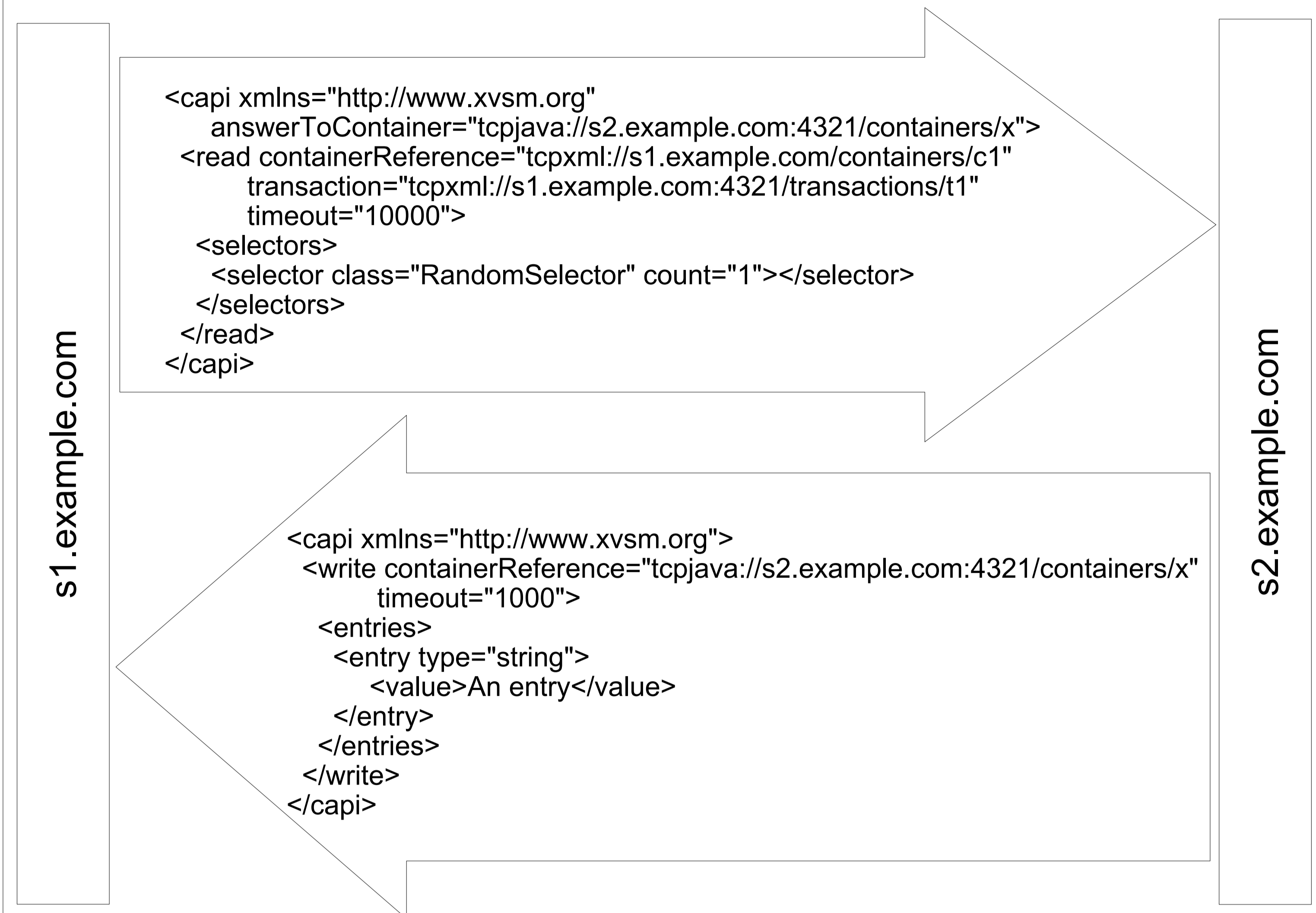


Abstract

Designing and implementing distributed applications is a complex process. A lot of aspects, like scalability, availability, security, synchronisation and heterogeneous environments have to be considered. This thesis describes MozartSpaces, the Java implementation of XVSM (eXtensible Virtual Shared Memory). XVSM is an extensible, distributed, space based computing middle-ware which addresses a lot of the issues very well. It supports developers in creating distributed applications by offering a natural, p2p based abstraction over the underlying physical network. Instead of exchanging messages like in conventional systems, communication on top of XVSM is realised by shared data structures. This thesis deals especially with the realisation of the transaction management, the extensibility through custom coordination models and the platform independent XML protocol.

