RabbitMQ, what's new

Gabriele Santomaggio - Developer @vmware

{codemotion}
Online Tech Conference
- Italian edition -

23-24-25 Marzo, 2021

CC BY NC SA

About me..

Developer @ vmware

- Gabriele Santomaggio
- @gsantomaggio
- Platform Team (RabbitMQ / Kubernetes)

Confidential | ©2021 VMware, Inc.

Maybe you don’t know…

RabbitMQ and vmware

- VMWare acquired Pivotal
- RabbitMQ is widely used (See Amazon and other cloud providers)
- Docker Image over 1 Billion of downloads
- A dedicated cloud team
  - Kubernetes
  - Operator
  - Mesh
  - Istio/Traefik/Envoy/etc..
  - Monitoring/Alarms
- Move to cloud native
- Tanzu RabbitMQ (https://www.rabbitmq.com/tanzu/)

Agenda

Overview

- **Streaming Queues**
- **Kubernetes**
- Real example
  - CGP / k8s
  - Traefik / HTTPS and AMQPS
  - Cert Manager / lets encrypt
  - TLS SNI for routing
    - one streaming cluster
    - standard cluster

Confidential | ©2021 VMware, Inc.

Stream Overview

A new type of data structure in RabbitMQ

- Persistent and replicated
- Models an append-only log
- Non-destructive consumer semantics
- ***Can be used as a regular AMQP 0.9.1 queue or through a new binary protocol***
- ***Strong points:***
  - Large fan-outs
  - Replay / Time-travelling
  - Throughput Performance (several orders of magnitude higher throughput)
  - Large logs

Confidential | ©2021 VMware, Inc.

Why Streams in RabbitMQ?

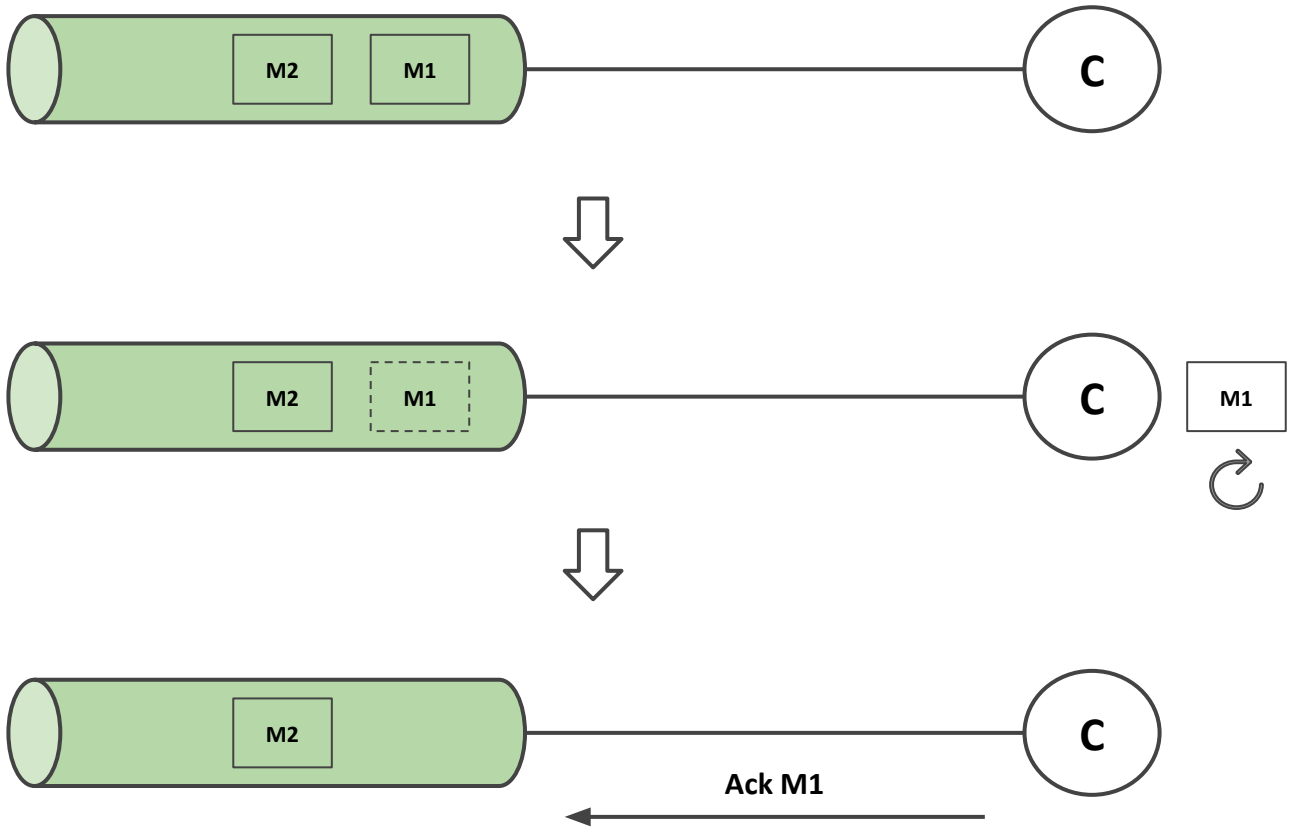
“Traditional” queues are not suited for some use cases

