COMMUNITY DAY

CZECH REPUBLIC

aws

COMMUNITY DAY

EDA for the Real World

From Architecture to Implementation

Author

Sandro Volpicella

Date

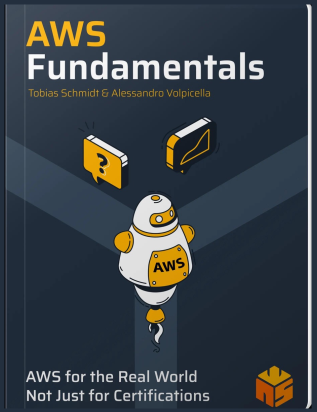
2025-04-29

aws
COMMUNITY DAY

whoami?

- Fullstack Software Engineer (AWS, Serverless) from Munich 🇩🇪
- Built up the whole EDA for hashnode with 1m+ events per month
- AWS Community Builder 🧡
- Author: AWS Fundamentals & CloudWatch Book

Author
Sandro Volpicella
@sandro_vol



AWS Fundamentals
Tobias Schmidt & Alessandro Volpicella
AWS for the Real World
Not Just for Certifications

The CloudWatch Book
Tobias Schmidt & Alessandro Volpicella
Do You Really Need This Expensive Third-Party Provider?

aws

COMMUNITY DAY

Theory & Architecture

1. What is EDA?
2. EDA on AWS
3. EventBridge Introduction
4. Envelope
5. Subscription pattern

Hands-On & TypeScript

6. Validating events
7. Errors & Retries
8. Archive & Replay
9. Idempotency
10. Testing
11. More & Resources

aws

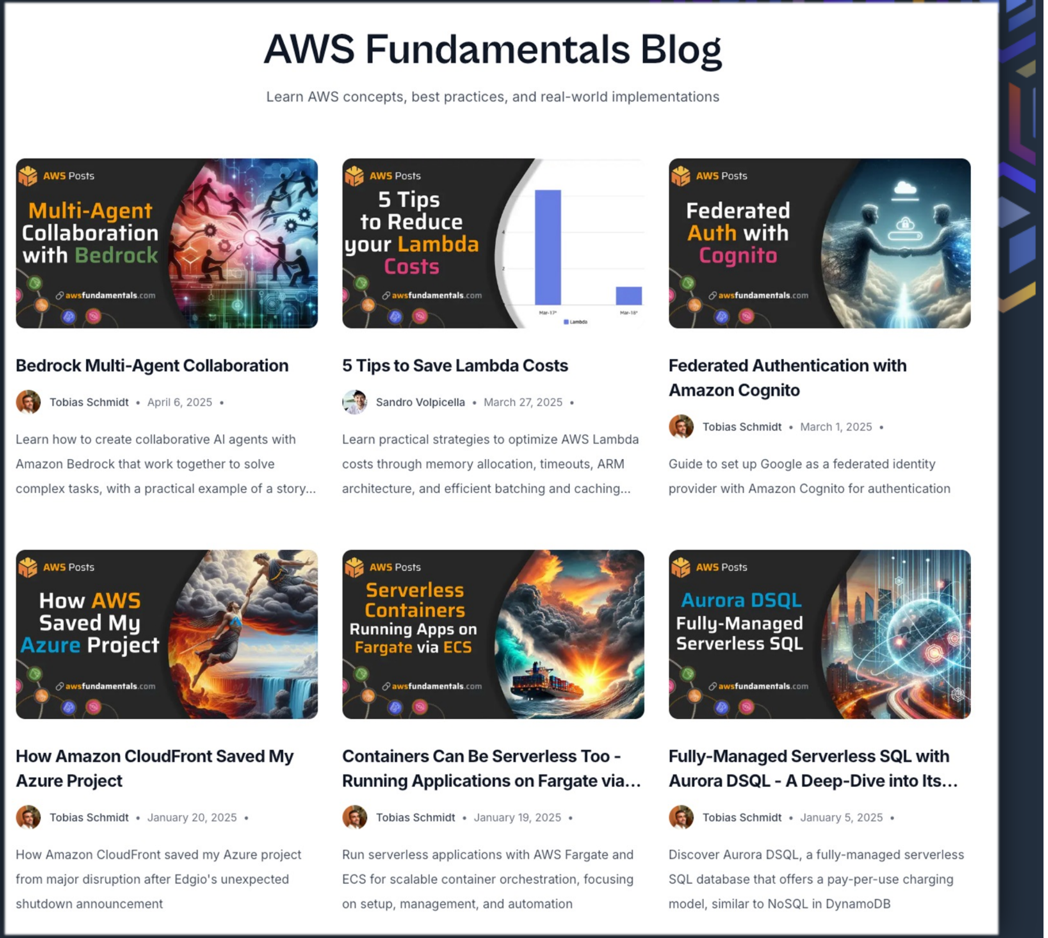
COMMUNITY DAY

Example Application - Blog

All examples will be on a blogging platform

Our blogging platform can:
- people can sign up
- create blogs
- read blogs
- publish blog post

Think of hashnode / medium / dev.to



AWS Fundamentals Blog

Learn AWS concepts, best practices, and real-world implementations

AWS Posts
Multi-Agent Collaboration with Bedrock
awsfundamentals.com
AWS Posts
5 Tips to Reduce your Lambda Costs
awsfundamentals.com
AWS Posts
Federated Auth with Cognito
awsfundamentals.com

Bedrock Multi-Agent Collaboration

Tobias Schmidt • April 6, 2025 •

Learn how to create collaborative AI agents with Amazon Bedrock that work together to solve complex tasks, with a practical example of a story...

5 Tips to Save Lambda Costs

Sandro Volpicella • March 27, 2025 •

Learn practical strategies to optimize AWS Lambda costs through memory allocation, timeouts, ARM architecture, and efficient batching and caching...

Federated Authentication with Amazon Cognito

Tobias Schmidt • March 1, 2025 •

Guide to set up Google as a federated identity provider with Amazon Cognito for authentication

AWS Posts
How AWS Saved My Azure Project
awsfundamentals.com
AWS Posts
Serverless Containers Running Apps on Fargate via ECS
awsfundamentals.com
AWS Posts
Aurora DSQL Fully-Managed Serverless SQL
awsfundamentals.com

How Amazon CloudFront Saved My Azure Project

Tobias Schmidt • January 20, 2025 •

How Amazon CloudFront saved my Azure project from major disruption after Edgio's unexpected shutdown announcement

Containers Can Be Serverless Too - Running Applications on Fargate via...

Tobias Schmidt • January 19, 2025 •

Run serverless applications with AWS Fargate and ECS for scalable container orchestration, focusing on setup, management, and automation

Fully-Managed Serverless SQL with Aurora DSQL - A Deep-Dive into Its...

Tobias Schmidt • January 5, 2025 •

Discover Aurora DSQL, a fully-managed serverless SQL database that offers a pay-per-use charging model, similar to NoSQL in DynamoDB

What is EDA?

aws

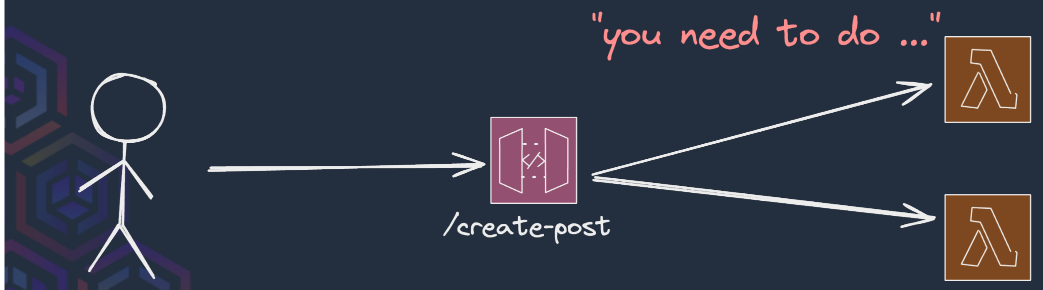
COMMUNITY DAY

What is EDA?

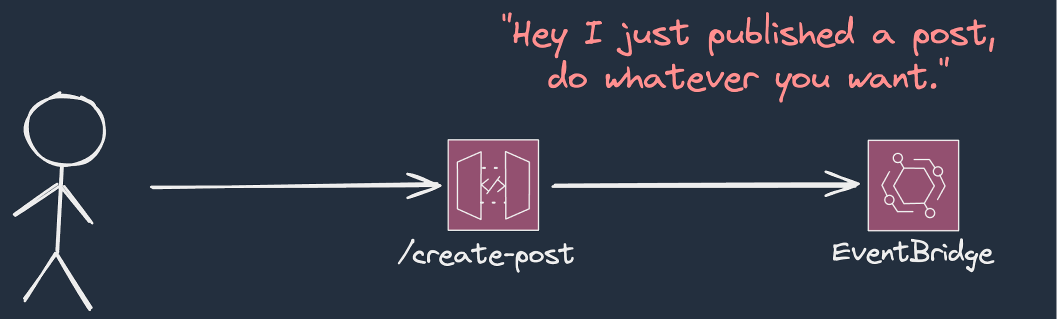
- Systems **react** to events instead of being told what to do
- Technical Goal: Decoupling of your system
- Real-World Goal: Feature velocity, Resilience, Reliability

