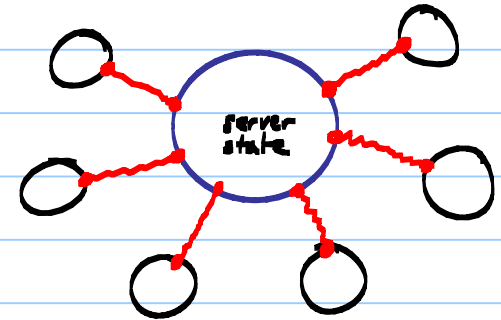


Client-Server Framework (continued)

Note Title

- Want:
- many simultaneous clients
 - Application defines:
 - message types
 - logic for when to send messages
 - handle messages received
 - internal app. state at both clients & server



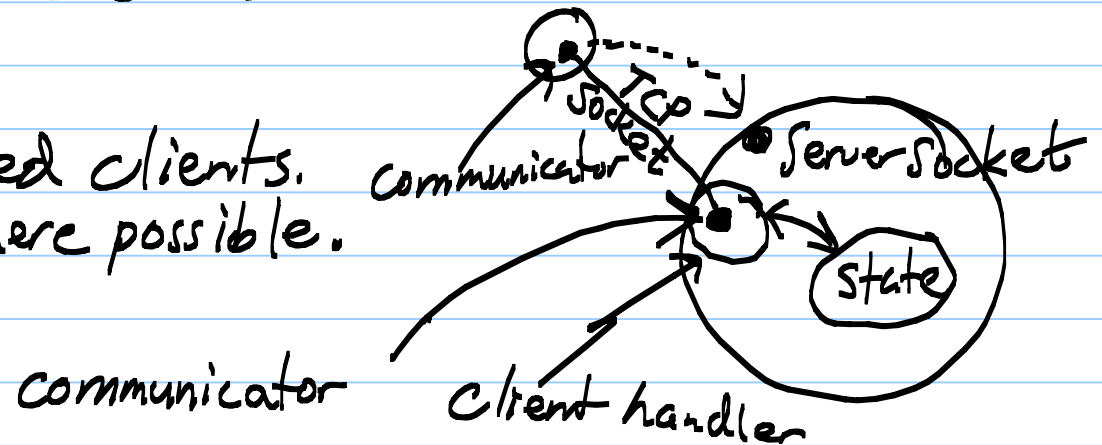
- Framework provides
 - communication set-up (sockets, streams)
 - thread management (at the server)
 - a way to plug in:
 - message types
 - logic
 - internal state
- API for handling sending & receiving of messages

Ideas:

- Reuse comm support at client + server. \Rightarrow Communicator class
- Handle reactive message processing using a listener. \Rightarrow MessageListener interface
- Also allow active message consumption. \Rightarrow receive() method
- Use a factory method to create client handlers at the server. \Rightarrow AbstractServer with createClientHandler
- Provide an abstract server that applications can extend. \Rightarrow

Plus:

- \Rightarrow • Keep track of connected clients.
- \Rightarrow • Handle exceptions where possible.

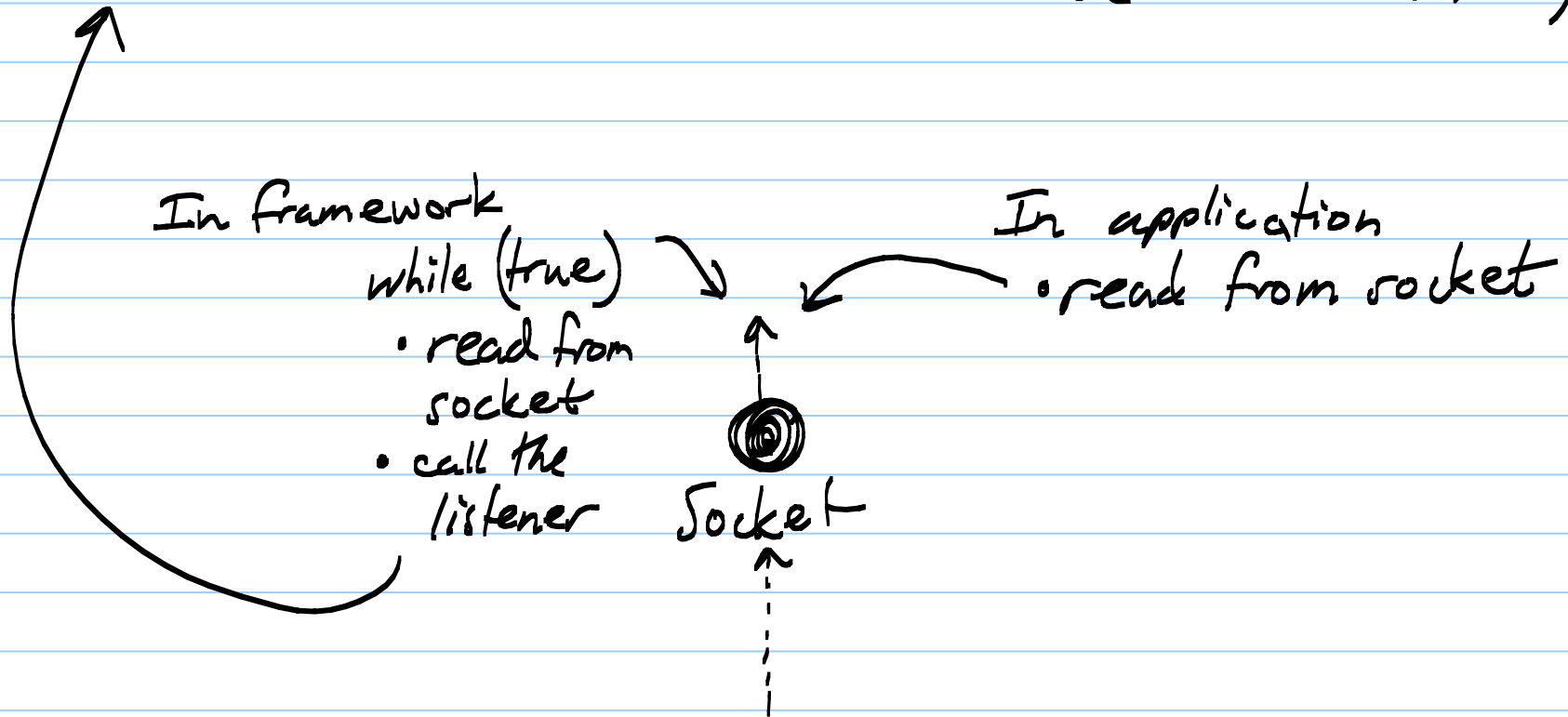


Passive communication

(receive using a listener)

Active comm.

(Ask for messages when we want them.)



Reuse communicator at server & client

↓
built from
a socket that
results from
ss.accept()

↓
needs to
connect to
server

∴ Two constructors