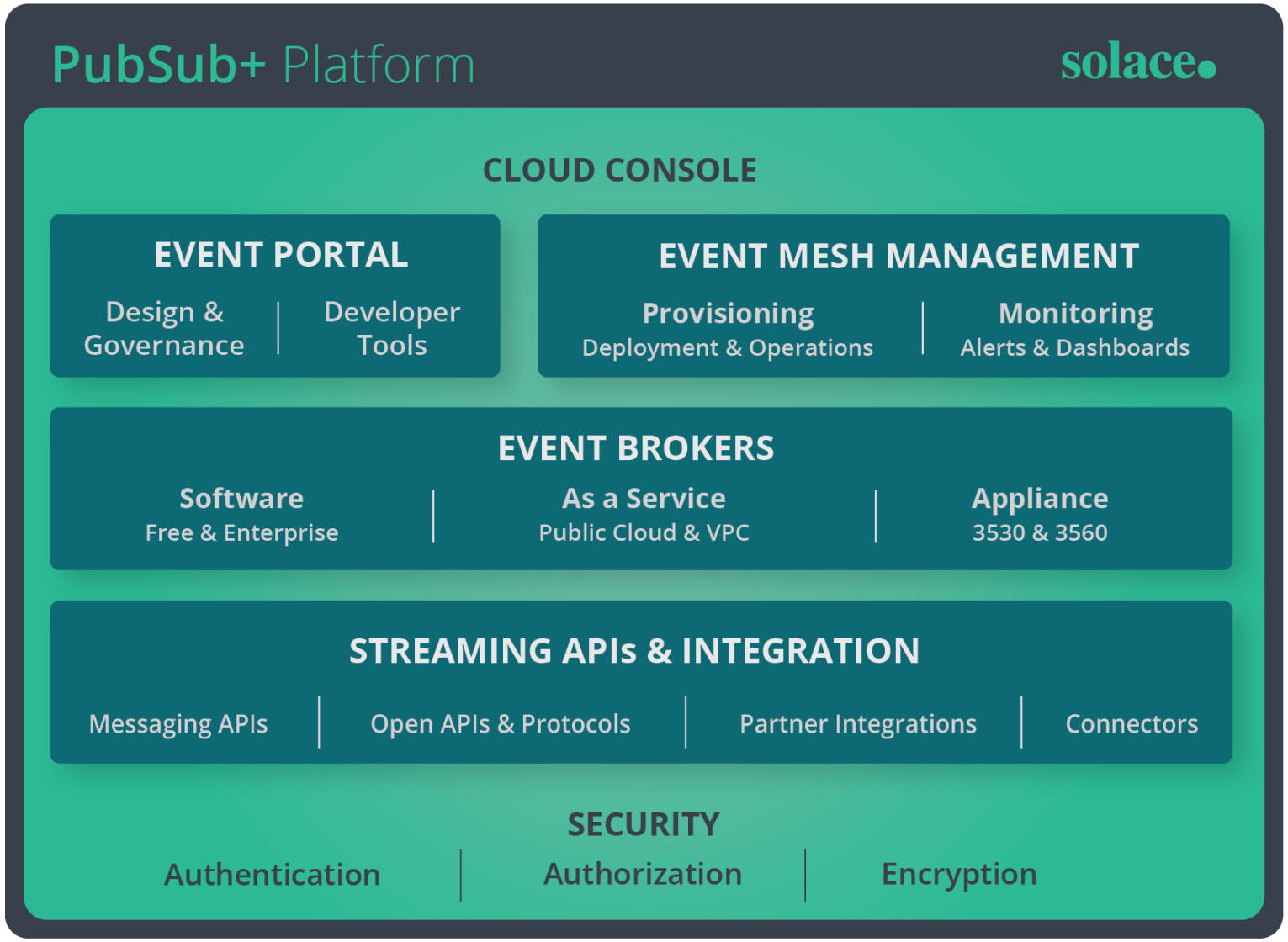
PubSub+ Platform Overview

solace.

PubSub+ Platform Overview



Event-Driven Basics

solace.

Terminology & Concepts

Click on the button below to learn more.

- 01 – Events
- 02 – Event Streams
- 03 – Commands
- 04 – Events & Messages
- 05 – Loose Coupling
- 06 – Event Channels
- 07 – Clients
- 08 – Quality of Service

solace.

Events

Events are notifications of change of state.