aws COMMUNITY DAY

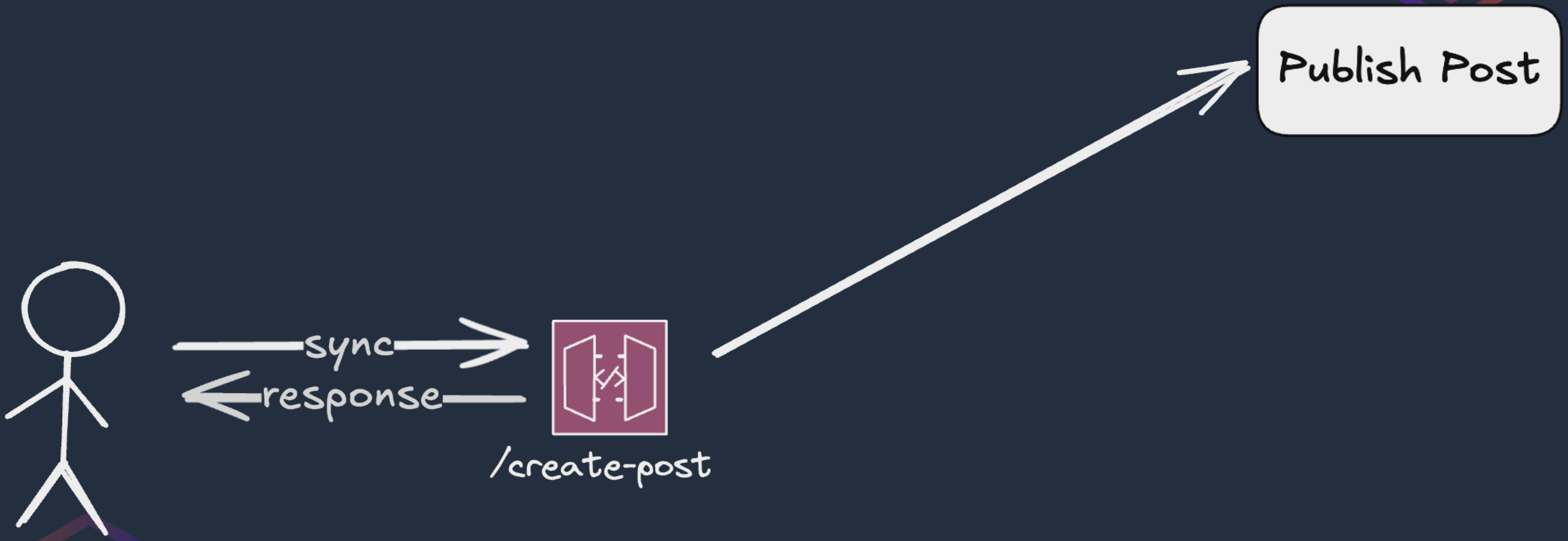
Function-Driven

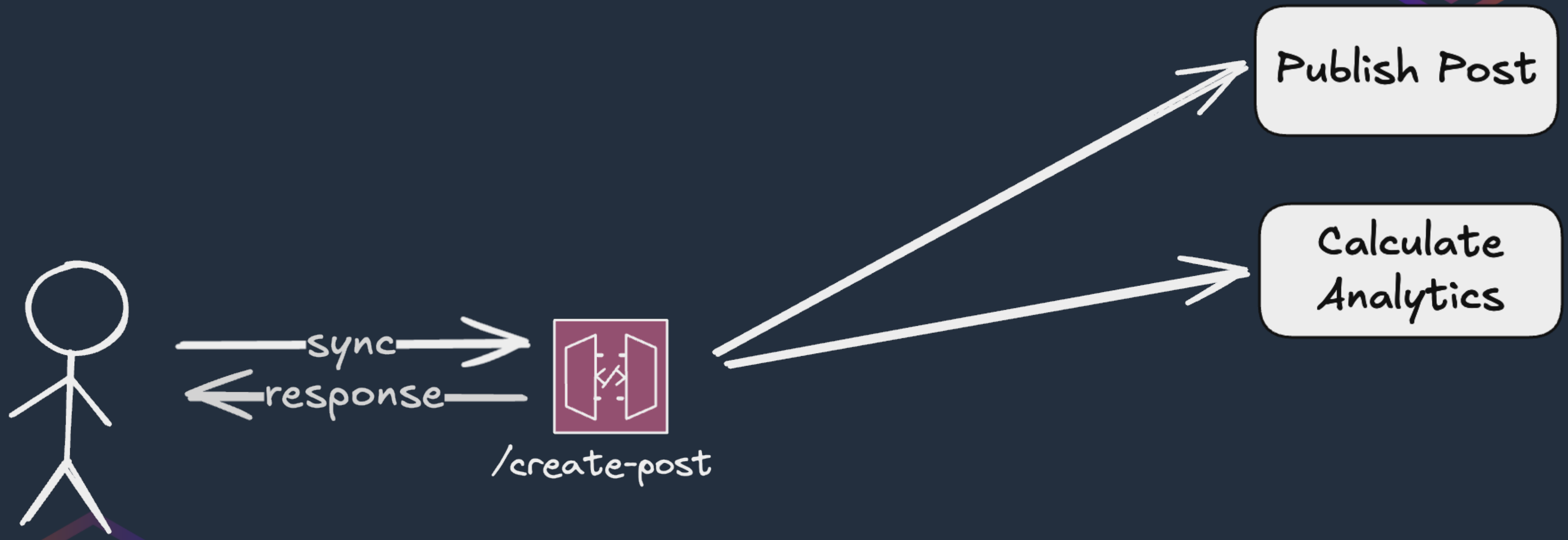


Event-Driven



aws COMMUNITY DAY





sync

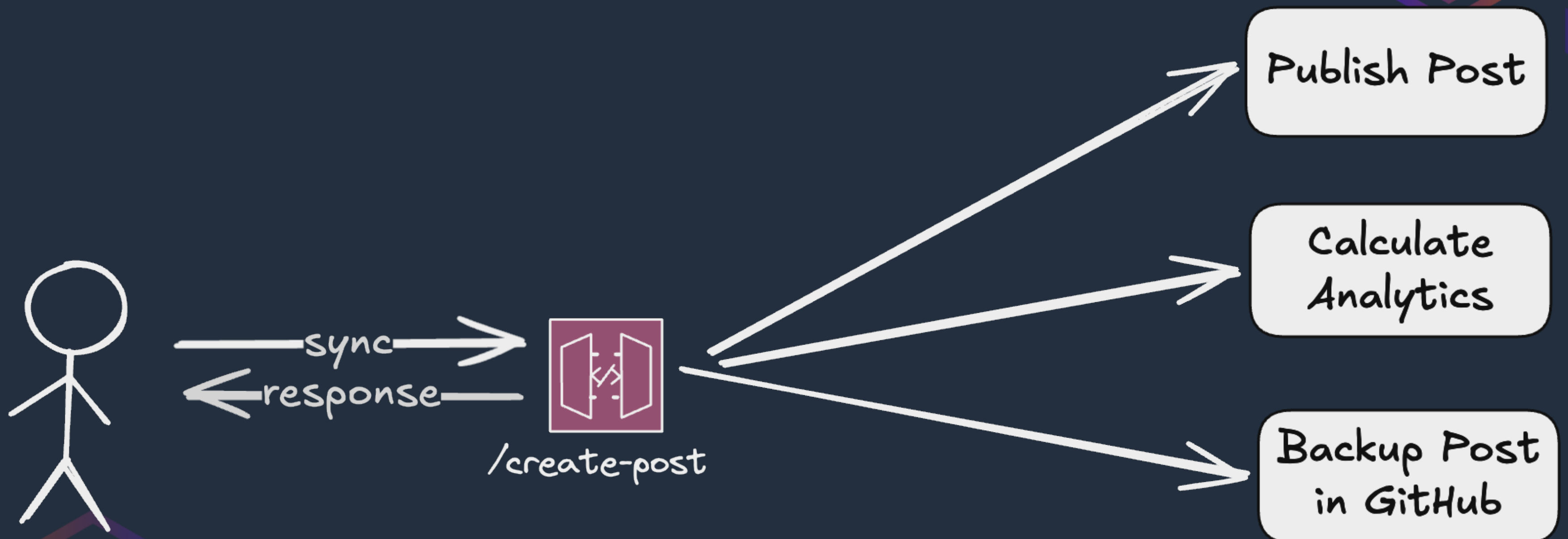
response

/create-post

Publish Post

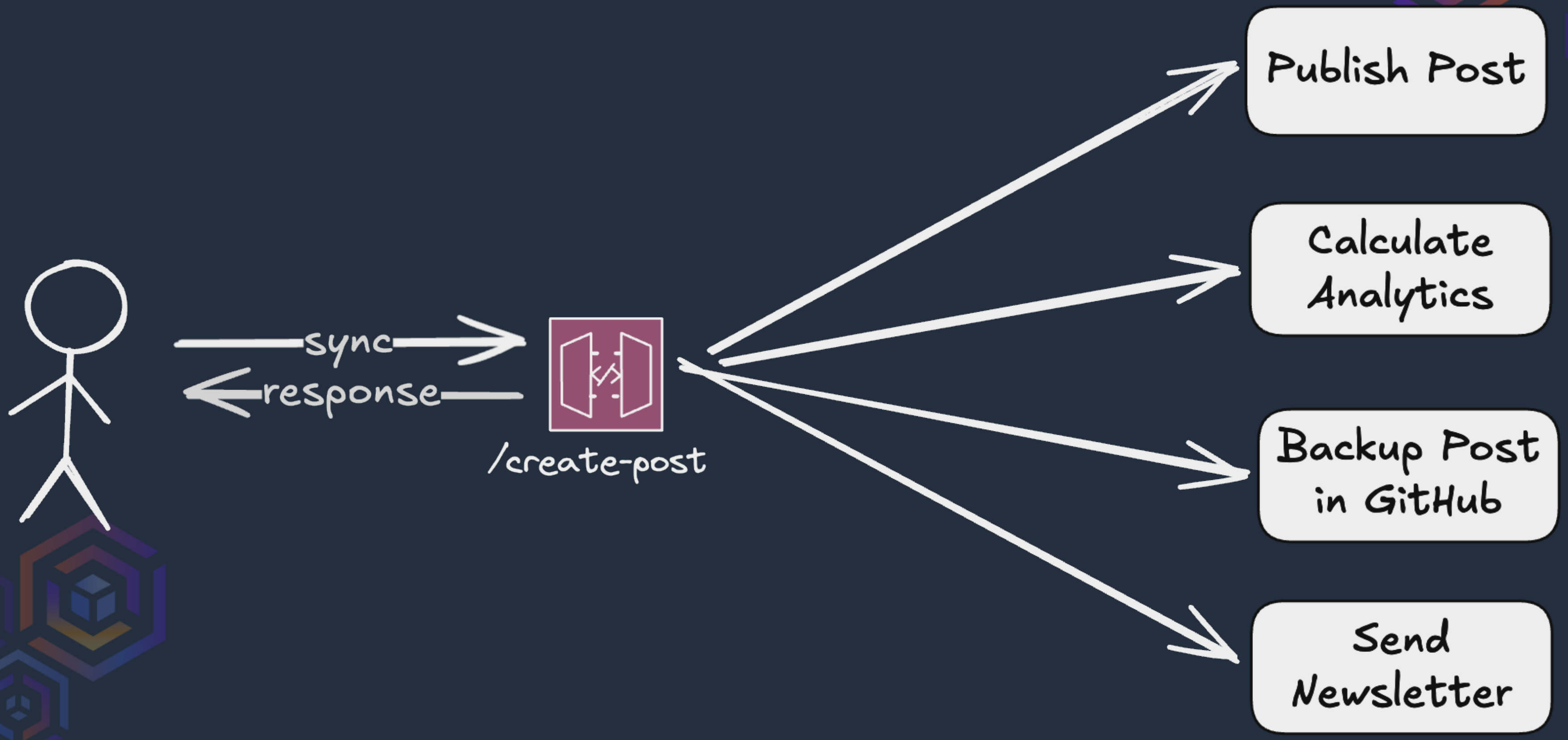
Calculate Analytics

aws
COMMUNITY DAY



aws

COMMUNITY DAY



sync

response

/create-post

Publish Post

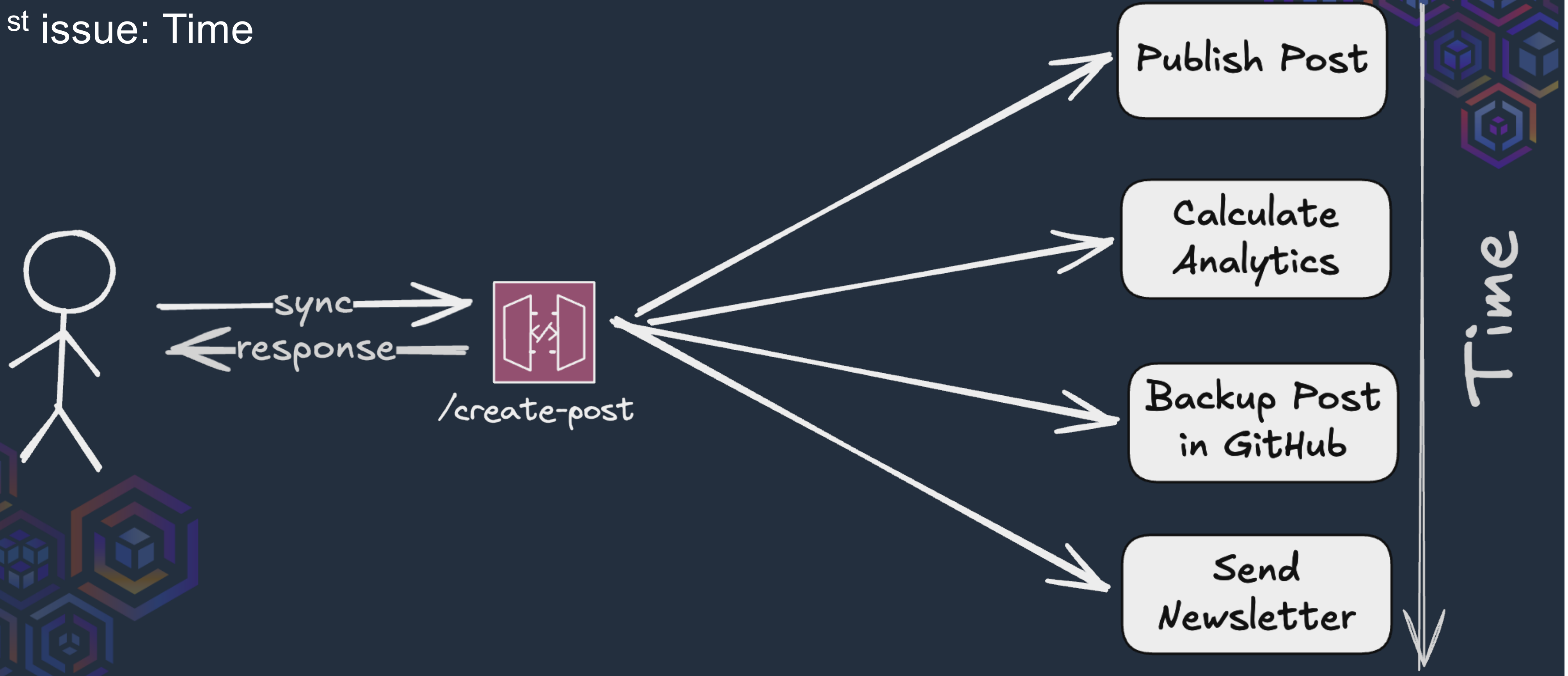
Calculate Analytics

Backup Post in GitHub

Send Newsletter

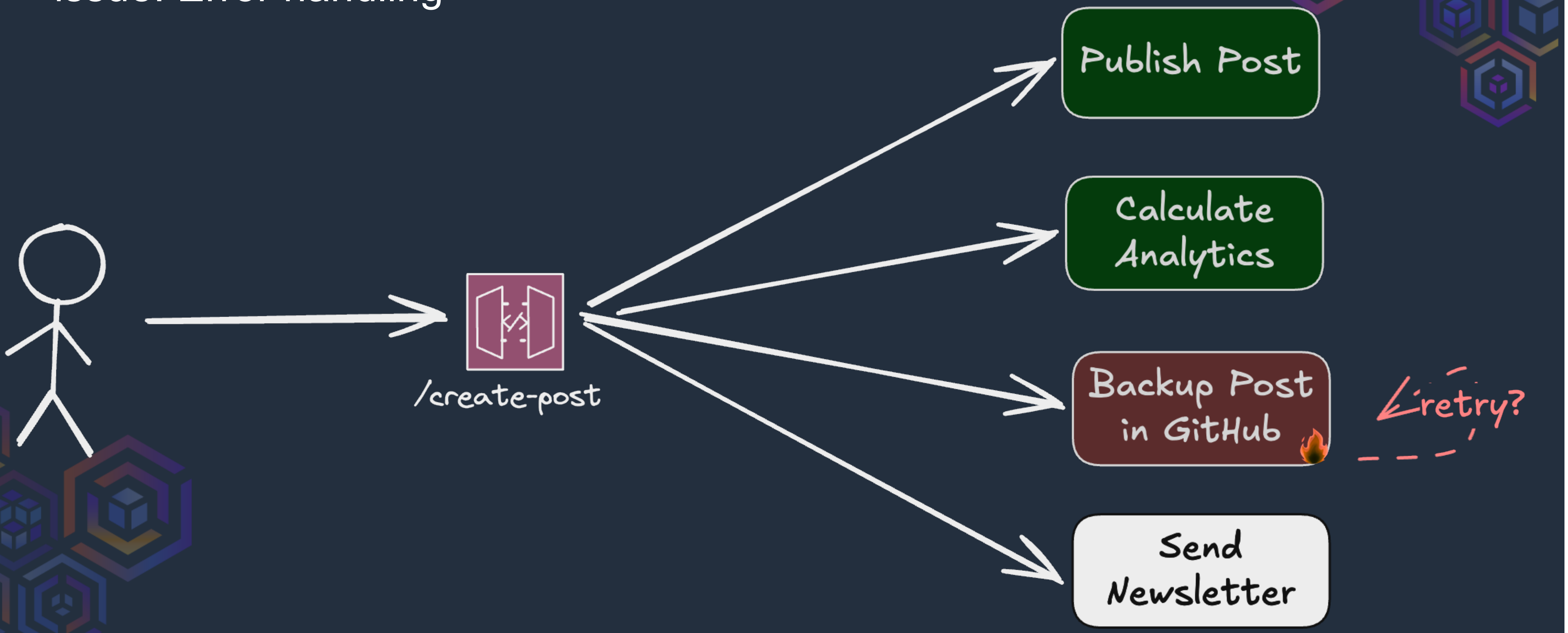
AWS COMMUNITY DAY

1st issue: Time



aws COMMUNITY DAY

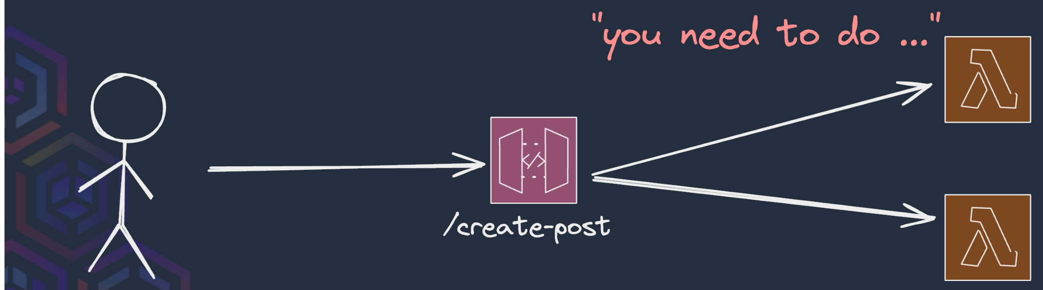
2nd issue: Error handling



aws

COMMUNITY DAY

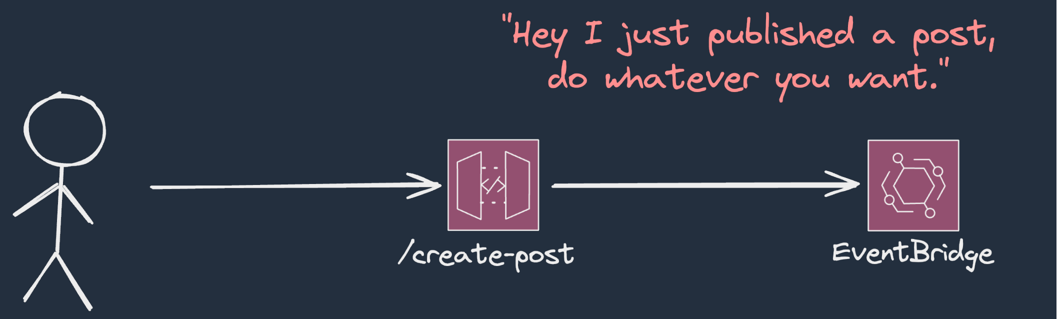
Function-Driven



"you need to do ..."

/create-post

Event-Driven



"Hey I just published a post, do whatever you want."

/create-post

EventBridge

How to EDA on AWS?

aws

COMMUNITY DAY

How to EDA on AWS?

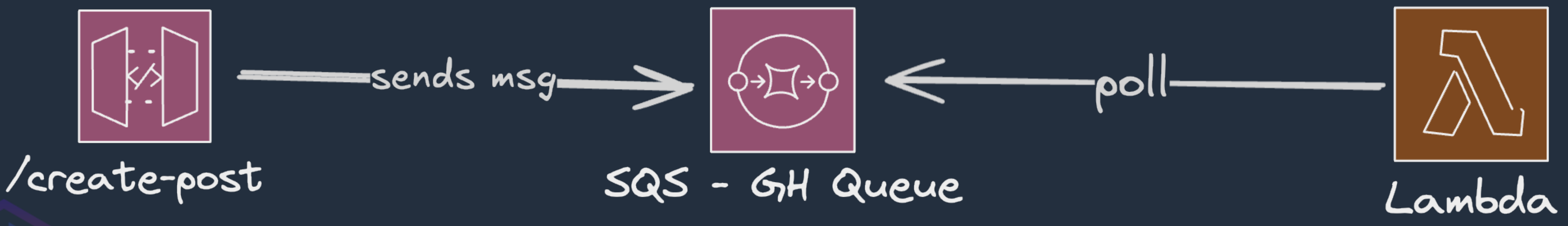
SQS

SNS

EventBridge

aws COMMUNITY DAY

How to EDA on AWS? - SQS

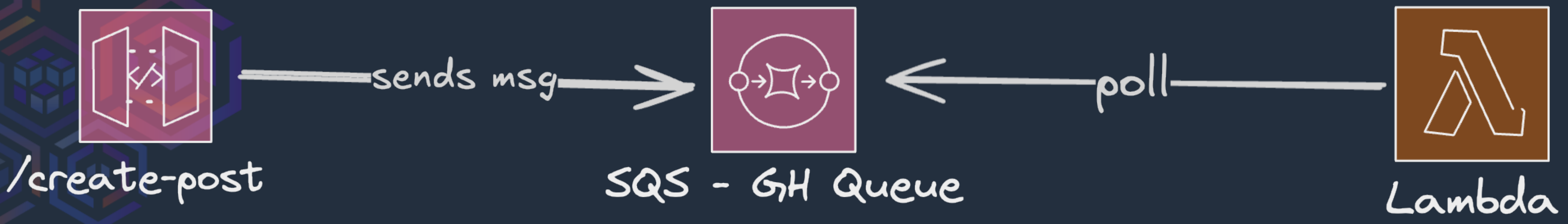


aws COMMUNITY DAY

How to EDA on AWS? - SQS

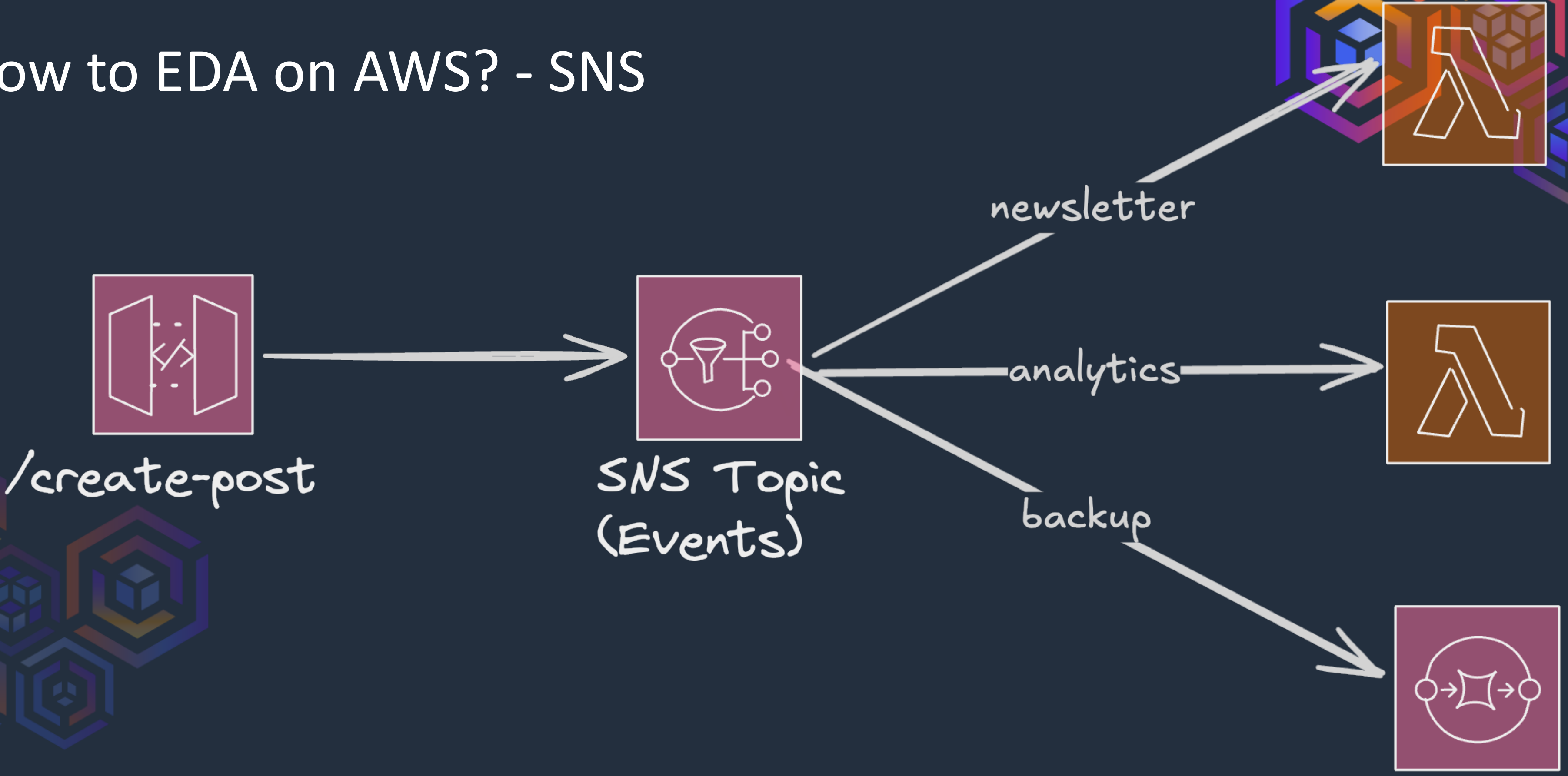
✅ **Async**
✅ **Better error handling**

❌ **One** queue for **one** task
❌ **Coupling**



aws COMMUNITY DAY

How to EDA on AWS? - SNS

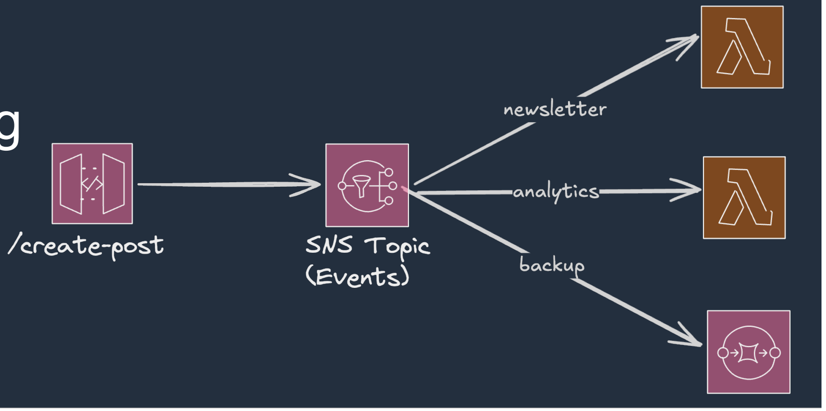


aws COMMUNITY DAY

How to EDA on AWS? - SNS

✅ **Coupling**: Decoupled Producers from Consumers
✅ **Publish/Subscribe**: Consumers easily subscribe/unsubscribe from topics

❌ **Routing** via filtering
❌ **Archive & Replay:** Saving all events & replaying them

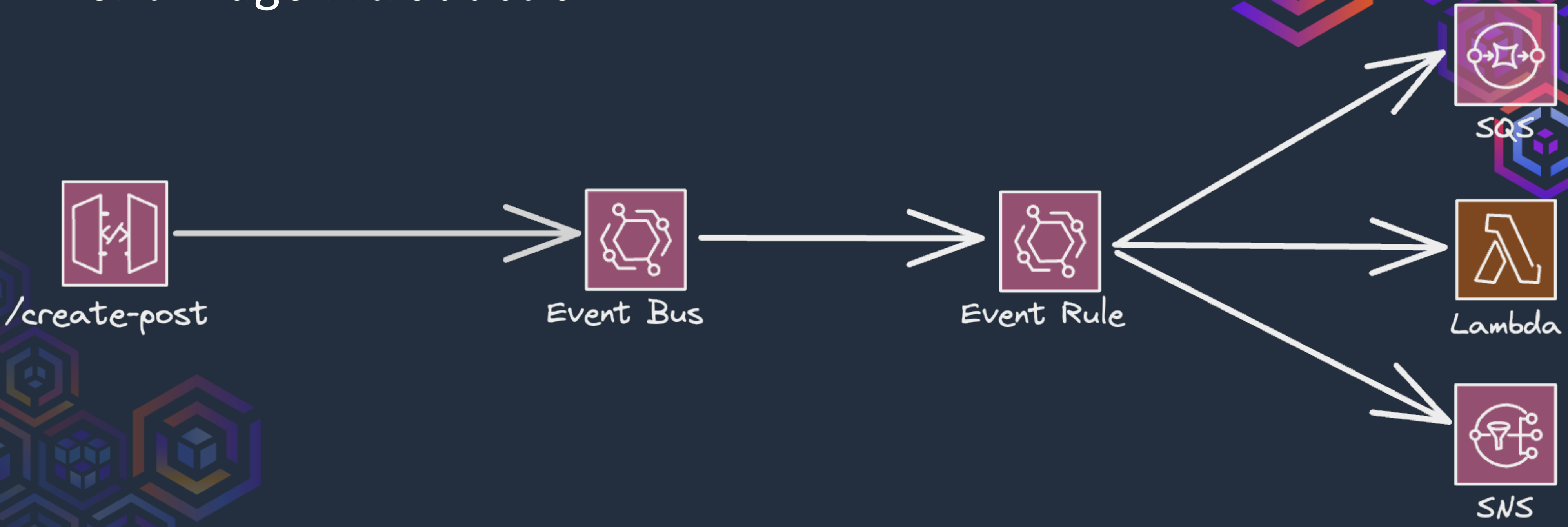


EventBridge Introduction

aws

COMMUNITY DAY

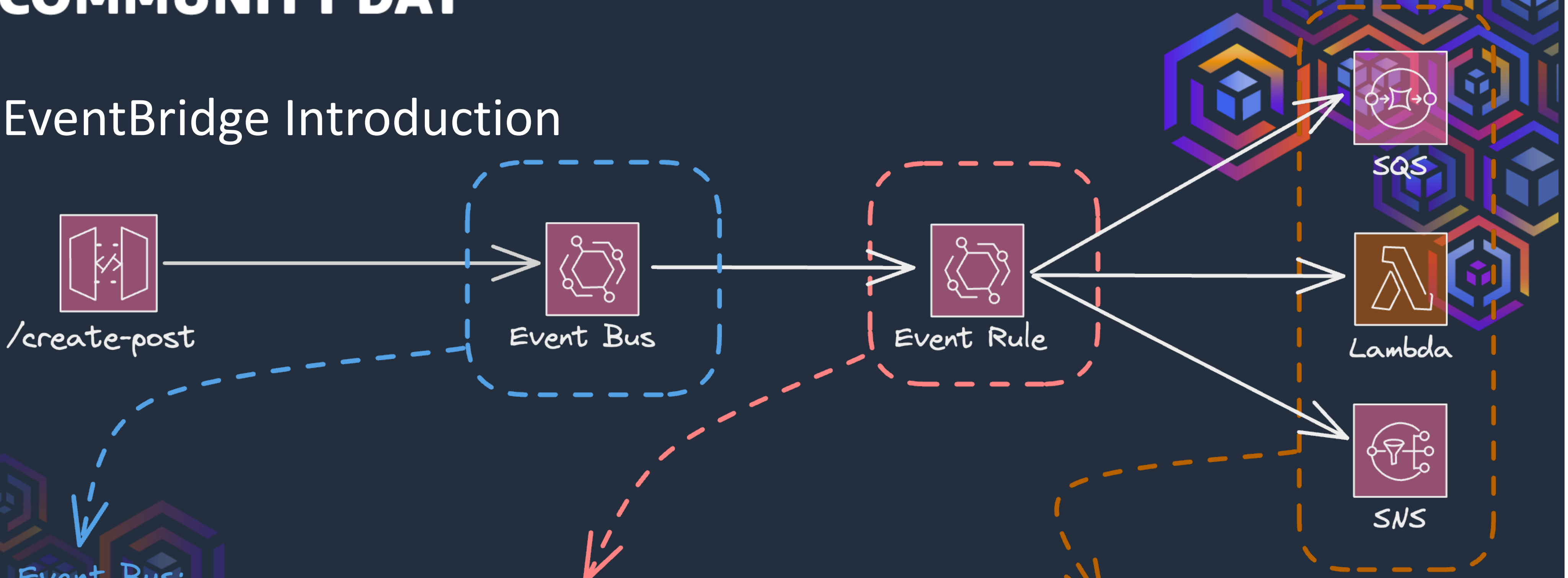
EventBridge Introduction



aws

COMMUNITY DAY

EventBridge Introduction



Event Bus:
- default
- custom
- partner (stripe, shopify, mongodb)

Event Rule:
subscribe on event "createPost"
if isGithubEnabled: true

Targets:
- Consumers of events
- Lambda, SQS, SNS, API, etc.

