https://github.com/soggie/naws

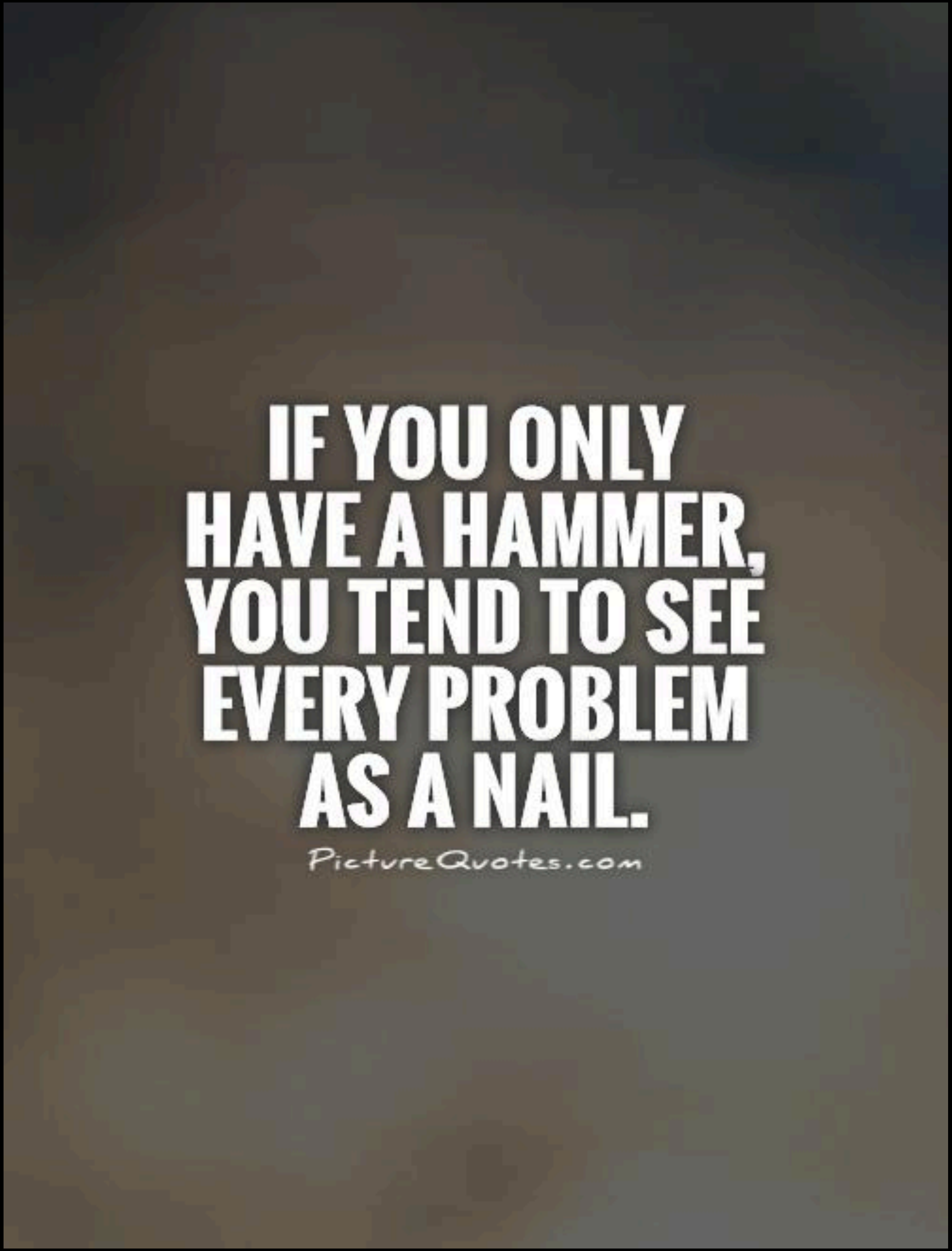
Leveraging ZeroMQ for Node.js

Another tool in your toolbox

About Me

- Ruben Tan
- @roguejs (twitter)
- Works in GrabTaxi (Malaysia)

github.com/soggie/naws



IF YOU ONLY
HAVE A HAMMER,
YOU TEND TO SEE
EVERY PROBLEM
AS A NAIL.

ZeroMQ

- “zero” as in none, nada, nil, 無
- MQ as in Message Queue

Why call ZeroMQ as so when it is not a message queue?

I know. Wrong meme. Shhh.

