Decouple and Scale Applications Using Amazon SQS and Amazon SNS

Modern Apps

Compute

Database

Messaging

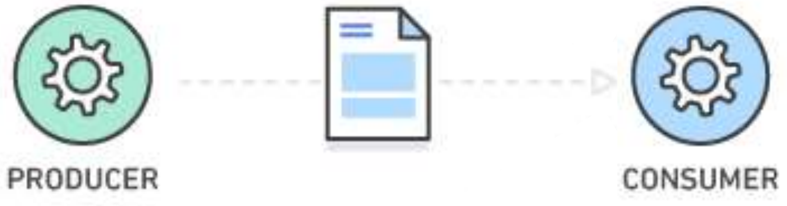
Modern Apps

Compute

Database

Messaging

Message



JSON Message

```json
{
    "bookingId": "456ab773-dccb-4bc9-87b7-322ff5c29eab",
    "bookingNumber": "CDG-64453",
    "locationId": "563890",
    "customer": {
        "id": "8943",
        "email": "jacque@gmail.com"
    },
    "stayStart": "2017-09-04",
    "stayEnd": "2017-09-06",
    "price": {
        "amount": "67.80",
        "currency": "EUR"
    }
}
```

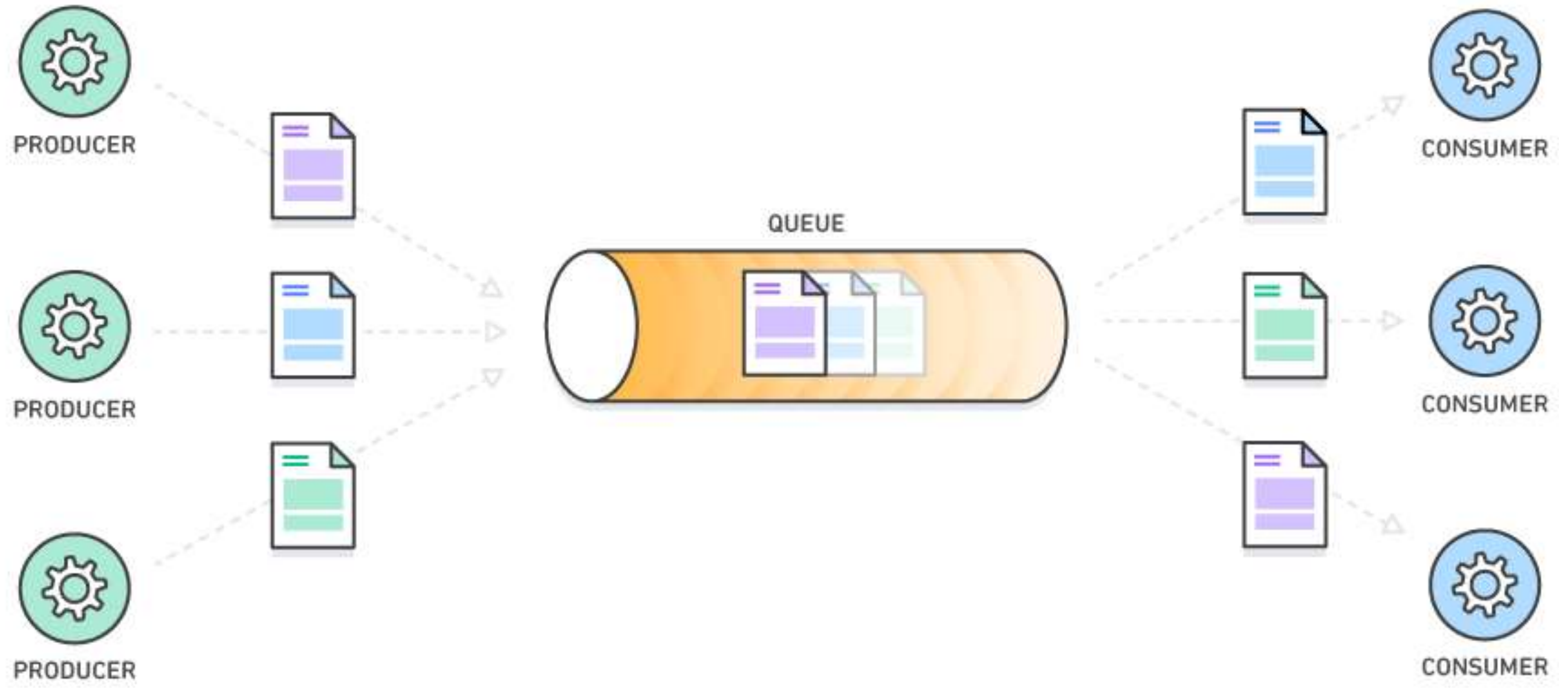
XML Message

```xml
<booking>
    <bookingId>456ab773-dccb-4bc9-87b7-322ff5c29eab</bookingId>
    <bookingNumber>CDG-64453</bookingNumber>
    <locationId>563890</locationId>
    <customer>
        <id>8943</id>
        <email>jacque@gmail.com</email>
    </customer>
    <stayStart>2017-09-04</stayStart>
    <stayEnd>2017-09-06</stayEnd>
    <price>
        <amount>67.80</amount>
        <currency>EUR</currency>
    </price>
</booking>
```

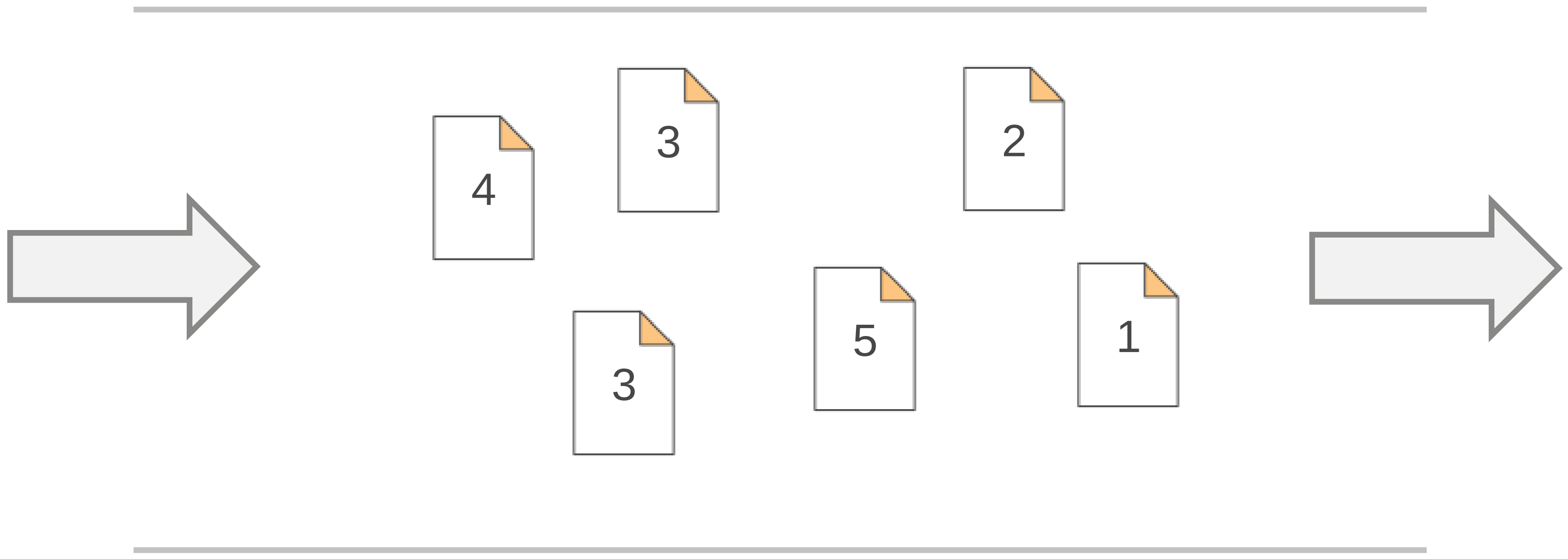
Payload and Attributes

- Key/value pairs
- With business meaning:
  - `CustomerId=6445`
  - `MessageType=NewBooking`
- With technical meaning:
  - `SourceHost=ip-12-34-56-78.us-west-2.compute.internal`
  - `ProgramName=WebServer-PID:9989`

