FS20

DISTRIBUTED SYSTEMS

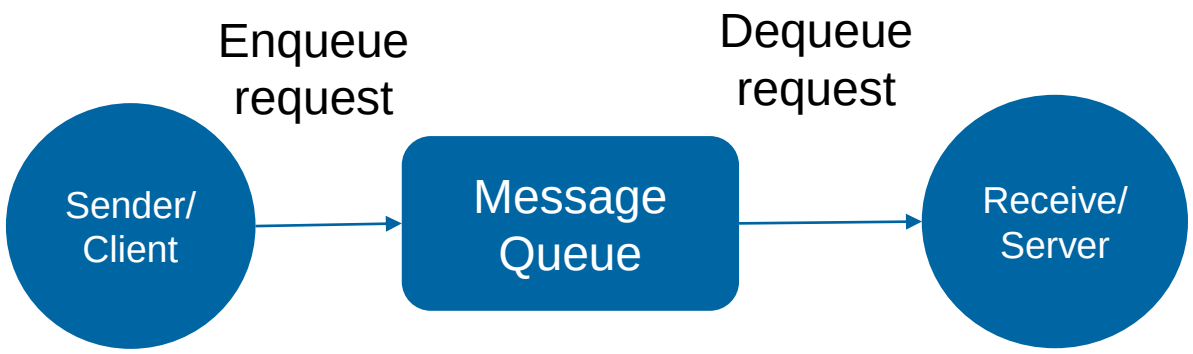
Message Queues

T. Bocek
thomas.bocek@hsr.ch

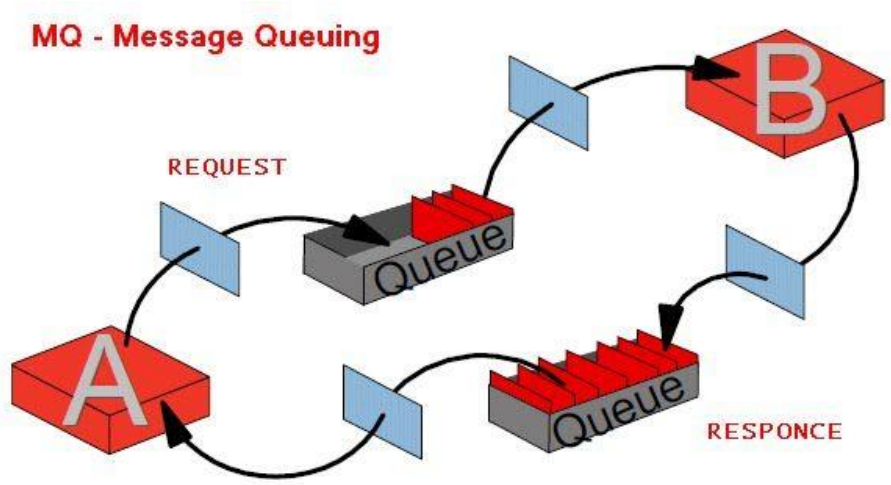
HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL
FHO Fachhochschule Ostschweiz

Message Queues

- **Message Queues in Distributed Systems**
  - Communicating without continuous connection
  - Decoupling: hardware is not reliable (lecture 1)
  - Systems can be heterogeneous (small, large machines)
- **Messages placed on queue by one systems, taken by another system**



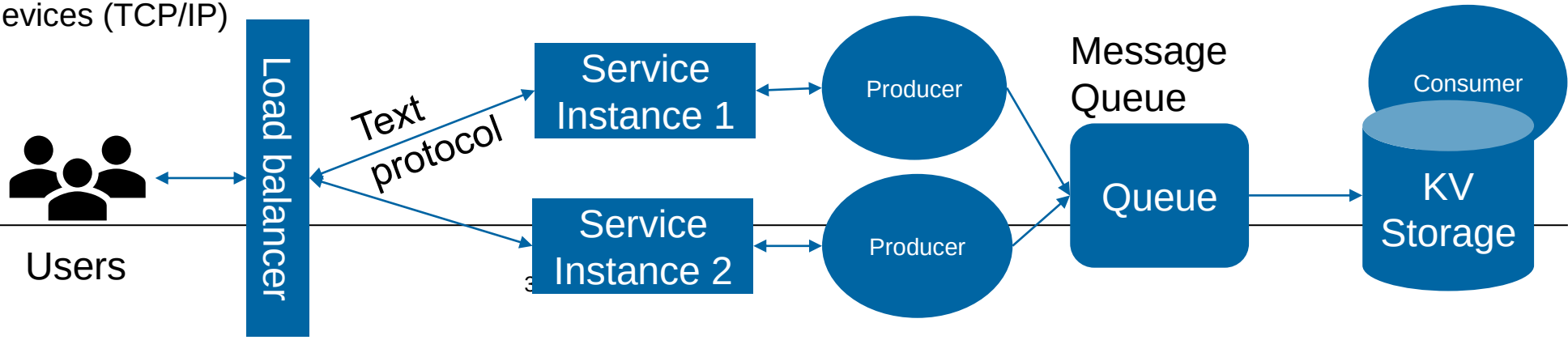
- **Channels in Go (synchronization of goroutines)**
- **Message queues and mailboxes**
  - Similar to Email
  - Can be used for IPC
  - Asynchronous communications protocol
    - Sender and receiver do not need to interact with message queue at the same time
  - Messages in queues may be persisted