- Typically, events represent the change of state of something of interest to the business.
- Events are records of something that has happened.
- Events can't be changed, that is, they are immutable. (We can't change something that has happened in the past).

Examples of Events

- **The thermostat in a home notifying the owner temperature is high**
- **A change in stock price in a trading system**
- **A rideshare app like Uber and Lyft notifying that the driver has arrived to pick up the passenger**

Event Streams

An event stream is a continuous unbounded series of events.

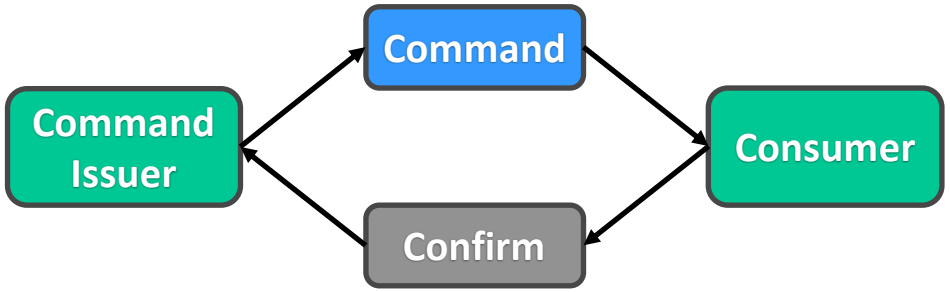
- The start of the stream may have occurred before we started to process the stream.
- The end of the stream is at some unknown point in the future.
- Events are ordered by the point in time at which each event occurred.

solace

Examples of Event Streams

- **An order processing application receiving a stream of orders being placed by different customers on an ecommerce website**
- **A rideshare app sending the driver location information to the passenger**
- **A video stream from a security camera**

Command



A command, is an instruction to do something.

- Typically, commands are directed to a particular consumer.
- The consumer runs the required command or process, and passes back a confirmation to the issuer stating that the command has been processed.

solace

Examples of Commands

- **A user unlocking their connected car through a mobile app**
- **A train control system applying brakes on the train automatically because the driver missed a signal**
- **A connected home turning on the air conditioning because the thermostat notified a high temperature in the home**

Event vs Messages

A Message consists of two basic parts

- Header – describes the data being transmitted, its origin, its destination, etc.
- Body – the data being transmitted.

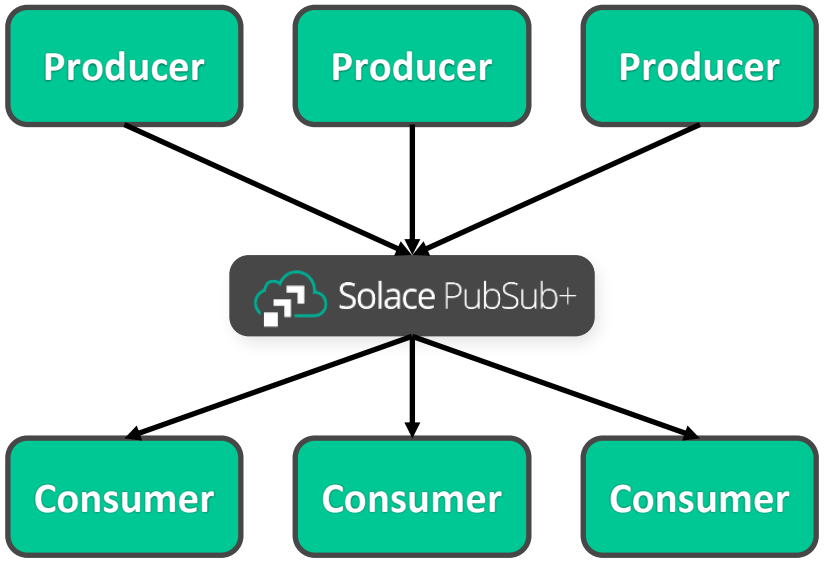
Events can be visible to a person or business whereas Messages are visible at the transport level between applications and messaging systems.

solace.