- Originally conceived as a stock trading messaging solution
- Moved on to generalizing its purpose
- Adopted BSD Socket API and semantics
- Smart end-point, dump network

- ZeroMQ is a networking library with **message queue semantics**
- Used primarily for connectivity between services
- Can be used to facilitate **brokerless** message passing **topologies**

Crash Course!

- Socket basics
  - Buffering (high watermark)
  - Load balancing (outgoing/incoming strategy)
- PUSH socket
- PULL socket
- PUB socket
- SUB socket
- We won’t cover: REQ-REP, XPUB-XSUB, ROUTER-DEALER, STREAM, PAIR, as well as DEVICES

0MQ Sockets

- Supports:
  - inproc, IPC, TCP, pgm, epgm
- Maintains it’ s own internal queue
  - **High watermark** controls buffer size
- Has a **MUTE** strategy
- Has different **INCOMING** and **OUTGOING** strategies

MUTE

- Connection states:
  - connect() ==> READY
  - bind() ==> MUTE
  - accept() ==> READY
- 2 kinds of MUTE state: block or drop

BLOCK

DROP

Exhibit A - REQ-REP

https://github.com/soggie/naws/tree/master/test/talk/exhibit-a.js

-
  ```
  // Bind the REP socket
  reply.bind('inproc://exhibit-a');
  ```

-
  ```
  // Make the REQ socket connect to the REP socket
  request.connect('inproc://exhibit-a');
  ```

-
  ```
  request.on('message', function (msg) {
      console.log('---- recv: ' + msg.toString());
  });
  ```

-
  ```
  reply.on('message', function (msg) {
      console.log('-- recv: ' + msg.toString());
  ```

-
  ```
      work(); // Some heavy duty work
  ```

-
  ```
      reply.send(msg);
      console.log('--- send: ' + msg.toString());
  });
  ```

-
  ```
  setInterval(function () {
      var msg = color(ran(1000, 1000) + '');
      console.log('- send: ' + msg);
      request.send(msg);
  }, 1000);
  ```