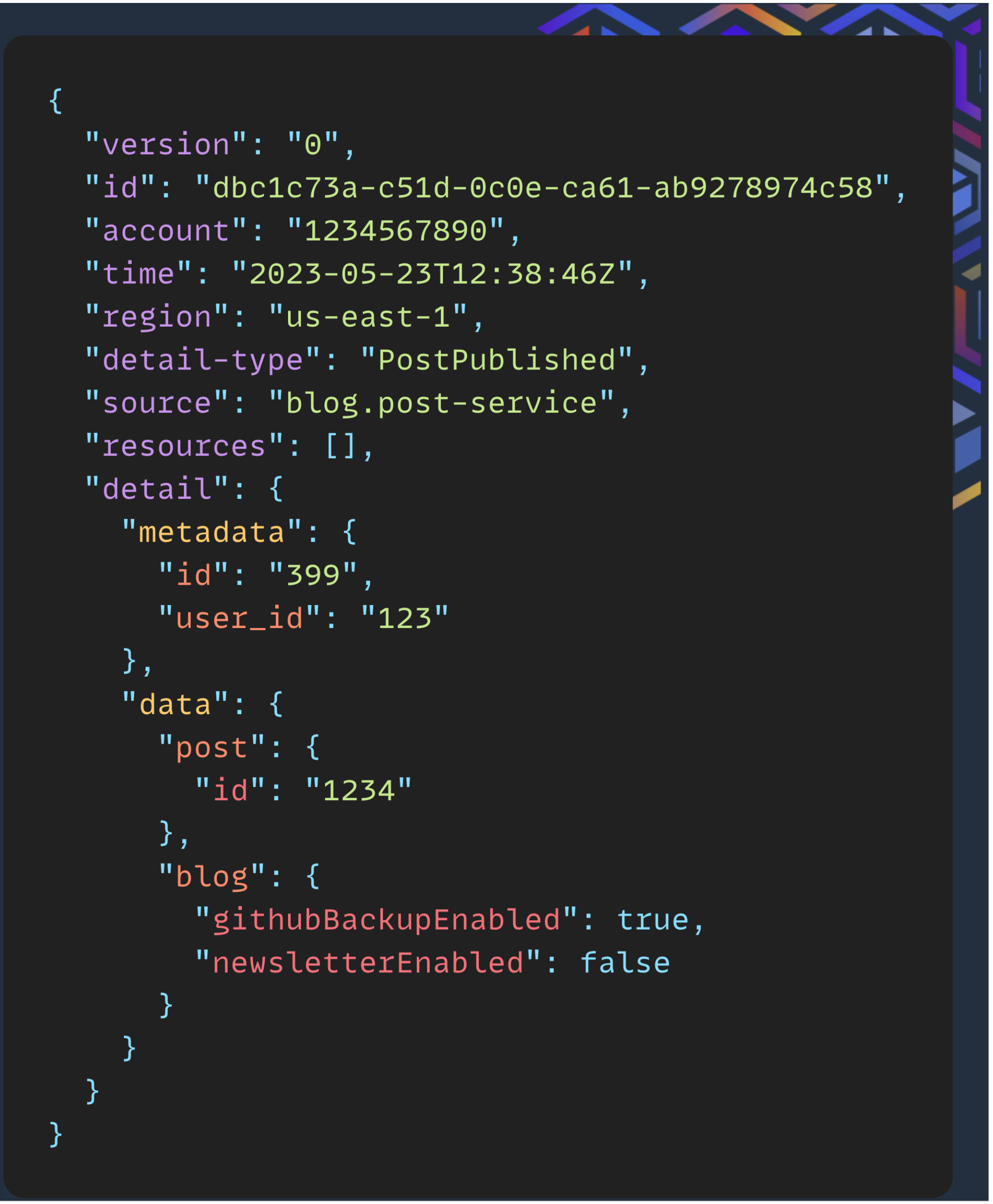
Event Envelope

aws

COMMUNITY DAY

Event Envelope

How does an event look like?



```json
{
  "version": "0",
  "id": "dbc1c73a-c51d-0c0e-ca61-ab9278974c58",
  "account": "1234567890",
  "time": "2023-05-23T12:38:46Z",
  "region": "us-east-1",
  "detail-type": "PostPublished",
  "source": "blog.post-service",
  "resources": [],
  "detail": {
    "metadata": {
      "id": "399",
      "user_id": "123"
    },
    "data": {
      "post": {
        "id": "1234"
      },
      "blog": {
        "githubBackupEnabled": true,
        "newsletterEnabled": false
      }
    }
  }
}
```

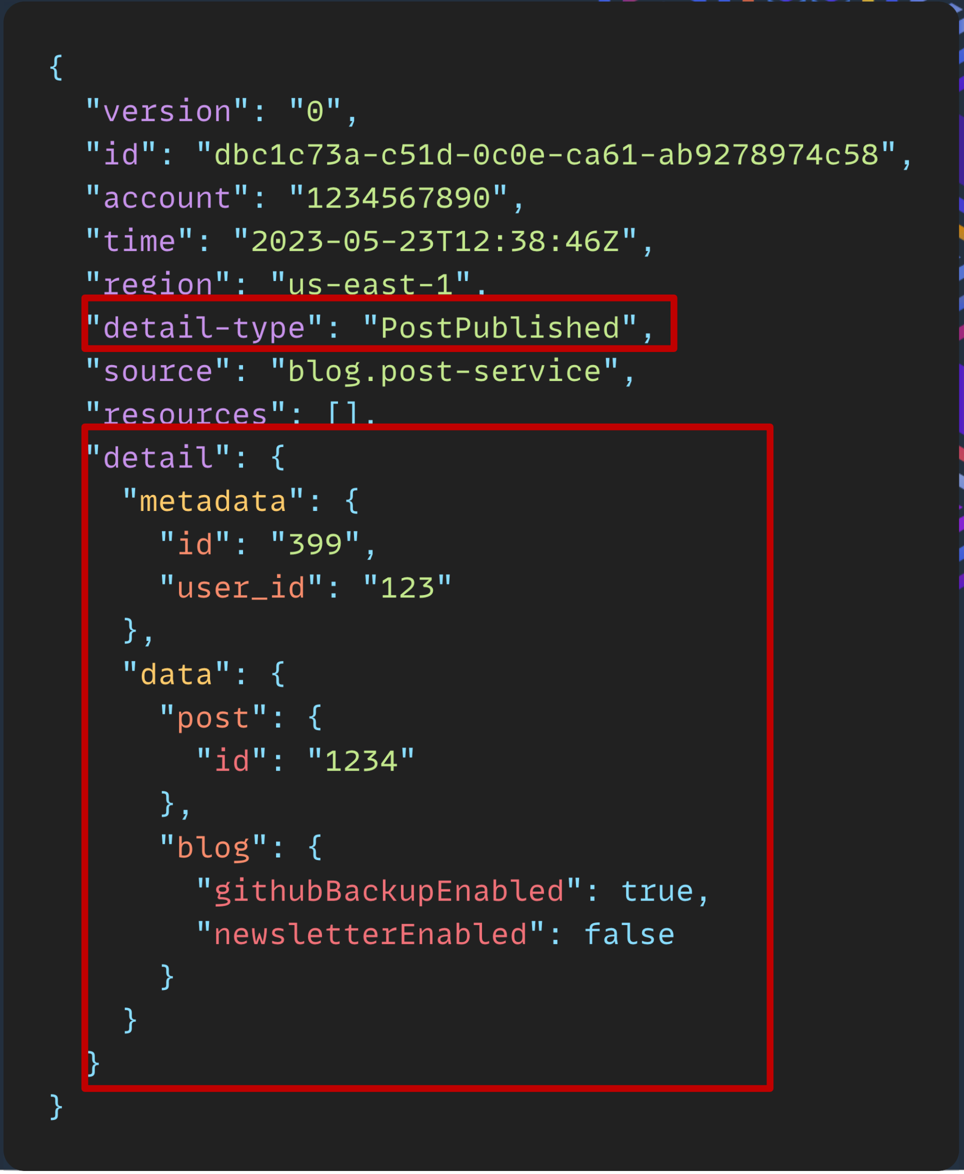
aws

COMMUNITY DAY

Event Envelope

How does an event look like?

1. detail-type: “Type of event”
2. detail: data



```json
{
  "version": "0",
  "id": "dbc1c73a-c51d-0c0e-ca61-ab9278974c58",
  "account": "1234567890",
  "time": "2023-05-23T12:38:46Z",
  "region": "us-east-1",
  "detail-type": "PostPublished",
  "source": "blog.post-service",
  "resources": [],
  "detail": {
    "metadata": {
      "id": "399",
      "user_id": "123"
    },
    "data": {
      "post": {
        "id": "1234"
      },
      "blog": {
        "githubBackupEnabled": true,
        "newsletterEnabled": false
      }
    }
  }
}
```

Event Envelope

Why?

- **Idempotency** – Deduplicate your events
- **Versioning** – Schema will evolve, handle versions
- **Observability** – Trace events through your system

Event Envelope – Two approaches

1. Metadata / Data
2. CloudEvents

aws

COMMUNITY DAY

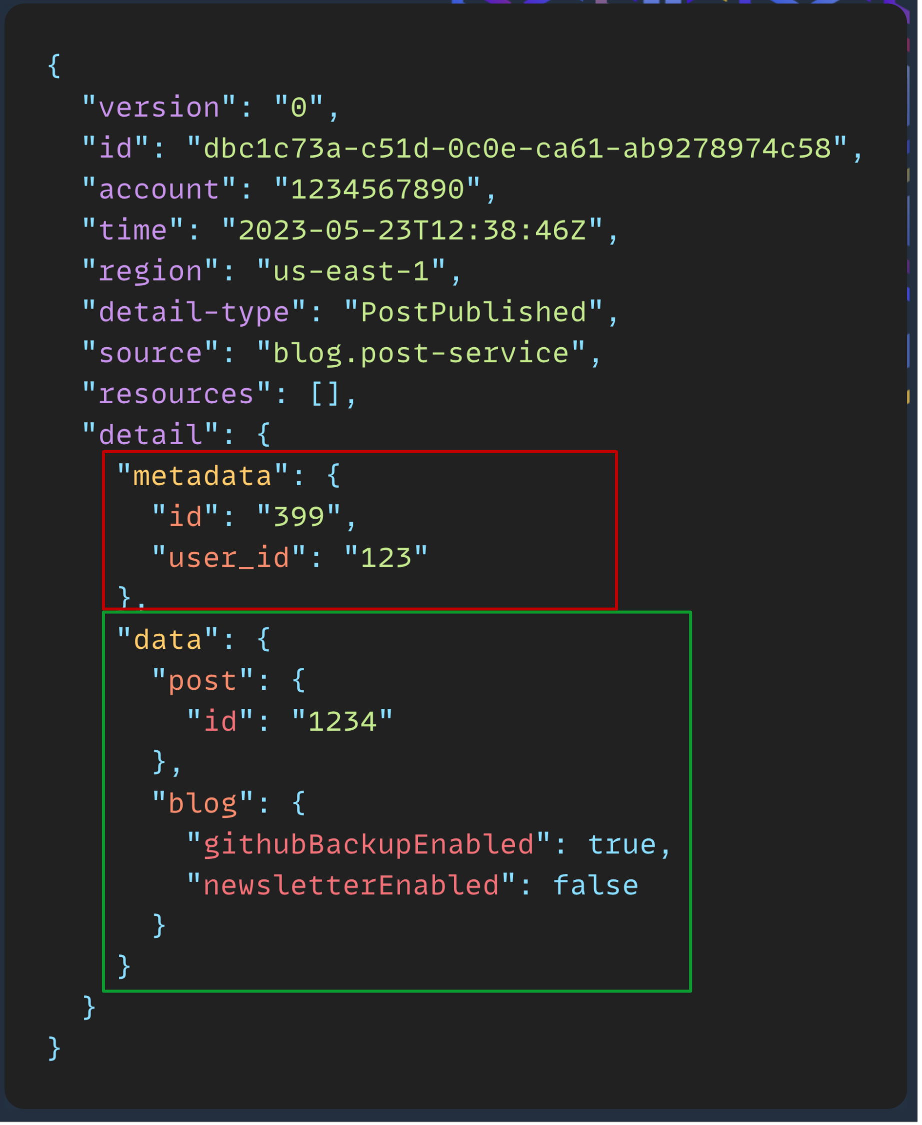
Event Envelope – Metadata / Data

Metadata – Common fields across all events

- id
- user
- date

Data – Unique for this event

- post
- blog



```
{
  "version": "0",
  "id": "dbc1c73a-c51d-0c0e-ca61-ab9278974c58",
  "account": "1234567890",
  "time": "2023-05-23T12:38:46Z",
  "region": "us-east-1",
  "detail-type": "PostPublished",
  "source": "blog.post-service",
  "resources": [],
  "detail": {
    "metadata": {
      "id": "399",
      "user_id": "123"
    },
    "data": {
      "post": {
        "id": "1234"
      },
      "blog": {
        "githubBackupEnabled": true,
        "newsletterEnabled": false
      }
    }
  }
}
```

aws

COMMUNITY DAY

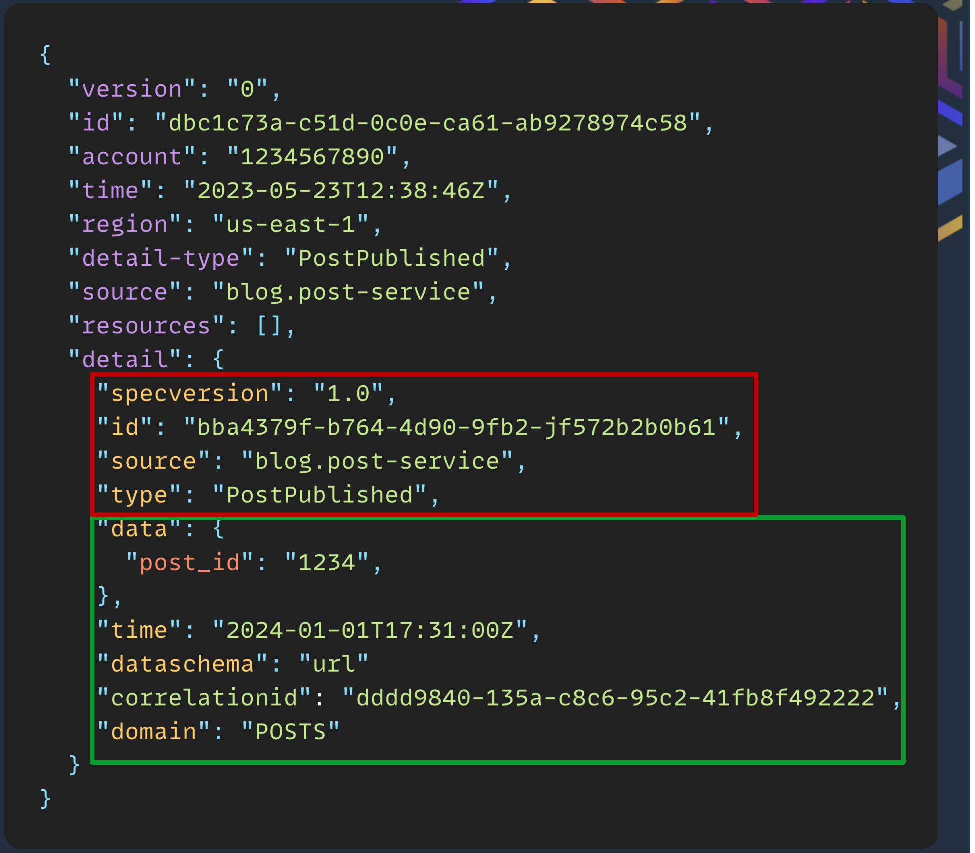
Event Envelope – CloudEvents(.io)

Required Fields

- id
- source
- type

Optional Fields – Unique for this event

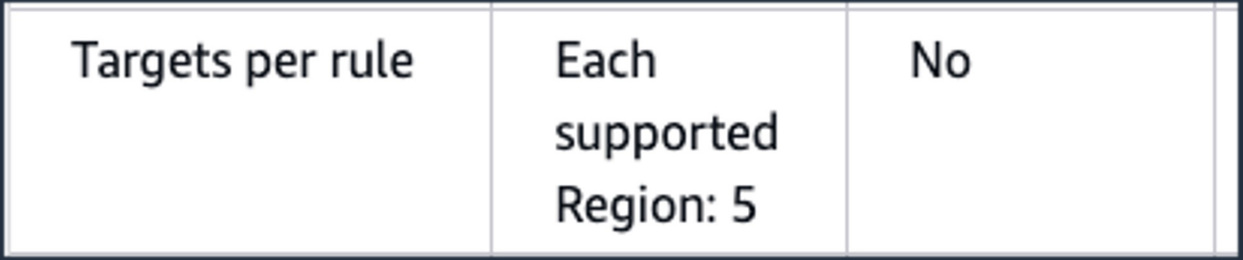
- data
- time
- correlationid



```
{
  "version": "0",
  "id": "dbc1c73a-c51d-0c0e-ca61-ab9278974c58",
  "account": "1234567890",
  "time": "2023-05-23T12:38:46Z",
  "region": "us-east-1",
  "detail-type": "PostPublished",
  "source": "blog.post-service",
  "resources": [],
  "detail": {
    "specversion": "1.0",
    "id": "bba4379f-b764-4d90-9fb2-jf572b2b0b61",
    "source": "blog.post-service",
    "type": "PostPublished",
    "data": {
      "post_id": "1234",
    },
    "time": "2024-01-01T17:31:00Z",
    "dataschema": "url"
    "correlationid": "dddd9840-135a-c8c6-95c2-41fb8f492222",
    "domain": "POSTS"
  }
}
```

Subscription Pattern

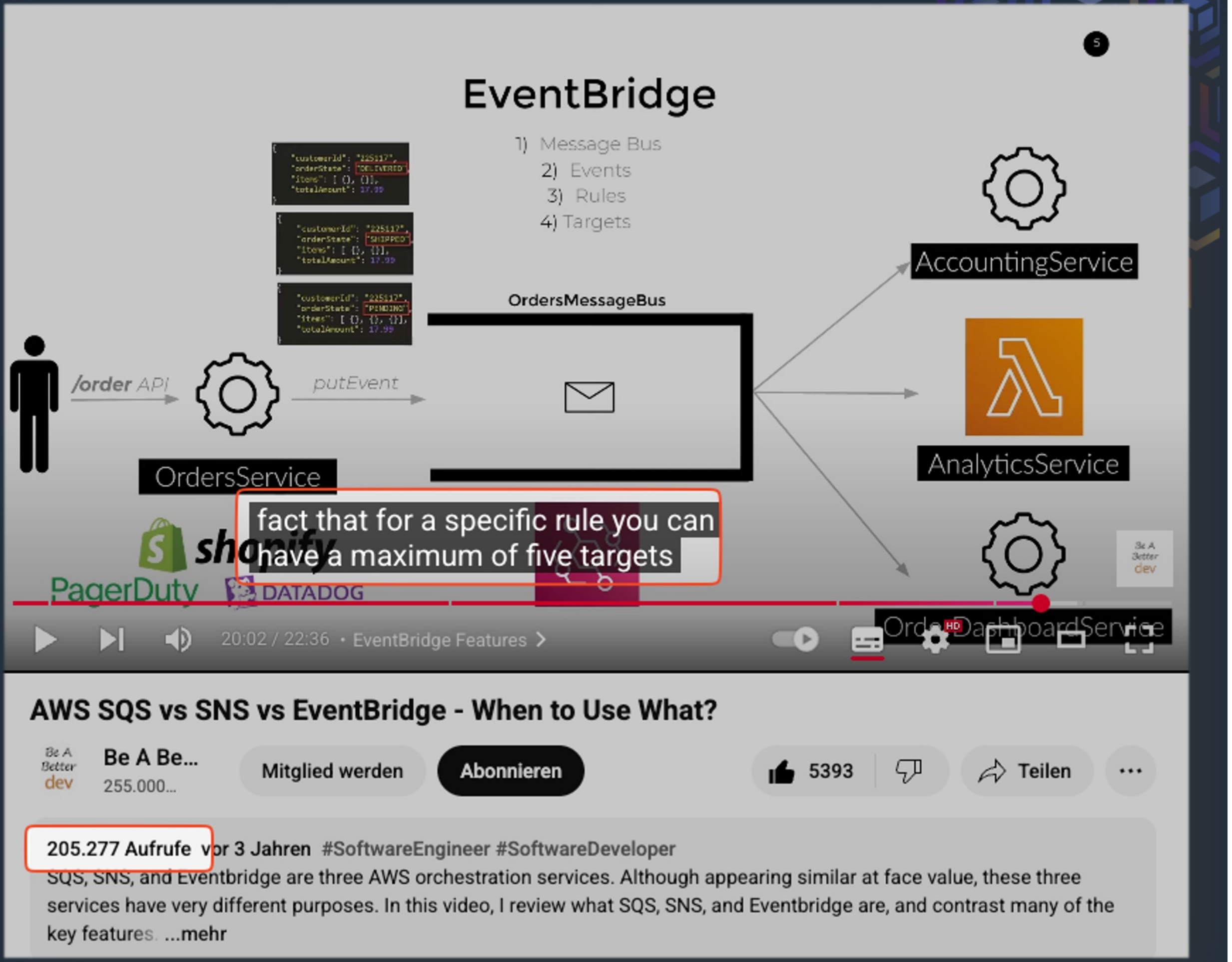
5 targets per rule



Targets per rule	Each supported Region: 5	No

aws COMMUNITY DAY

5 targets per rule



fact that for a specific rule you can have a maximum of five targets

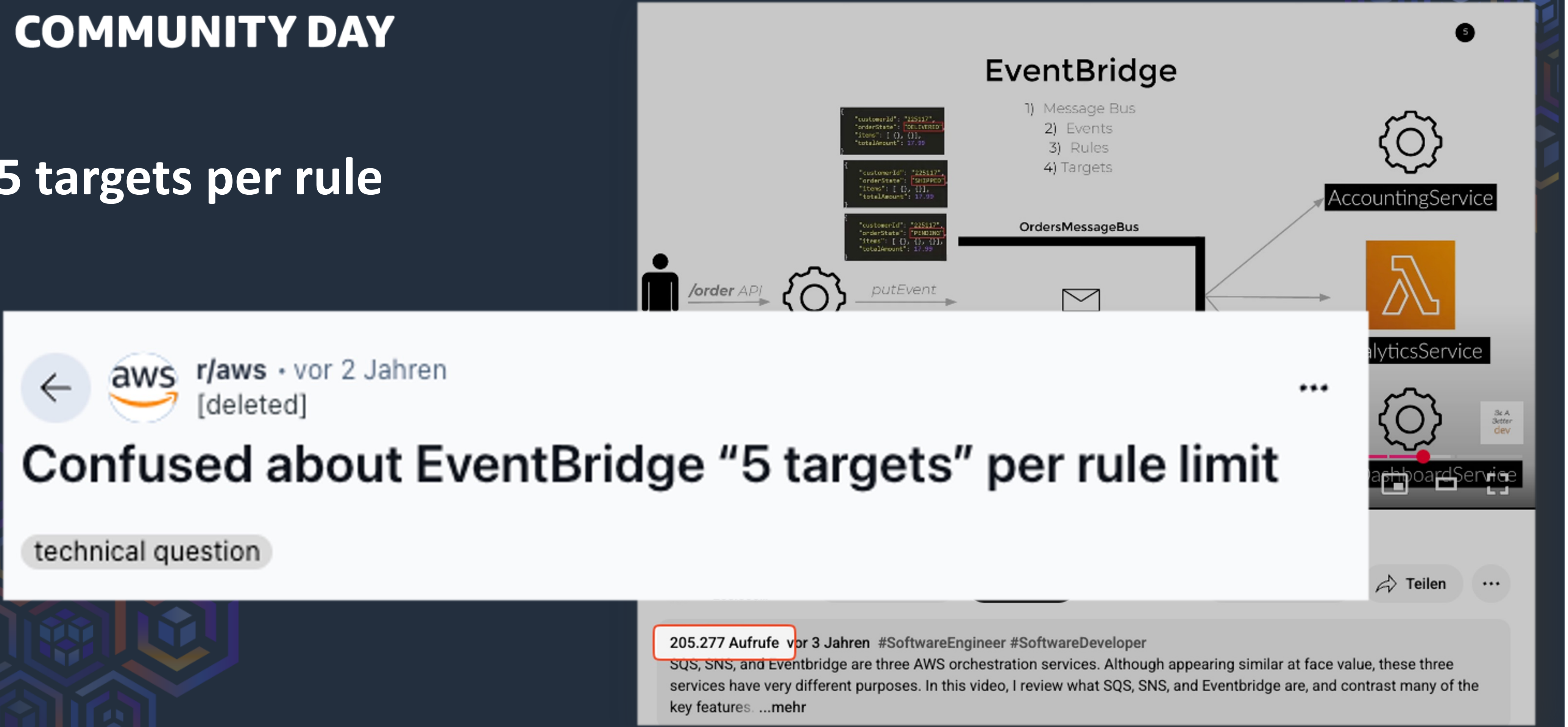
AWS SQS vs SNS vs EventBridge - When to Use What?

