Architecting Stream Processing Solutions Using Google Cloud Pub/Sub

GETTING STARTED WITH CLOUD PUB/SUB

Vitthal Srinivasan
CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Messaging middleware

Decouples senders and receivers

Global, replicated, autoscaling

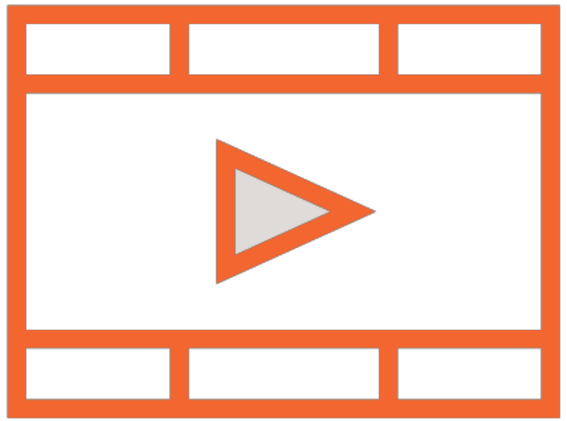
Publishers publish to topics

Subscribers listen on subscriptions

Complex pricing

Prerequisites and Course Outline

Prerequisites: Basic Cloud Computing



Choosing and Implementing Google Cloud Compute Solutions

Architecting Google Cloud Storage Configurations

Course Outline

Getting started

- Terminology: publishers, subscribers, messages
- Concepts and message flow
- Architectural overview

Working with Pub/Sub

- Publishers, topics, subscribers, subscriptions
- Web console and gcloud
- Syncing to timestamp and snapshots

Using client libraries

- Python client libraries for publishers and subscribers
- Async callbacks, batching and flow control
- Creating a Hangouts bot using Pub/Sub

Scenarios: SpikeySales.com

Hypothetical Online Retailer

- Flash sales of trending products
- Spikes in user traffic

SpikeySales on the GCP

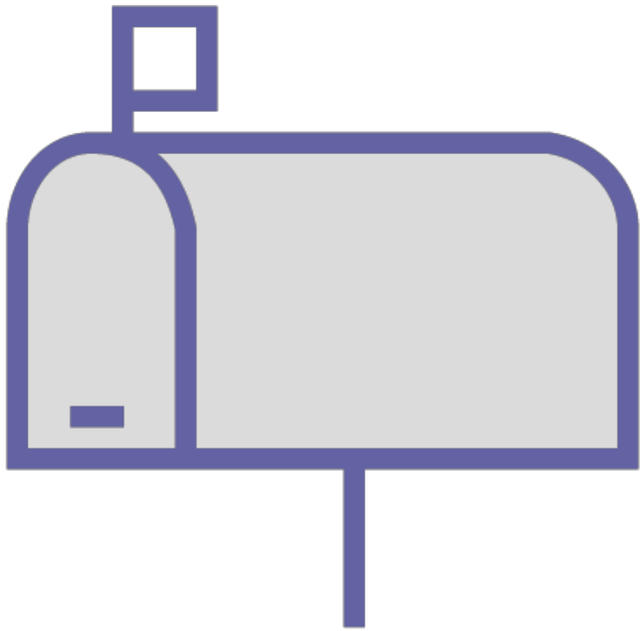
- Cloud computing fits perfectly
- Pay-as-you-go
- No idle capacity during off-sale periods

Pub/Sub

Cloud Pub/Sub

Simple, global, reliable real-time messaging and streaming ingestion service on the GCP.

Pub/Sub



Many-to-many asynchronous messaging

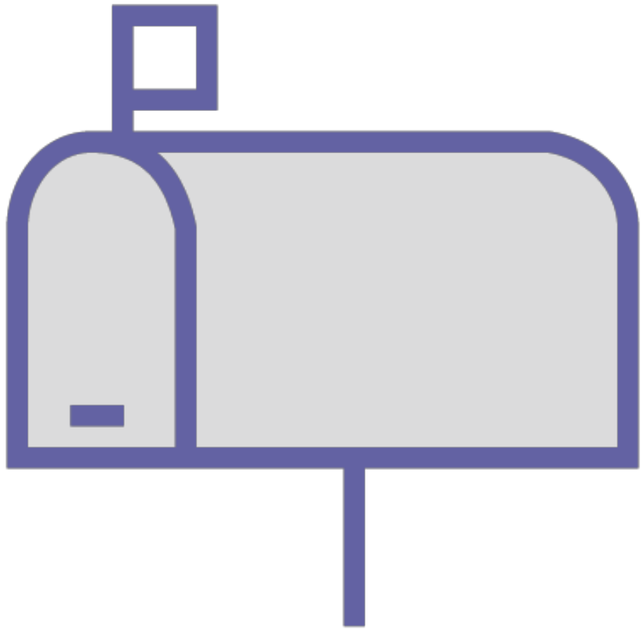
Decouples senders and receivers

Reliable, scalable, secure

At-least-once delivery

Exactly-once processing (in Dataflow)

