

Poison Messages, Dead-Lettering & Error Handling



Alan Smith

Active Solution

youtube.com/user/CloudCasts

Overview



Communication Problems

Transient Fault Handling

Dead-Lettering

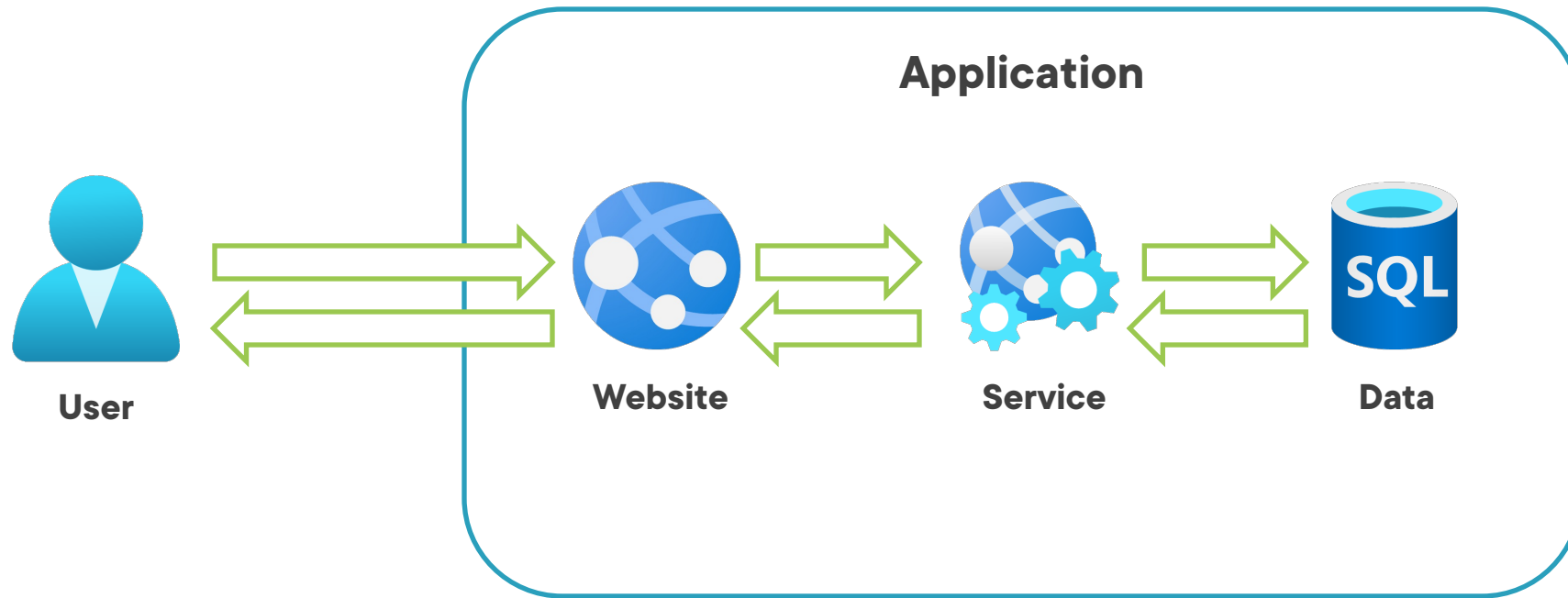
Demo: Dead-Lettering Messages

Handling Poison Messages

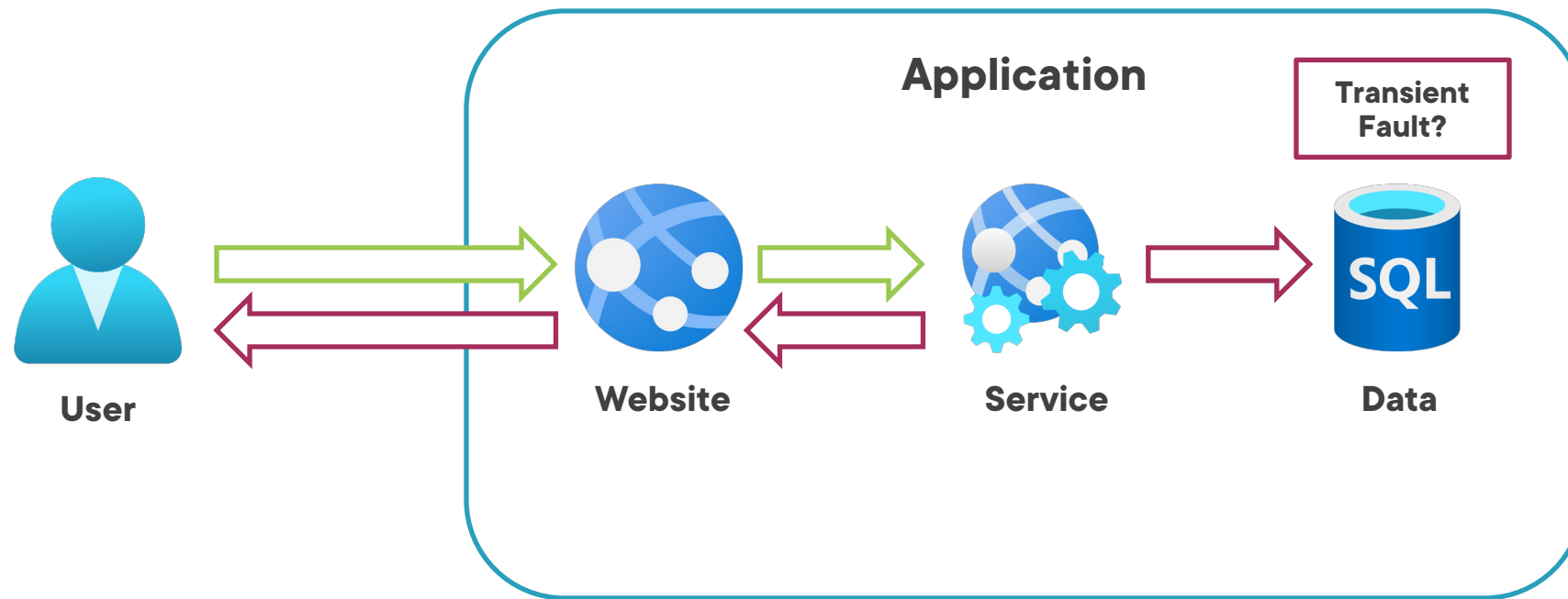
