

Apache Pulsar and its enterprise use cases

Yahoo Japan Corporation
Nozomi Kurihara

Sep., 27th, 2018

Who am I?

Nozomi Kurihara

- Software engineer at Yahoo! JAPAN (April 2012 ~)
- Working on internal messaging platform using Apache Pulsar
- Committer of Apache Pulsar



Agenda

- 1. What is Apache Pulsar?**
- 2. Why is Apache Pulsar useful?**
- 3. How does Yahoo! JAPAN uses Apache Pulsar?**

What is Apache Pulsar?

Agenda

1. What is Apache Pulsar?

- › History & Users
- › Pub-Sub messaging
- › Architecture
- › Client libraries
- › Topic
- › Subscription
- › Sample codes

2. Why is Apache Pulsar useful?

3. How does Yahoo! JAPAN uses Apache Pulsar?

Apache Pulsar

Flexible pub-sub system backed by durable log storage

History:

- › 2014 Development started at Yahoo! Inc.
- › 2015 Available in production in Yahoo! Inc.
- › Sep. 2016 Open-sourced (Apache License 2.0)
- › June 2017 Moved to Apache Incubator Project

Competitors:

- › Apache Kafka
 - › RabbitMQ
 - › Apache ActiveMQ
 - › Apache RocketMQ
- etc.

Users:

- › Oath Inc. (Yahoo! Inc.)
 - › Comcast
 - › The Weather Channel
 - › Mercado Libre
 - › Streamlio
 - › Yahoo! JAPAN
- etc.