205.277 Aufrufe vor 3 Jahren #SoftwareEngineer #SoftwareDeveloper

SQS, SNS, and Eventbridge are three AWS orchestration services. Although appearing similar at face value, these three services have very different purposes. In this video, I review what SQS, SNS, and Eventbridge are, and contrast many of the key features. ...mehr

COMMUNITY DAY

5 targets per rule



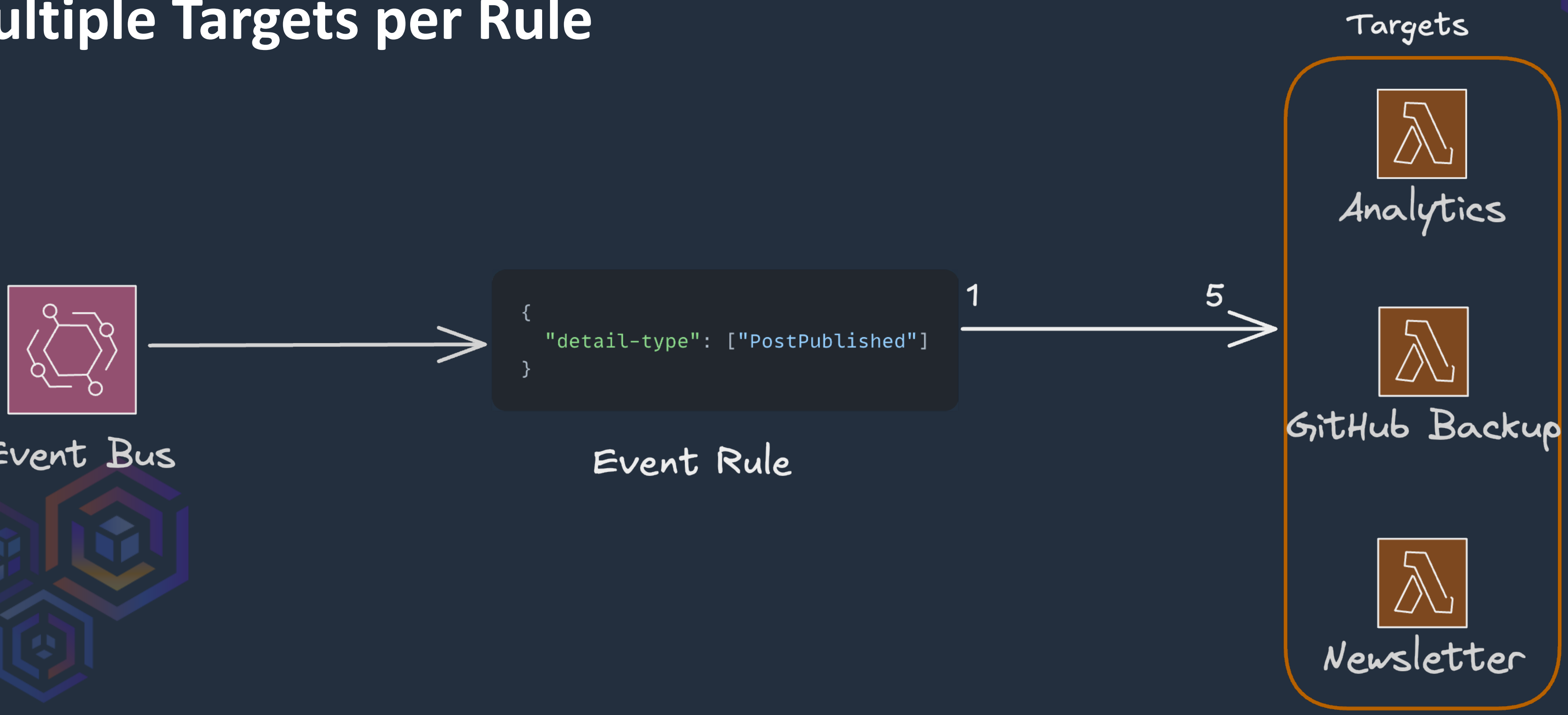
r/aws • vor 2 Jahren
[deleted]

Confused about EventBridge "5 targets" per rule limit

technical question

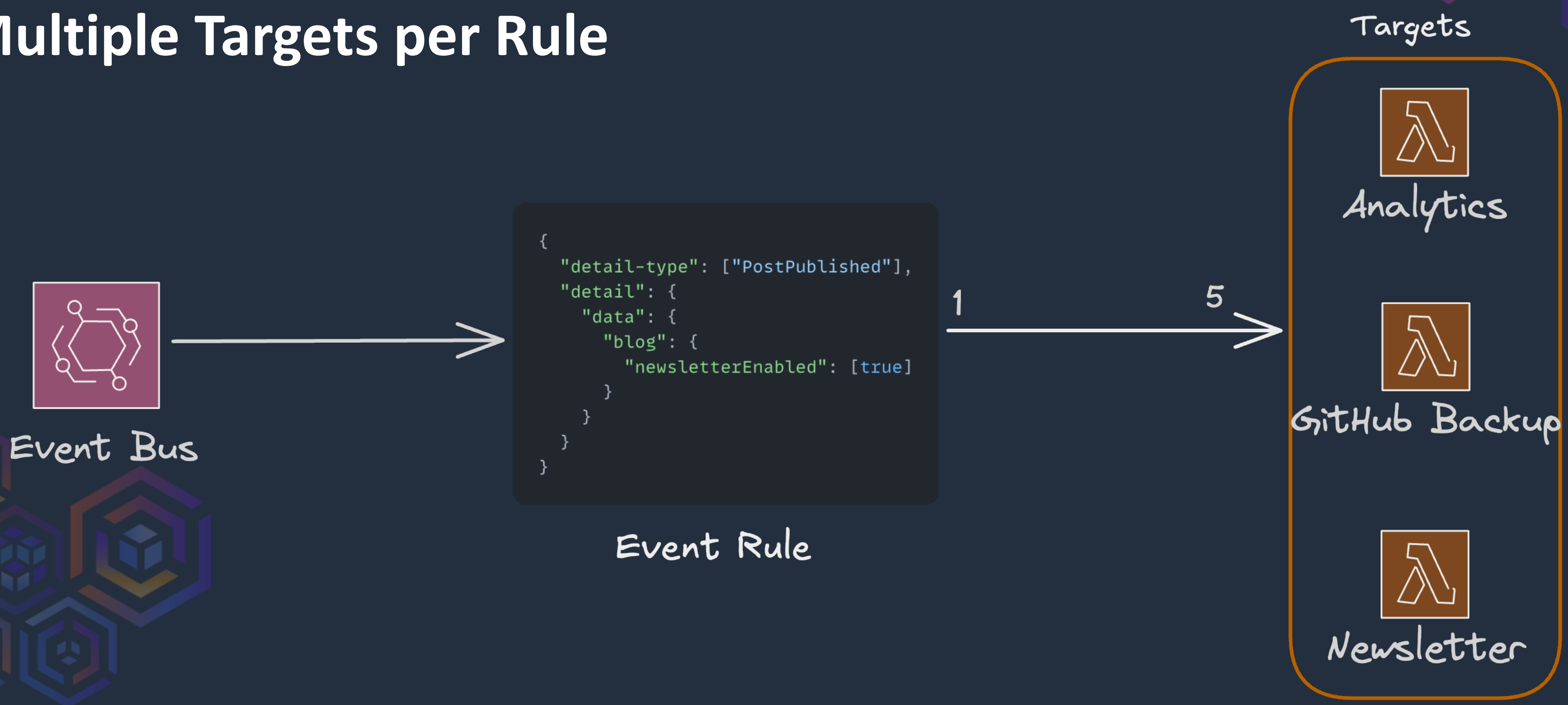
205.277 Aufrufe vor 3 Jahren #SoftwareEngineer #SoftwareDeveloper
SQS, SNS, and Eventbridge are three AWS orchestration services. Although appearing similar at face value, these three services have very different purposes. In this video, I review what SQS, SNS, and Eventbridge are, and contrast many of the key features. ...mehr

Multiple Targets per Rule



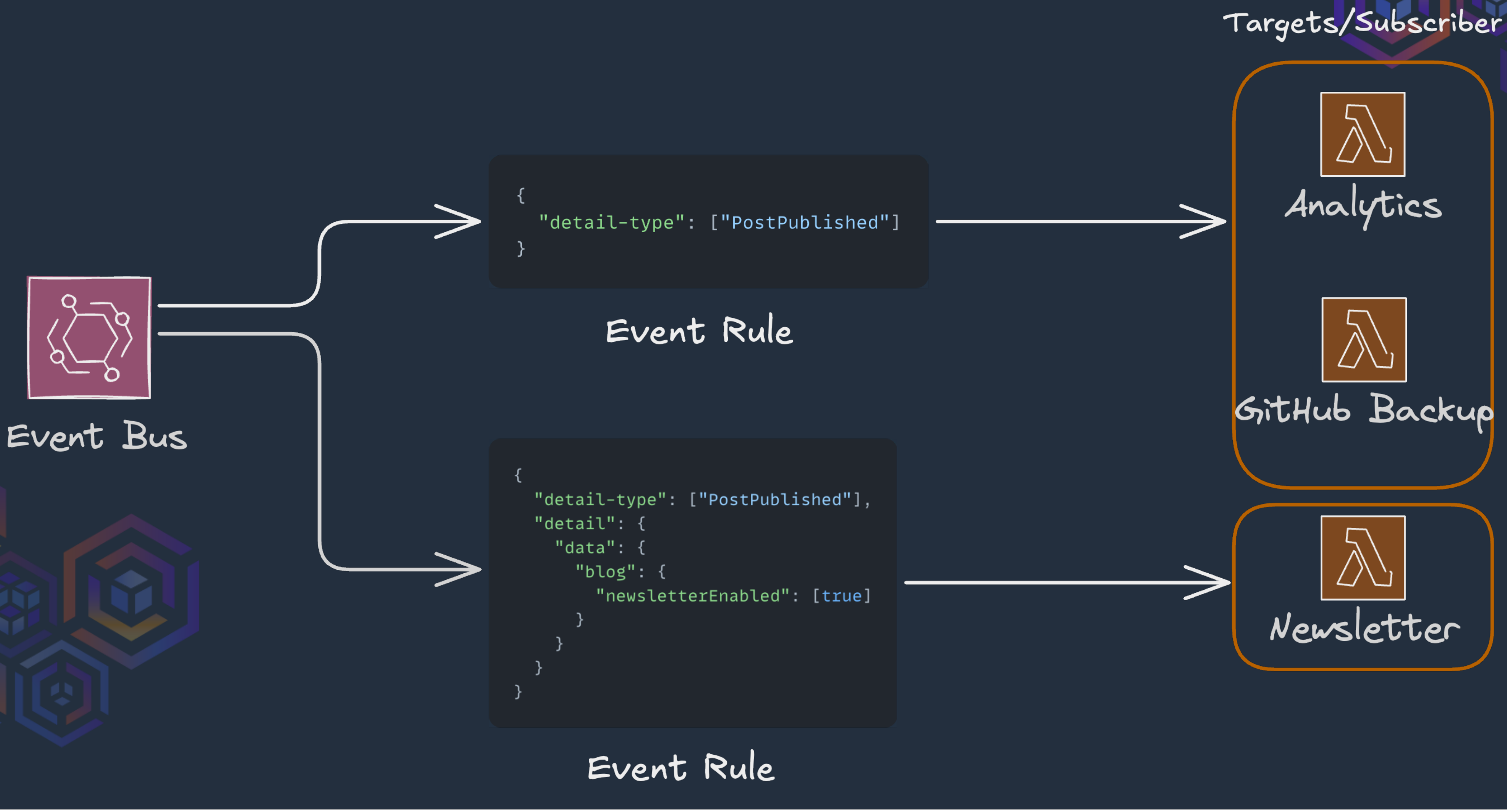
```
{
  "detail-type": ["PostPublished"]
}
```

Multiple Targets per Rule



```
{
  "detail-type": ["PostPublished"],
  "detail": {
    "data": {
      "blog": {
        "newsletterEnabled": [true]
      }
    }
  }
}
```

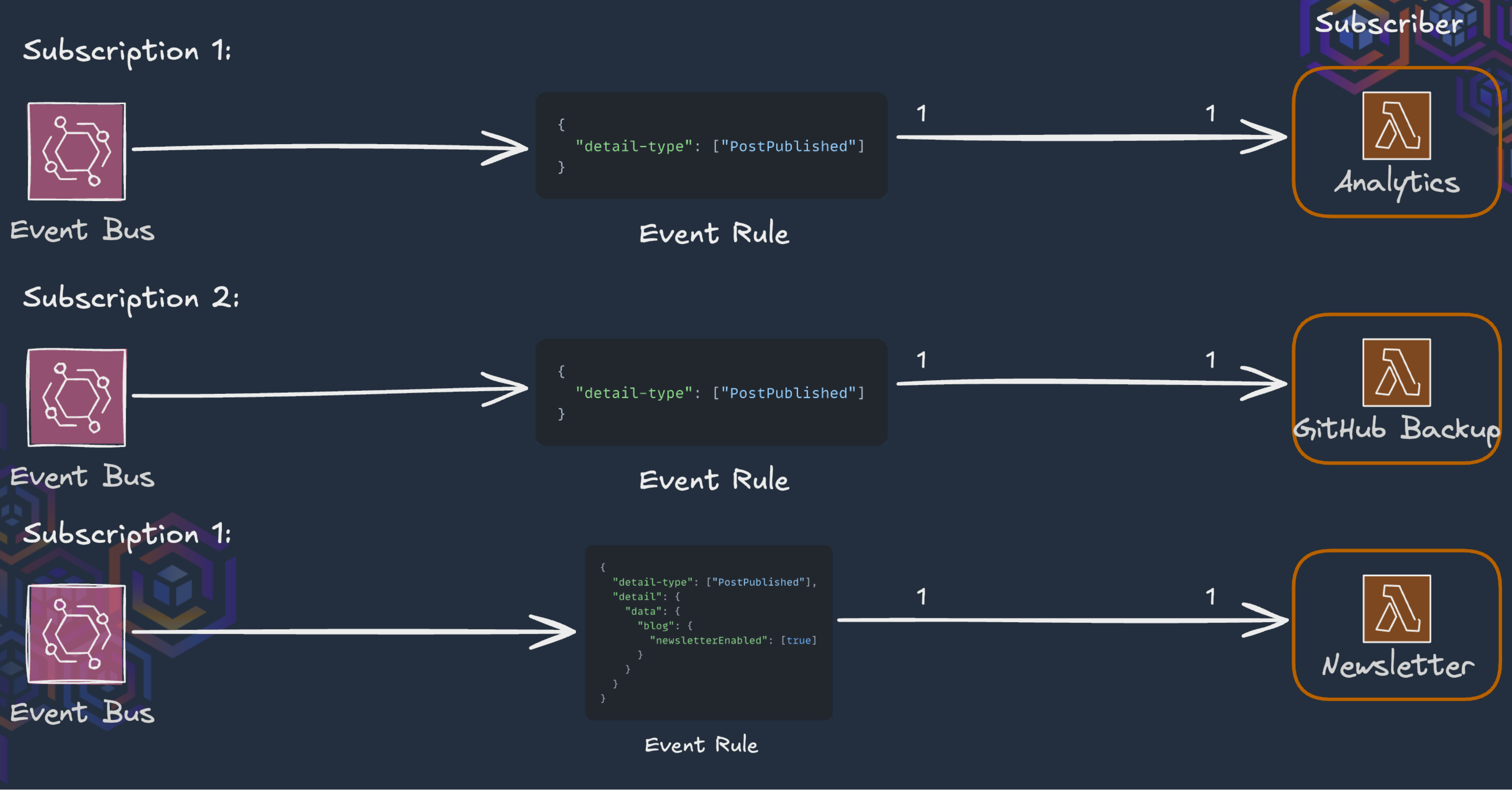
aws COMMUNITY DAY



```
{
  "detail-type": ["PostPublished"]
}
```

```
{
  "detail-type": ["PostPublished"],
  "detail": {
    "data": {
      "blog": {
        "newsletterEnabled": [true]
      }
    }
  }
}
```

aws COMMUNITY DAY



Subscription 1:

Event Bus

```
{
  "detail-type": ["PostPublished"]
}
```

Event Rule

Subscription 2:

Event Bus

```
{
  "detail-type": ["PostPublished"]
}
```

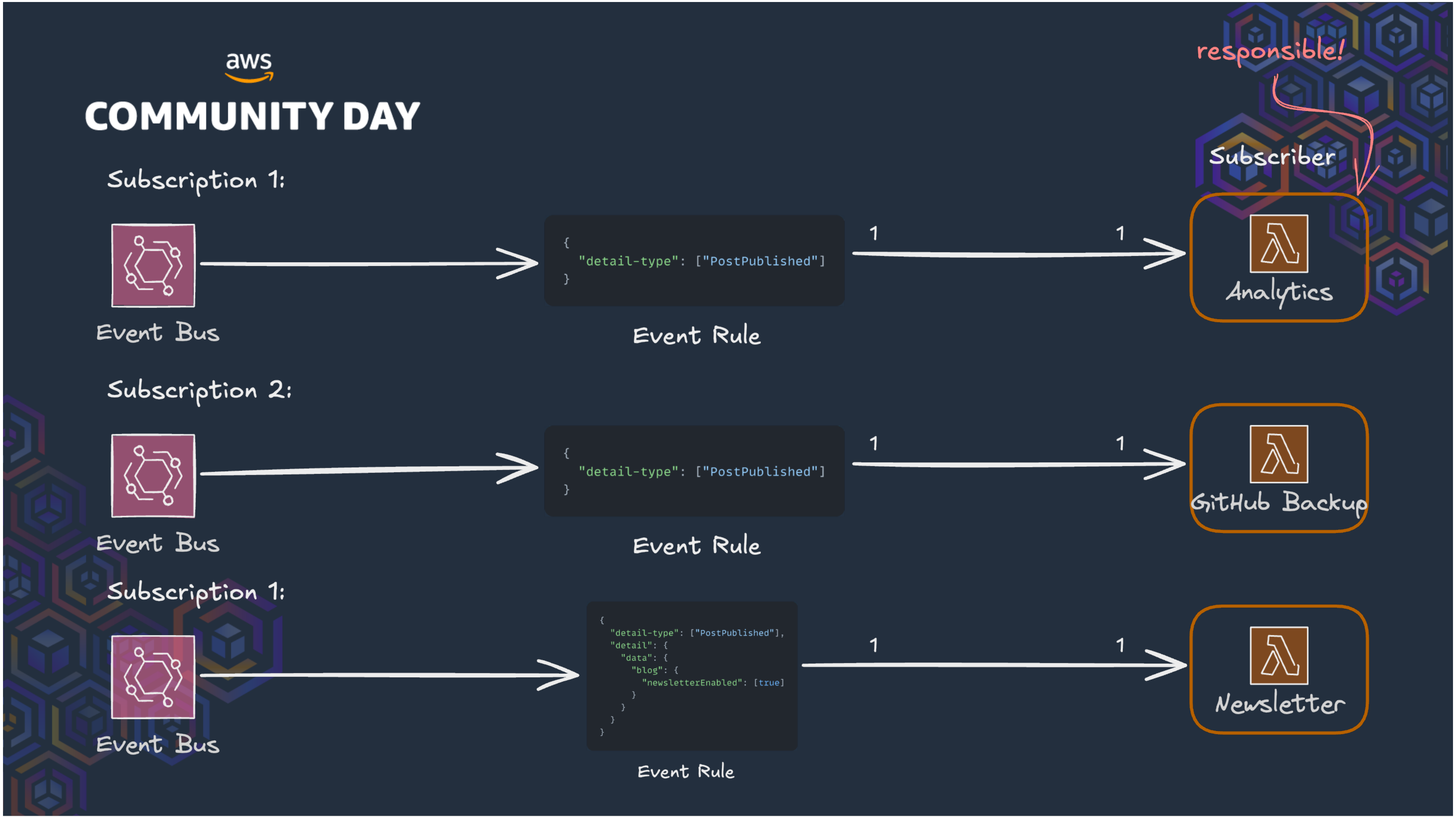
Event Rule

Subscription 1:

Event Bus

```
{
  "detail-type": ["PostPublished"],
  "detail": {
    "data": {
      "blog": {
        "newsletterEnabled": [true]
      }
    }
  }
}
```

Event Rule



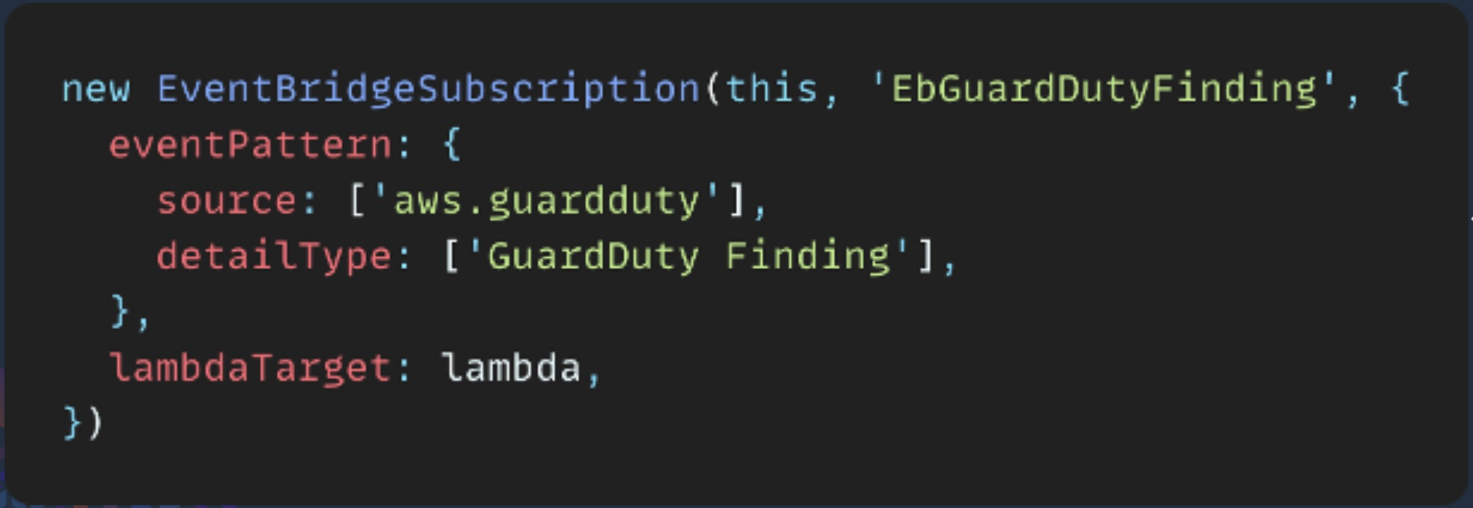
```
{
  "detail-type": ["PostPublished"]
}
```

```
{
  "detail-type": ["PostPublished"]
}
```