Pub/Sub



Serverless

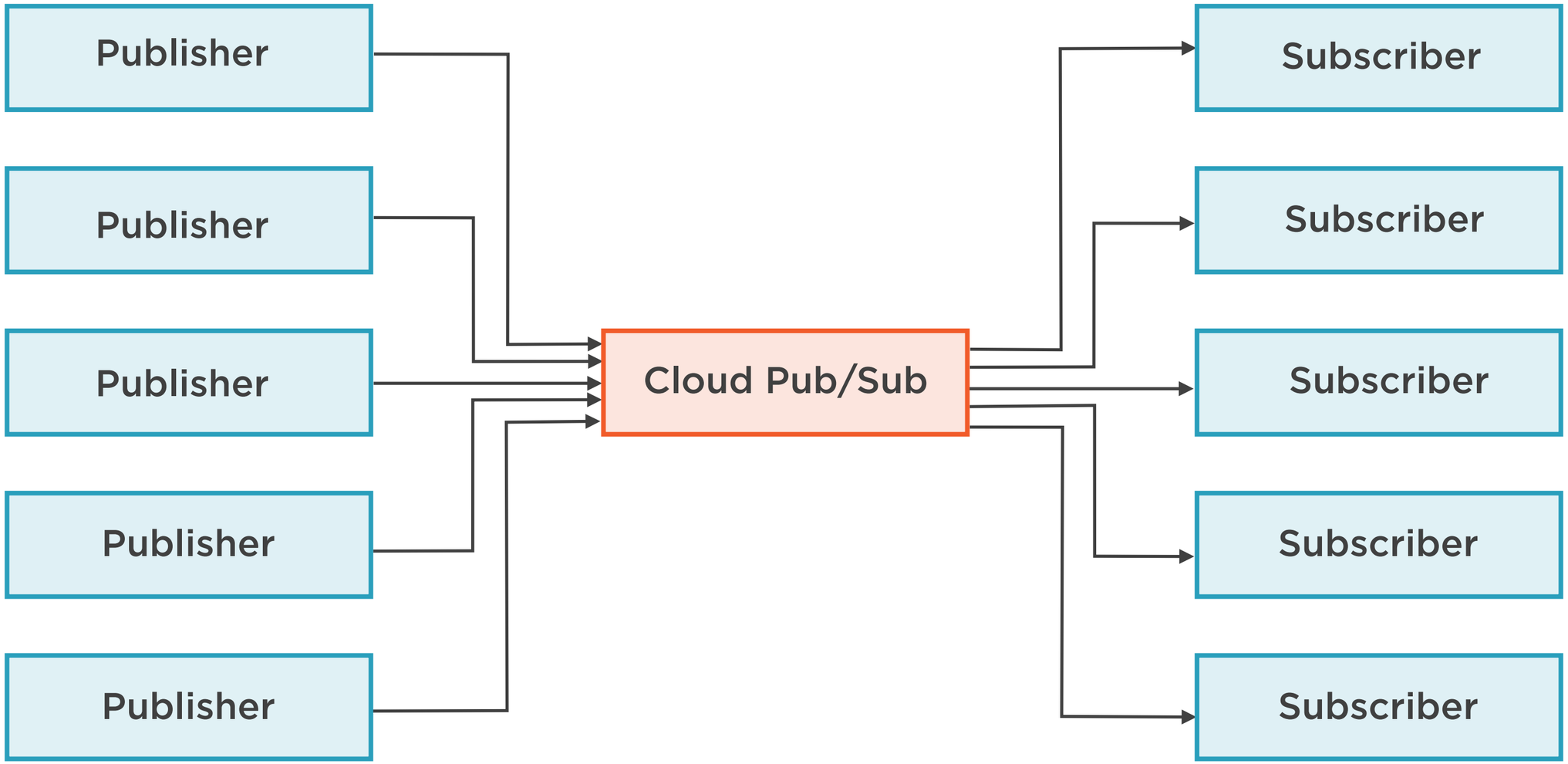
Autoscaling along all axes

- Senders
- Receivers
- Number and size of messages

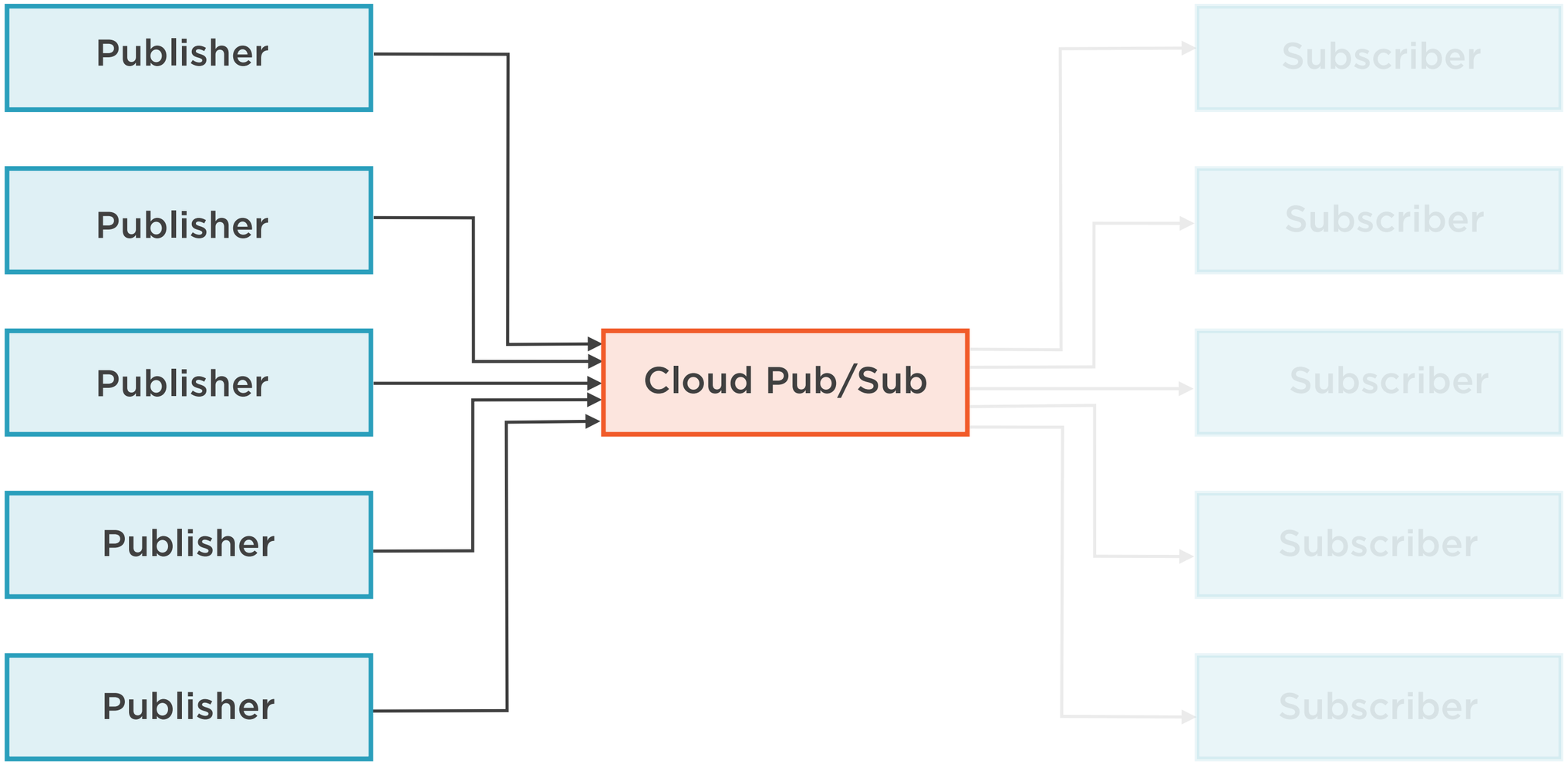
Replicated

Load-balanced internally

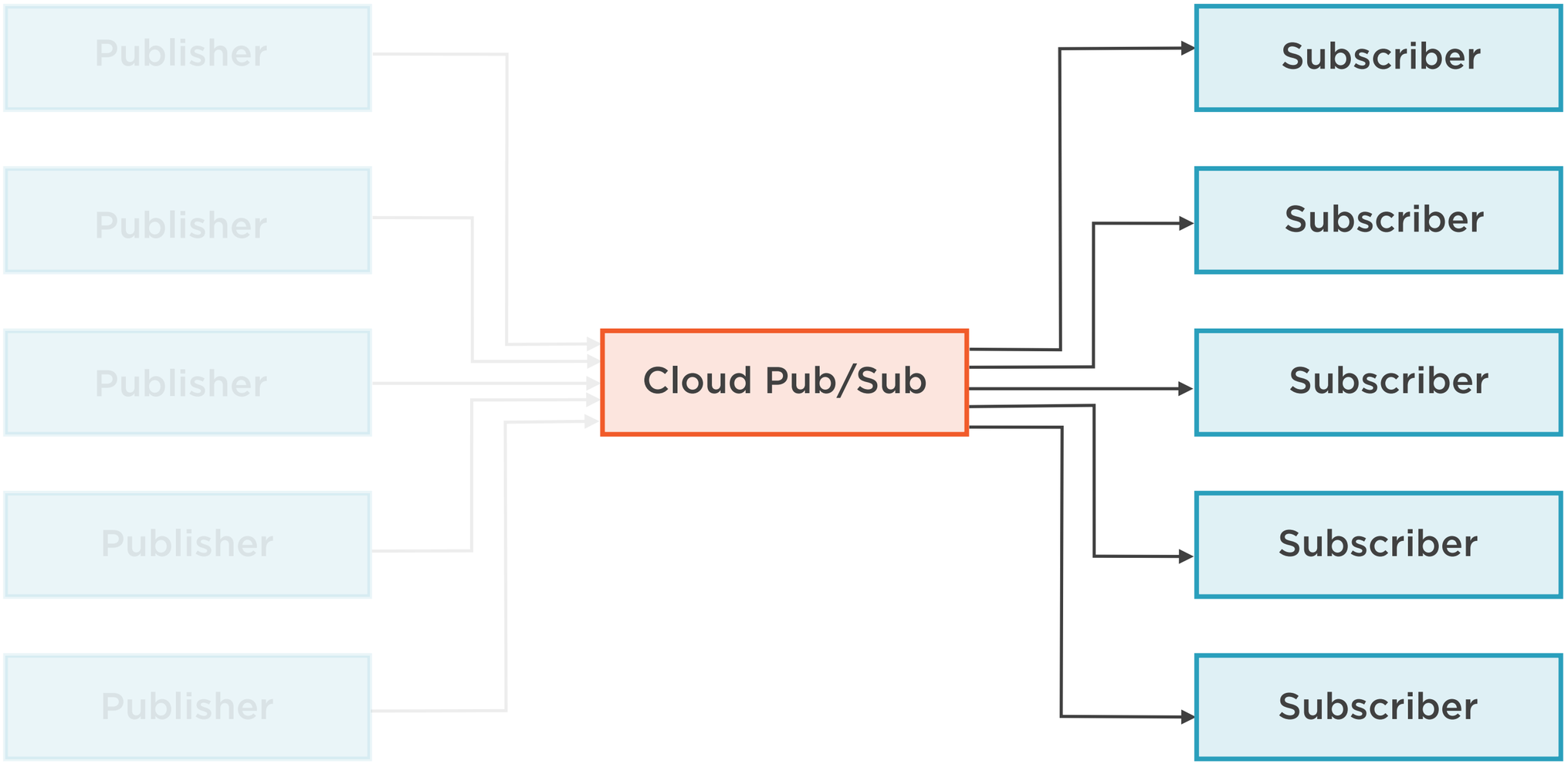
Messaging Middleware



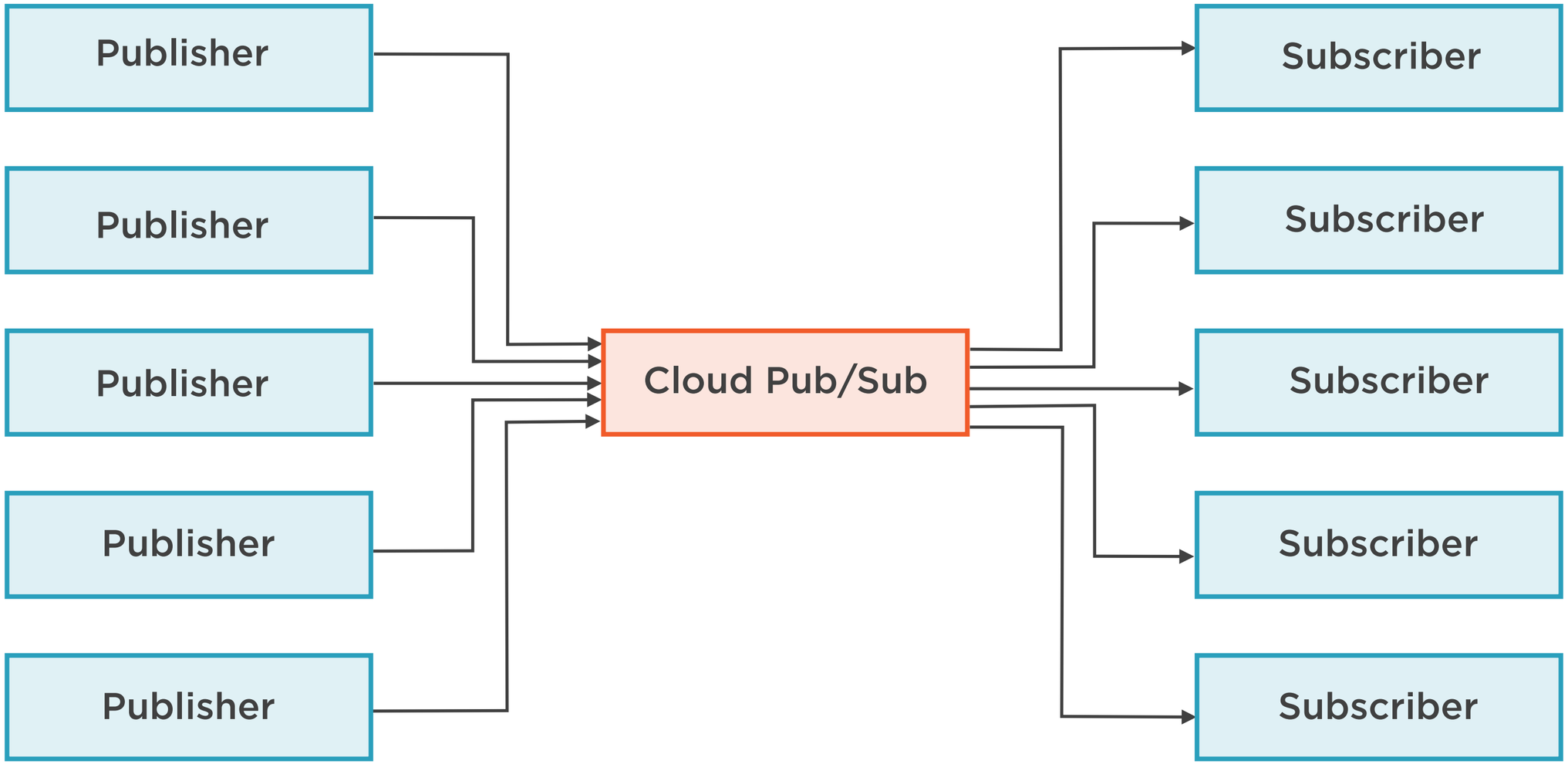
Messaging Middleware



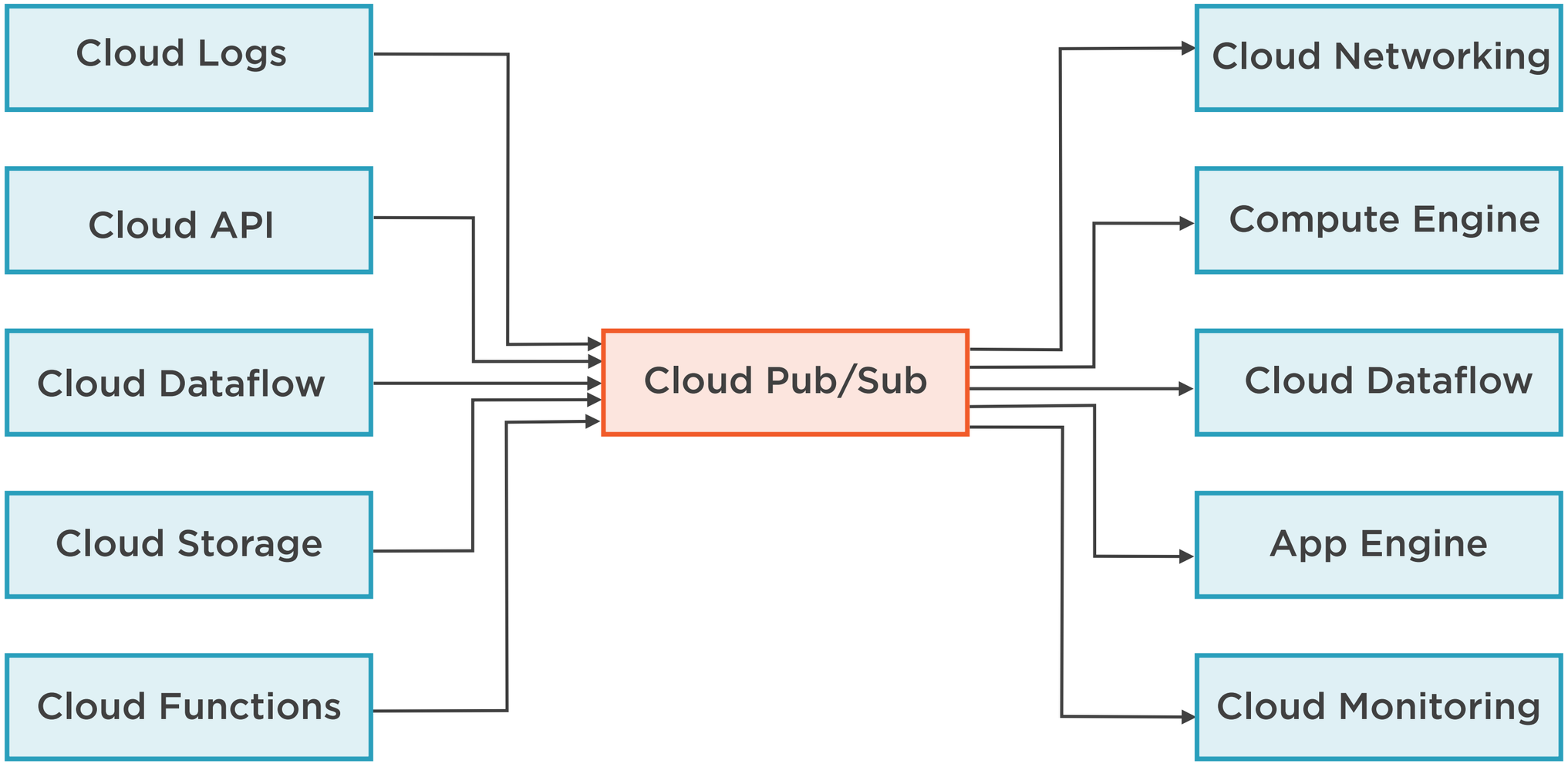
Messaging Middleware



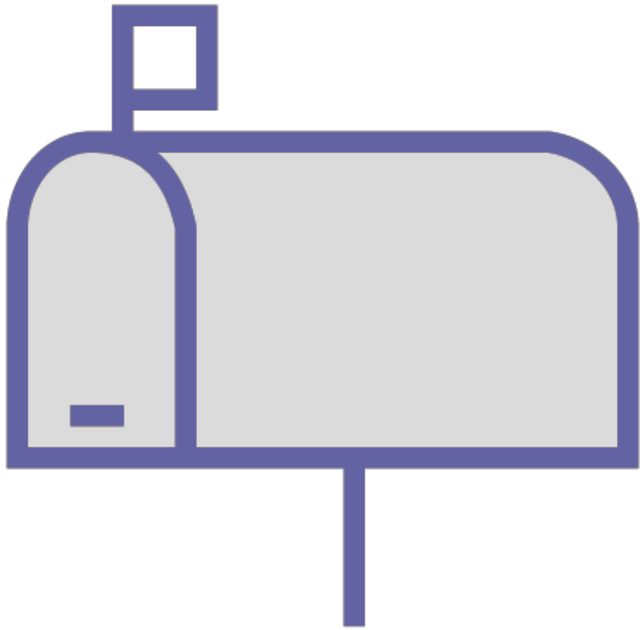
Messaging Middleware



Works with All GCP Services



Pub/Sub

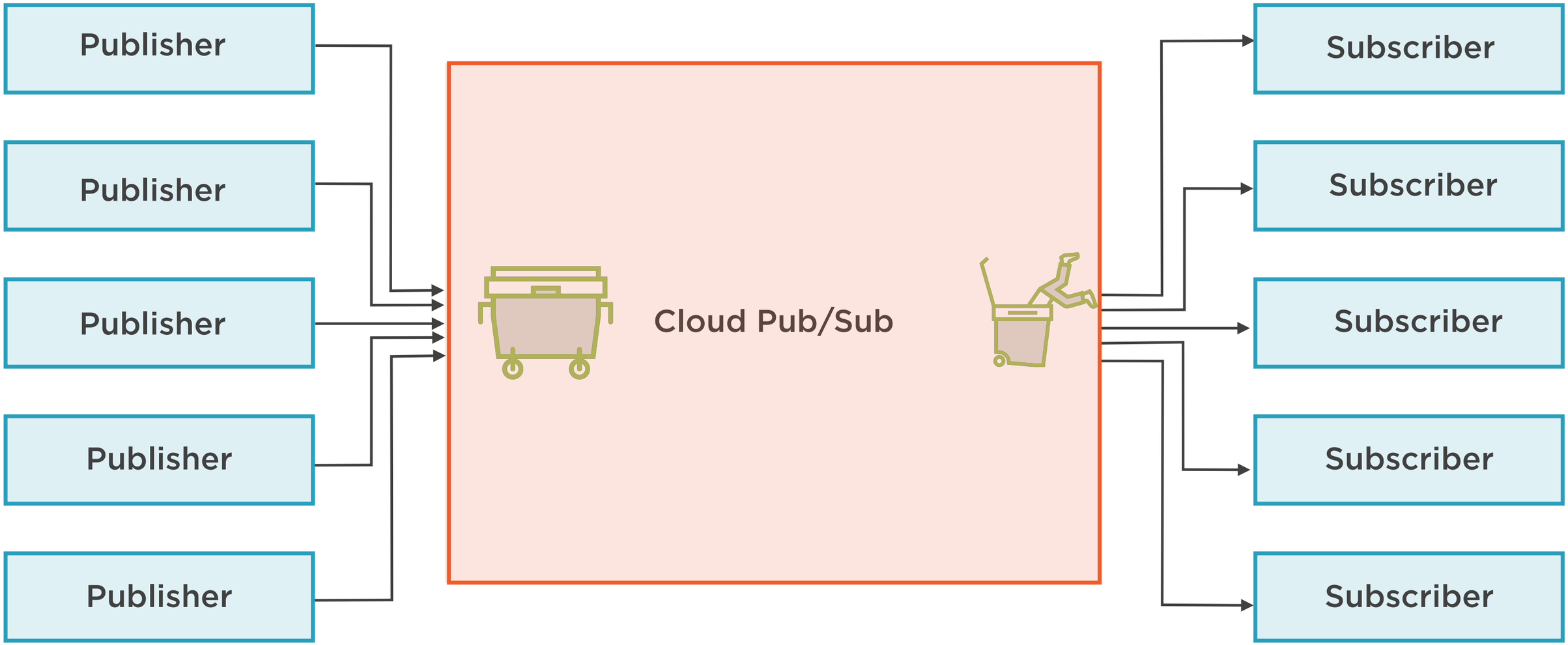


Publisher apps create and send messages on a **Topic**

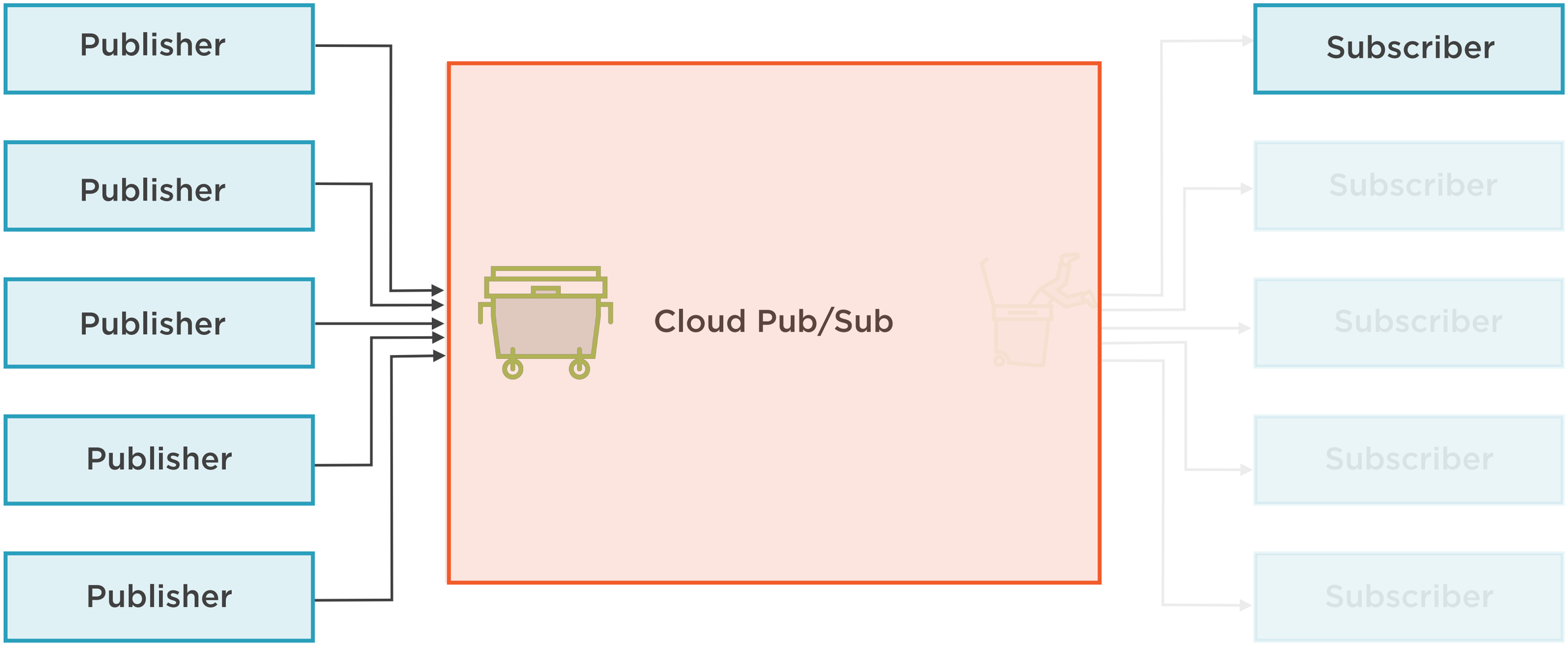
Subscriber apps subscribe to a topic to receive messages

Subscription is a queue (message stream) to a subscriber

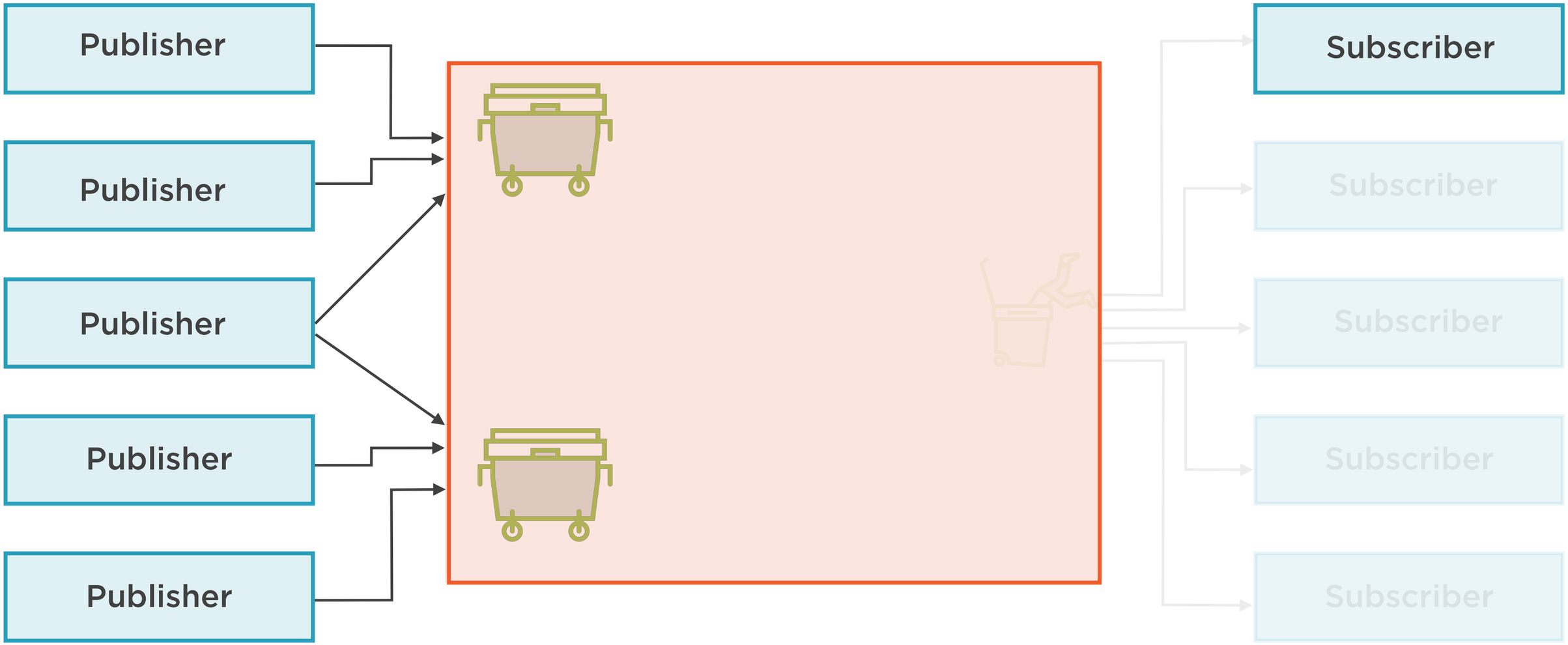
Publishers, Topics, Subscribers, Subscriptions



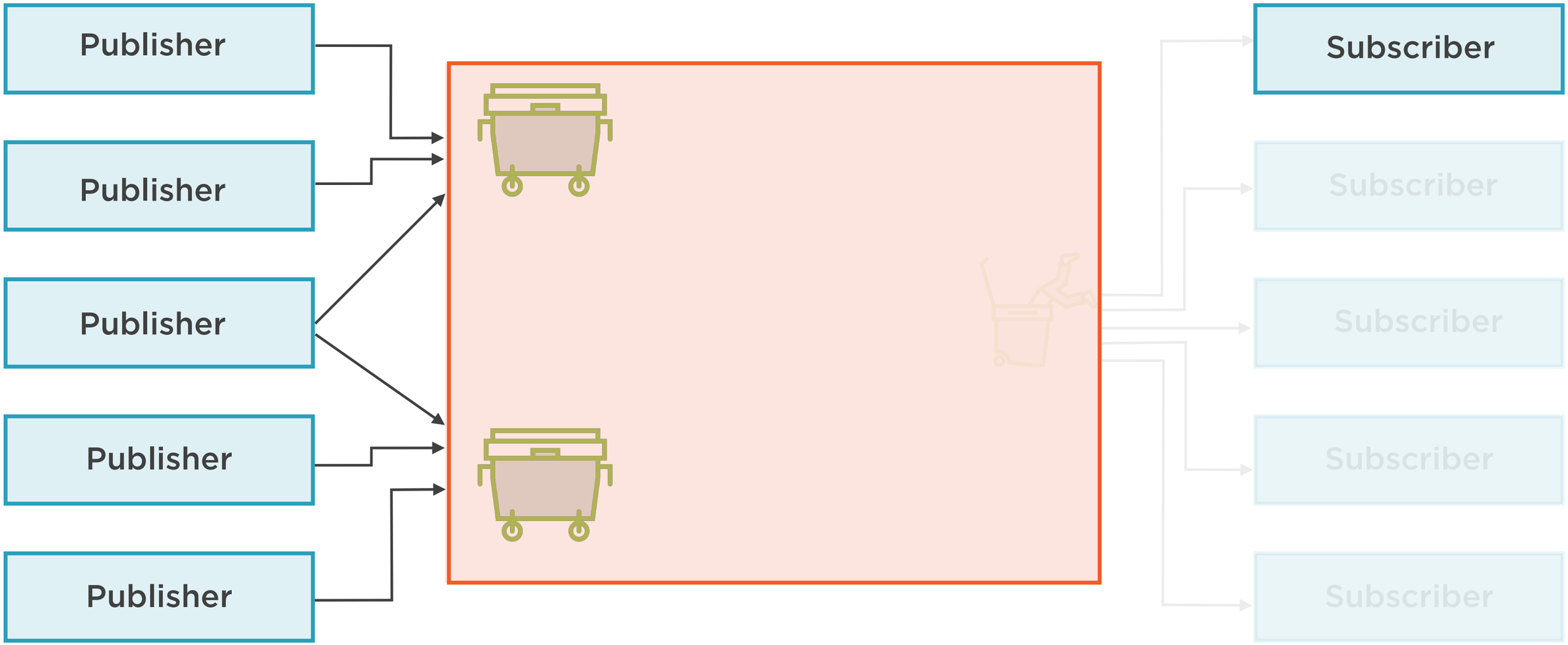
Publishers Publish to a Topic



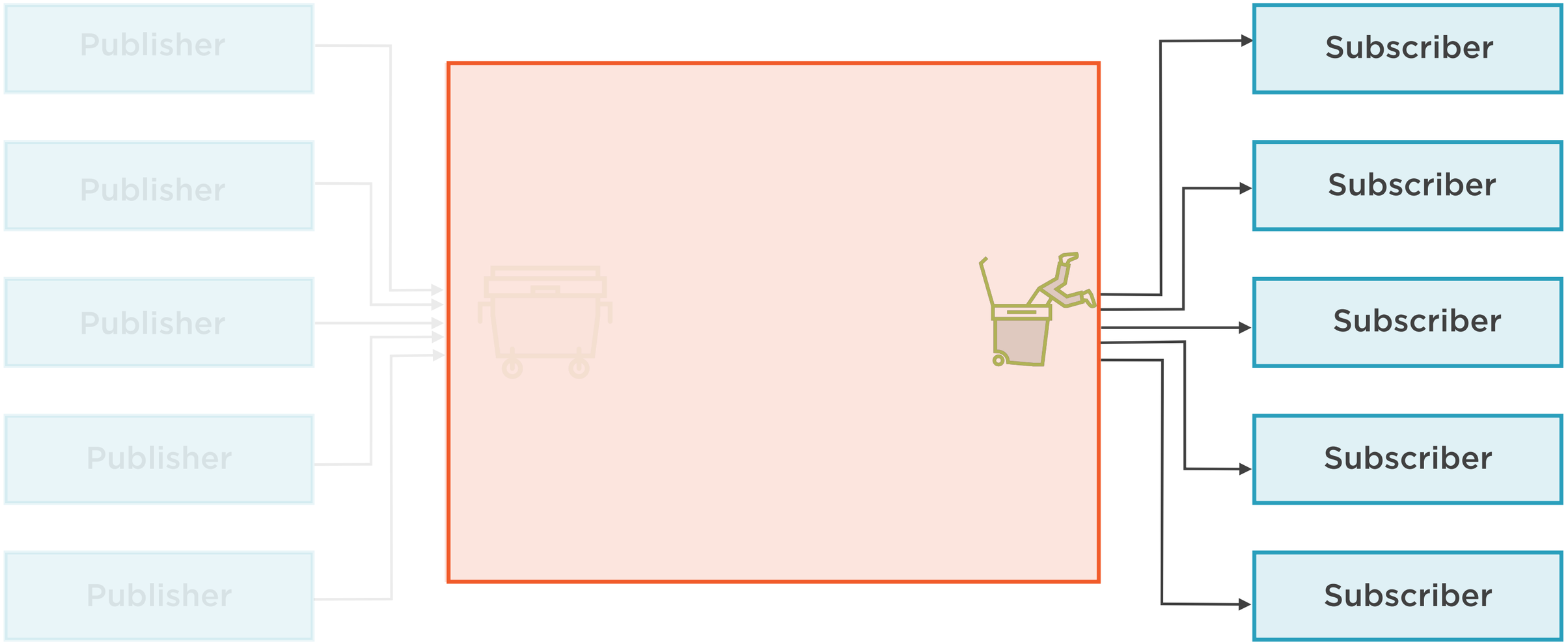
A Topic Can Have Multiple Publishers



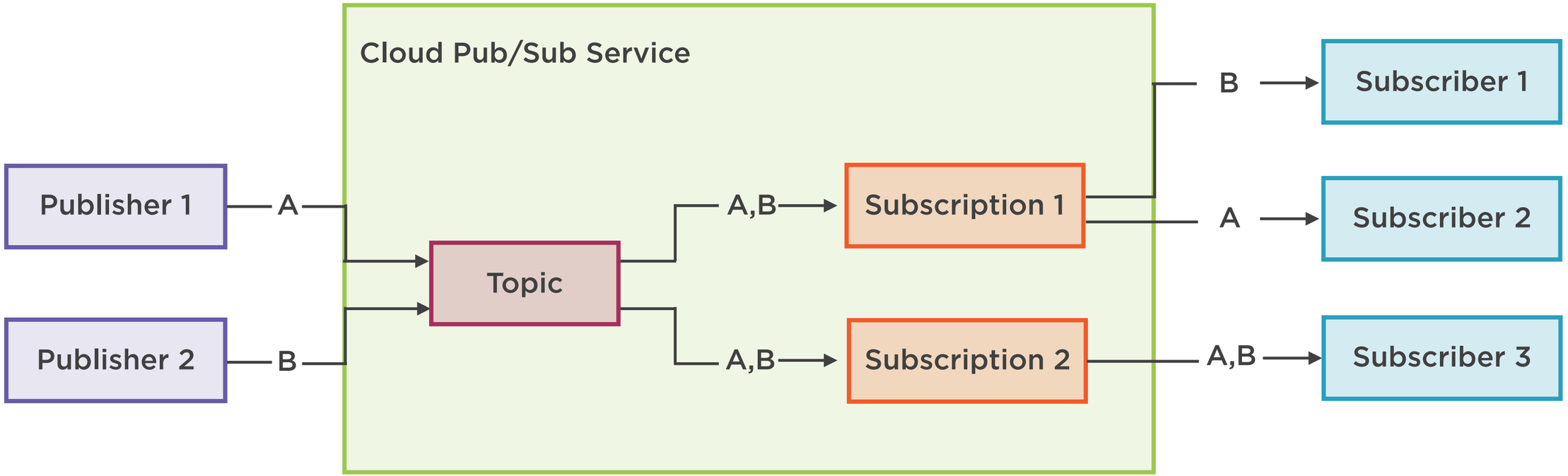
A Publisher Can Publish to Multiple Topics