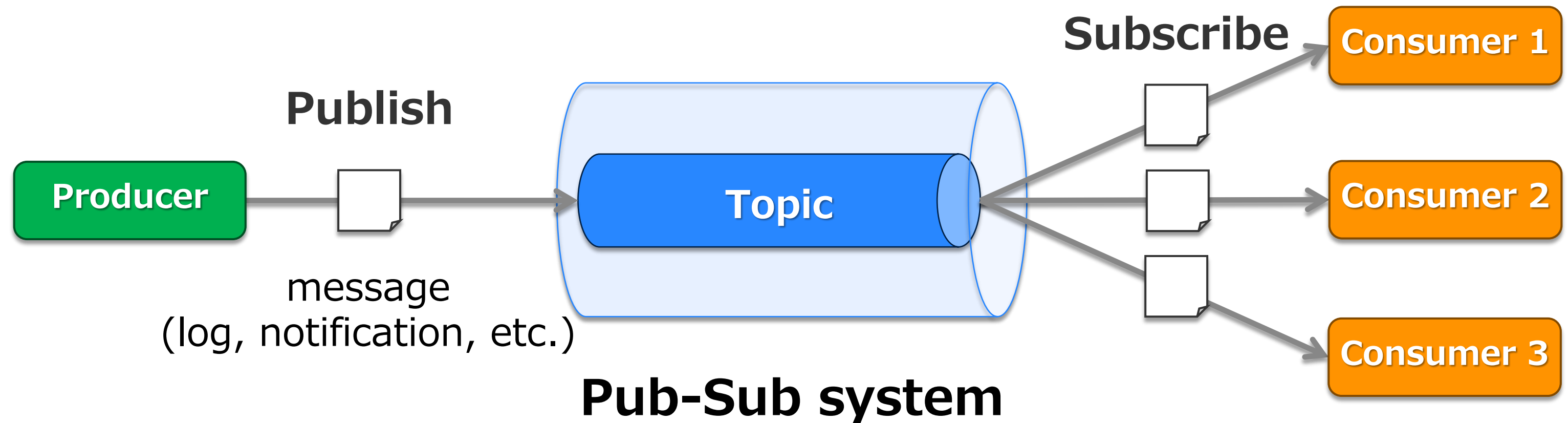
Pub-Sub messaging

Message transmission from one system to another via **Topic**

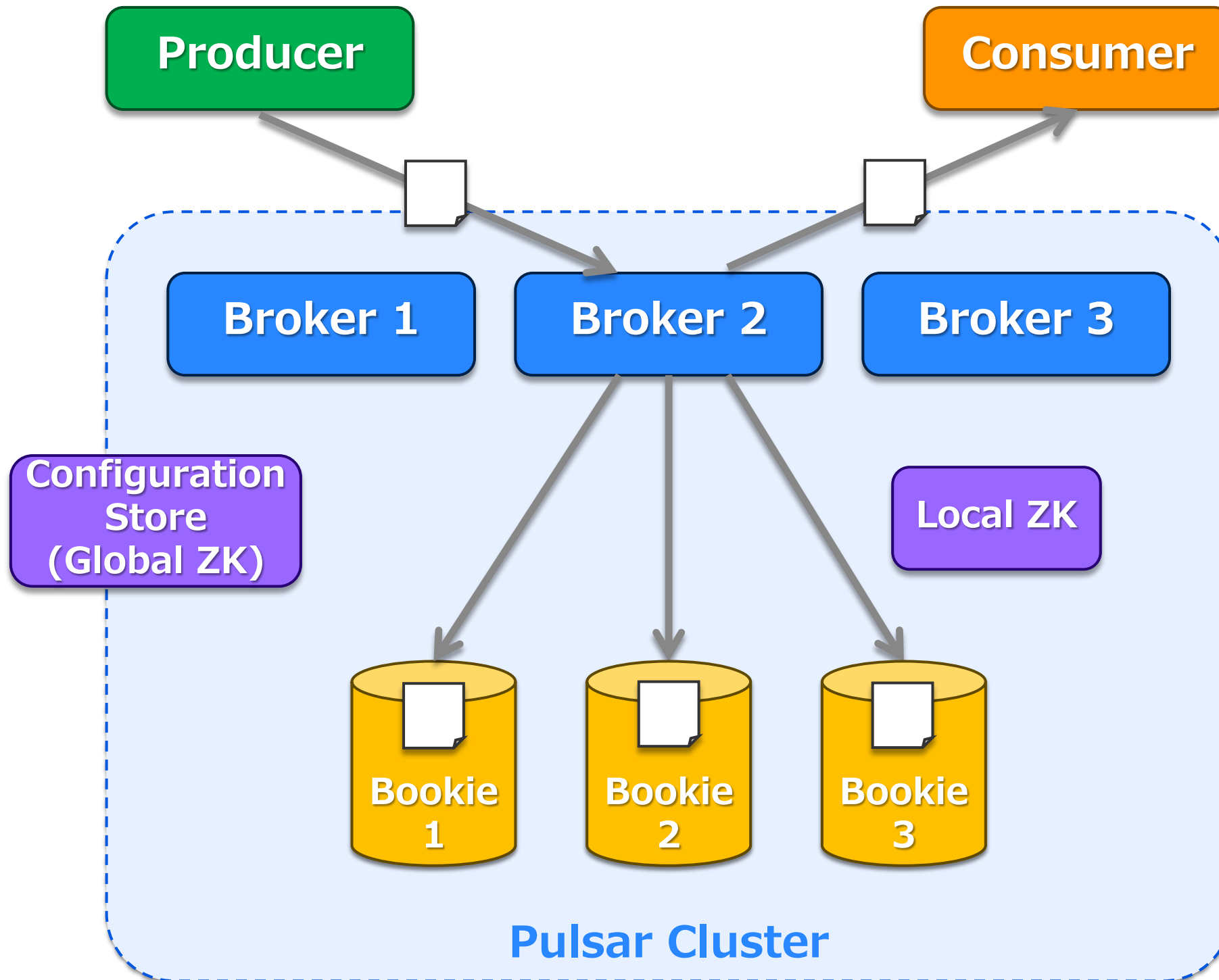
Producers publish messages to Topics

Consumers receive only messages from Topics to which they subscribe

Decoupled (no need to know each other) → asynchronous, scalable, resilient



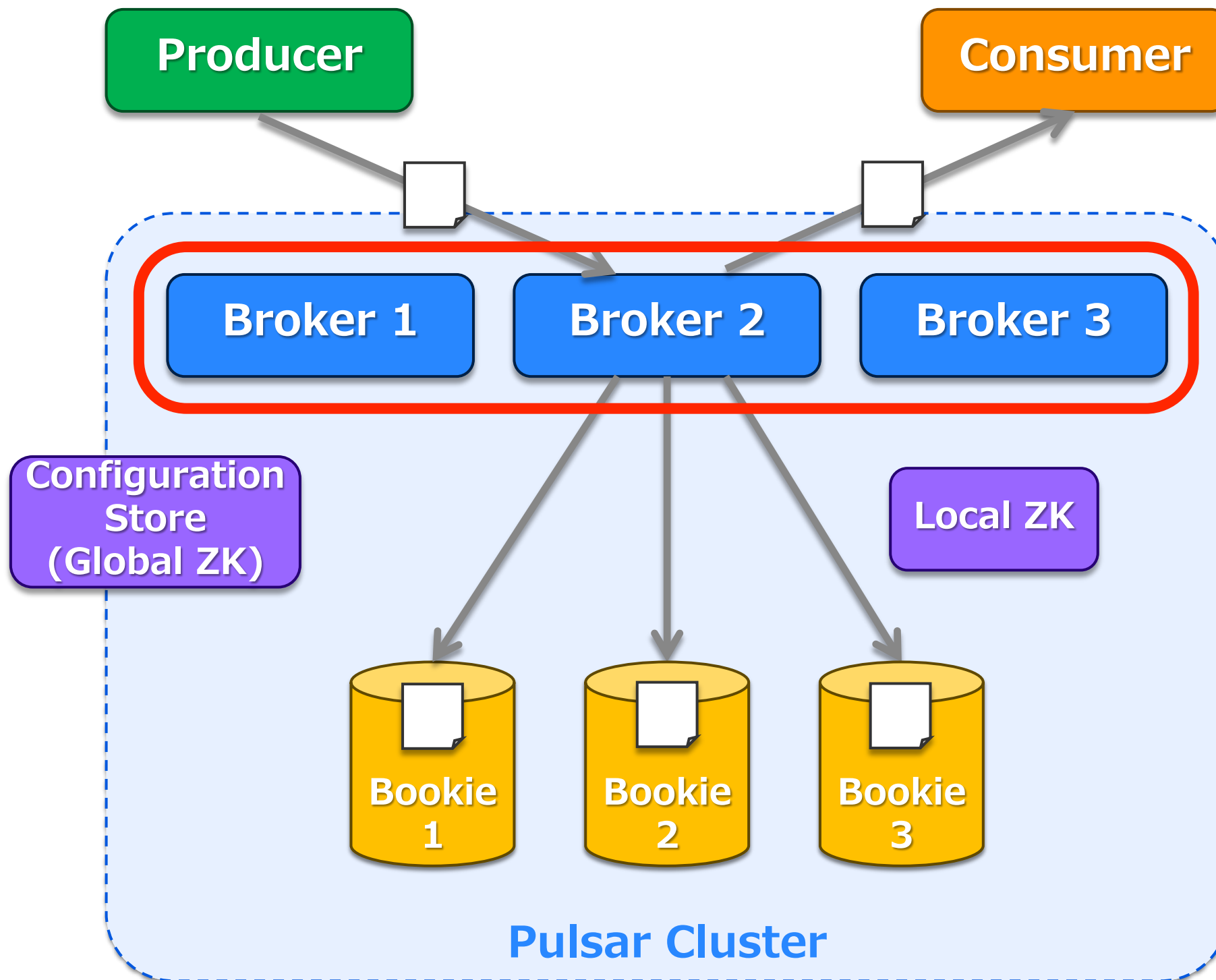
Architecture



■ 3 components:

- Broker
- Bookie
- ZooKeeper

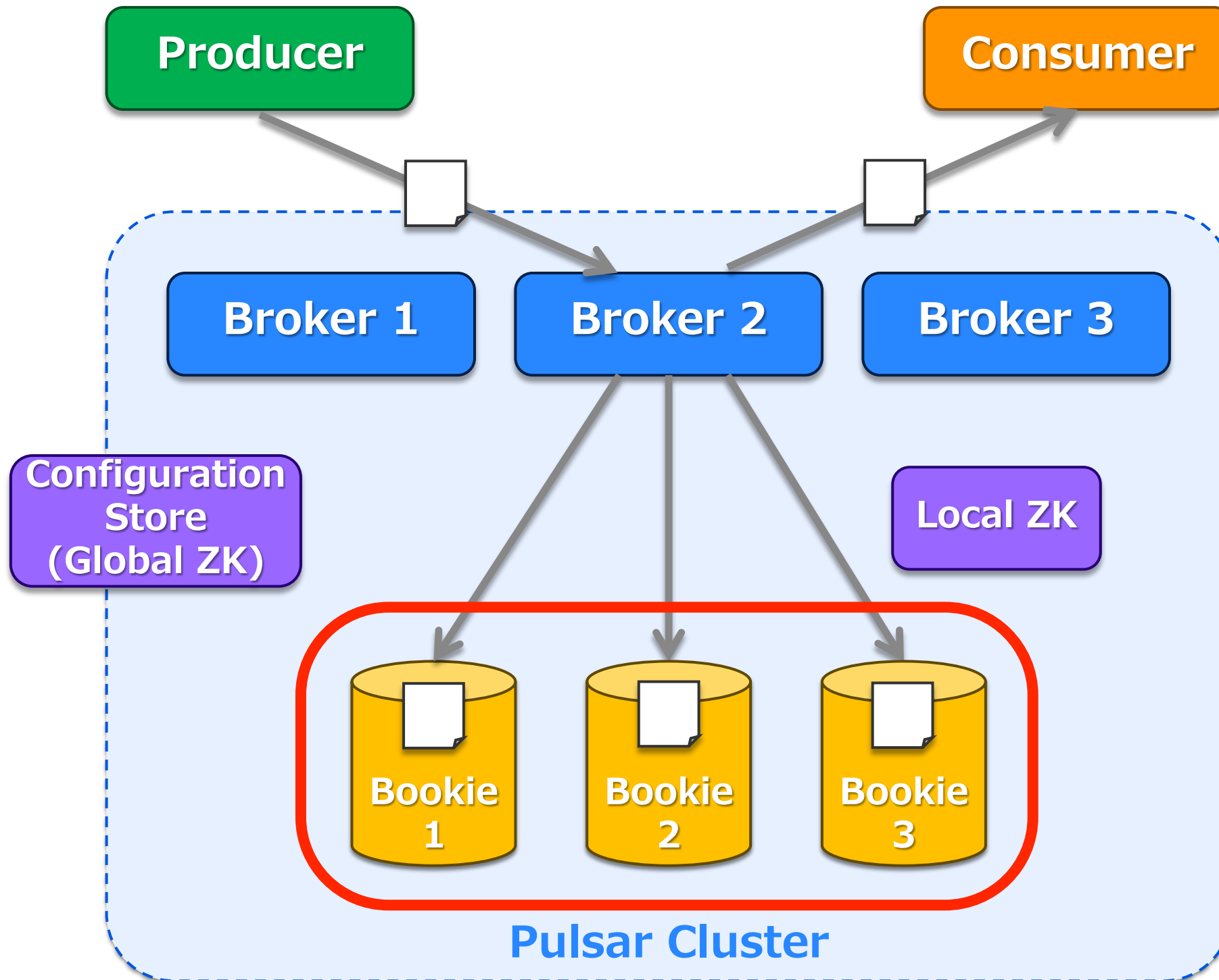
Architecture - Broker



■ Broker

- Serving node for clients' requests
- No data locality (**stateless**)

Architecture - Bookie



■ Bookie (Apache BookKeeper)

- Storage node for messages
- Durable, Scalable, Consistent, Fault-tolerant, Low-latency

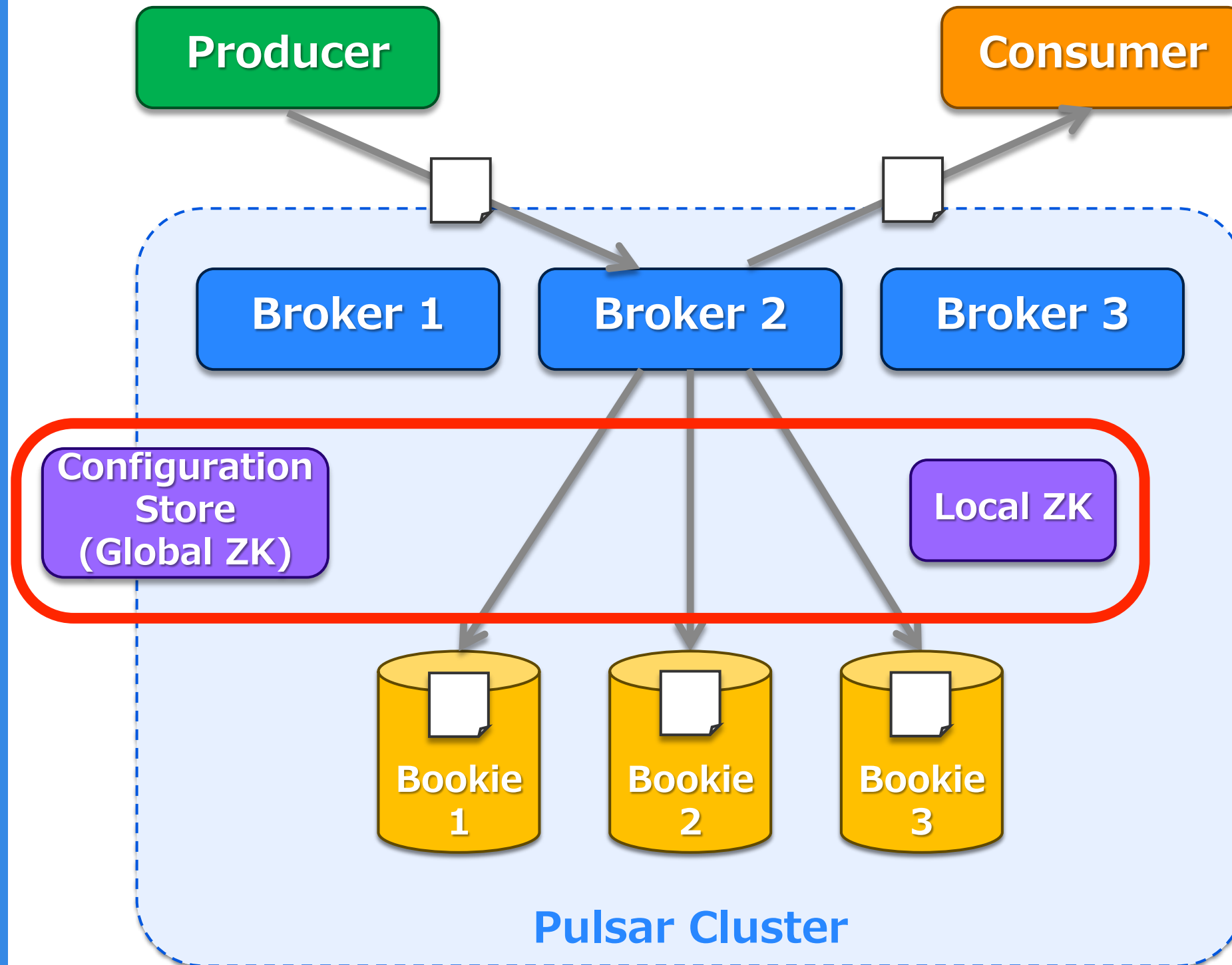


Apache BookKeeper:

distributed write-ahead log system

Copyright © 2016 - 2018 [The Apache Software Foundation](#),
licensed under the [Apache License, version 2.0](#).

Architecture - ZooKeeper



■ Apache ZooKeeper

- ▶ Store metadata and configuration
- ▶ **Local ZK**: within local cluster
- ▶ **Configuration Store**: across all clusters



Copyright © 2016 - 2018 [The Apache Software Foundation](#),
licensed under the [Apache License, version 2.0](#).