- **Standards**
  - Sun Microsystems' JMS specification
    - Early attempt, disadvantage API spec / Java only
  - 3 standards being used - specification includes wire protocol
    - Advanced Message Queuing Protocol (**AMQP**) – feature-rich message queue protocol (HTTP)
    - Streaming Text Oriented Messaging Protocol (**STOMP**) – simple, text-oriented message protocol (HTTP)
    - **MQTT** (formerly MQ Telemetry Transport) - lightweight message queue protocol especially for embedded devices (TCP/IP)

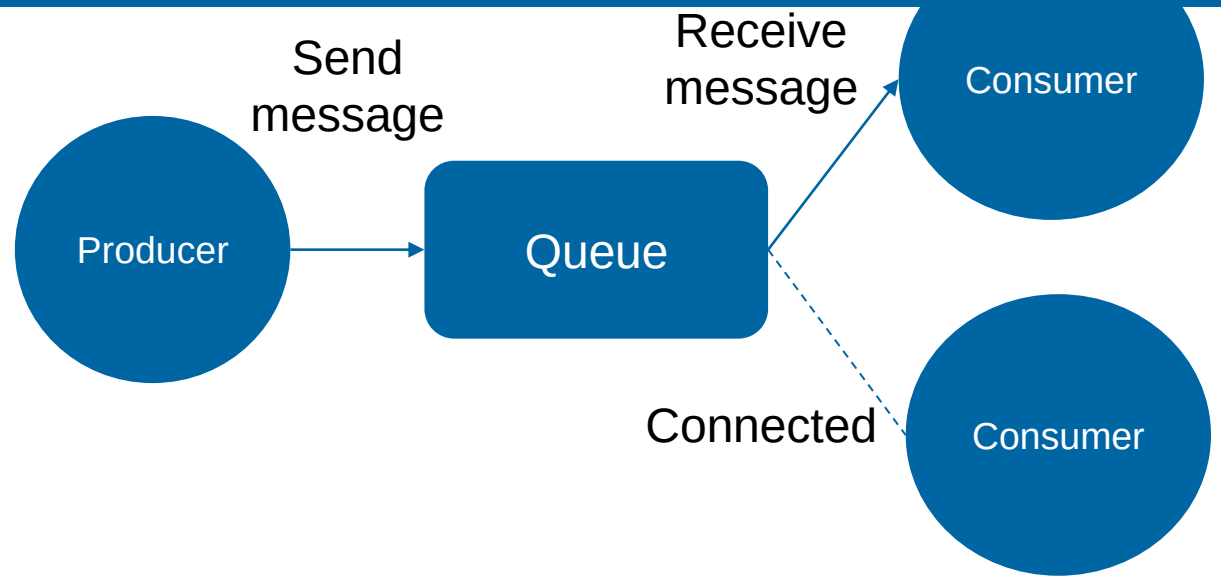
- **RabbitMQ (AMQP, STOMP, MQTT), ActiveMQ (AMQP, STOMP, MQTT), QPID (AMQP), ZeroMQ (ZMTP, runs without broker)**
  - Message Queues in CT
  - Stateless-service will only start if storage backend runs

