Indirect communication

Message queues

Characteristics

Whereas groups and publish/subscribe systems provide a one-to-many style of communication, messages queues provide a point-to-point service using the concept of a **message queue as an indirection**.

Message queues are also referred to as **Message-Oriented Middleware**.

Commercial middleware such as

- IBM WebSphere MQ
- Microsoft MSMQ
- Oracle Stream Advanced Queuing (AQ)

Applications

Freie Universität Berlin

Enterprise application integration (EAI)

- is an integration framework composed of a collection of technologies and services which form a **middleware** (e.g. distributed message queues) to enable integration of systems and applications across the enterprise

Transaction Processing Systems

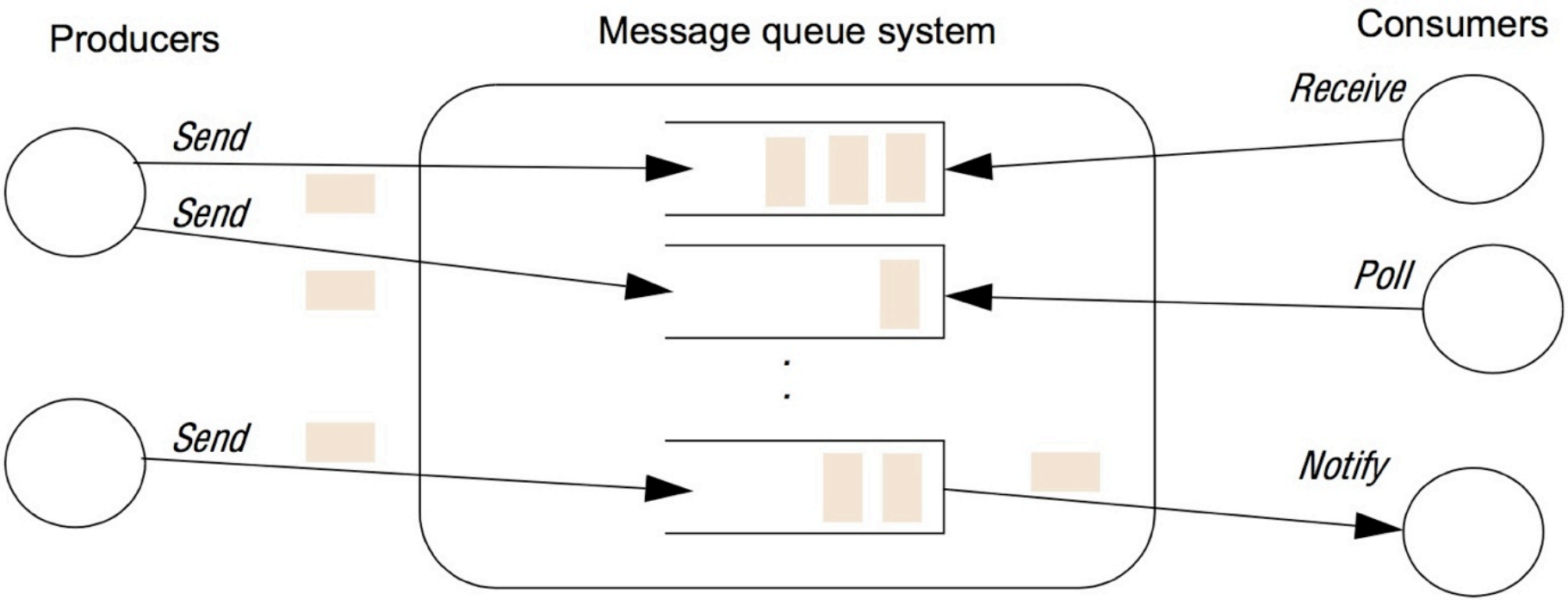
- Message queues have intrinsic support of transactions for transactions

Freie Universität Berlin

Message Queues

The programming model

Programming model



Queuing policy typically FIFO, priority or other selection criteria (e.g. metadata based)

Styles of “Receive”

blocking receive

- will block until an appropriate message is available

non-blocking receive (a polling operation)

- will check the status of the queue and return a message if available, or a not available indication otherwise

notify operation

- will issue an event notification when a message is available in the associated queue.