More about Messages…

- **Message can transmit events, commands, or any piece of raw data**
- **In an event-driven solution an event becomes part of the payload of the message, a.k.a. Event Message**
- **Event Messages are routed from applications creating them to consumer applications by an Event Broker, such as the PubSub+ Event Broker**

Loose Coupling



Event producers can publish event without any knowledge of who is consuming the events

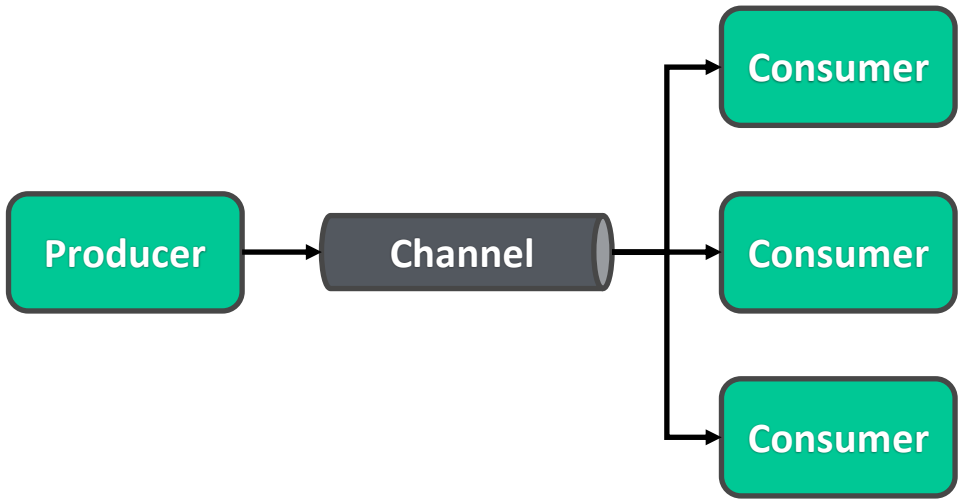
- Event consumers don't need to be aware of the event producers
- Because of this, producers and consumers can be implemented in different languages or be on different platforms

solace.

Loose coupling does not mean “no coupling”

- **Consumers establishes what data it needs and the type and format of the data**
- **Producers establishes an implicit contract with potential consumers**
- **For example:**
  - an event notification in JSON format must conform to a certain schema that must be known by both the consumer and the producer

Event Channels



Channels are logical addresses in an Event Broker

- An application producing an event, publishes it to an Event Channel
- An application interested in receiving the event, subscribes to that Event Channel to retrieve the event
- A broker can have multiple channels that have different purposes to communicate different events

solace.

Types of Event Channels

Topics

- Allow one-to-many communication (**Publish-Subscribe**)
- Typically used when you want to send a single event to multiple consumers

Queues

- Allow one-to-one communication (**Point-to-Point**)
- Typically used when you want to send a single event to exactly one consumer

More on Topics

- **Support a hierarchical naming scheme**

- **For example:**
  - myhome/livingroom/temperature
  - myhome/livingroom/humidity

- **Consumers can subscribe to multiple topics using Wildcard**
  - myhome/>
  - myhome/livingroom/*

Key things to remember about Topics

- **Support a hierarchical naming scheme**

- **For example:**
  - myhome/livingroom/temperature
  - myhome/livingroom/humidity

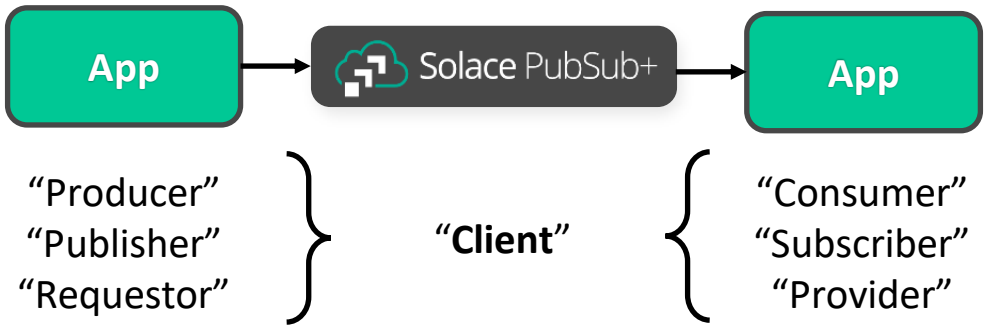
- **Consumers can subscribe to multiple topics using Wildcard**
  - myhome/>
  - myhome/livingroom/*

More on Queues

- **Ensure that only one receiver consumes any given event**
- **Can have multiple receivers but only one of them can successfully consume a particular event**
- **Provides persistence for events so that they are not lost even if the broker or consuming application crashes**
  - Producers receive a confirmation from the broker, a.k.a. acknowledgment, that the event has been persisted on the queue
  - Consumers send an acknowledgment to the broker that they have consumed and processed the event, and only then the event is removed from the queue