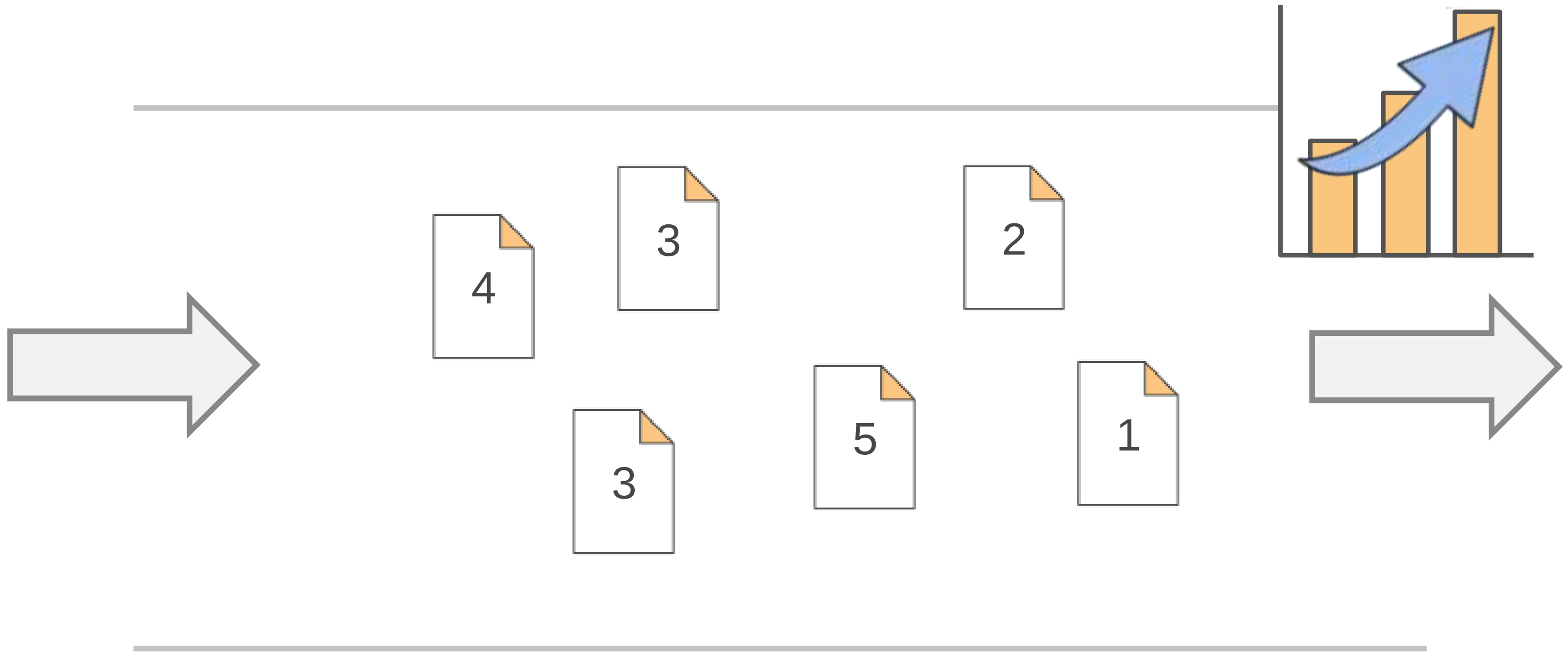
Queues


PRODUCER
PRODUCER
PRODUCER
QUEUE
CONSUMER
CONSUMER
CONSUMER

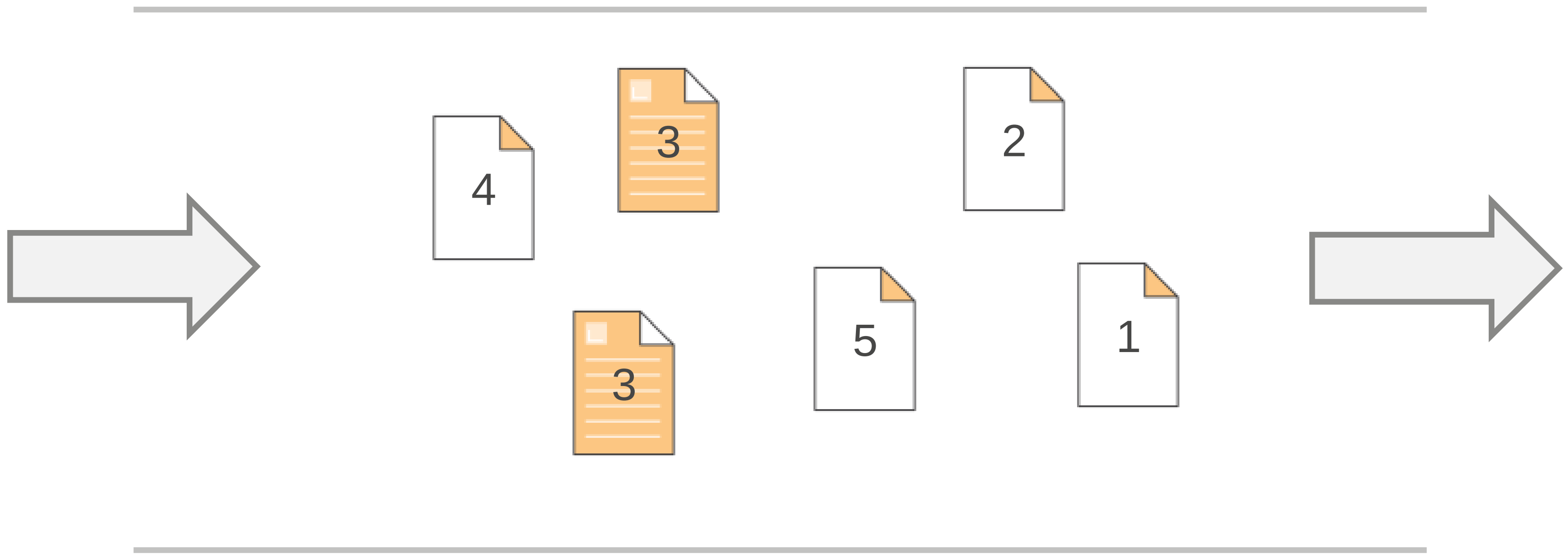
Amazon Simple Queue Service (Amazon SQS): Standard Queue



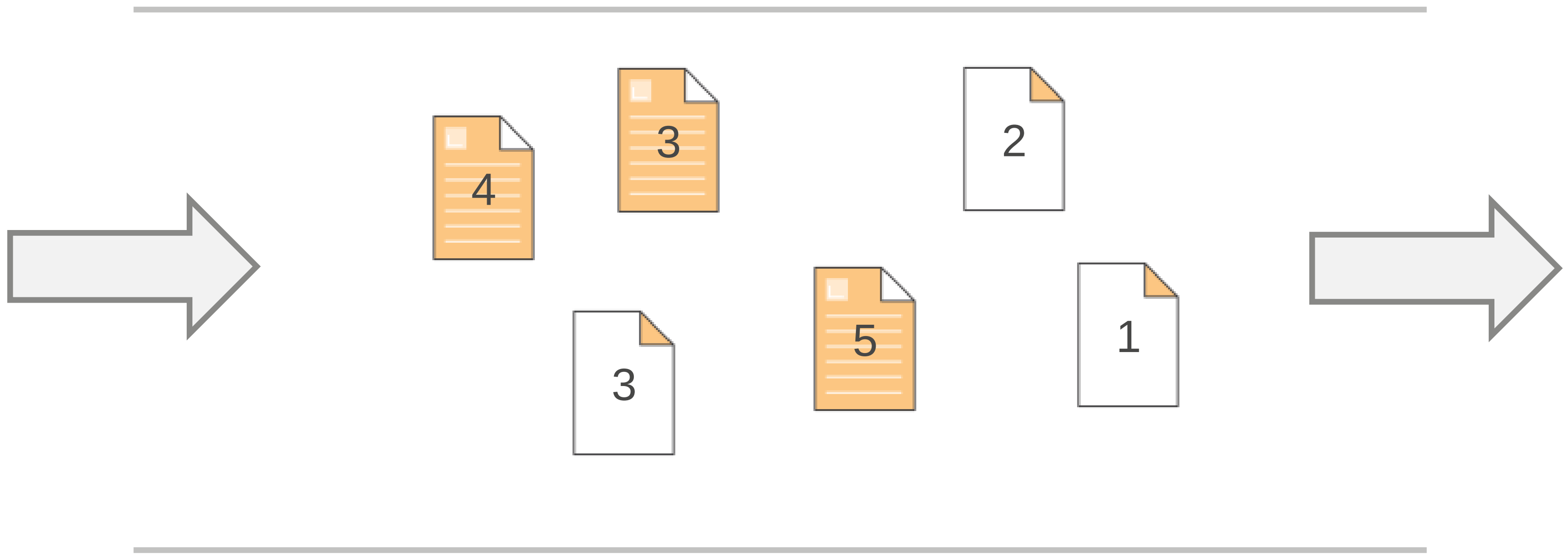
Amazon Simple Queue Service (Amazon SQS): Standard Queue



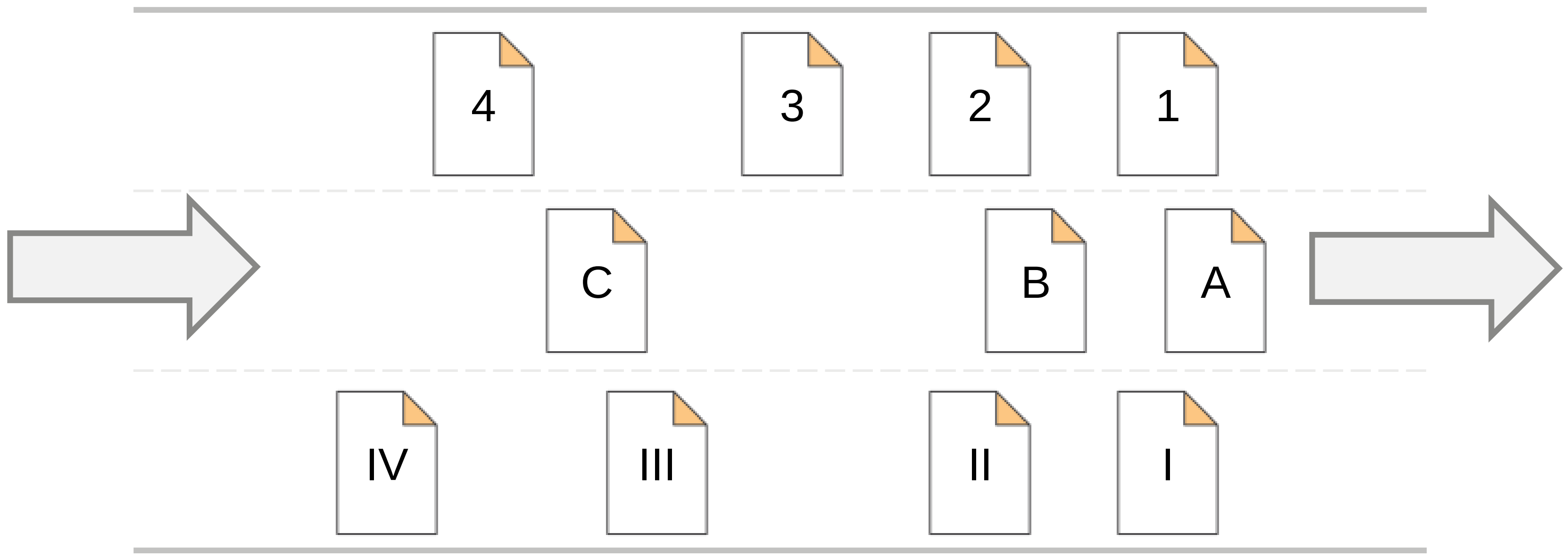
Amazon Simple Queue Service (Amazon SQS): Standard Queue



