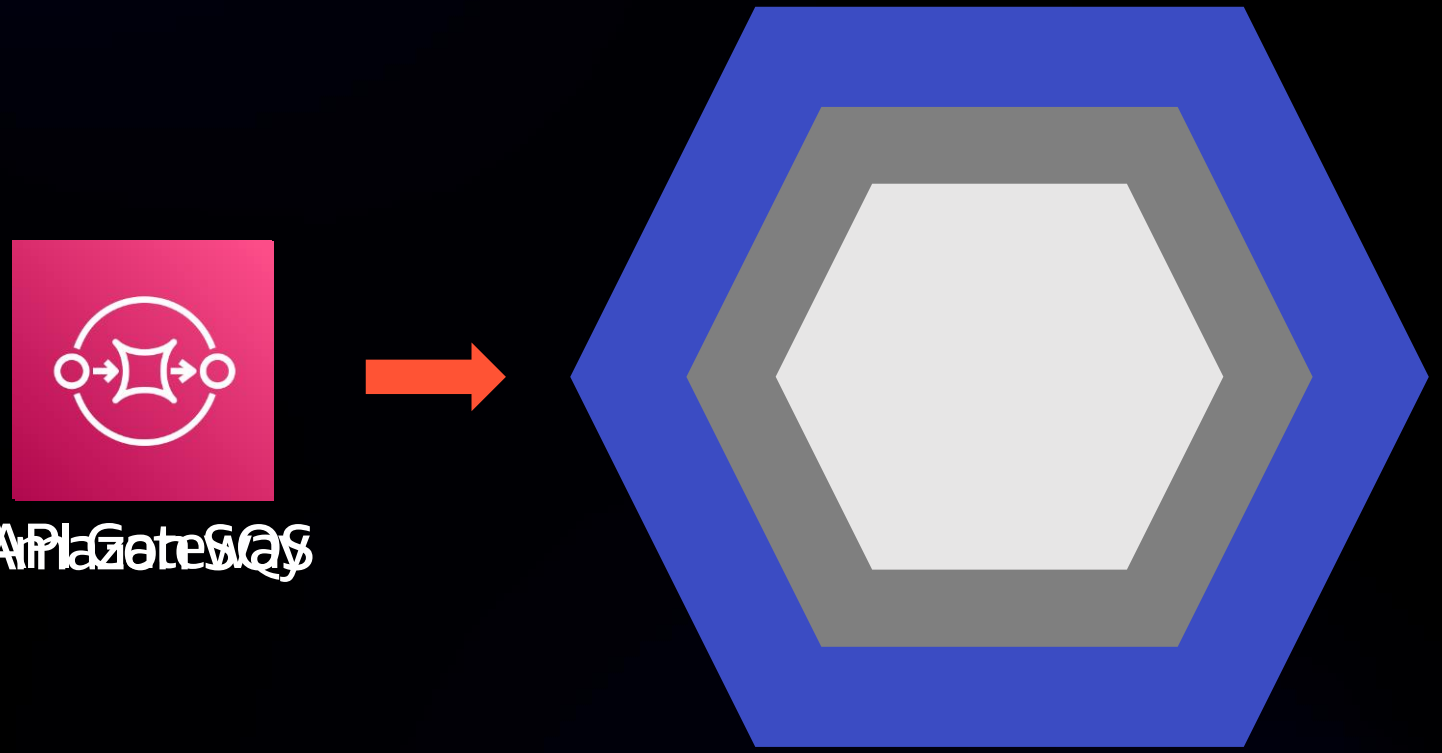
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Testing business logic in isolation