XML Communication Protocol

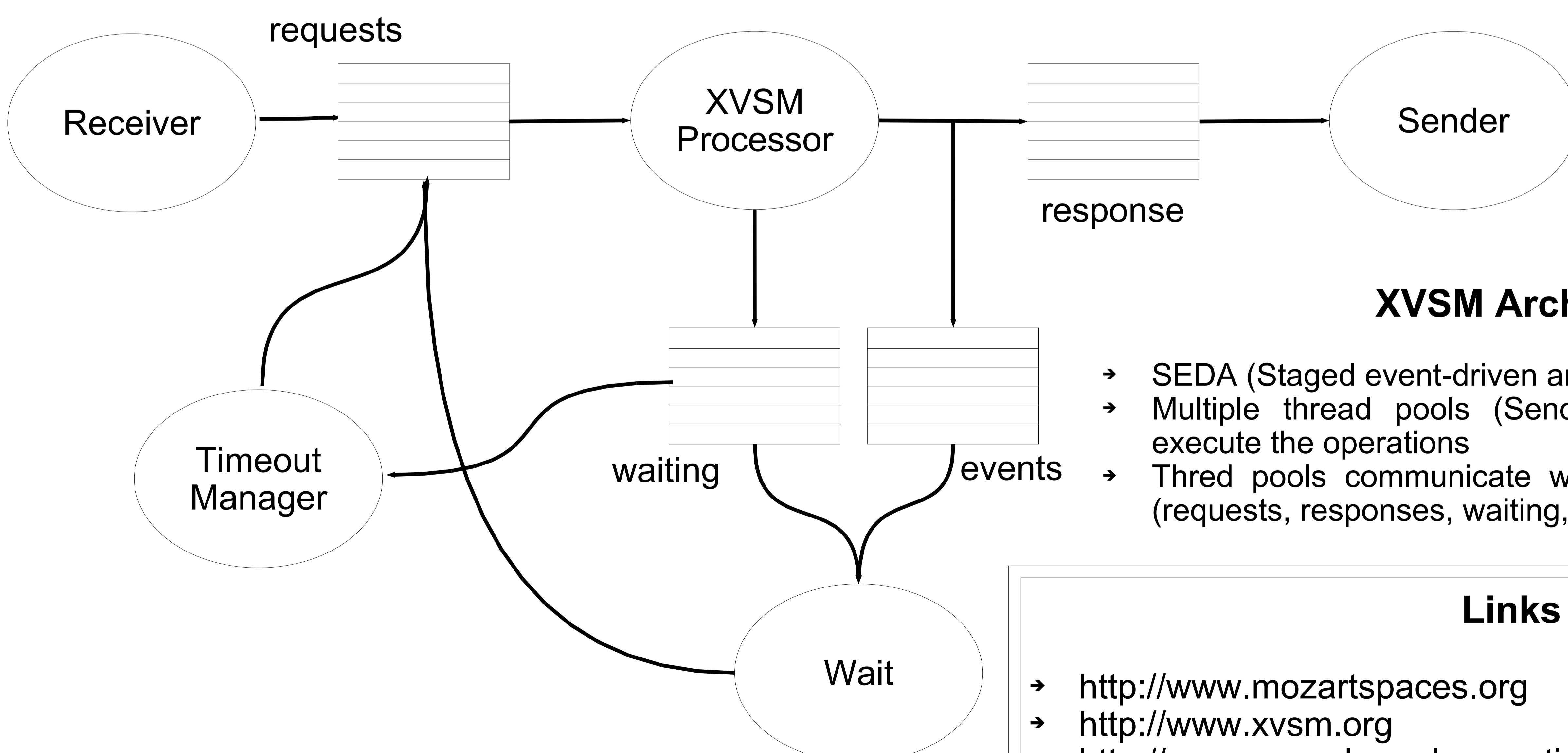
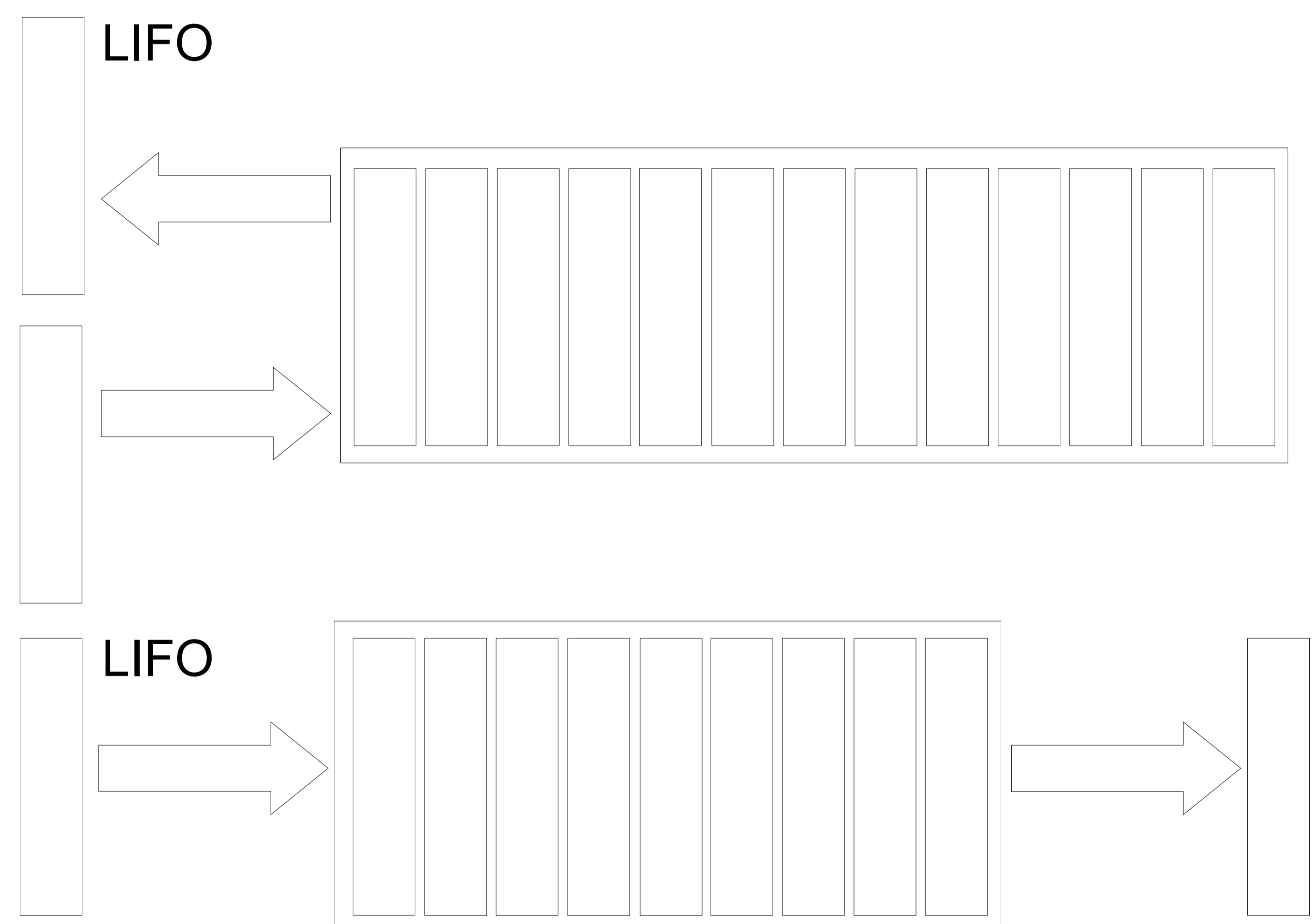
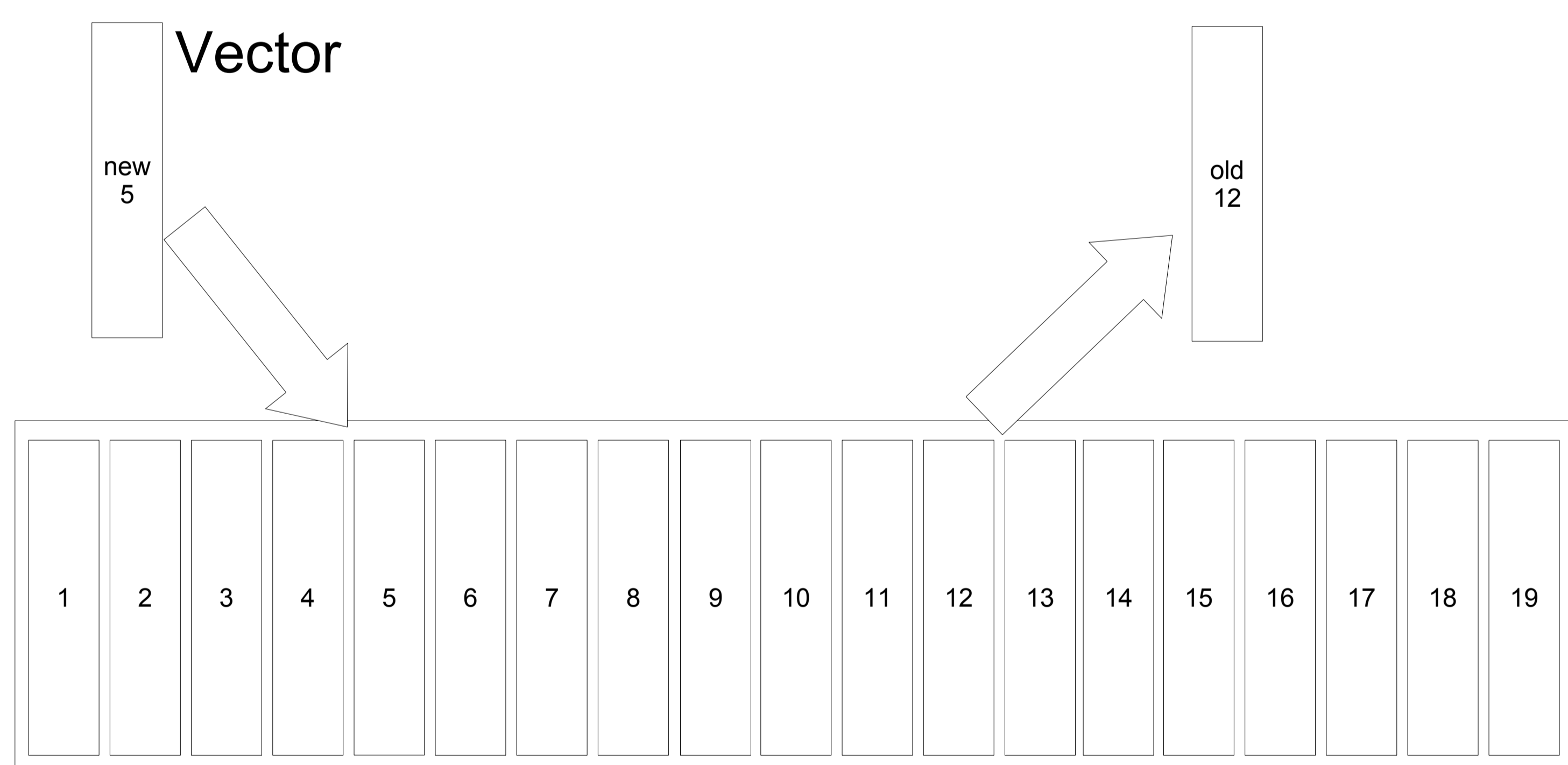
→ XVSM nodes communicate with each other using a platform independent XML protocol



Customisable Coordination Models

User can implement custom coordination models

- Realisation of a specific order
 - first in – first out: realisation of a queue
 - last in – first out: realisation of a stack
 - Vector: a list of entries (expanded on insertion and shrank on deletion)
 - ...
- Implementation of complex pattern
 - auctions
 - voting
 - ...



XVSM Architecture

- SEDA (Staged event-driven architecture) based architecture
- Multiple thread pools (Sender, Wait, XVSM Processor) execute the operations
- Three thread pools communicate with each other using queues (requests, responses, waiting, events)

Links

- <http://www.mozartspaces.org>
- <http://www.xvsm.org>
- <http://www.spacebasedcomputing.org>