Demo: Handling Poison Messages

Communication Problems

Distributed Application

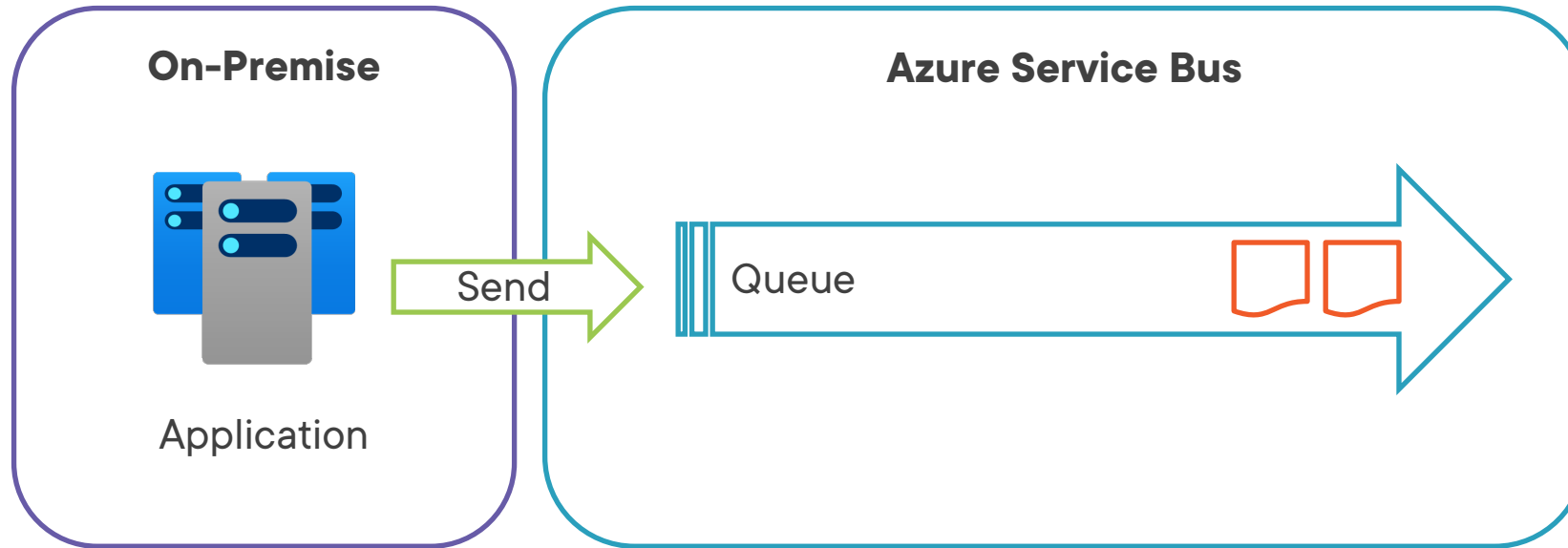


Handling Faults



Transient fault – Error that occurs due to a temporary condition

Handling Faults – Sending Messages



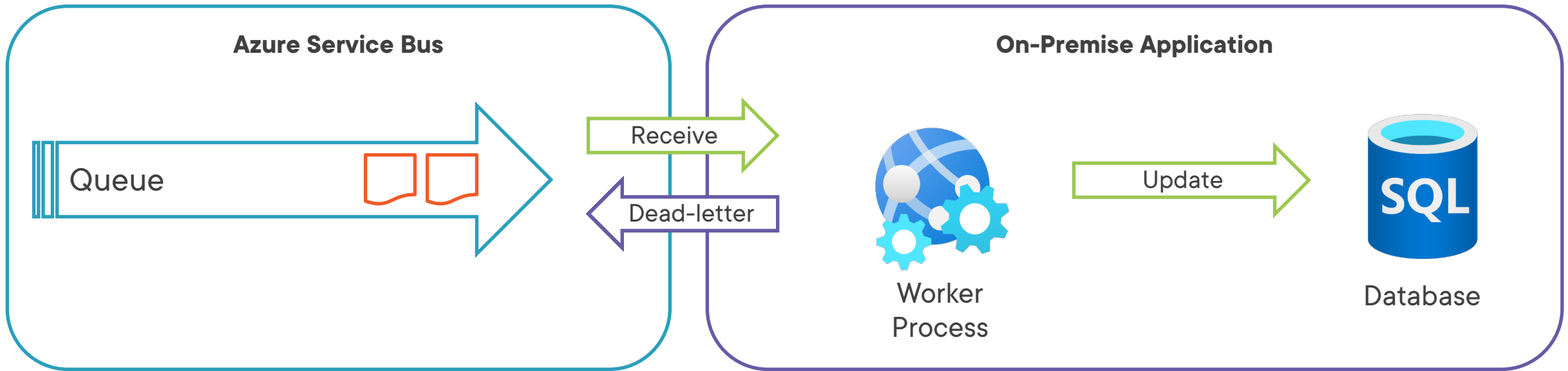
Possible Errors

- Network error