Client libraries

Supported client libraries:

Java

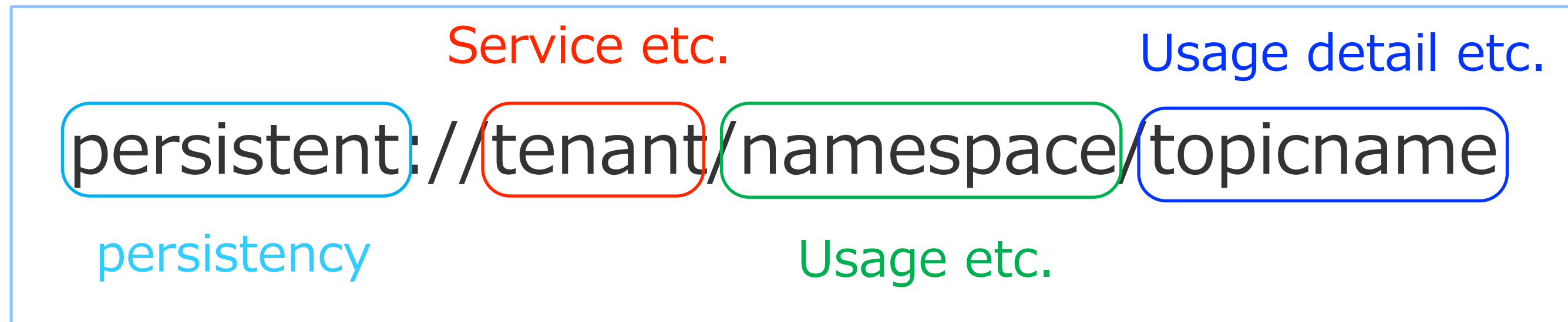
C++

Python

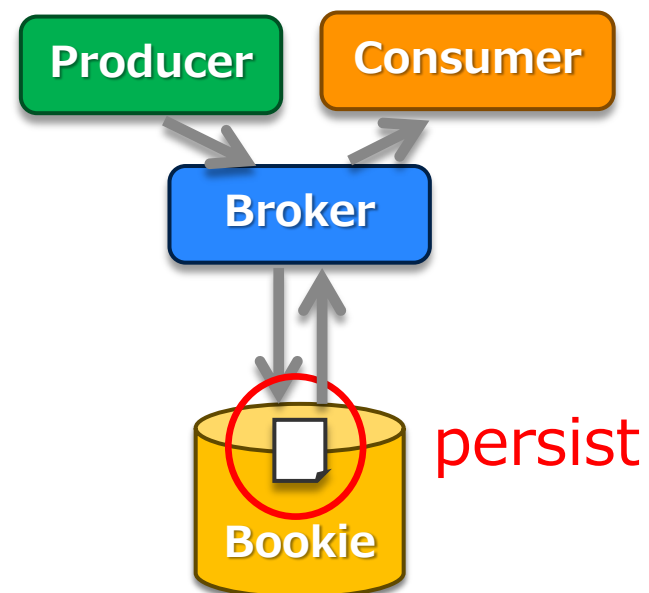
Go

Also can use **WebSocket API**

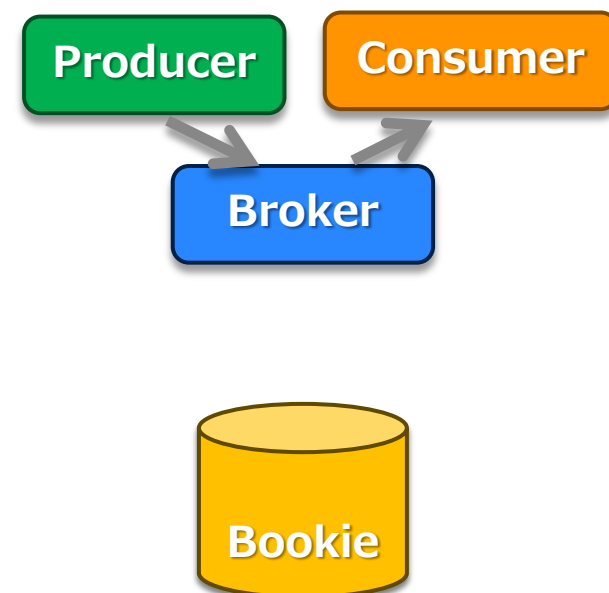
Topic URI



persistent



non-persistent



Examples:

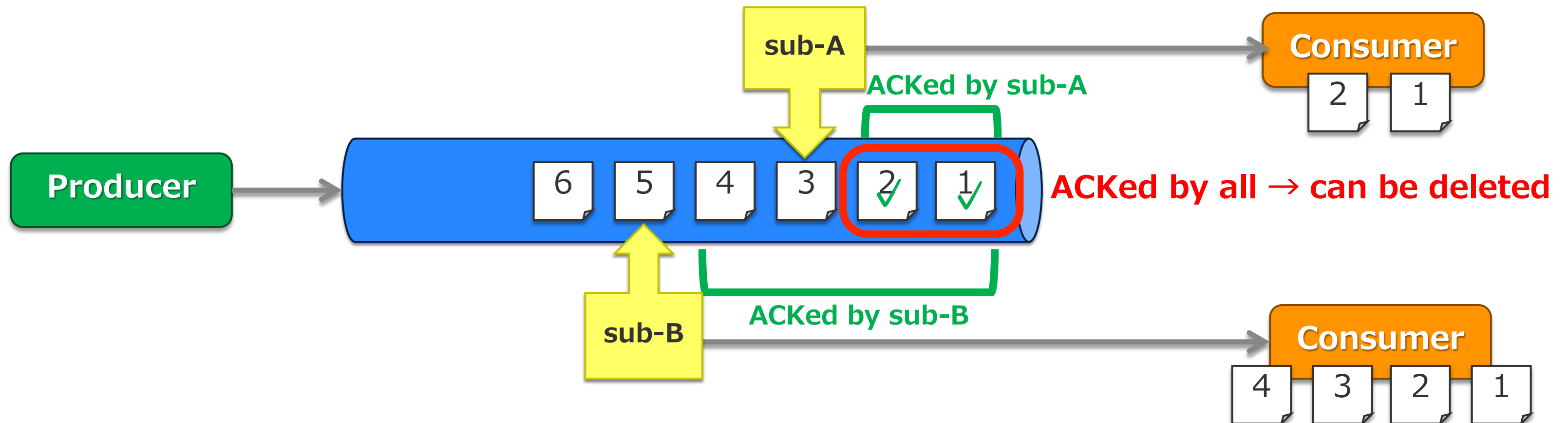
`persistent://shopping/user-log/buy`
`persistent://auction/notifications/bid`
`persistent://news/sports/articles`

...

Subscription

- **Acknowledgement(ACK)**: confirmation that message is received
- **Cursor**: which messages are acknowledged
- **Subscription**: cursor and which consumers are connecting

Note: Each message is kept unless all subscriptions acknowledge



Sample code(Java) - Producer

```
// Create client by specifying the broker URI
PulsarClient client = PulsarClient.builder()
    .serviceUrl("pulsar://localhost:6650")
    .build();

// Create producer by specifying the topic URI
Producer<byte[]> producer = client.newProducer()
    .topic("persistent://my-tenant/my-ns/my-topic")
    .create();

// Send a message
producer.send("My message".getBytes());
```

Sample code(Java) - Consumer

```
// Create client by specifying the broker URI
PulsarClient client = PulsarClient.builder()
    .serviceUrl("pulsar://localhost:6650")
    .build();

// Create consumer by specifying the topic URI and subscription name
Consumer consumer = client.newConsumer()
    .topic("persistent://my-tenant/my-ns/my-topic")
    .subscriptionName("my-subscription")
    .subscribe();

// Receive a message
Message msg = consumer.receive();
System.out.printf("Message received: %s", new String(msg.getData()));

// Acknowledge the message so that it can be deleted by the broker
consumer.acknowledge(msg);
```


Why is Apache Pulsar useful?

Agenda

1. What is Apache Pulsar?

2. Why is Apache Pulsar useful?

- › High performance
- › Scalability
- › Multi-tenant
- › Geo-replication
- › Pulsar Functions
- › Pulsar on Kubernetes
- › Apache Pulsar vs. Apache Kafka