Properties

Persistent messages

- Messages are stored until they are consumed

Reliable delivery

- Integrity: the message received is the same as the one sent, and no messages are delivered twice
- Validity: message is eventually received (but not guarantees about the timing of delivery)

Transactional: „all steps in a transaction or nothing“

- Typically supported by additional external tranaction service

+ (message transformation, security, …)

Freie Universität Berlin

Message Queues

Implementation issues

Centralized vs. Decentralized

Centralized

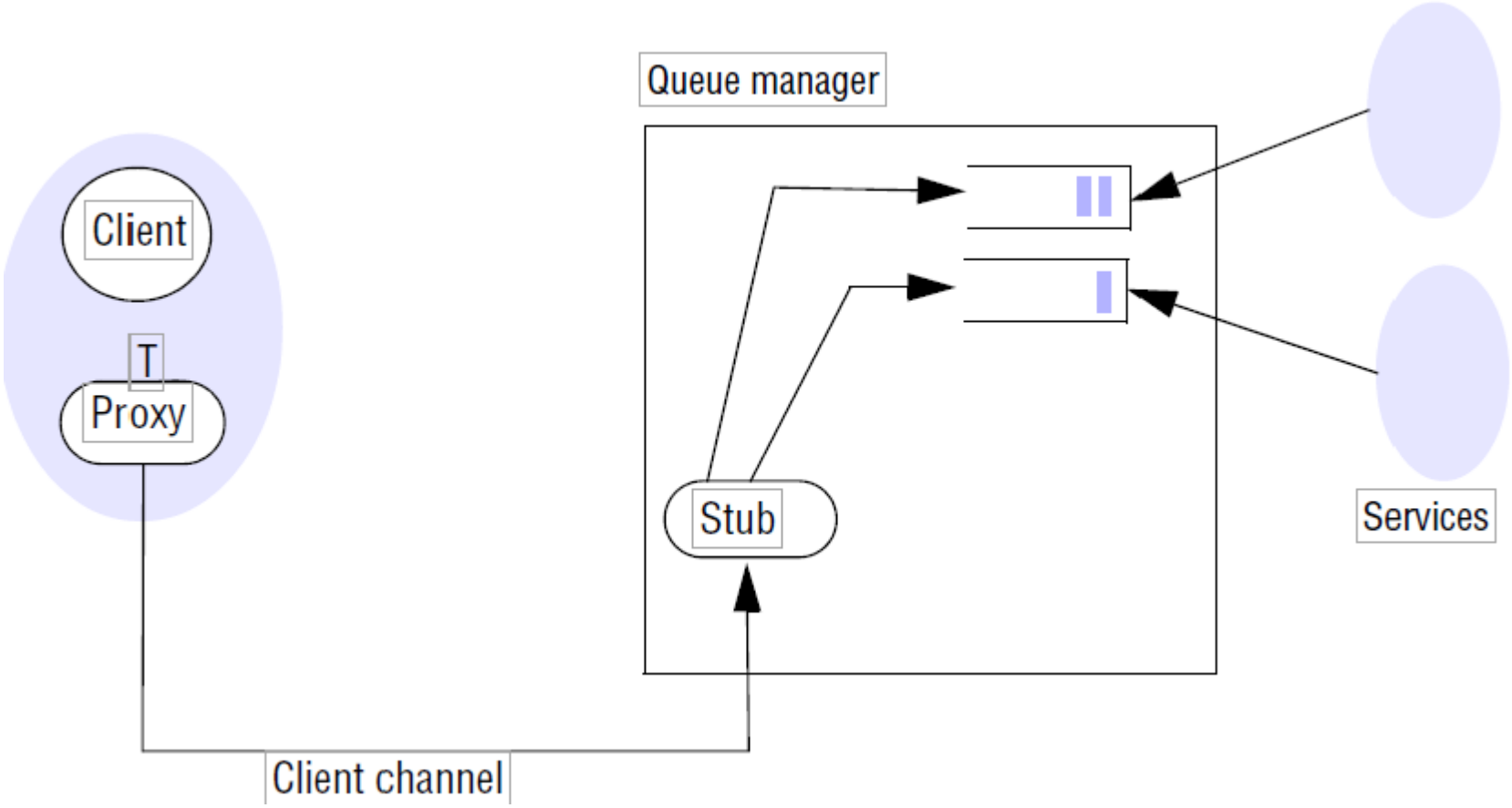
one or more message queues managed by a centralized queue manager located at a given node

- Advantage: simplicity
- Drawback: manager can become heavy-weight, bottelneck, single-point-of-failure

Decentralized

federated structure with **message channel** as a unidirectional connection between two queue managers that is used to forward messages asynchronously from one queue to another. A message channel is managed by a **message channel agent** (MCA) at each end

Centralized Queue Manager



Client Channels issue commands of the **Message Queue Interface** (MQI) on the proxy and sent them transparently to the **Message Queue Manager** via RPC.

Decentralized Topologies

Freie Universität Berlin

Decentralized queue managers allow network topologies, e.g., trees, meshes or a bus-based networks

Example: Hub-and-spoke approach

- Often used in large scale geographically distributed deployments
  - ability to connect to a local spoke over a high-bandwidth connection
- One queue manager is **hub** providing main services
  - placed somewhere appropriate in the network, on a node with sufficient resources to deal with the volume of traffic
- Clients connect through queue managers called **spokes** which relay messages to the queues of the hub
  - placed strategically around the network to support different clients