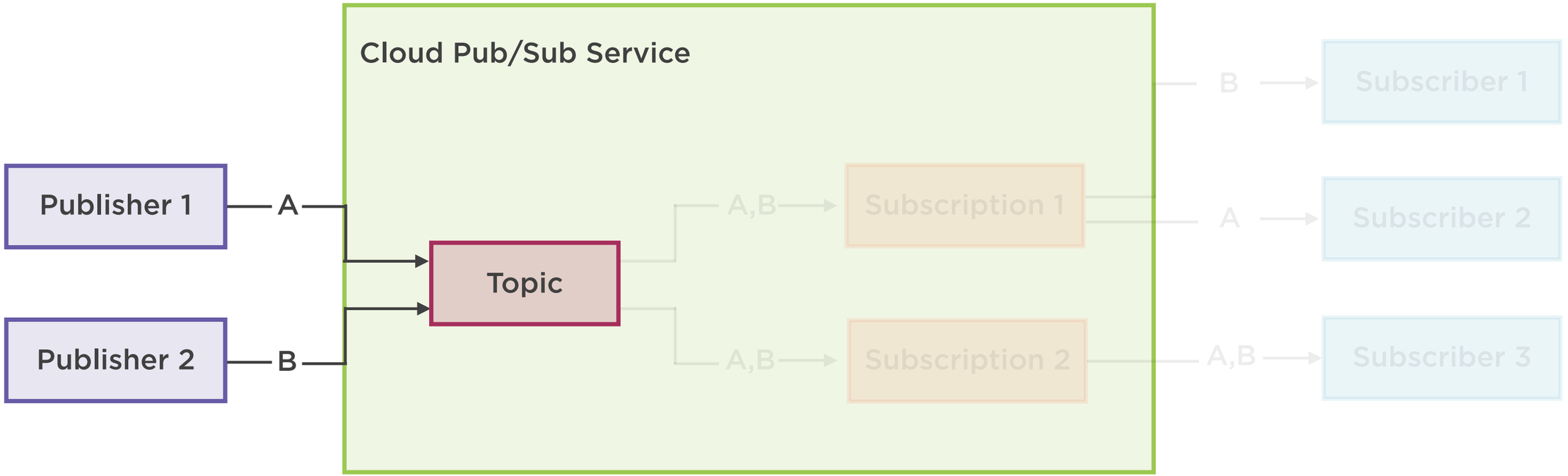
Subscribers Receive Messages from Subscriptions



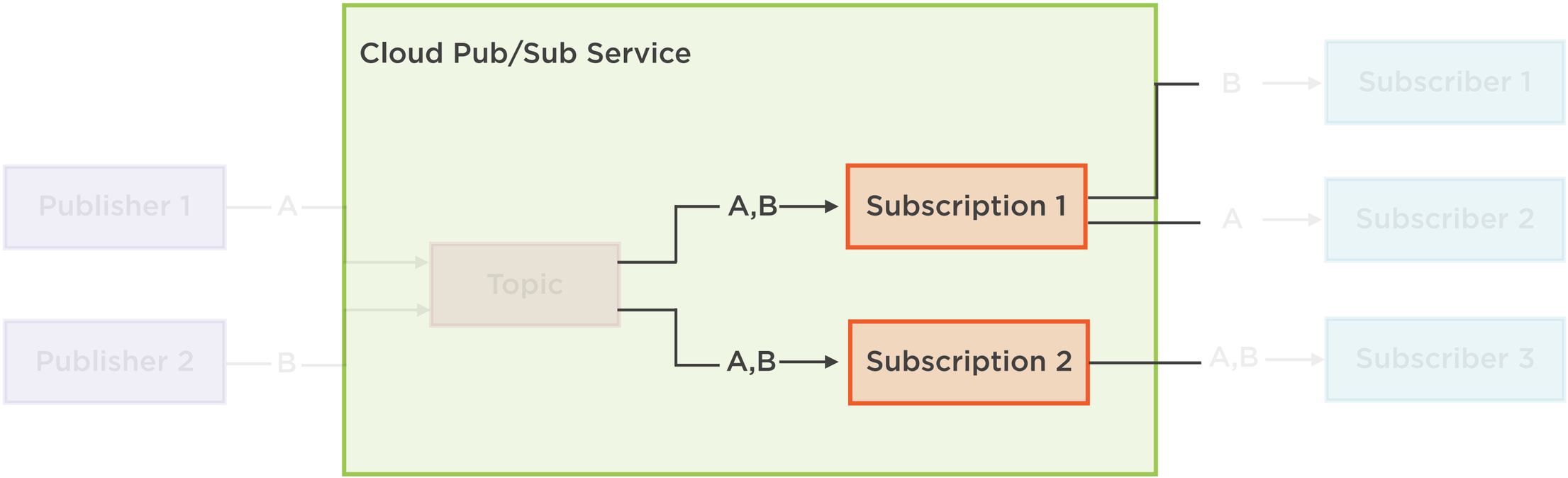
Topics and Subscriptions



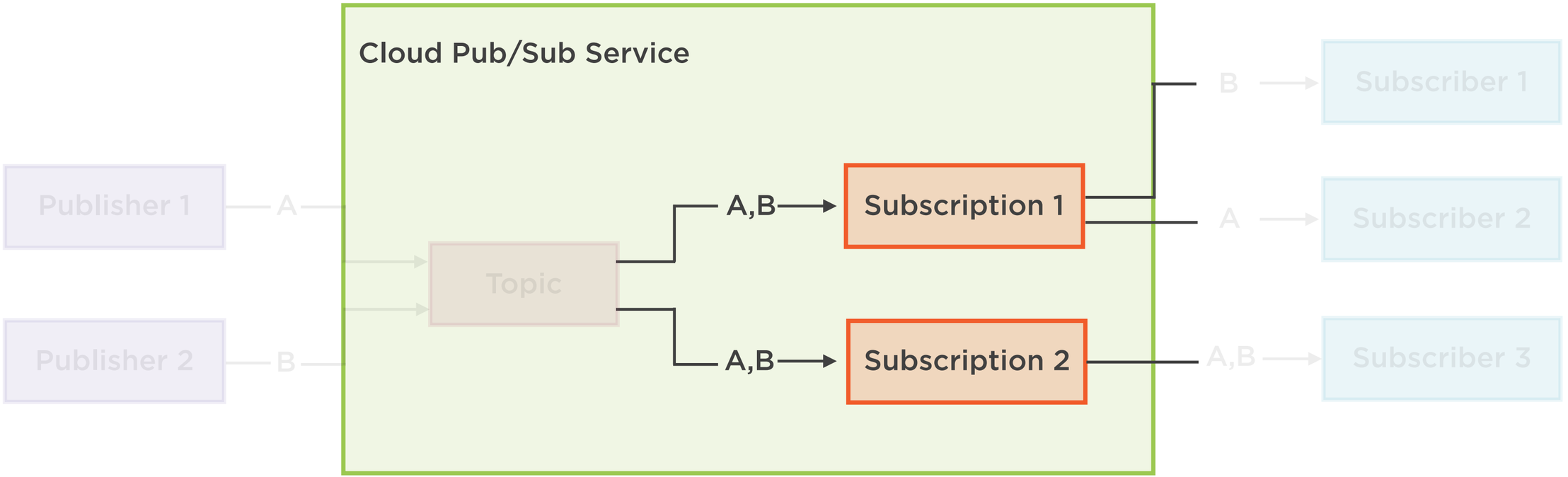
Two Messages on a Topic



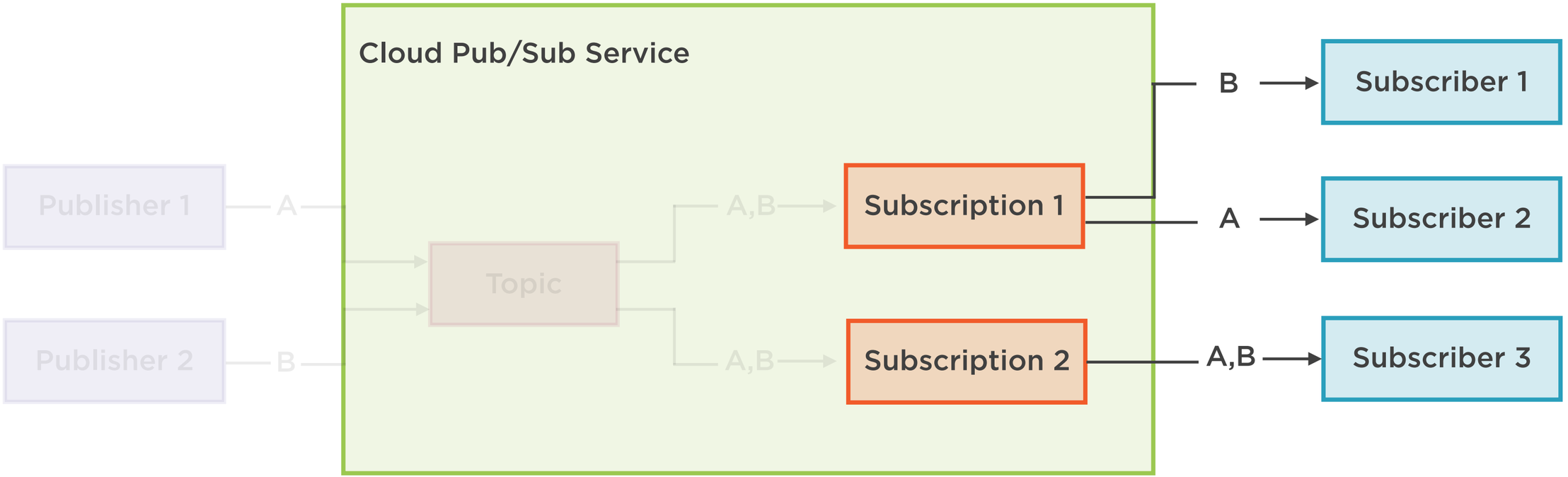
Two Subscriptions for the Topic



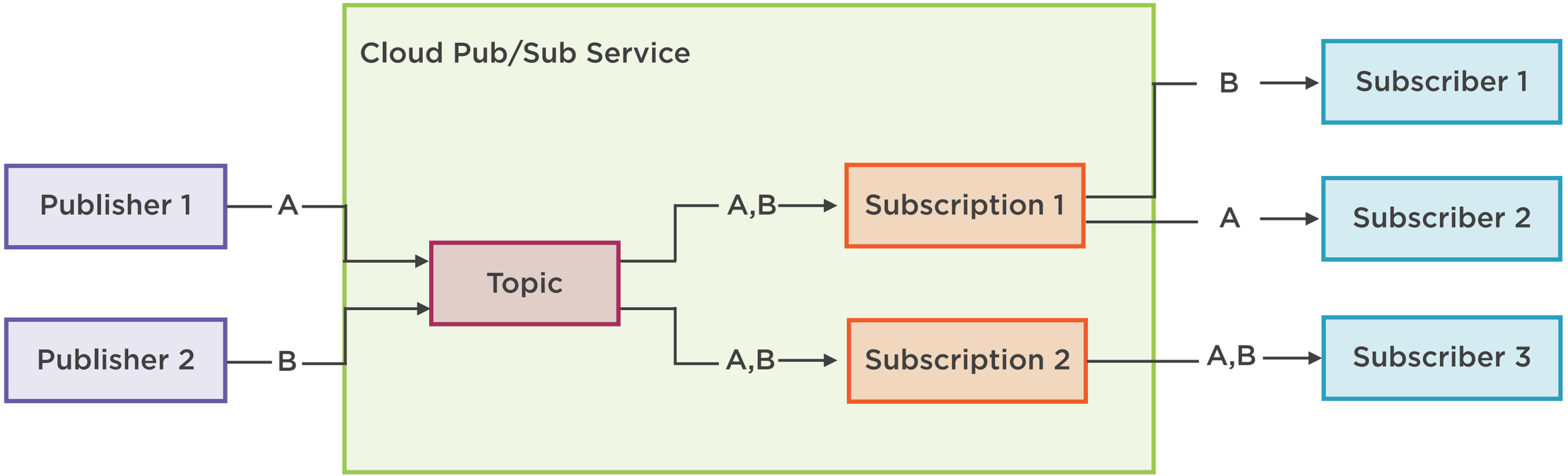
Same Messages Available on Both Subscriptions



