

Distributed Systems



indirect communication
Johan Montelius

Indirect Communication

- Communication that is not coupled in time, space:
 - group communication
 - publish/subscribe
 - message queues or forwarding
 - shared memory or tuple space



Space uncoupling

- The sender does not know, or need to know the receiver and vice versa.
 - several receivers, a group
 - multi/broad-casting
 - logical receiver



Time uncoupling

- The sender and receiver need not exist at the same time.
 - mail box
 - shared media
- Is this the same as asynchronous communication?



Group communication

- More than simple broadcast
 - the group is well defined and managed
 - ordered delivery of messages
 - fault tolerant, delivery guarantees, failures...

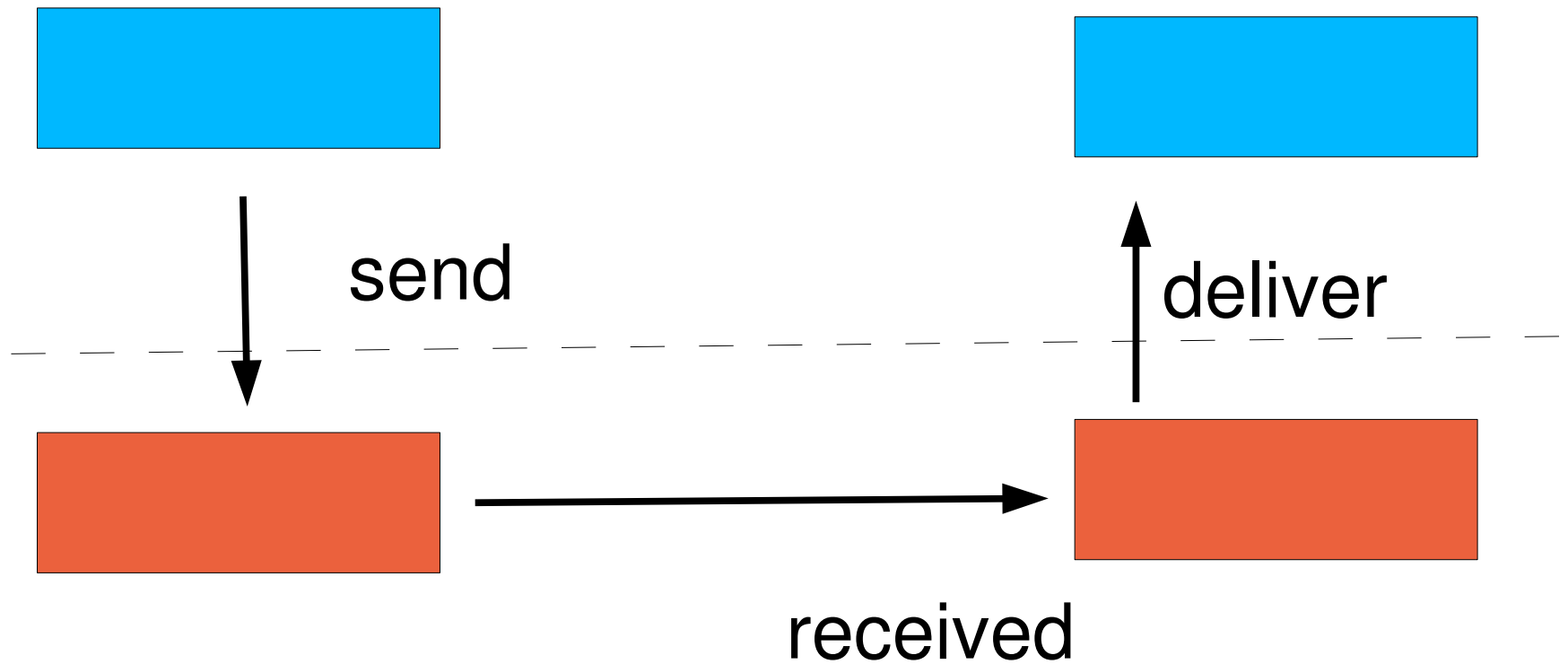


Group communication



- Advantages
 - make use of underlying network, a group message becomes a multicast
 - semantics
- Closed vs Open
 - Can a process outside of the group send messages to the group?
- Management
 - Processes join and leave the group coordinated by a *group membership service*.