- Throttling exception
- Internal service error in service bus
- Security exception
- Endpoint not found exception

Options

- Try to resend message
- Report an error

Handling Faults – Processing Messages



Possible Errors

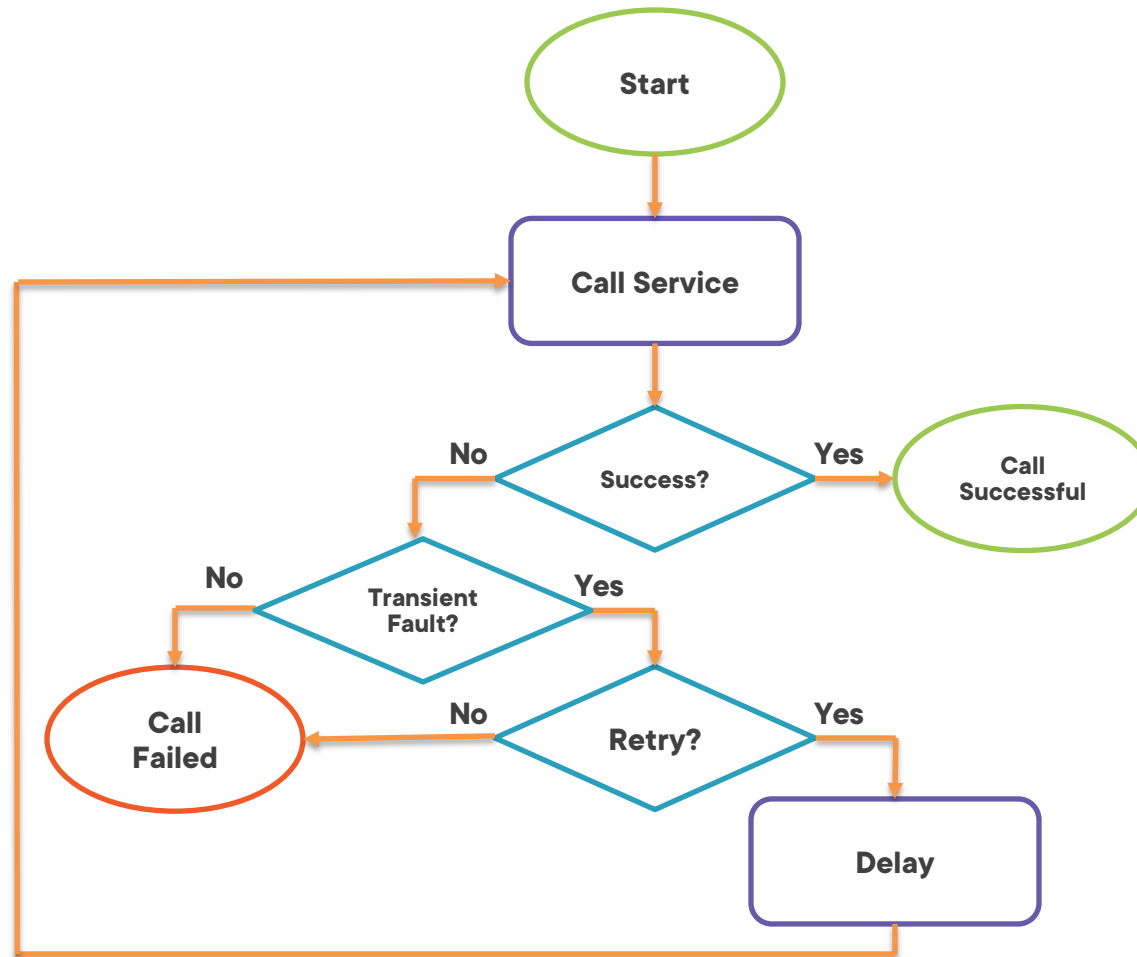
- Data integrity violation
- Security exception
- Throttling exception