Multiple Subscribers Share a Subscription



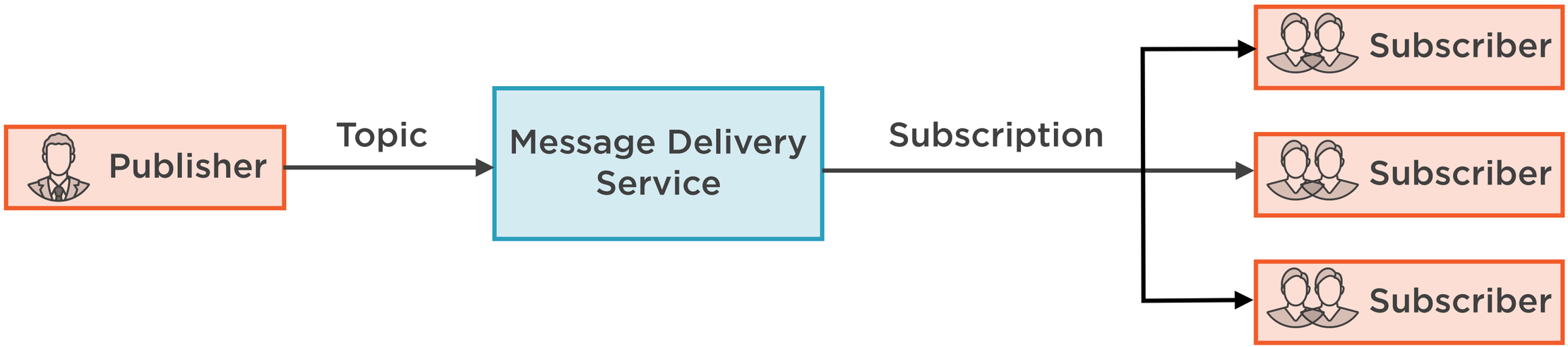
Fan-in and Fan-out



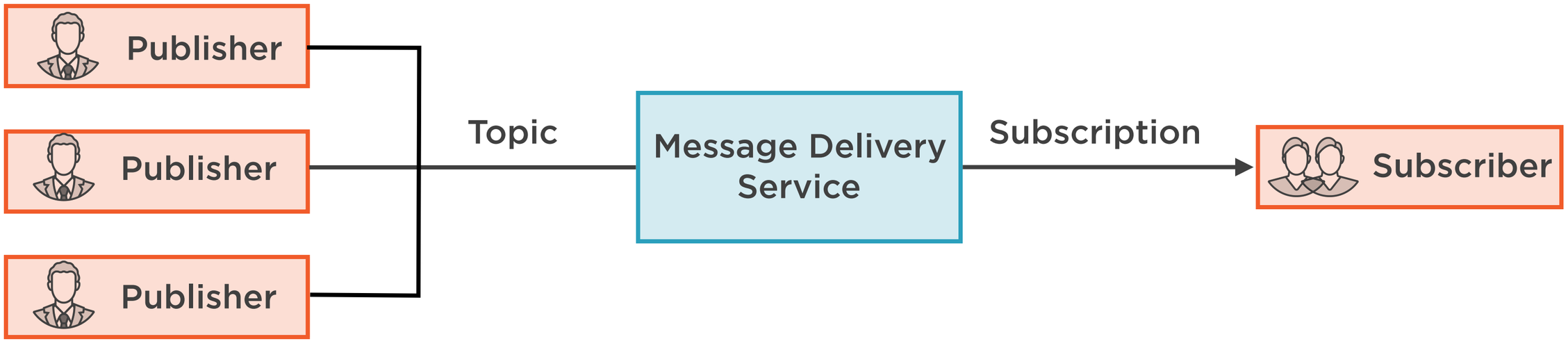
Publishers and Subscribers

Single Publisher, Single Subscriber

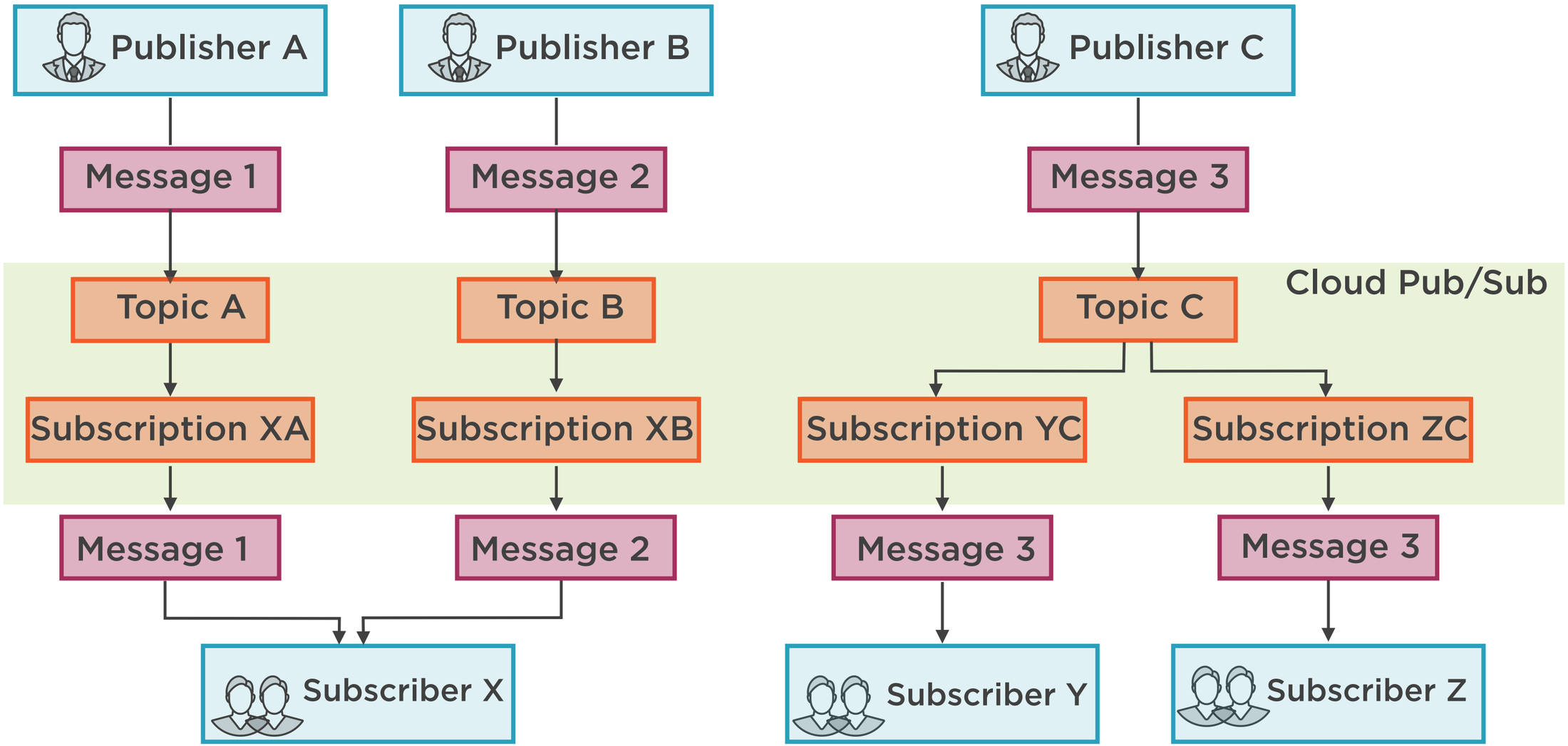
Single Publisher, Multiple Subscribers



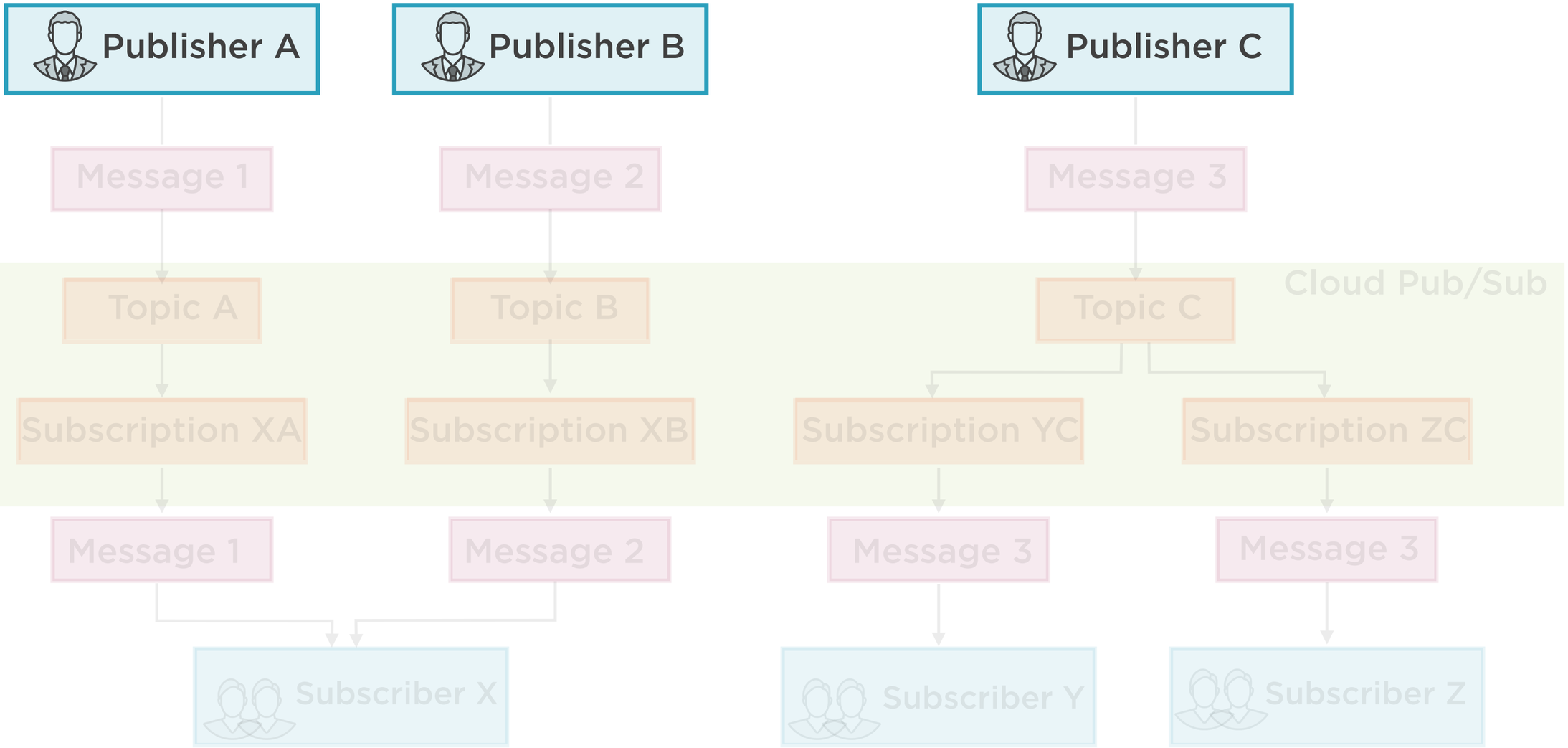
Multiple Publishers, Single Subscriber



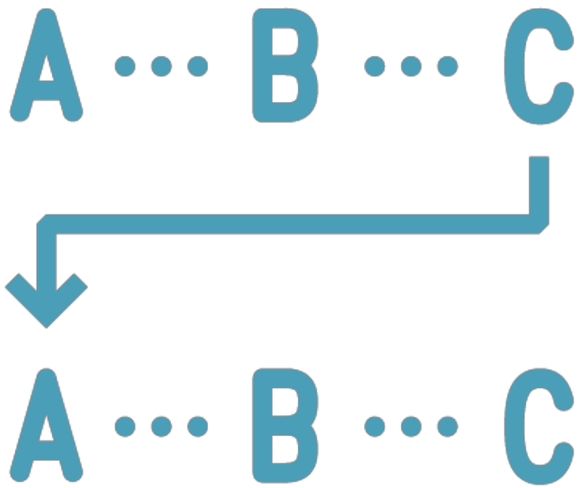
Publishers and Subscribers



Publishers and Subscribers



Publishers



App Engine and Cloud Function apps

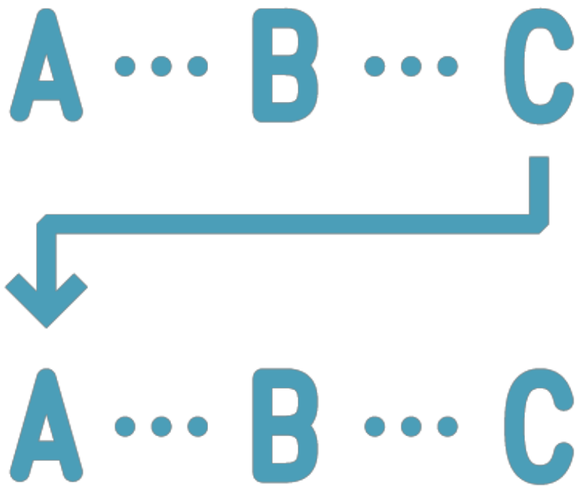
GCE, GKE apps

App running on third party network

Any mobile or web app

Any application that can make HTTPS requests to googleapis.com

Publishers

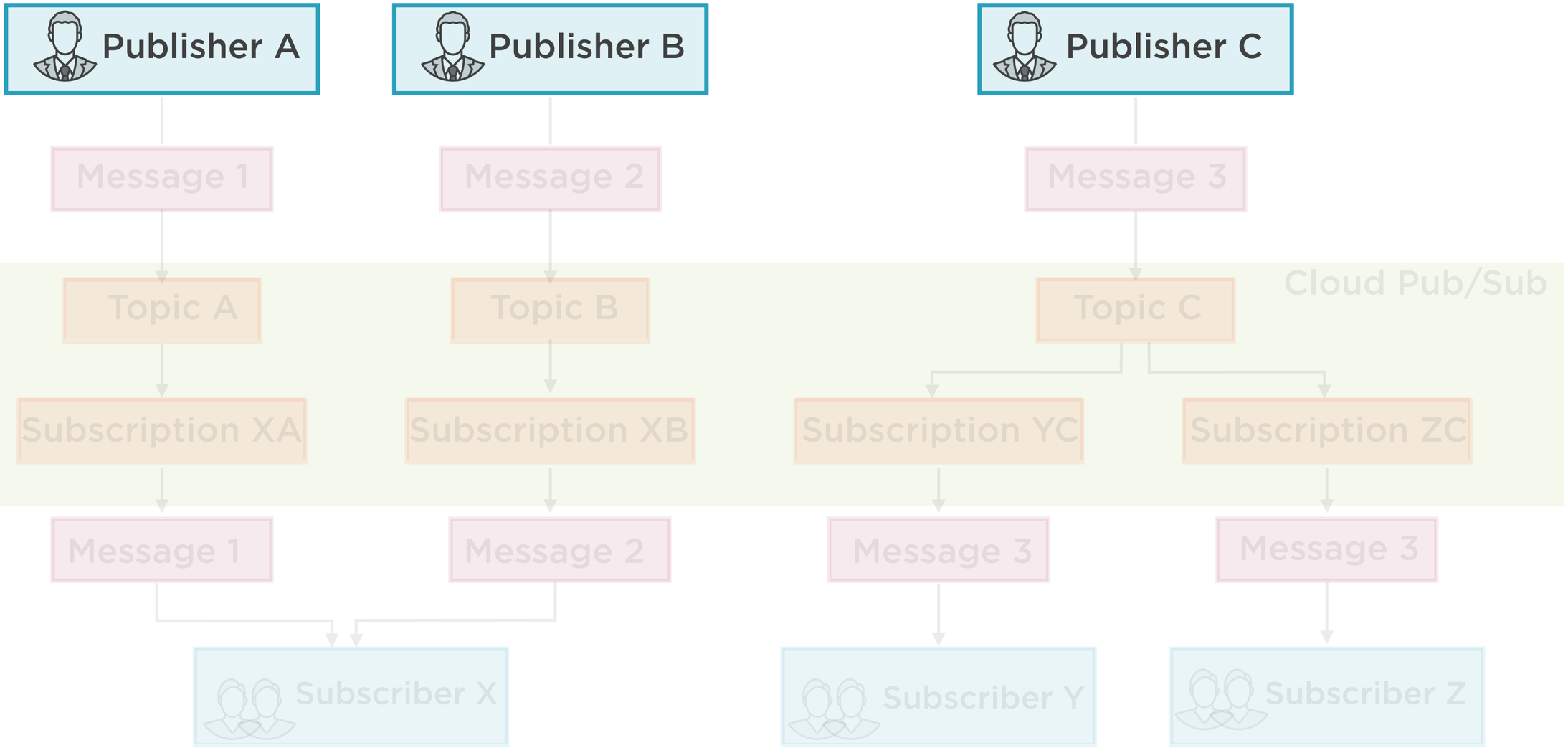


Publisher apps create and send messages on a **Topic**

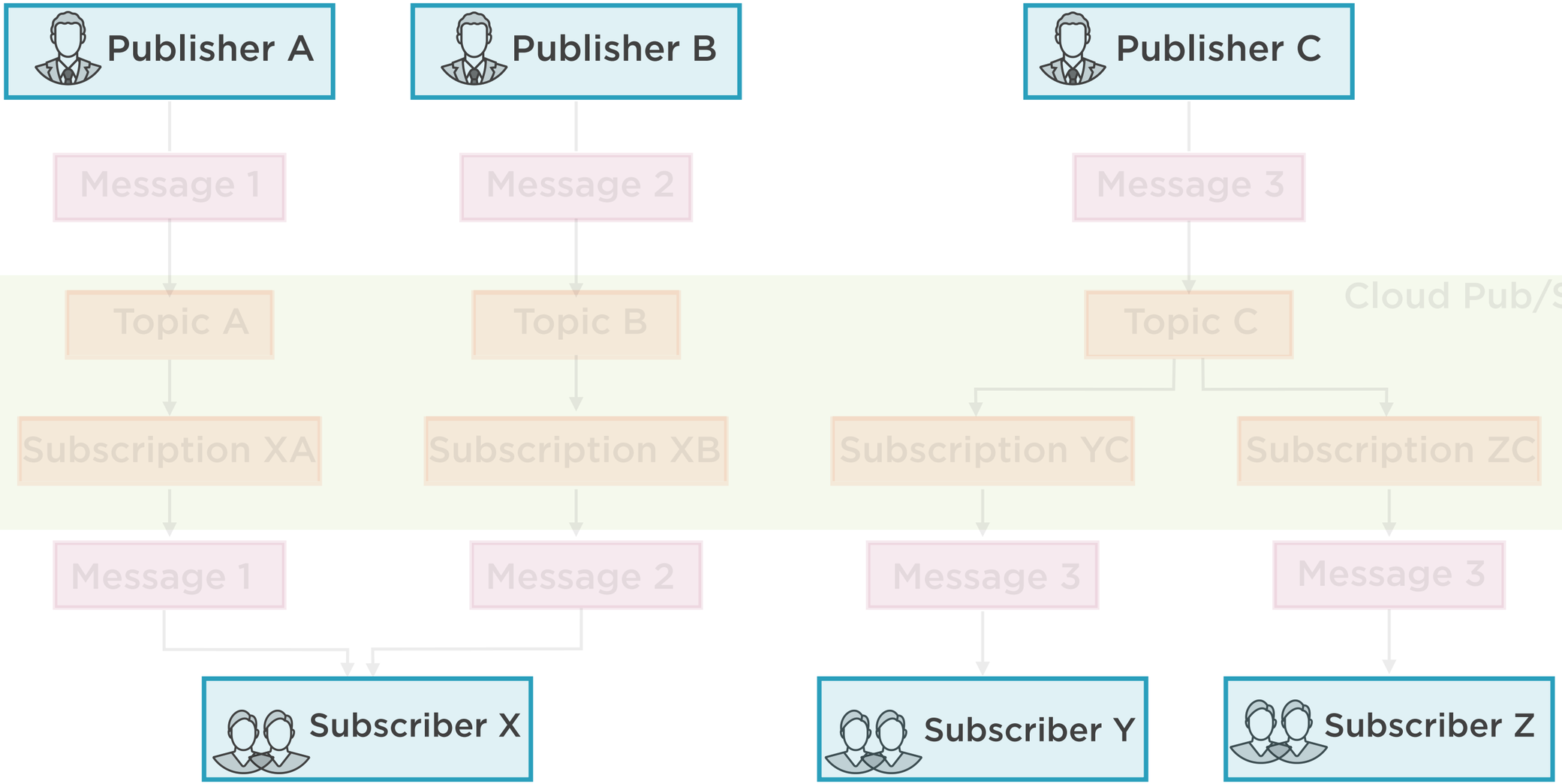
Messages persisted in a message store - until delivered/acknowledged

One queue per subscription

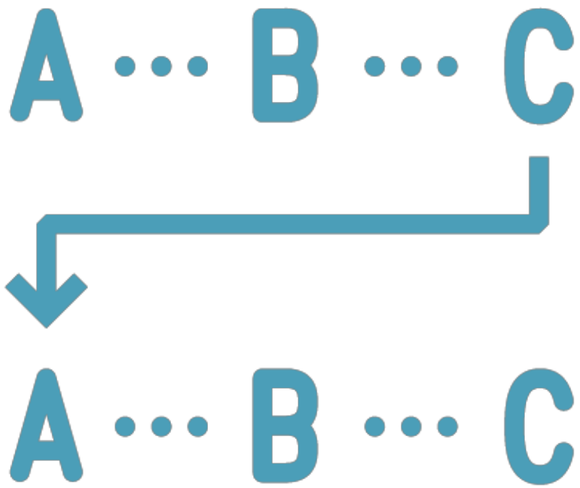
Publishers and Subscribers



Publishers and Subscribers



Subscribers



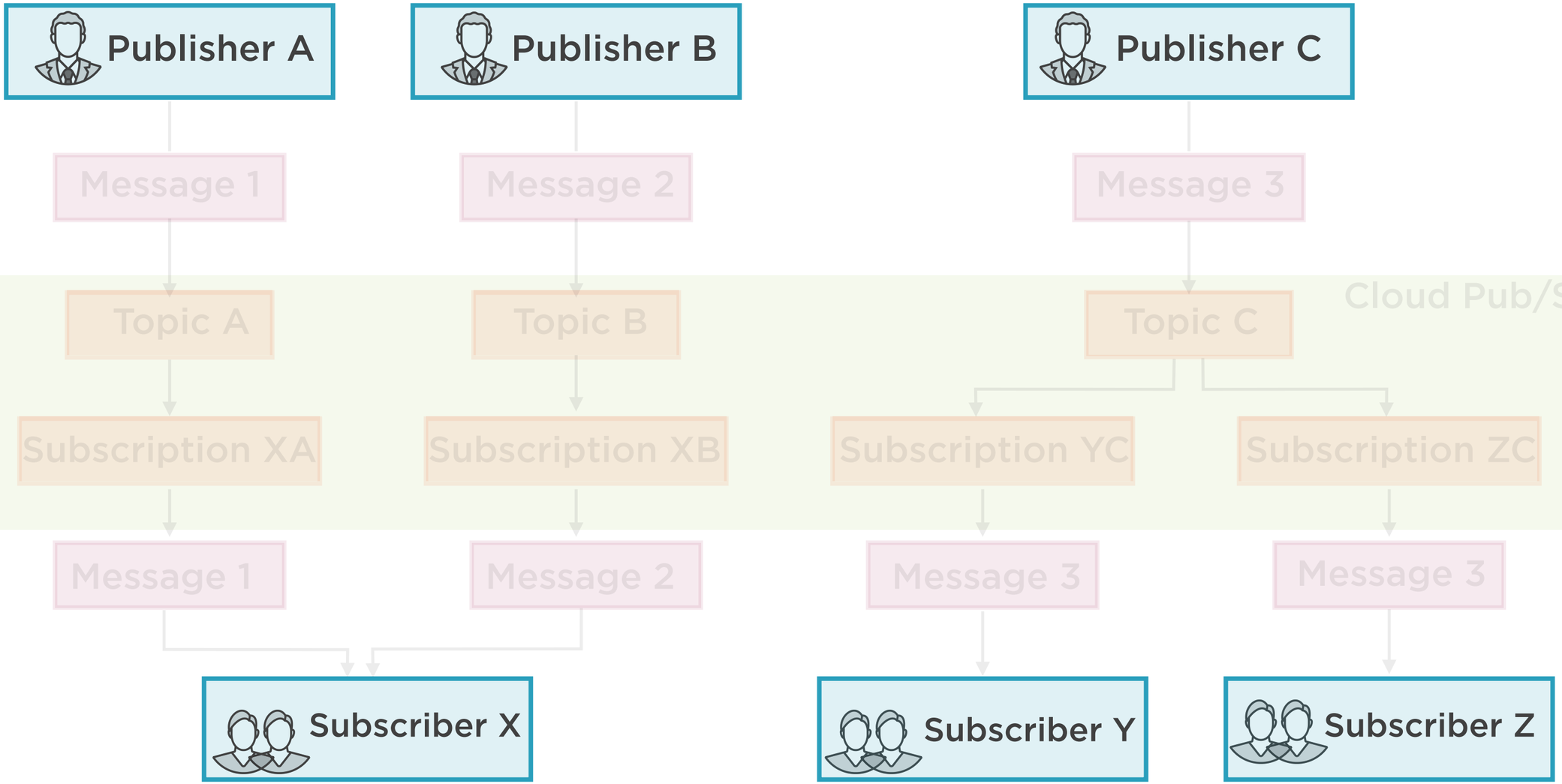
Subscriber apps subscribe to a topic to receive messages