Send/Receive/Deliver



Reliability



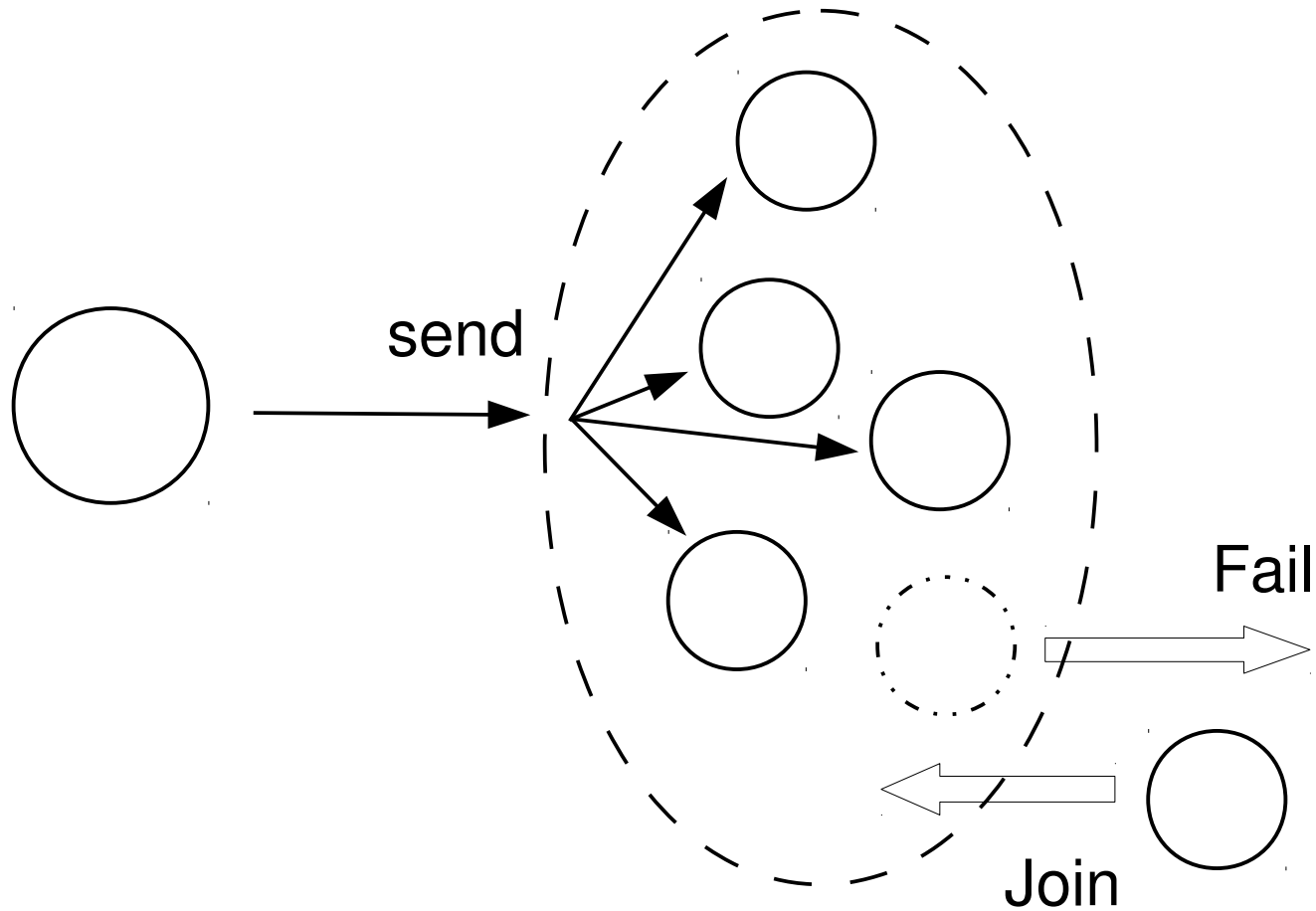
- Integrity
 - A delivered message is sent message.
- Validity
 - A send message is eventually delivered.
- Agreement
 - If one member delivers a message then all members deliver the message.



