

goto;

GOTO EDA Day 2022

📍 Codenode | London

📅 September 1st, 2022



BROUGHT TO YOU BY
AWS AND PARTNERS





GOTO
Guide

Let us help you

Ask questions
through **the app**



remember to rate the session

THANK YOU





Event design and event first development

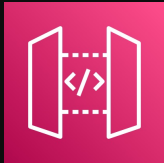
David Boyne

SDA, Aws Serverless

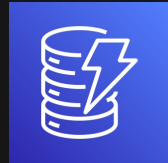
The adoption of event-driven architecture (EDA) is growing



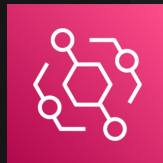
Event-driven architectures becoming more accessible



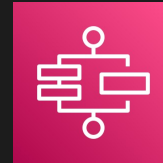
Amazon
API Gateway



Amazon
DynamoDB



Amazon
EventBridge

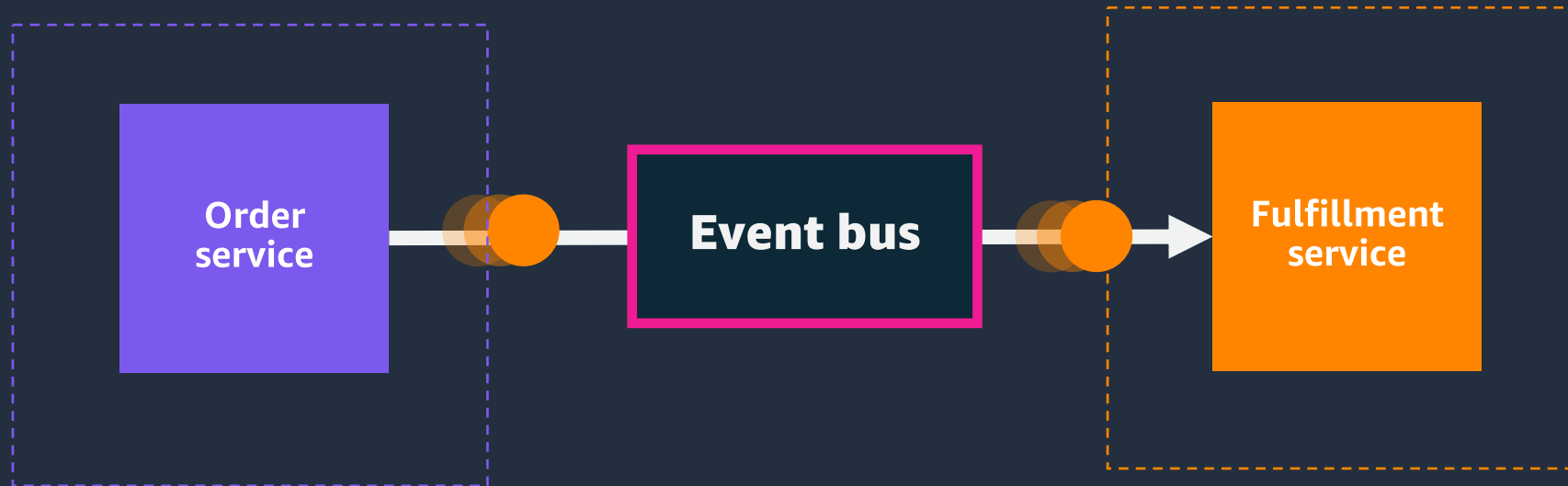


AWS Step
Functions

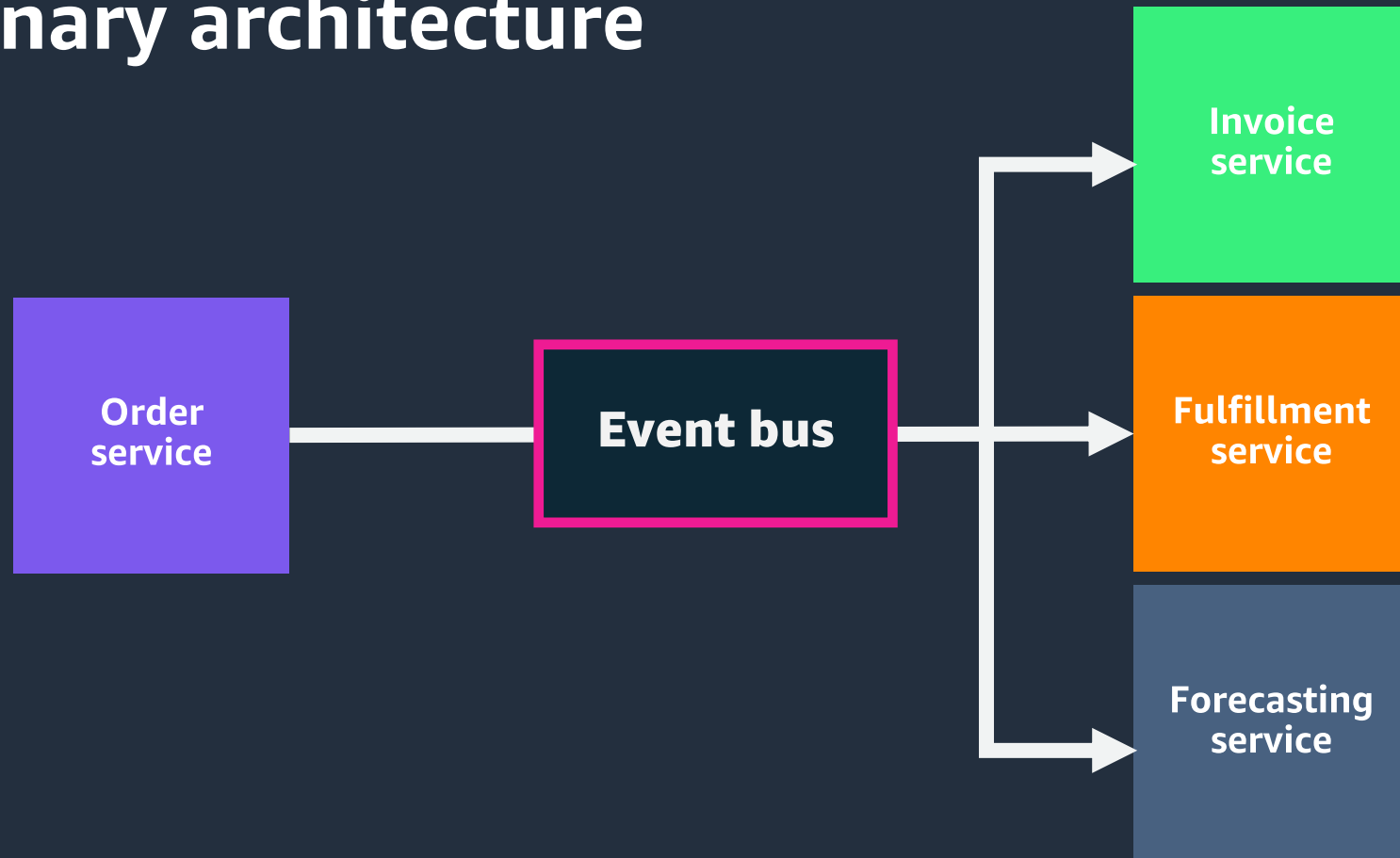


AWS
Lambda

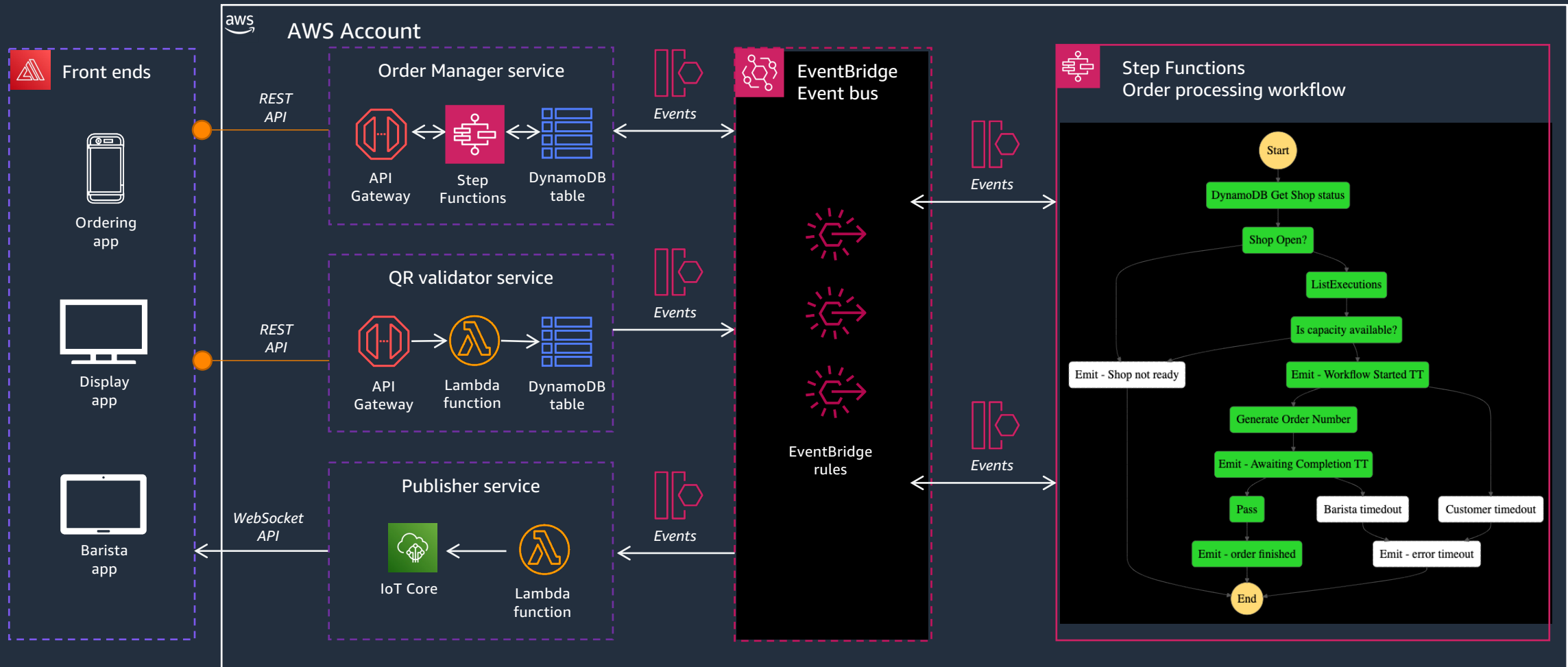
Decoupled architectures



Evolutionary architecture

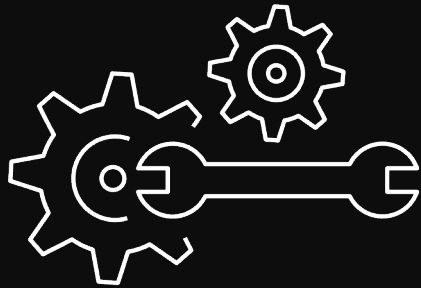


Event-driven architectures provide agility

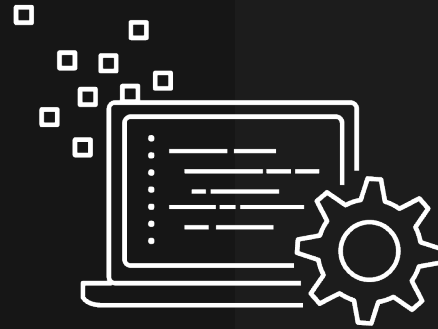


serverlesspresso

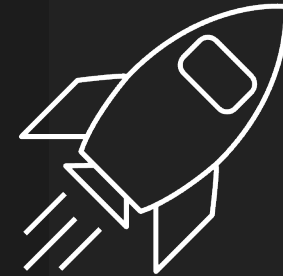
Benefits of event-driven architectures (EDA)



Loose coupling



Extensibility



Agility

As you scale your event-driven architecture problems start to emerge...

Common problems with event-driven architectures

Producer / consumer responsibilities



Coupling services with events



Lack of governance and standards



Common problems with event-driven architectures

Unclear event intent



Exposed domain information



Focusing on implementation
vs behavior



How you design your events impacts your architecture

Agenda

- Where do problems start?
- Understanding event patterns and tradeoffs
- Moving to event first thinking
- Summary and resources

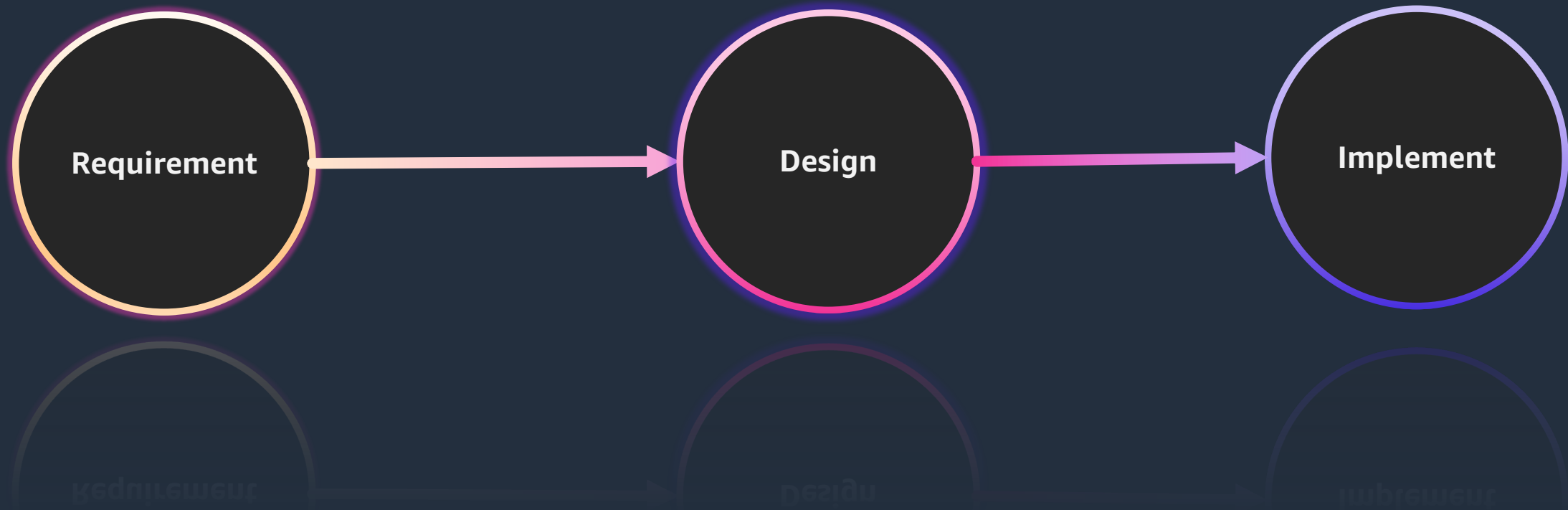
Who am I?

