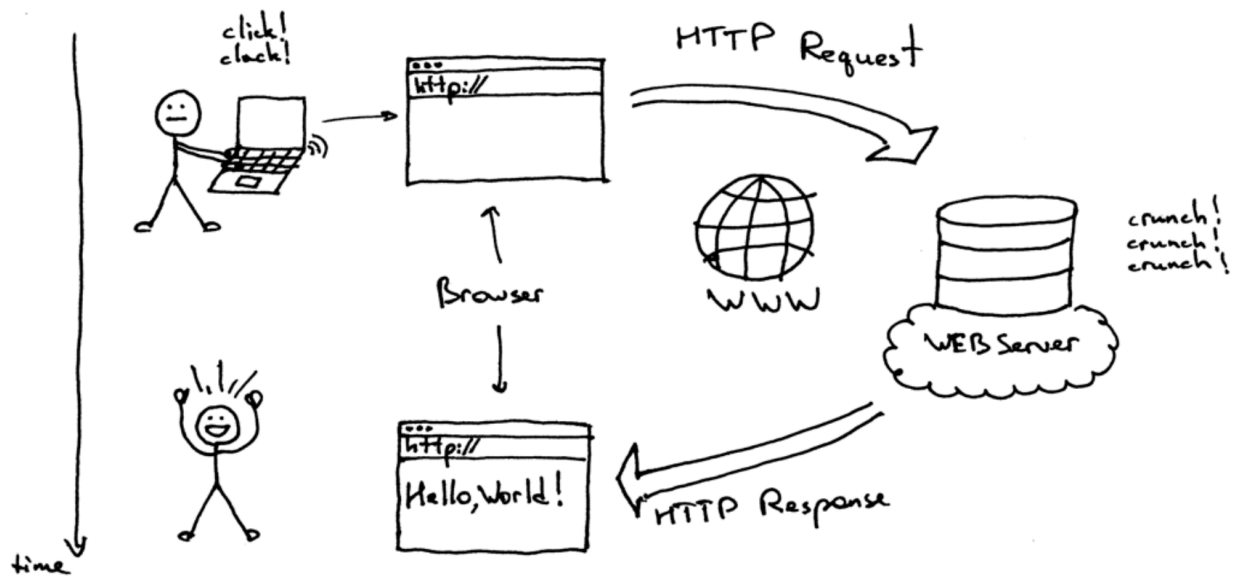


http-server

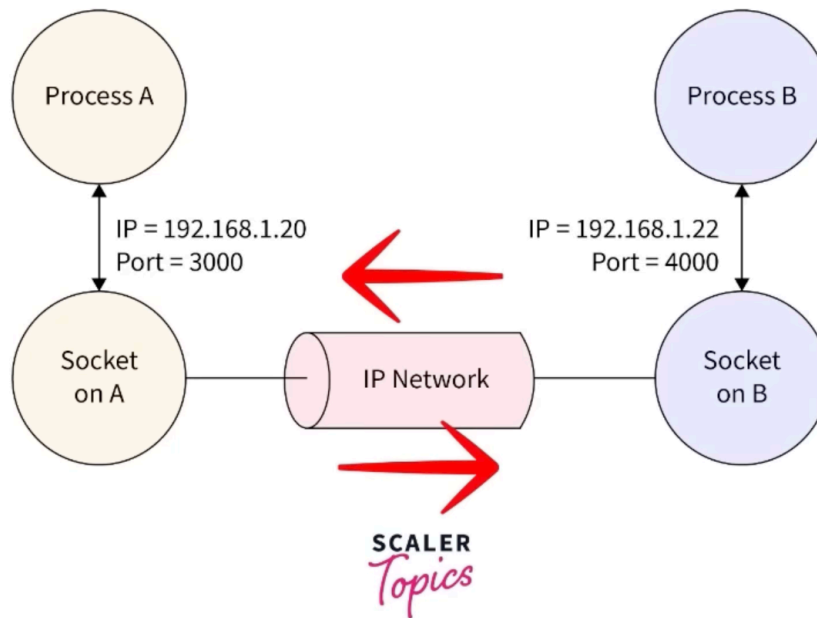
`https://localhost:8888/hello`

- Protocol - http
- Host Name - localhost
- Port - 8888
- Path - hello



Sockets

Sockets are endpoints in a 2-way communication channel.