- Consumption is destructive
- This does not play well for some scenarios
- Streams are meant to complement traditional queues
- Streams expand the use cases RabbitMQ can address

Confidential | ©2021 VMware, Inc.

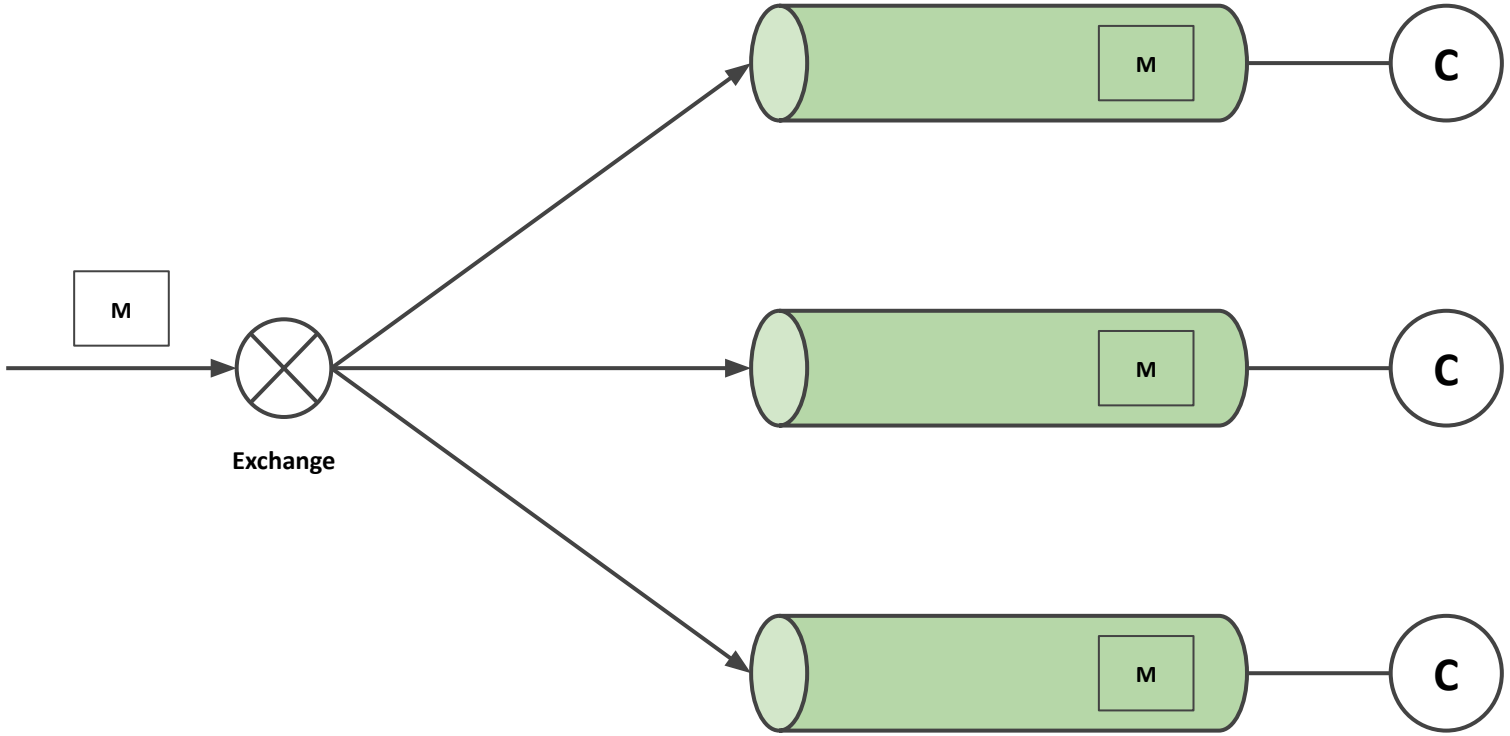
Consuming with traditional queue

A message is removed from the queue once acknowledged ("destructive consuming").