- David Boyne
 - Twitter: @boyney123
- Senior Developer Advocate – AWS Serverless
- Father of 2
- X Guitar Player
- Previously
 - Startups, Open Source, Tooling, and EDA

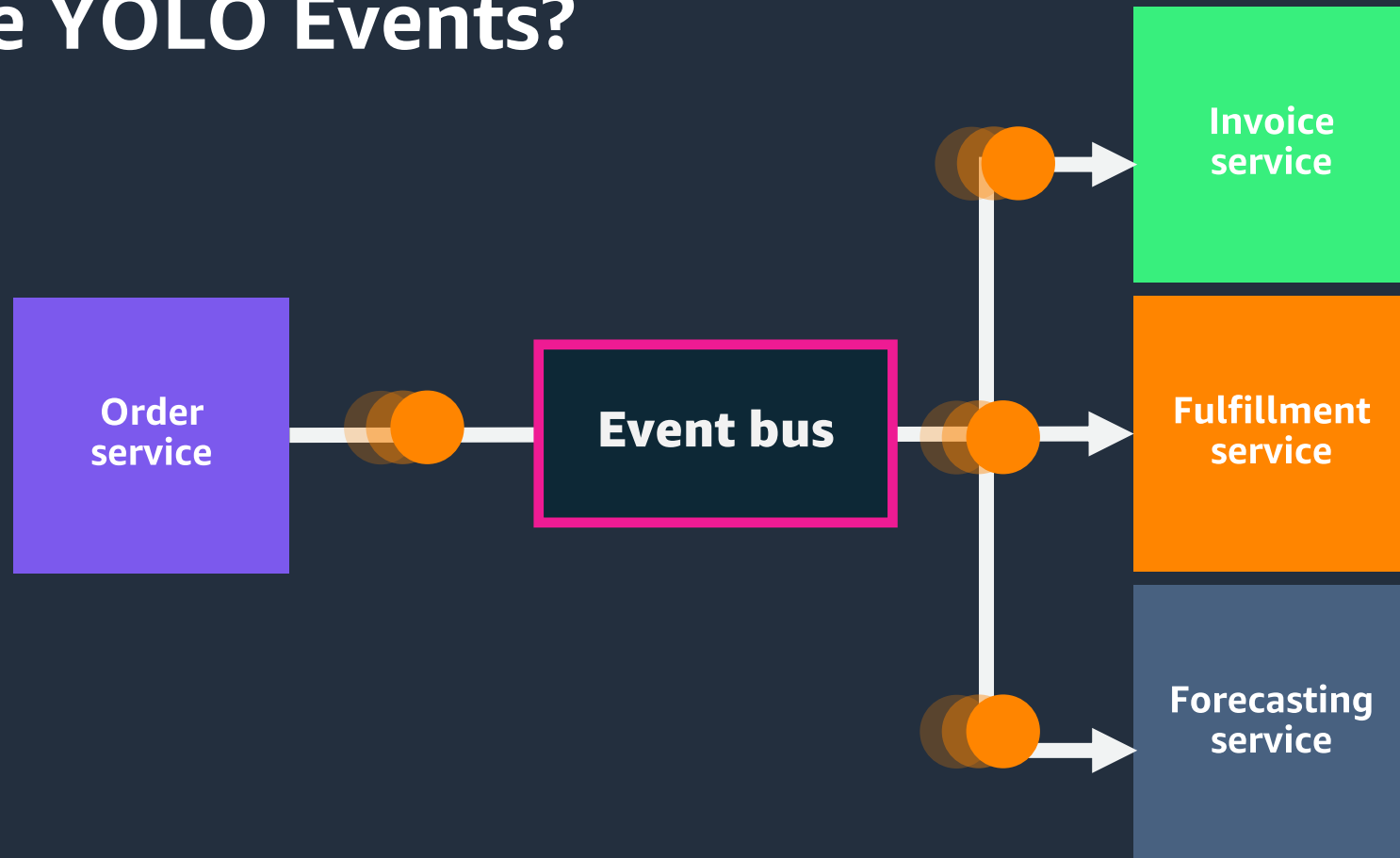


1. Where do problems start?

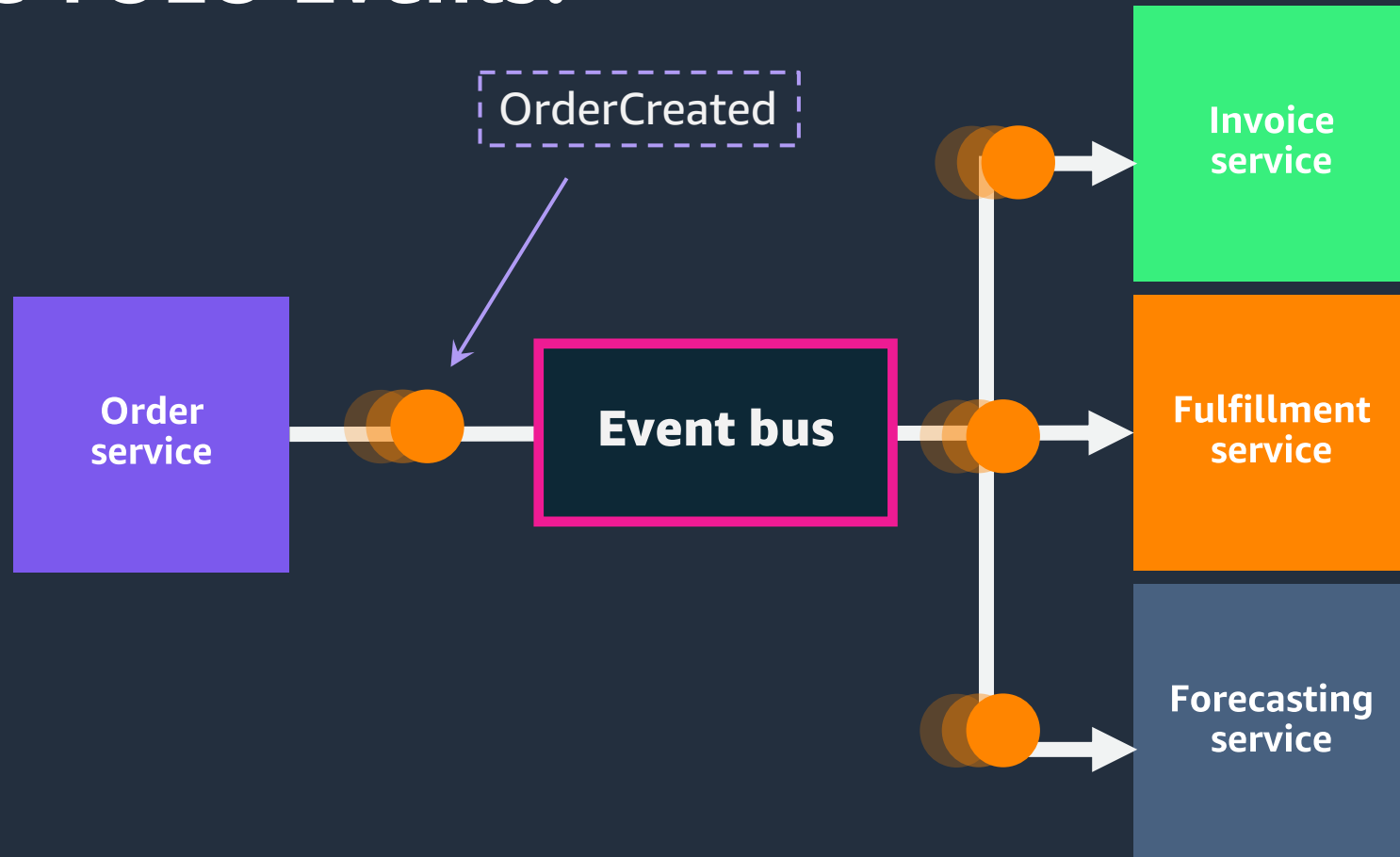
Designing and developing events within the team



What are YOLO Events?



What are YOLO Events?



OrderCreated Event

Order
service

```
{
  "version": "1",
  "id": "07fffe3b-4b25-48a7-b4a8-432bcc3bfb2c",
  "detail-type": "OrderCreated",
  "source": "myapp.orders",
  "account": "123456789",
  "time": "2022-06-01T00:00:00Z",
  "region": "us-west-1",
  "detail": {
    "data": {
      "orderId": "3c947443-fd5f-4bfa-8a12-2aa348e793ae",
      "userId": "09586e5c-9983-4111-8395-2ad5cfd3733b"
    }
  }
}
```

What is the event detail-type?

Order
service

```
{
  "version": "1",
  "id": "07fffe3b-4b25-48a7-b4a8-432bcc3bfb2c",
  "detail-type": "OrderCreated",
  "source": "myapp.orders",
  "account": "123456789",
  "time": "2022-06-01T00:00:00Z",
  "region": "us-west-1",
  "detail": {
    "data": {
      "orderId": "3c947443-fd5f-4bfa-8a12-2aa348e793ae",
      "userId": "09586e5c-9983-4111-8395-2ad5cfd3733b"
    }
  }
}
```

event name

What is the event source?

Order
service

```
{
  "version": "1",
  "id": "07fffe3b-4b25-48a7-b4a8-432bcc3bfb2c",
  "detail-type": "OrderCreated",
  "source": "myapp.orders",
  "account": "123456789",
  "time": "2022-06-01T00:00:00Z",
  "region": "us-west-1",
  "detail": {
    "data": {
      "orderId": "3c947443-fd5f-4bfa-8a12-2aa348e793ae",
      "userId": "09586e5c-9983-4111-8395-2ad5cfd3733b"
    }
  }
}
```

application domains

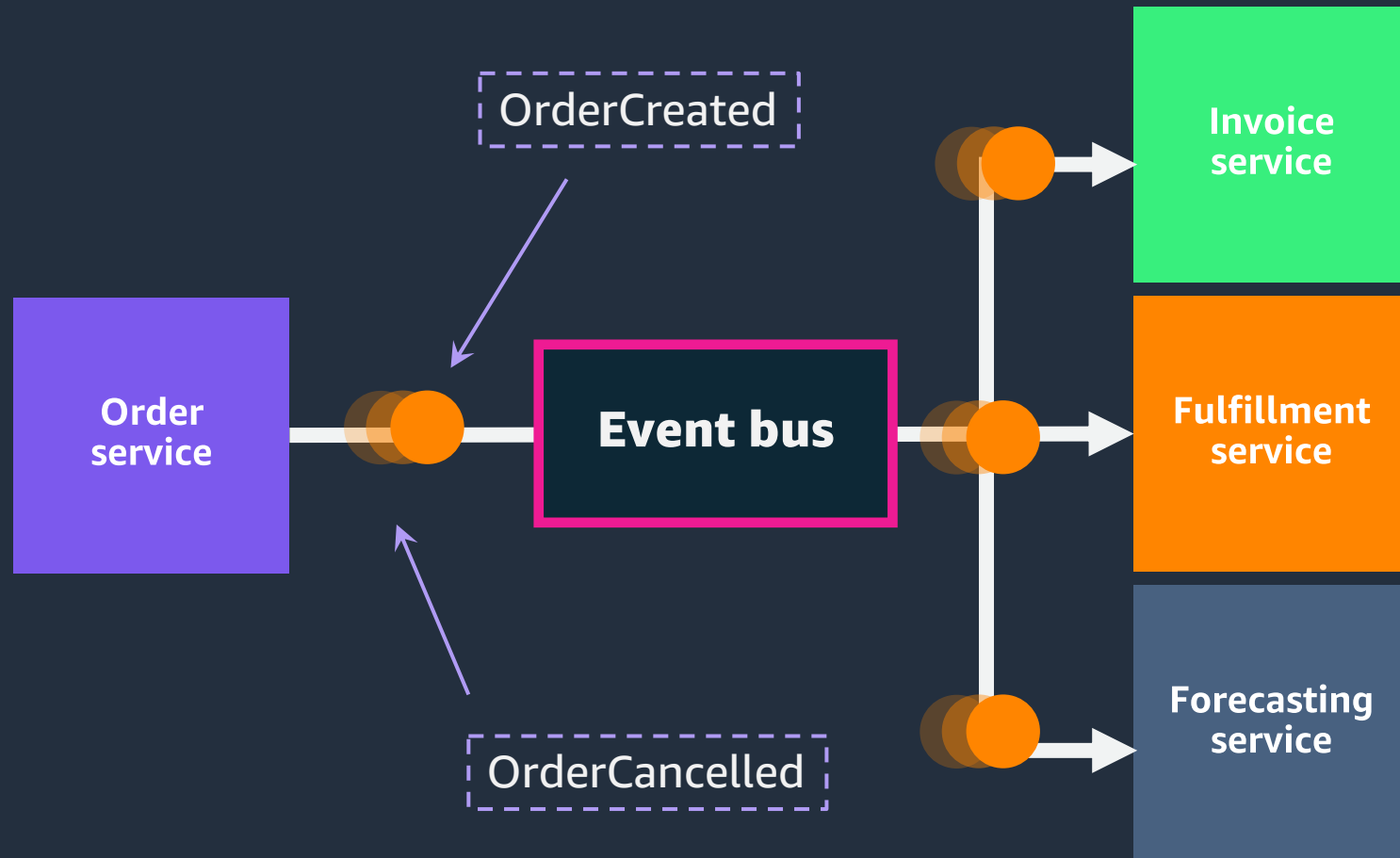
Wat is the event detail?

Order
service

```
{
  "version": "1",
  "id": "07fffe3b-4b25-48a7-b4a8-432bcc3bfb2c",
  "detail-type": "OrderCreated",
  "source": "myapp.orders",
  "account": "123456789",
  "time": "2022-06-01T00:00:00Z",
  "region": "us-west-1",
  "detail": {
    "data": {
      "orderId": "3c947443-fd5f-4bfa-8a12-2aa348e793ae",
      "userId": "09586e5c-9983-4111-8395-2ad5cfd3733b"
    }
  }
}
```

