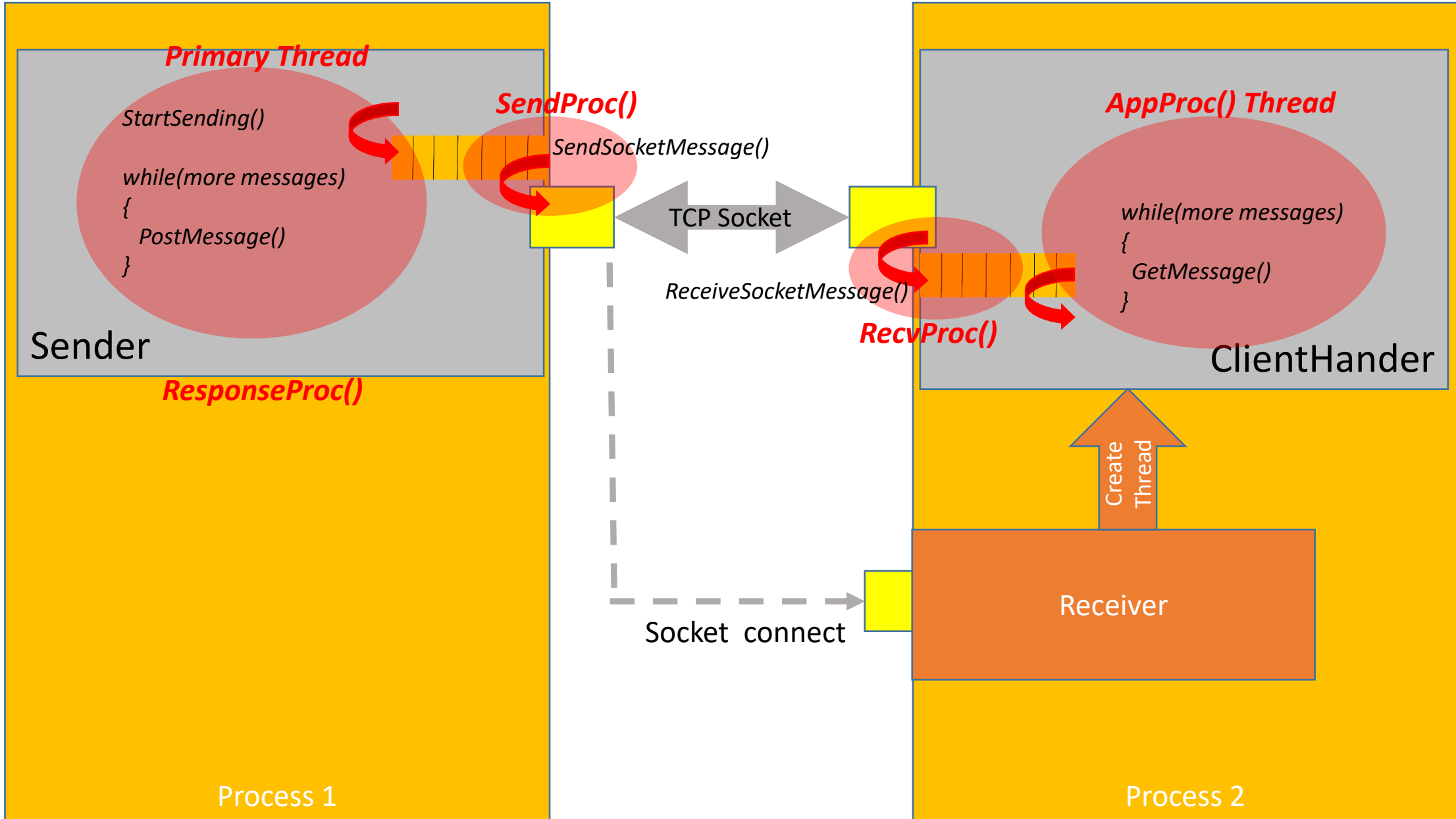


Message Passing Library (MPL)