Options

- Abandon message
- Dead-letter message
- Do nothing

Transient Fault Handling

Transient Fault Handling Flowchart

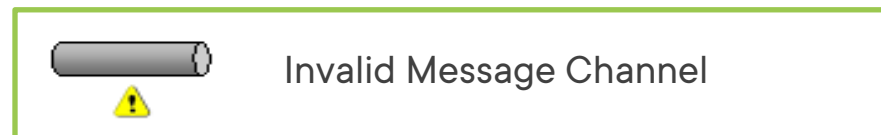


Dead-Lettering

Enterprise Integration Patterns



The messaging system moves messages to a dead-letter channel.



The receiving application moves messages to an invalid message channel.

Dead-lettering Scenarios

Processing Failure

- A receiving application fails repeatedly to process a message

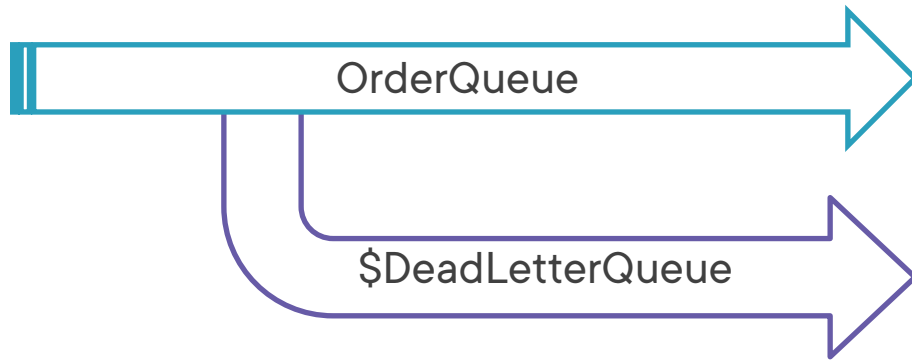
Poison Messages

- The content of a message makes it not possible to process

Expired Messages

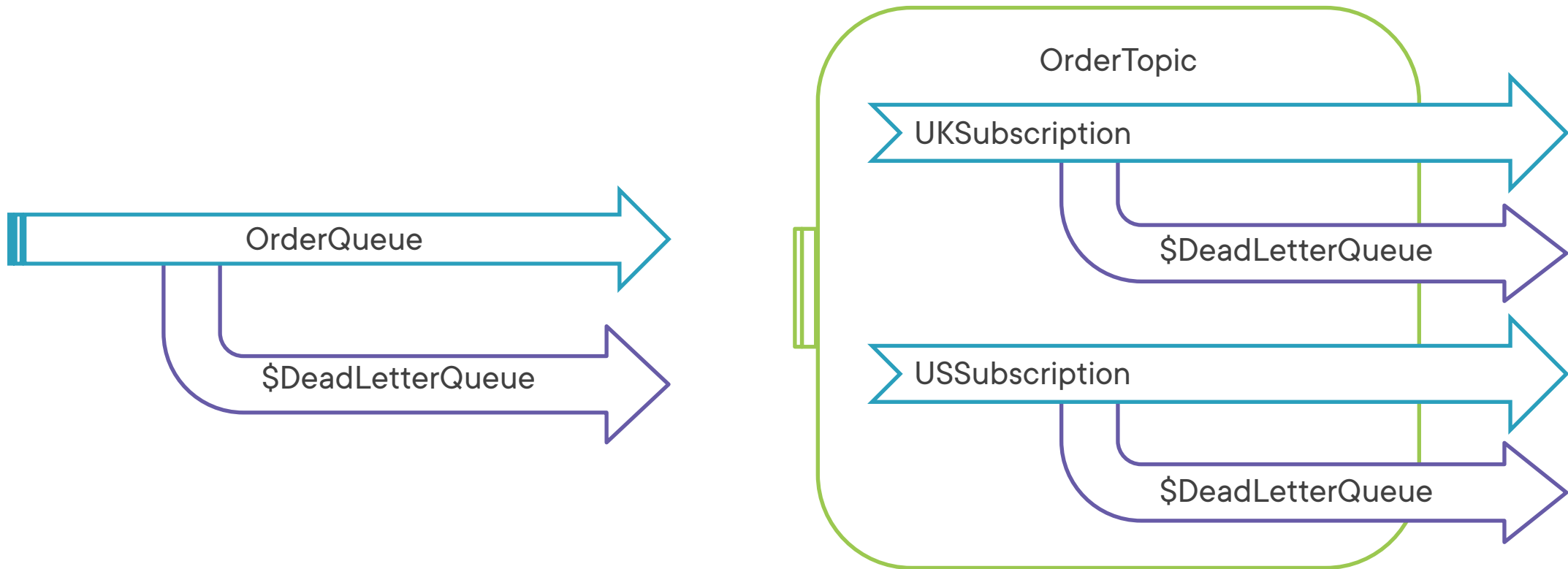
- A message has been present in the messaging system past its expiration time

Dead-letter Sub-Queues



- Queues
 - Each queue has a dead-letter sub-queue

Dead-letter Sub-Queues



- Queues
 - Each queue has a dead-letter sub-queue
- Topics and Subscriptions
 - Each subscription has a dead-letter sub-queue

Dead-letering in the Azure Service Bus

Implicit Dead-Lettering

Explicit Dead-Lettering

Dead-letering in the Azure Service Bus

Implicit Dead-Lettering

Max delivery count is exceeded

Explicit Dead-Lettering

Dead-letering in the Azure Service Bus

Implicit Dead-Lettering

Max delivery count is exceeded

Message expiration

- When enabled on queue or subscription

Explicit Dead-Lettering

Dead-lettering in the Azure Service Bus

Implicit Dead-Lettering

Max delivery count is exceeded

Message expiration

- When enabled on queue or subscription

Routing failure exception

- When enabled on subscription

Explicit Dead-Lettering

Dead-lettering in the Azure Service Bus

Implicit Dead-Lettering

Max delivery count is exceeded

Message expiration

- When enabled on queue or subscription

Routing failure exception

- When enabled on subscription

Explicit Dead-Lettering

Dead-lettered by receiving application

- Reason and description can be specified

Maximum Delivery Count

MaxDeliveryCount Property

Configurable on Queues and Subscriptions

Messages that repeatedly fail processing can be dead-lettered automatically

Default value is 10

However...

These dead-lettered messages contain no meaningful information as to the cause of processing failure