← custom data

Evolution of YOLO event

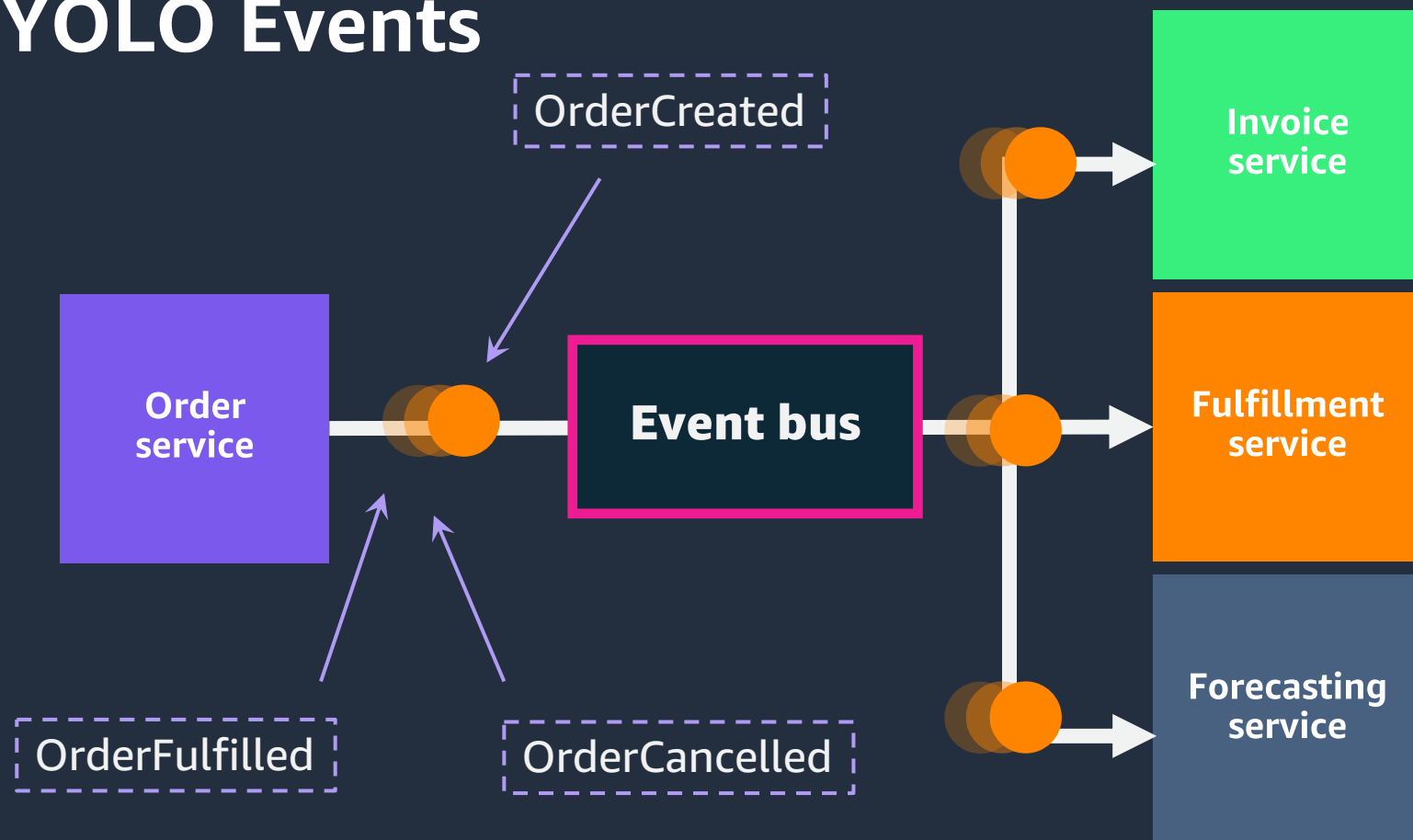


Building the OrderCancelled Event

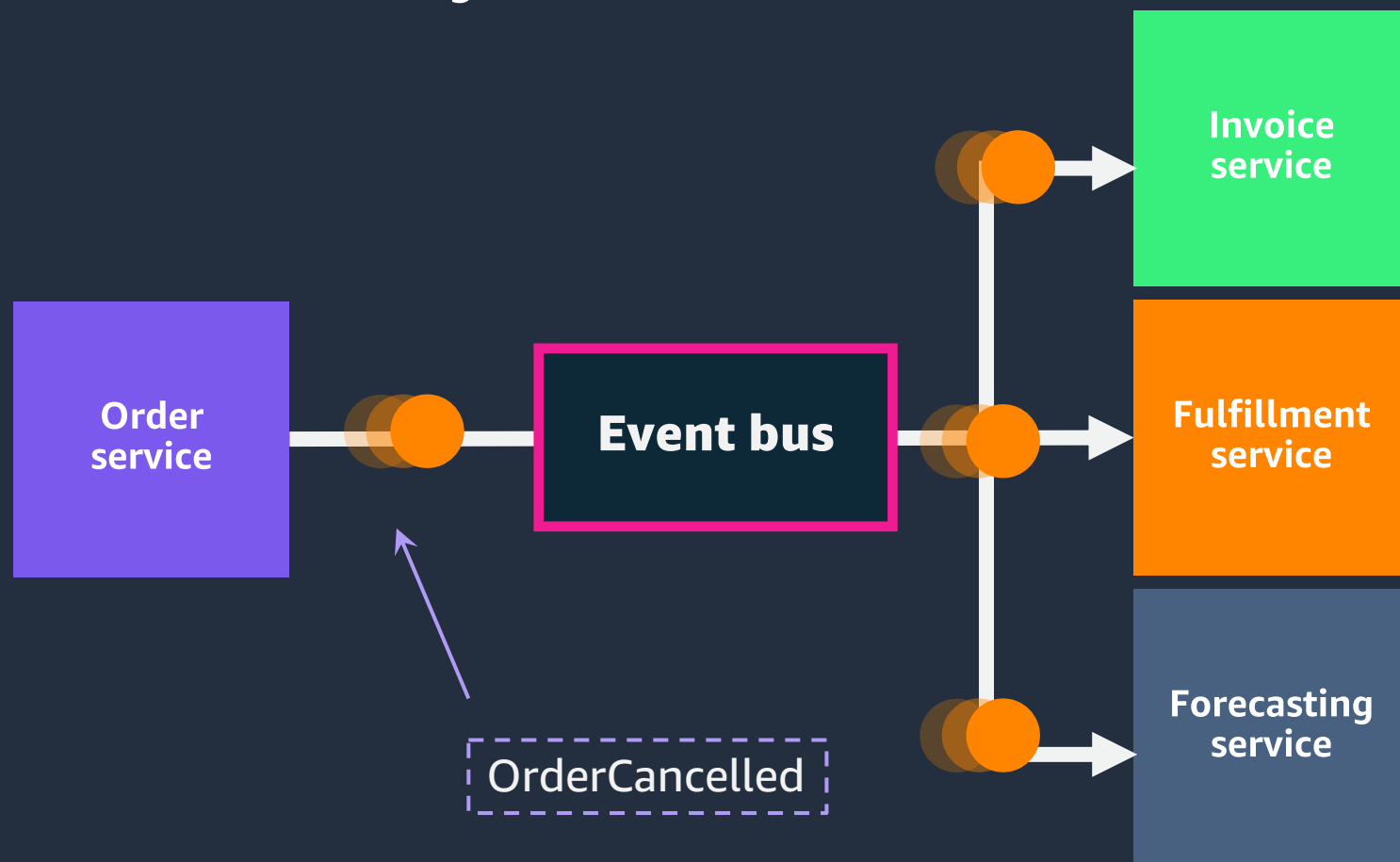
Order
service

```
{
  ...,
  "detail-type": "OrderCancelled",
  "source": "myapp.orders",
  "account": "123456789",
  "time": "2022-06-01T00:00:00Z",
  "region": "us-west-1",
  "detail": {
    "data": {
      "id": "3c947443-fd5f-4bfa-8a12-2aa348e793ae",
      "reason": "No longer needed",
      "type": "T_W",
      "user": {
        "id": "09586e5c-9983-4111-8395-2ad5cfd3733b",
        "firstName": "David",
        "lastName": "Boyne"
      }
    }
  }
}
```

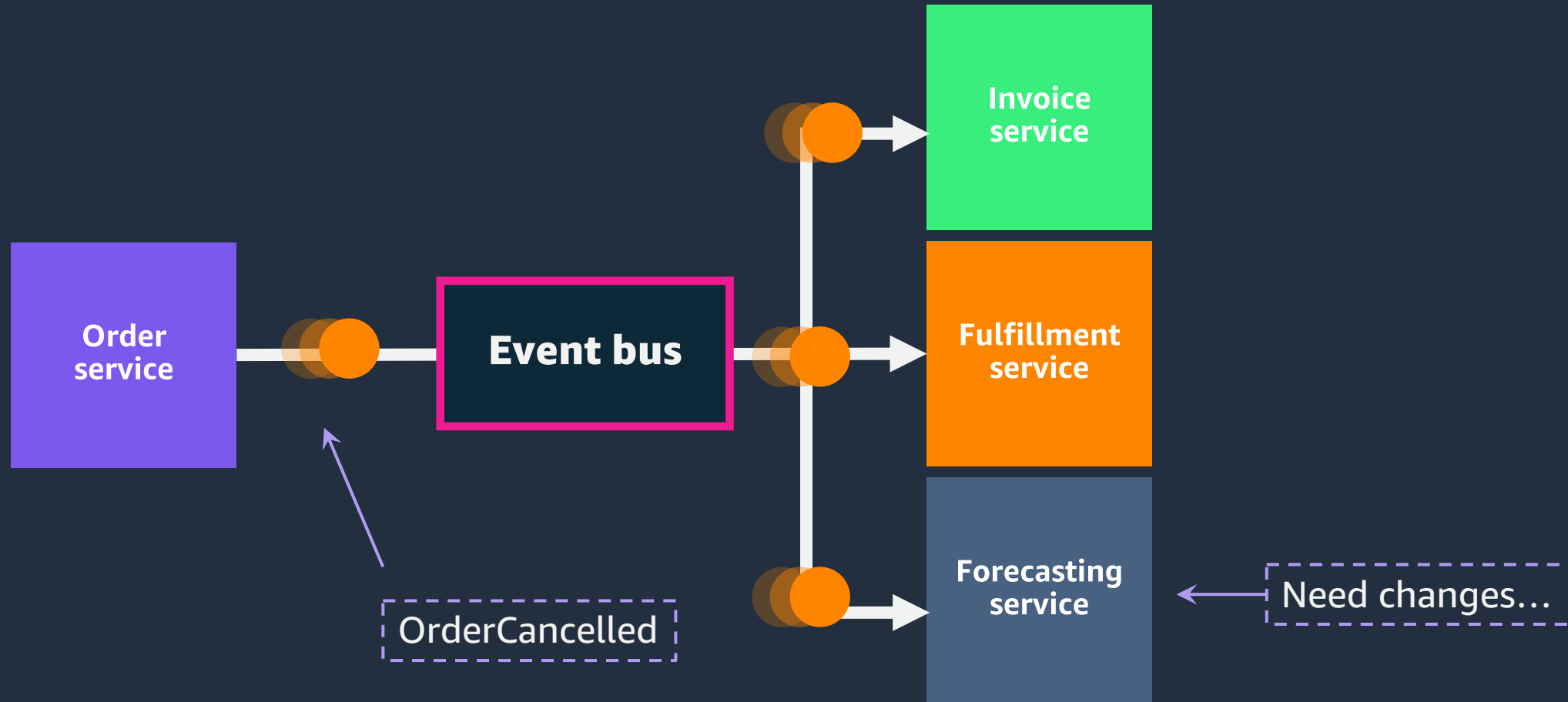
Future YOLO Events



Consumer Driven Payload



Requiring changes to event contract



Requests to change the reason field

Order
service

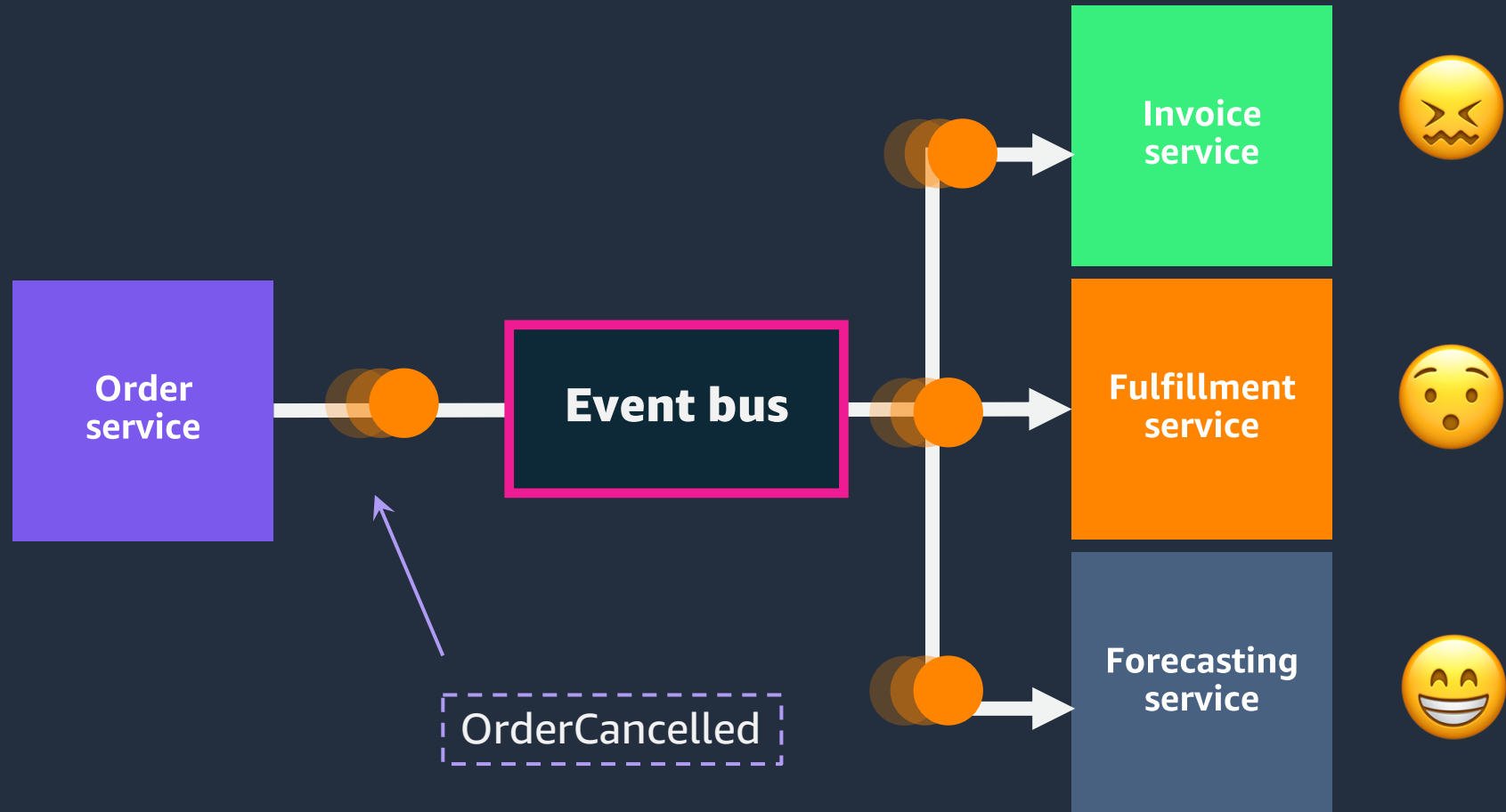
```
{
  ...,
  "detail-type": "OrderCancelled",
  "source": "myapp.orders",
  "account": "123456789",
  "time": "2022-06-01T00:00:00Z",
  "region": "us-west-1",
  "detail": {
    "data": {
      "id": "3c947443-fd5f-4bfa-8a12-2aa348e793ae",
      "reason": "No longer needed",
      "type": "T_W",
      "user": {
        "id": "09586e5c-9983-4111-8395-2ad5cfd3733b",
        "firstName": "David",
        "lastName": "Boyne"
      }
    }
  }
}
```

