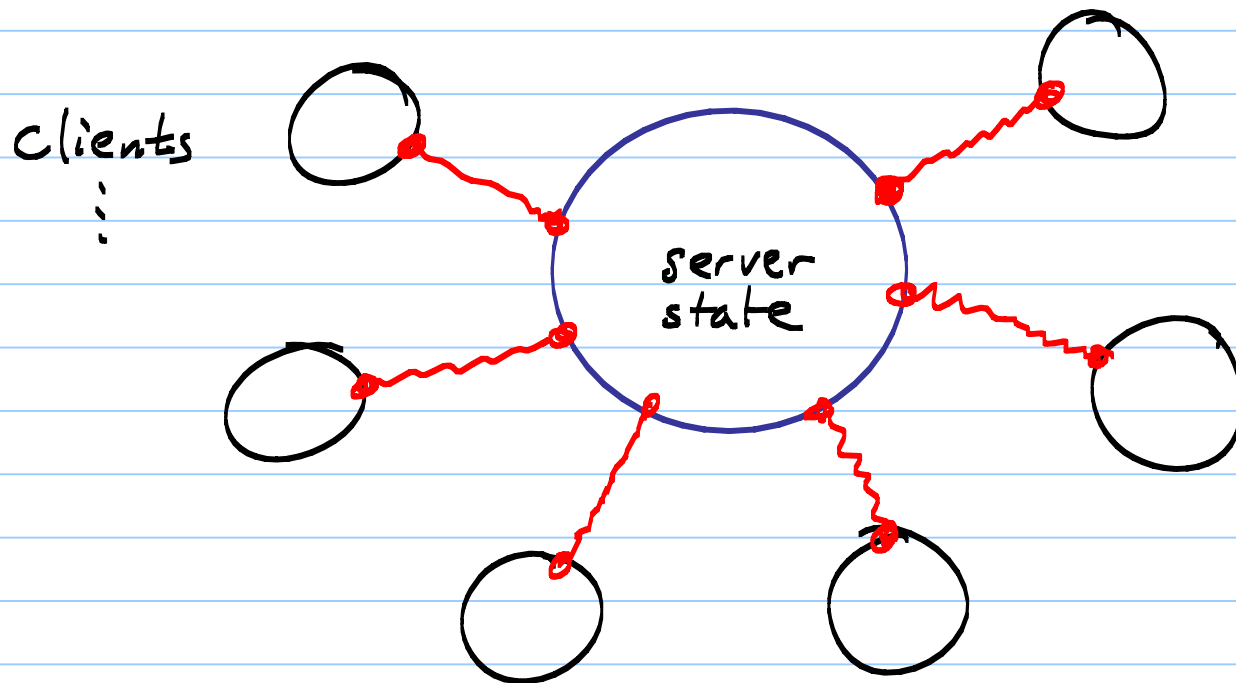


Designing a Multi-User Client-Server Framework

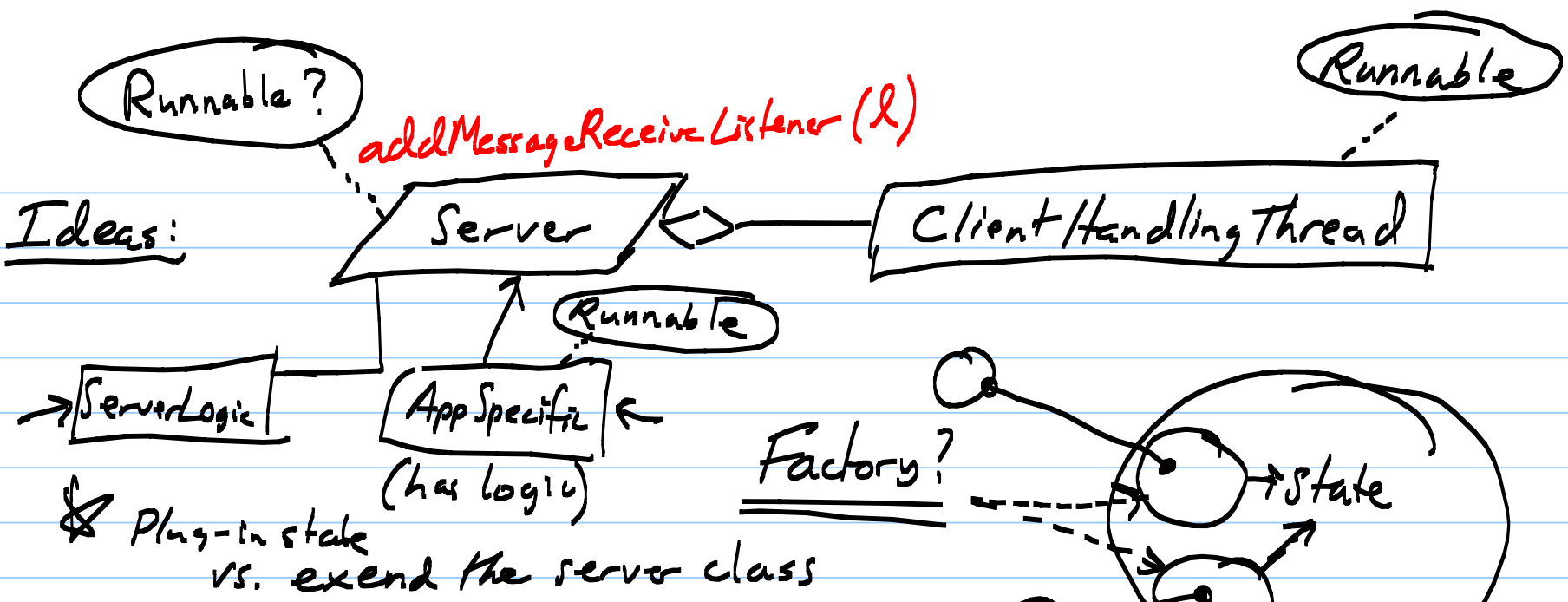
Note Title



- Distributed collaboration, games, social networking
- Distributed computation
- ...

Want:

- many simultaneous client
- 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



? save/load support in API

sendMessage (M...)
receiveMessage (M...)

blocking call
event-driven

Both?