Maybe not the best way to handle processing failures

Consider this as a safety net

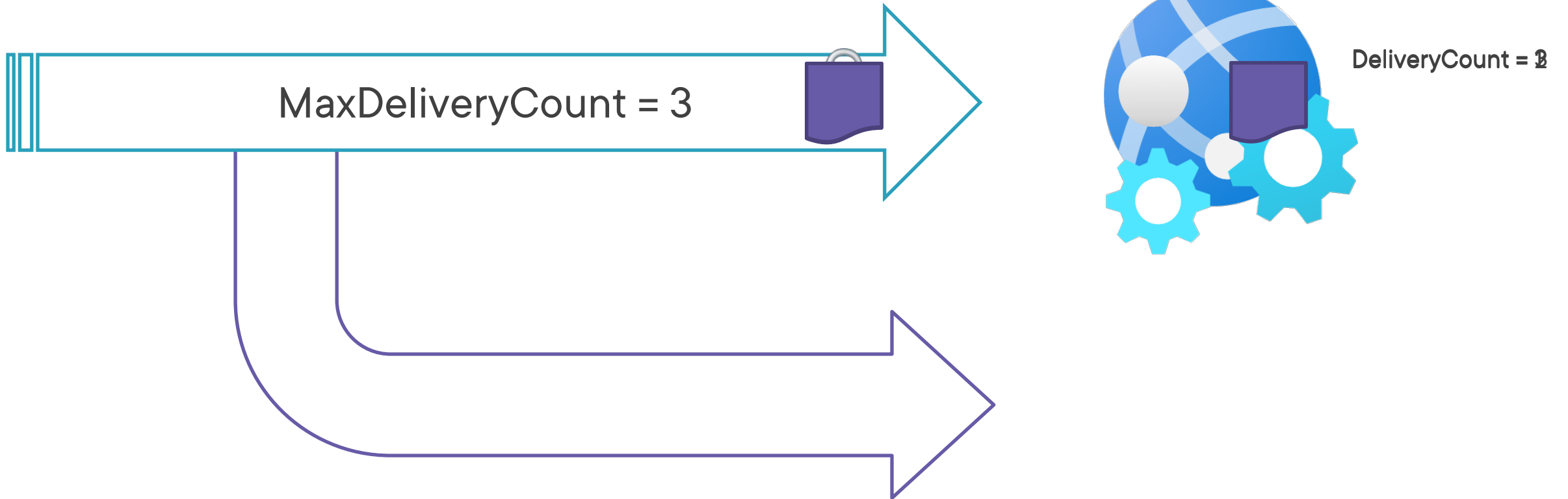
- Strive not to need it

Maximum Delivery Count

```
var subscriptionOptions = new CreateSubscriptionOptions
    ("topic", "subscription")
{
    MaxDeliveryCount = 5
};

await administrationClient.CreateSubscriptionAsync(subscriptionOptions);
```

MaxDeliveryCount Property



Explicitly Dead-Lettering Messages

- Call `DeadLetterMessageAsync`
 - Specify dead-letter reason and error description description

```
// Explicitly dead-letter a message.  
await receiver.DeadLetterMessageAsync(message, "Invalid order", "Error in billing address");
```

Dead-Lettered Message Properties



- Properties are added to the message properties collection
 - DeadLetterReason
 - DeadLetterErrorDescription