Changes to reason field

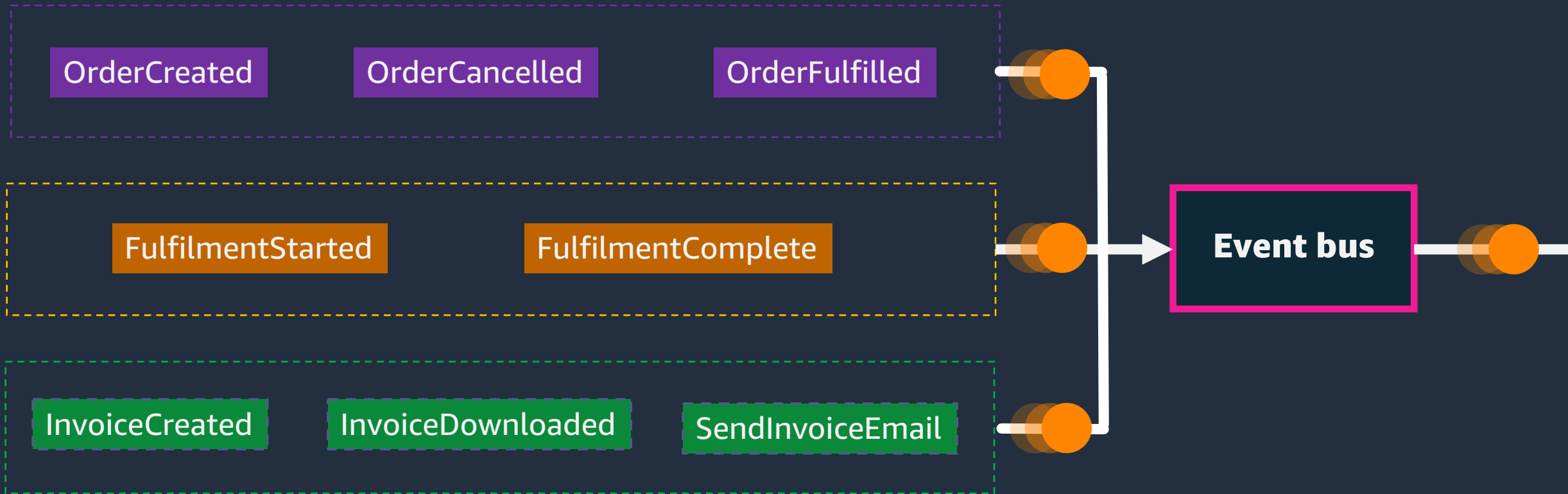
Order
service

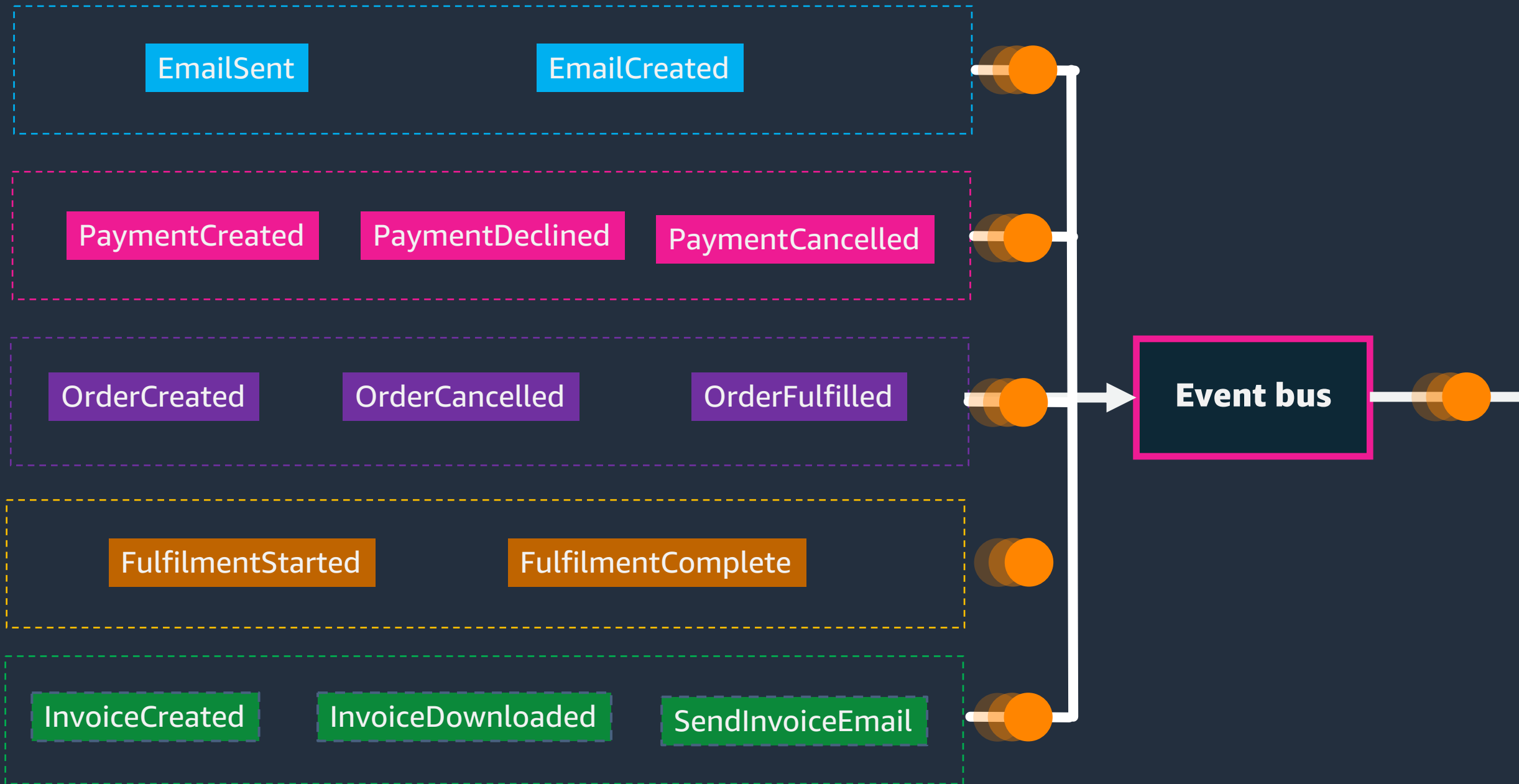
```
{
  ...,
  "detail": {
    "data": {
      "id": "3c947443-fd5f-4bfa-8a12-2aa348e793ae",
      "reason": "No longer needed",
      "type": {
        "id": "T_W",
        "description": "User cancelled order",
        "userCancelled": true,
        "systemCancelled": false
      },
      "user": {
        "id": "09586e5c-9983-4111-8395-2ad5cfd3733b",
        "firstName": "David",
        "lastName": "Boyne"
      }
    }
  }
}
```

Breaking contracts with schemas



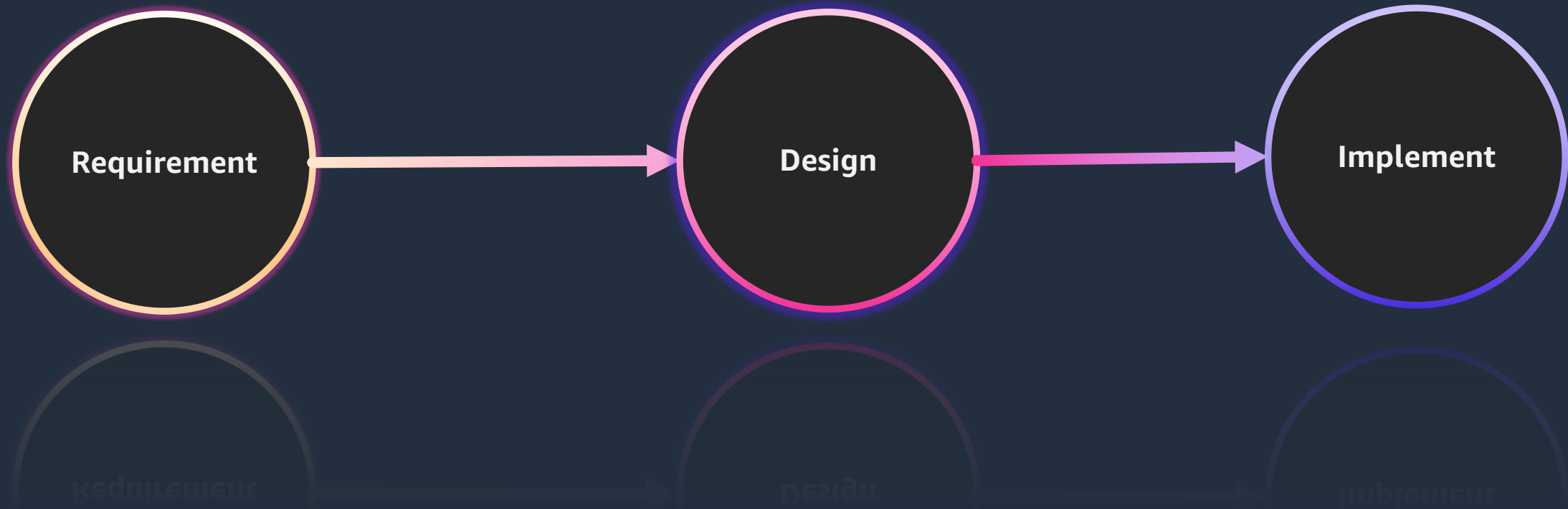
Value driven development





How can event design help avoid common problems?

Changing the way we think and design our events



2. Understanding event patterns and tradeoffs

“

**What information should we
put into our events? ”**

Everyone building event-driven applications

Event structure is down to you

Order
service

```
{  
  "version": "1",  
  "id": "07fffe3b-4b25-48a7-b4a8-432bcc3bfb2c",  
  "detail-type": "OrderCreated",  
  "source": "myapp.orders",  
  "account": "123456789",  
  "time": "2022-06-01T00:00:00Z",  
  "region": "us-west-1",  
  "detail": {  
    ...  
  }  
}
```

← you define the contract

Start by adding metadata to your event

Order
service

```
{
  "version": "0",
  "id": "0d079340-135a-c8c6-95c2-41fb8f496c53",
  "detail-type": "OrderCreated",
  "source": "myapp.orders",
  "account": "123451235123",
  "time": "2022-02-01T18:41:53Z",
  "detail": {
    "metadata": {
      "correlation_id": "dddd9340-135a-c8c6-95c2-41fb8f492222",
      "service": "orders-service",
      "domain": "SHOP"
    },
    "data": {
      ...
    }
  }
}
```

standard data

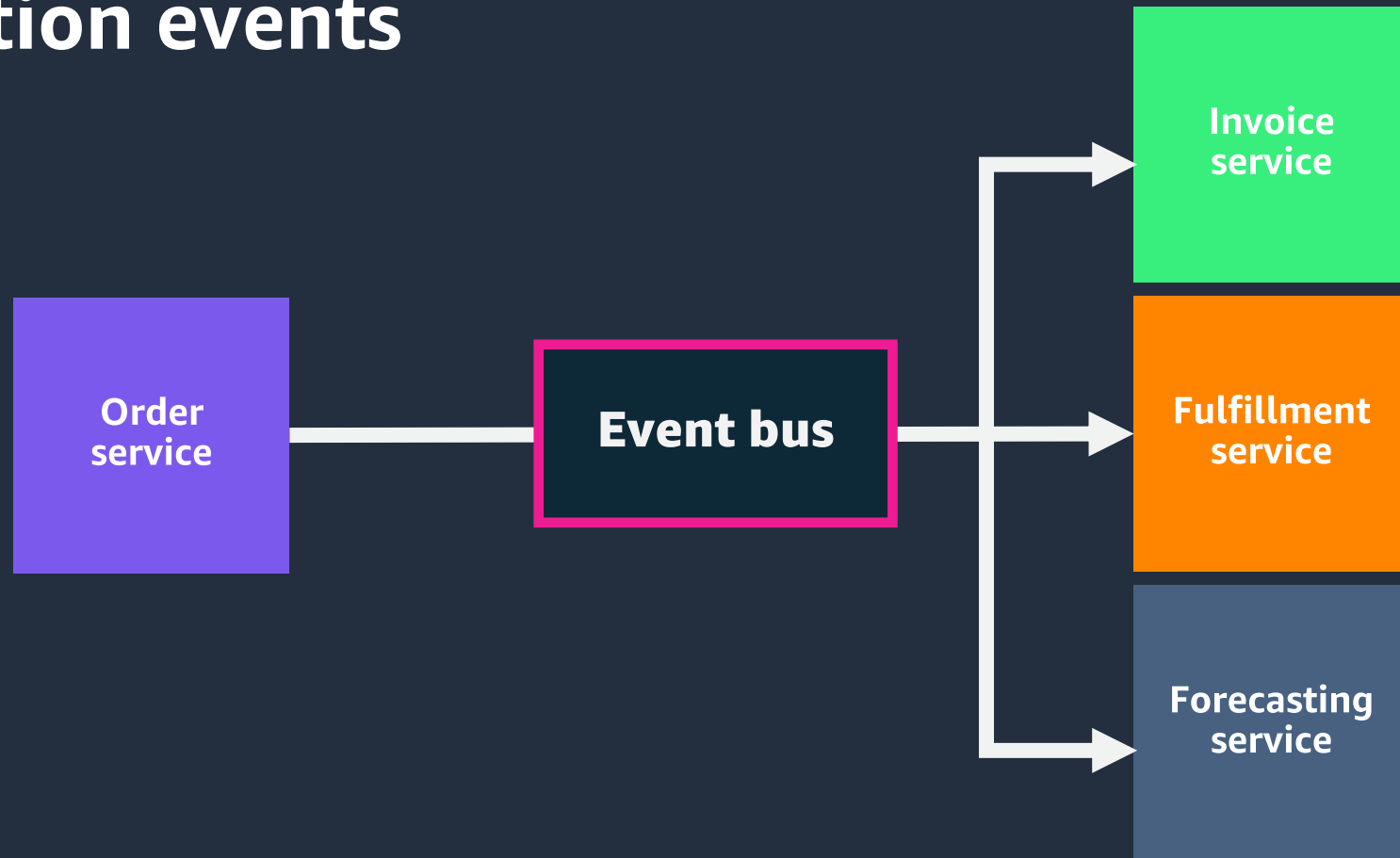


Custom library to handle events



Rules & Filters

Notification events



Notification event for OrderCreated

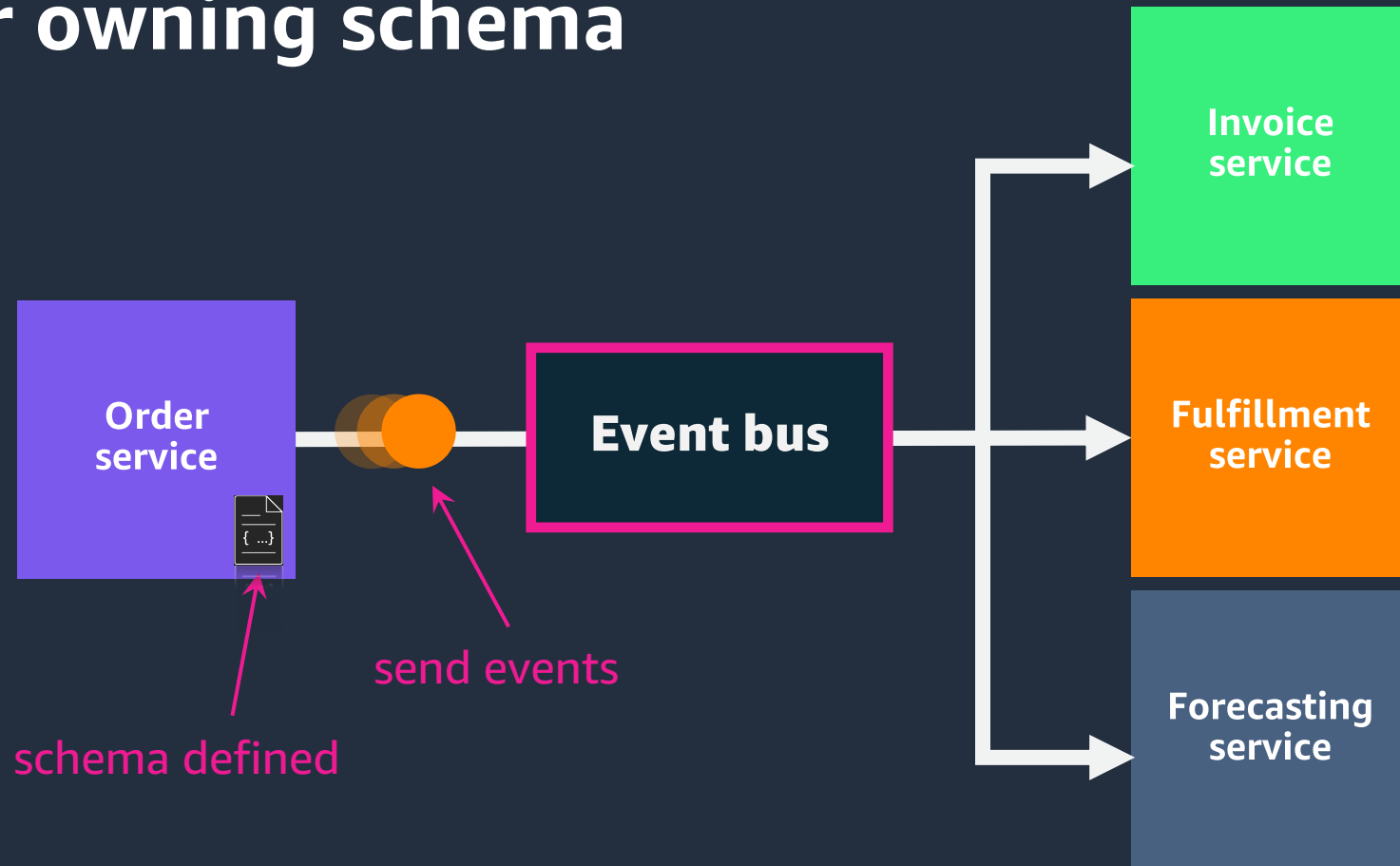
Order
service

```
{
  "version": "1",
  "id": "07fffe3b-4b25-48a7-b4a8-432bcc3bfb2c",
  "detail-type": "OrderCreated",
  "source": "myapp.orders",
  "account": "123456789",
  "time": "2022-06-01T00:00:00Z",
  "region": "us-west-1",
  "detail": {
    "metadata": {
      "domain": "ORDERS"
    },
    "data": {
      "orderId": "3c947443-fd5f-4bfa-8a12-2aa348e793ae",
      "userId": "09586e5c-9983-4111-8395-2ad5cfd3733b"
    }
  }
}
```