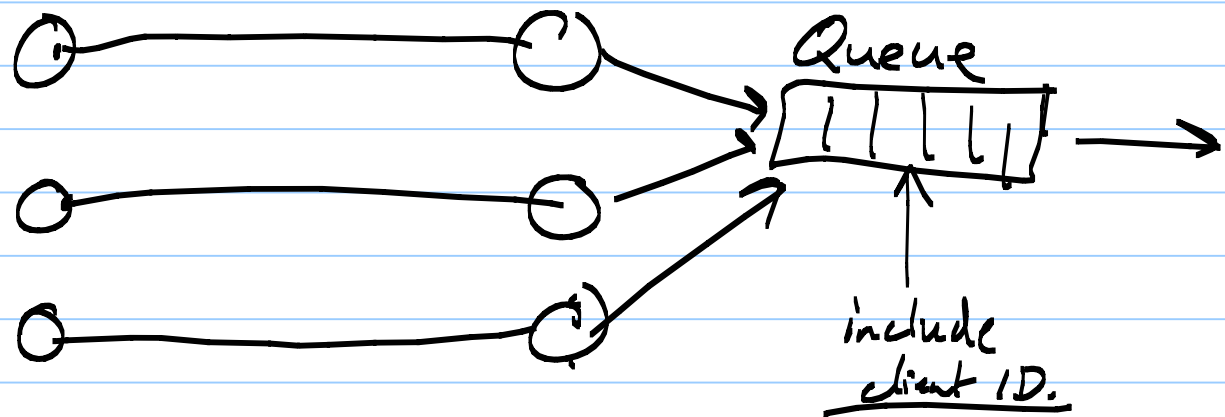
MessageReceiveListener

Appl. 1

• Create our own Queues? - Disadv: overhead
- Adv: Funnel + dispatch

Server Side
Client/Handler

Clients

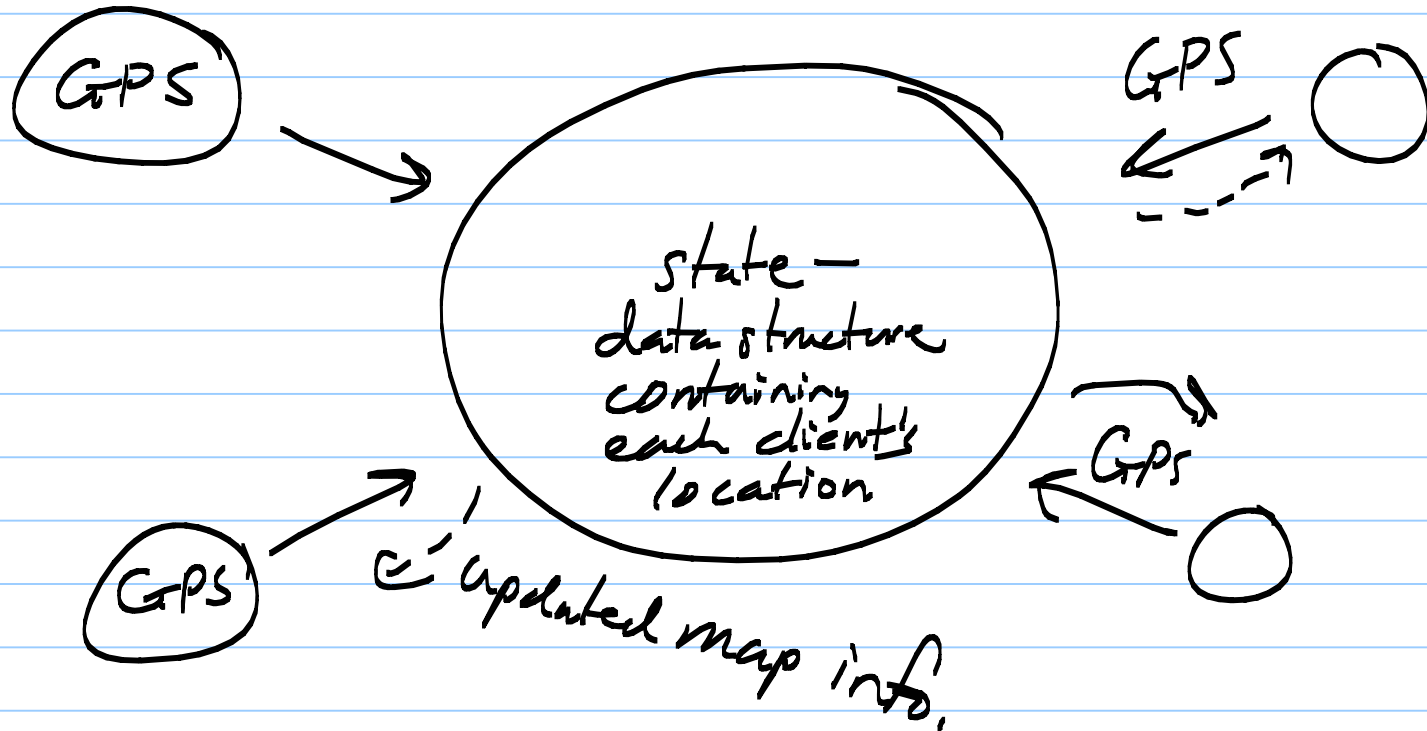


Server Logic