PUSH

- INCOMING:
  none
- OUTGOING:
  round-robin
- MUTE:
  blocks

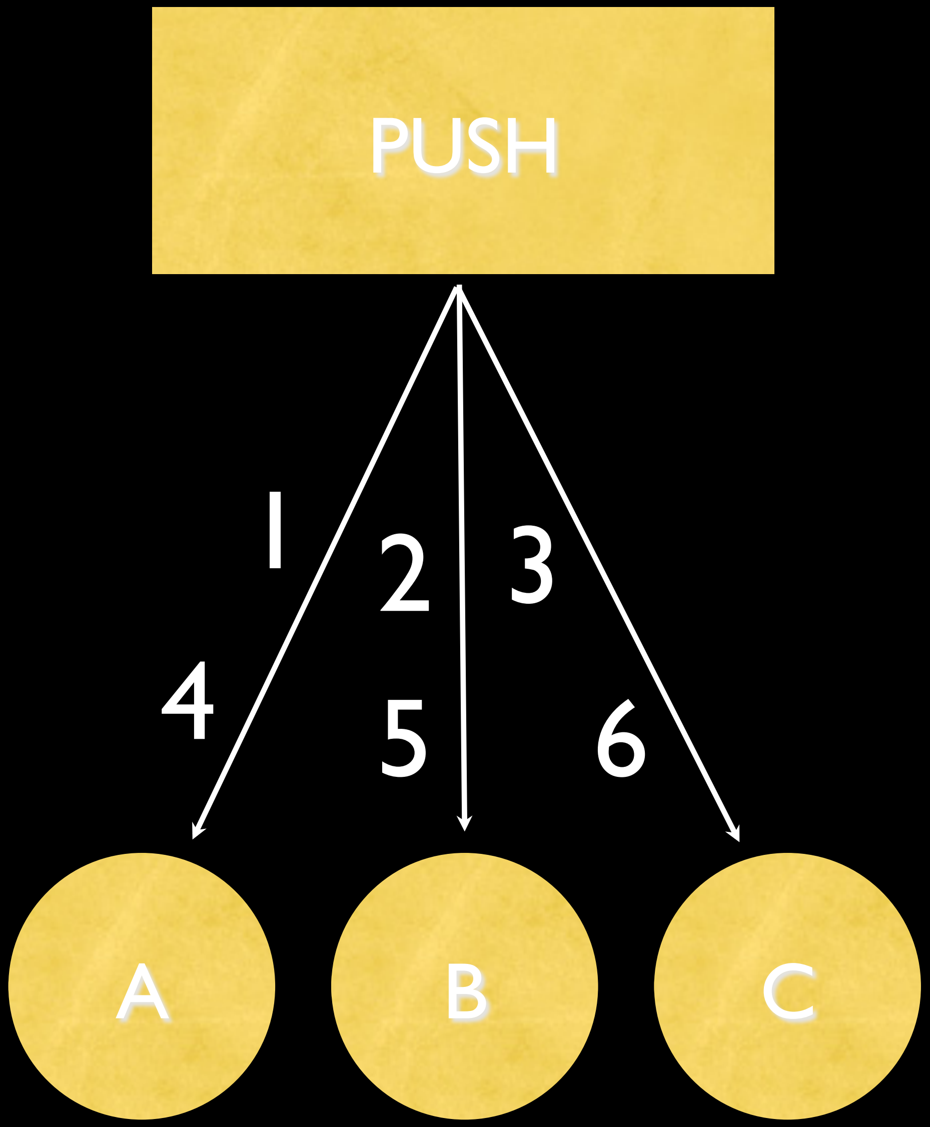


Exhibit B

https://github.com/soggie/naws/tree/master/test/talk/exhibit-b.js

PULL

- INCOMING
  fair-queue
- OUTGOING
  none
- MUTE
  block (??)