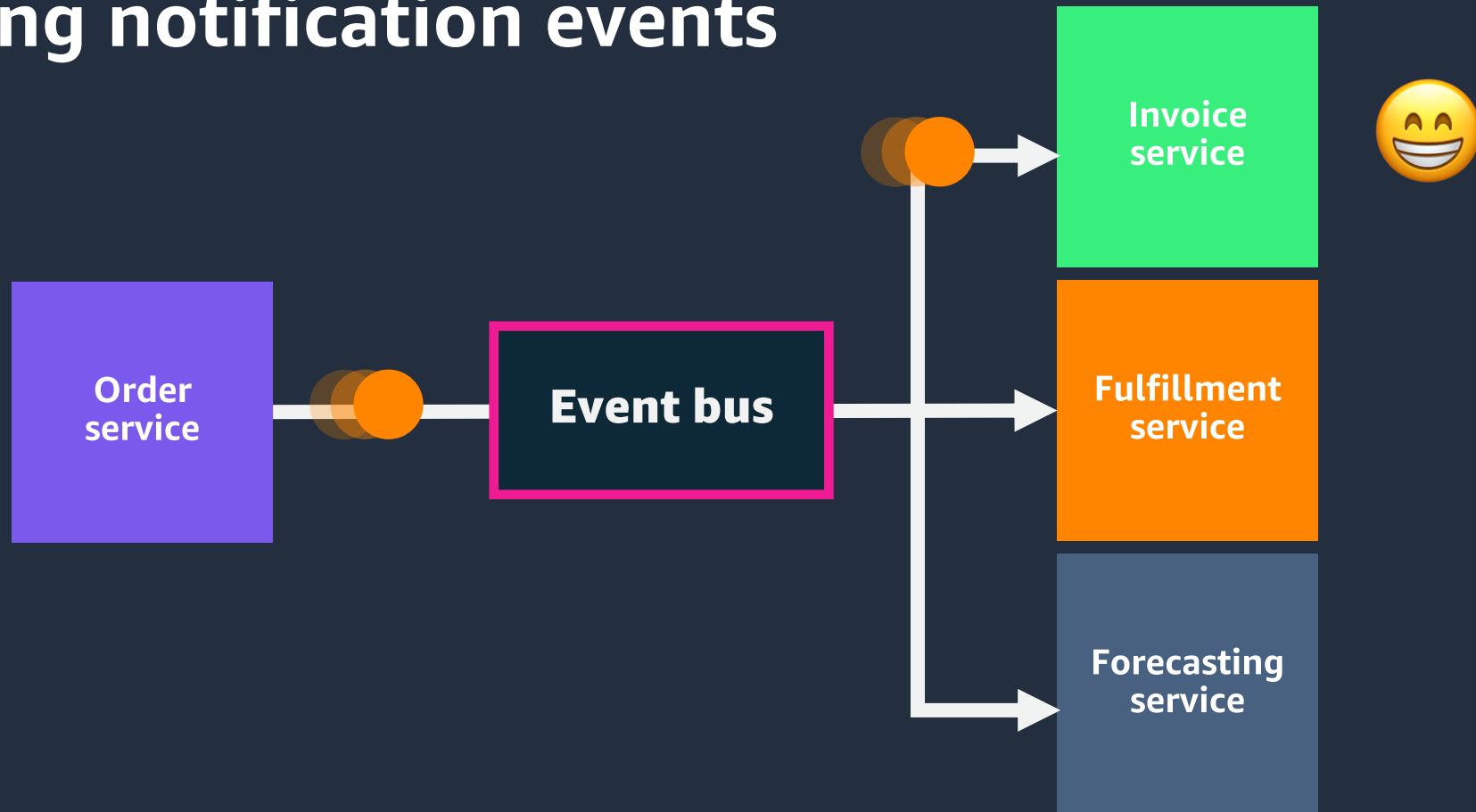
id to downstream services, assumed it's OK



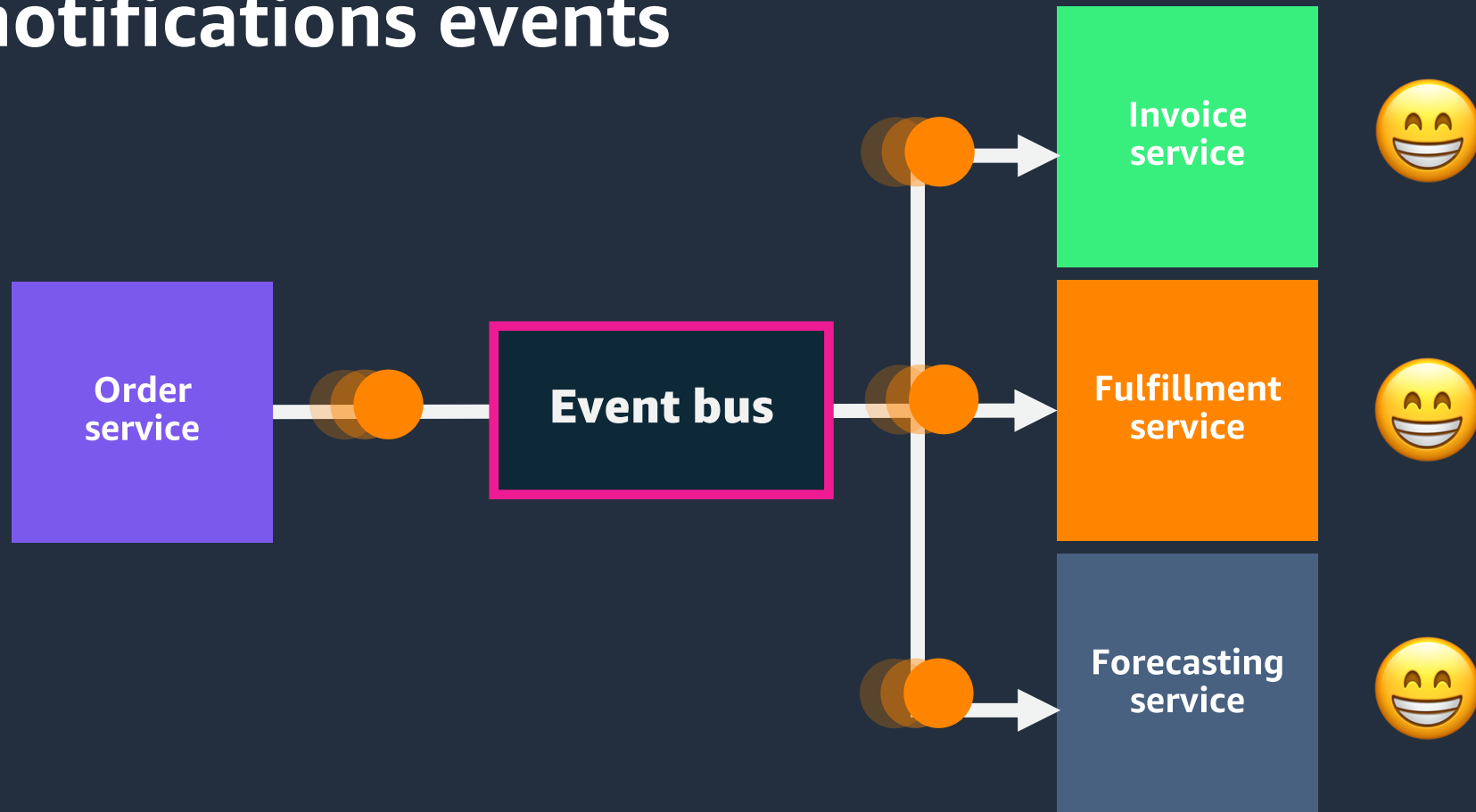
Producer owning schema



Consuming notification events



Scaling notifications events



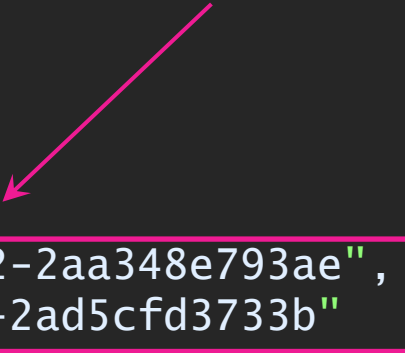
Notification events are
simple contracts, not
designed for consumers, **easy
to get started** but come with
a cost...

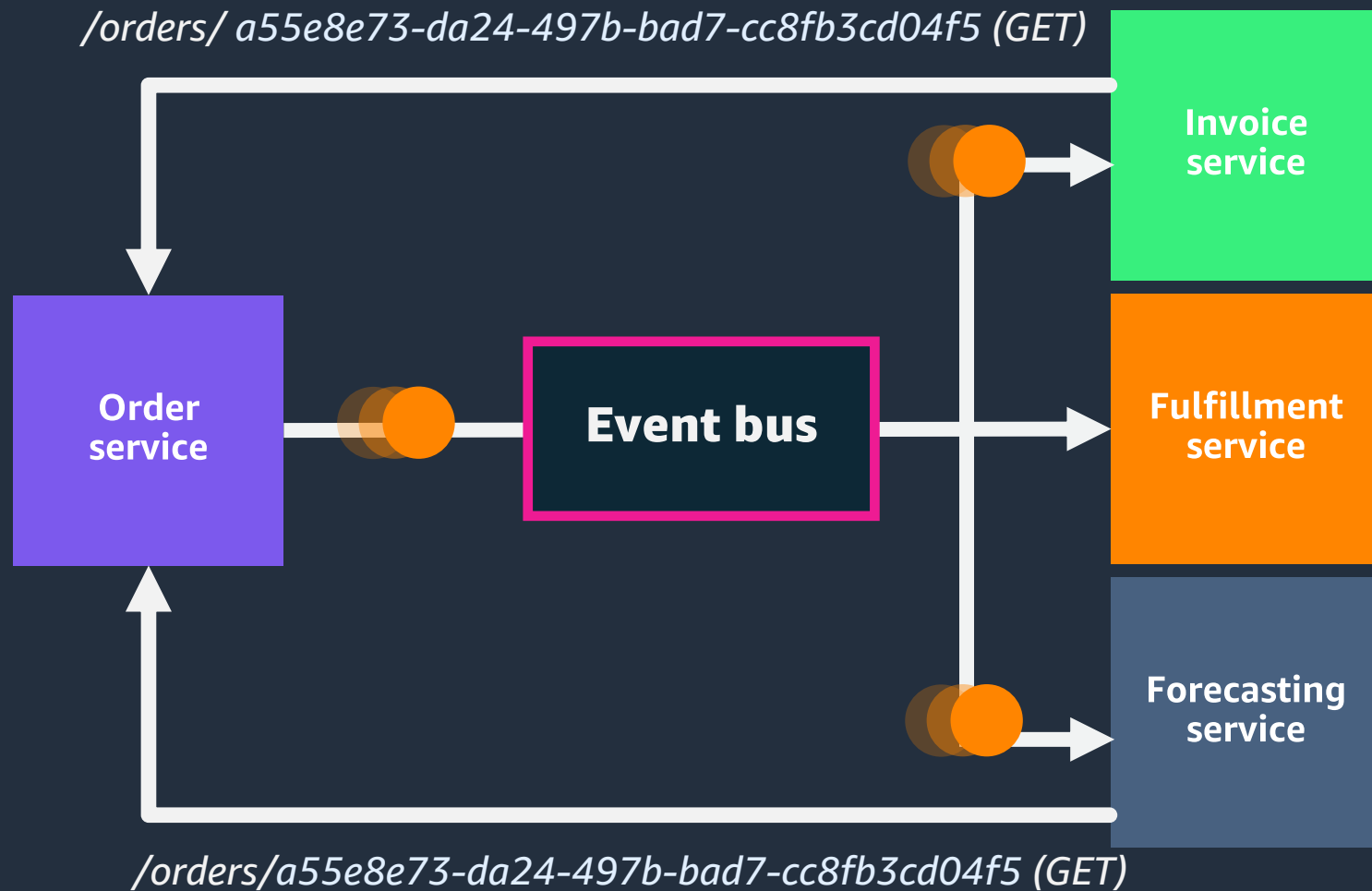
Notification Event: State...?

Order
service

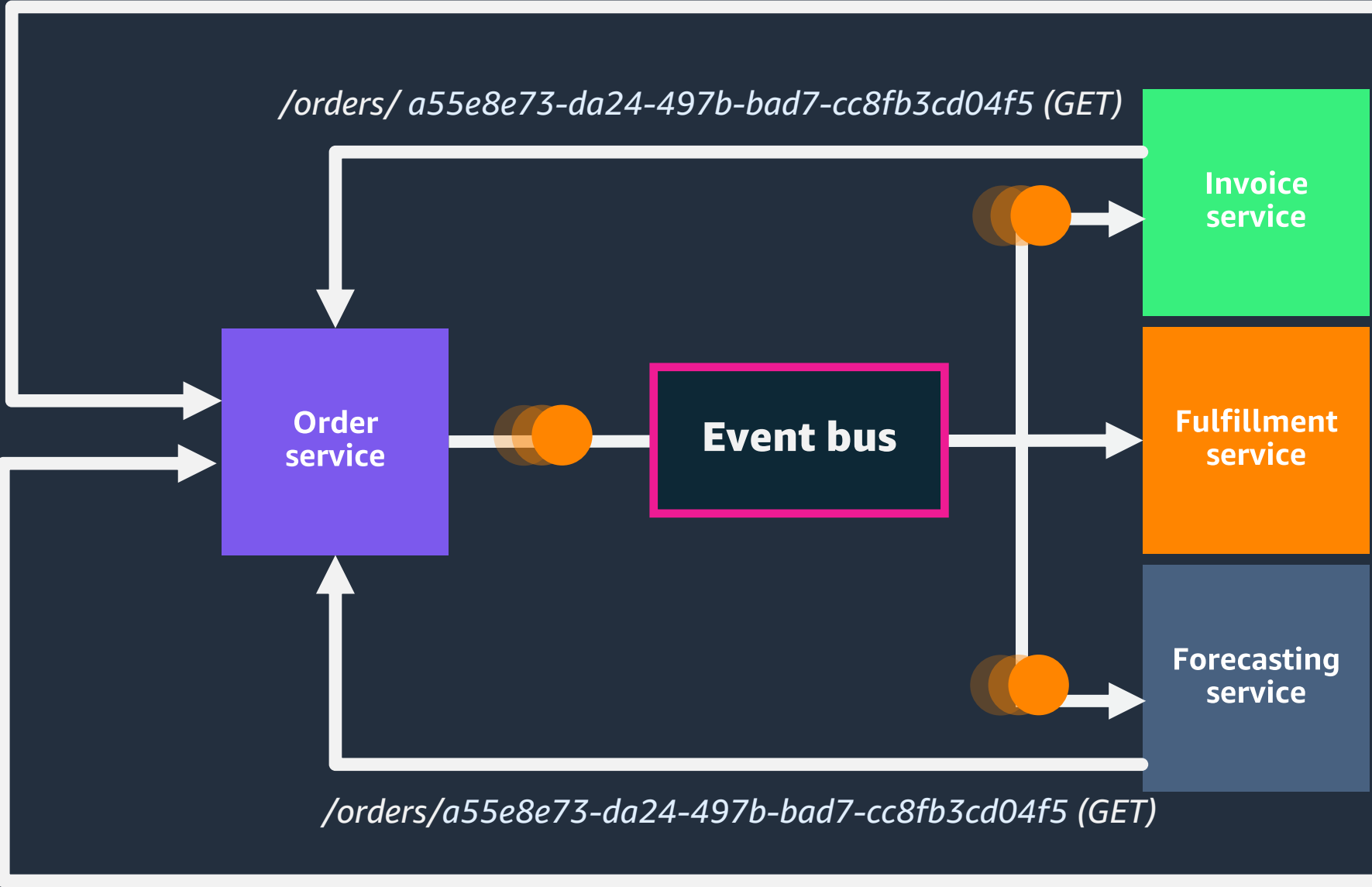
```
{
  "version": "1",
  "id": "07fffe3b-4b25-48a7-b4a8-432bcc3bfb2c",
  "detail-type": "OrderCreated",
  "source": "myapp.orders",
  "account": "123456789",
  "time": "2022-06-01T00:00:00Z",
  "region": "us-west-1",
  "detail": {
    "metadata": {
      "domain": "ORDERS"
    },
    "data": {
      "orderId": "3c947443-fd5f-4bfa-8a12-2aa348e793ae",
      "userId": "09586e5c-9983-4111-8395-2ad5cfd3733b"
    }
  }
}
```

how useful is this?