loop to receive
& process messages

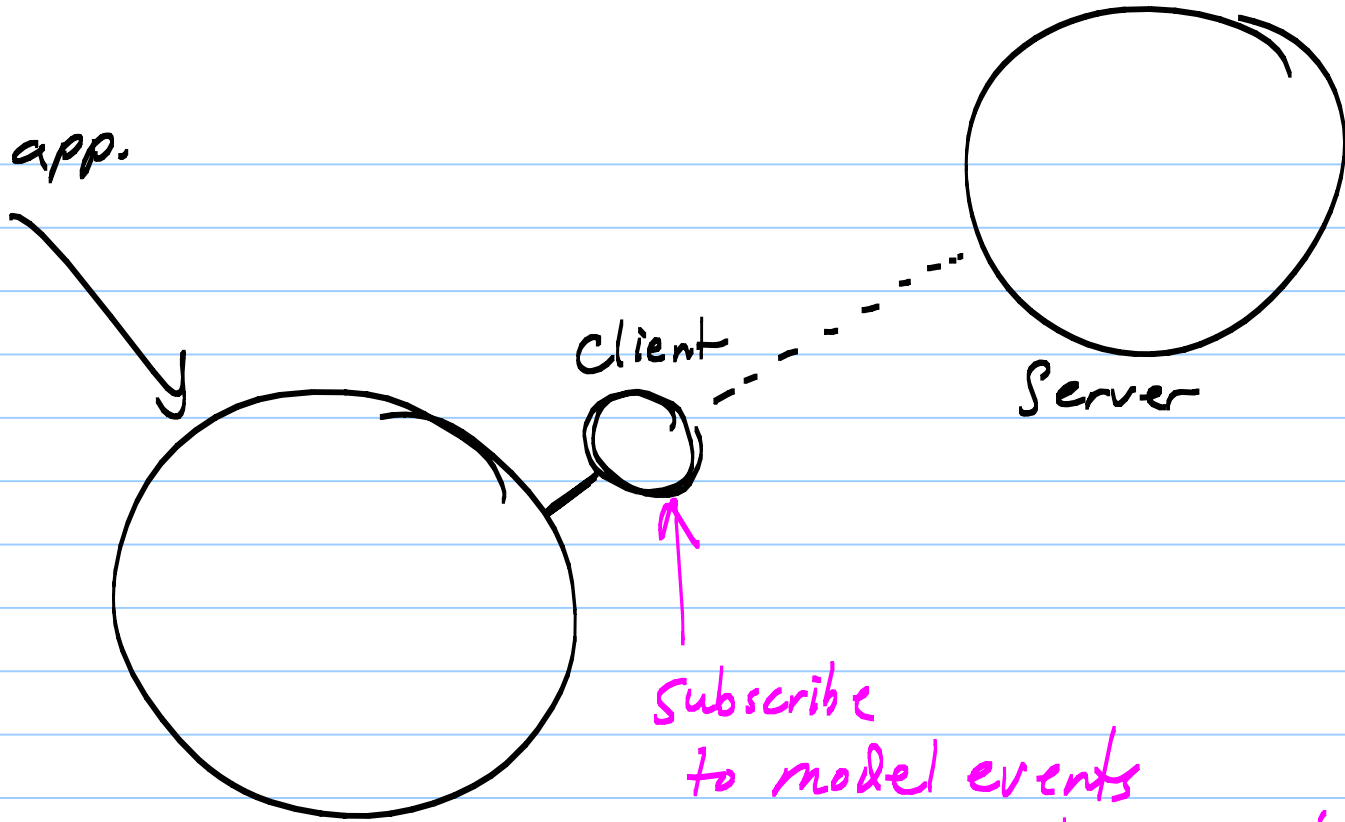
simple logic

Marauder's Map



Message types: NewCoordinates (x,y)
MapUpdate: (-...)

Existing app.



Subscribe
to model events
from existing app's model