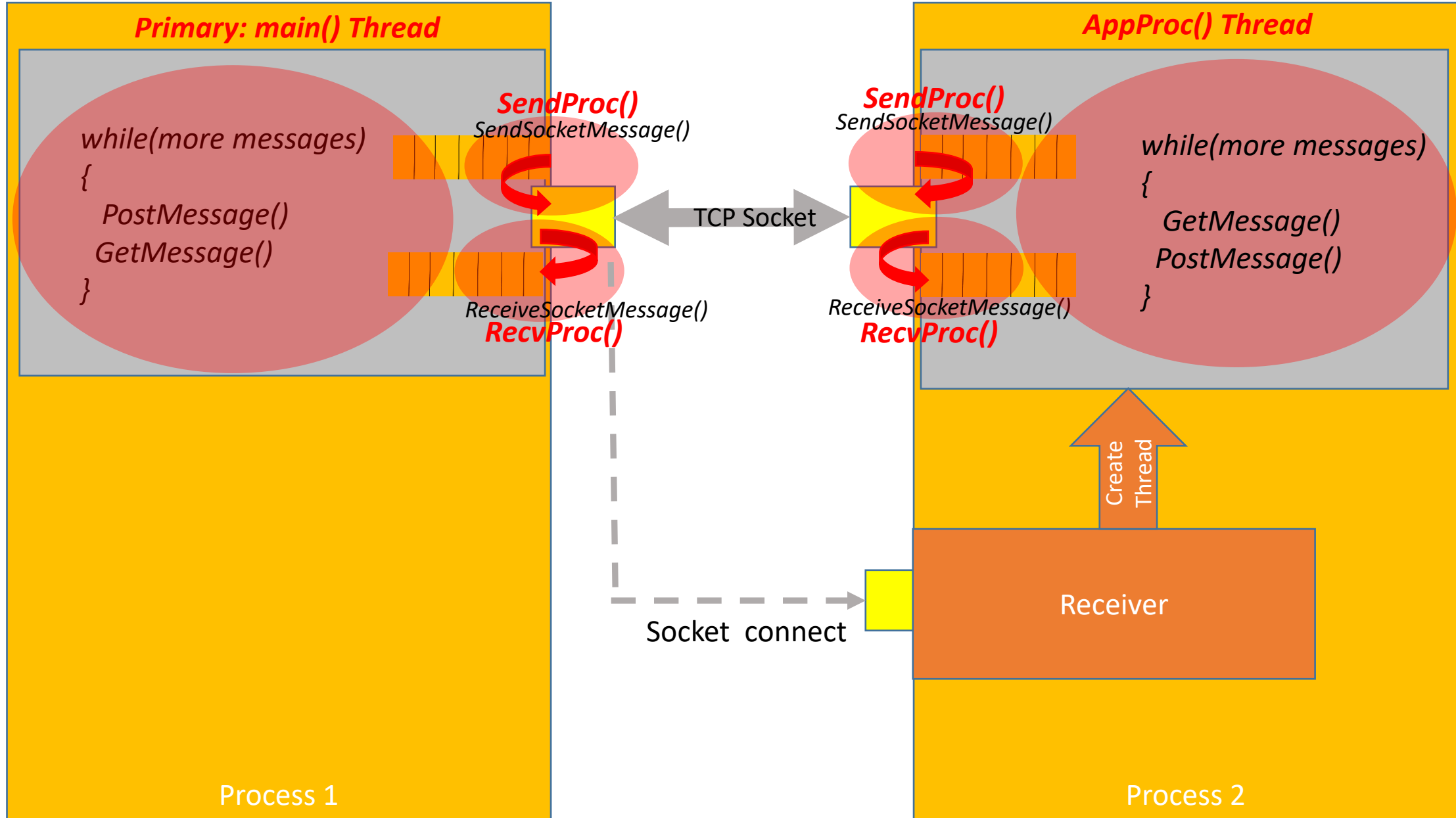
Basis

- Encapsulate details of TCP client/server model
- Abstractions enable lightweight message passing framework
 - Sender - encapsulate TCP client-side processing
 - ClientHandler – encapsulate TCP server-side processing
 - Receiver – Service Host Object for ClientHandler
- Why?
 - A flexible (reusable) messaging pass library
 - Develop complex network/distributed applications in native code (C++)