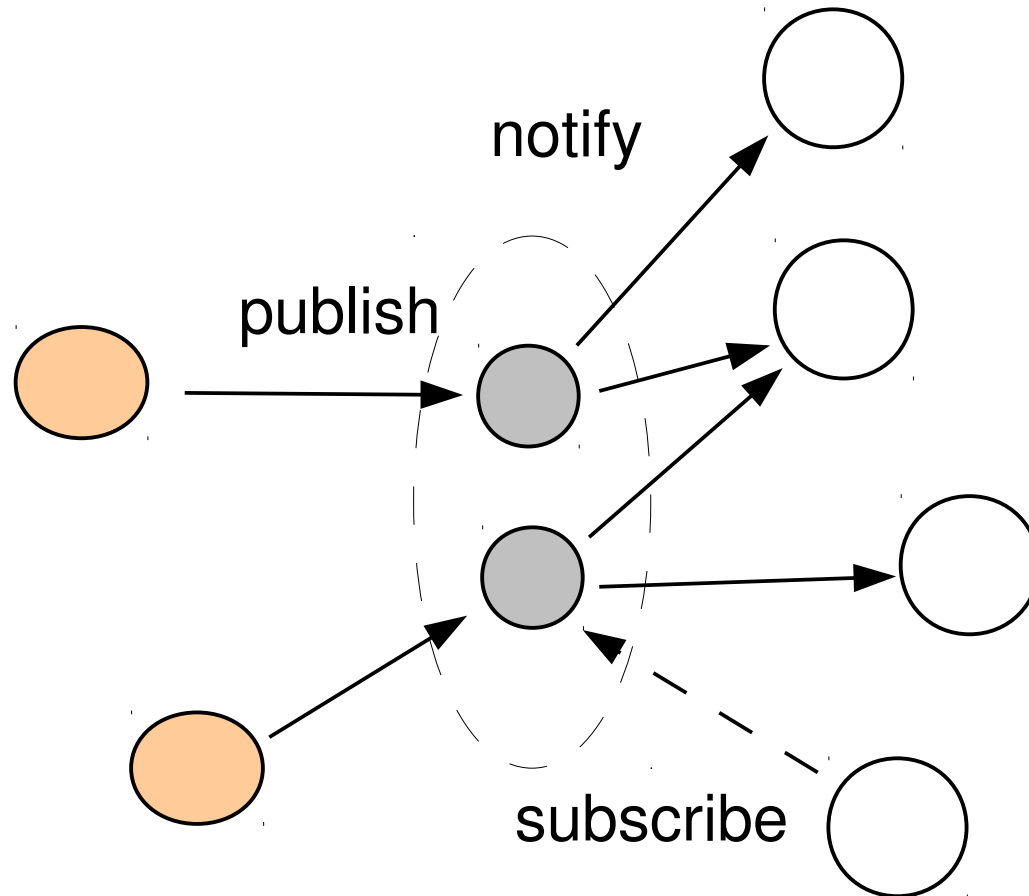
Order of delivery

- FIFO
 - In the order send by sender.
- Causal order
 - In a logical order.
- Total order
 - In the same order by all members.

Group membership service



Publish-subscribe systems





Subscription model

- Channel
 - publisher has to select a *channel*
- Topic
 - each message belongs to a topic
- Content
 - the subscriber decides
- Type
 - programming language dependent

Implementations



- Centralized solution
 - simple, single point of failure
- Distributed
 - Scales well (especially if we relax delivery guarantees)
 - Strategies
 - Flooding
 - Filtering
 - Advertisement
 - Rendezvous
 - Gossip



Message queues

- Sender sends message to a message queue (FIFO)
- Receiver read from queue (on given or default queue)
 - blocking, non-blocking, event
- Messages remain in queue until read (different from publish/subscribe)
- Erlang
 - Message passing is the only way to communicate.

Shared memory



- Why not use a model that we know?
 - read and write to memory
- Problems
 - data representation
 - synchronized access
 - cache consistency
- Should it be transparent?

Tuple space