/orders/07fffe3b-4b25-48a7-b4a8-432bcc3bfb2c (GET)



/orders/34ff821c-e01d-4d49-b944-90b84579a70f (GET)

Summary of notification events



Simplicity

*Simple contracts,
less assumptions*



Coupling

*With requests vs
contracts*



Pull model

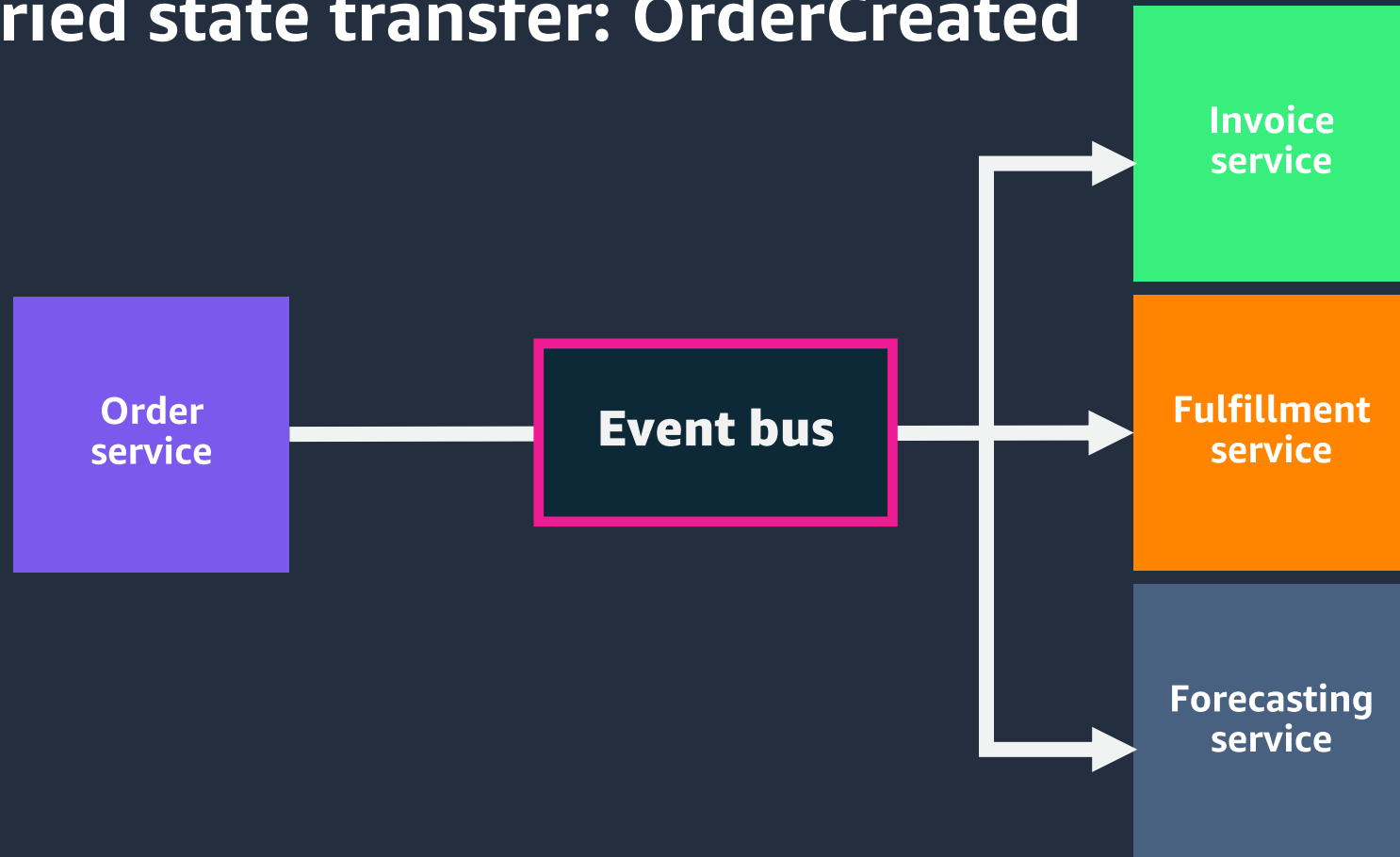
*Consumer request
information from source*



Limited filtering

*EventBridge rules
limited*

Event-carried state transfer: OrderCreated



Event-carried state transfer: OrderCreated

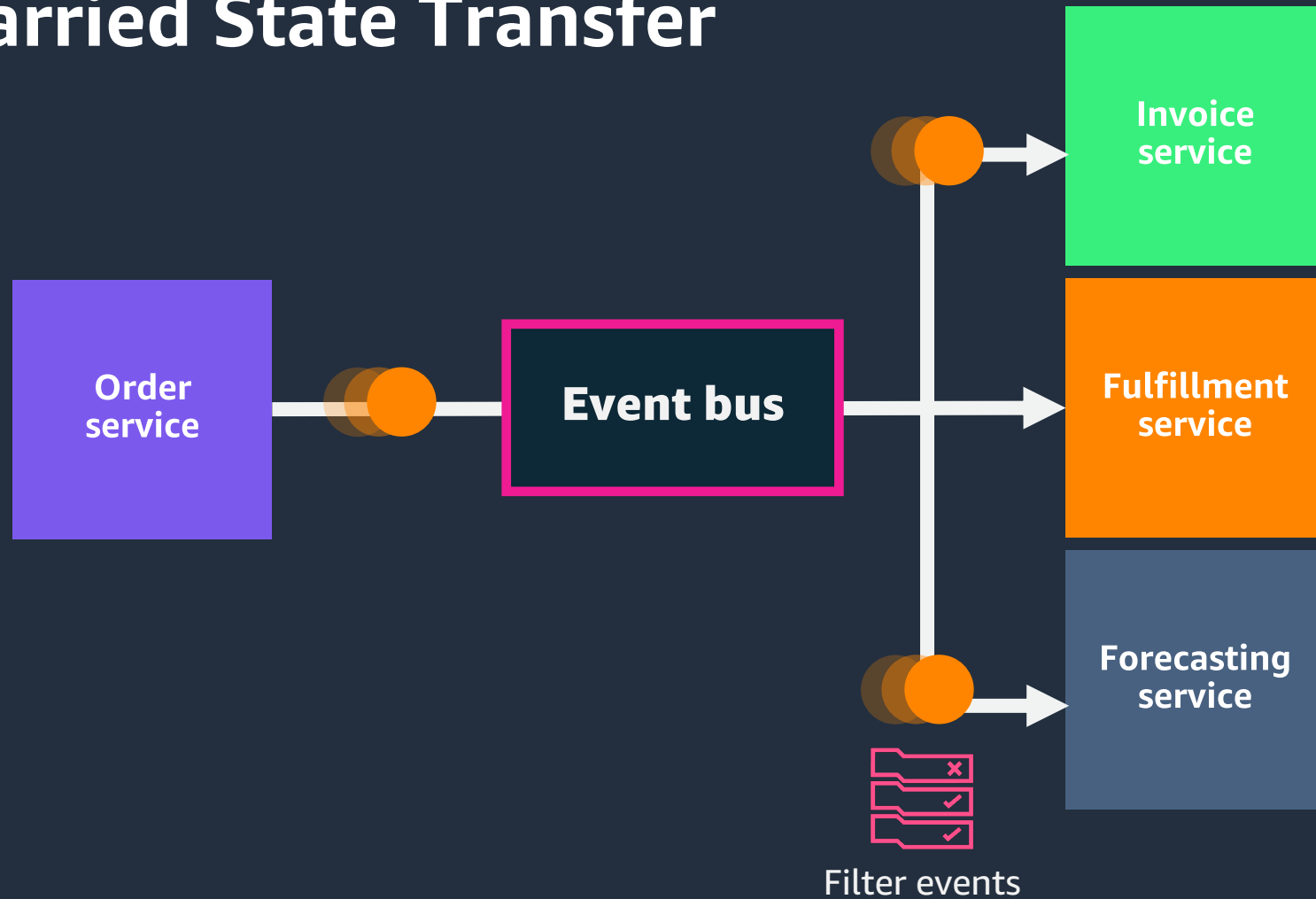
Order
service

```
{
  "version": "1",
  ...
  "detail": {
    "metadata": {
      "domain": "ORDERS"
    },
    "data": {
      "order": {
        "id": "3c947443-fd5f-4bfa-8a12-2aa348e793ae",
        "amount": 50,
        "deliveryAddress": {
          "postCode": "PE1111"
        }
      },
      "user": {
        "id": "09586e5c-9983-4111-8395-2ad5cfd3733b",
        "firstName": "Dave",
        "lastName": "Boyne",
        "email": "dboyne@dboyne.com"
      }
    }
  }
}
```