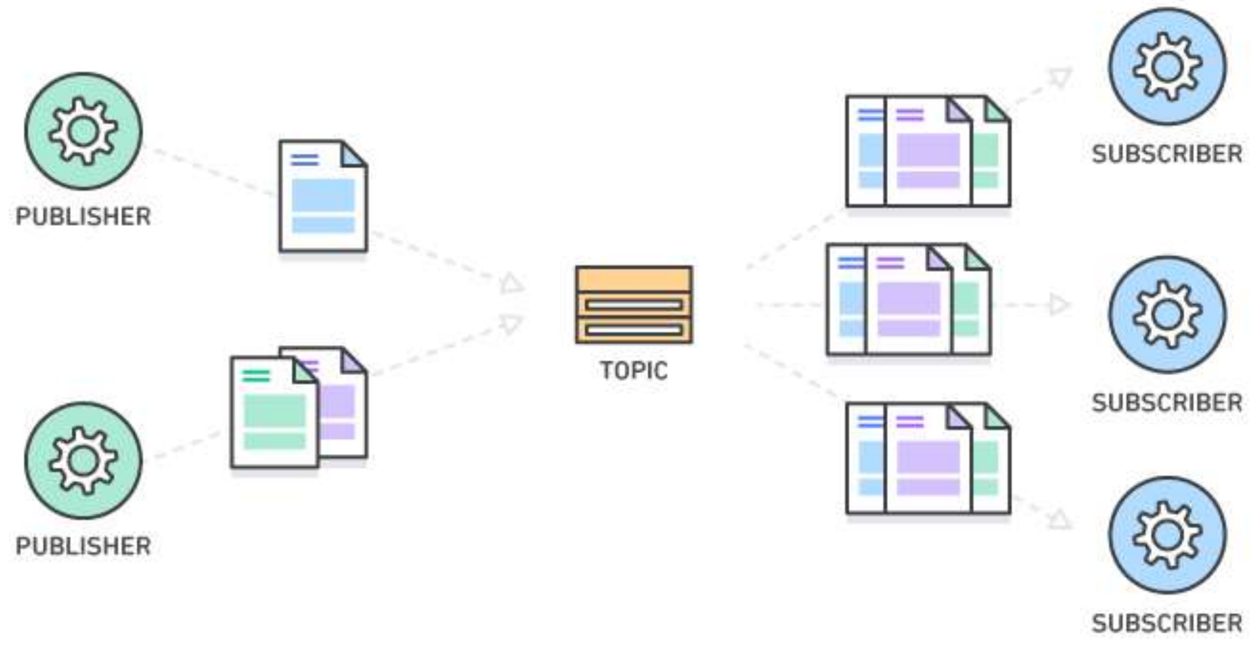
Amazon Simple Queue Service (Amazon SQS): Standard Queue



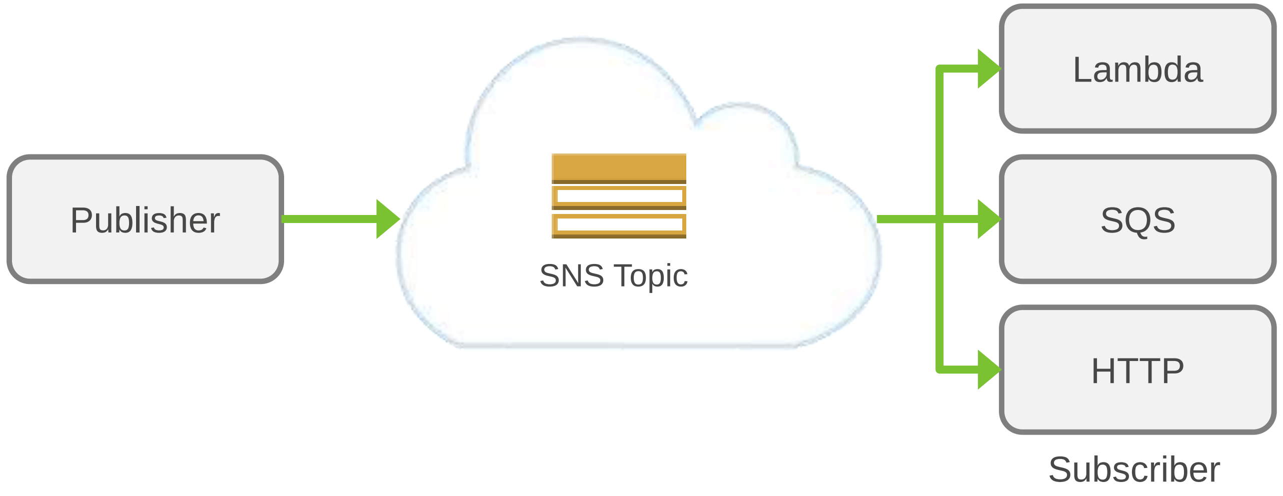
Amazon Simple Queue Service (Amazon SQS): FIFO Queue