3. How does Yahoo! JAPAN uses Apache Pulsar?

High performance

High throughput & low latency even if

huge number of topics

high reliability is required

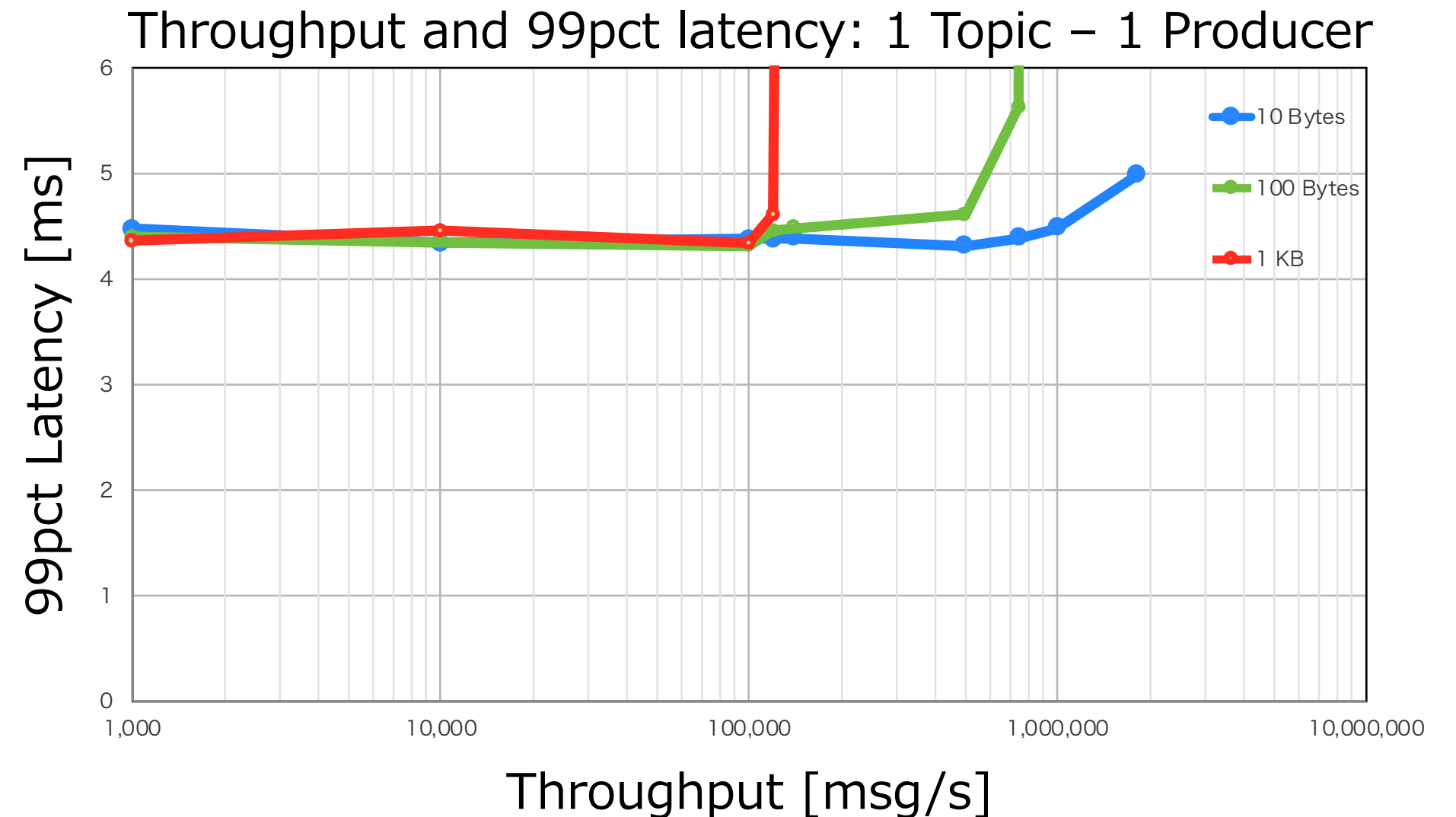
ex. At Oath Inc.

Requirements:

- Topics: **2.3 million**
- Messages : **100 billion [msg/day]**
- Message loss is not acceptable
- Order needs to be guaranteed

Performance(1KB message):

- Throughput: **100 thousand [msg/s]**
- Latency: **5 [ms]**



Graph is from:

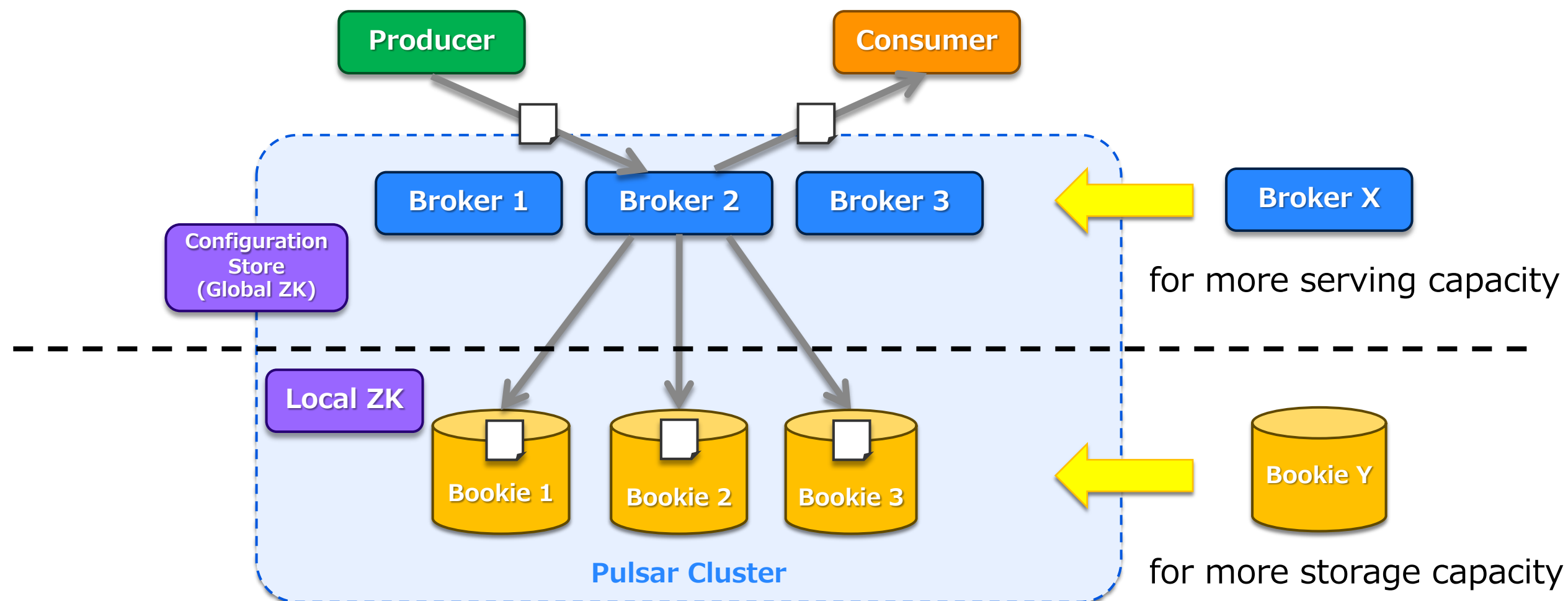
<https://yahooeng.tumblr.com/post/150078336821/open-sourcing-pulsar-pub-sub-messaging-at-scale>

Scalability

Just adding Brokers/Bookies increases serving/storage capacity

Add Brokers when more producer/consumer's requests need to be served

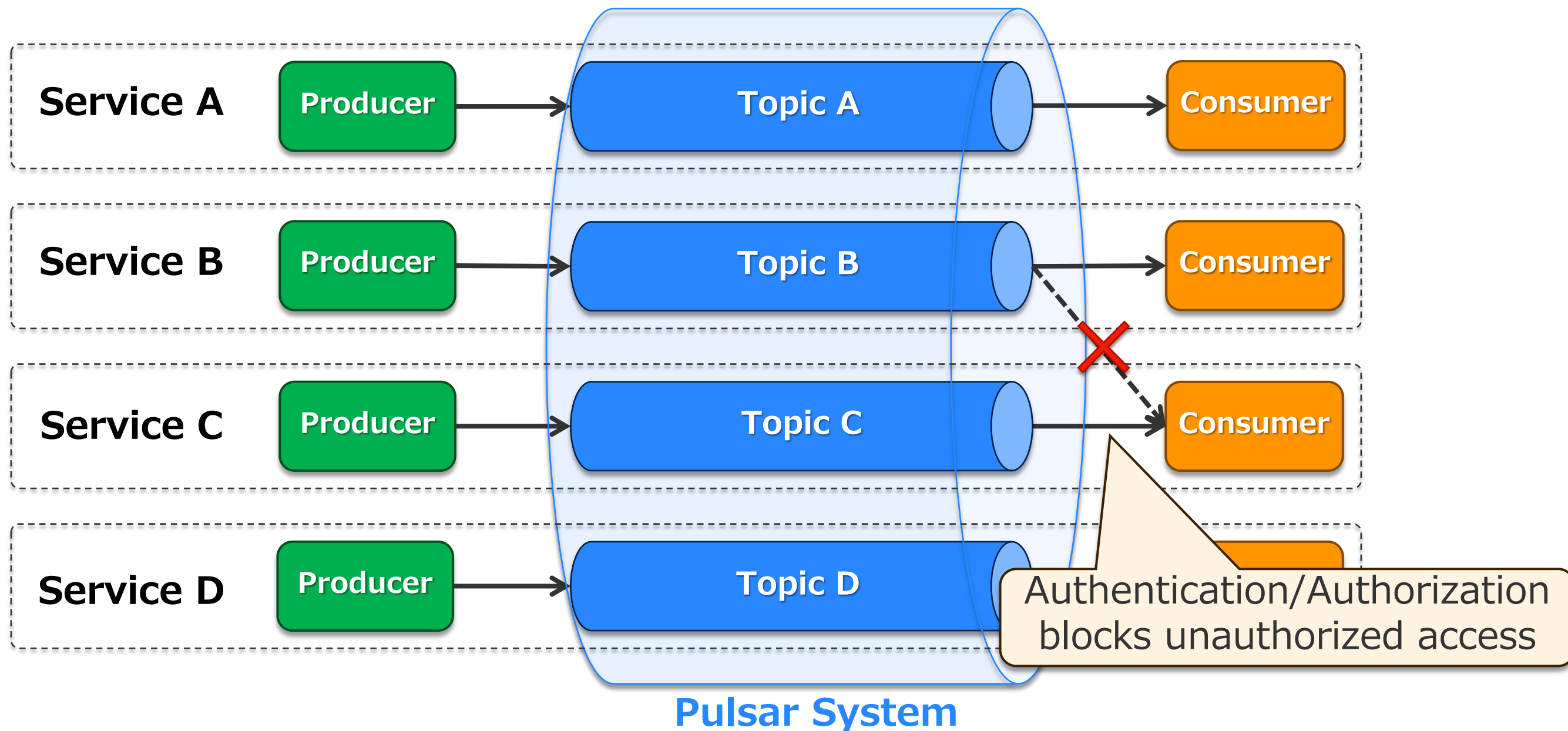
Add Bookies when more messages need to be stored