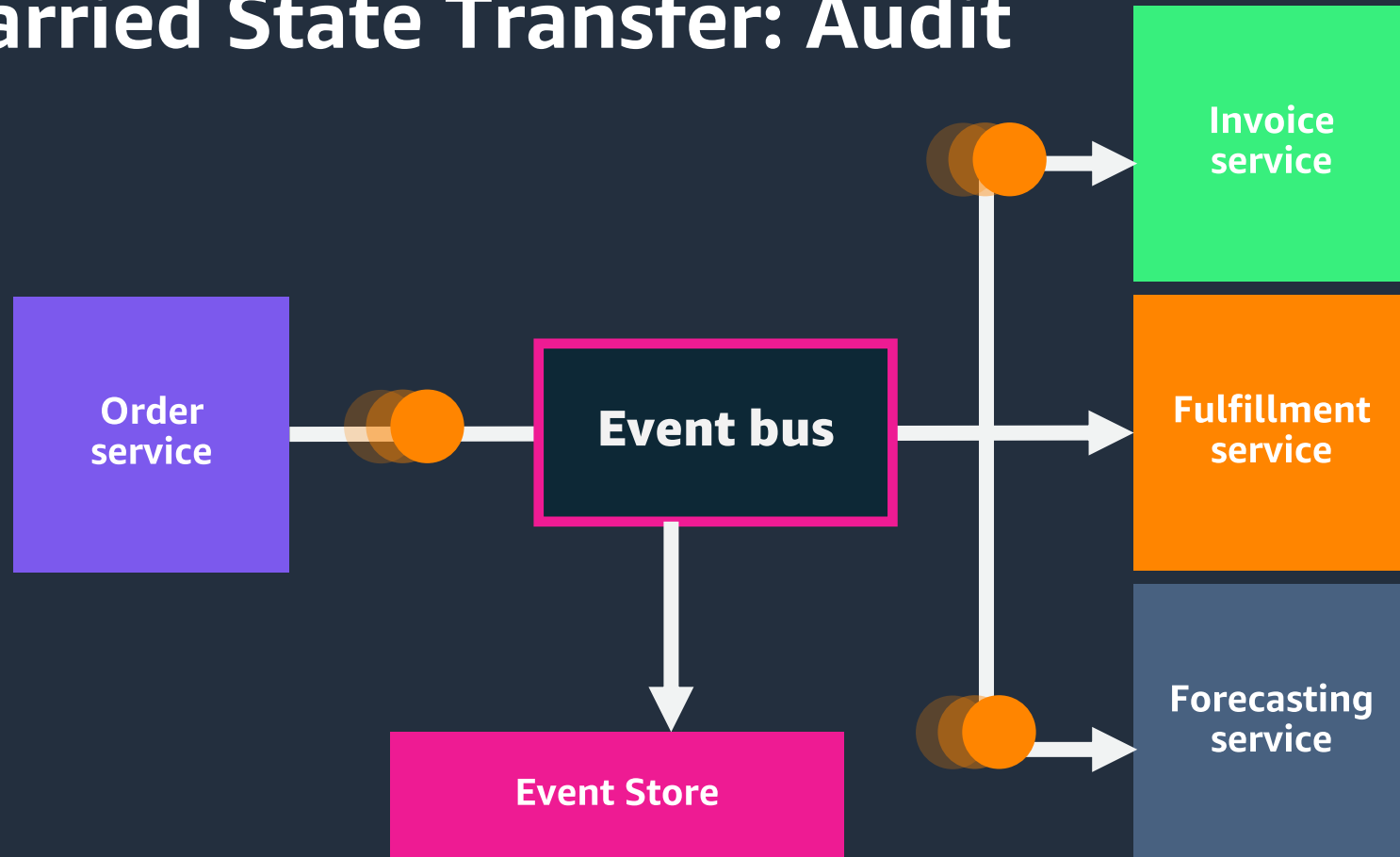
order information is sent

user information sent

Event-Carried State Transfer

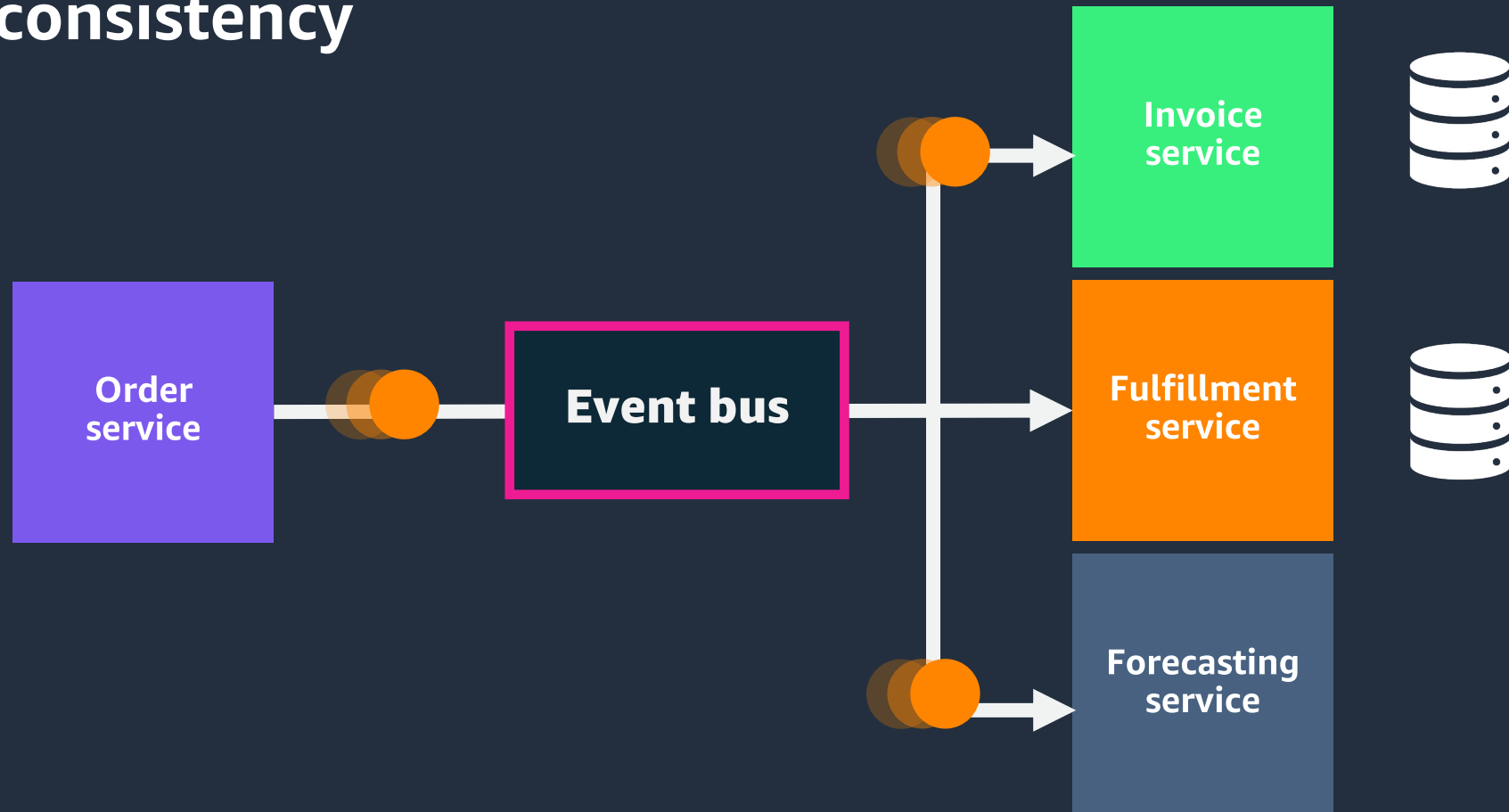


Event-Carried State Transfer: Audit



ECST events allow consumers to be **autonomous, flexible with filters / rules, enriched** but come with a cost...

Eventual consistency



Exposing implementation details

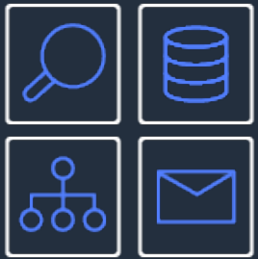
Order
service

```
{
  "version": "1",
  ...
  "detail": {
    "metadata": {
      "domain": "ORDERS"
    },
    "data": {
      "order": {
        "id": "3c947443-fd5f-4bfa-8a12-2aa348e793ae",
        "amount": 50,
        "deliveryAddress": {
          "postCode": "PE1111"
        }
      },
      "user": {
        "id": "09586e5c-9983-4111-8395-2ad5cfd3733b",
        "firstName": "Dave",
        "lastName": "Boyne",
        "email": "dboyne@dboyne.com"
      }
    }
  }
}
```

amount in what?, address details enough?

PII information

Summary of event carried state transfer



Contracts

*Events represent
the system contract*



Coupling

*With contracts vs
requests*



Push Model

*State pushed to
consumers*



Filtering

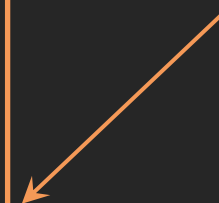
*Enriched rules on
event bus*

Additional event types : delta event

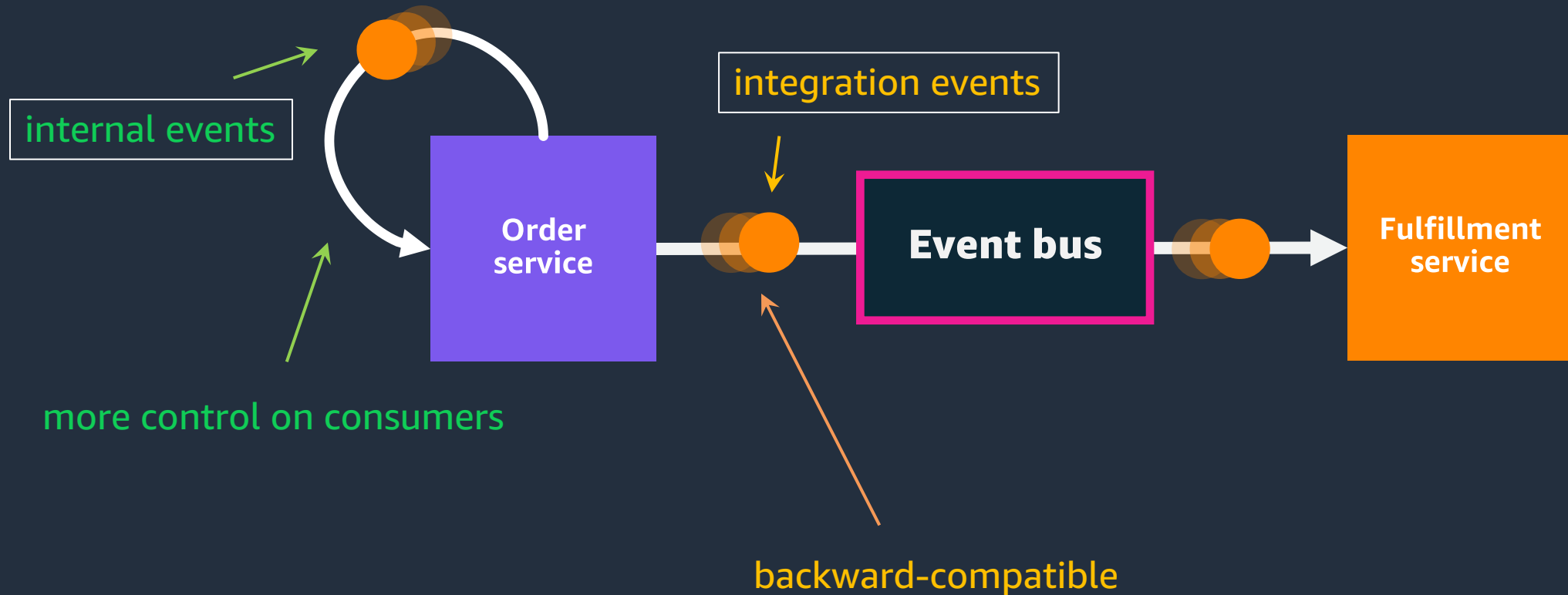
Order
service

```
{
  "version": "1",
  "id": "07fffe3b-4b25-48a7-b4a8-432bcc3bfb2c",
  "detail-type": "OrderUpdated",
  "source": "myapp.orders",
  "account": "123456789",
  "time": "2022-06-01T00:00:00Z",
  "region": "us-west-1",
  "detail": {
    "metadata": {
      "domain": "ORDERS"
    },
    "data": {
      "orderId": "3c947443-fd5f-4bfa-8a12-2aa348e793ae",
      "previous": {
        "amount": 50
      },
      "new": {
        "amount": 100
      }
    }
  }
}
```

Only send changes



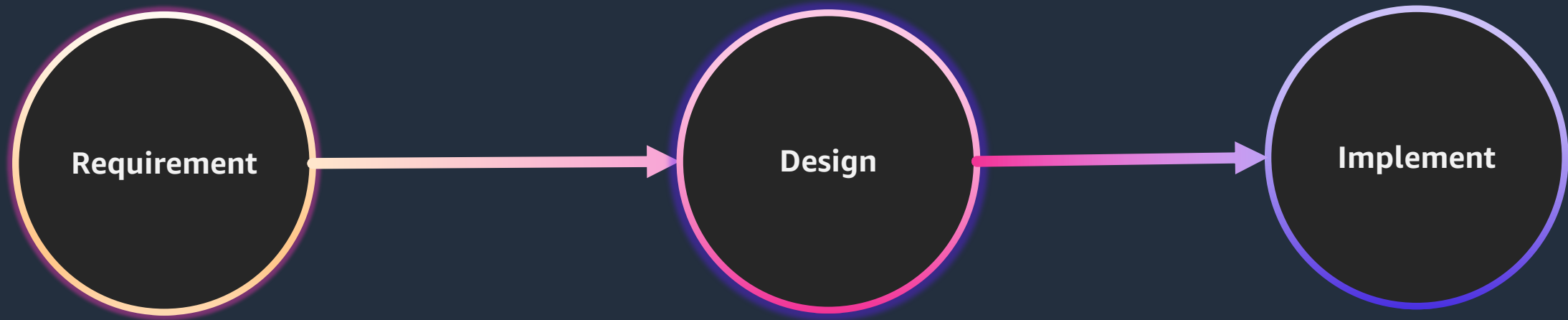
Understanding internal and integration events



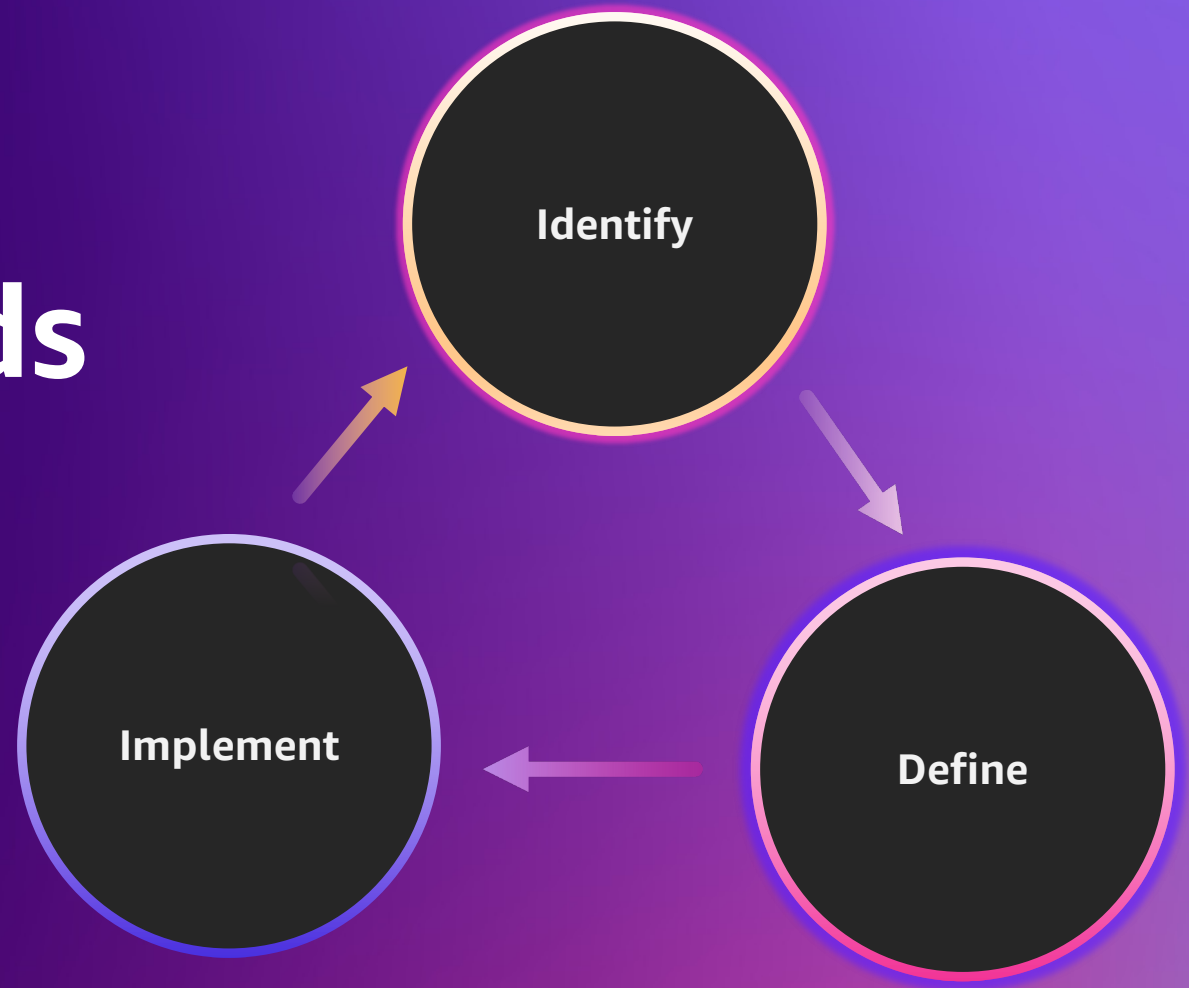
Which event pattern should you use?

3. Moving to event first thinking

Changing the way we design events

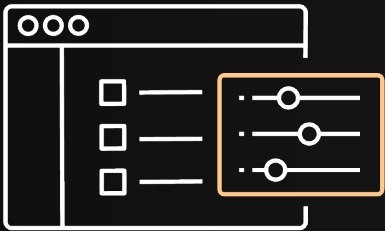


Moving towards event first thinking



Event first thinking: Identify

Identify



Behavior of system

*Understand system
behavior*



Event storming

*Collaborate with
domain experts*



Domain

*Identify bounded
context and domains*

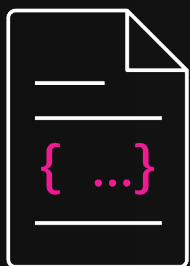


Ubiquitous language

*Speak the same
language*

Event first thinking: Define

Define



Event standards

Define standards in event payloads



Boundaries

Clear boundaries between domains



Documentation

Events, producers, and consumers



Naming Conventions

Align to clearly defined domains.