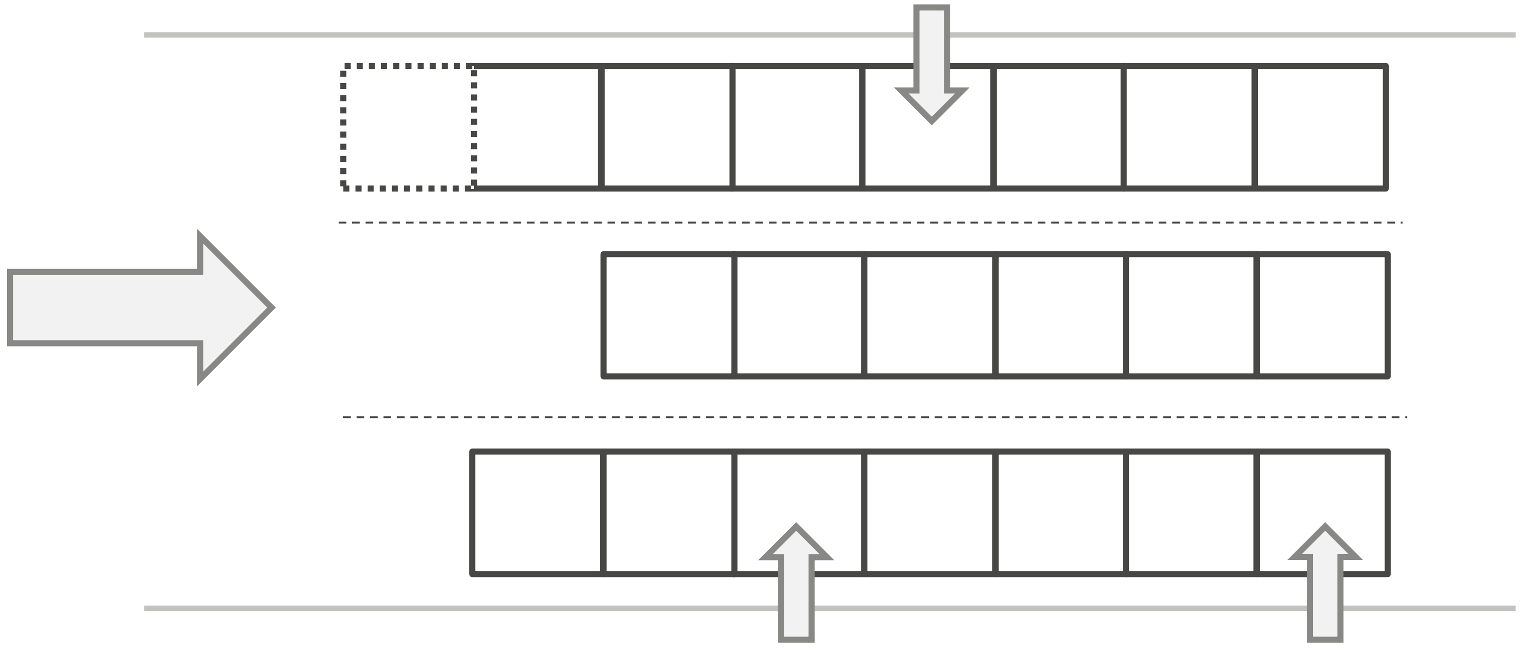
Pub/Sub Messaging



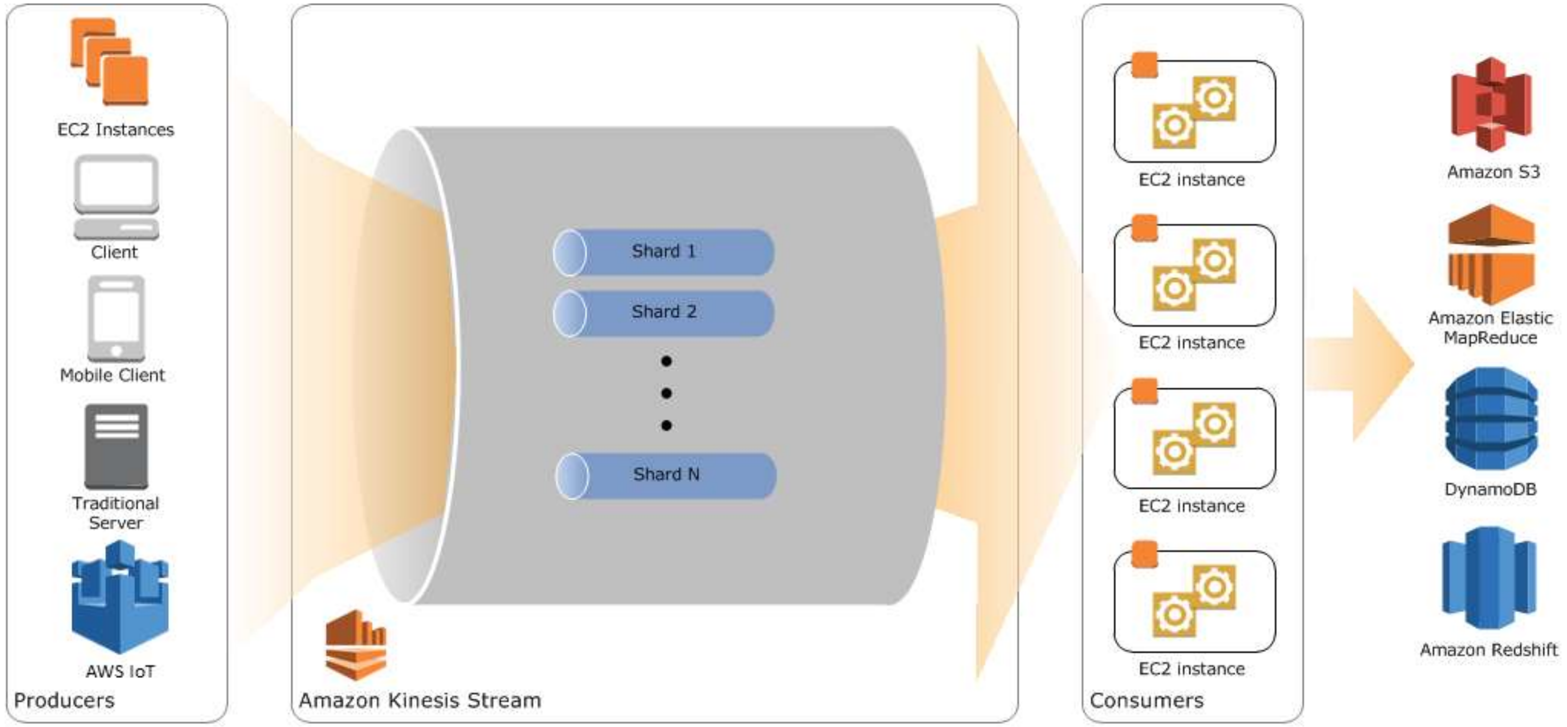
Amazon Simple Notification Service (Amazon SNS)