Clients

An application that sends or receives an event is called a *client* of the broker

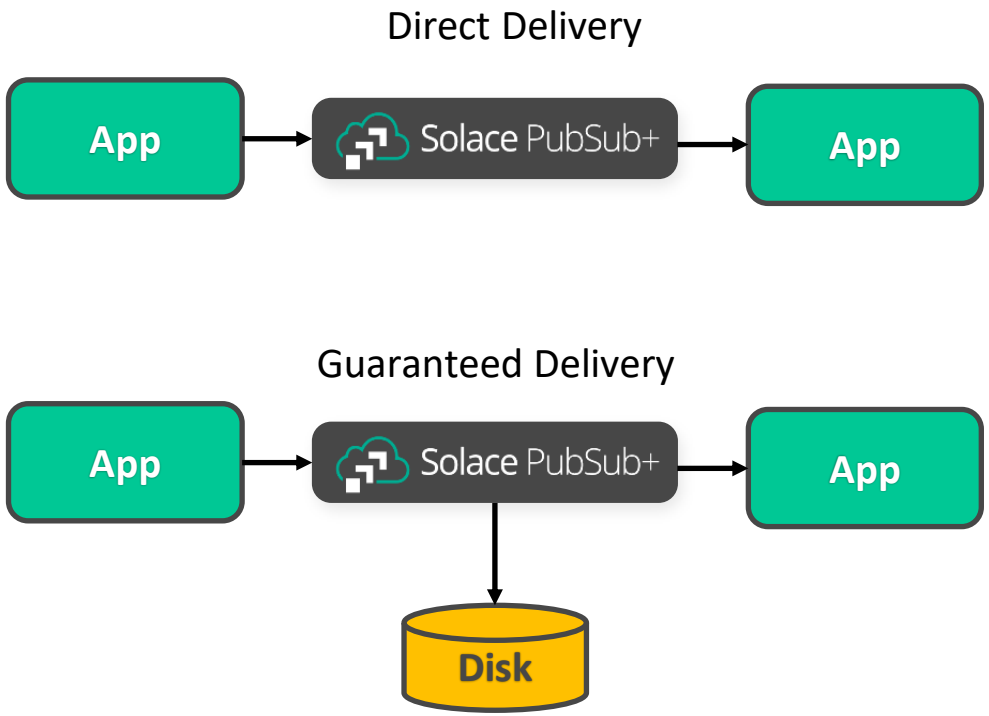


Terms that are often used to describe a client:

- Sender and receiver
- Producer and consumer
- Publisher and subscriber
- Requestor and provider

solace

Quality of Service



Quality of service when publishing determines if an event should be transmitted reliably or not.

There are two qualities of service in PubSub+ Event Brokers:

1. Direct Delivery
2. Guaranteed Delivery

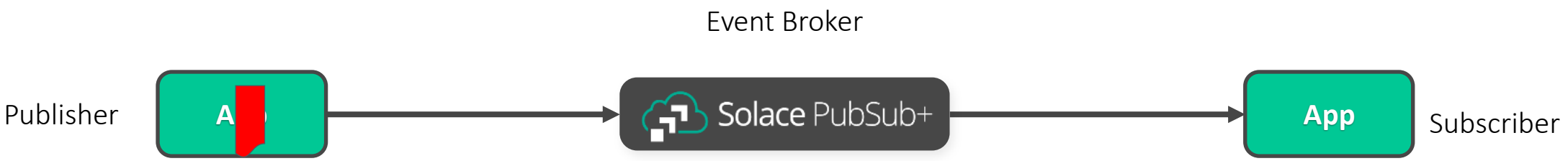
solace.