Fan-out with traditional queues

Fan-out requires several queues.

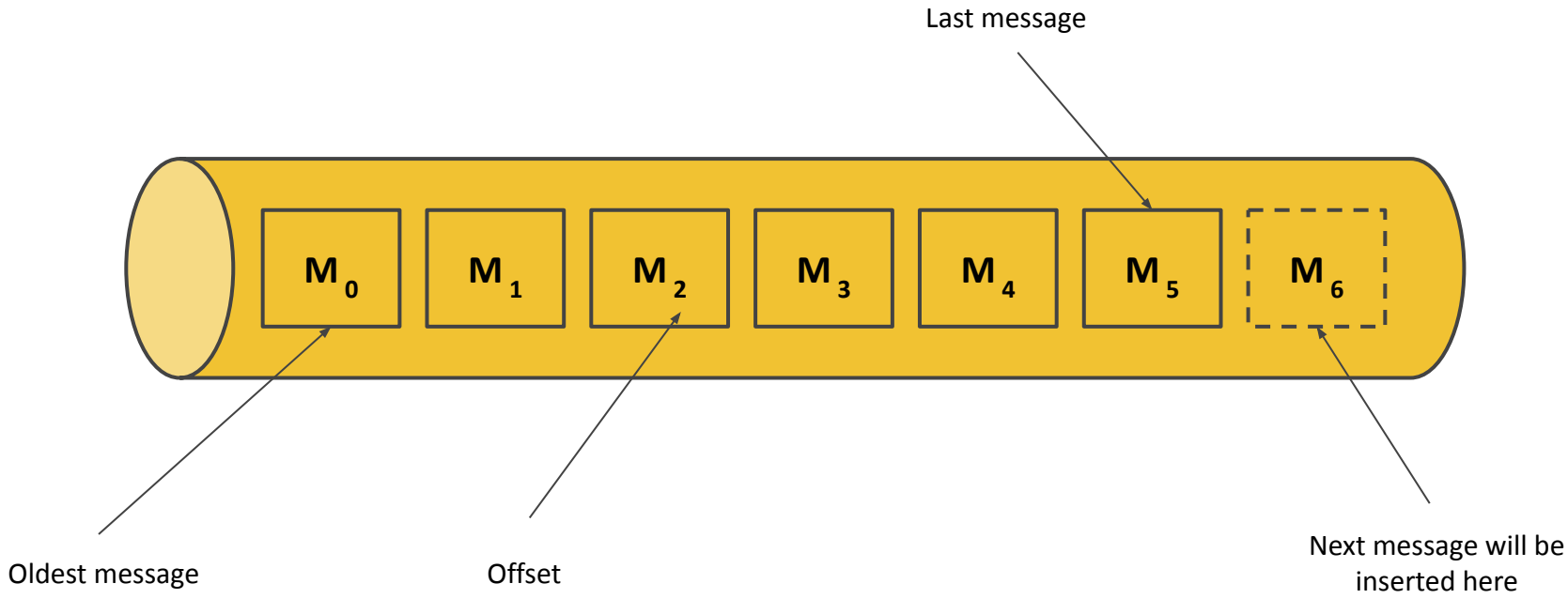


Confidential | ©2021 VMware, Inc.

The Log Abstraction

A stream models an append-only log

- FIFO data structure
- No destructive reading



Stream Creation with AMQP 0.9.1

A stream is exposed as an AMQP 0.9.1 queue

- Any AMQP 0.9.1 client library can be used to create a stream

```
channel.queueDeclare(
  "my-stream",
  true,           // durable
  false, false, // not exclusive, not auto-delete
  Collections.singletonMap("x-queue-type", "stream")
);
```