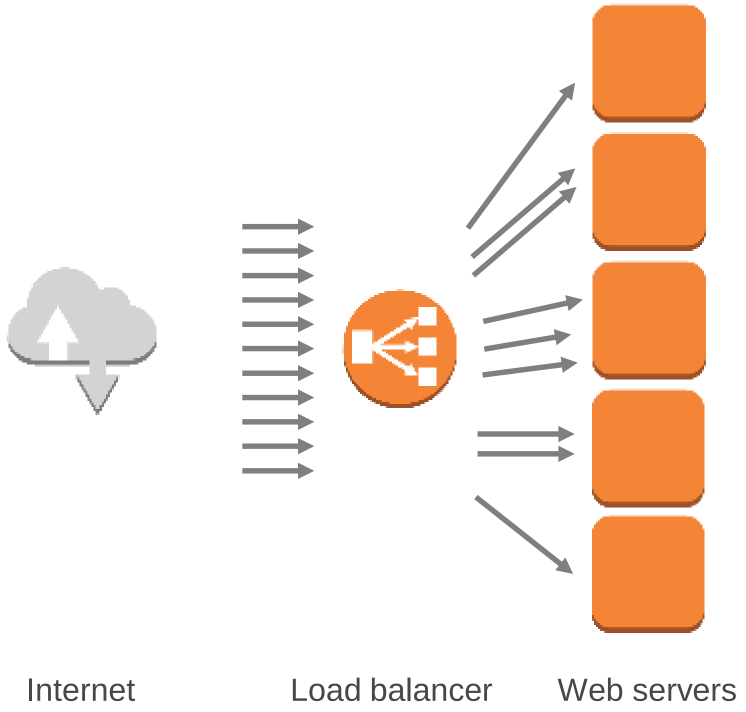
Streams



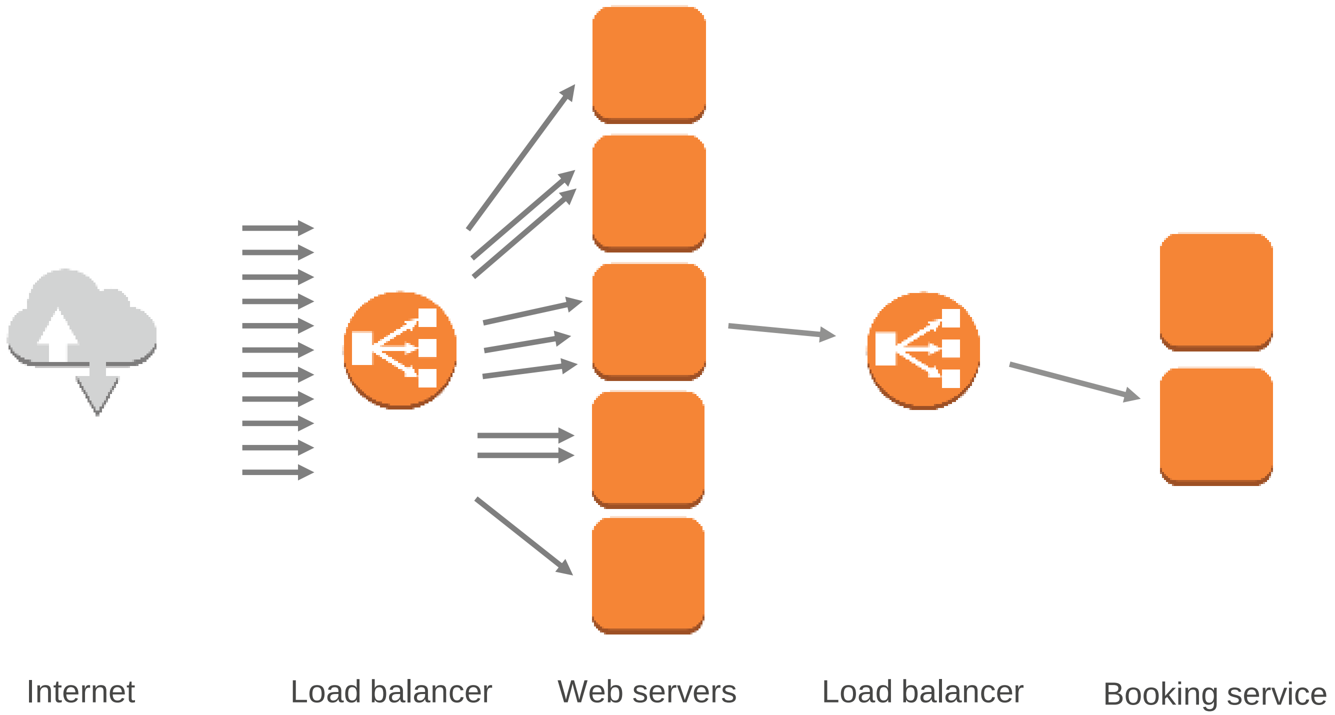
Amazon Kinesis



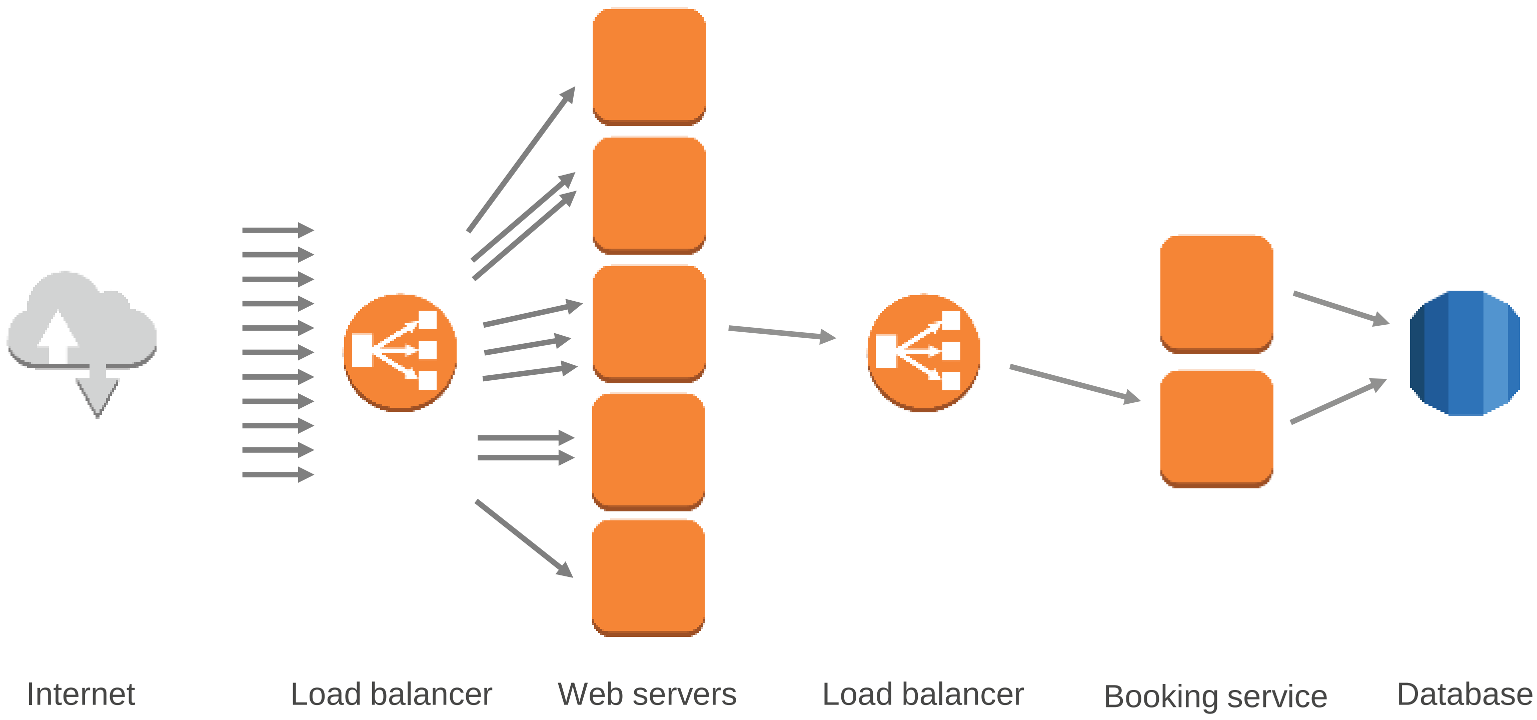
Service to Service Communication



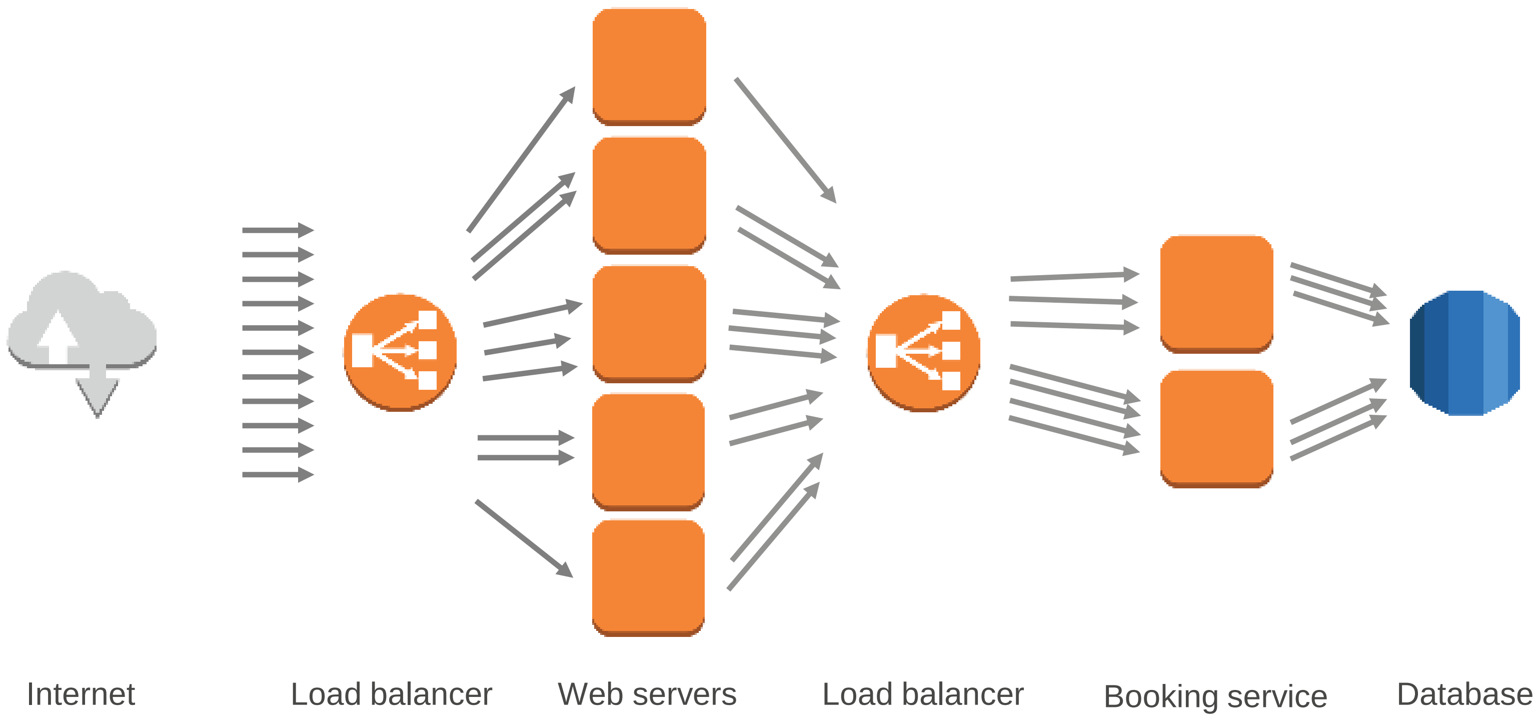
Service to Service Communication



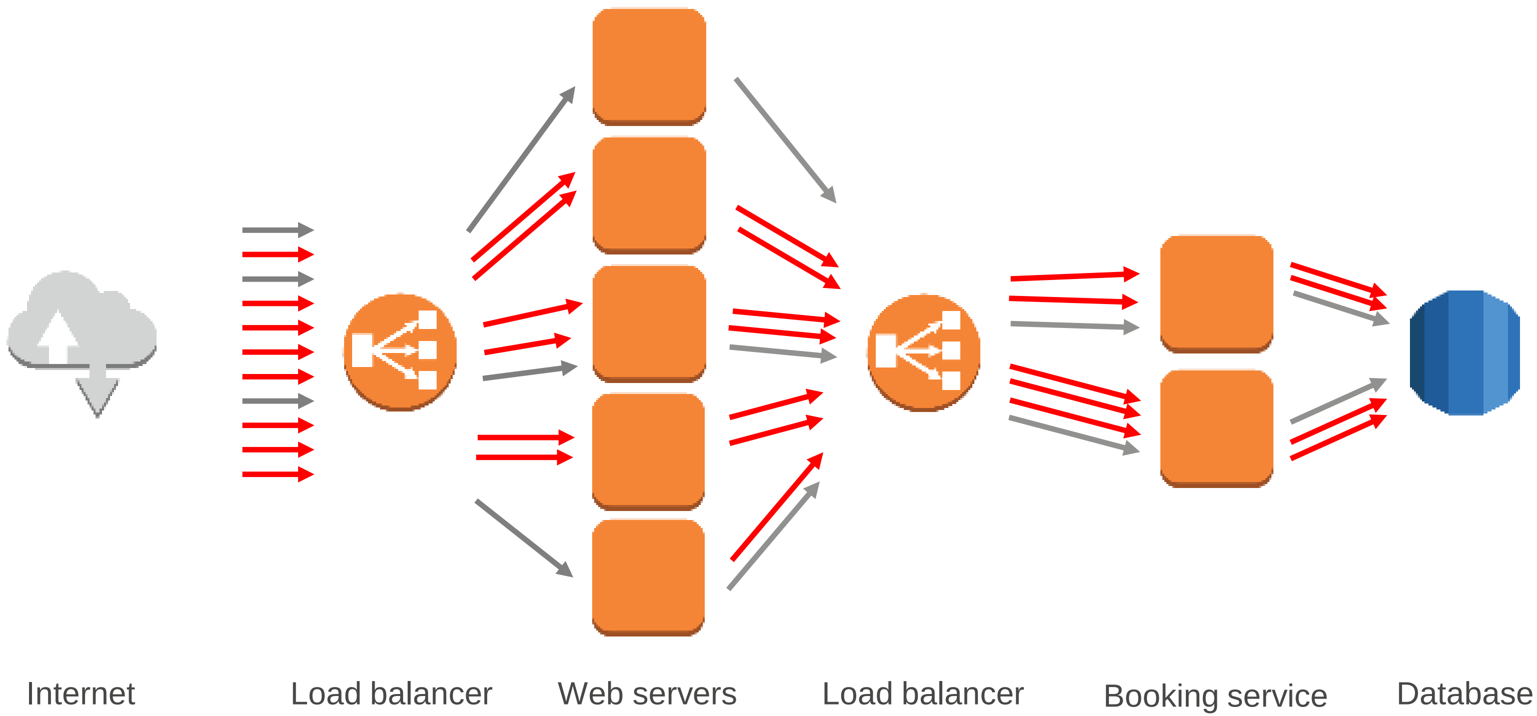
Service to Service Communication



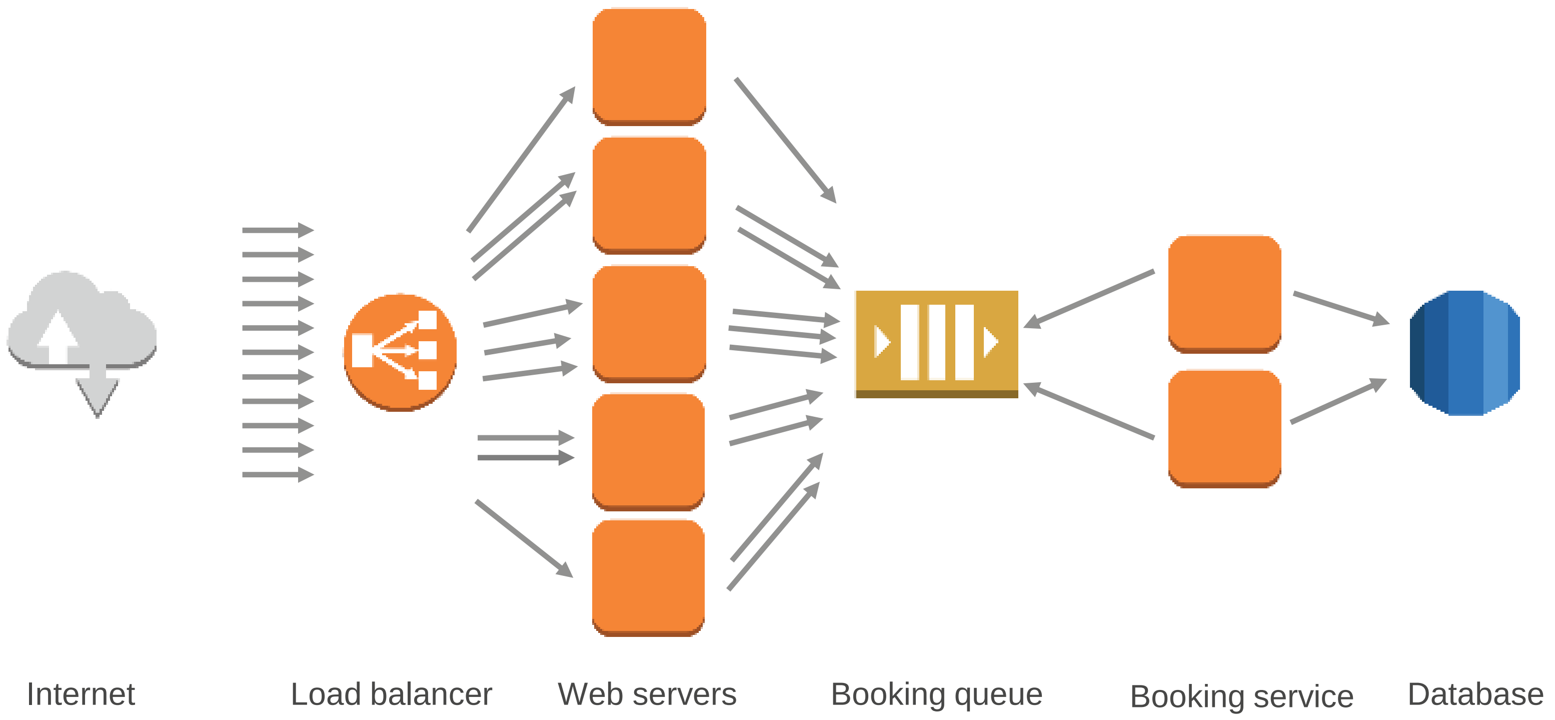
Capacity Imbalance



Overload


Internet
Load balancer
Web servers
Load balancer
Booking service
Database

Queue as a Safe and Fast Buffer



Demo

Improving Synchronous Latency

Booking takes up to 3 seconds, it’s too slow!

Improving Synchronous Latency

Booking takes up to 3 seconds, it’s too slow!

Let’s break it down:

Change status in database	30 ms
Notify external booking supplier	800 ms
Prepare a PDF invoice	900 ms
Send a confirmation e-mail with large PDF	500 ms