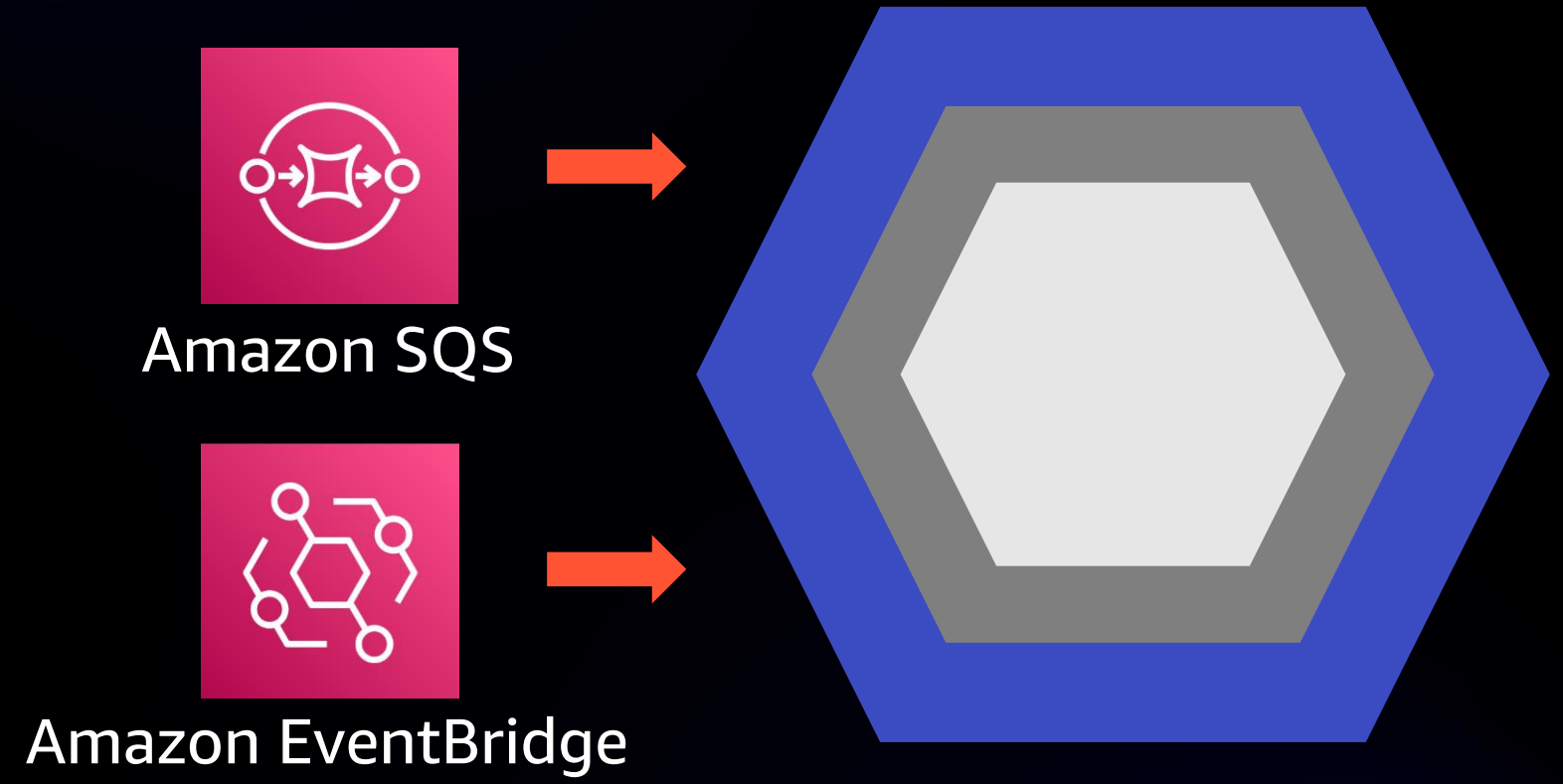
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Changing trigger



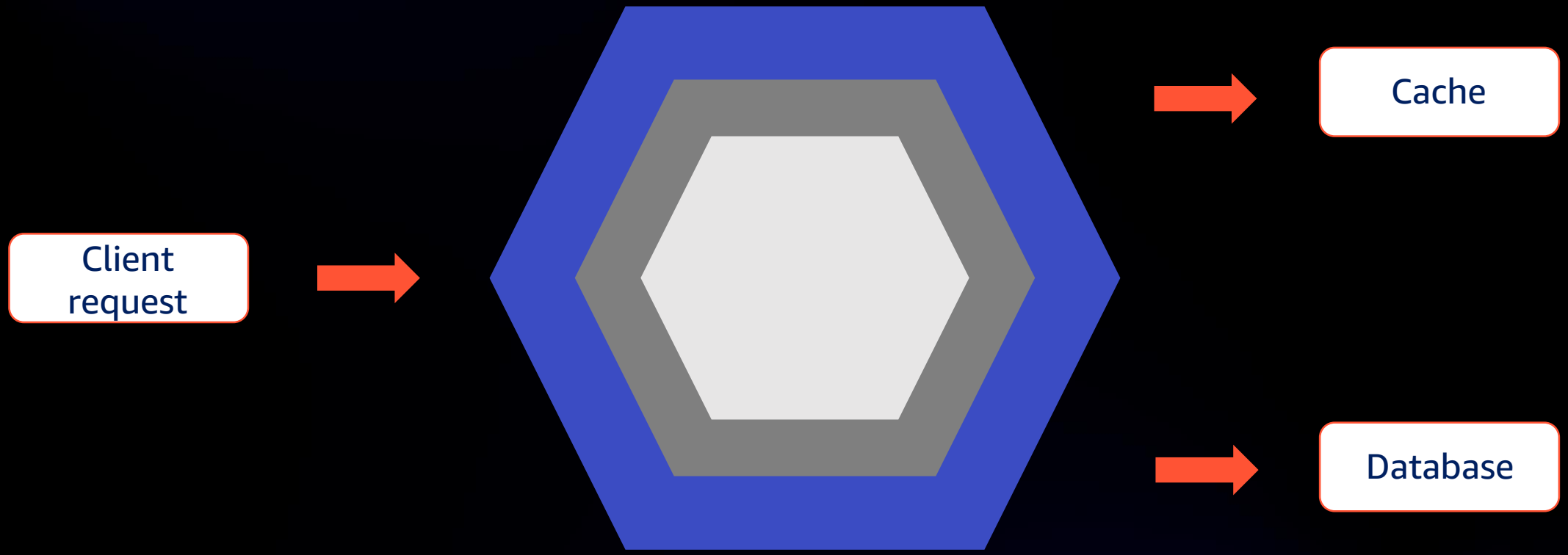
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Adding another trigger



