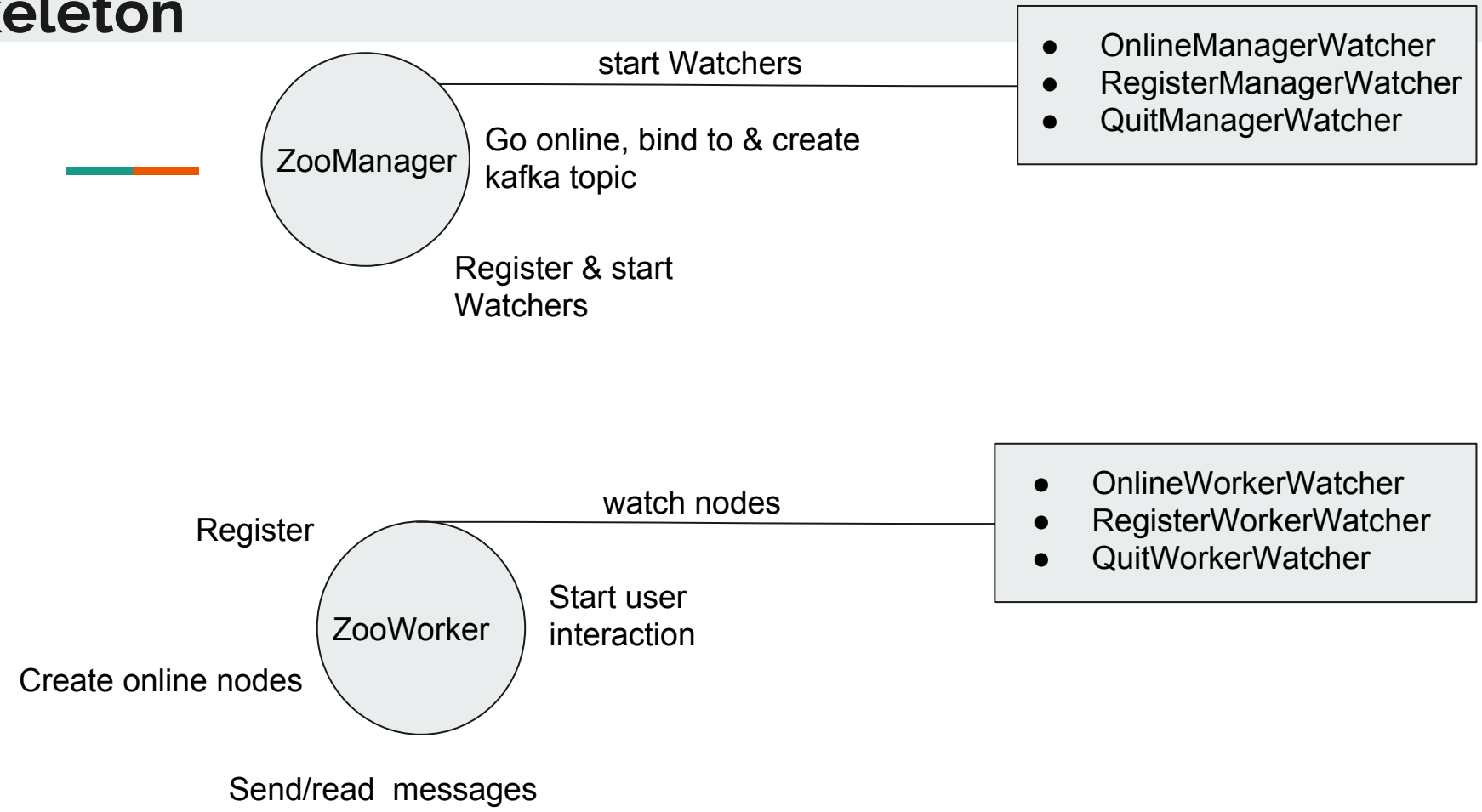
Large scale systems project

Ettore Puccetti, Nicolae Righeriu

Distributed messaging system, where users can:

- Register and connect to the system
- 1 Manager (m), n writers (W)

Skeleton


start Watchers
ZooManager
Go online, bind to & create kafka topic
Register & start Watchers
OnlineManagerWatcher
RegisterManagerWatcher
QuitManagerWatcher
watch nodes
Register
ZooWorker
Start user interaction
Create online nodes
Send/read messages
OnlineWorkerWatcher
RegisterWorkerWatcher
QuitWorkerWatcher

User interaction

1 - Register

2 - go online

3 - quit

4 - send messages

5 - see topics

6 - read messages

Action: Register

- M: watch the children of /request/enroll
- W: create and watch /request/enroll/w_id:-1
- M: create /registry/w_id
  - If create success M: set /request/enroll/w_id:1
  - If user already registered M: set /request/enroll/w_id:2
  - If create fails M: set /request/enroll/w_id:0
- W: check the watched event data, if 1 or 2 => successful registered.
- W: remove /request/enroll/w_id

Action: Quit

- M: watch the children of /request/quit
- W: create and watch /request/quit/w_id:-1
- M: delete /registry/w_id
    - If delete success M: set /request/quit/w_id:1
    - If user was not registered M: set /request/quit/w_id:2
    - If delete fails M: set /request/quit/w_id:0
- W: check the watched event data, if 1 or 2 => successful deleted.
- If success:
    - M: delete kafka topic associated with W_id
- W: remove /request/quit/w_id

Action: Go online

- M: watch the children of /online
- W: create ephemeral node /online/w_id
- M&W: check if /registry/w_id exists.
- M: if w_id registered and first time online create /topic/w_id in kafka
- Sender
  - W: publish messages to /topic/w_id/msg:content
- Receiver
  - W: consume messages from /topic/w_id/

Action: Go offline

- M: watch the children of /online
- W: close connection.

Demo