Once acknowledged by subscriber, message deleted from queue

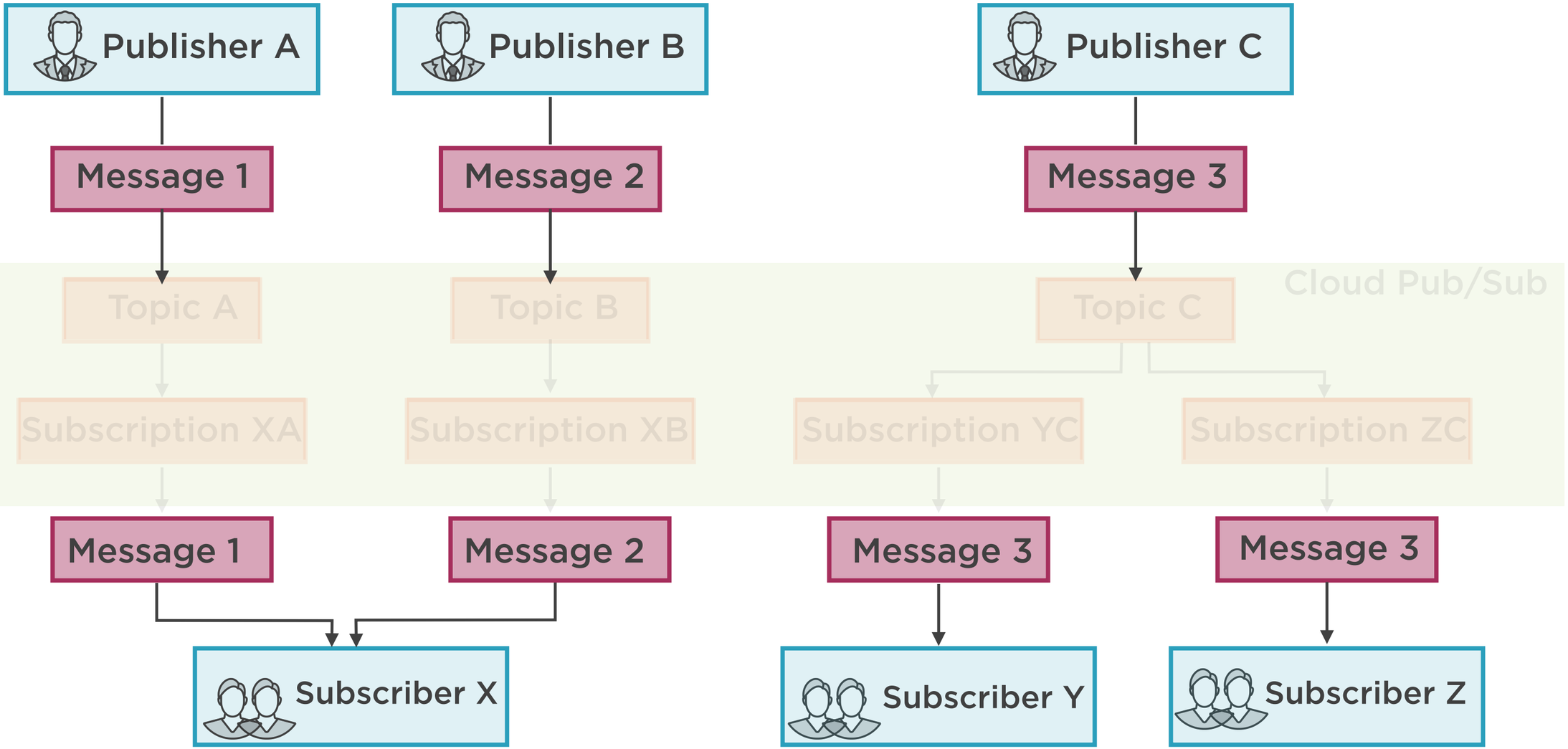
Push and Pull subscriptions

Messages, Topics and Subscriptions

Publishers and Subscribers



Publishers and Subscribers



Messages

Message data

Attributes: Key-value pairs set by publisher

Organized into topics

Stored until delivered and acknowledged

Messages

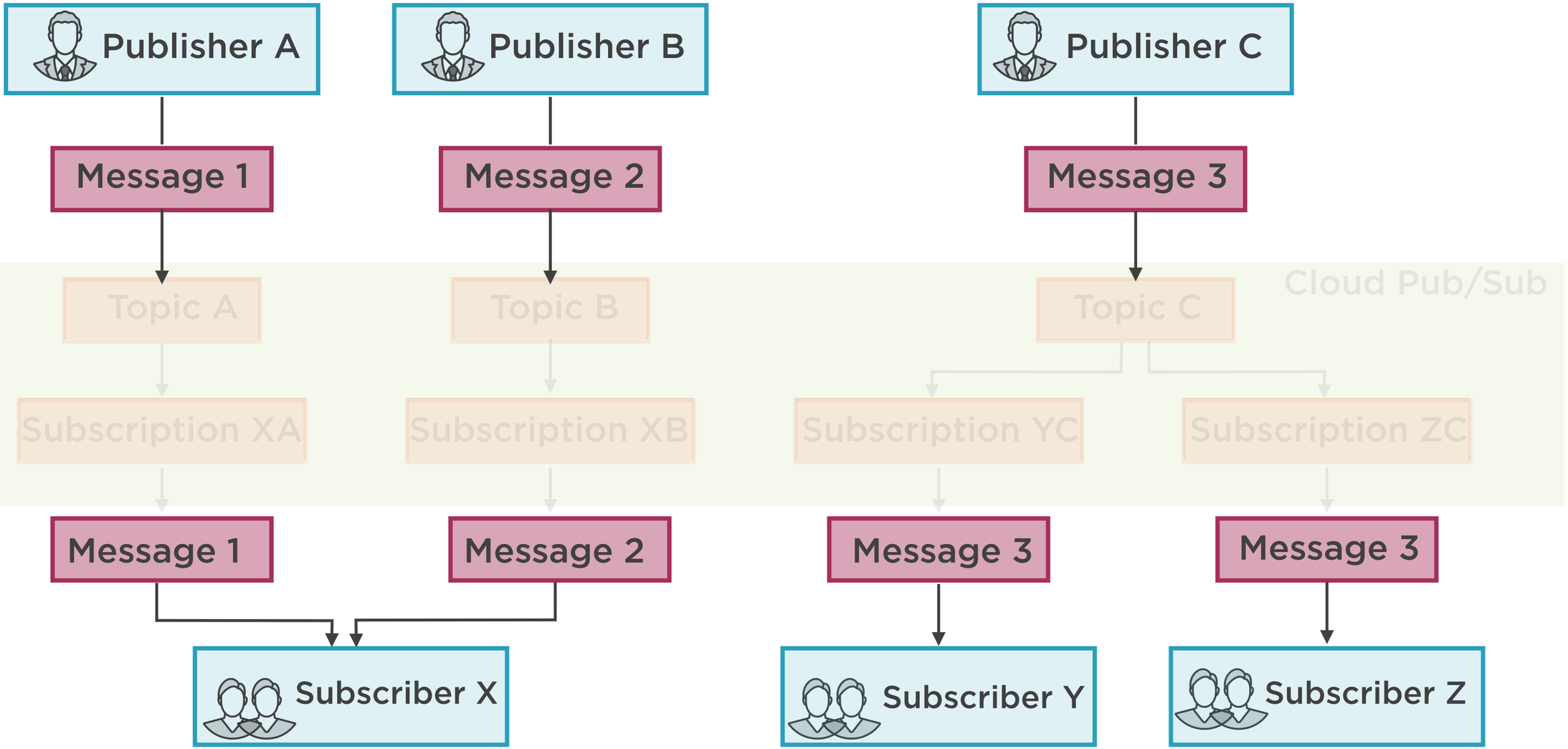
Pub/Sub provides

- At-least-once delivery
- No guaranteed ordering

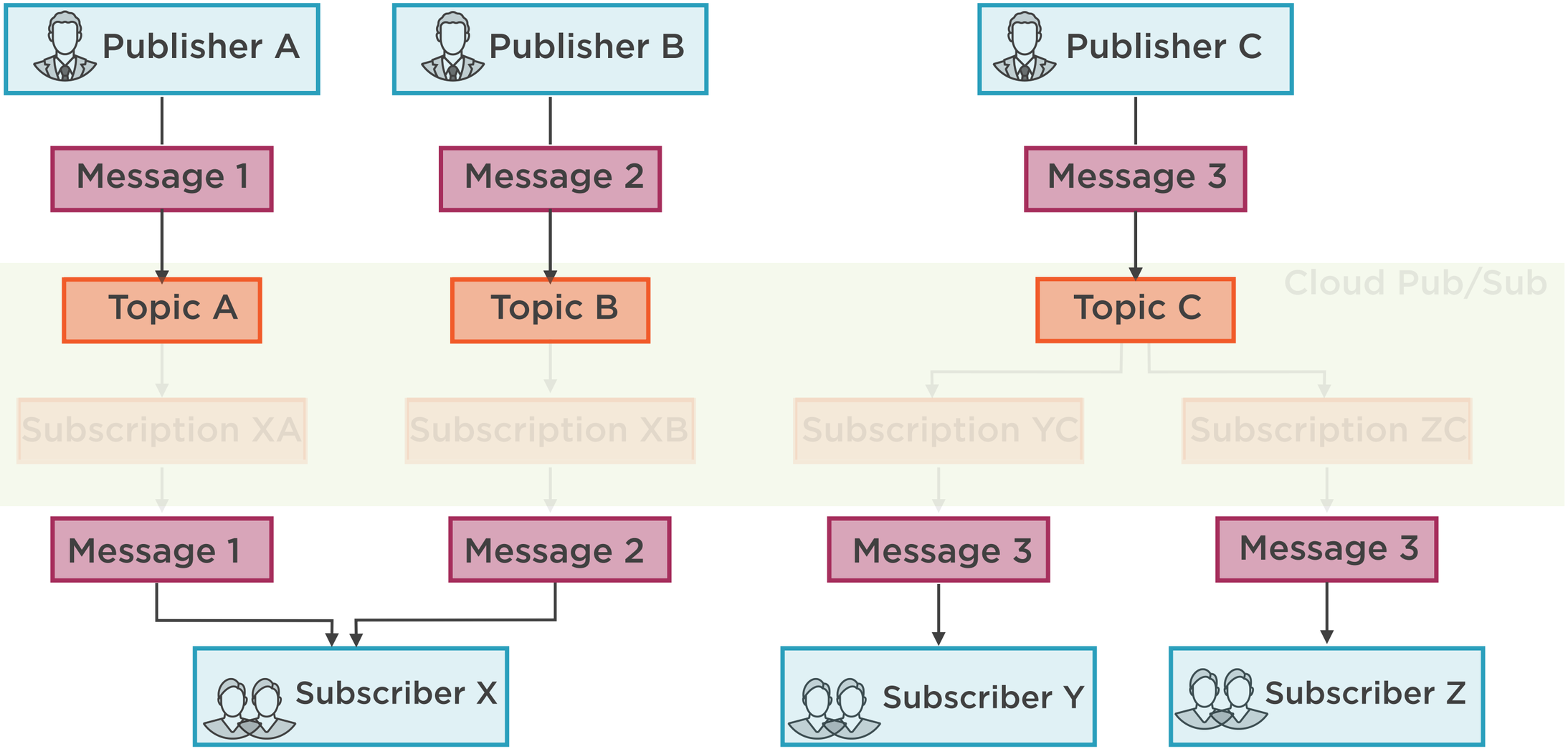
Use Dataflow with Pub/Sub

- Ordering
- De-duplication

Publishers and Subscribers



Publishers and Subscribers



Topics

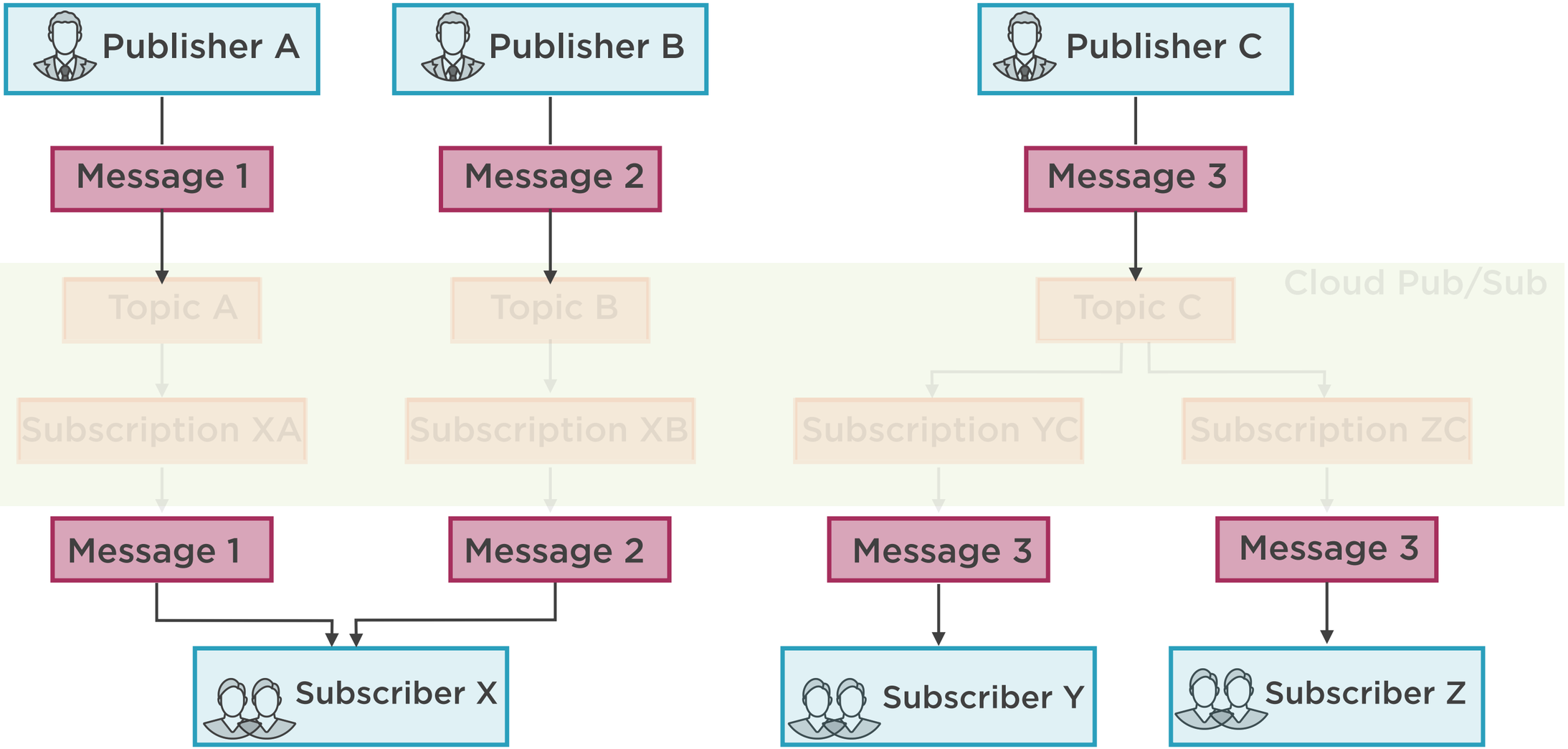
Named resources to which publishers send messages

Used to organize messages

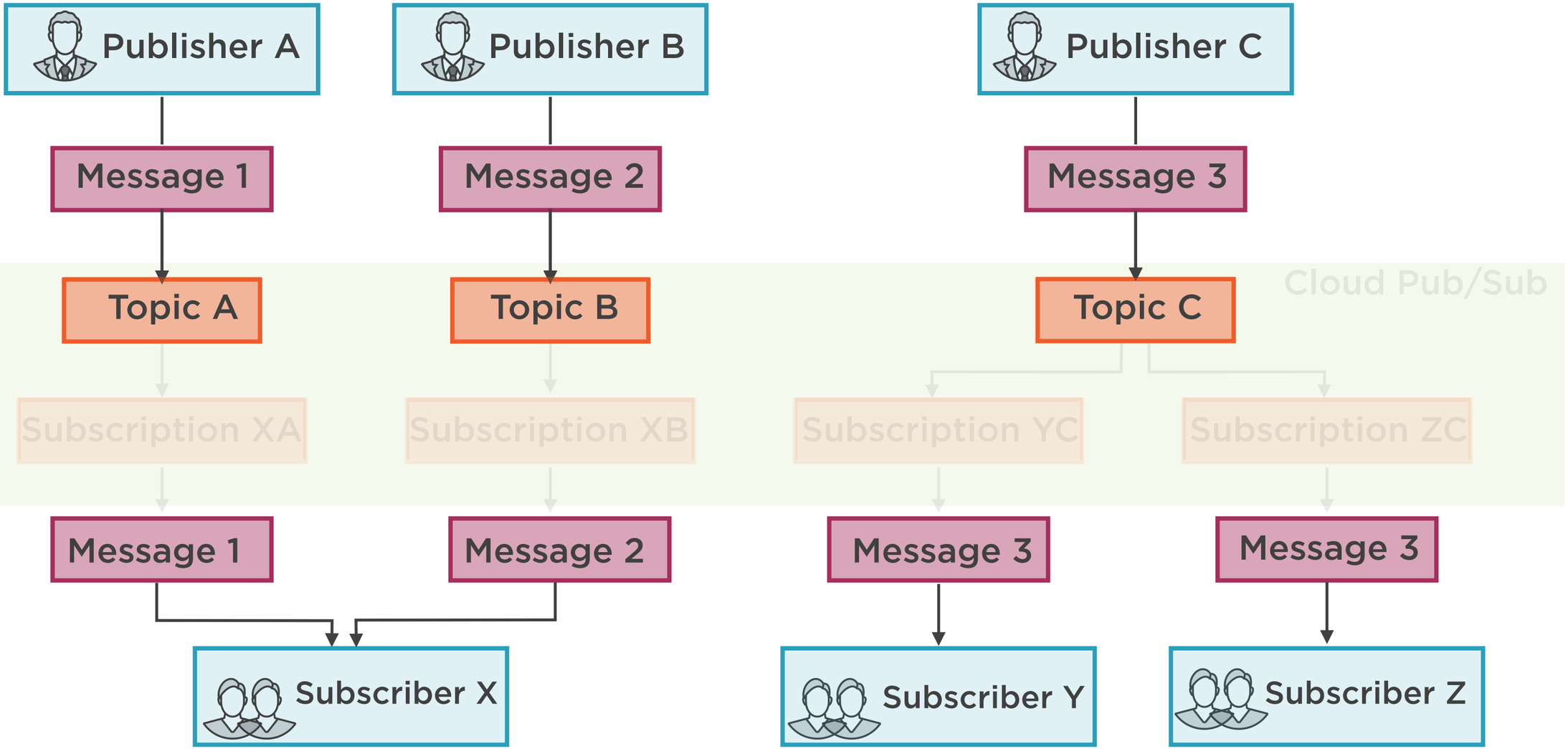
Publishers send messages to topics

Subscribers subscribe to topics

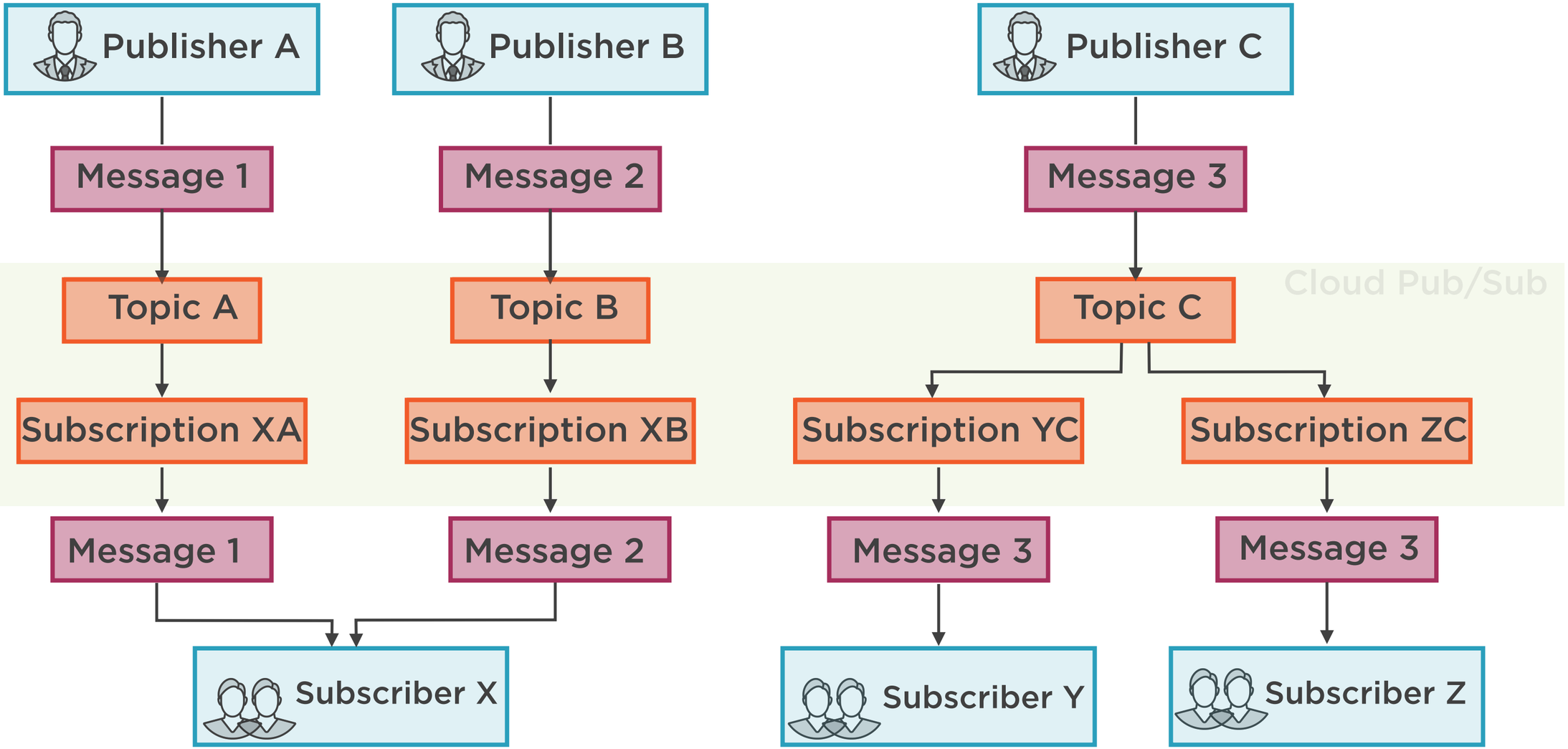
Publishers and Subscribers



Publishers and Subscribers



Publishers and Subscribers



Subscriptions

Pub/Sub sends each message on a topic to each subscription individually

Message persisted inside Pub/Sub

Not deleted until at least one subscriber per subscription has acknowledged

Subscriptions

Pub/Sub sends each message on a topic to each subscription individually

Each subscription then might

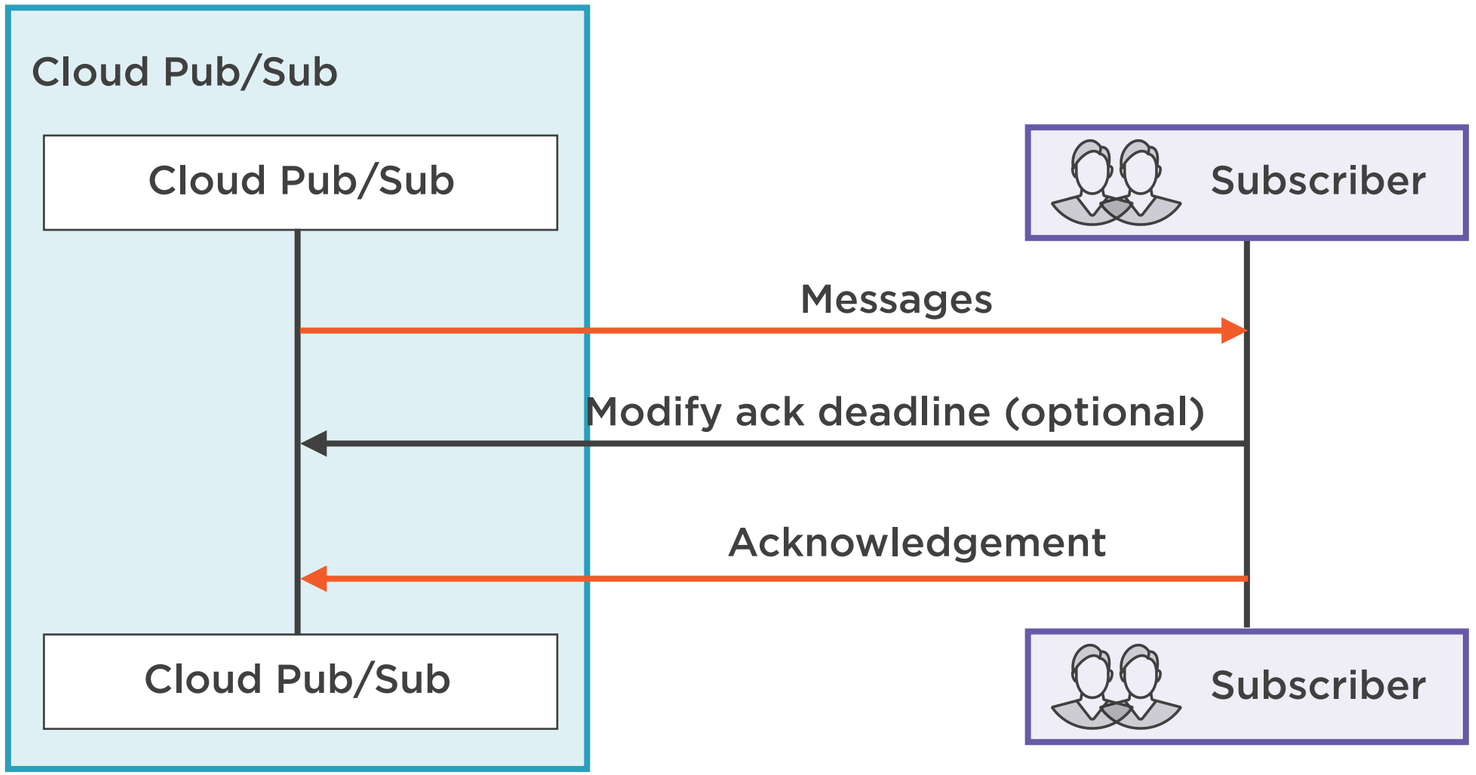
- Either push messages to subscriber endpoint
- Or wait for subscriber to pull them at a later point