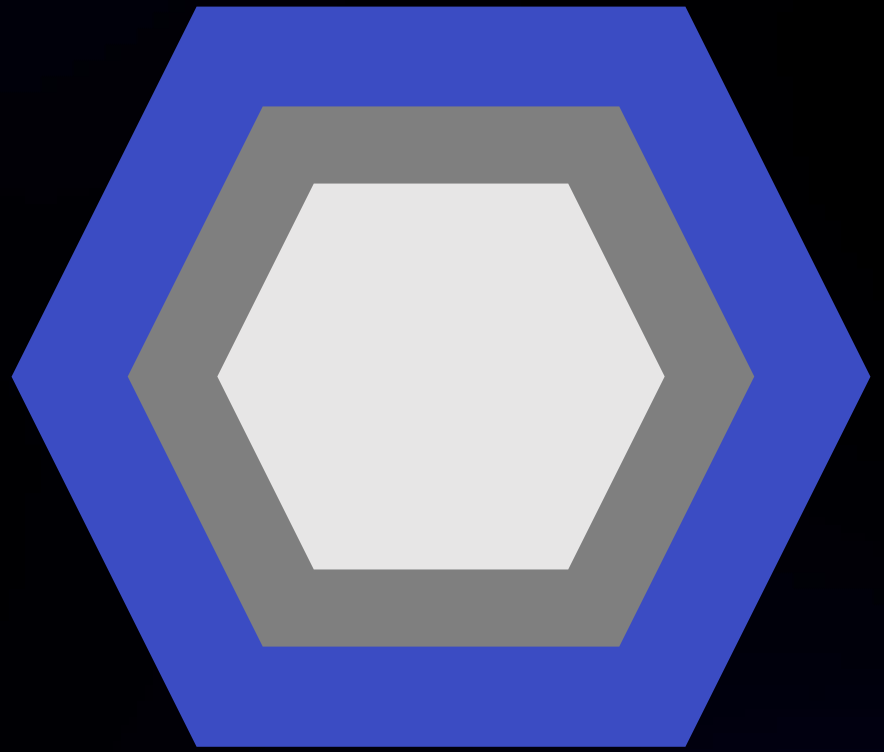
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

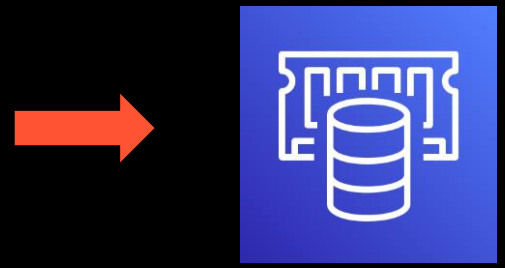
Cache-aside pattern



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Migration to AWS Managed Services