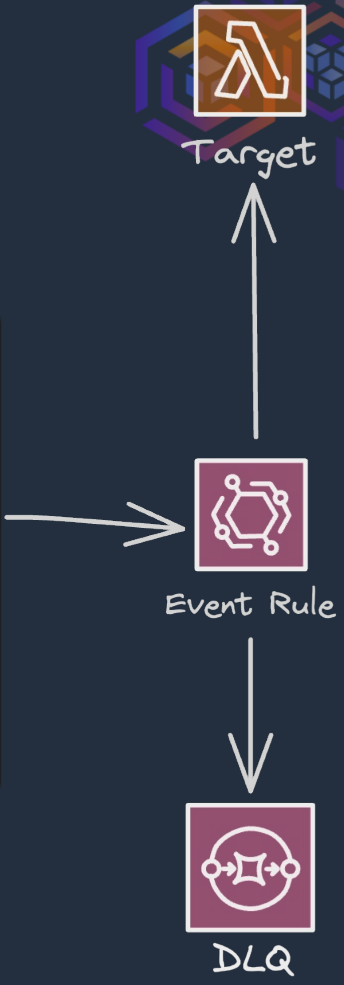
```
{
  "detail-type": ["PostPublished"],
  "detail": {
    "data": {
      "blog": {
        "newsletterEnabled": [true]
      }
    }
  }
}
```

aws COMMUNITY DAY

Subscription Pattern



```
new EventBridgeSubscription(this, 'EbGuardDutyFinding', {
  eventPattern: {
    source: ['aws.guardduty'],
    detailType: ['GuardDuty Finding'],
  },
  lambdaTarget: lambda,
})
```



“Hands-On” Agenda

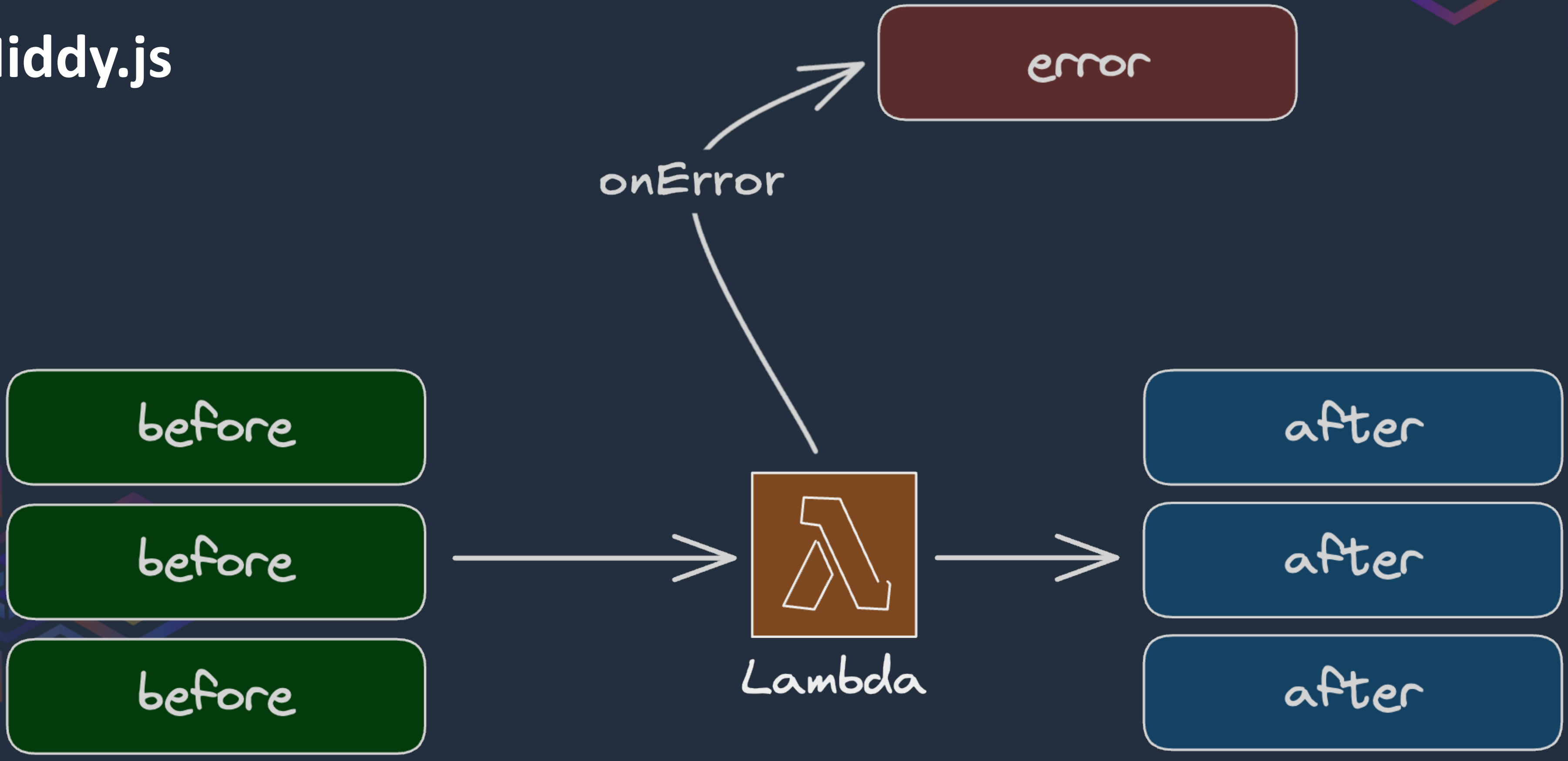
- Event validation
- Idempotency
- Error handling
- Testing

Middlewares!

Middy.js

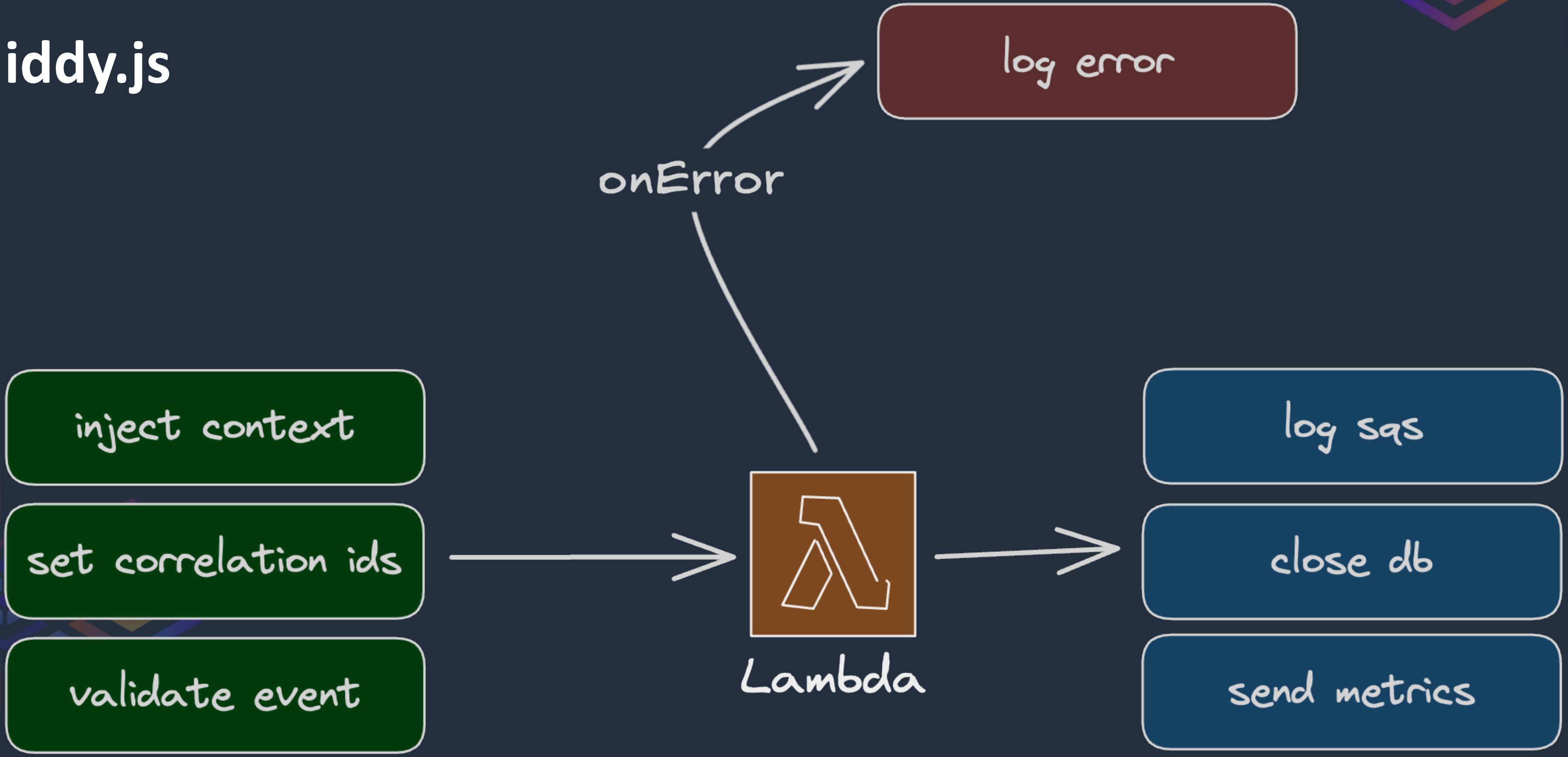
aws COMMUNITY DAY

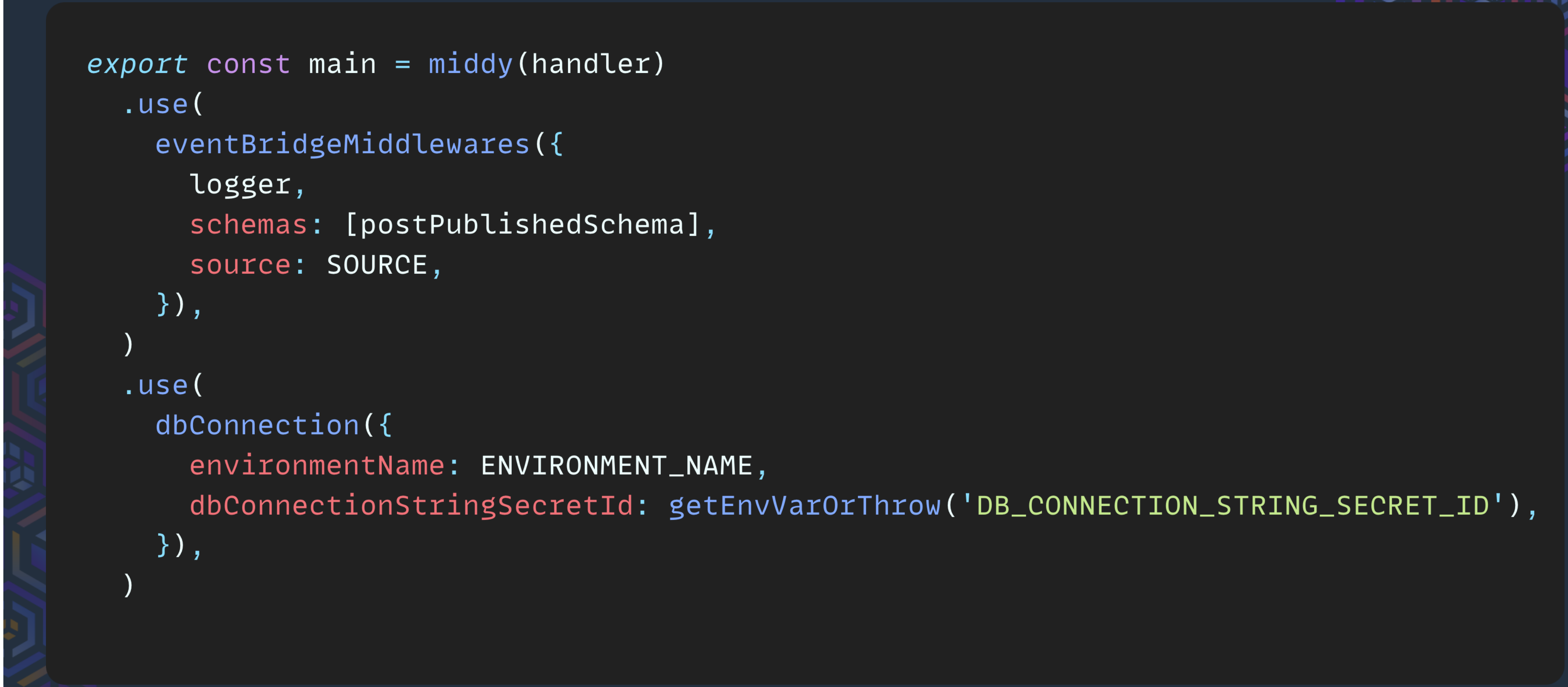
Middy.js



aws COMMUNITY DAY

Middy.js



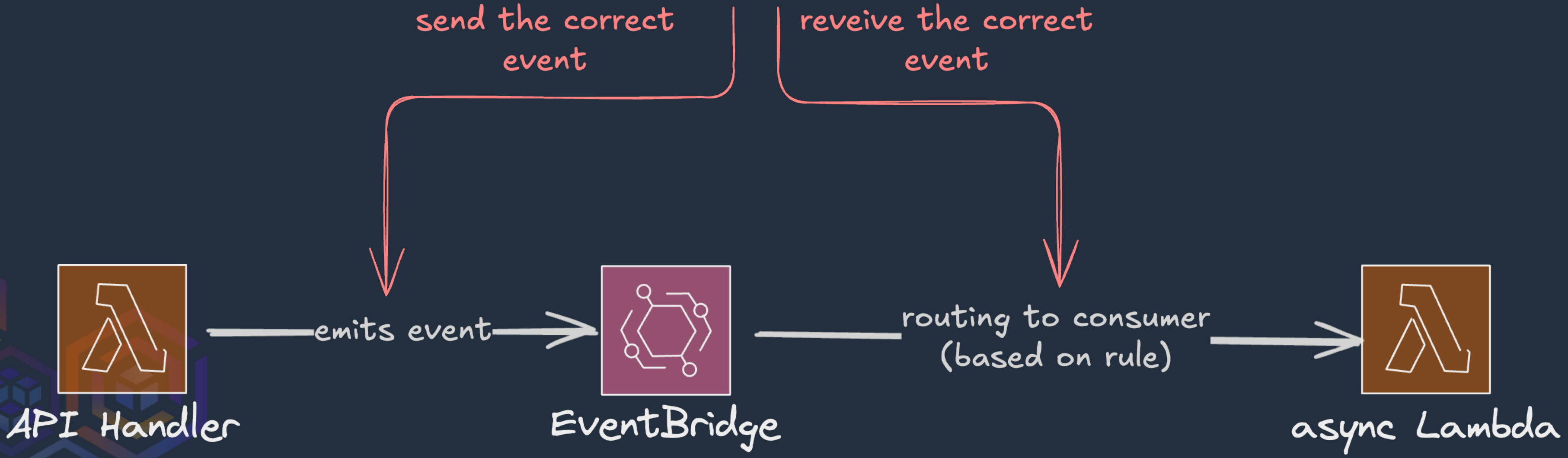


```
export const main = middy(handler)
  .use(
    eventBridgeMiddlewares({
      logger,
      schemas: [postPublishedSchema],
      source: SOURCE,
    }),
  )
  .use(
    dbConnection({
      environmentName: ENVIRONMENT_NAME,
      dbConnectionStringSecretId: getEnvVarOrThrow('DB_CONNECTION_STRING_SECRET_ID'),
    }),
  )
```

Validation

aws COMMUNITY DAY

Validation



send the correct event

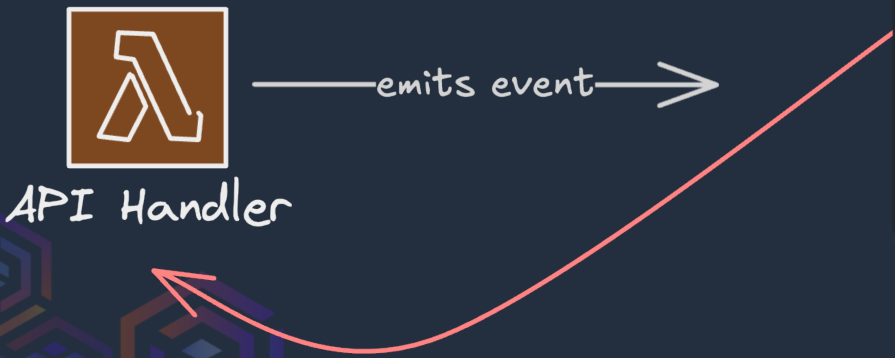
reveive the correct event

emits event

routing to consumer (based on rule)

aws

COMMUNITY DAY

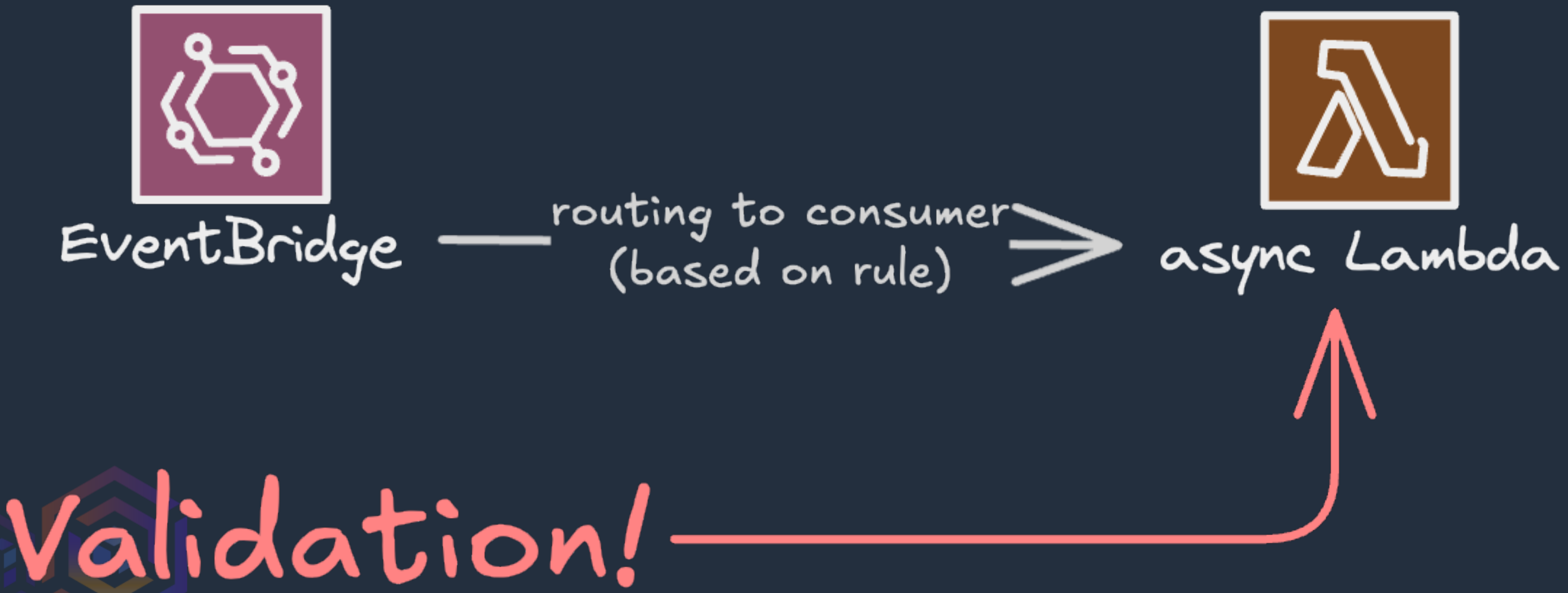


```
await PostEventEmitter.published({
  requestId: this.requestId,
  authenticatedUserId,
  post: {
    id: result._id
  },
  blog: {
    isNewsletterActivated: result.isNewsletterActivated,
    isGitHubBackupEnabled: result.isGitHubBackupEnabled,
  },
})
```

Validation!

aws

COMMUNITY DAY



EventBridge

routing to consumer
(based on rule)

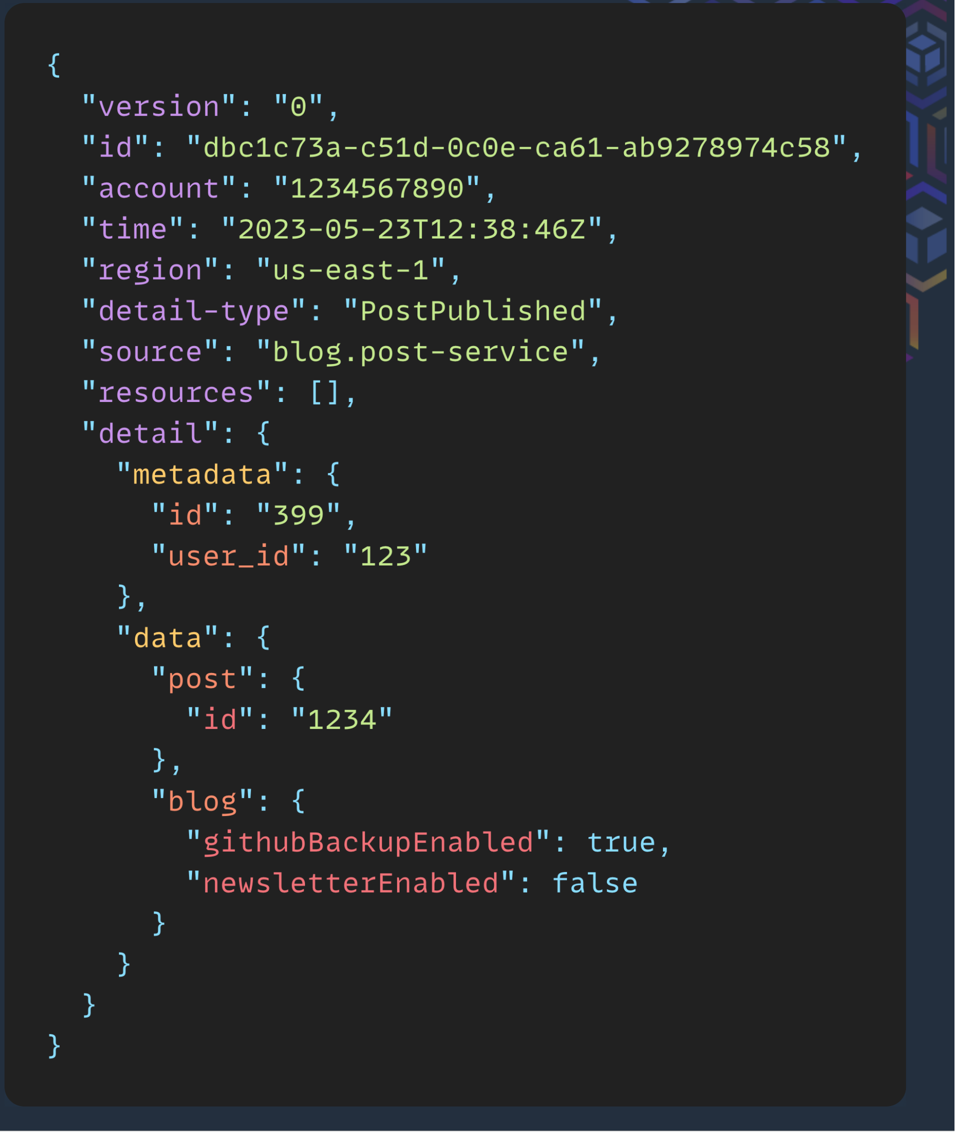
async Lambda

Validation!