Subscriptions

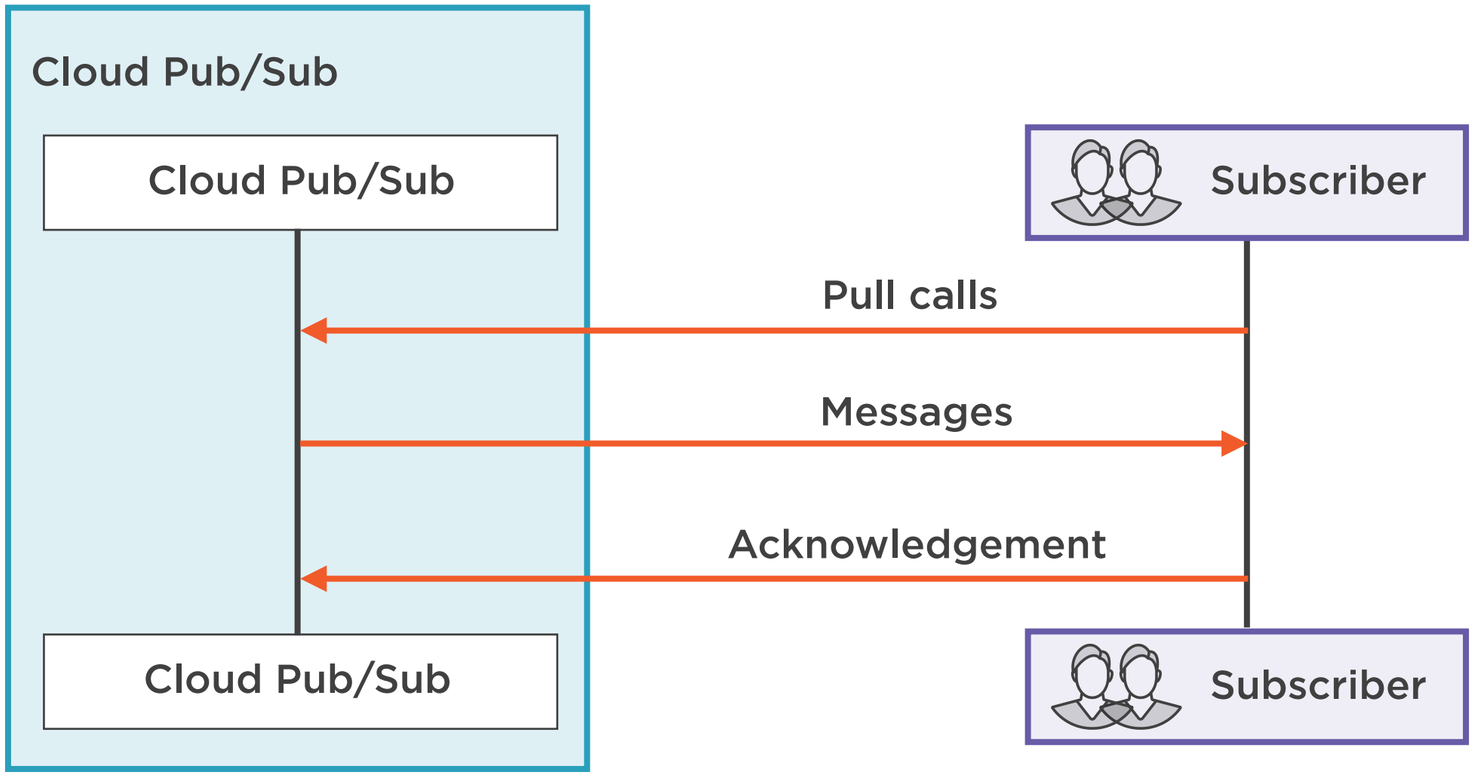
Push subscriptions: Any app that can make HTTPS requests to googleapis.com