Amazon ElastiCache

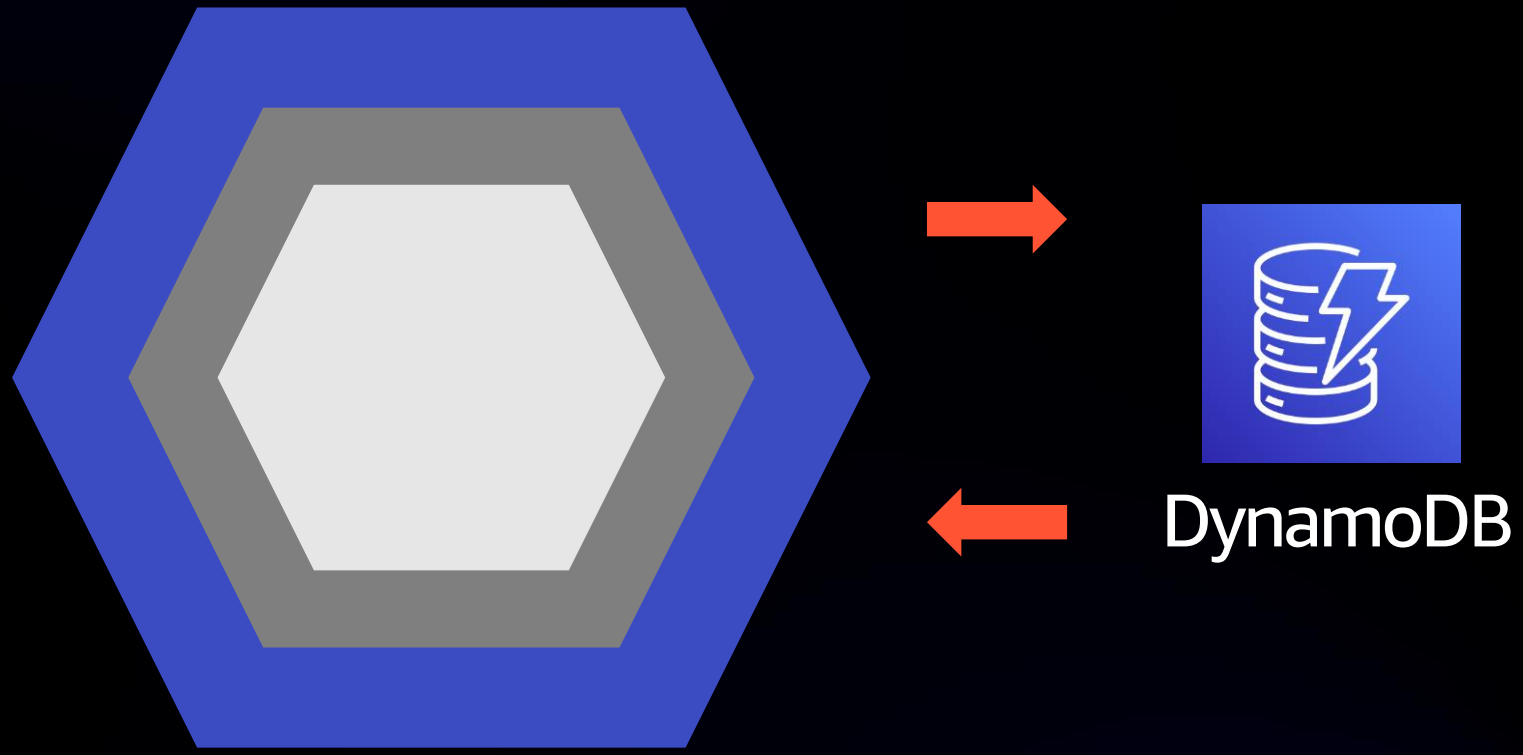


Redis

Redis is a trademark of Redis Labs Ltd. Any rights therein are reserved to Redis Labs Ltd. Any use by AWS is for referential purposes only and does not indicate any sponsorship, endorsement, or affiliation between Redis and AWS.

