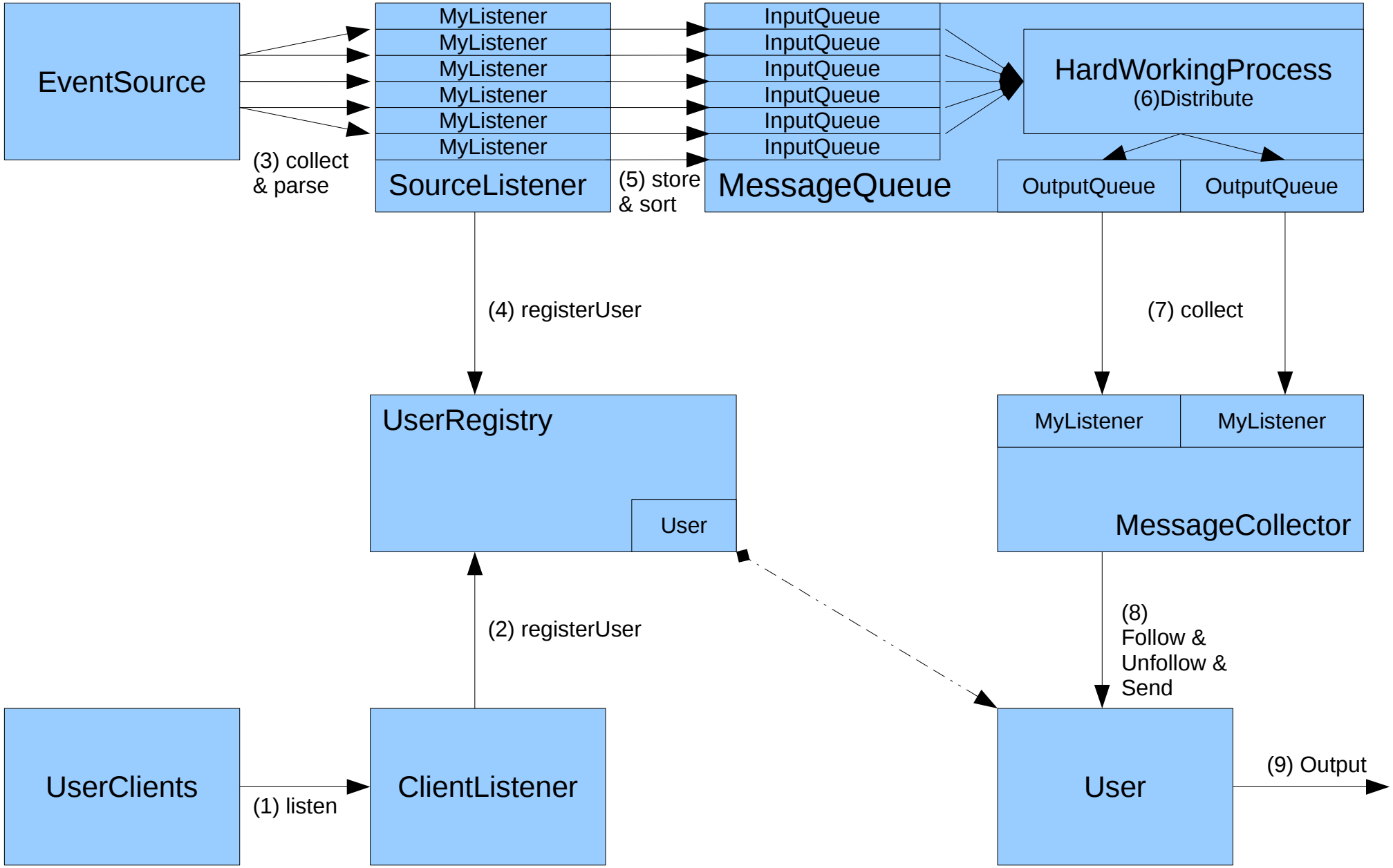
EventSource
(3) collect
& parse
MyListener
MyListener
MyListener
MyListener
MyListener
MyListener
SourceListener
(5) store
& sort
InputQueue
InputQueue
InputQueue
InputQueue
InputQueue
InputQueue
MessageQueue
HardWorkingProcess
(6)Distribute
OutputQueue
OutputQueue
(4) registerUser
(7) collect
UserRegistry
User
MyListener
MyListener
MessageCollector
(2) registerUser
(8)
Follow &
Unfollow &
Send
UserClients
(1) listen
ClientListener
User
(9) Output

1. A listener will collect all incoming user requests.
2. New users will be registered in the user registry.
2a) Every user, that was registered this way owns a socket for outgoing messages.
3. Several listeners will collect & parse all incoming messages.
4. For every unknown target/source a new user will be registered.
4a) Every user, that was registered this way will store messages internally until a socket is open.
5. Every listener will order & store the messages in his own queue (in MessageQueue)
6. A HardWorkingProcess will distribute all messages to OutputQueues.
7. Severall listeners will collect messages from their own queue (in MessageQueue)
8. Follow/Unfollow operations will be performed and the message will be send out or stored
9. If a socket for this user was open, the message was send out.

Start in folder "./bin": java soundcloud.server.SoundCloudStarter <args>

Command Line Options:
-mct n
    Number of threads in SourceListener
-slt n
    Number of threads in MessageCollector
-uport n
    Port for user requests
-eport n
    Port for event messages