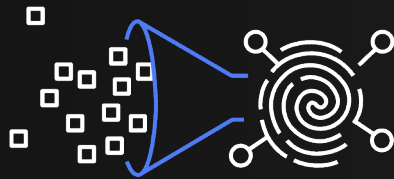
Event first thinking: Implement

Implement



Libraries

Develop shared code to help



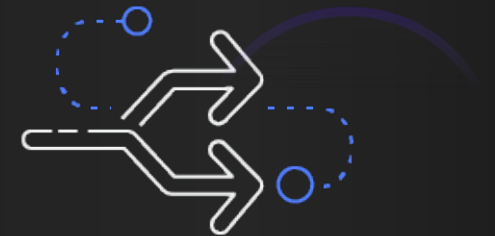
Event Patterns

Consider tradeoffs of design choices



Domain vs Integration

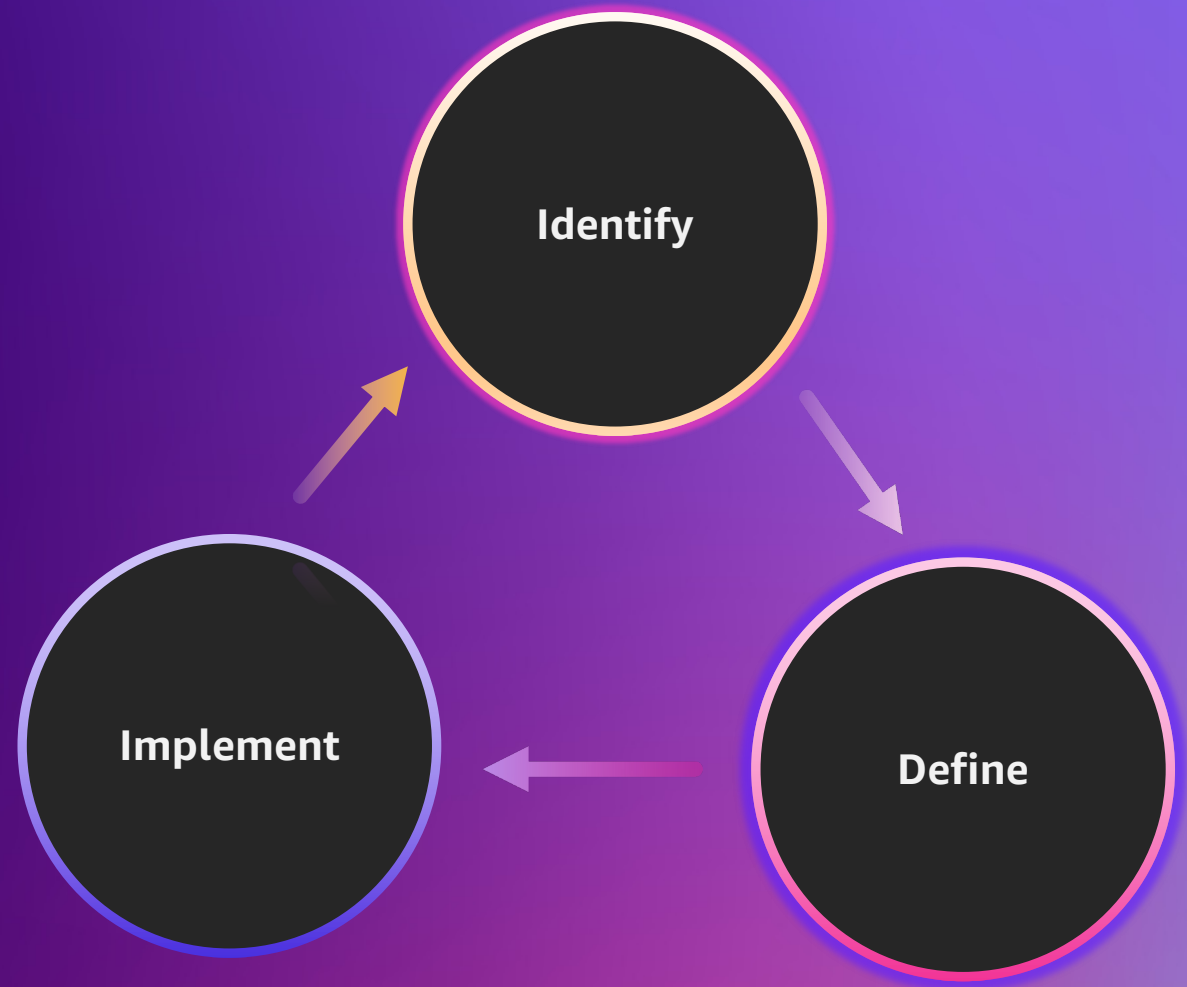
Internal vs external events



Domain Events

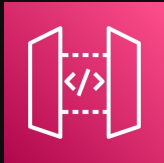
Events that the business understands

Scale EDA with event first thinking

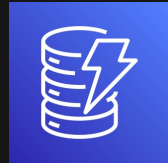


Summary

Adoption and accessibility of event-driven architecture



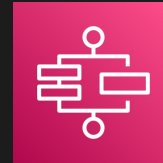
Amazon
API Gateway



Amazon
DynamoDB



Amazon
EventBridge

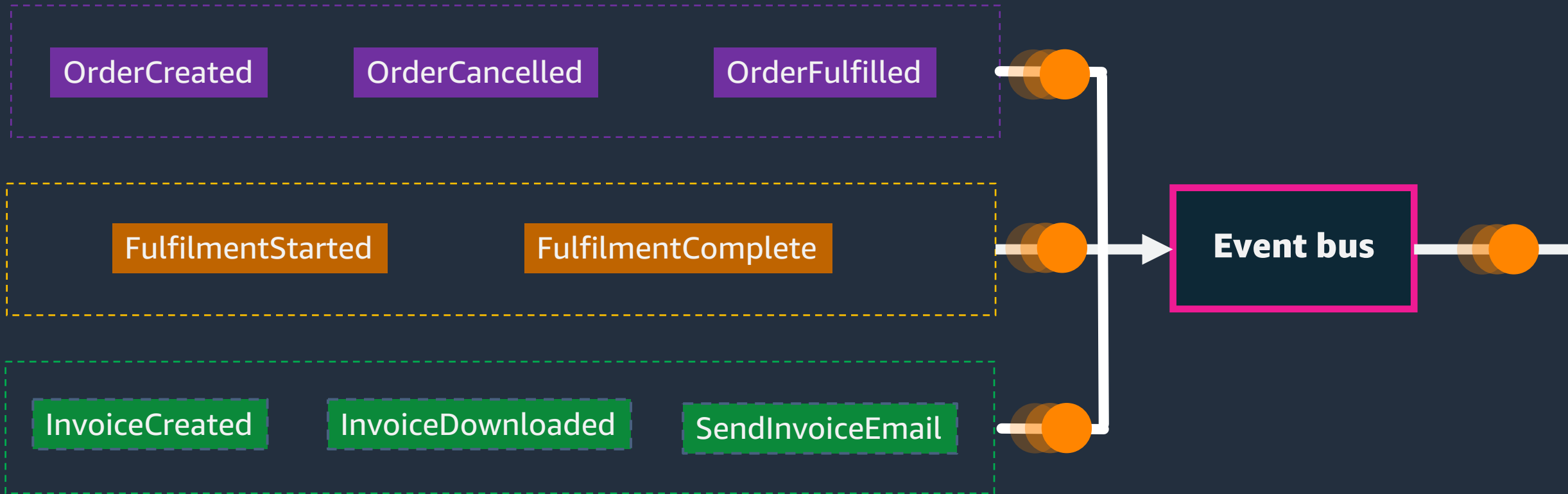


AWS Step
Functions

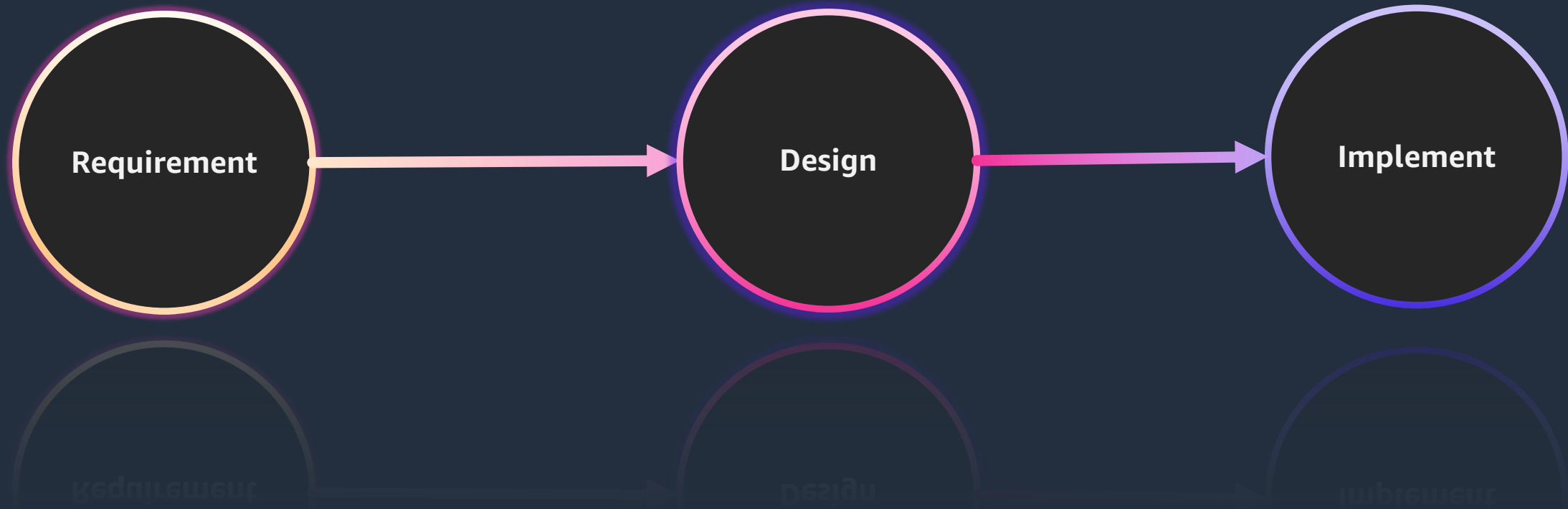


AWS
Lambda

Problems as we start scaling our patterns



Review the way we think about events



How to scale your event design?

Identify

Behavior of system

Event storming

Ubiquitous language

Define

Event standards

Documentation

Naming conventions

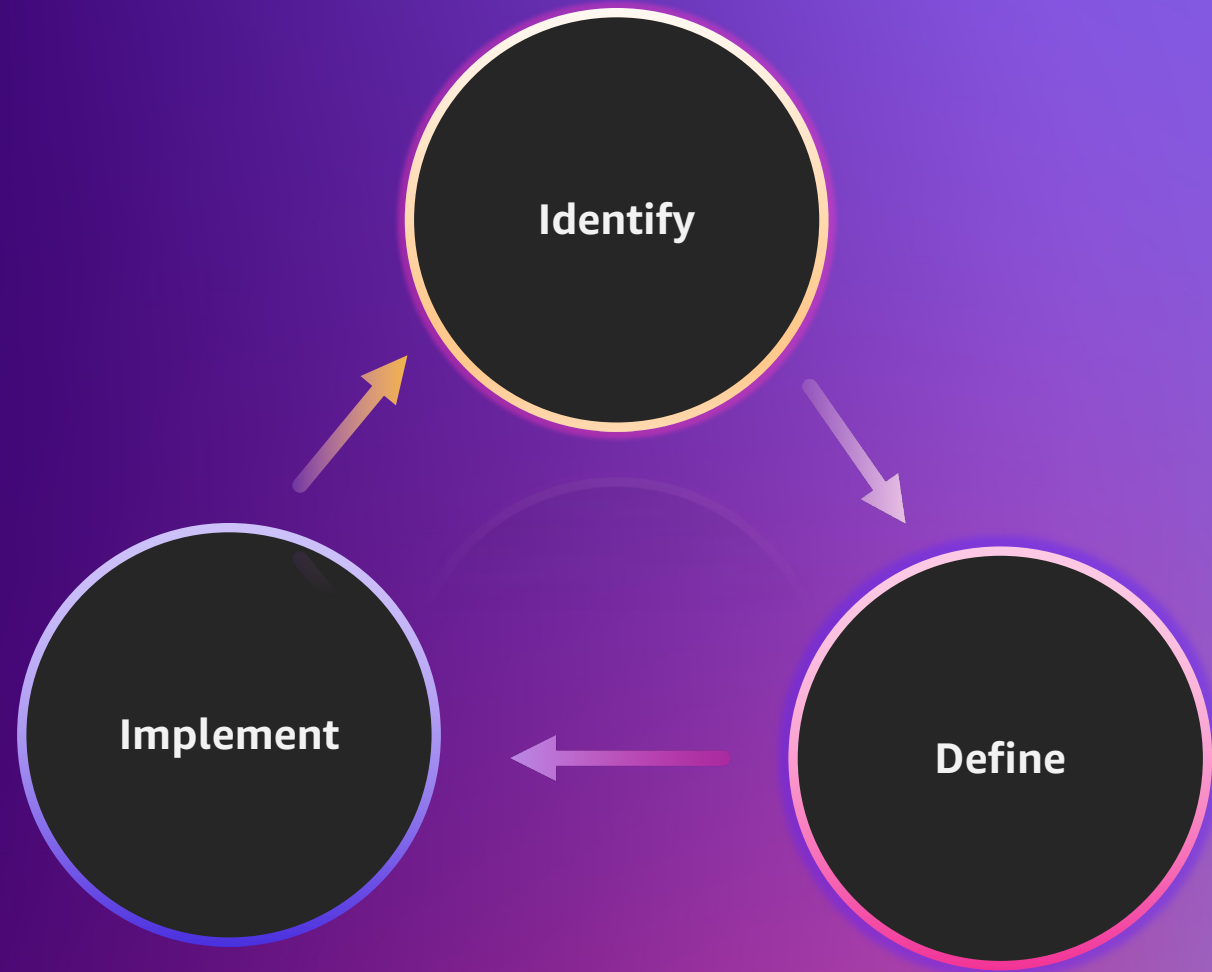
Implement

Libraries

Domain Events

Event patterns

Event first thinking



Resources



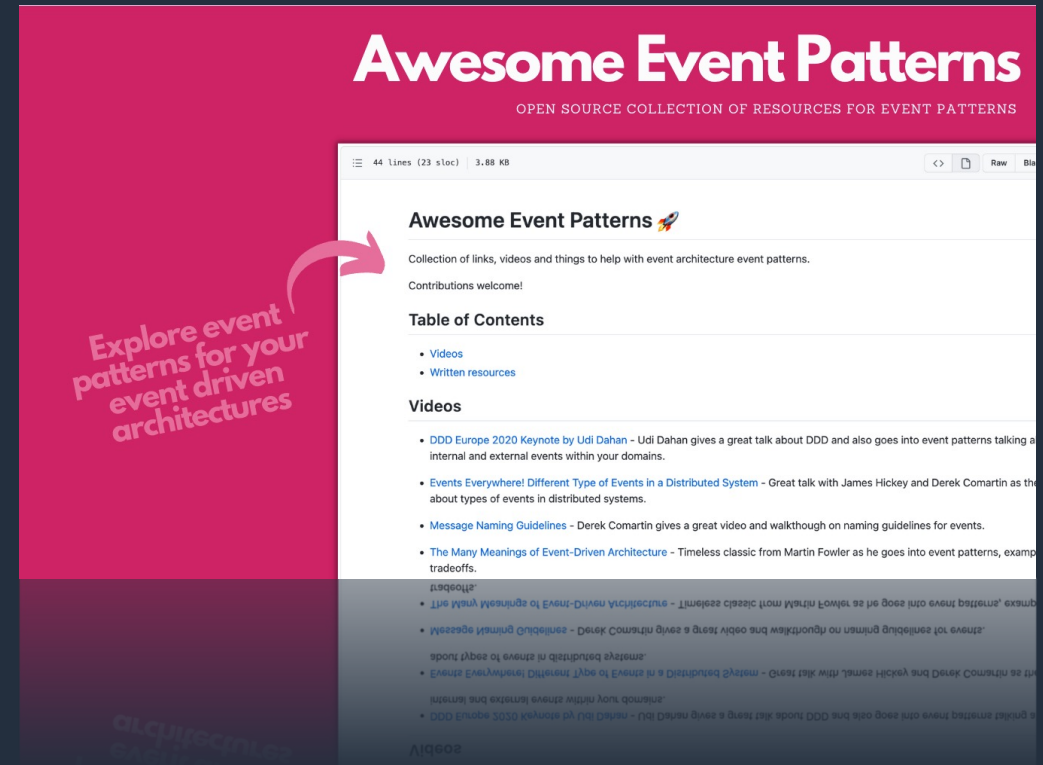
[Awesome Event Patterns](#)

Collection of videos and reading material around event patterns



[Awesome EventBridge](#)

Videos, Tutorials, Tools, Patterns for Amazon EventBridge





GOTO
Guide

★★★★★

Remember to **rate the session**



David Boyne

 @boyney123

THANK YOU





Thank you!

David Boyne

 @boyney123