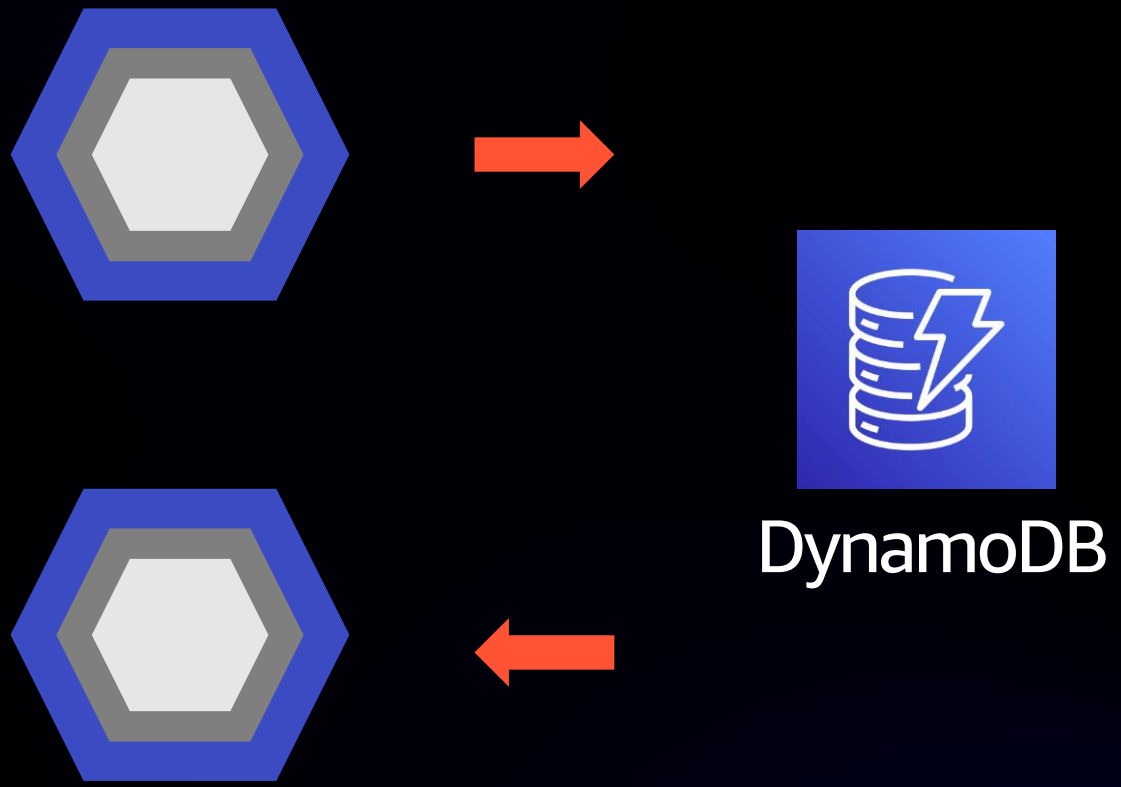
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CQRS pattern



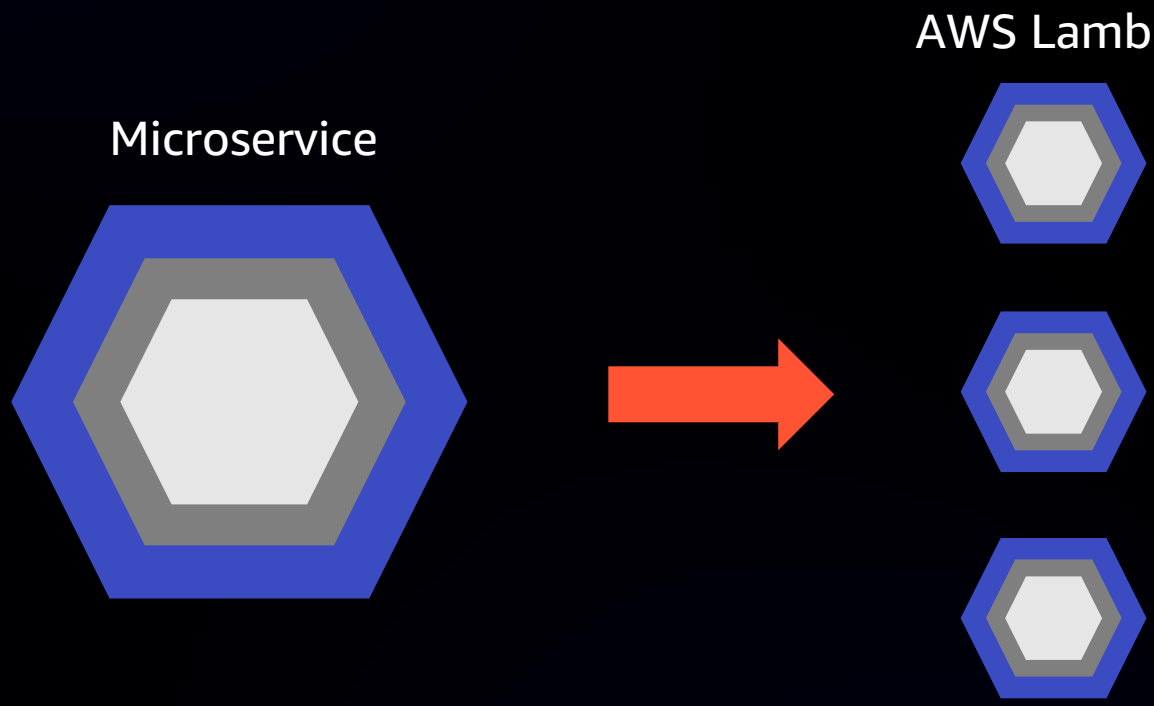
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CQRS pattern