COMMUNITY DAY

Three things we need to validate incoming events

1. **Event**
2. Schema
3. Middleware



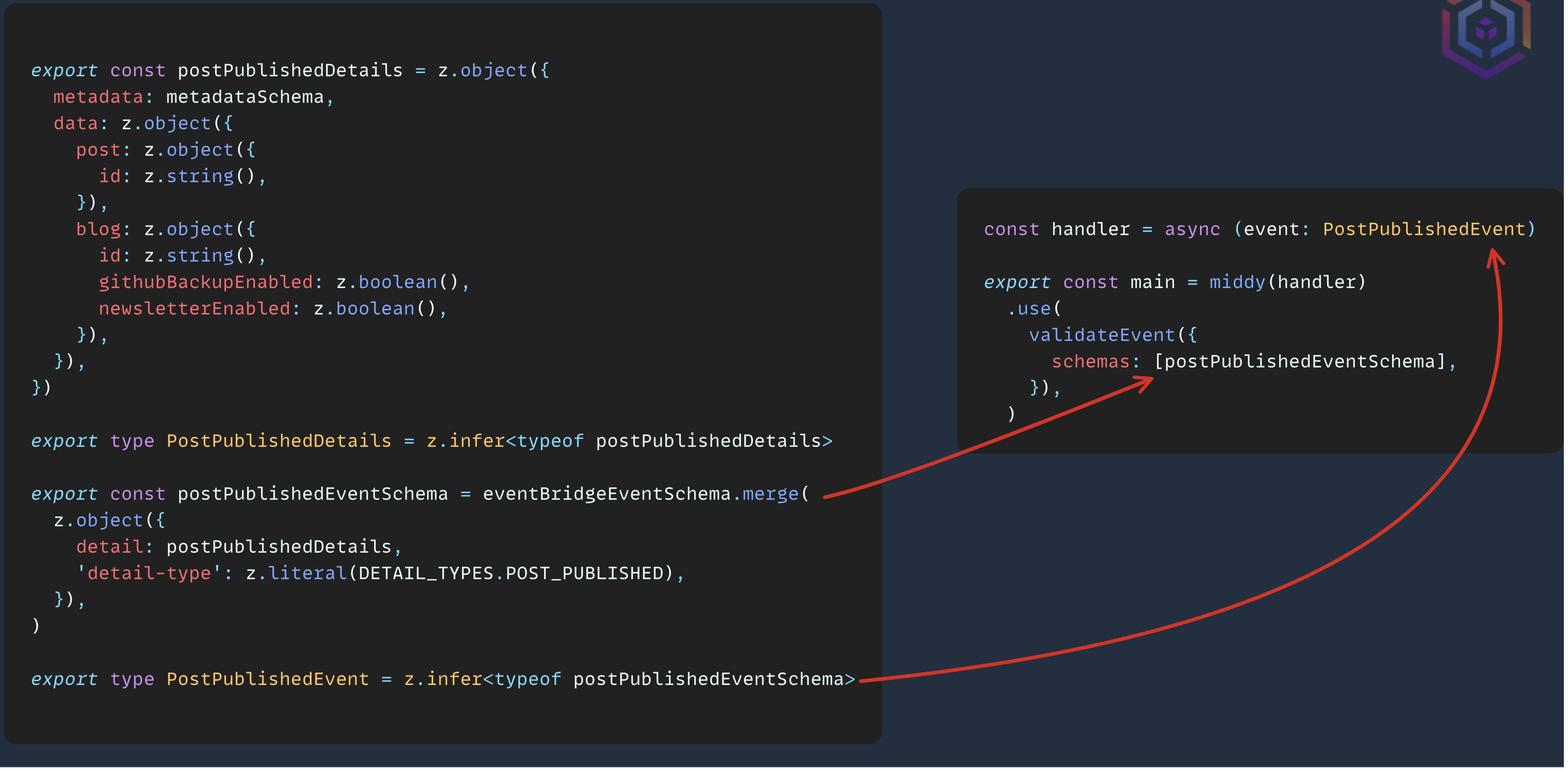
```json
{
  "version": "0",
  "id": "dbc1c73a-c51d-0c0e-ca61-ab9278974c58",
  "account": "1234567890",
  "time": "2023-05-23T12:38:46Z",
  "region": "us-east-1",
  "detail-type": "PostPublished",
  "source": "blog.post-service",
  "resources": [],
  "detail": {
    "metadata": {
      "id": "399",
      "user_id": "123"
    },
    "data": {
      "post": {
        "id": "1234"
      },
      "blog": {
        "githubBackupEnabled": true,
        "newsletterEnabled": false
      }
    }
  }
}
```

aws

COMMUNITY DAY

Three things we need to validate incoming events

1. Event
2. **Schema**
3. Middleware



```
export const postPublishedDetails = z.object({
  metadata: metadataSchema,
  data: z.object({
    post: z.object({
      id: z.string(),
    }),
    blog: z.object({
      id: z.string(),
      githubBackupEnabled: z.boolean(),
      newsletterEnabled: z.boolean(),
    }),
  }),
})

export type PostPublishedDetails = z.infer<typeof postPublishedDetails>

export const postPublishedEventSchema = eventBridgeEventSchema.merge(
  z.object({
    detail: postPublishedDetails,
    'detail-type': z.literal(DETAIL_TYPES.POST_PUBLISHED),
  }),
)

export type PostPublishedEvent = z.infer<typeof postPublishedEventSchema>
```

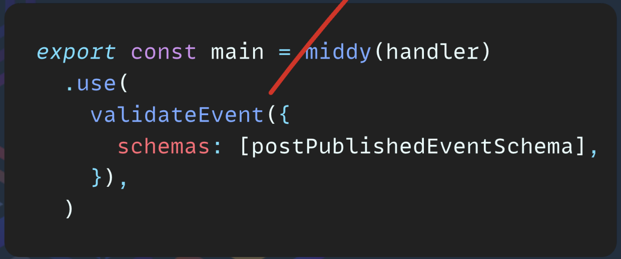
```
const handler = async (event: PostPublishedEvent)

export const main = middy(handler)
  .use(
    validateEvent({
      schemas: [postPublishedEventSchema],
    }),
  )
```

aws

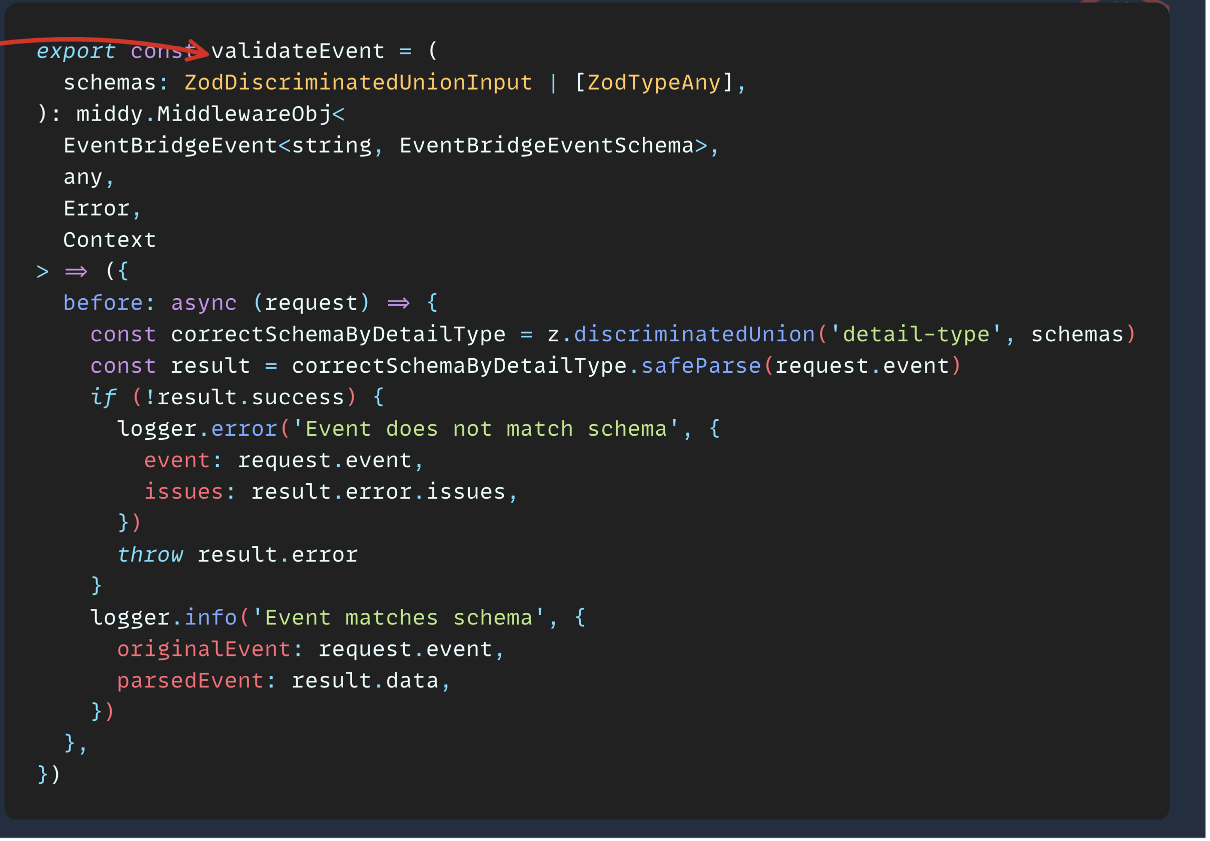
COMMUNITY DAY

Three things we need to validate incoming events

1. Event
2. Schema
3. **Middleware**



```
export const main = middy(handler)
  .use(
    validateEvent({
      schemas: [postPublishedEventSchema],
    }),
  )
```

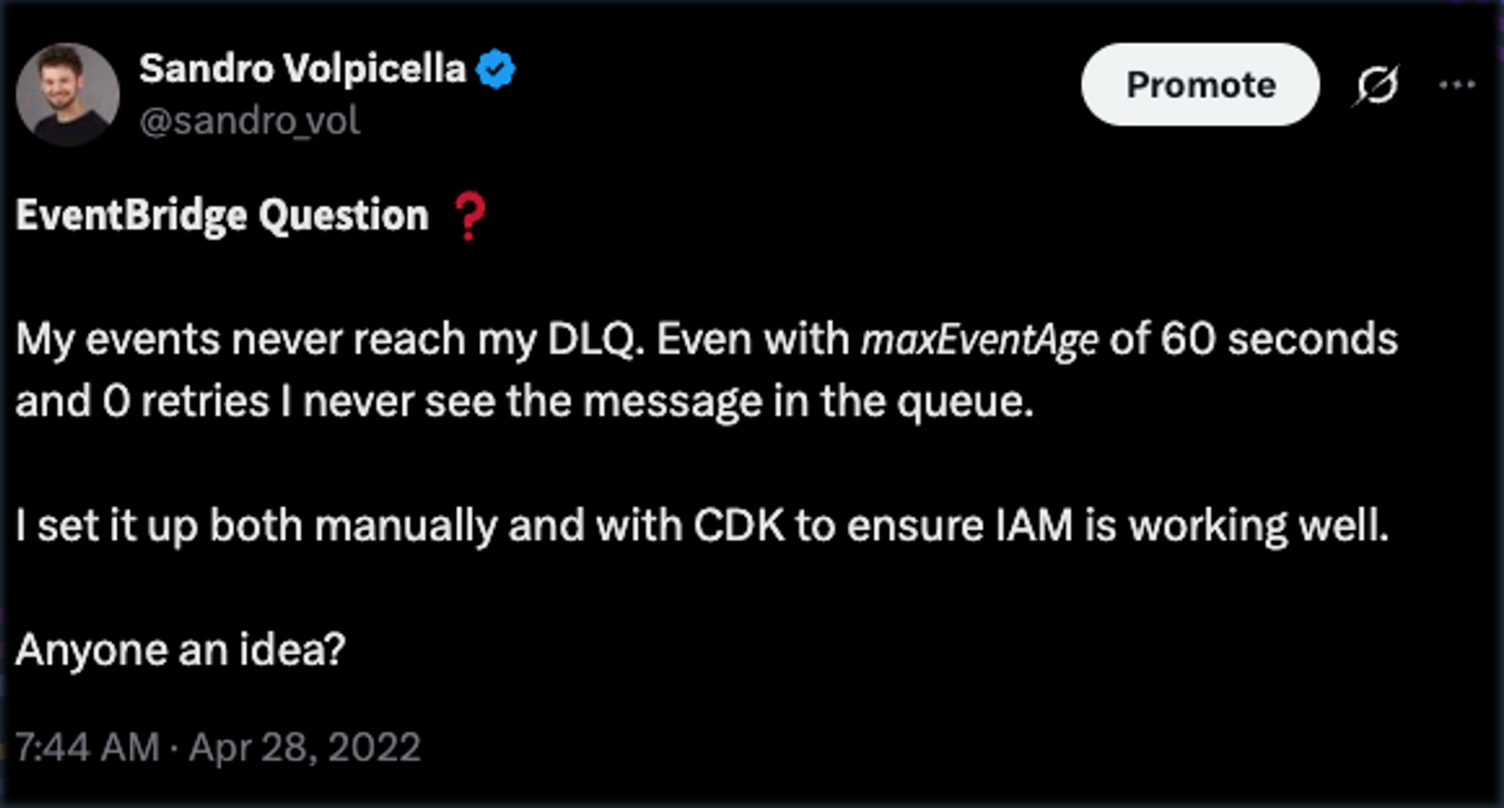


```
export const validateEvent = (
  schemas: ZodDiscriminatedUnionInput | [ZodTypeAny],
): middy.MiddlewareObj<
  EventBridgeEvent<string, EventBridgeEventSchema>,
  any,
  Error,
  Context
> => ({
  before: async (request) => {
    const correctSchemaByDetailType = z.discriminatedUnion('detail-type', schemas)
    const result = correctSchemaByDetailType.safeParse(request.event)
    if (!result.success) {
      logger.error('Event does not match schema', {
        event: request.event,
        issues: result.error.issues,
      })
      throw result.error
    }
    logger.info('Event matches schema', {
      originalEvent: request.event,
      parsedEvent: result.data,
    })
  },
})
```

Retry & Error Handling

aws

COMMUNITY DAY



Sandro Volpicella @sandro_vol

EventBridge Question ❓

My events never reach my DLQ. Even with *maxEventAge* of 60 seconds and 0 retries I never see the message in the queue.

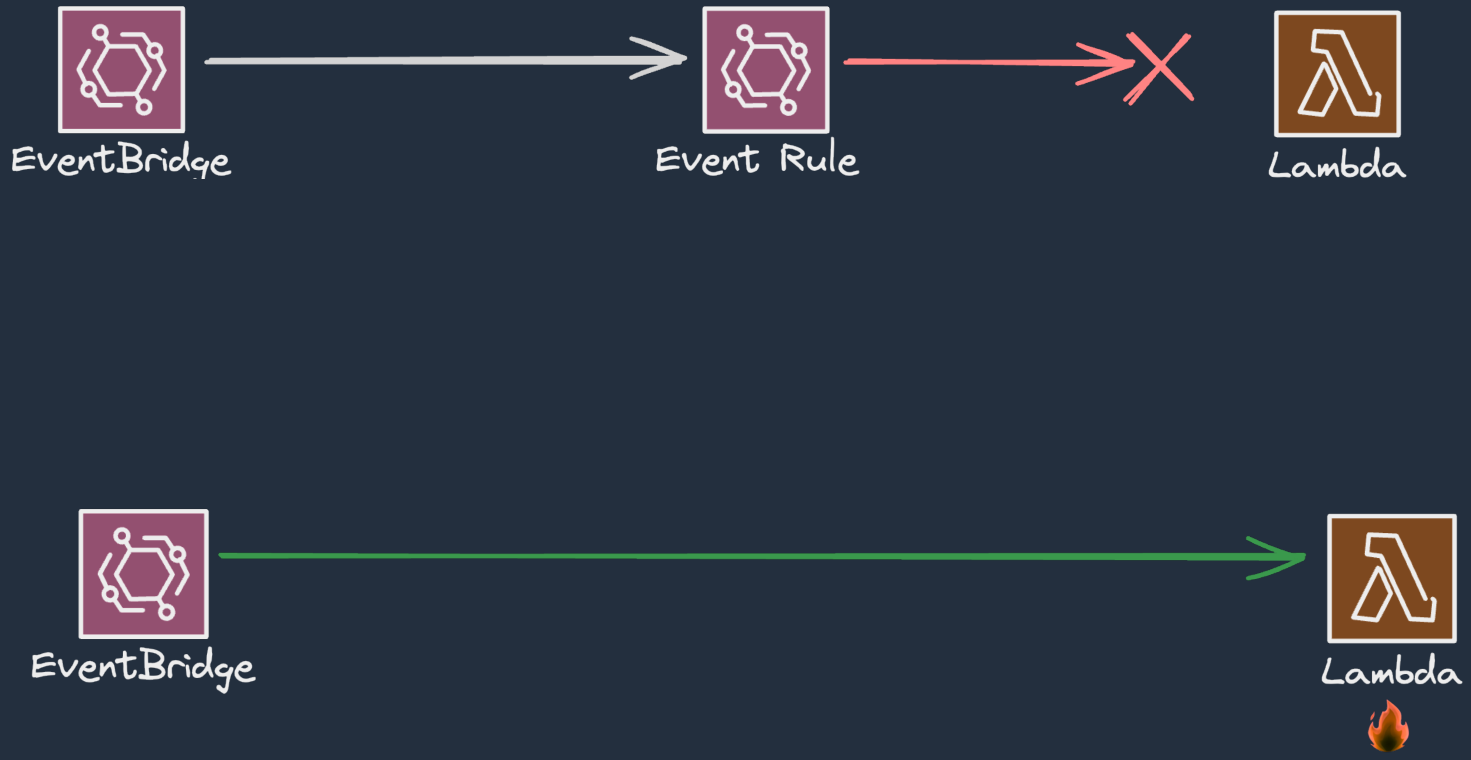
I set it up both manually and with CDK to ensure IAM is working well.

Anyone an idea?

7:44 AM · Apr 28, 2022

aws

COMMUNITY DAY



"Server" errors
(or delivery errors)

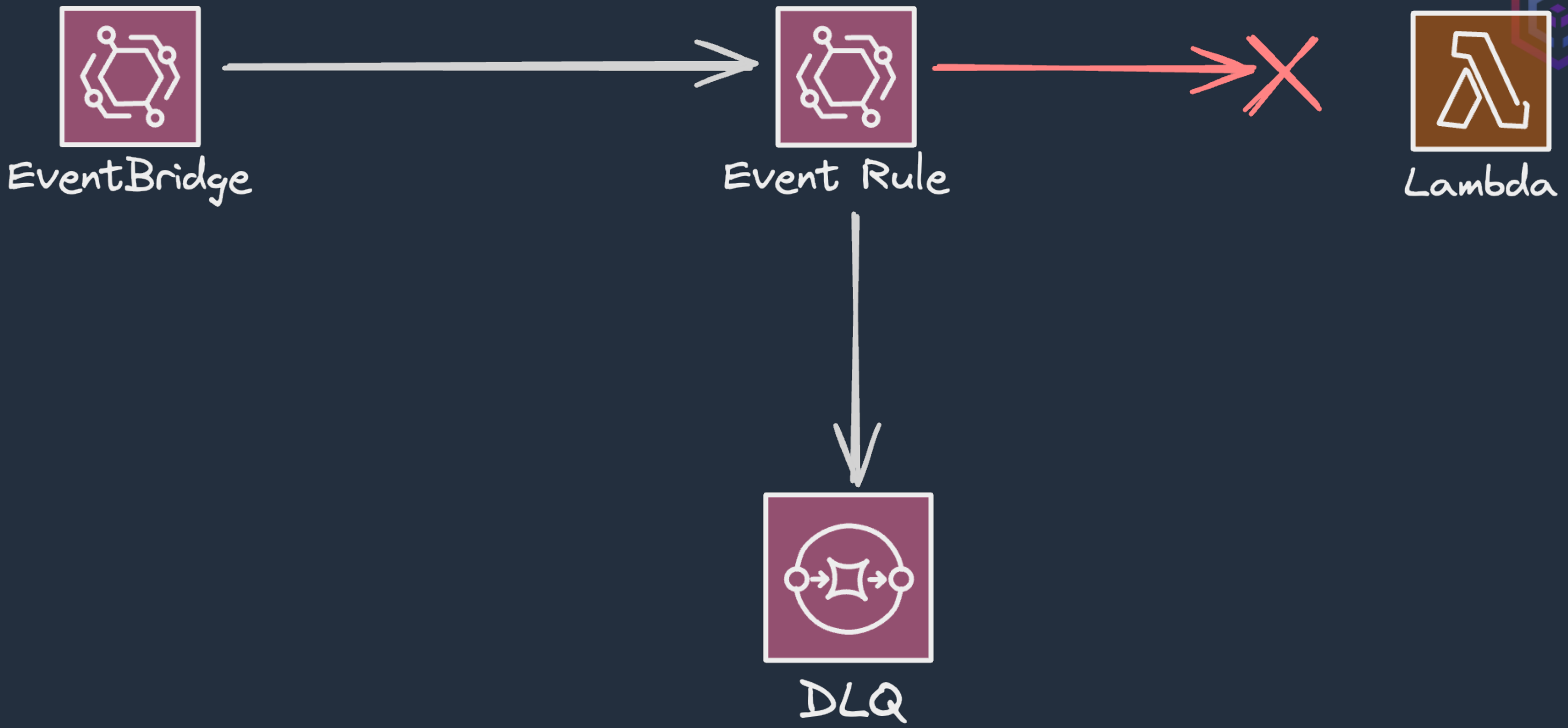
"Consumer" Errors

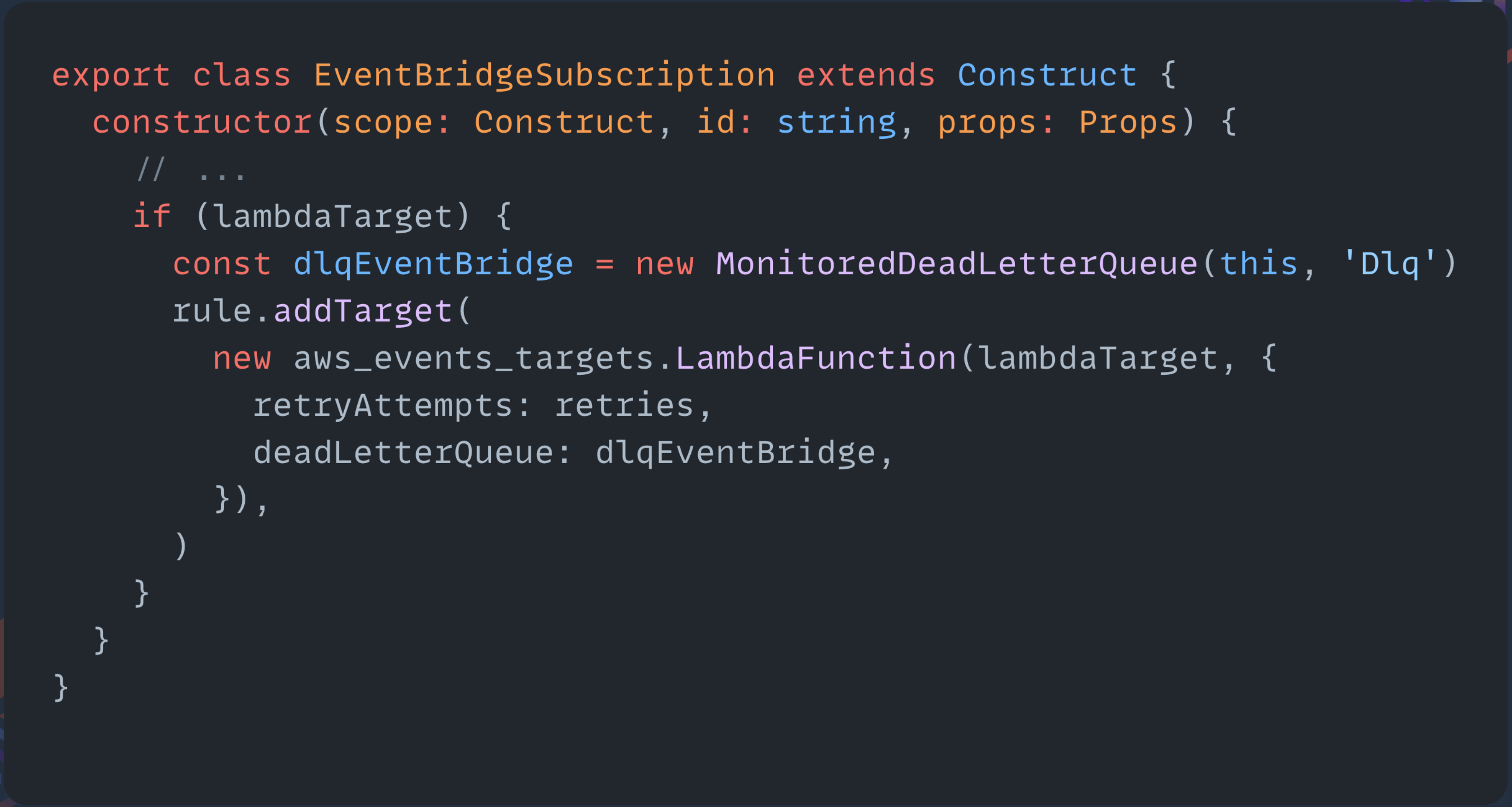
aws

COMMUNITY DAY

Server Errors

"Server" errors
(or delivery errors)