Use Background Thread

Synchronous:

Change status in database	30 ms

Background thread:

Notify external booking supplier	800 ms
Prepare a PDF invoice	900 ms
Send a confirmation e-mail with large PDF	500 ms

Use Background Thread

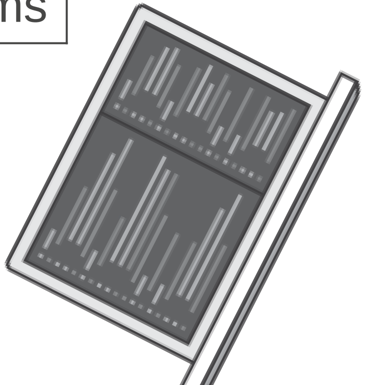
Synchronous:

Change status in database	30 ms

Background thread:

Notify external booking supplier	800 ms
Prepare a PDF invoice	900 ms
Send a confirmation e-mail with large PDF	500 ms

Work can get lost!



Safe Asynchronous Tasks

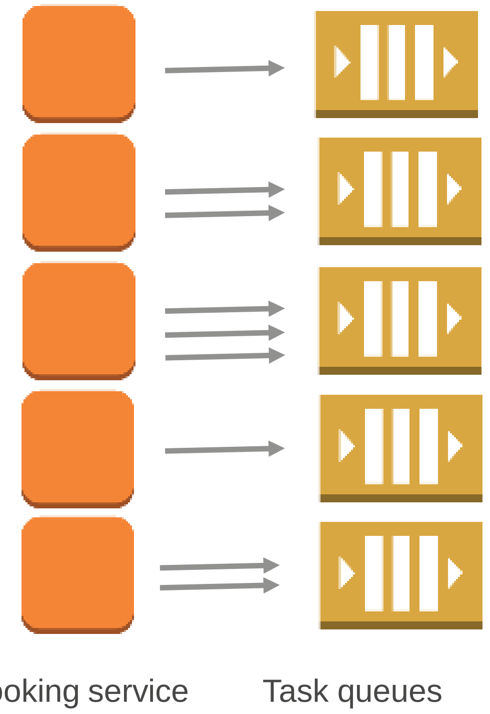
Synchronous:

Change status in database	30 ms
Store task in queue	**10 ms**

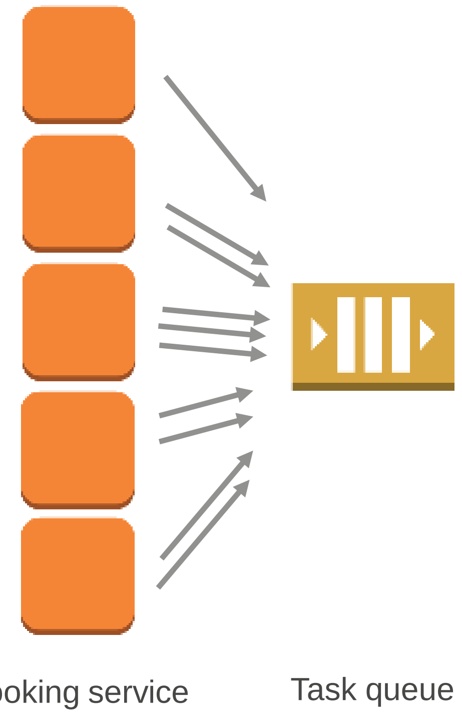
Background poller:

Get next task from queue	**10 ms**
Notify external booking supplier	800 ms
Prepare a PDF invoice	900 ms
Send a confirmation e-mail with large PDF	500 ms
Delete task from queue	**10 ms**