Consuming from a Stream with AMQP 0.9.1

The consume operation can specify an offset parameter

- Any AMQP 0.9.1 client library can be used to consume from a stream

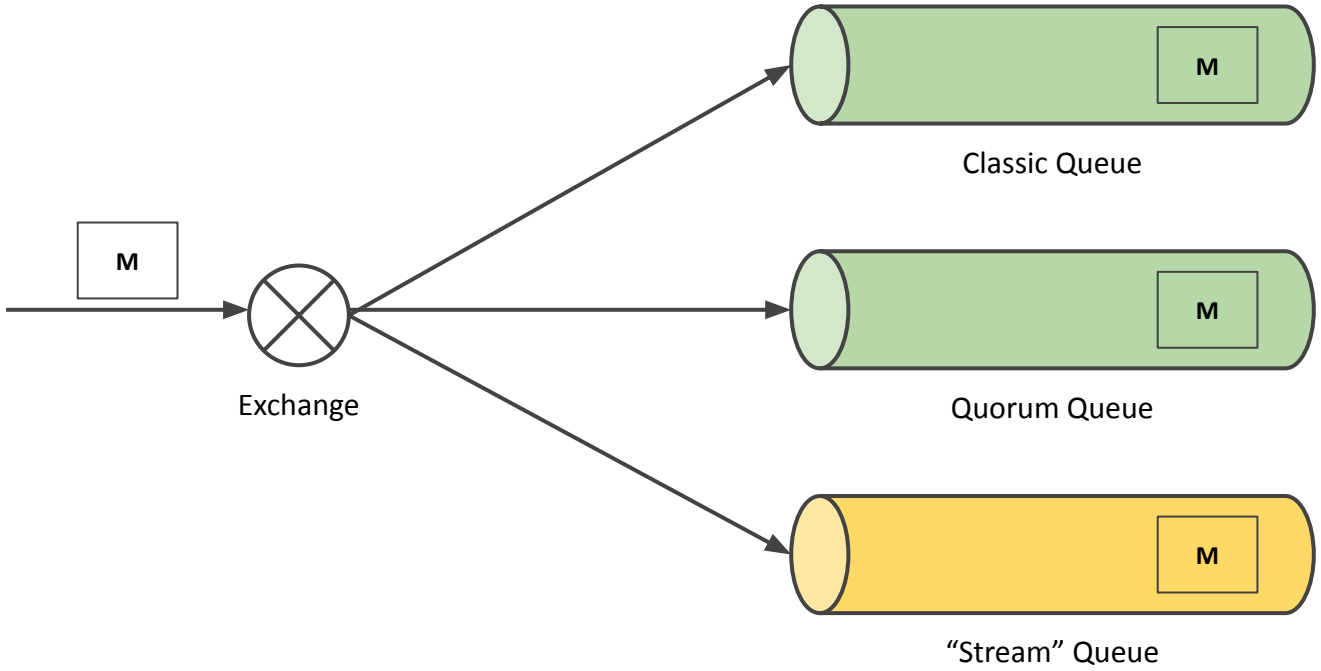
```
channel.basicConsume(
  "my-stream",
  false, // not auto-ack
  Collections.singletonMap("x-stream-offset", 0),
  (s, delivery) -> { }, // delivery callback
  s -> { } // cancel callback
);
```


Confidential | ©2021 VMware, Inc.

Stream in the AMQP Model

A stream is exposed as a regular AMQP 0.9.1 queue

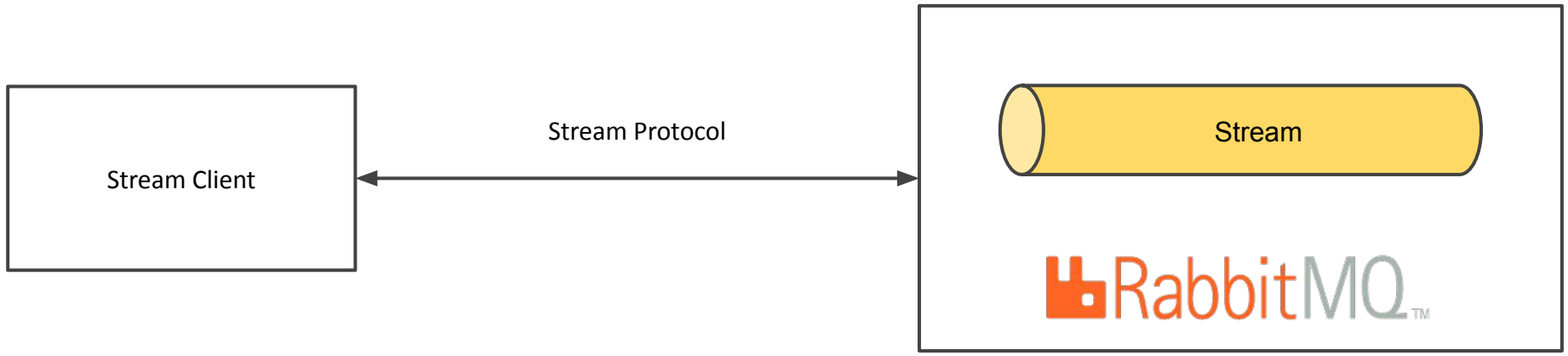
- Can enjoy all the AMQP model features, like routing
- Remains limited by the protocol though
  - Protocol aiming at interoperability