Pull subscribers: Webhook endpoint that can accept HTTPS POST requests

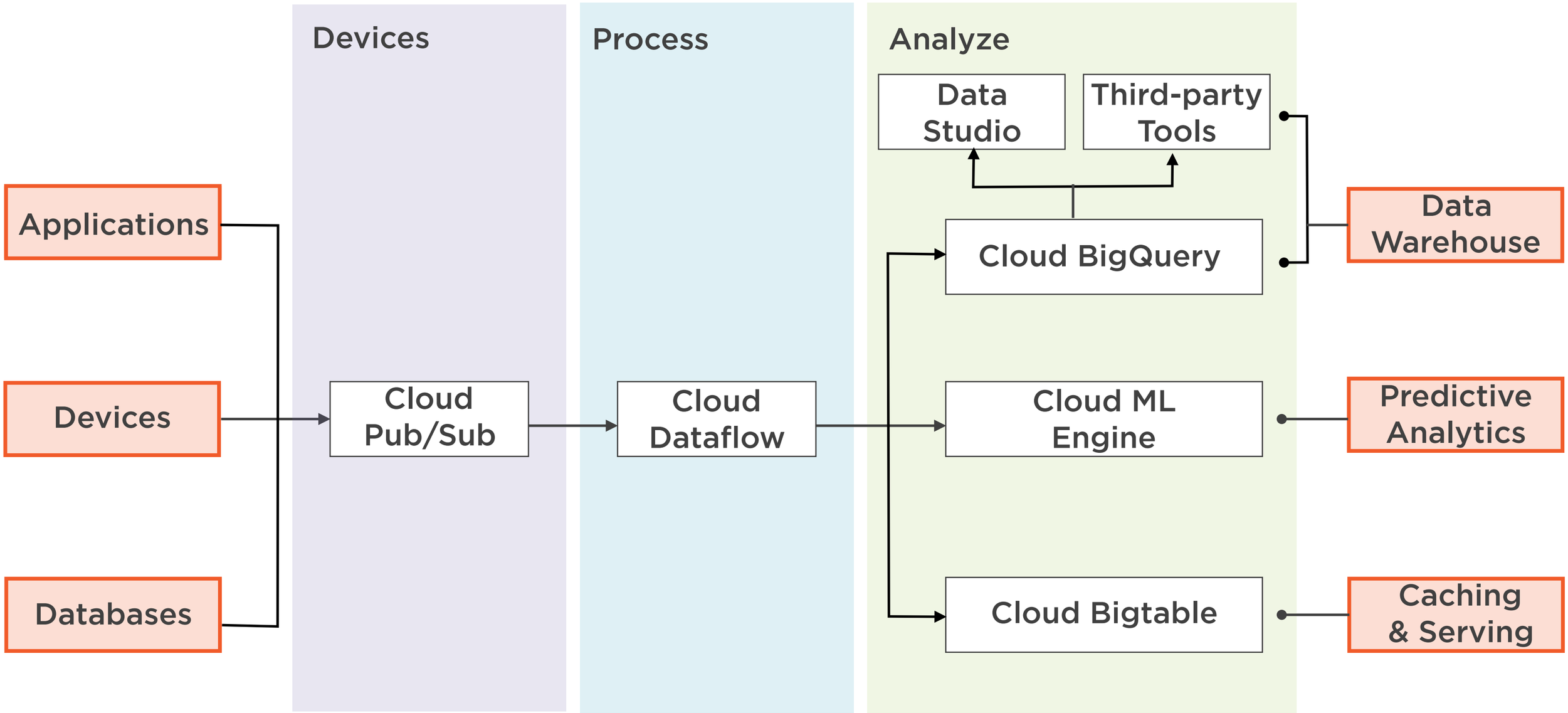
Push Subscriptions



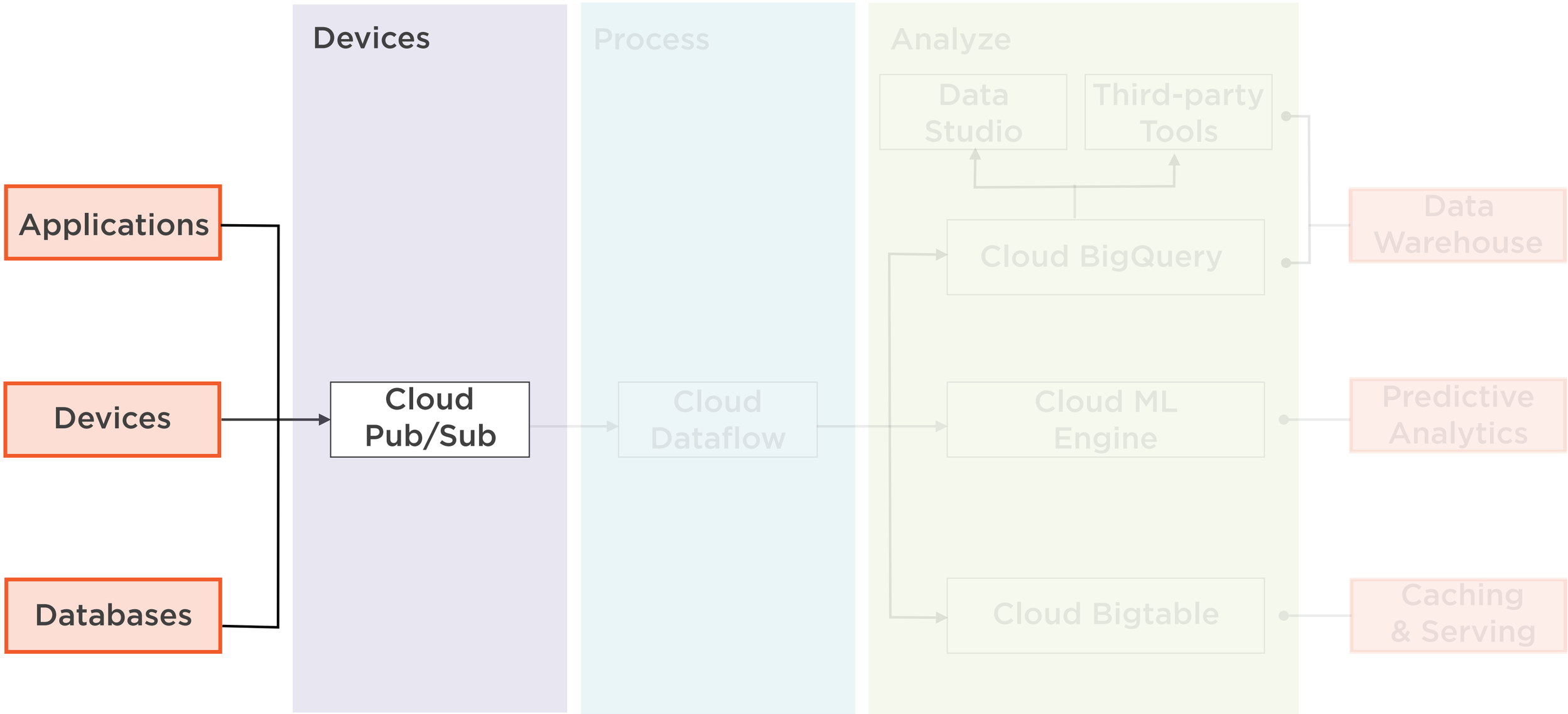
Pull Subscriptions



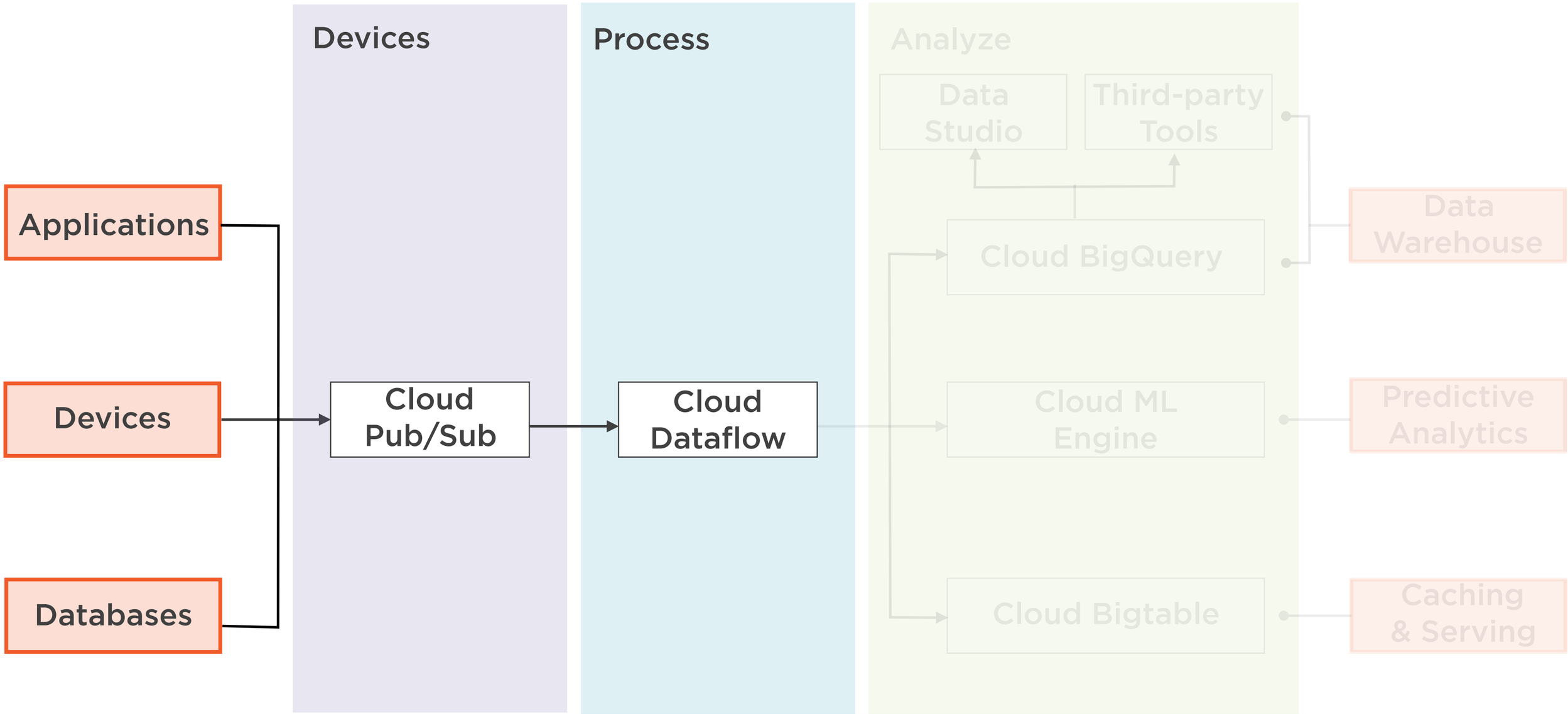
Streaming Analytics



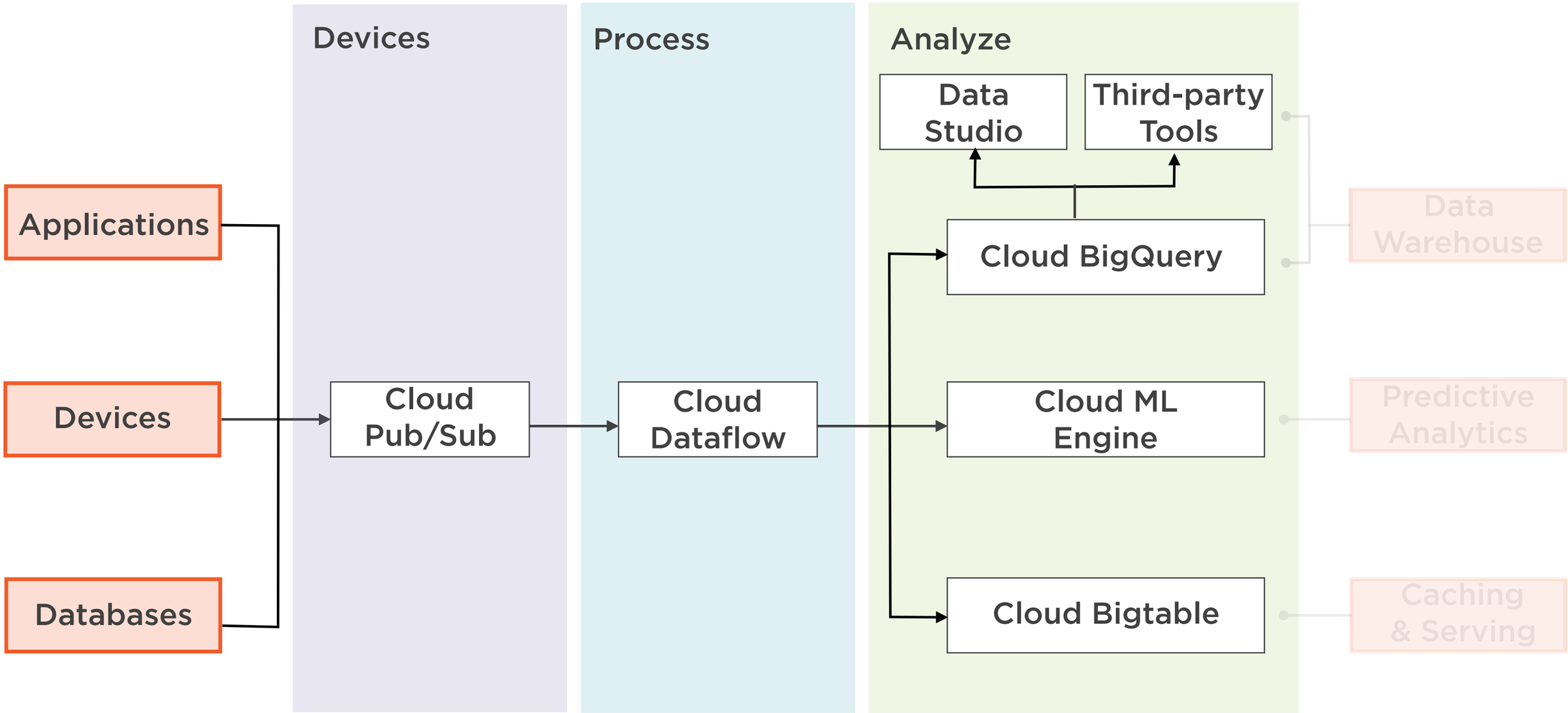
Stream Into Pub/Sub



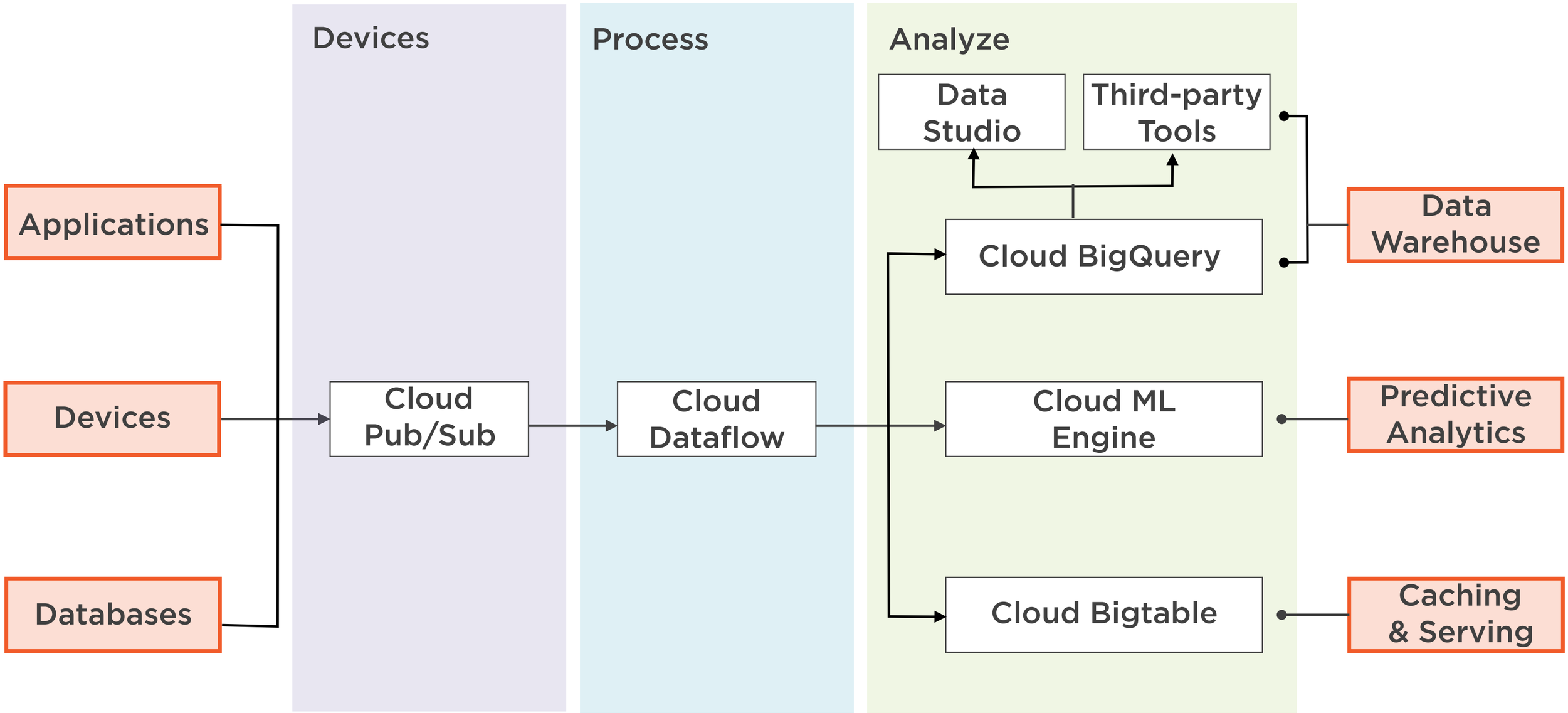
Transform Using Dataflow



Analyze and Visualize Data

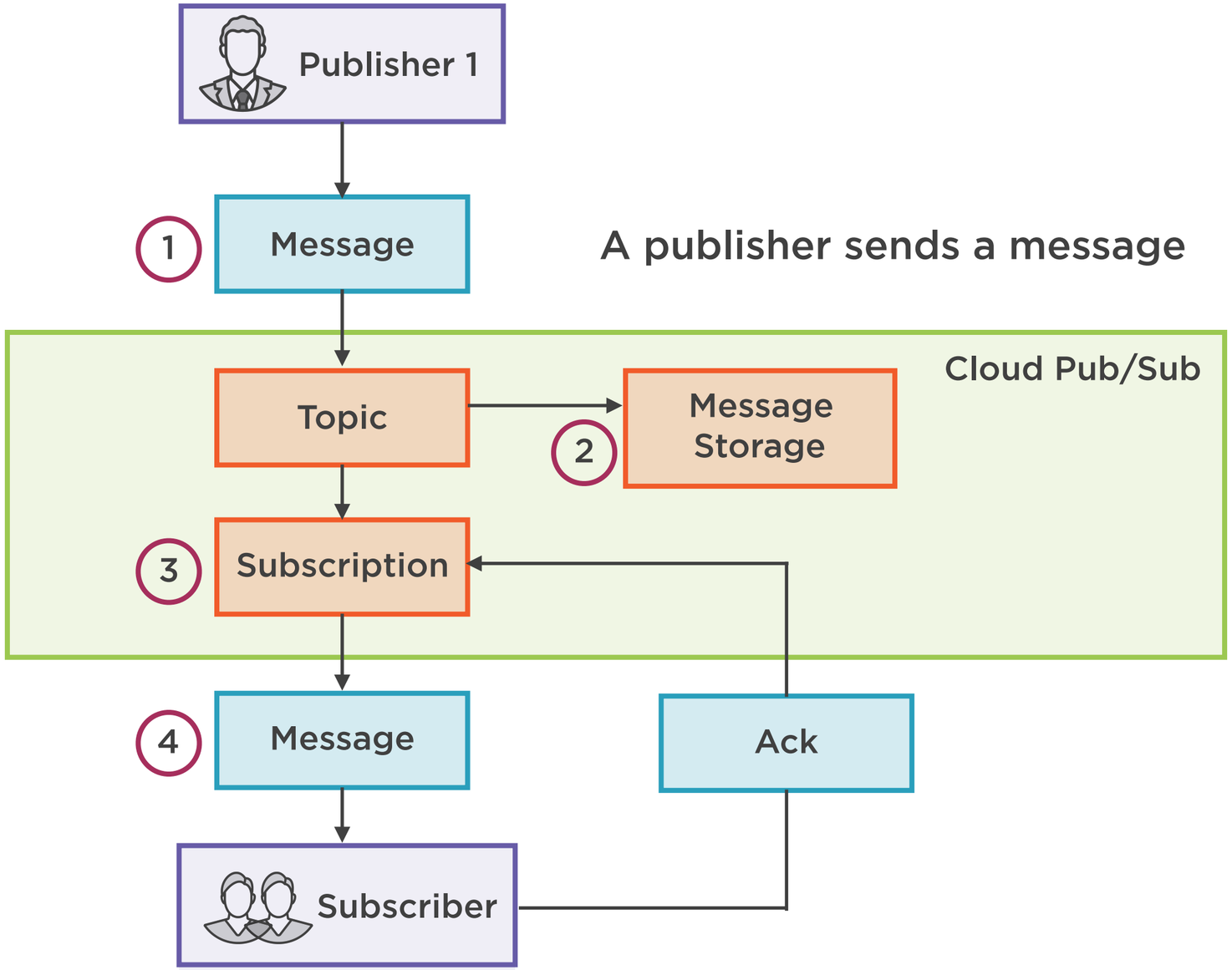


Store in a Data Warehouse

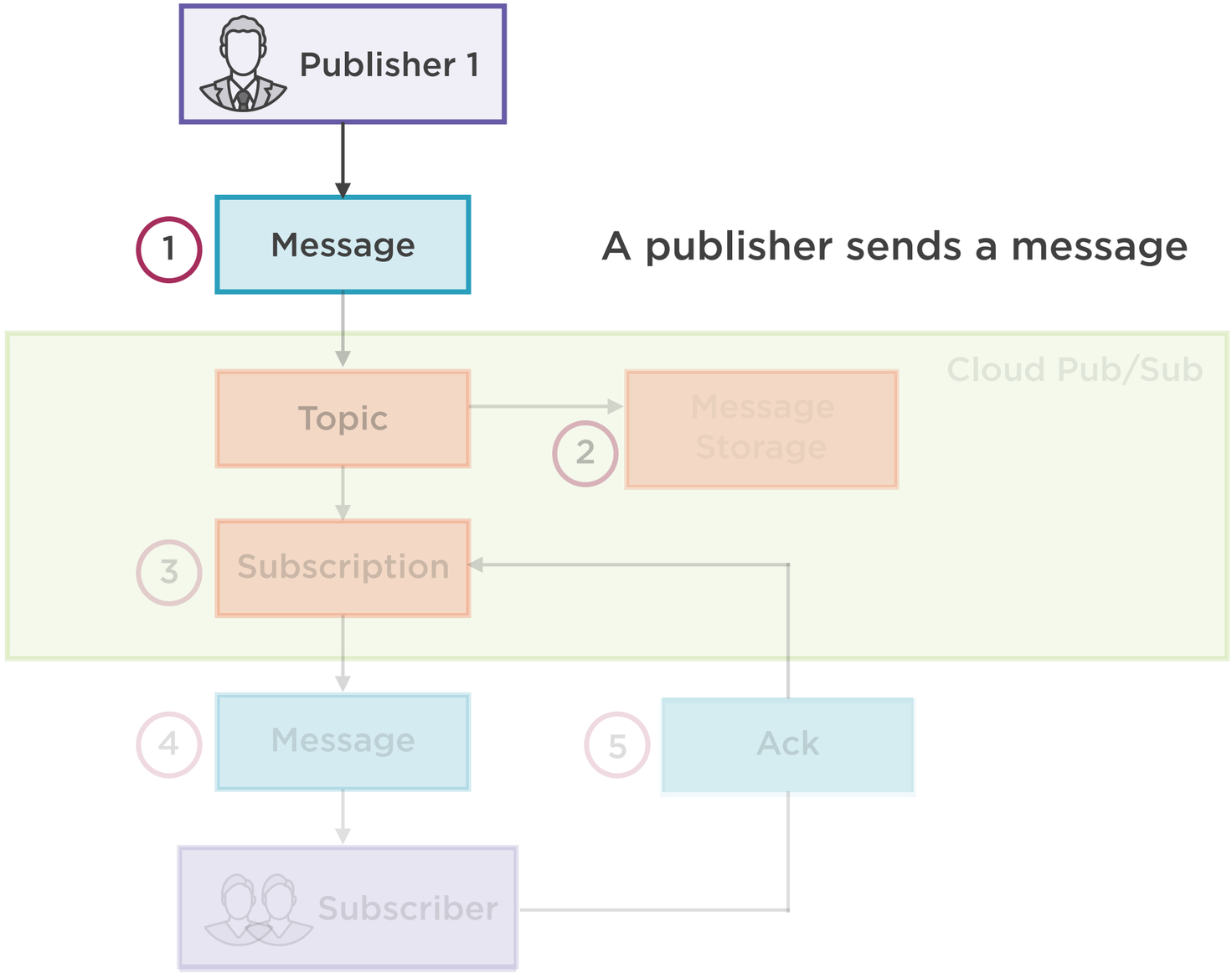


Pub/Sub Message Flow

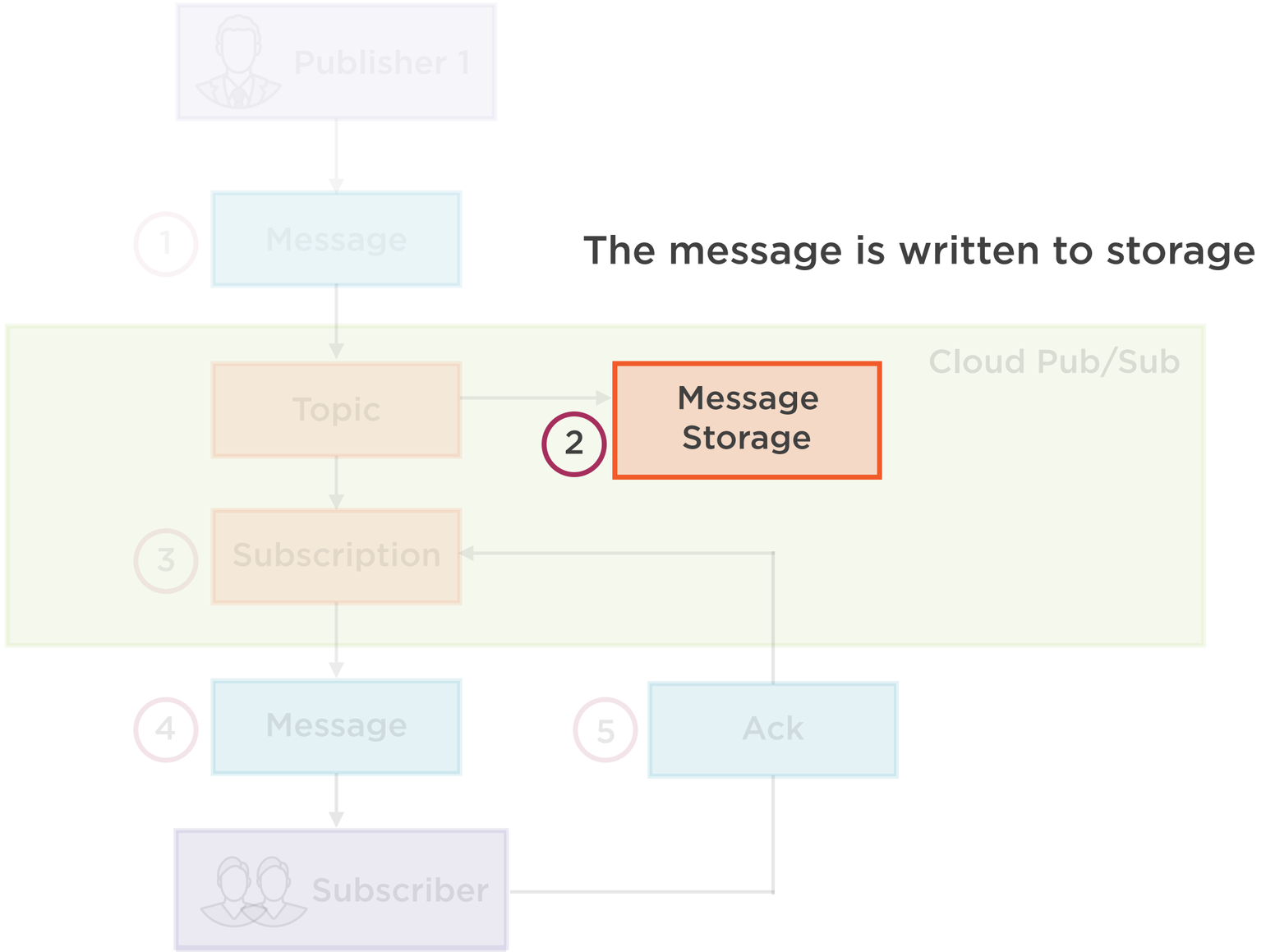
Pub/Sub Messaging



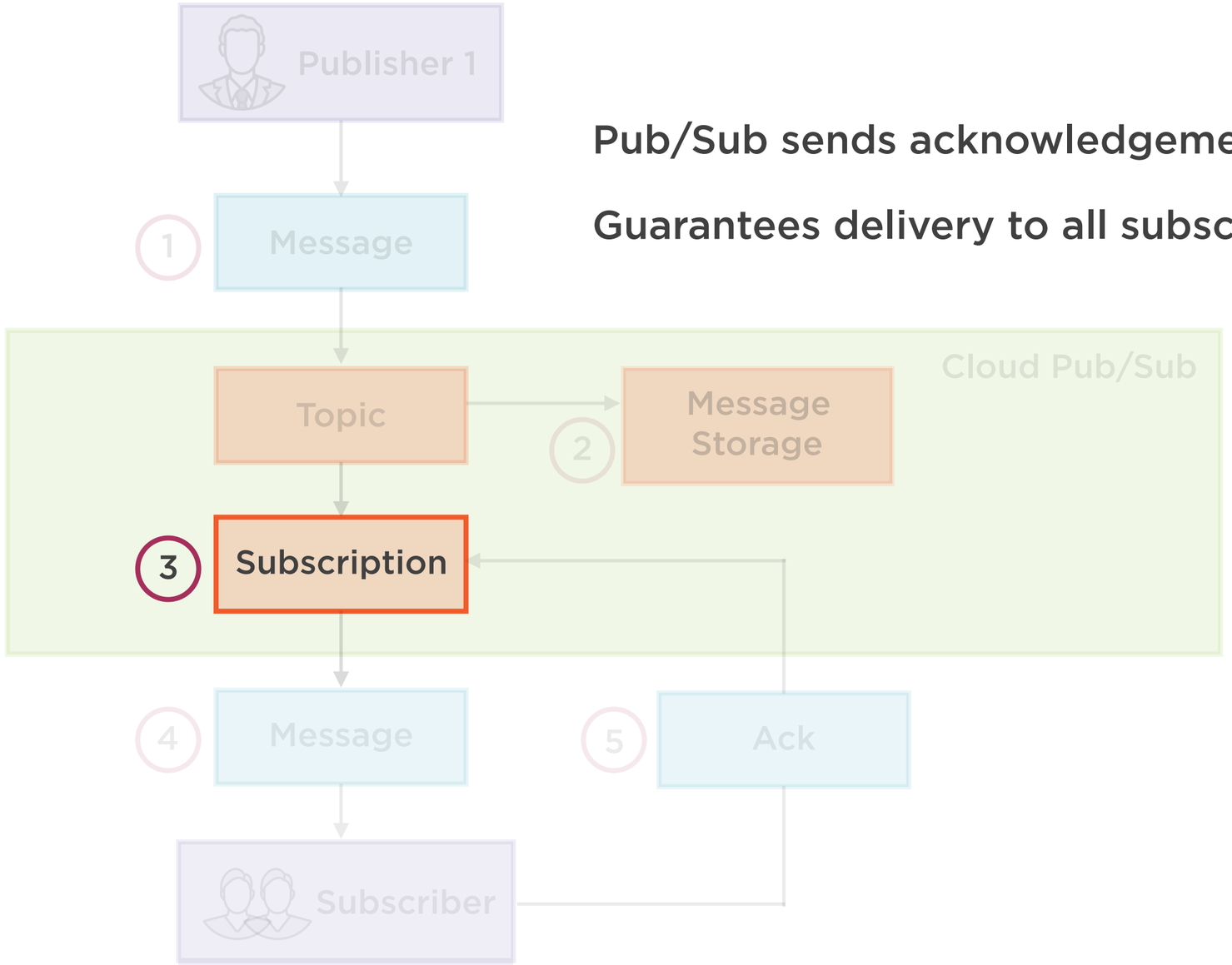
Pub/Sub Messaging



Pub/Sub Messaging



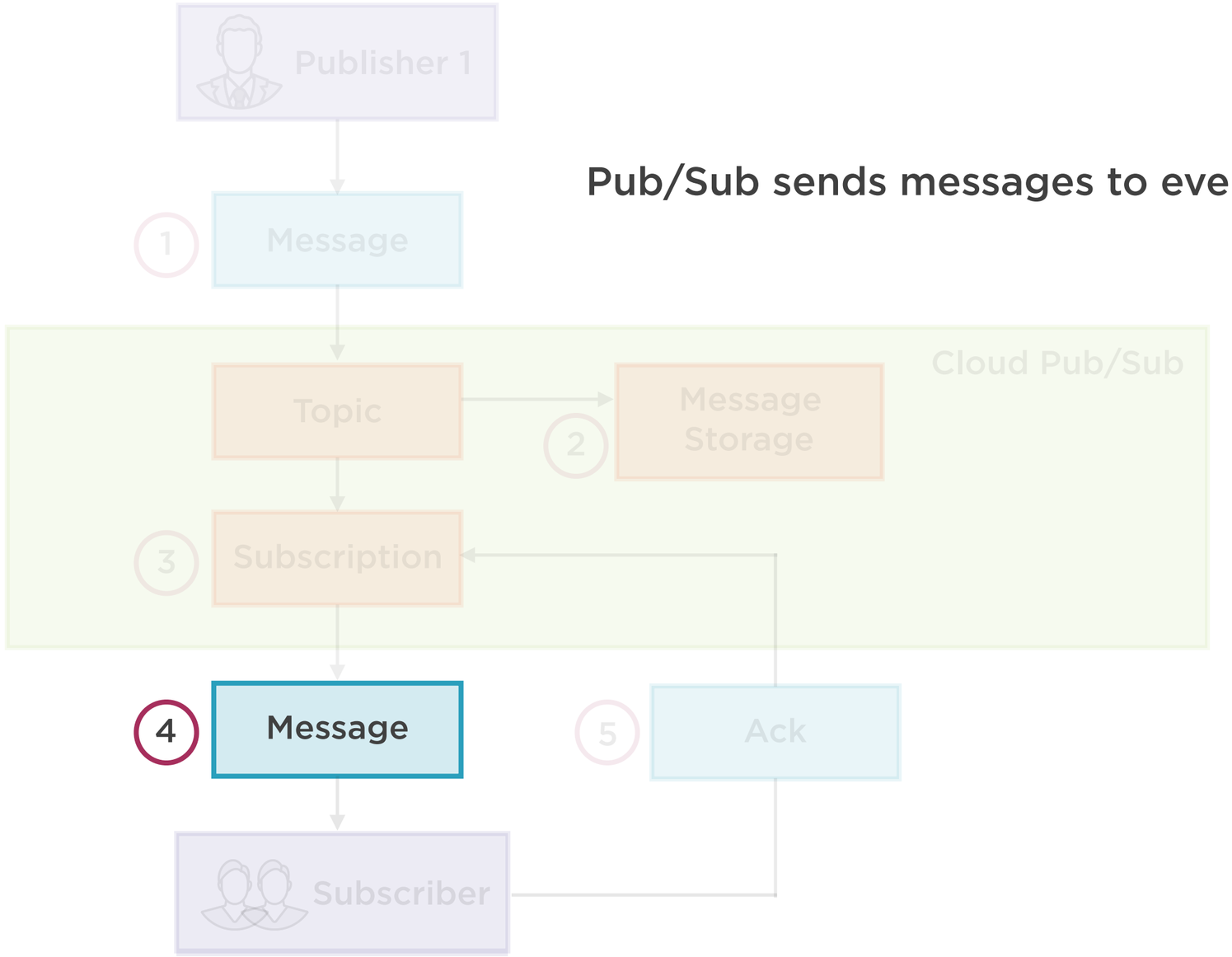
Pub/Sub Messaging



Pub/Sub sends acknowledgement to publisher

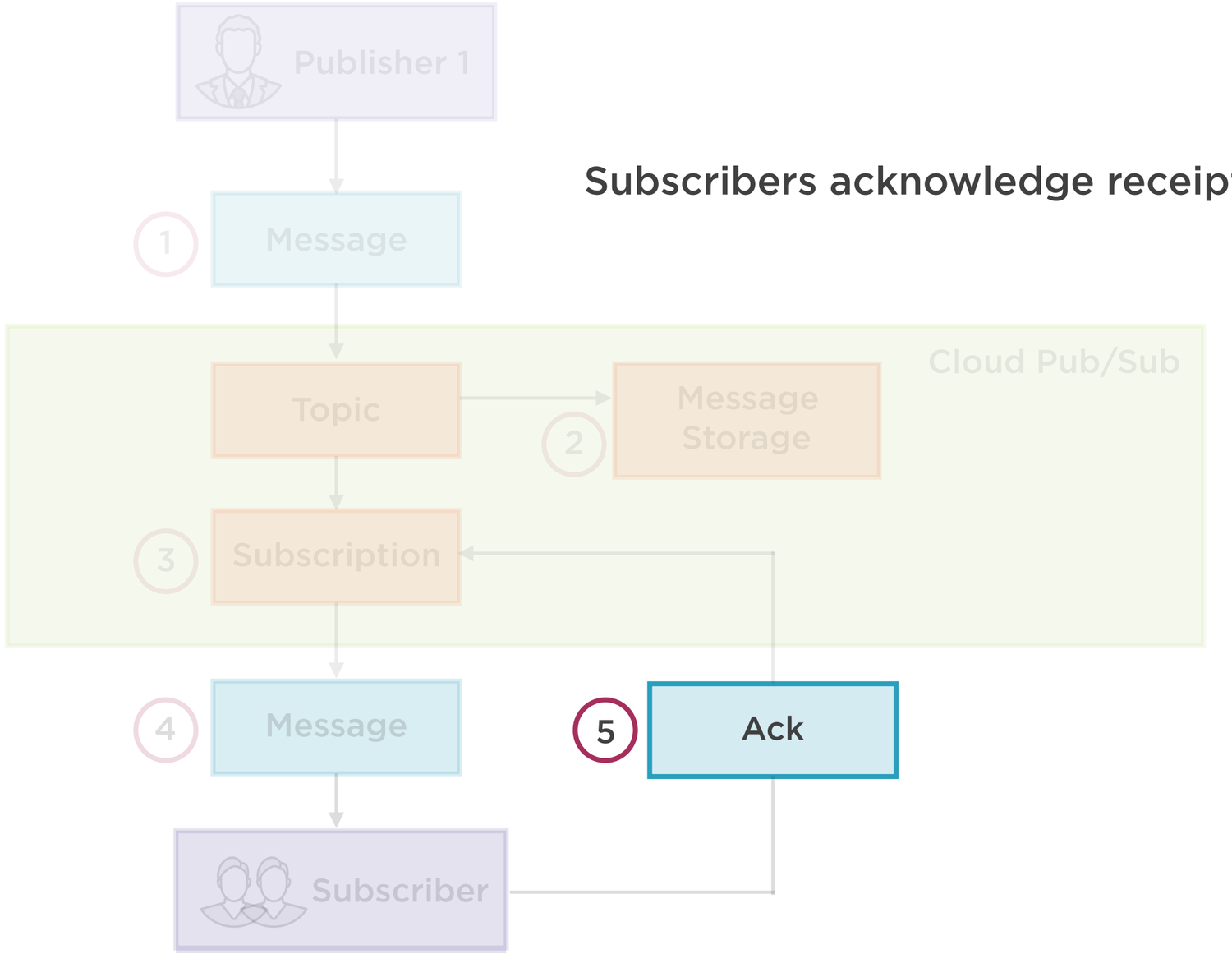
Guarantees delivery to all subscribers

Pub/Sub Messaging



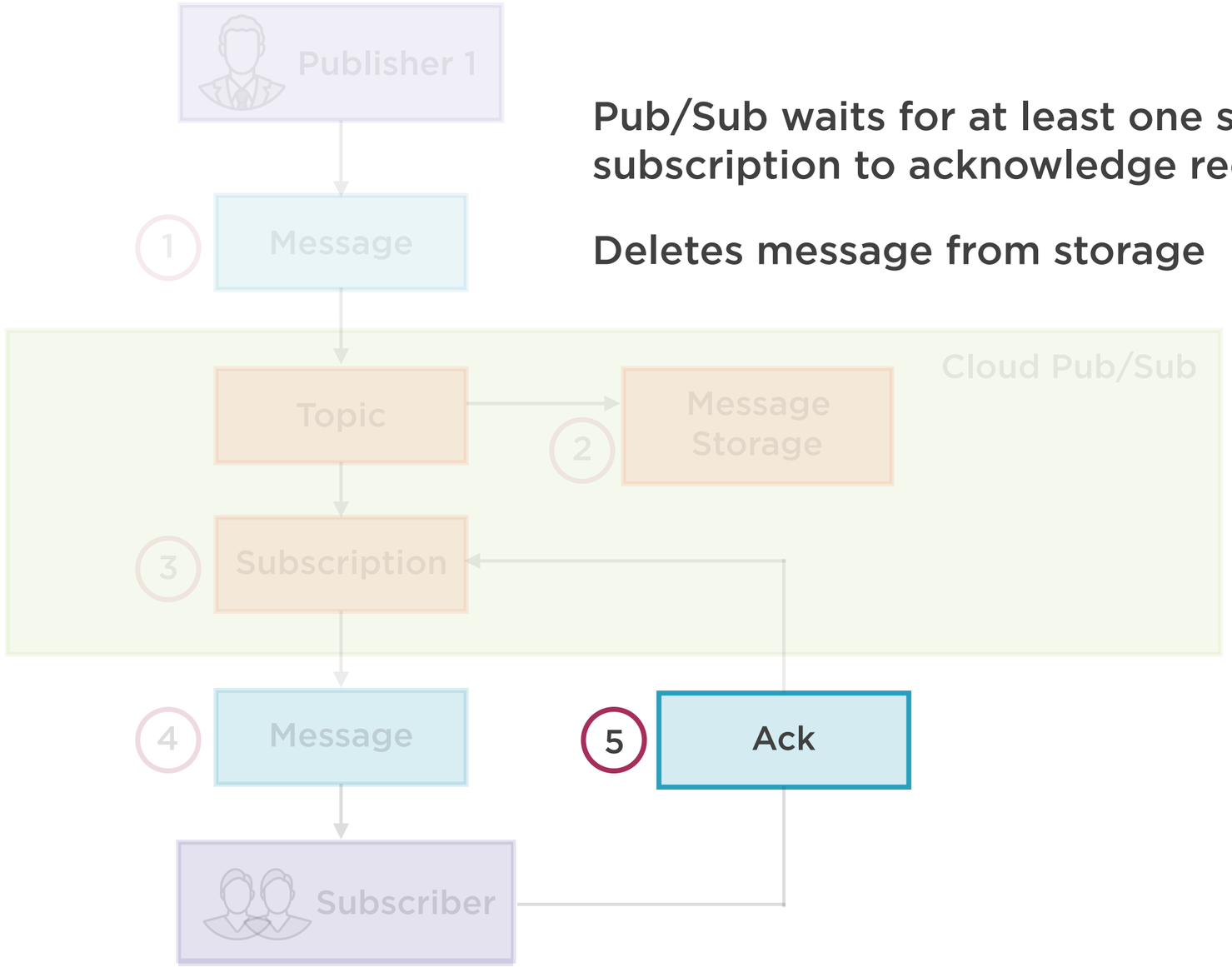
Pub/Sub sends messages to every subscription

Pub/Sub Messaging



Subscribers acknowledge receipt to Pub/Sub

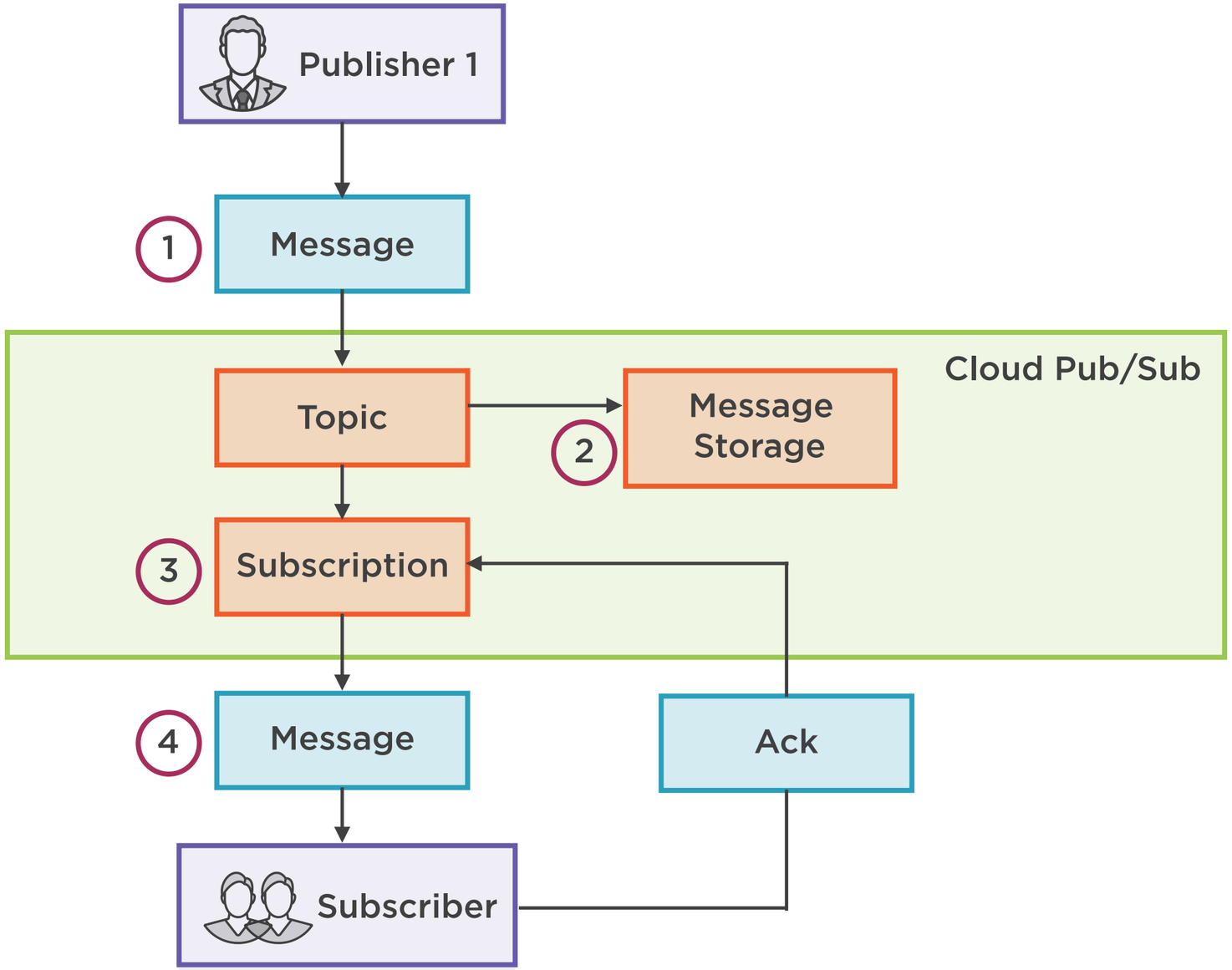
Pub/Sub Messaging



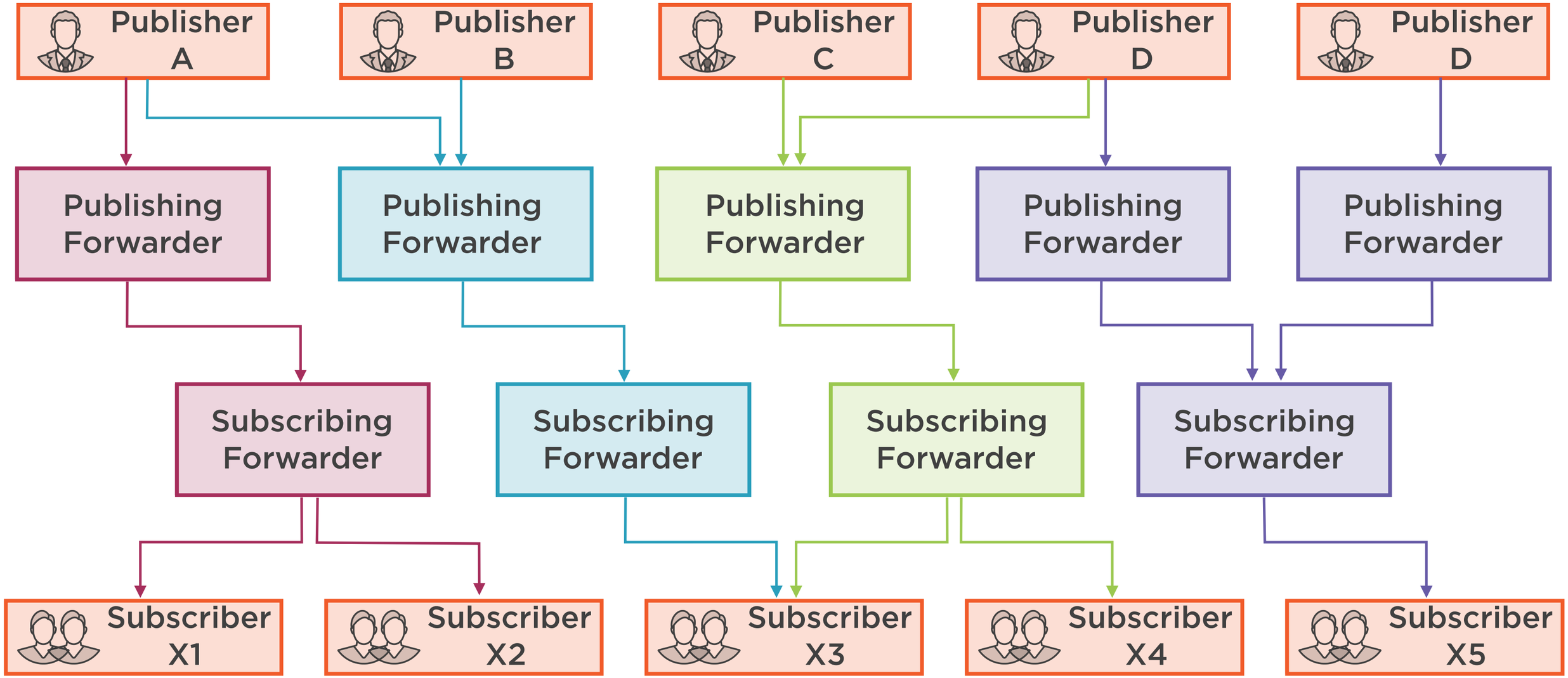
Pub/Sub waits for at least one subscriber on each subscription to acknowledge receipt

Deletes message from storage

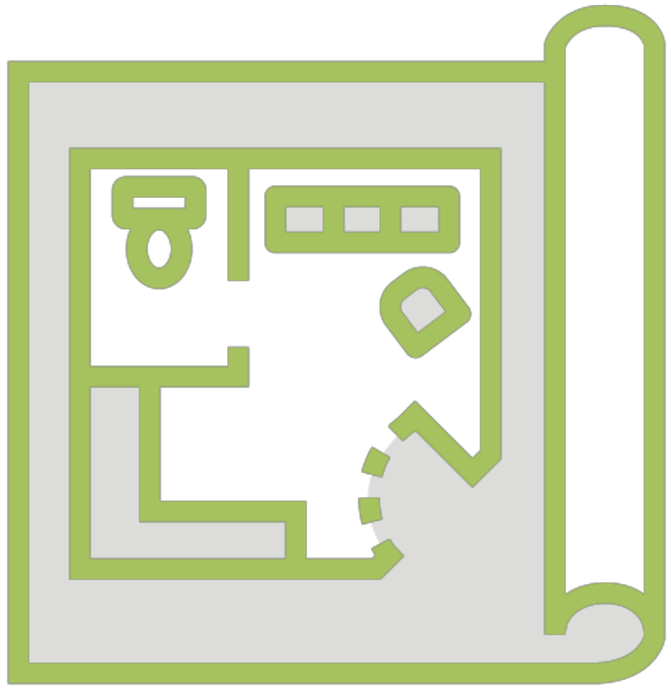
Pub/Sub Messaging



Pub/Sub Messaging



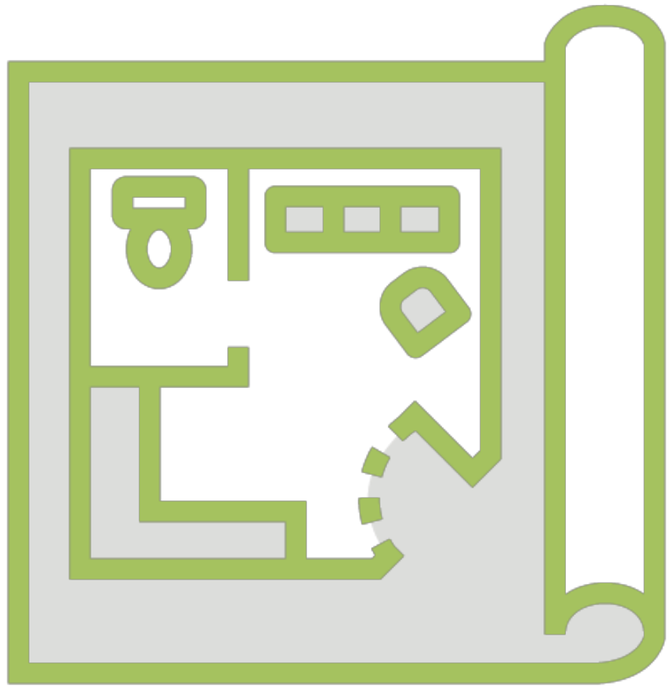
Architecture



Data plane

Control plane

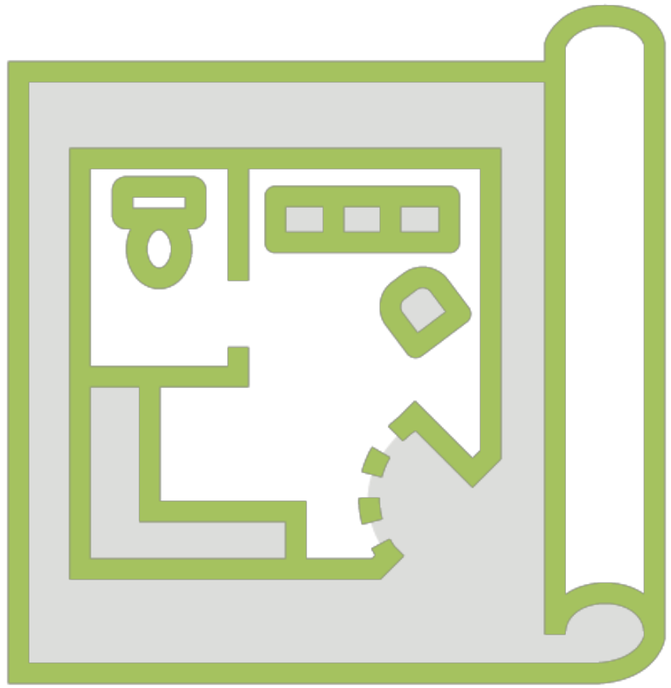
Data Plane



Data plane moves messages between publishers and subscribers

- Servers in data plane are called forwarders
- Publishing and subscribing forwarders

Control Plane



Control plane assigns publishers and subscribers to servers on the data plane

- Servers in control plane are called routers

Replaying Messages

Seek

Extends subscriber functionality by allowing you to to alter the acknowledgement state of messages in bulk.

Seek to a Timestamp

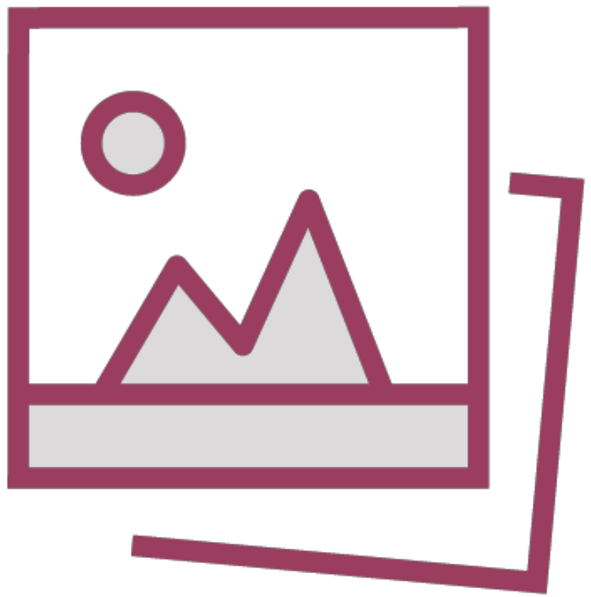
Marks every message received before the timestamp as acknowledged

Marks every message received after the timestamp as unacknowledged

To seek to timestamp in past, must enable retain acknowledged messages

To discard messages seek to a time in the future

Seek to a Snapshot

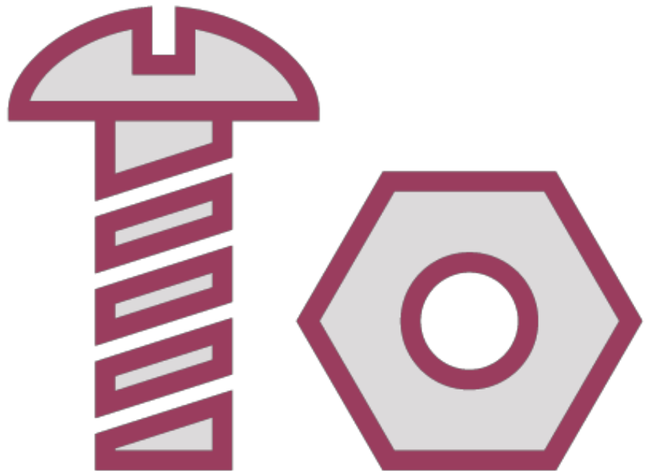


Captures the message acknowledgement state of a subscription

Retains all unacknowledged messages in the subscription at the time of snapshot

Also retains all messages received after the snapshot was created

Use Cases



- Update subscriber code safely
- Recover from unexpected subscriber problems
- Bulk acknowledge unneeded messages to avoid unnecessary processing
- Testing subscribers on known data

Pricing

Pricing

Message ingestion

Delivery

Seek-related storage

- Snapshots
- Retained acknowledged messages

https://cloud.google.com/pubsub/pricing

Pricing Schedule

Monthly data volume	Price