Multi-tenancy

Multiple services can share one Pulsar system

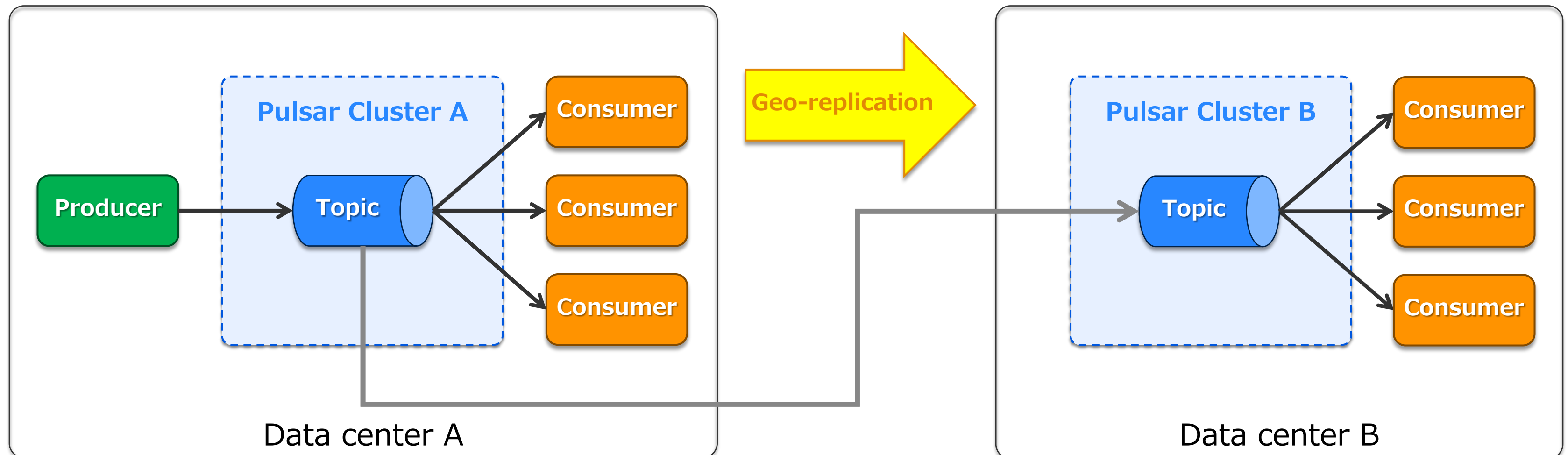
Just use Pulsar as a "Tenant" → no need to maintain own messaging system
Authentication/Authorization mechanism protects messages from interception



Geo-replication

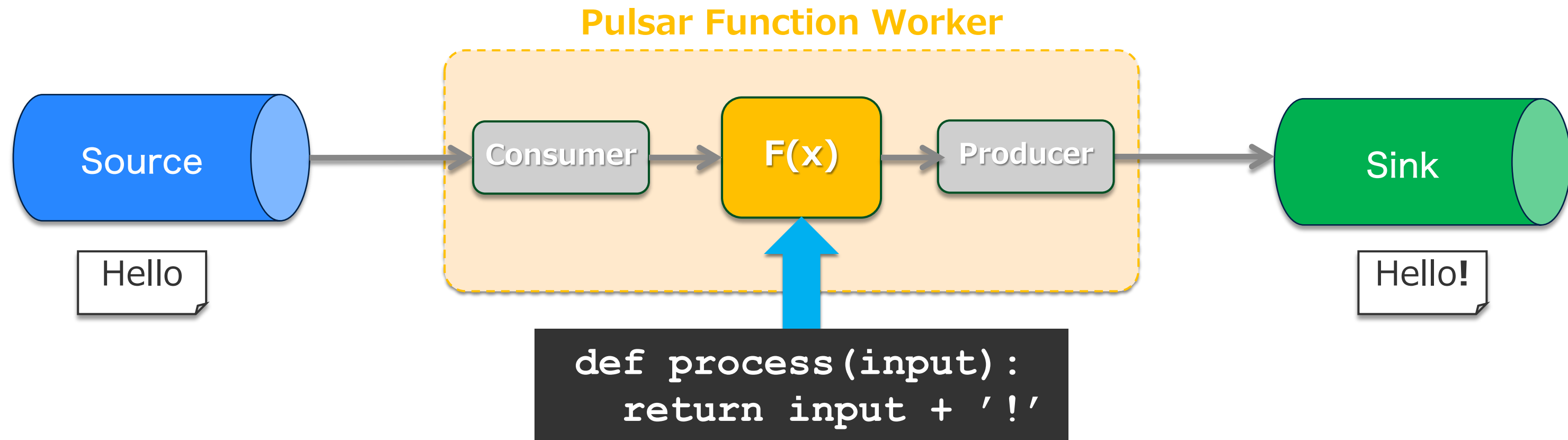
Pulsar can replicate messages to another cluster

1. Producers only have to publish messages to Pulsar in the same data center
2. Pulsar asynchronously replicates messages to another cluster
3. Consumers can receive messages from the same data center



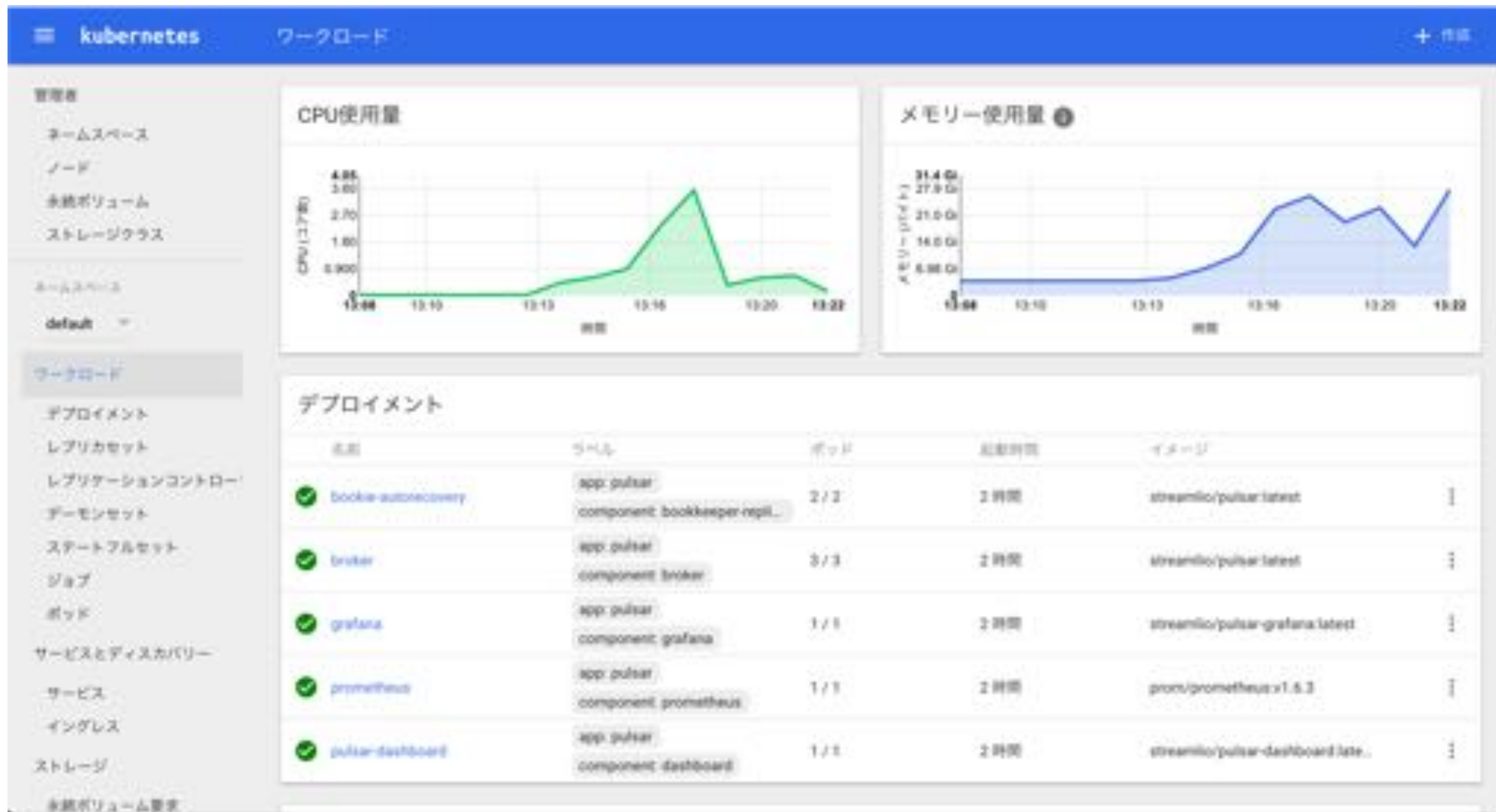
Pulsar Functions

- **Lightweight compute framework** (e.g., AWS lambda, Google Cloud Functions)
- **No need to launch extra systems** (e.g., Apache Heron, Apache Storm, Apache Spark)
- All you have to do is to implement your logic and deploy it to Pulsar cluster
- Java and Python are available and also other languages will be supported in the future



Pulsar on Kubernetes

- Pulsar can be easily deployed in Kubernetes clusters (e.g., AWS, GKE)
- See <https://pulsar.incubator.apache.org/docs/latest/deployment/Kubernetes/>



Apache Pulsar vs. Apache Kafka

	Apache Pulsar	Apache Kafka
Architecture image: <ul style="list-style-type: none"> • # of partition=1 • # of storage=3 • # of replication=2 		
Replication unit	Segment (smaller than Partition)	Partition
Data distribution	Distributed and balanced across all Bookies	Kept in only leader and replica Brokers
Max capacity	Not limited by any single node	Limited by the smallest broker's capacity
Data Rebalancing when scale	Not required	Required
Geo-replication	Built-in	Need to start an extra process: MirrorMaker