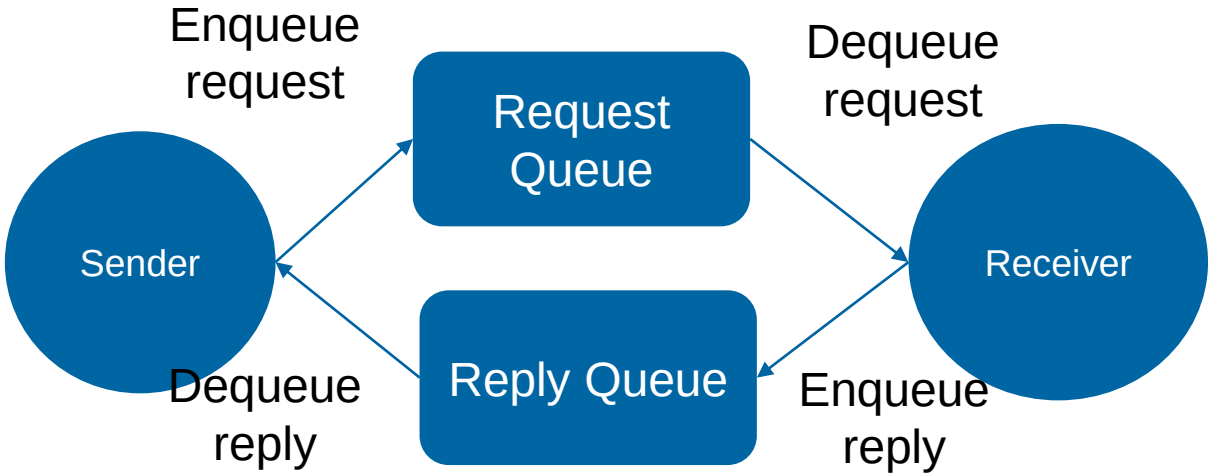
→ Decouple it

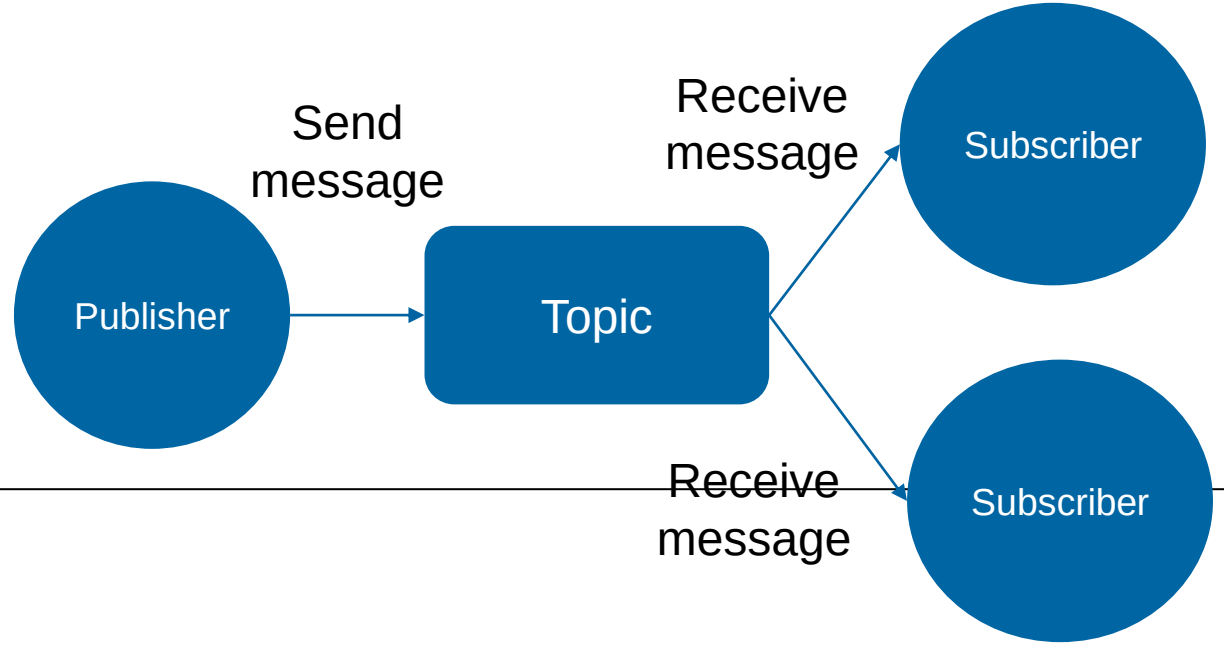


Message Queues

- **Topologies (Terminology!)**
  - Point-to-point, Producer-Consumer (can be used for load balancing)
  - Publish-Subscribe
  - Bidirectional Queues
- **Terminology**
  - one-to-many, many-to-one, fanout, …