First 10GB	Free
Next 50TiB	$60/TiB
Next 100TiB	$50/TiB
Beyond 150TiB	$40/TiB

1024 KB messages + 2 subscriptions at rate of 1 MiB/second

Pricing Scenario

Message ingestion and delivery

Ingesting 1 MiB/second + Delivery 2MiB/second = 3MiB/second

Per-second Usage

Message ingestion and delivery

3 MiB/s x 3600 x 24 x 30 = 7.416 TiB/Month

Per-month Usage

Message ingestion and delivery

7.416 TiB - 10 GB (free tier) = 7.406 TiB

Free Tier

Deduct from data usage

7.406 TiB * $60/TiB = $444.36/month

Monthly Pub/Sub Bill

Overall costs of operation

Fine Print

Minimum billable volume per request is 1 KB

Cross-project billing

- Bills payable by project hosting resource
- Subscriptions, topics

Seek-related Message Storage

A subscription can be configured to retain acknowledged messages

Makes it possible to re-process messages using seek

Seek-related Message Storage

A snapshot of a subscription is created

Storage fees are charged for storing all snapshot's unacknowledged messages

Message storage fees, at a rate of $0.27/GiB-month

Steady 1 MiB/second incoming data with negligible backlog and 7 day retention

Pricing Scenario: Retained Acknowledgements

1MiB/second x 3600 seconds/hour x 24 hours/day = 86.4 GiB/day

Pricing Scenario

Daily data usage

86.4 GiB/day * 7 days = 605GiB

Pricing Scenario

Weekly data usage

605 GiB-month * $0.27/GiB-month = $163

Retained Acknowledged Messages

Total monthly bill incurred for retained messages

Snapshot Fees

When snapshot is created, one-time fee applied to backlog of unacknowledged messages

Fee equivalent to storing that backlog for the full seven days is charged

In addition to incremental charge for new messages

60 MiB published in first minute stored for 7 days - 1 minute

60 MiB published in second minute stored for 7 days - 2 minutes

...

Over 7 days, backlog size reaches 605GiB

Pricing Scenario: Snapshots

Backlog size reaches 605 GiB when snapshot expires after 7 days

Average size of backlog at any point in time is 1/2 x 605 GiB

1/2 x 605 GiB x 7 days = 2118 GiB-days

Pricing Scenario: Snapshots

Weekly backlog size calculation

2118 GiB-days x (1/30 months/day) x $0.27/GiB-month = $19

Snapshot Message Storage Fees

Snapshots potentially cheaper than acknowledged message retention

Fees for Unacknowledged Messages

When snapshot is created, one-time fee applied to backlog of unacknowledged messages

Fee equivalent to storing that backlog for the full seven days is charged

In addition to incremental charge for new messages

10 GiB unacknowledged message backlog

Snapshot message storage fees

Assumption for pricing scenario

10 GiB x 7 days / 30 days/month x 0.27/GiB-month = $0.63

One-time Fee for Unacknowledged Backlog

Total monthly fee

Summary

Messaging middleware

Decouples senders and receivers

Global, replicated, autoscaling

Publishers publish to topics

Subscribers listen on subscriptions

Complex pricing