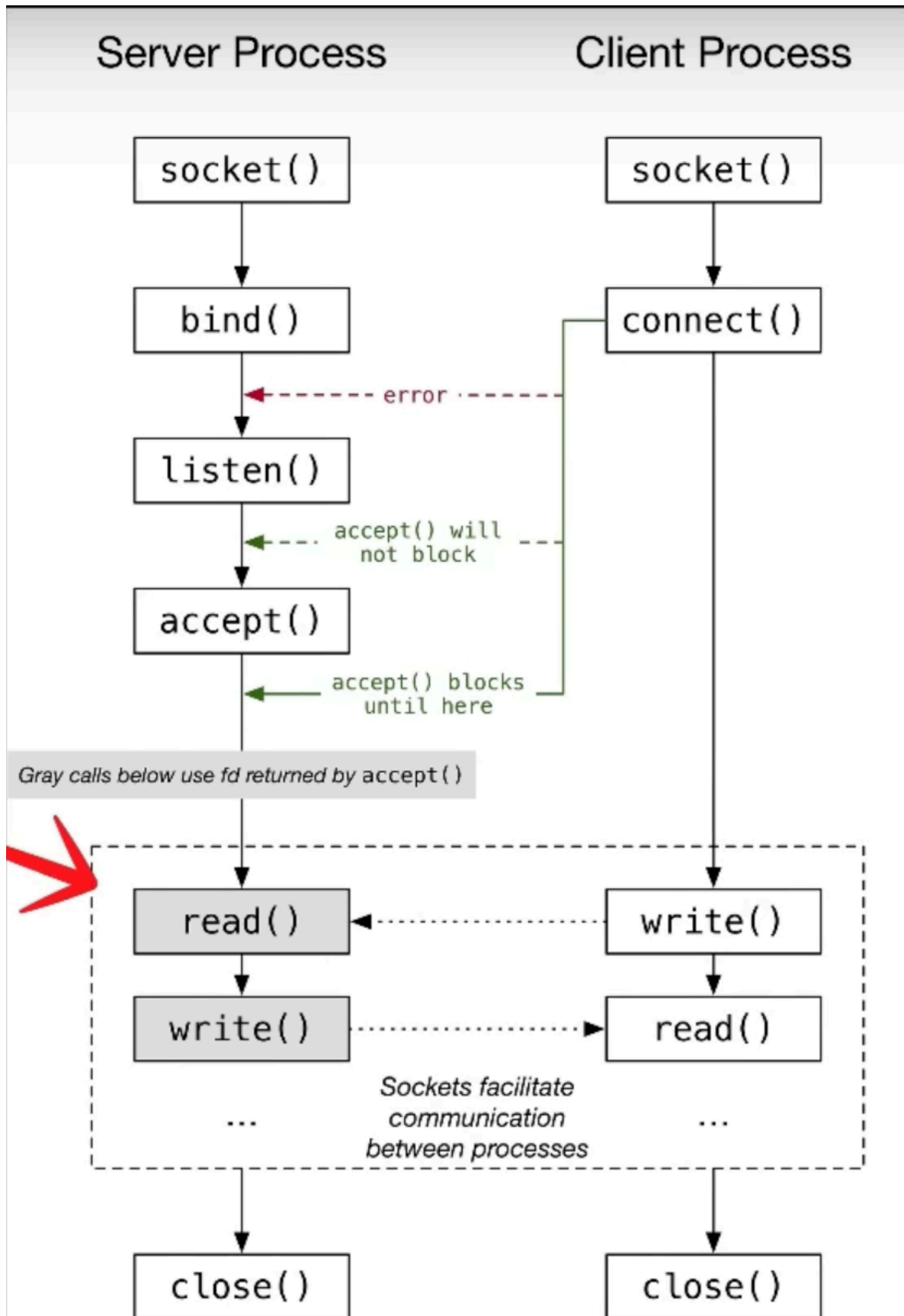
Socket = IP Address + Port Number

IP Address → 192.168.1.255

Port Number → 3389

Socket Address = 192.168.1.255:3389

Socket Lifecycle



- Server Side Socket Lifecycle
 - **Socket Creation**
 - Created using `socket()`
 - Example: `int sockfd = socket(AF_INET, SOCK_STREAM, 0);`
 - **Binding**
 - Bound to IP and port via `bind()`
 - **Listening**
 - Listens for connections using `listen()`
 - **Accepting Connections**
 - Accepts clients with `accept()` , creating a **new socket** for each
 - **Data Exchange**
 - Communicates using `send()` and `recv()`
 - **Closing Connection**
 - Closes sockets using `close()`
- Client Side Socket Lifecycle
 1. **Socket Creation**
 - The client creates a socket using `socket()` .
 2. **Connecting to Server**
 - The client initiates a connection to the server using `connect()` .
 3. **Data Exchange**
 - The client sends and receives data using `send()` and `recv()` .
 4. **Closing Connection**
 - When finished, the client closes the socket using `close()` .

<code>socket()</code>	Reserves a network endpoint with OS
<code>connect()</code>	Initiates a connection to target IP/port
<code>send()</code>	Sends data over the network via that connection
<code>recv()</code>	Reads response data from the network buffer
<code>close()</code>	Tells OS to release the socket

Resources

1. <https://www.youtube.com/watch?v=NvZEE-mZsul>
2. <https://freedium.cfd/https://medium.com/from-the-scratch/http-server-what-do-you-need-to-know-to-build-a-simple-http-server-from-scratch-d1ef8945e4fa>