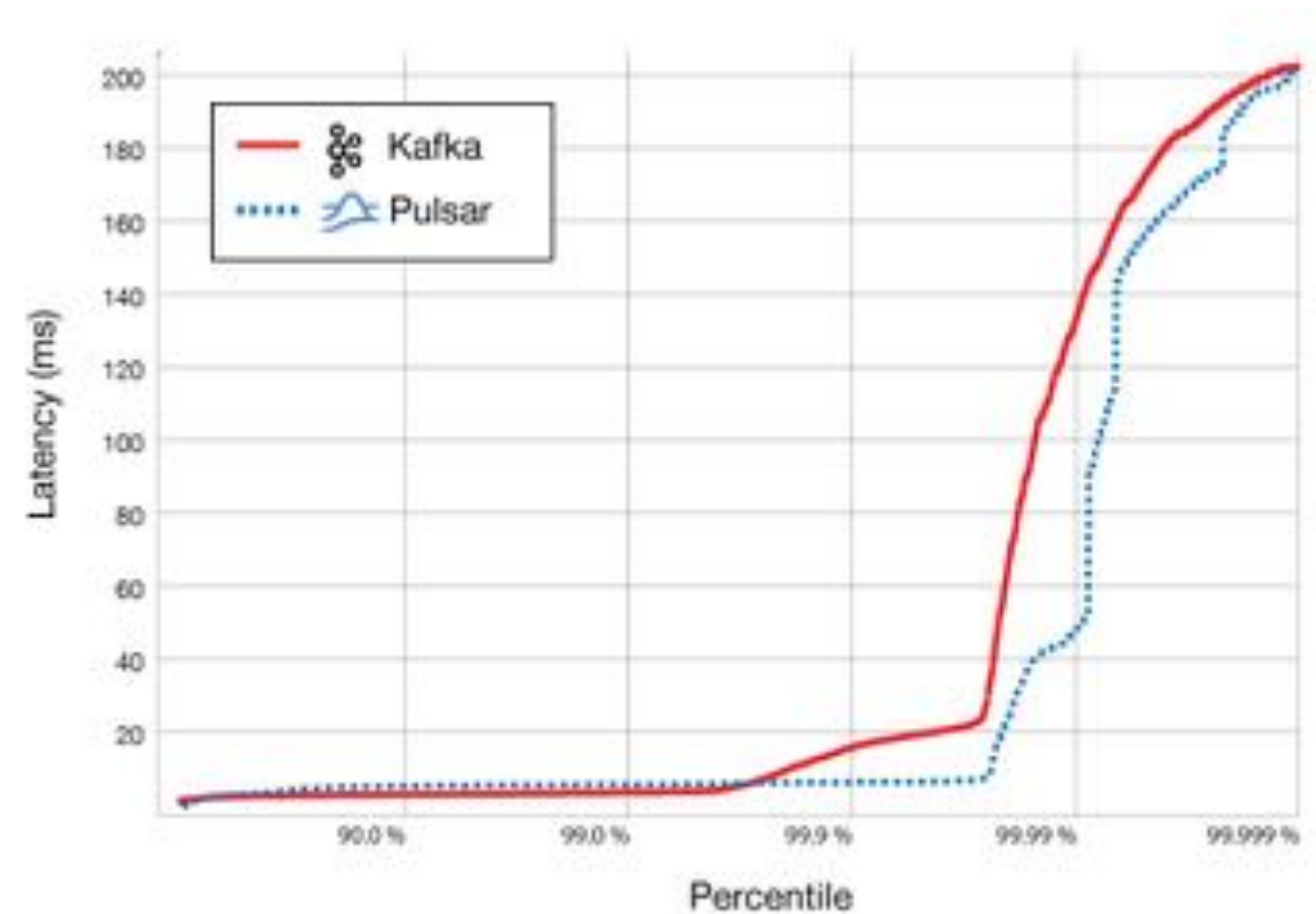
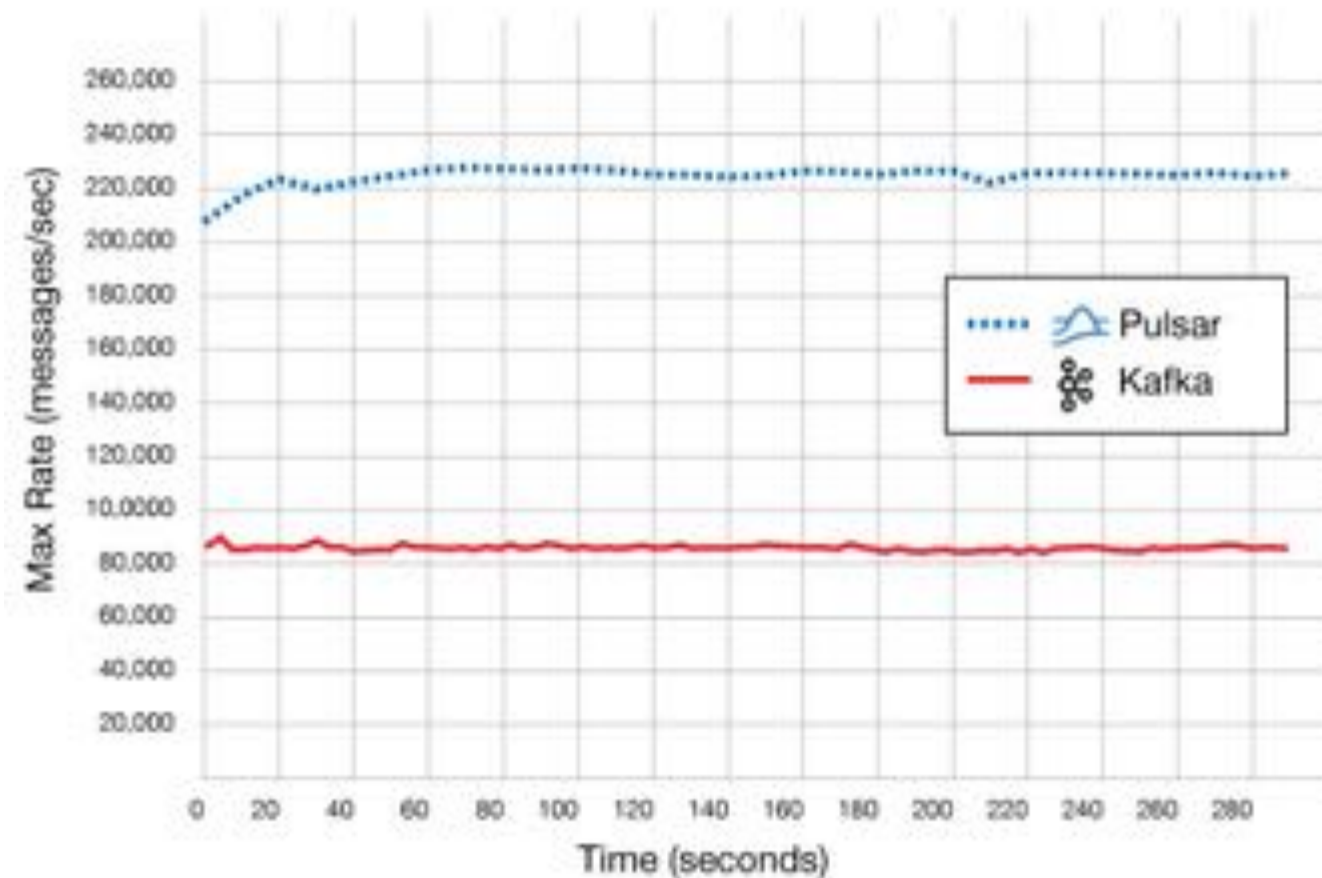
OpenMessaging Benchmark

<http://openmessaging.cloud/docs/benchmarks/>

- A suite of tools that can easily compare various messaging systems

<https://info.streaml.io/benchmarking-streaming-messaging-platforms>

- Pulsar showed better performance than Kafka



How does Yahoo! JAPAN use Apache Pulsar?

Agenda

1. What is Apache Pulsar?

2. Why is Apache Pulsar useful?

3. How does Yahoo! JAPAN uses Apache Pulsar?

- › About Yahoo! JAPAN
- › System architecture
- › Self-service tool
- › Case 1 - Notification
- › Case 2 - Job queuing
- › Case 3 - Log collection

Yahoo! JAPAN

🏠 ホームページに設定する

ヘルプ

Yahoo! BB
きっず版
アプリ版



トラベル



ヤフオク!



ショッピング



プレミアム



カード



メール

ウェブ 画像 動画 辞書 知恵袋 地図 リアルタイム 一覧

検索

▶ 奈良への旅がおトク、宿泊料金が最大50%オフ ▶ サプリメントや健康食品のセールを開催中 ▶ 首都圏のレストランがネットで予約可能

主なサービス

- ショッピング
- ヤフオク!
- LOHACO
- 旅行、ホテル予約
- 一休.com
- 一休.comレストラン
- ニュース
- 天気
- スポーツナビ
- ファイナンス
- テレビ
- GYAO!
- ゲーム
- Yahoo!モバゲー
- 地図
- 路線
- 食べログ
- 求人、アルバイト
- 不動産
- 自動車
- 掲示板
- ブログ
- 美容、ダイエット
- 恋愛、婚活
- お気に入り
- 電子書籍
- 映画
- 占い
- 地域

動画 **LIVE** ニュース 経済 エンタメ スポーツ

9時49分更新

- 敷金やツケ 民法どう変わる? **NEW**
- 野良猫に餌やり禁止 賛否続々 **NEW**
- トルコ最大級の地下都市発見 **NEW**
- 皆既月食 夜桜と競演珍しく
- 奇策に選手も驚き 日本代表
- ママタレ 割れる子供の顔出し
- 運転見合わせの路線、再開へ
- 優先席付近 電源オフは必要か



カレーの街、神保町
9月7日9時46分配信
curry topics

もっと見る トピックス一覧

話題なう 新スイーツが | 猫密度 国で違い?