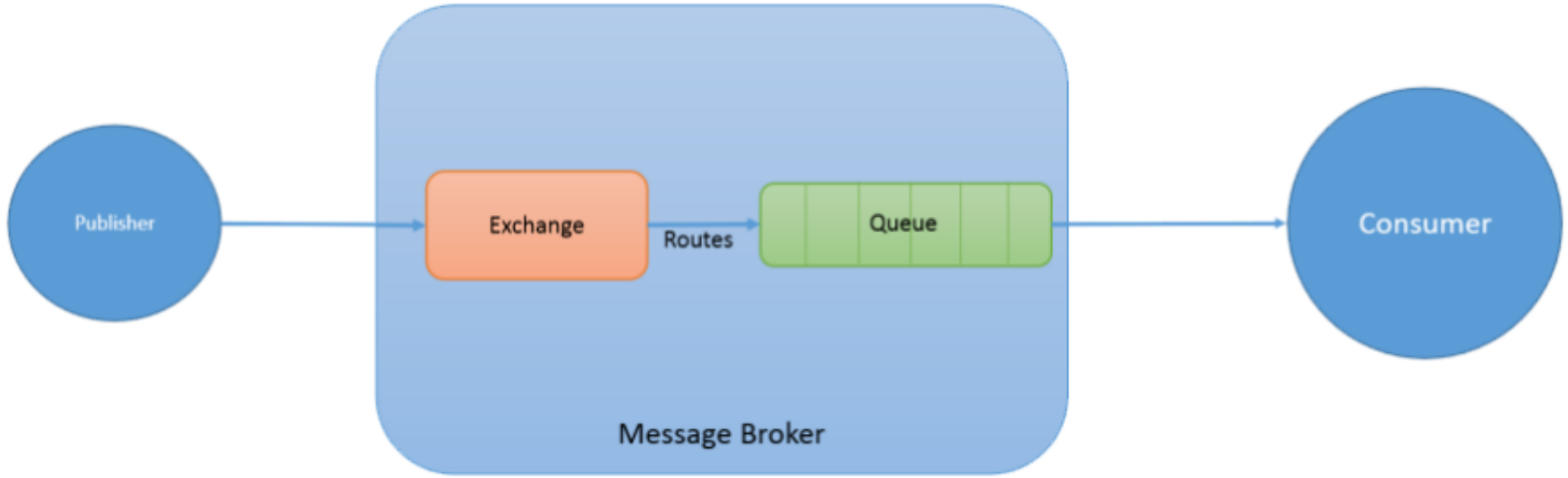






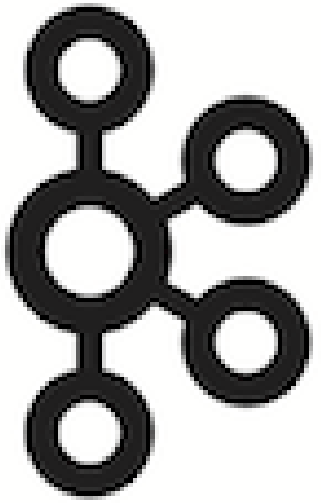
Message Queue / RabbitMQ

- **Message Brokers, e.g. in AMQP**
  - Exchange: routing keys selects relevant queue
  - Queue: messages are stored, consumer can get the message (durable or transient)
  - Example RabbitMQ
    - Ubuntu: apt install rabbitmq-server
    - systemctl start rabbitmq-server
  - Management
    - rabbitmq-plugins enable rabbitmq_management



Message Queue / Apache Kafka

- **Apache Kafka: Distributed streaming platform / Message Queue**
  - Read and write streams of data (message queue)
  - Store streams of data in a distributed, replicated fault-tolerant cluster
  - Stream processing (Stream API) – aggregation or joining streams
  - Real-time data and streaming apps
- **Kafka uses ZooKeeper**
  - Distributed hierarchical key-value storage
  - Fast Processing in "read-dominant" workloads
  - Scalable: performance can be improved by adding nodes.
  - Used at Rackspace, Yahoo, Reddit, eBay, Swisscom…
- **Example – github**
  - Consumer / producer / streaming
  - Streams / Table



Questions?

Dr. Thomas Bocek
thomas.bocek@hsr.ch