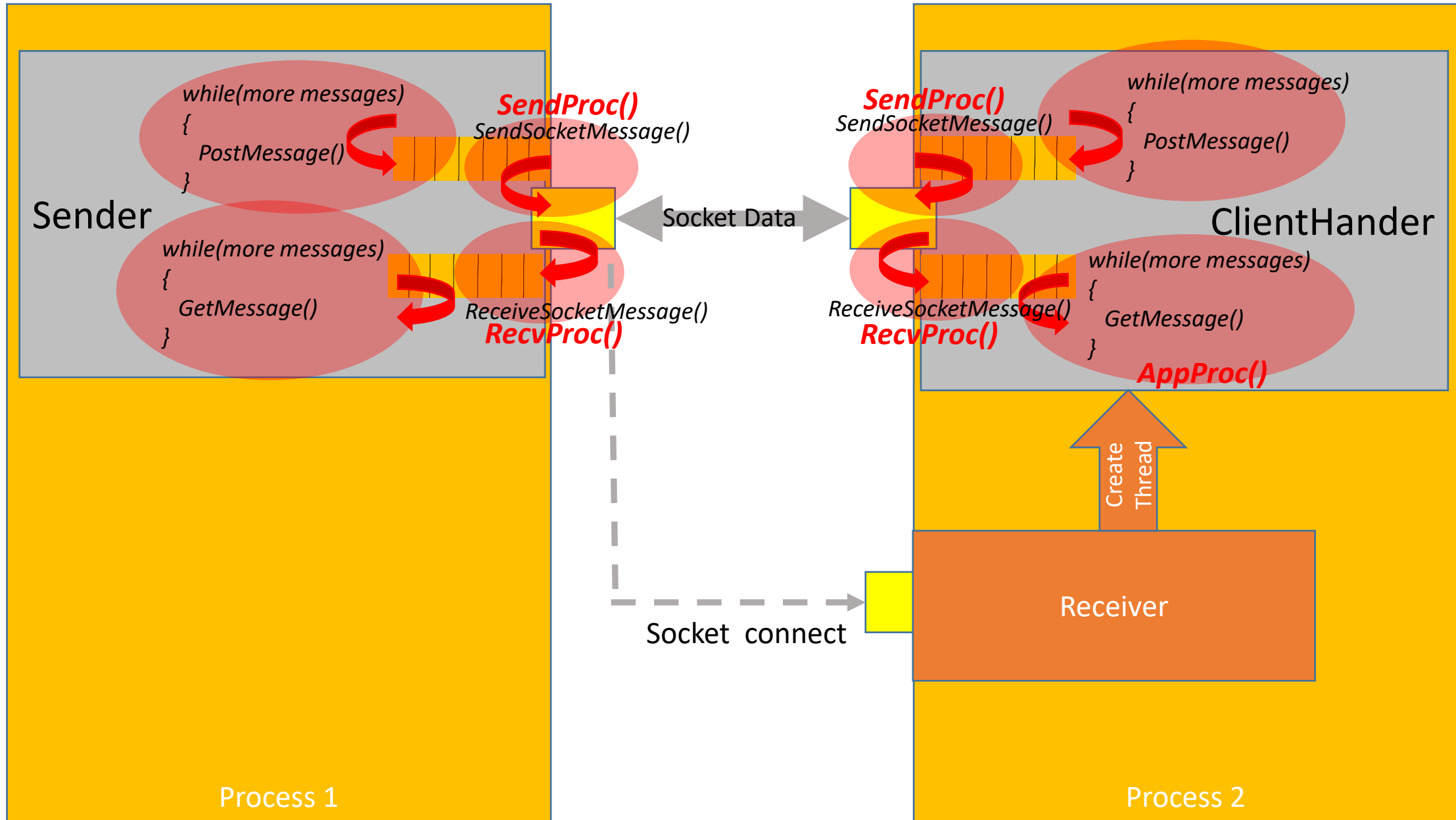
MPL Concept: Half Duplex (Unidirectional) Post/Get



MPL Concept: Synchronous Full Duplex (Bidirectional)



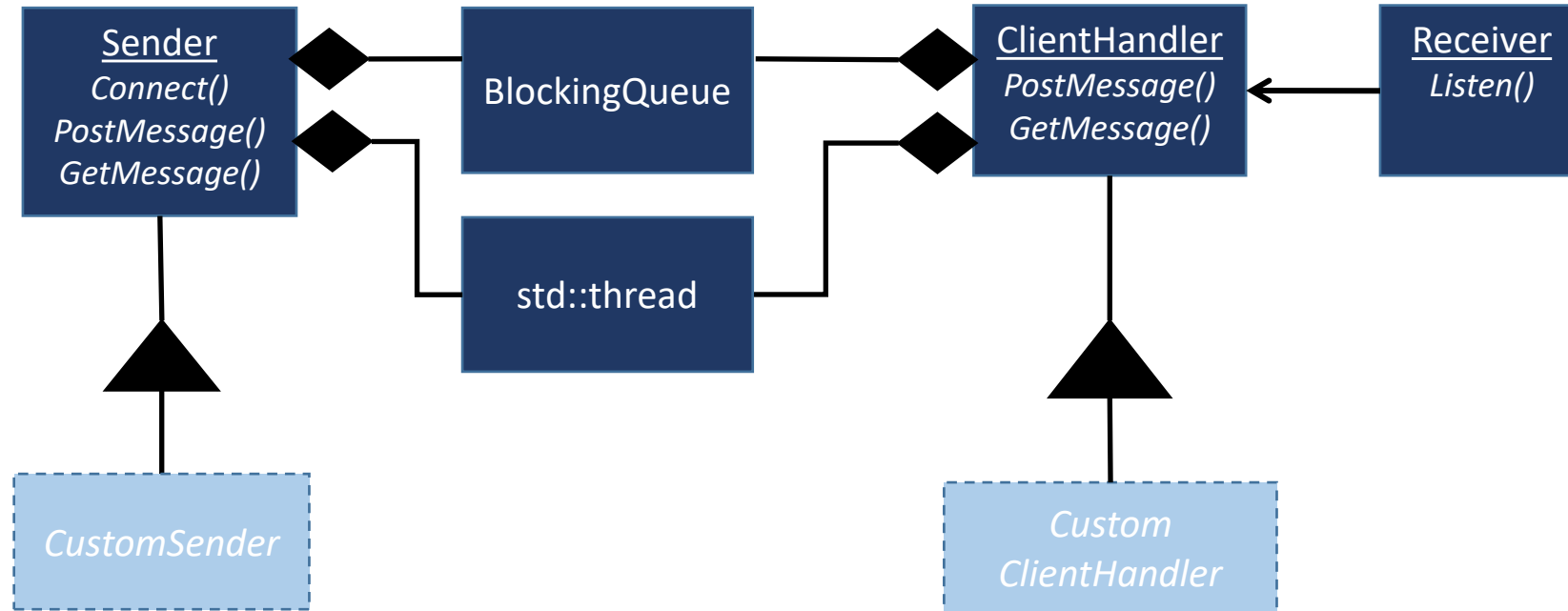
MPL Concept: Asynchronous Full Duplex (Bidirectional)