Stream Plugin

The Stream Plugin will ship with RabbitMQ

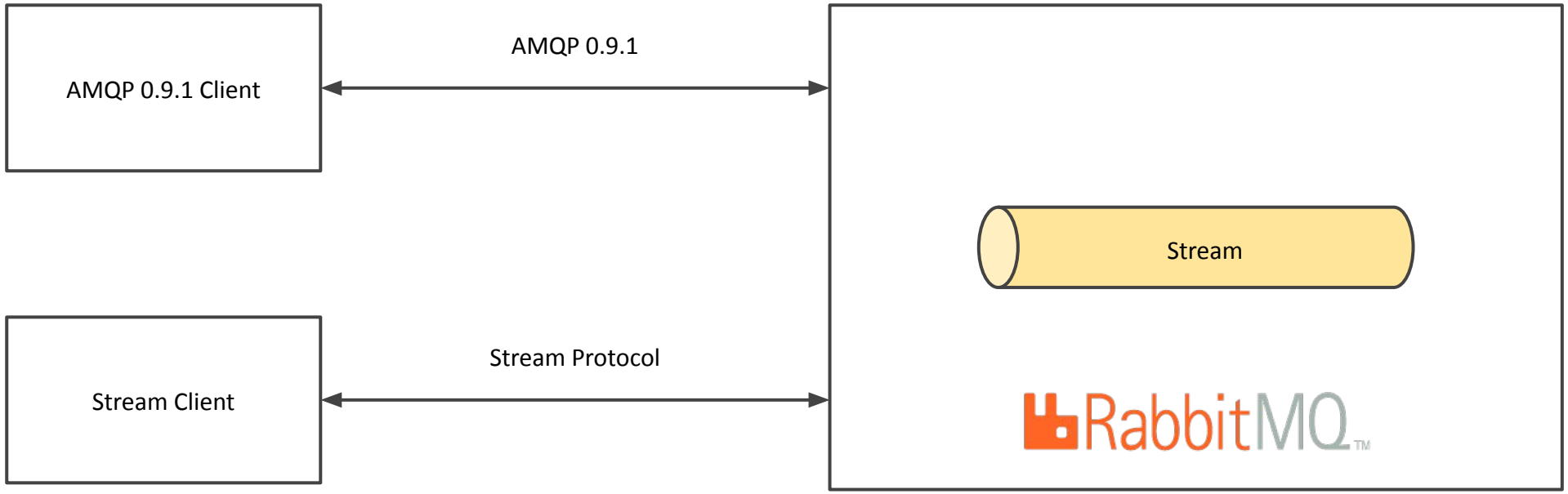
- The stream client directly interacts with streams
- Publish to/consume from streams
  - no exchange involved for publishing



Stream Interoperability

A stream is accessible from AMQP 0.9.1 and the stream protocol

- Client from different protocols can interact with the same stream
- It remains accessible from other protocolos as well (MQTT, STOMP)



Confidential | ©2021 VMware, Inc.

Performance

Simplistic benchmarks, stream protocol almost 25 times faster

- AMQP 0.9.1, local node, in-memory queue, 1000 outstanding confirms (with PerfTest)

```
id: test-180050-924, sending rate avg: 42573 msg/s
id: test-180050-924, receiving rate avg: 42541 msg/s
```

- Stream Protocol, local node, stream is persisted (with Stream PerfTest)

```
Summary: published 1279444 msg/s, confirmed 1279019 msg/s,
consumed 1279019 msg/s, latency 95th 12161 µs, chunk size 2910
```

- 42 K (AMQP) vs 1,279 K (stream protocol) => throughput ~ 30 higher

Current State

Under ongoing development

- Not production-ready, still in alpha
- Merged into RabbitMQ master (for RabbitMQ 3.9)
- Can be tested and evaluated
- Feedback welcome (see “Getting Started”)

Confidential | ©2021 VMware, Inc.

Start Experimenting!