今、話題の「男前インテリア」



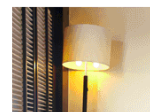
まず押さえない
定番インテリア



深みがあるヴィ
ンテージ



多肉植物でカ
フェっぽく



失敗しない間接
照明のコツ

あなたにおすすめ



思わず欲しくなっ
ちゃう、おもしろ
文房具



Yahoo!ショッピング

一番売れているラ
ーメンは果たして
どれ?



おしゃれなティー
ポットで優雅なテ
ィータイム



見て楽しい、大人
女子向け文房具セ
レクション

ネットで話題の無料動画

▶ ライフプラン別、老後資金の正解例

Yahoo! JAPAN アプリ

タブレット版
iPad / Android

無料

明日の雨をお知らせ!

アプリはこちらから

ログイン

IDでもっと便利に[新規取得]

ログイン履歴

登録情報

メールアドレス取得

ポイント確認

カード新規入会

2018年9月7日 (金)

今日の天気 (東京)

10% | 19°C / 10°C

明日の天気 (東京)

10% | 21°C / 10°C

雨雲レーダー

新宿区

運行情報

事故・遅延情報はありません (16時05分)

カレンダー

今日の運勢 牡羊座 95点

スコアボード

プロ野球

Jリーグ

9月7日 (金) の試合はありません J1試合予定

Jリーグトップ

スポーツナビ

<https://www.yahoo.co.jp/>

Yahoo! JAPAN – 3 numbers

100+



services

150,000+



servers
(real)

71,300,000,000



PV/month
(average in 2017)

Why Yahoo! JAPAN chose Pulsar?

Large number of users

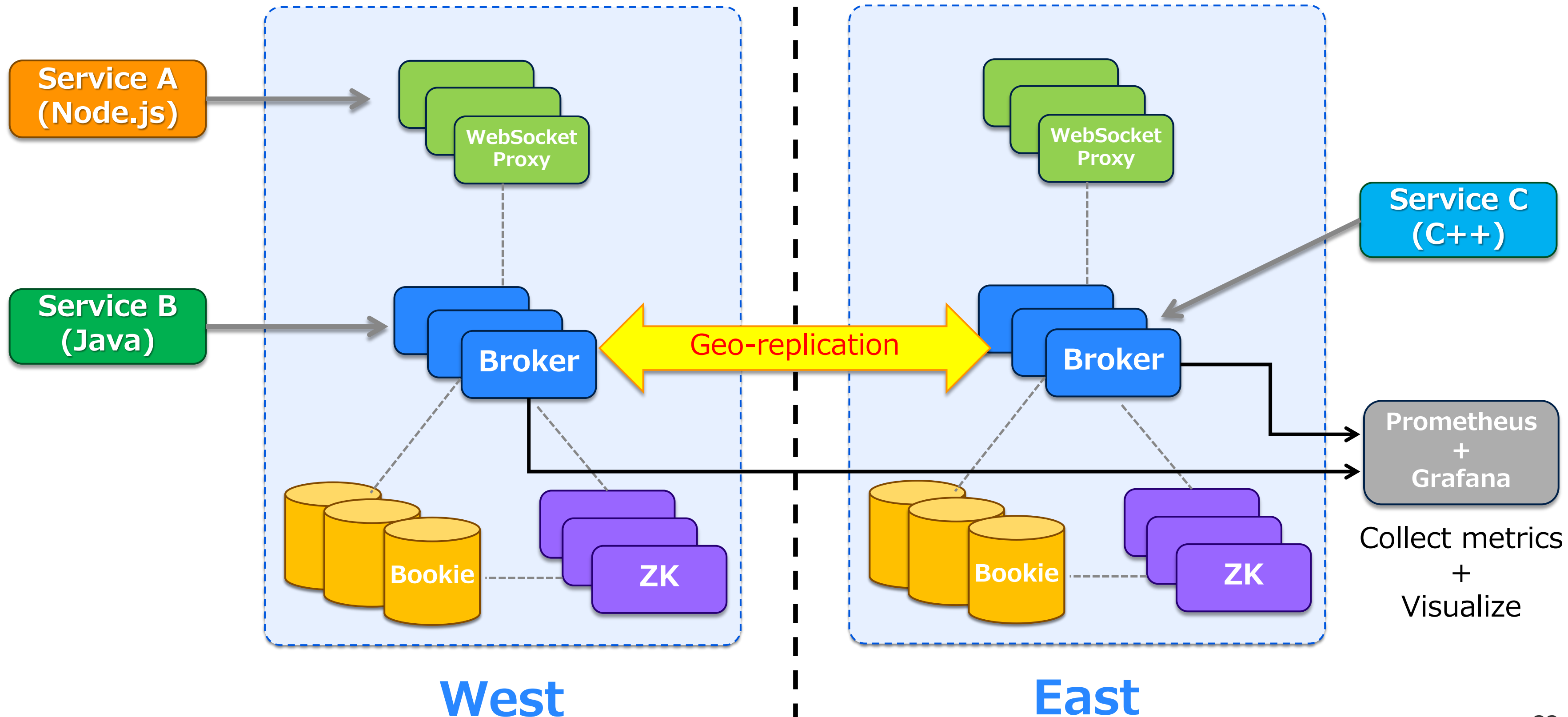
Large number of services

Sensitive/mission-critical messages

Multiple data centers

Pulsar meets all these requirements!

System architecture in Yahoo! JAPAN

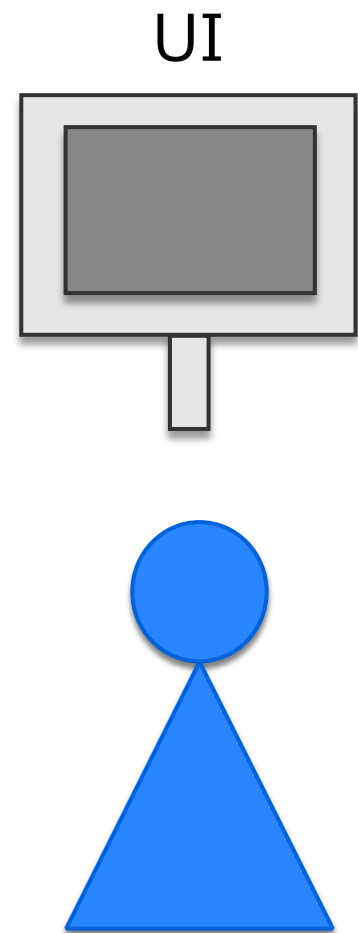


Monitoring by Prometheus



Self-service tool

Web UI to manage tenants, namespaces and topics (Yahoo! JAPAN internal)



プロパティ名 ⓘ
半角英数字・ハイフンで3-32文字

Create tenant

ネームスペース名 ⓘ
半角英数字・ハイフンで1-32文字

produce権限のロール ⓘ
ロールを入力
Athenz認証: \${テナントドメイン}.\${テナントサービス}をカンマ区切り

consume権限のロール ⓘ
ロールを入力
Athenz認証: \${テナントドメイン}.\${テナントサービス}をカンマ区切り

Create namespace

統計情報 ⓘ

Average msg size	44.22 bytes
Rate in	3.3333854350366 msg/s
Rate out	3.3333880293975 msg/s
Throughput in	147.40230393732 bytes/s
Throughput out	147.40241865996 bytes/s
Producer count	1
Storage size	3.3333854350366 bytes
Broker	http://...:4080

サブスクリプション一覧 ⓘ

pcf-sub [consumer情報を見る\(1\)](#)

Type	Backlog	Rate out	Throughput out	Rate expired	Unacked
Exclusive	0 msg/s	3.3333880293975 msg/s	147.40241865996 bytes/s	0 msg/s	210 msg/s

Consumer	Address	Rate out	Throughput out	Redeliver rate	Connected since	Available permits	Unacked	Blocked
55244	/172.21.19.7.64:59750	3.3333880293975 msg/s	147.40241865996 bytes/s	0 msg/s	2017-04-27 10:23:52.952+0900	790	210 msg	no