Demo

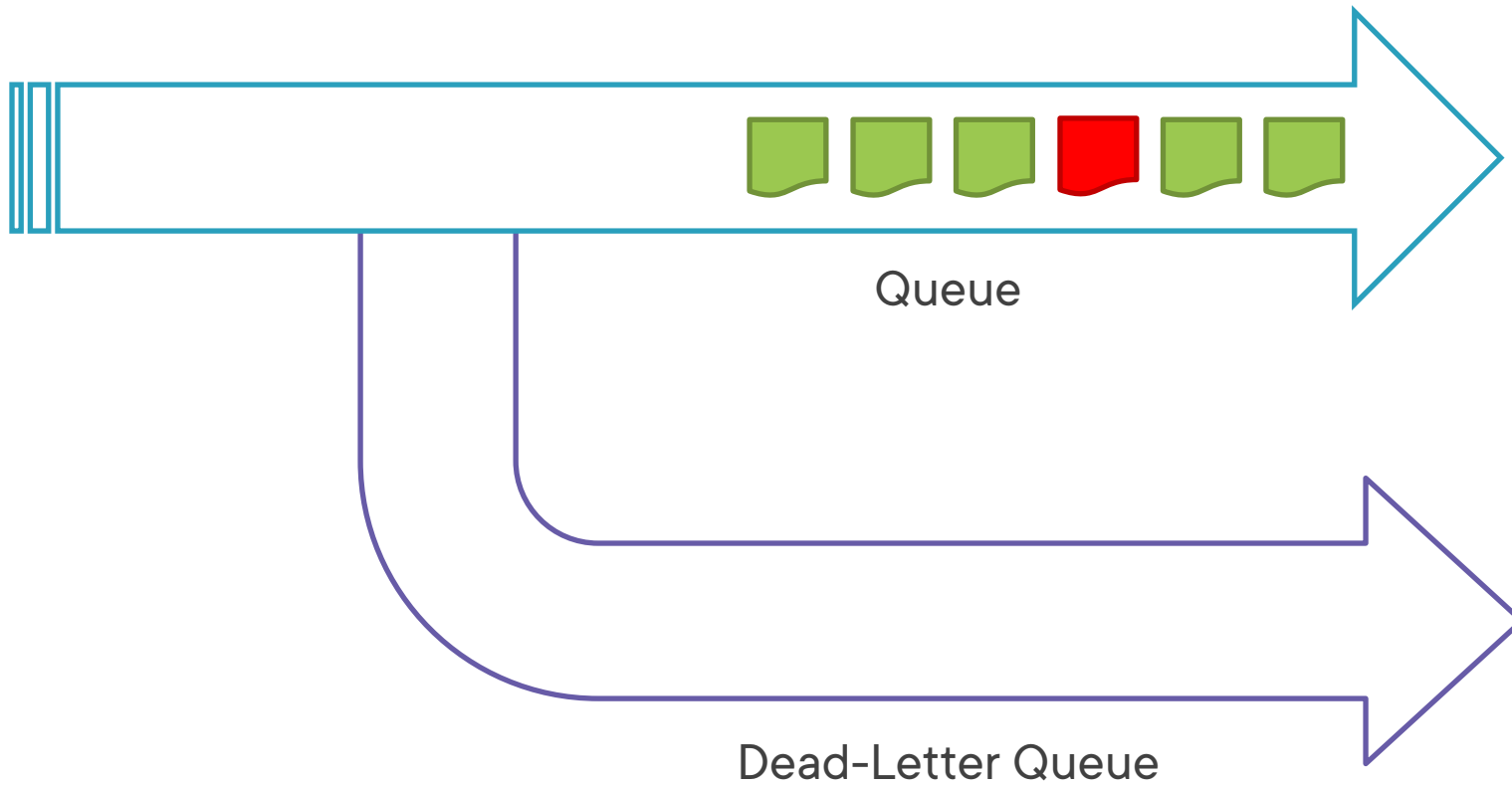


Demo: Dead-Lettering Messages

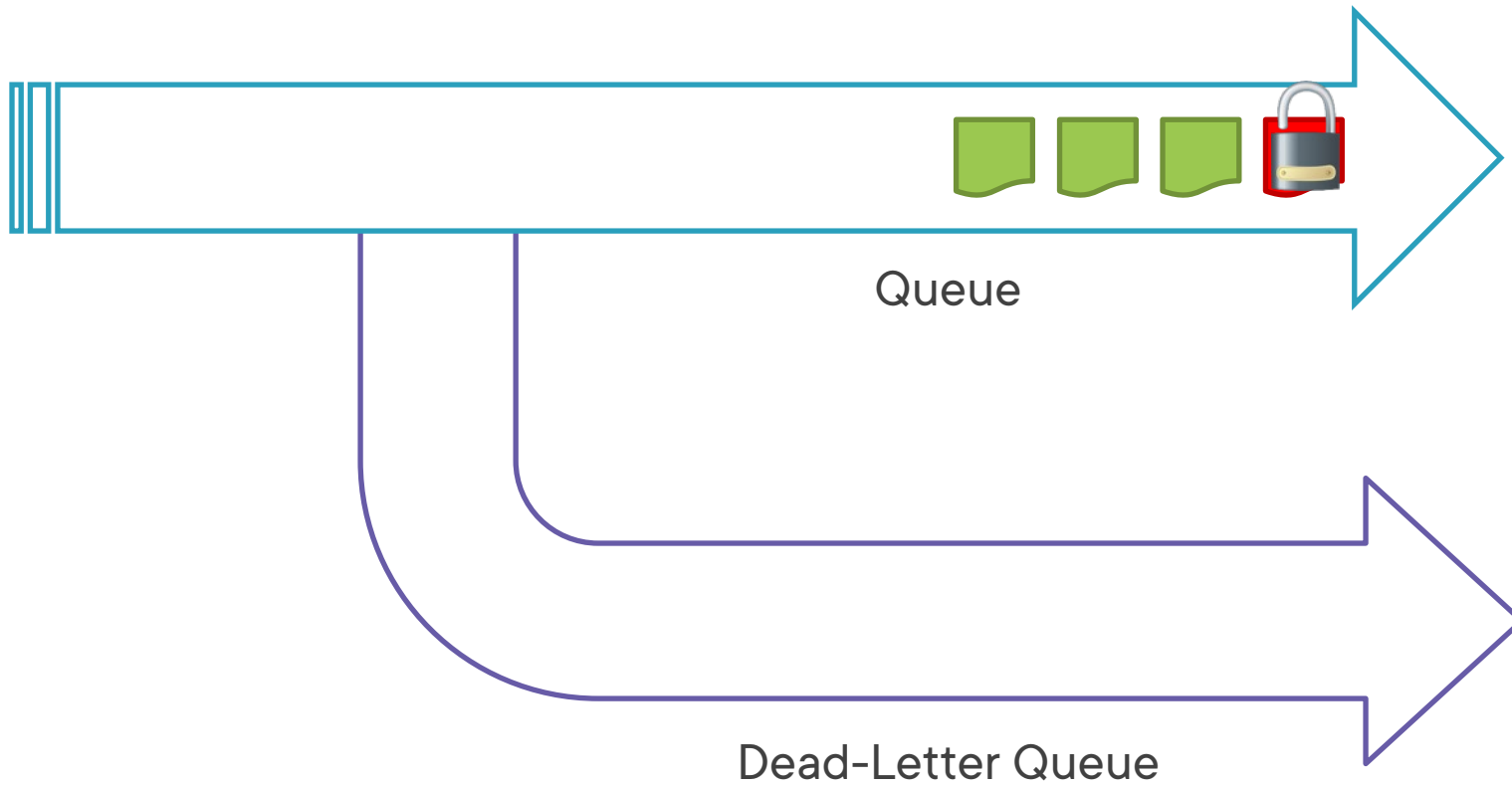
- Automatic Dead-Lettering of Messages
- Message Delivery Count Property
- Explicit Dead-Lettering of Messages