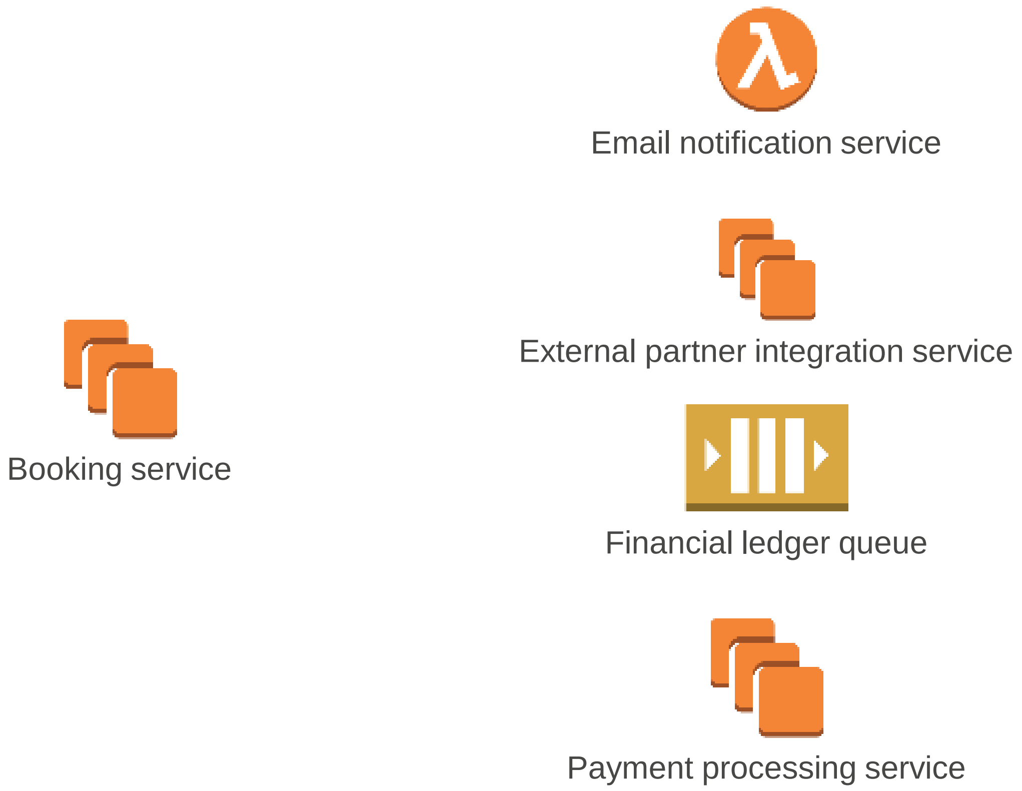
Private Task Queues



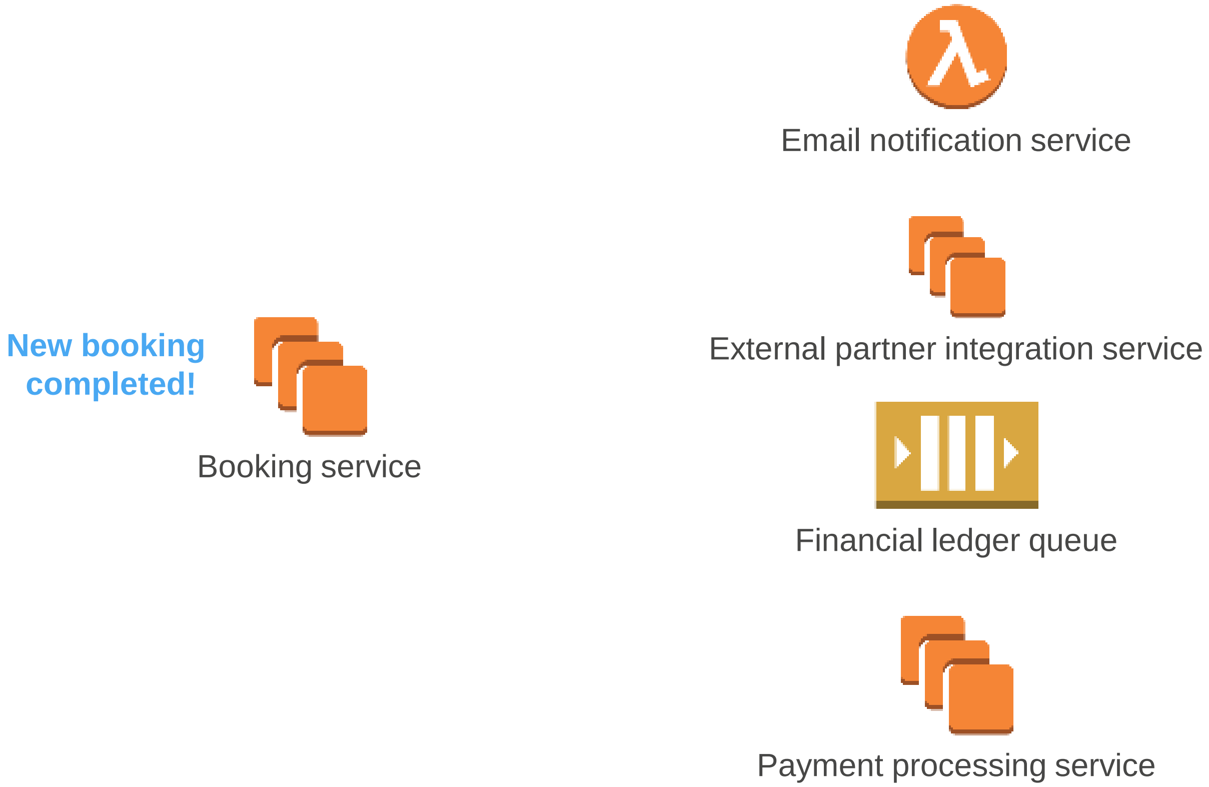
Shared Task Queue



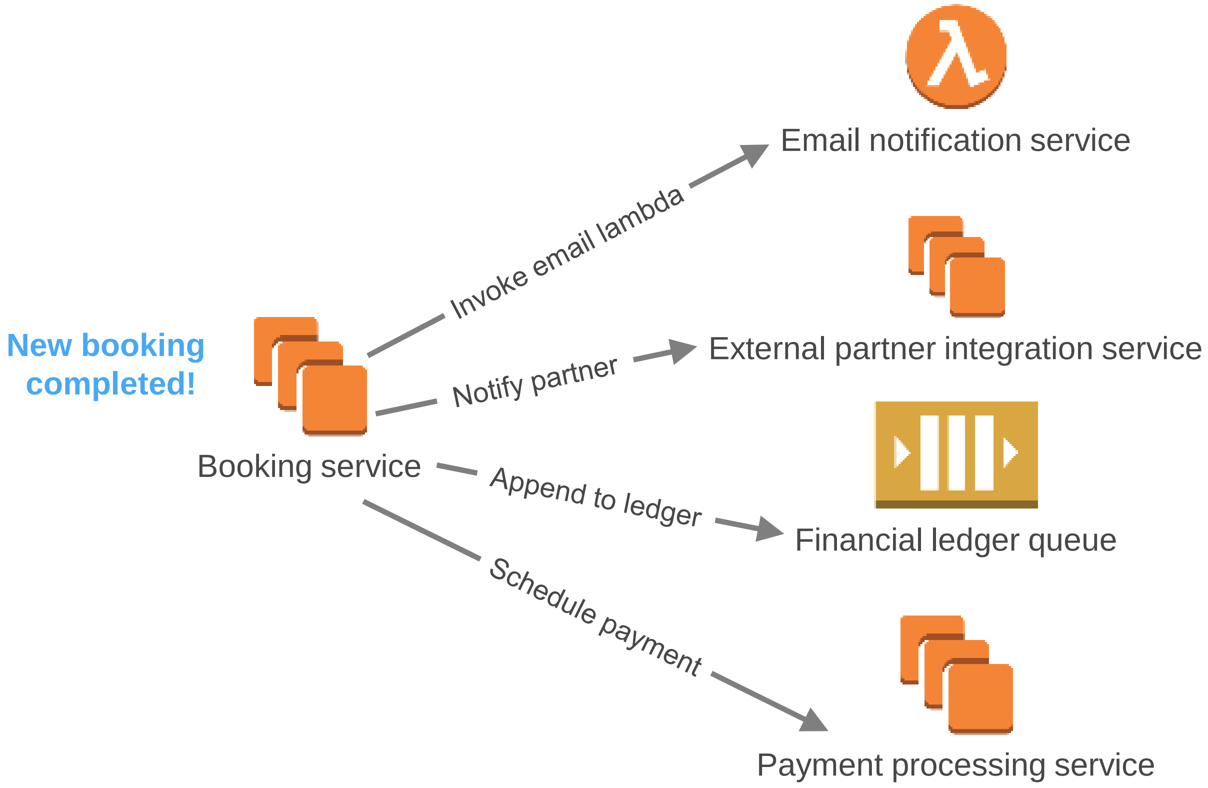
Microservices


Email notification service
External partner integration service
Booking service
Financial ledger queue
Payment processing service

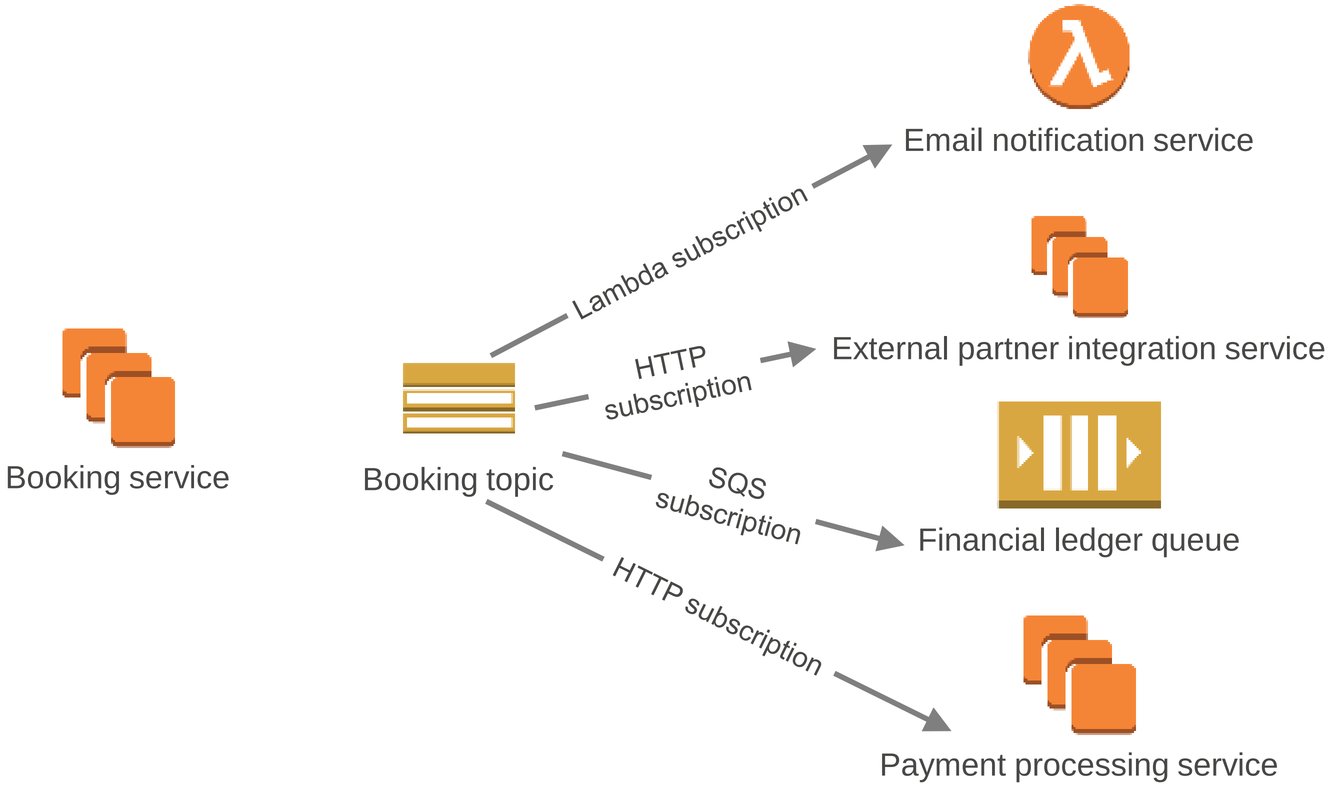
Notify Everyone!