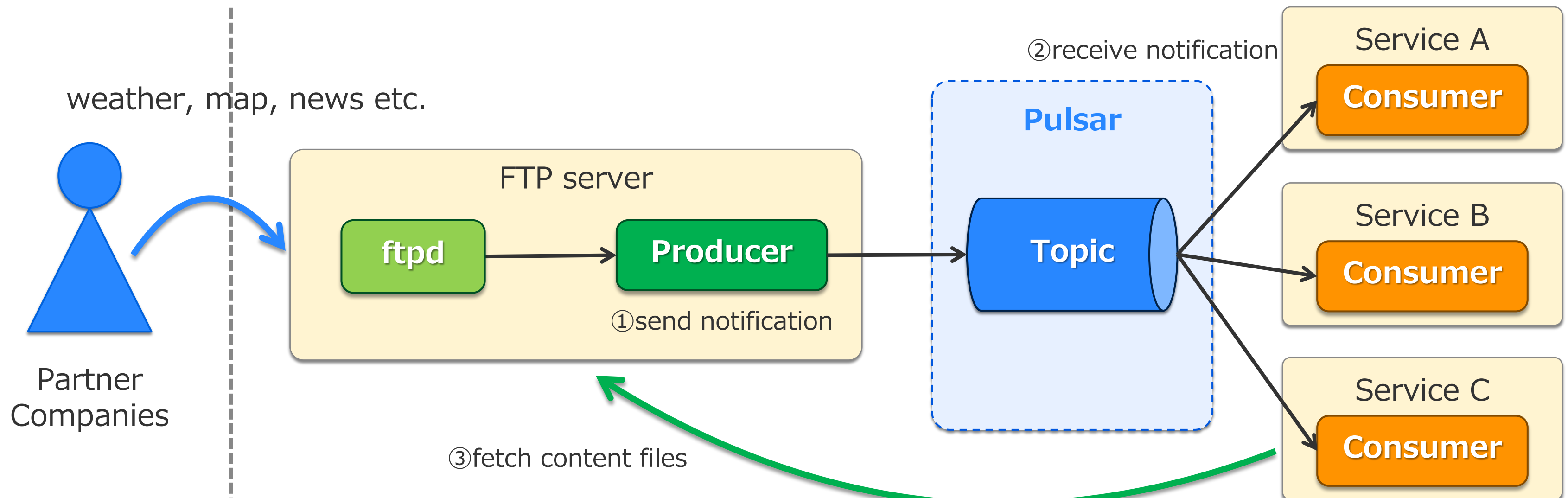
See topic stats

Case 1 – Notification of contents update

Various contents files are pushed from partner companies to Yahoo! JAPAN

Notification is sent to topic when contents are updated

Once services receive a notification, they then fetch contents from file server

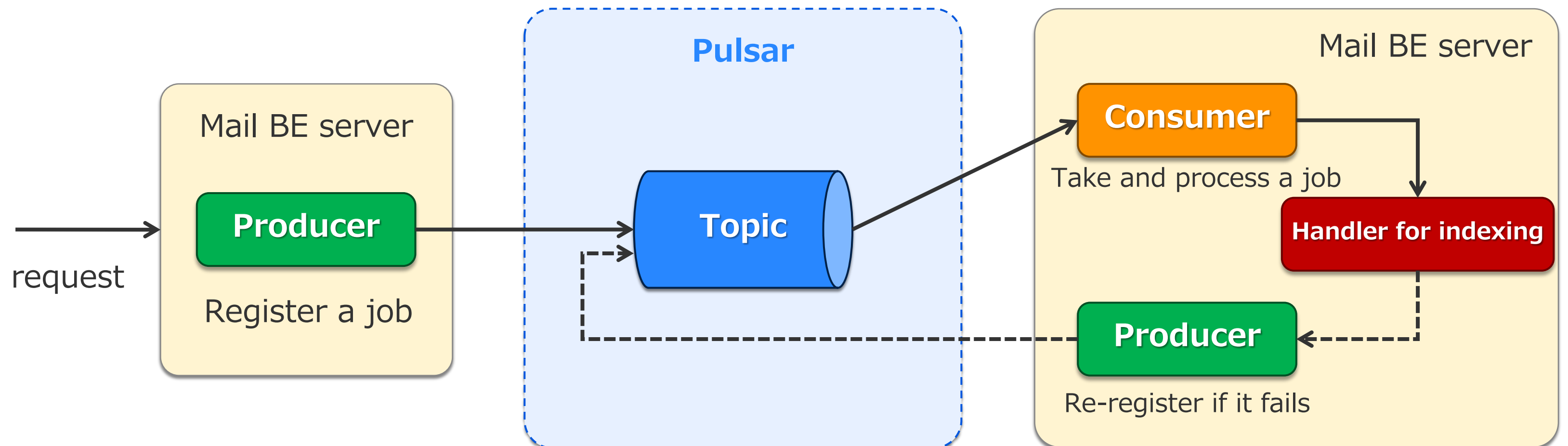


Case 2 – Job queuing in mail service

Indexing of mail can be heavy → let's execute it asynchronously

Producers register jobs to Pulsar

Consumers take jobs from Pulsar at their own pace

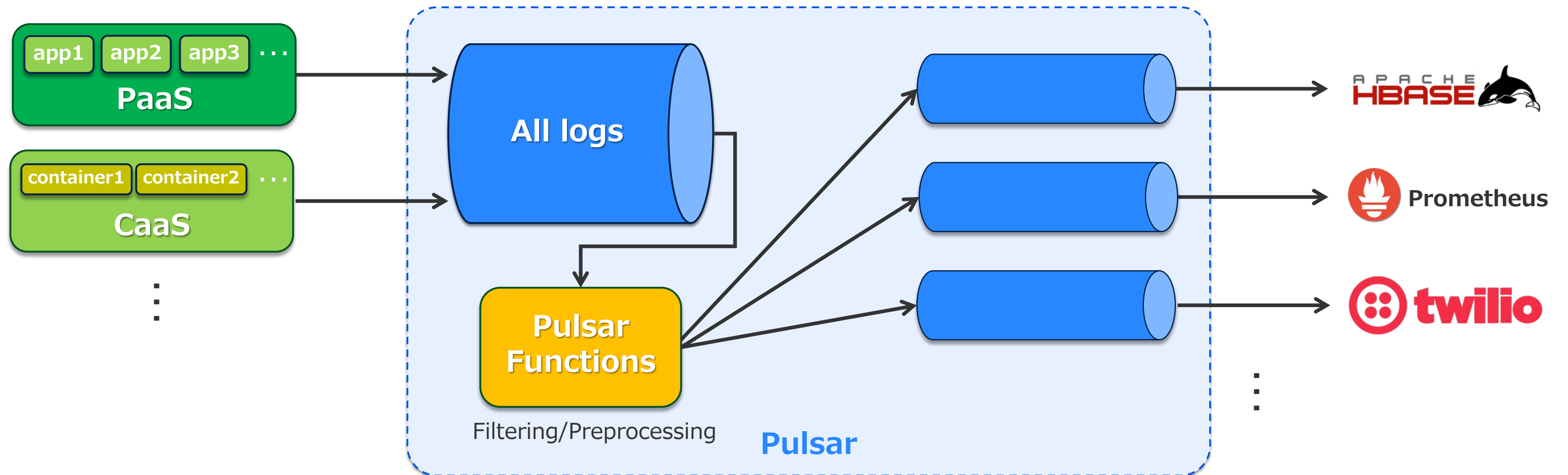


Case 3 – Log collection (under development)

Collect logs from all platforms: PaaS, CaaS, ...

Filtering / preprocessing by Pulsar Functions

Finally send to other DB/platforms: HBase, Prometheus, twilio, ...



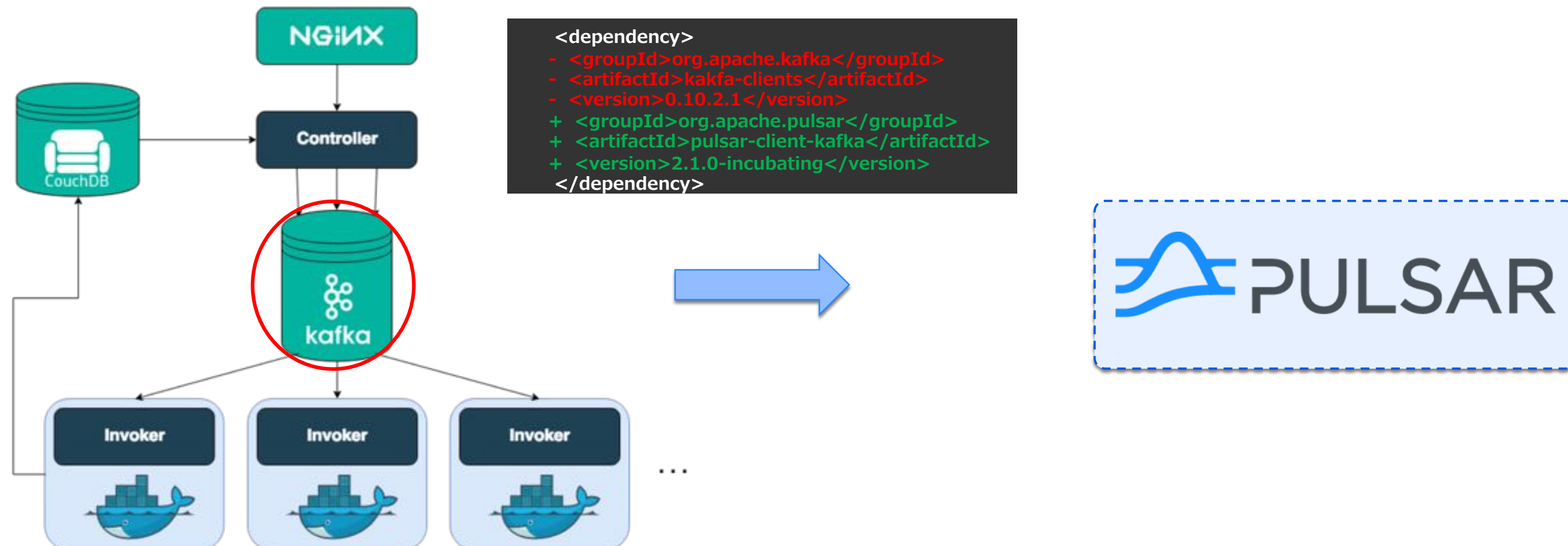
Case 4 – Migration from Kafka

We have an internal FaaS system using **Apache OpenWhisk**

Problem: FaaS team had to maintain **Apache Kafka**

Then they decided to migrate from Kafka to our internal Pulsar

Pulsar Kafka Wrapper needs no code change except configuration for migration



Conclusion

Conclusion

Apache Pulsar

- › Fast, Durable, Scalable pub-sub messaging
- › Has useful built-in features (Geo-replication, Pulsar Functions, etc.)

Feature releases (2.2.0)

- › Pulsar SQL

Documents

- › <https://pulsar.incubator.apache.org/>

Contact

- › <https://apache-pulsar.slack.com/>
- › users@pulsar.incubator.apache.org



References

- “Open-sourcing Pulsar, Pub-sub Messaging at Scale”
<https://yahooeng.tumblr.com/post/150078336821/open-sourcing-pulsar-pub-sub-messaging-at-scale>
- “Linked In Stream Processing Meetup – Apache Pulsar”
https://www.slideshare.net/KarthikRamasamy3/linked-in-stream-processing-meetup-apache-pulsar?qid=f7fade7e-632c-47fe-aa68-aea84a622866&v=&b=&from_search=2
- “Benchmarking Enterprise Streaming Data and Message Queuing Platforms”
<https://info.streaml.io/benchmarking-streaming-messaging-platforms>
- “Comparing Pulsar and Kafka: how a segment-based architecture delivers better performance, scalability, and resilience”
<https://streaml.io/blog/pulsar-segment-based-architecture/>

YAHOO!
JAPAN