MPL Specializations

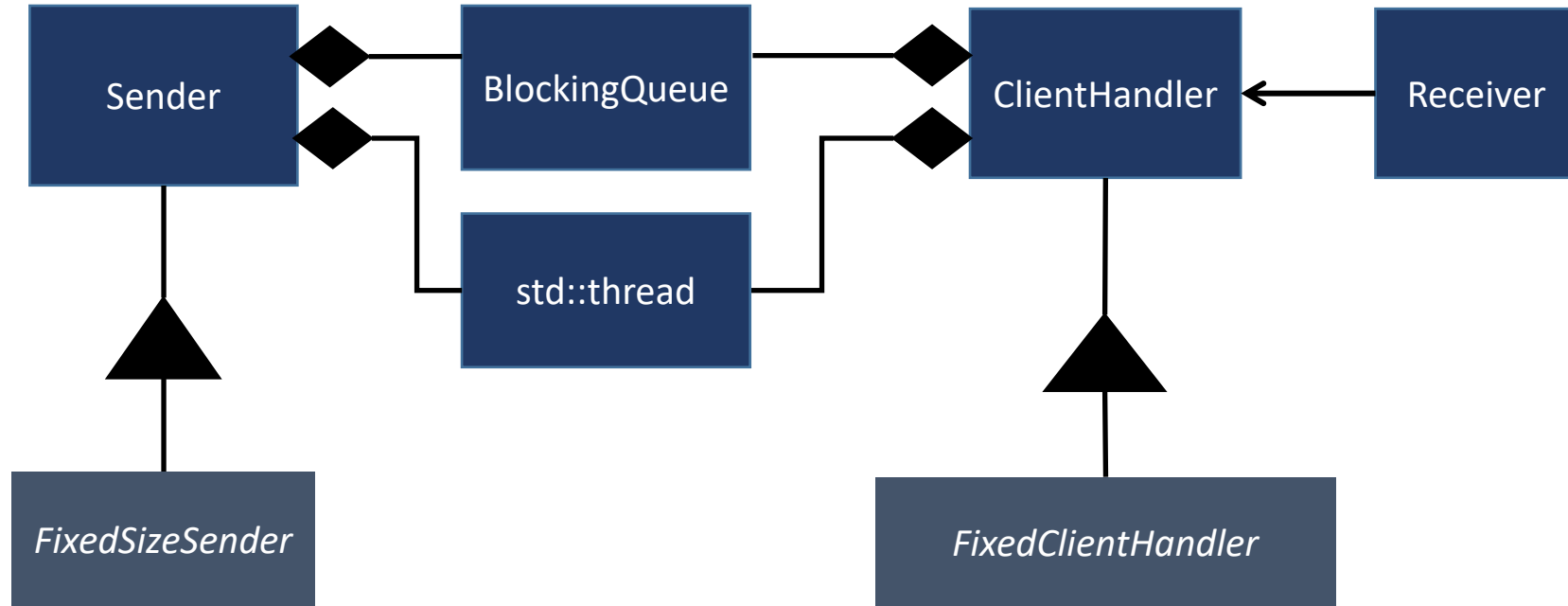
- **Default MPL: variable sized message passing**
 - Simple and flexible
- **Specialization 1: fixed size message passing**
 - Better performance (high throughput applications)
- **Specialization 2: SSL enabled (secure) message passing**
 - Security: based PKI (Public Key Infrastructure)
- **Specialization 3: Emphasize recovery**
 - High reliability applications

Base MPL framework



- TCP Client/Server Model...
 - Client-side and Server-side and not symmetric!
 - Client initiates, Server responds
 - Server listens, client connects
 - Notice *Sender and ClientHandler aren't much different!*

Fixed Size Message MPL



Secure MPL