New booking completed!
Booking service
Email notification service
External partner integration service
Financial ledger queue
Payment processing service

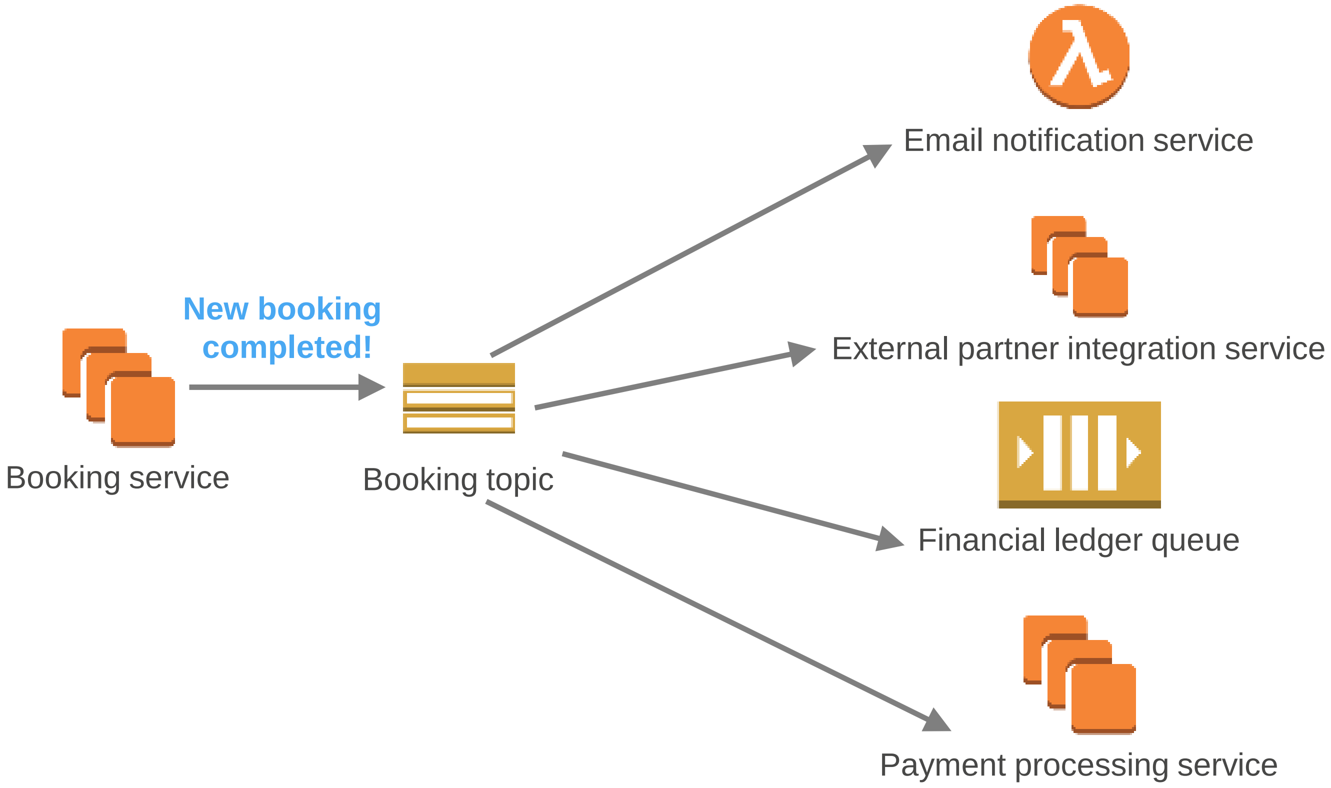
Notify Everyone!



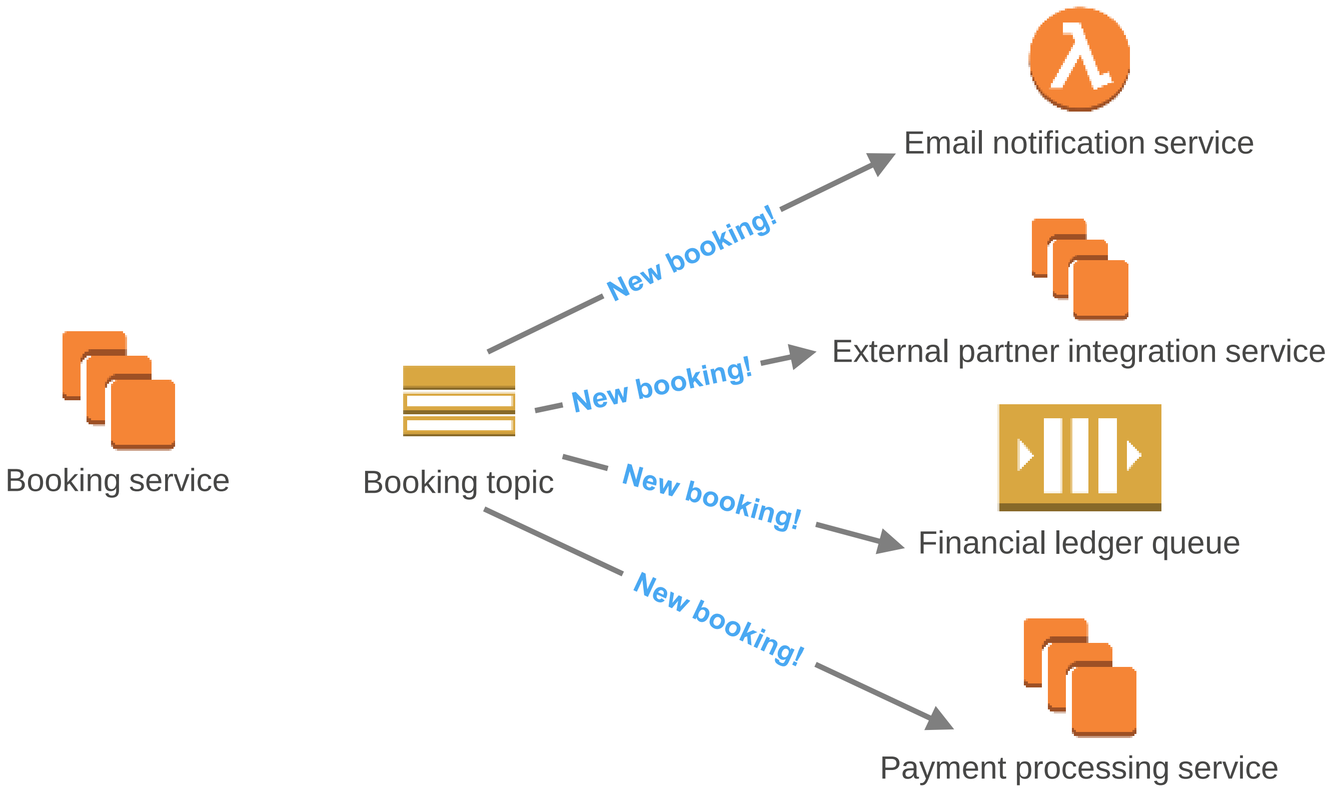
Decouple by Publishing Event through SNS



Decouple by Publishing Event through SNS


New booking completed!
Booking service
Booking topic
Email notification service
External partner integration service
Financial ledger queue
Payment processing service

Decouple by Publishing Event through SNS



There’s more! Queues in SQS

Long polling: instant push deliveries of messages

There’s more! Queues in SQS

Long polling: instant push deliveries of messages

Server side encryption

There’s more! Queues in SQS

Long polling: instant push deliveries of messages

Server side encryption

Dead letter queues

There’s more! Queues in SQS

Long polling: instant push deliveries of messages

Server side encryption

Dead letter queues

Easy monitoring with CloudWatch, with alarming

There's more! Queues in SQS

Long polling: instant push deliveries of messages

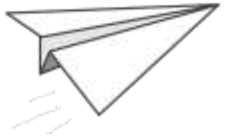
Server side encryption

Dead letter queues

Easy monitoring with CloudWatch, with alarming

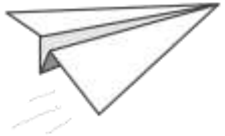
Integrated with other AWS services as a destination

There’s more! Pub/Sub messaging in SNS

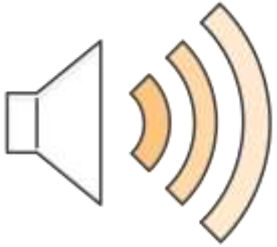


Multiple transports

There's more! Pub/Sub messaging in SNS



Multiple transports



Customizable delivery retries for HTTP

There’s more! Pub/Sub messaging in SNS

Multiple transports

Customizable delivery retries for HTTP

Failure notifications

There's more! Pub/Sub messaging in SNS

Multiple transports

Customizable delivery retries for HTTP

Failure notifications

Easy monitoring with CloudWatch, with alarming

There’s more! Pub/Sub messaging in SNS

Multiple transports

Customizable delivery retries for HTTP

Failure notifications

Easy monitoring with CloudWatch, with alarming

Integrated with other AWS services as a destination

Enterprises using Amazon SQS and SNS

HEARST
Capital One
News Corp
SUNCORP GROUP One Company Many Brands
SAMSUNG

THOMSON REUTERS
Nintendo
Adobe
The Weather Company
vodafone
NOVARTIS

amazon
The Washington Post
NETFLIX
NTT docomo
AUTODESK
The New York Times

theguardian
Expedia
EA
NOKIA
Earth Networks

BMW
COMCAST
mlbam
FT FINANCIAL TIMES
Johnson & Johnson
Nordstrom

For more information

www.aws.amazon.com/sqs

www.aws.amazon.com/sns

Thank you!