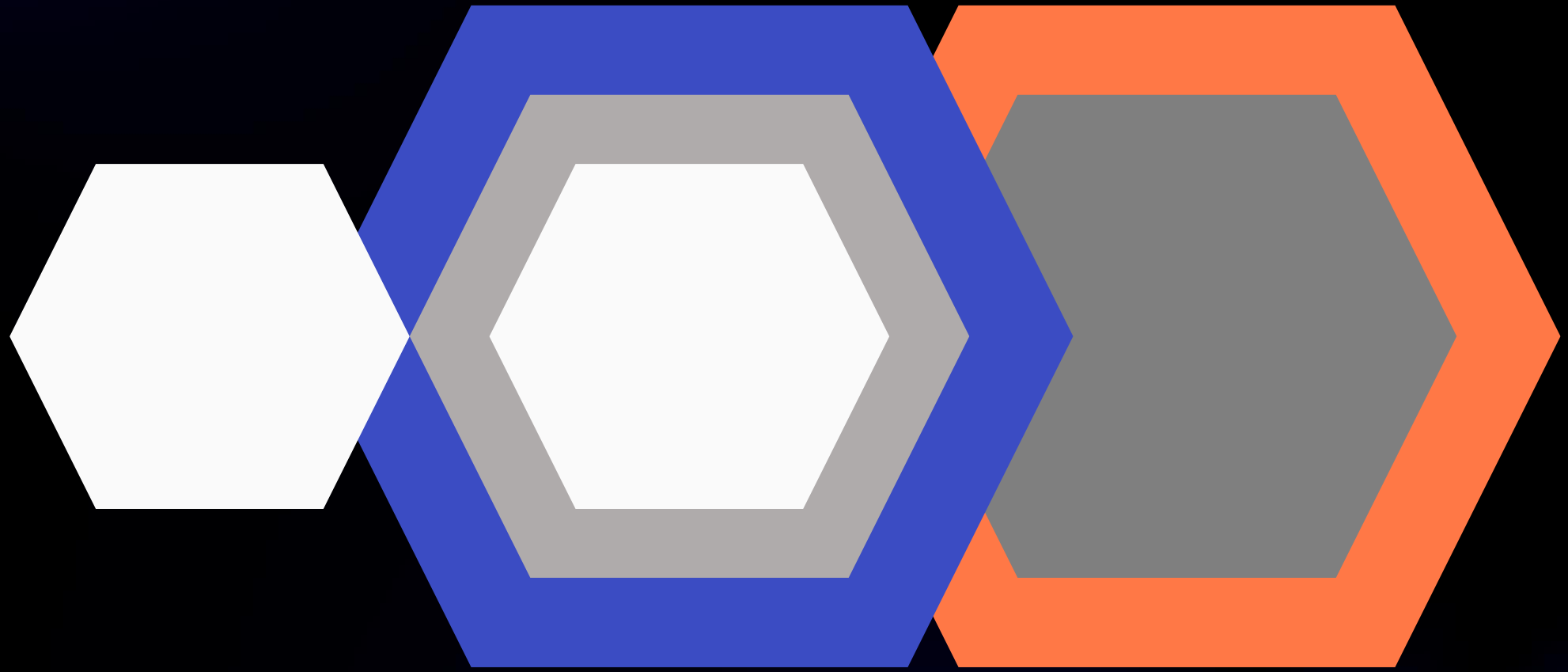
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Web applications modernization



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Hybrid architectures



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The evolution . . .

2005

Hexagonal architecture

2008

Onion architecture

2012

Clean architecture

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Is this the definitive architecture for AWS Lambda?

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Takeaway



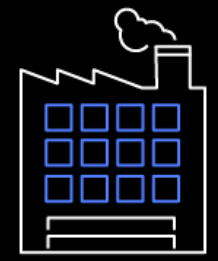
Separation of concerns



Infrastructure decoupled from business logic



Easy to test



Many use cases simplified

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Resources



https://go.aws/3EWz4Kf

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Luca Mezzalira

Twitter: @lucamezzalira
Email: lmezza@amazon.com

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.