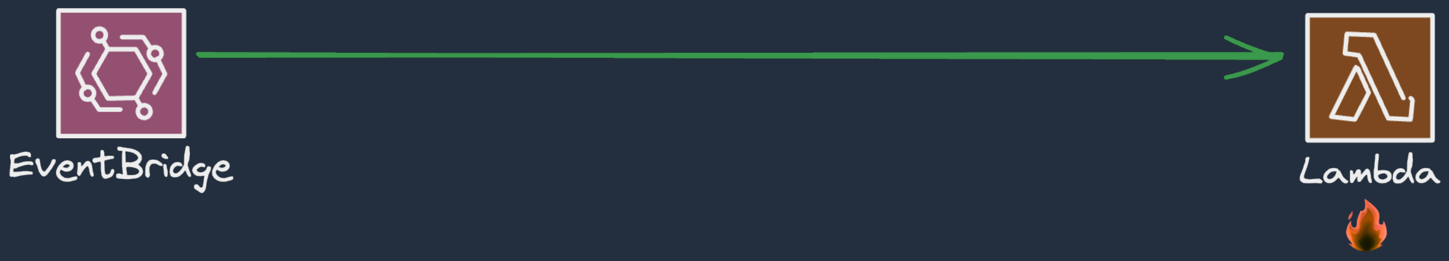
```
export class EventBridgeSubscription extends Construct {
  constructor(scope: Construct, id: string, props: Props) {
    // ...
    if (lambdaTarget) {
      const dlqEventBridge = new MonitoredDeadLetterQueue(this, 'Dlq')
      rule.addTarget(
        new aws_events_targets.LambdaFunction(lambdaTarget, {
          retryAttempts: retries,
          deadLetterQueue: dlqEventBridge,
        }),
      )
    }
  }
}
```

aws

COMMUNITY DAY

Retry & Errors

"Consumer" Errors



Retry & Errors – Lambda **Sync** vs. **Async**

Sync:

- Receiver waits for response
- Caller controls error handling
- API / Blocking

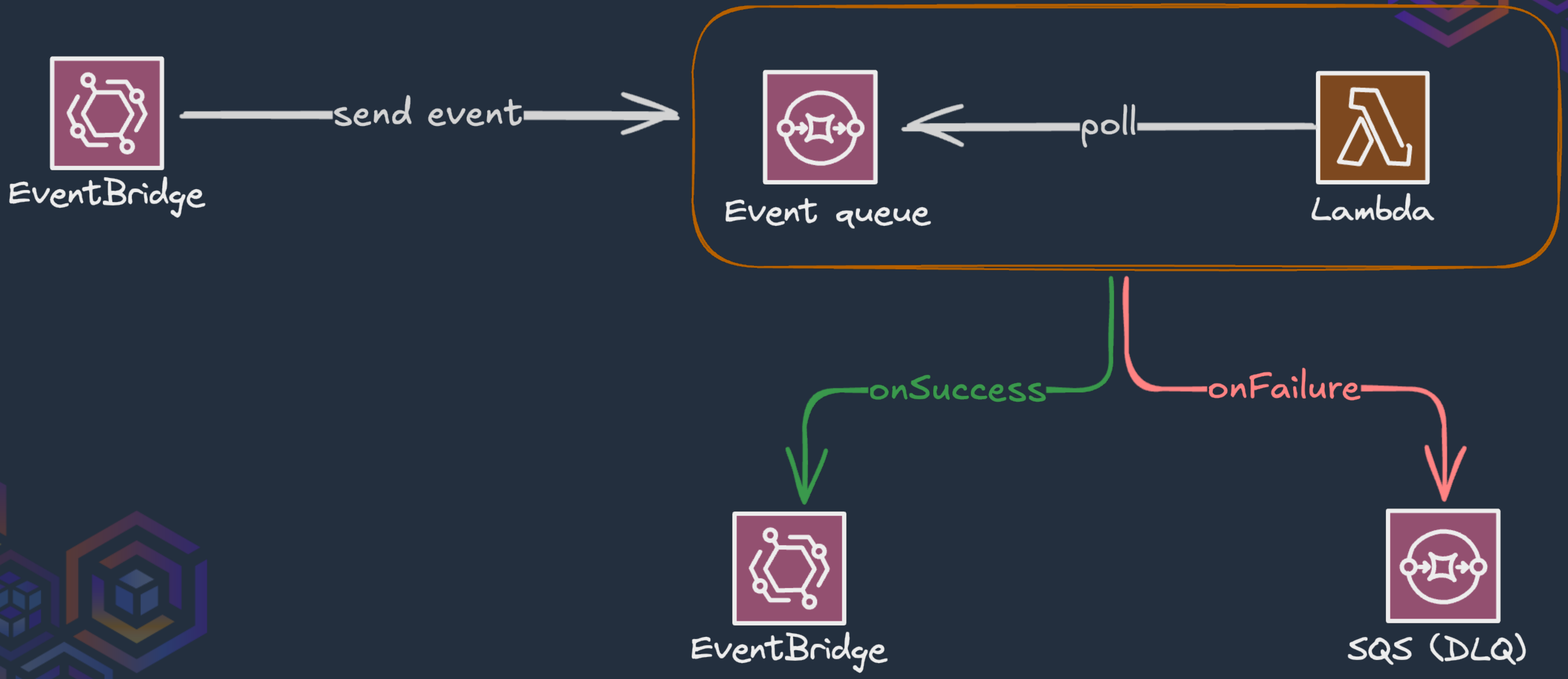
Async:

- Fire & Forget
- Consumer handles errors
- SQS & Lambda in one
- Retries
- Destinations

Service	sync	async
API Gateway	X	
(ESM) SQS	X	
Step Functions	X	X
Lambda	X	X
DynamoDB		X
EventBridge		X

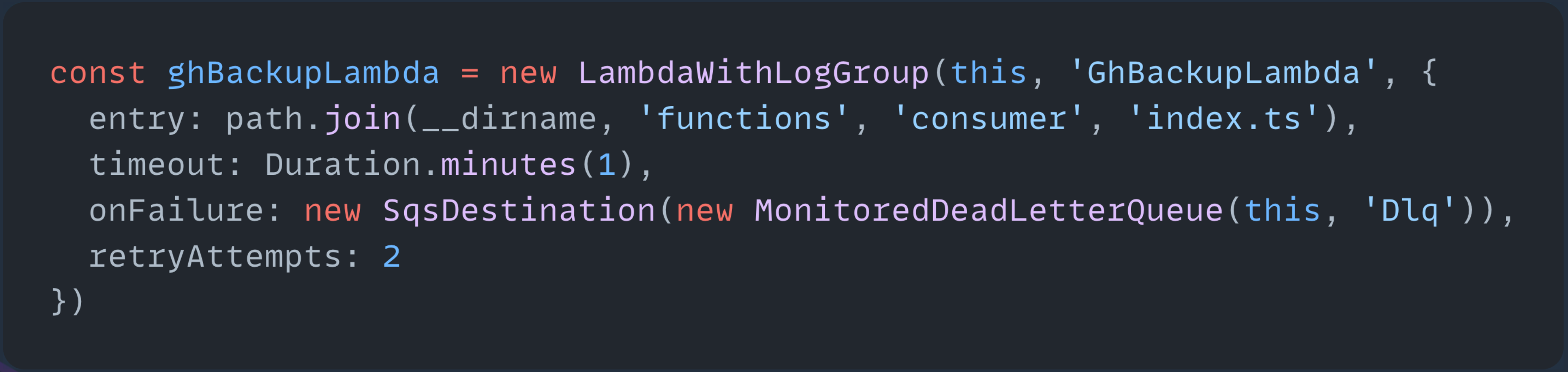
aws

COMMUNITY DAY



aws

COMMUNITY DAY



```
const ghBackupLambda = new LambdaWithLogGroup(this, 'GhBackupLambda', {
  entry: path.join(__dirname, 'functions', 'consumer', 'index.ts'),
  timeout: Duration.minutes(1),
  onFailure: new SqsDestination(new MonitoredDeadLetterQueue(this, 'Dlq')),
  retryAttempts: 2
})
```

Archive & Replay

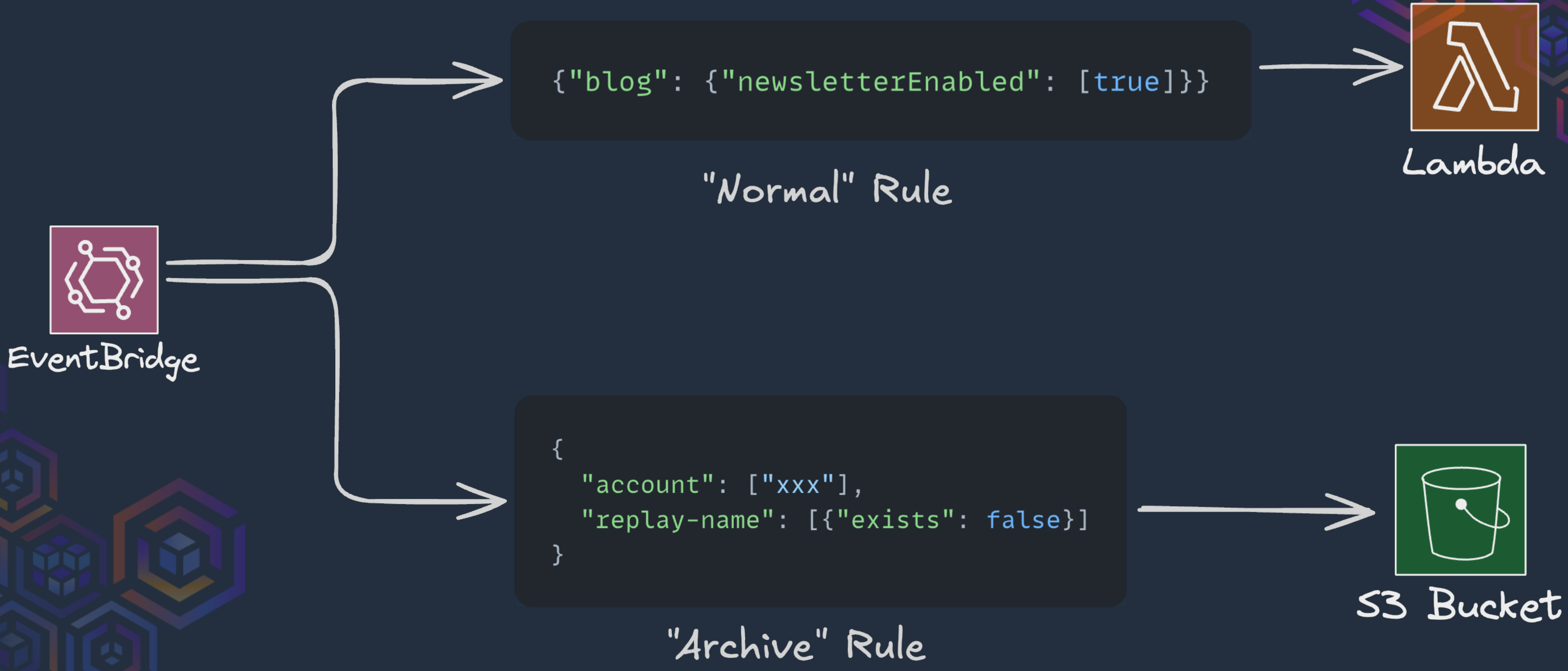
> “Everything fails, all the time.”

Werner Vogels – CTO Amazon

**there is no AWS slide without this quote*

aws

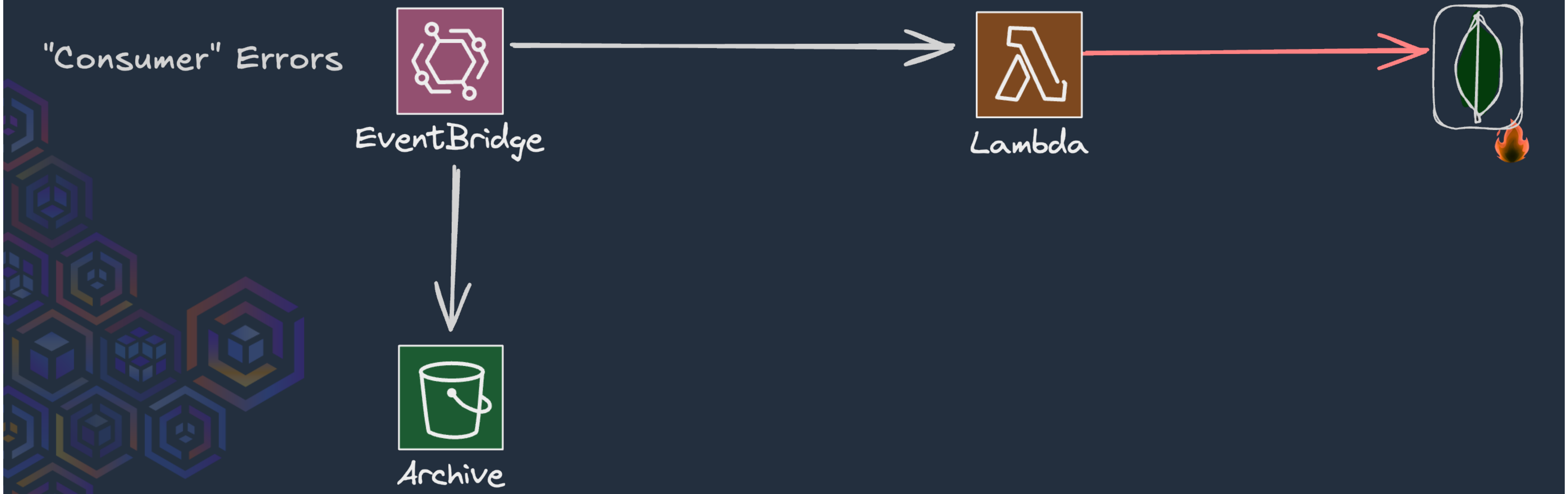
COMMUNITY DAY



```
{"blog": {"newsletterEnabled": [true]}}
```

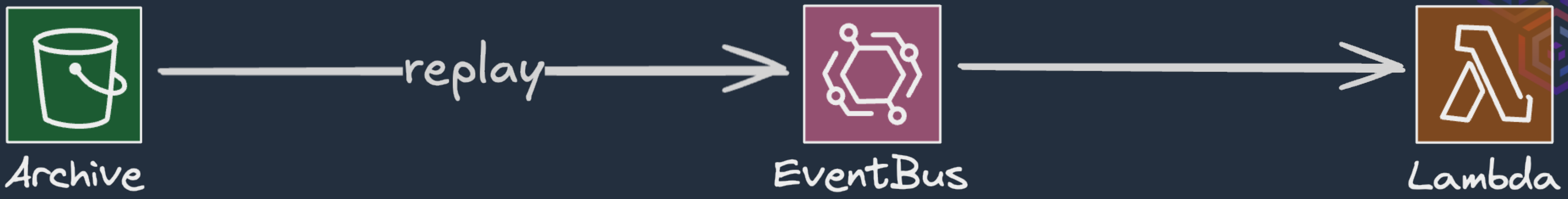
```
{
  "account": ["xxx"],
  "replay-name": [{"exists": false}]
}
```

aws COMMUNITY DAY



aws

COMMUNITY DAY

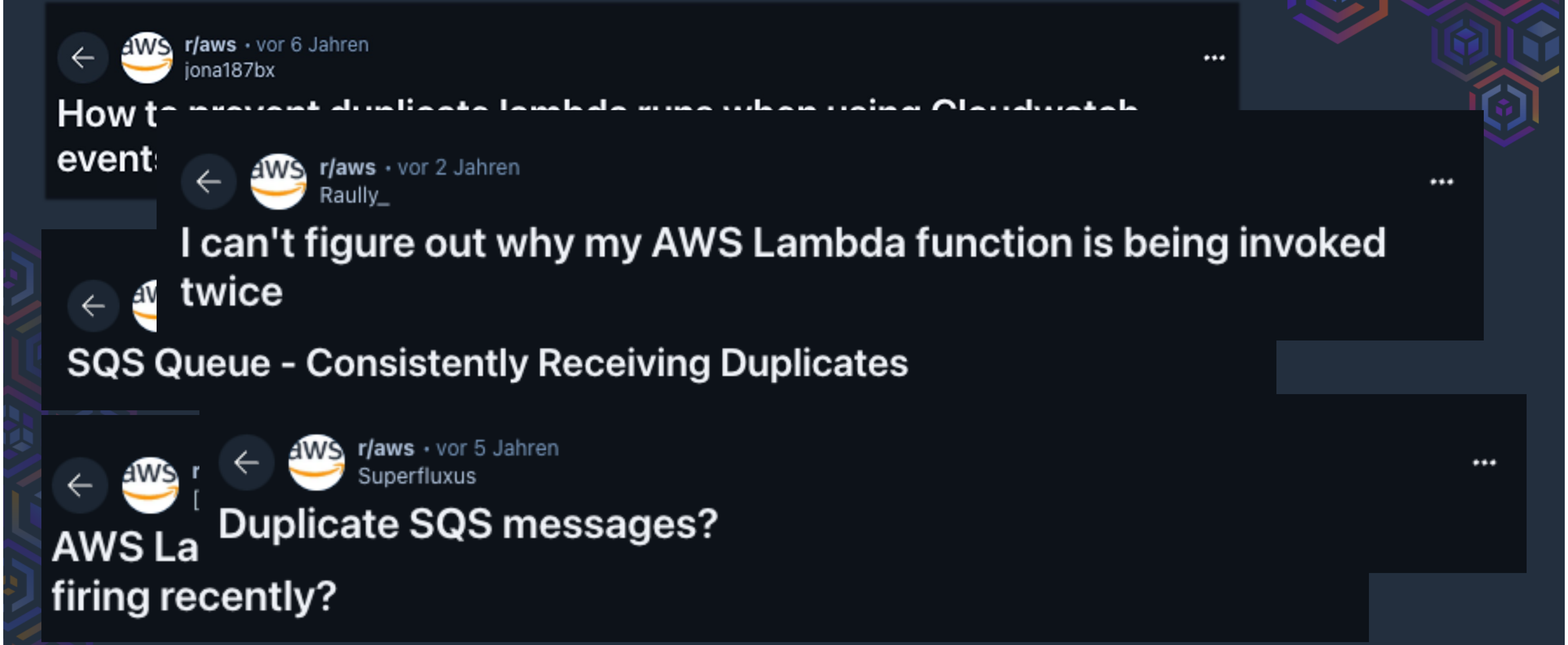


Replay from 10:10:00 - 10:20:00

- How do you know you haven't handled some events already?
- This is where idempotency comes in

Idempotency

(at-least-once processing / deduplication)



r/aws • vor 6 Jahren
jona187bx

How to prevent duplicate lambda runs when using Cloudwatch event

r/aws • vor 2 Jahren
Raully_

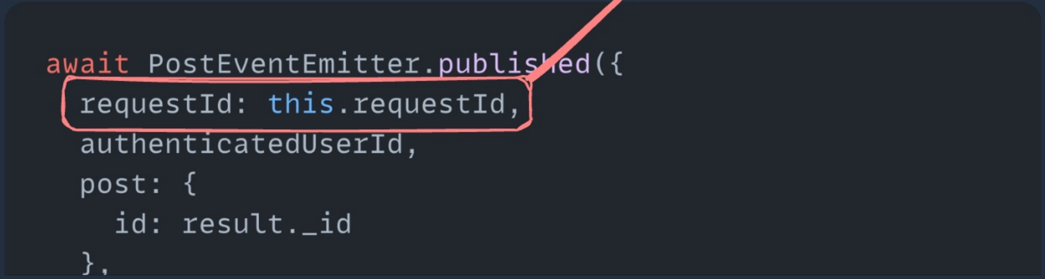
I can't figure out why my AWS Lambda function is being invoked twice

SQS Queue - Consistently Receiving Duplicates

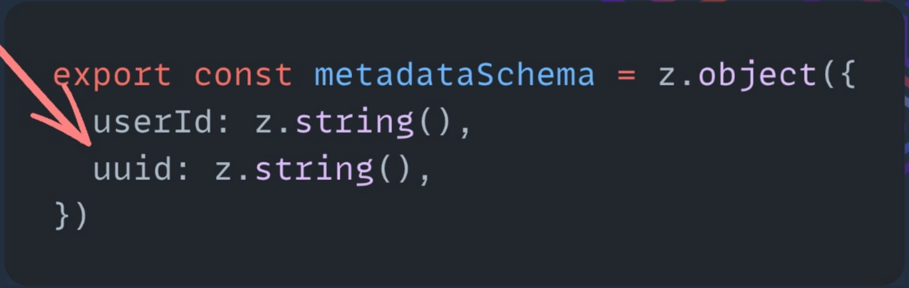
r/aws • vor 5 Jahren
Superfluxus

Duplicate SQS messages?