Use the stream plugin and the performance tool with Docker

```
# run the broker
docker run -it --rm --network host pivotalrabbitmq/rabbitmq-stream
# open another terminal and run the performance tool
docker run -it --rm --network host pivotalrabbitmq/stream-perf-test
```


Confidential | ©2021 VMware, Inc.

Kubernetes

Cluster Operator

DEMO

Confidential | ©2021 VMware, Inc.

Kubernetes

Going to cloud native

- Cluster operator
- Integration with prometheus
- Ready Grafana templates
- Service mesh integration

Confidential | ©2021 VMware, Inc.

Kubernetes

Cluster Operator

- Clusters Life cycle
- Easy deploy
- Easy TLS configuration
- Automatic PVC binding
- Default values
  - Note about production
- Override all the parameters
- Automatic procedures
  - Rebalance
- Enable Prometheus by default (HTTP/HTTPS)
- Easy Scaling
  - Note about scaling down

Kubernetes

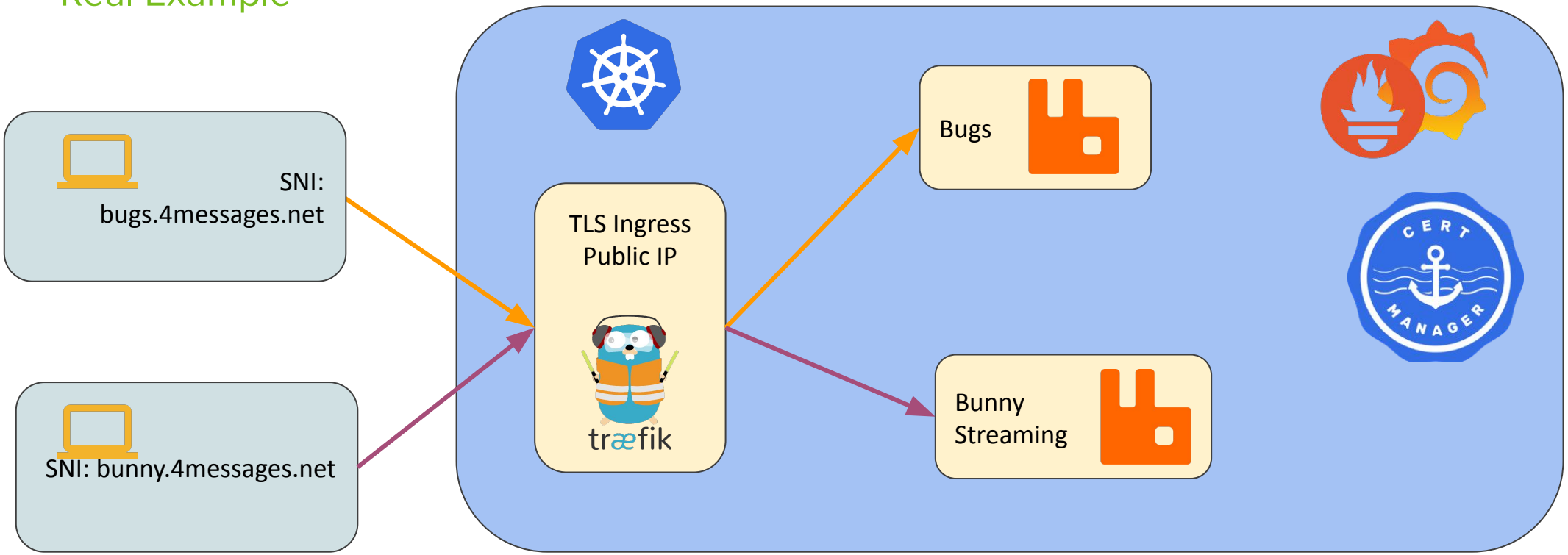
Cluster Operator

DEMO

Confidential | ©2021 VMware, Inc.

Kubernetes

Real Example



namecheap

CLOUDFLARE®

Let's Encrypt

Confidential | ©2021 VMware, Inc.

Kubernetes

Cluster Operator

DEMO

Confidential | ©2021 VMware, Inc.

Kubernetes

Prometheus/ Grafana

- 15691/metrics endpoint
- 15692/metrics endpoint
- Grafana templates ready on https://grafana.com/orgs/rabbitmq
- New templates are coming
  - Alarm
  - Integration

Confidential | ©2021 VMware, Inc.

Questions?

Confidential | ©2021 VMware, Inc.

Thank you!

gsantomaggio@vmware.com
@gsantomaggio

Confidential | ©2021 VMware, Inc.