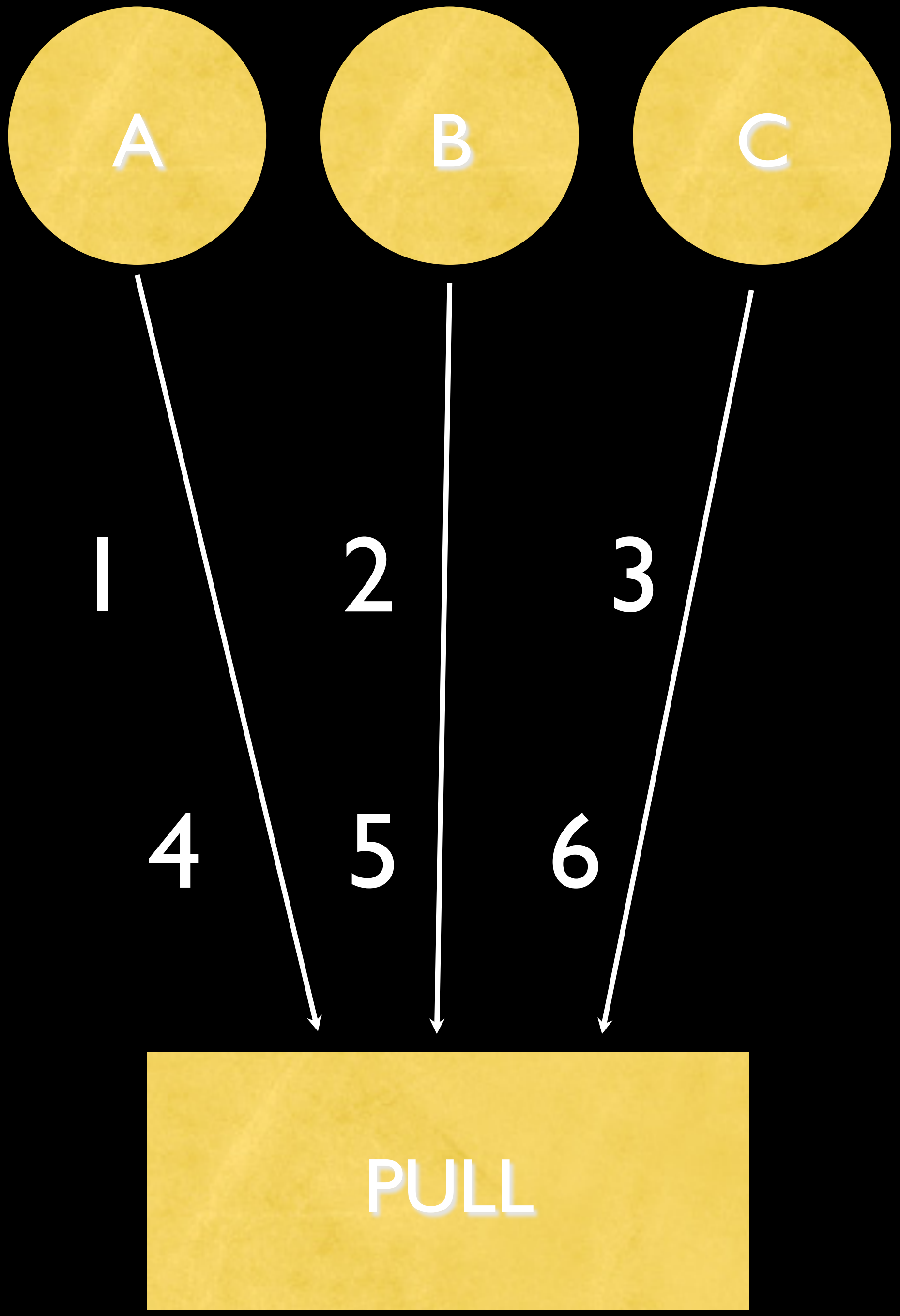


Exhibit C

https://github.com/soggie/naws/tree/master/test/talk/exhibit-c.js

PUB

- INCOMING

  none
- OUTGOING

  fan-out
- MUTE

  drop

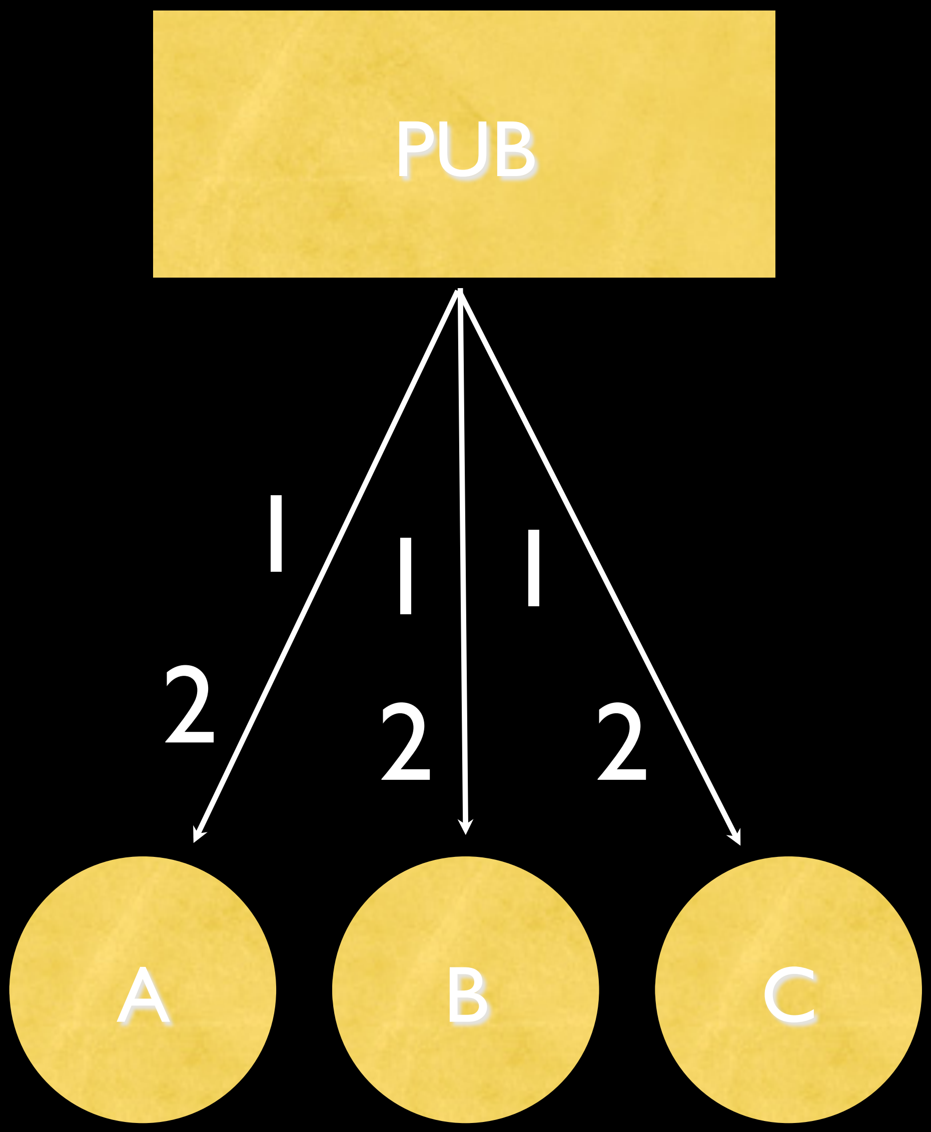


Exhibit D

https://github.com/soggie/naws/tree/master/test/talk/exhibit-d.js

SUB

- INCOMING

  fair-queue
- OUTGOING

  none
- MUTE

  none

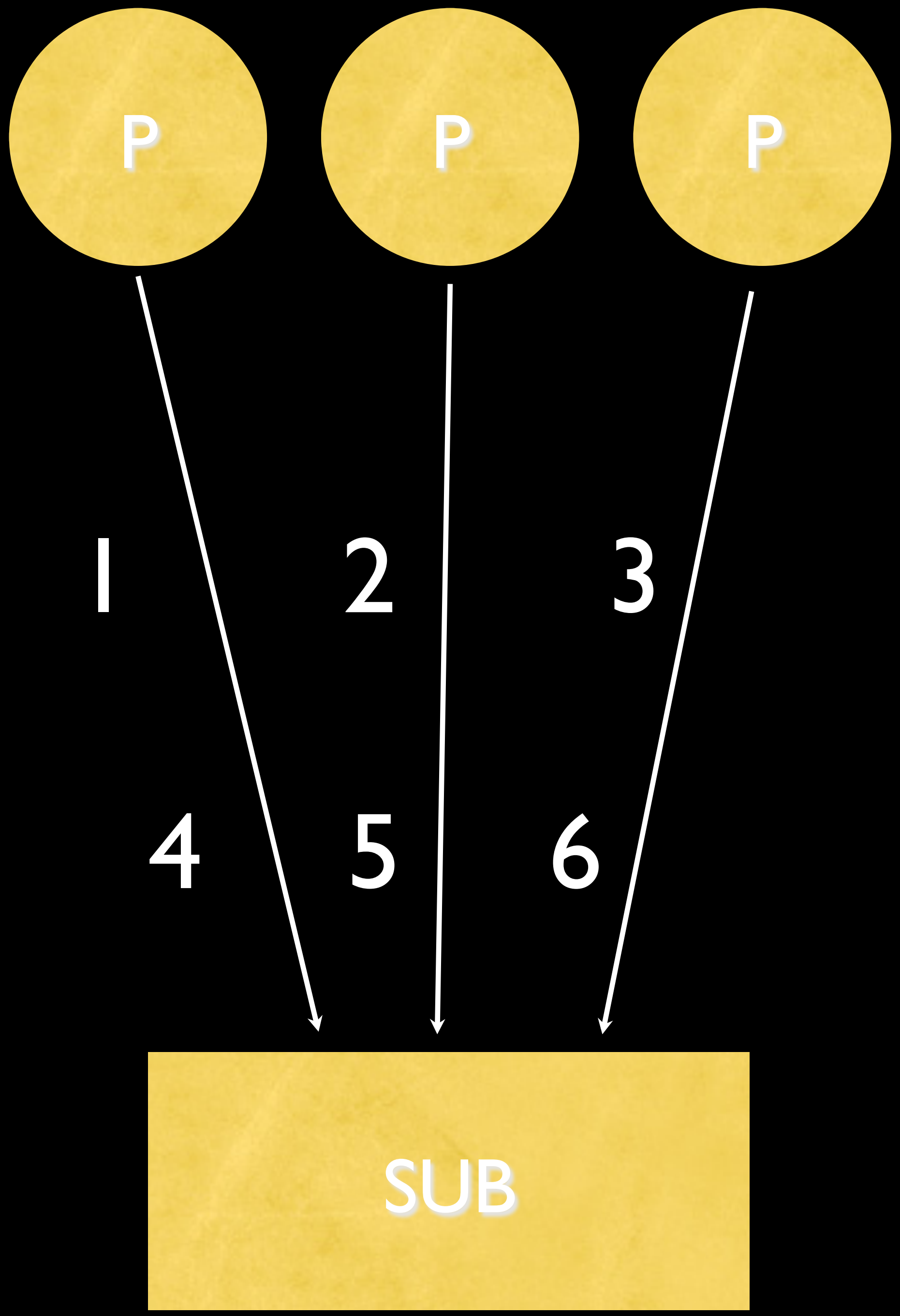


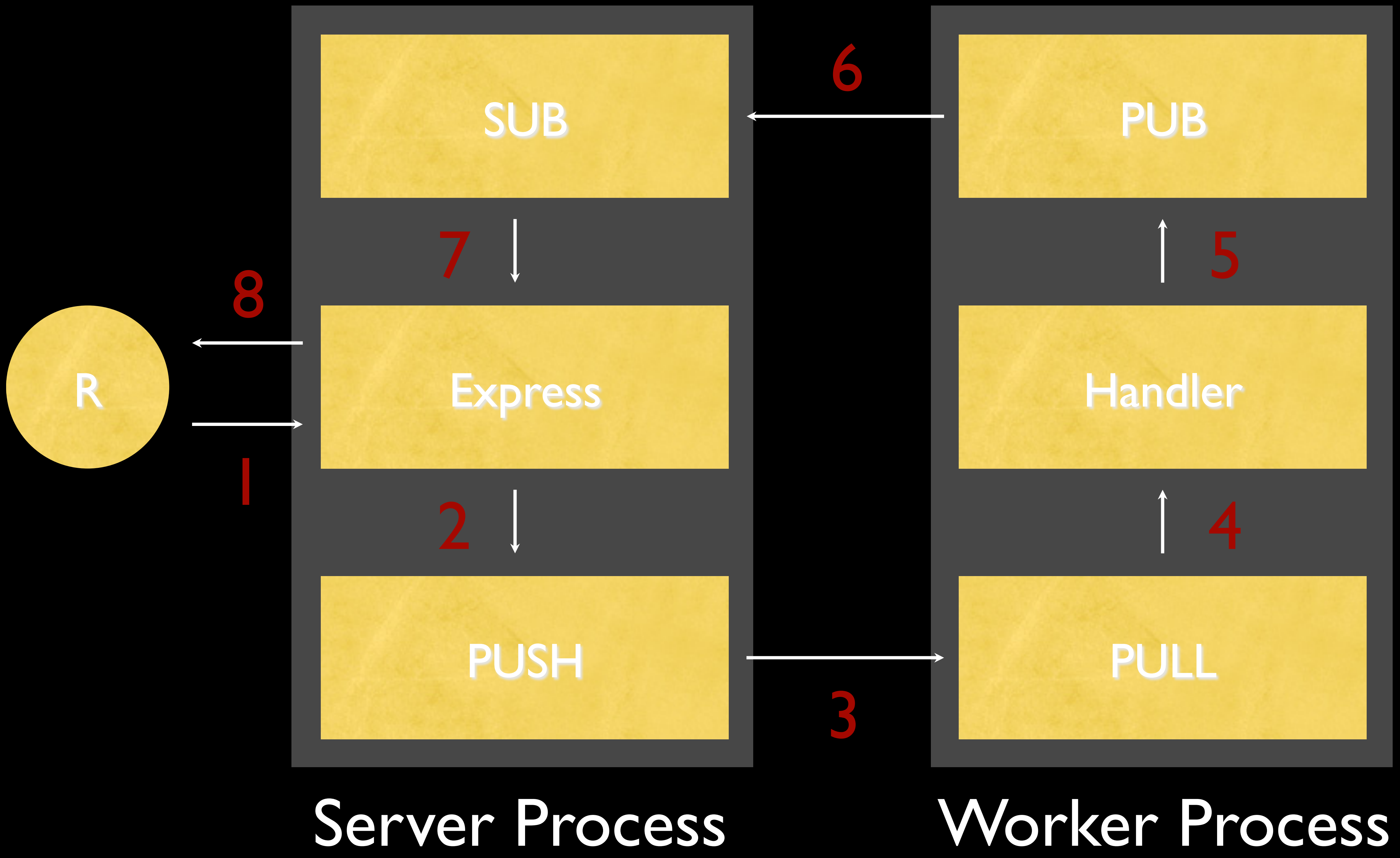
Exhibit E

https://github.com/soggie/naws/tree/master/test/talk/exhibit-e.js

Let’s Build Something

- A simple HTTP web server on top of express
- Uses worker processes to handle requests
- Can add or kill workers any time
- Rudimentary monitoring and reliability

Architecture



Conclusion

- zeromq is a lightweight networking library
- build messaging topologies you want
- nice async features enabling you to do fun stuff (add/kill workers)
- use to connect apps written in different languages/tech stacks
- a very useful tool to have and understand

Similar & References

- Nanomsg (http://nanomsg.org)
  - Martin Sustrik, fork/rewrite of 0mq
- Axon (https://github.com/visionmedia/axon)
  - Pure node.js 0mq sockets
- Theory of ZeroMQ
  http://250bpm.com/concepts
- ZeroMQ Internal Architecture
  http://www.aosabook.org/en/zeromq.html
- Dissecting Message Queues
  http://www.bravenewgeek.com/dissecting-message-queues/

THE END

Questions?