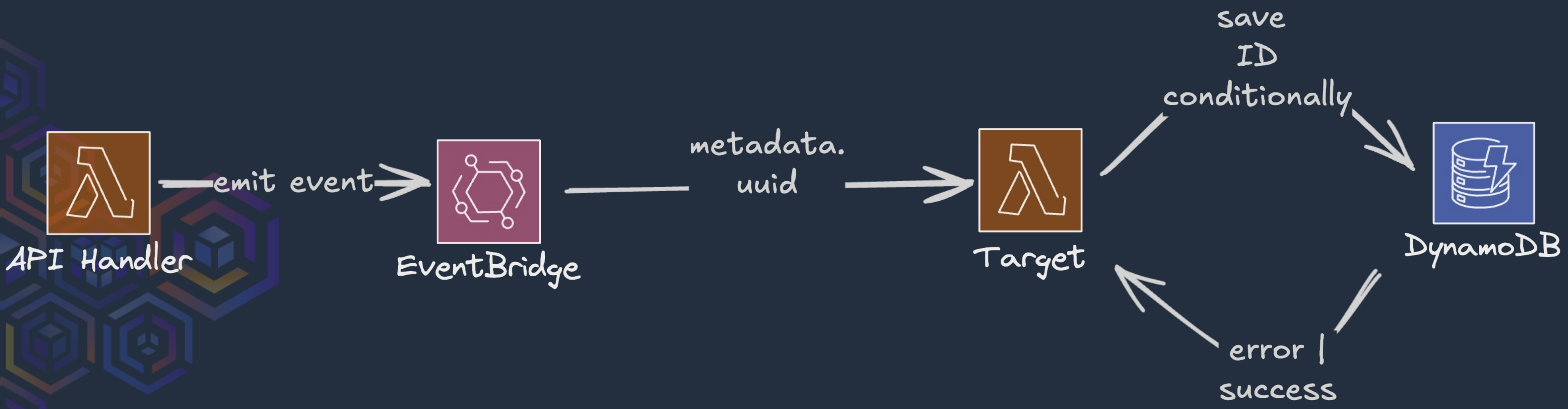
AWS La
firing recently?



```
await PostEventEmitter.published({
  requestId: this.requestId,
  authenticatedUserId,
  post: {
    id: result._id
  },
```

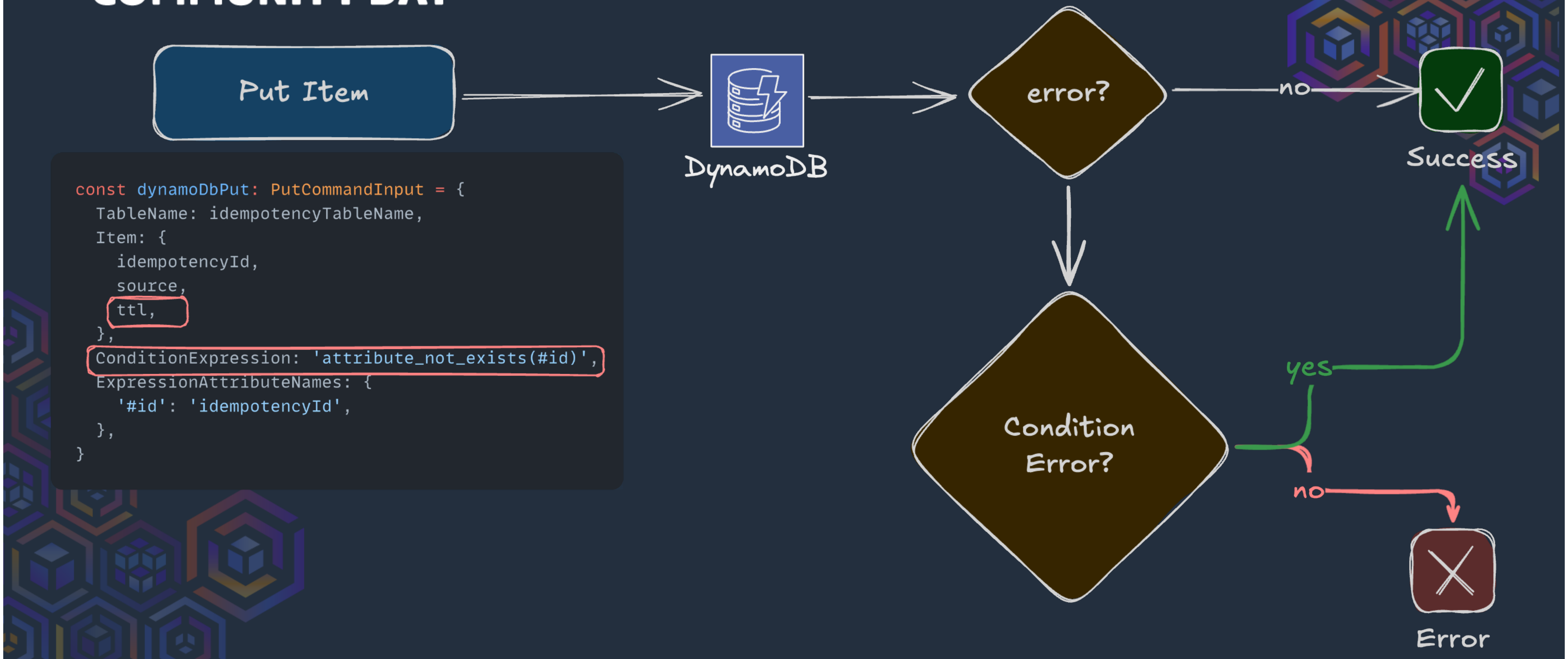


```
export const metadataSchema = z.object({
  userId: z.string(),
  uuid: z.string(),
})
```



aws

COMMUNITY DAY



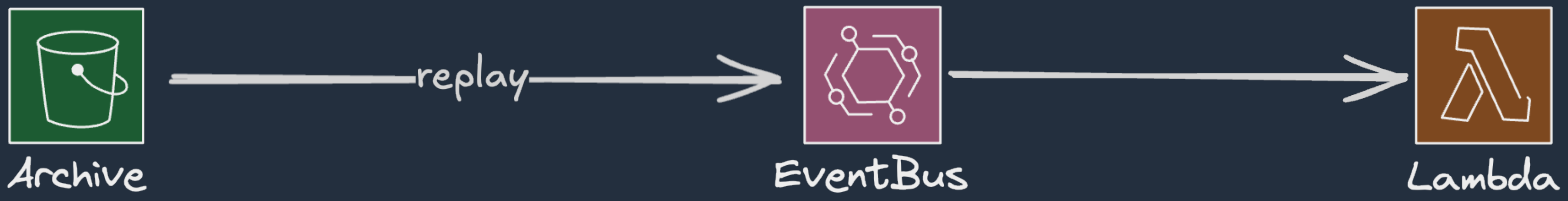
```
const dynamoDbPut: PutCommandInput = {
  TableName: idempotencyTableName,
  Item: {
    idempotencyId,
    source,
    ttl,
  },
  ConditionExpression: 'attribute_not_exists(#id)',
  ExpressionAttributeNames: {
    '#id': 'idempotencyId',
  },
}
```

aws

COMMUNITY DAY

- This is NOT idempotency -> It is deduplication

- Amazing Team @AWS: Powertools TypeScript (props to Andrea Amorosi)
  @aws-lambda-powertools/idempotency



Replay from 10:10:00 - 10:20:00

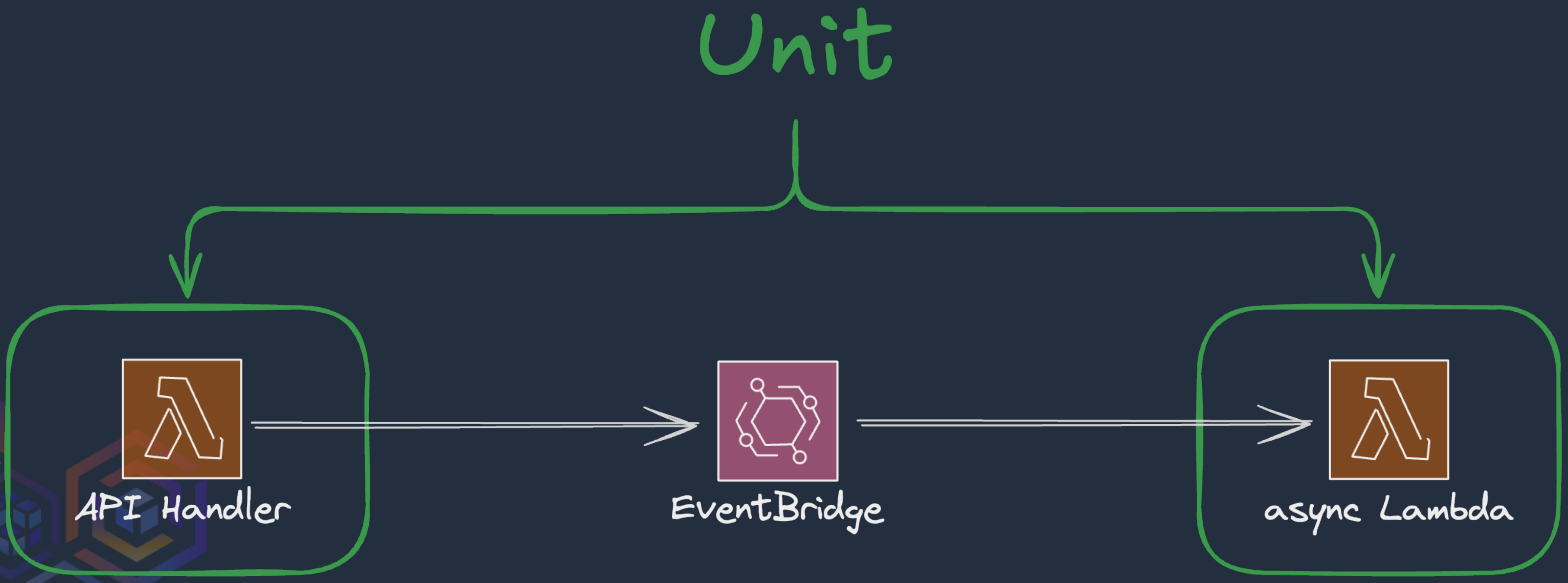
- How do you know you haven't handled some events already?
- This is where idempotency comes in
- Events will be deduplicated

Testing

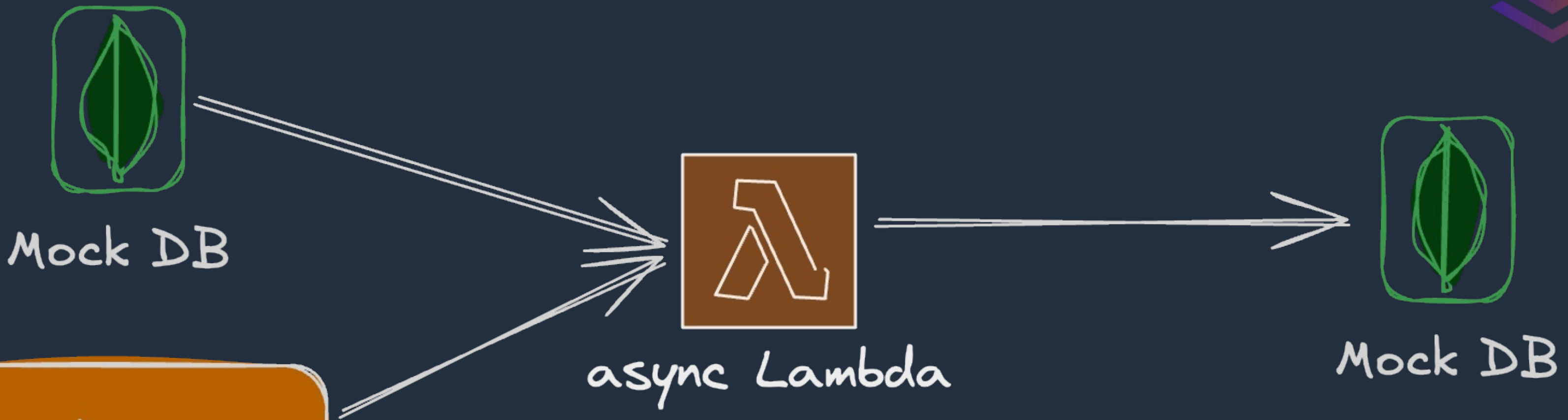
aws COMMUNITY DAY

Unit

Integration


Unit
API Handler
EventBridge
async Lambda

aws COMMUNITY DAY



ARRANGE

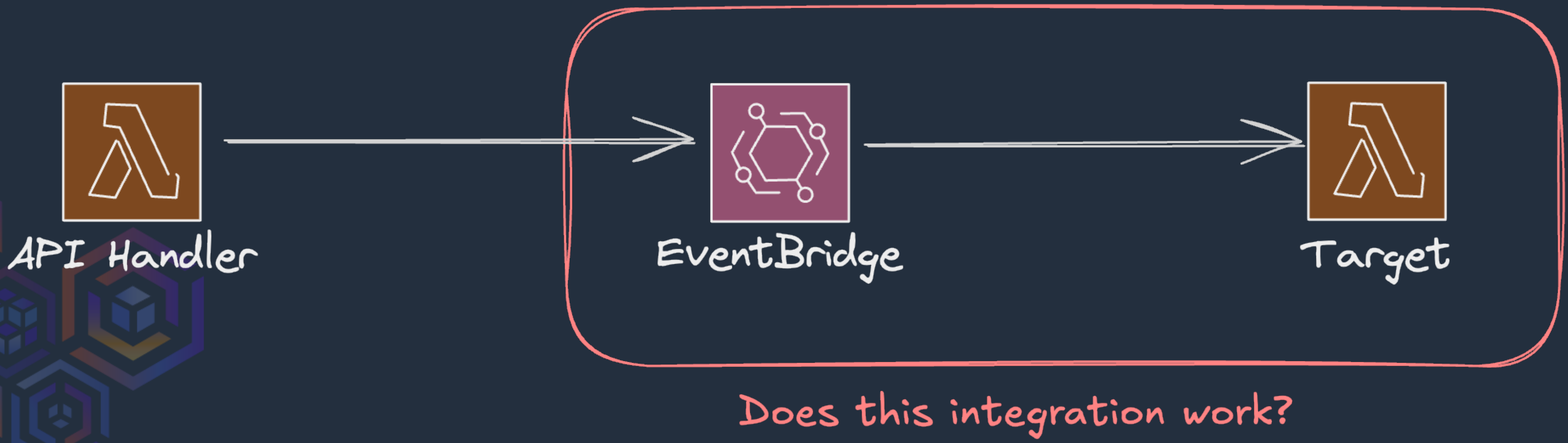
ACT

ASSERT

aws COMMUNITY DAY

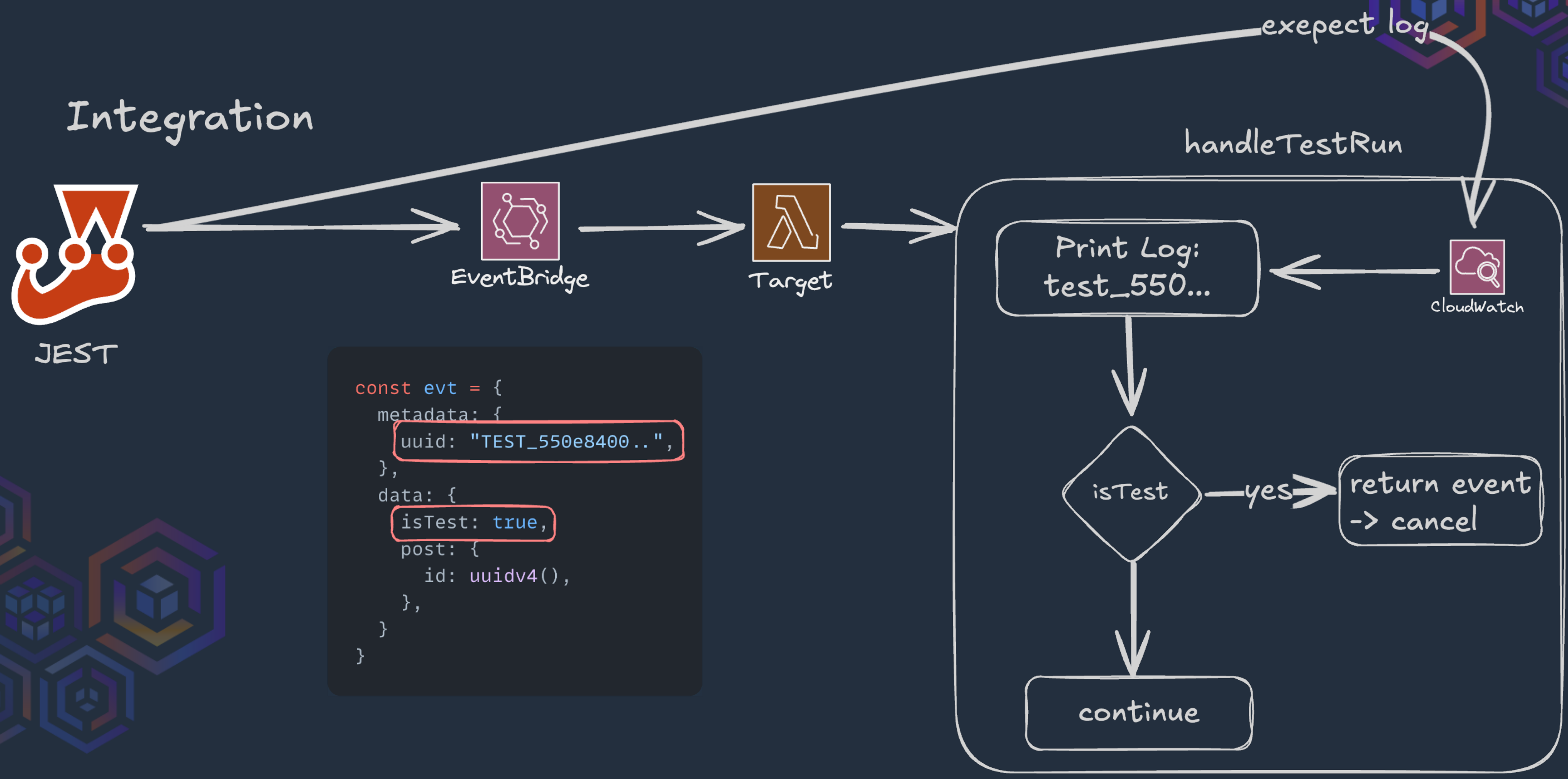
Integration

1. **Event Reaches Bus -> Bus forwards to target**
2. Target integration with other systems



COMMUNITY DAY

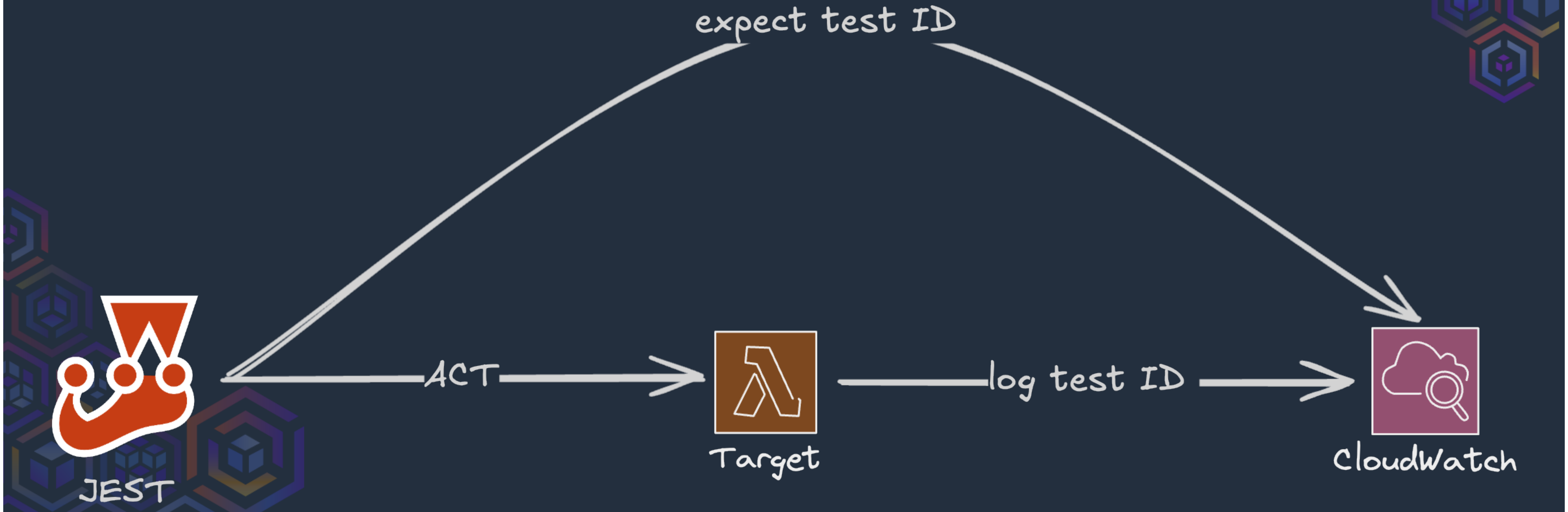
Integration



```
const evt = {
  metadata: {
    uuid: "TEST_550e8400..",
  },
  data: {
    isTest: true,
    post: {
      id: uuidv4(),
    },
  }
}
```

aws

COMMUNITY DAY



expect test ID

ACT

log test ID

JEST

Target

CloudWatch

More

aws

COMMUNITY DAY

- Create a rule log all group
- Use a log formatter across all your events
- Share your events in a npm package or monorepo
- Mock failures
- Use max concurrency
- Use as many alerts as needed and as few as possible (rather start too low than too high imo)
- Get started with it
- Idempotency without a Lambda in between is hard
- Not everything needs to be idempotent
- Think about full events vs. notification events (what is more expensive)
- Rather make the async consumer slow than the API – user performance is everything

Resources

aws

COMMUNITY DAY

- https://lumigo.io/blog/5-reasons-why-you-should-use-eventbridge-instead-of-sns/
- https://www.reddit.com/r/aws/comments/ystxxn/confused_about_eventbridge_5_targets_per_rule/
- https://www.youtube.com/watch?v=RoKAEzdcr7k
- https://engineering.hashnode.com/how-to-build-event-driven-architecture-on-aws
- https://www.youtube.com/watch?v=U5GZNt0iMZY&t=2346s
- https://www.youtube.com/watch?v=Wk0FoXTUEjo
- https://serverlessfirst.com/eventbridge-testing-guide/
- https://notes.paulswail.com/public/AWS

aws

COMMUNITY DAY

- Feedback appreciated!
- Freelance available

Author
Sandro Volpicella
@sandro_vol