Handling Poison Messages

Poison Messages

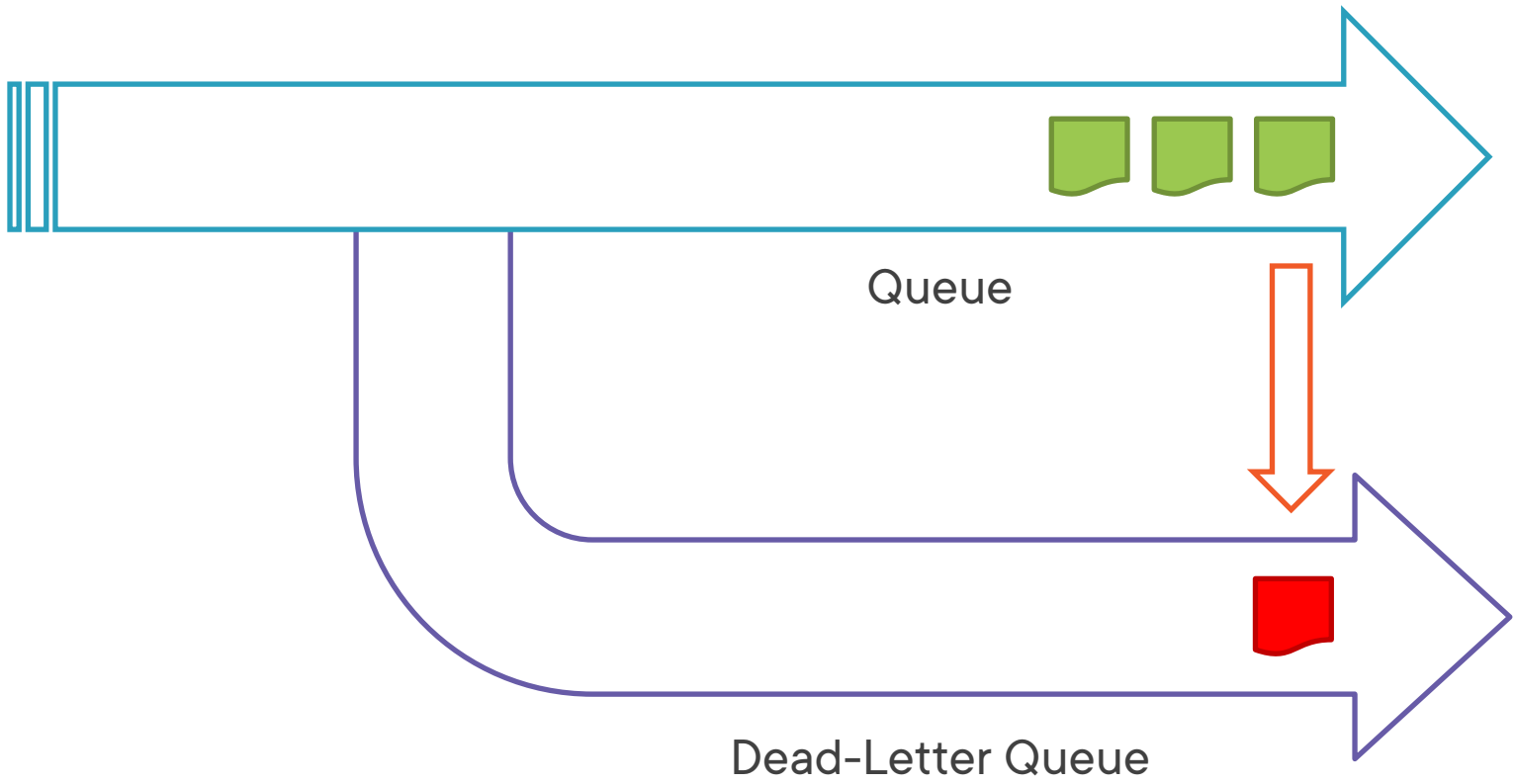


Poison Messages



Process
Message

Poison Messages



Dead-Letter
Message

Demo



Handling Poison Messages

- Poison Messages
- Handling Poison Messages

Summary



Dead-Lettering Messages

- Don't rely on max delivery count
- Take control of dead-lettering explicitly
- Providing reason and description of error
- Consider leveraging lock duration to handle transient faults

Handling Poison Messages

- Dead-letter explicitly providing reason and description of error