Direct Delivery

1 Send Message

2 Process Message

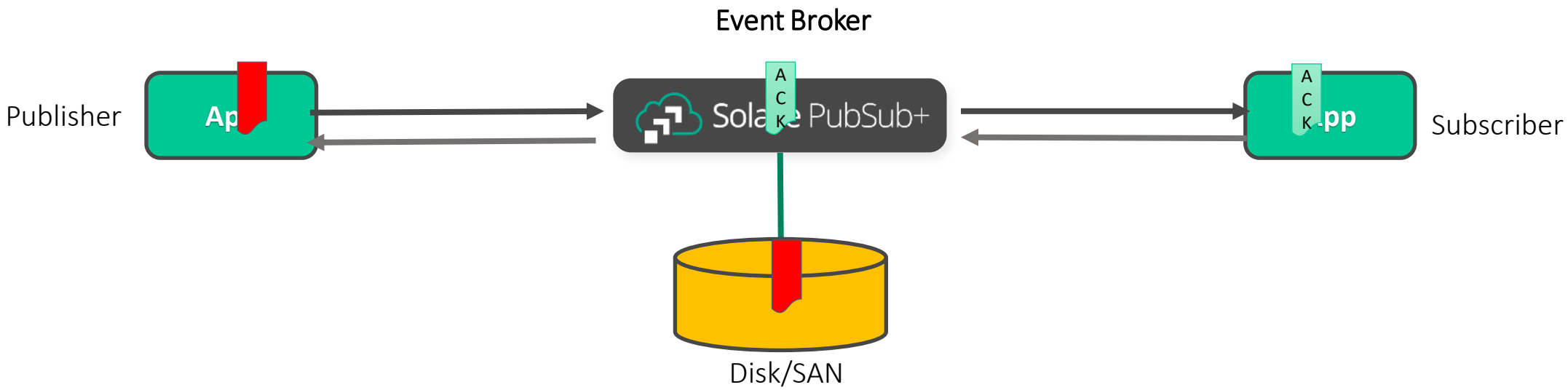
3 Receive Message



Fire and Forget

Guaranteed Delivery

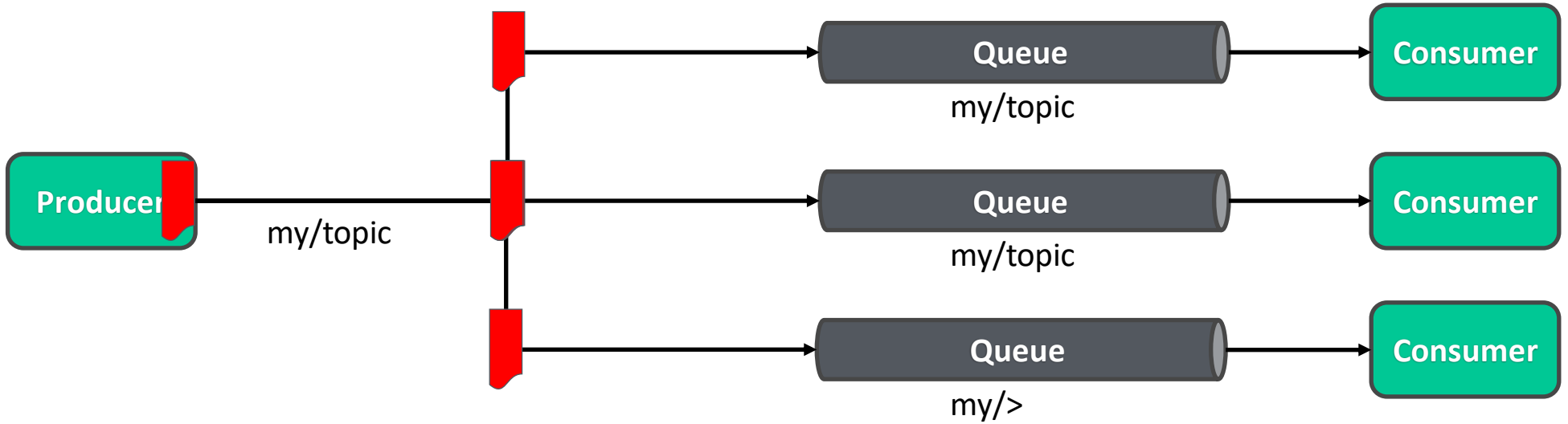
1. Send Message
2. Process and Acknowledge
3. Receive Message
4. Process and Acknowledge



Store and Forward

Queues and Guaranteed Delivery

- **To use Guaranteed Delivery you need:**
  - PubSub+ Event Broker configured with disk for storage
  - Queue that acts as a virtual endpoint to store the events
- **Recommended for apps to publish events to Topics**
- **Topics can be mapped to Queues to persist the events**



Hands-on Activity

- **Create PubSub+ Cloud Console account**
  - Skip if you already have tried PubSub+ Cloud before, or already part of an **Enterprise** PubSub+ Cloud org, and in this case confirm you still have access

solace.