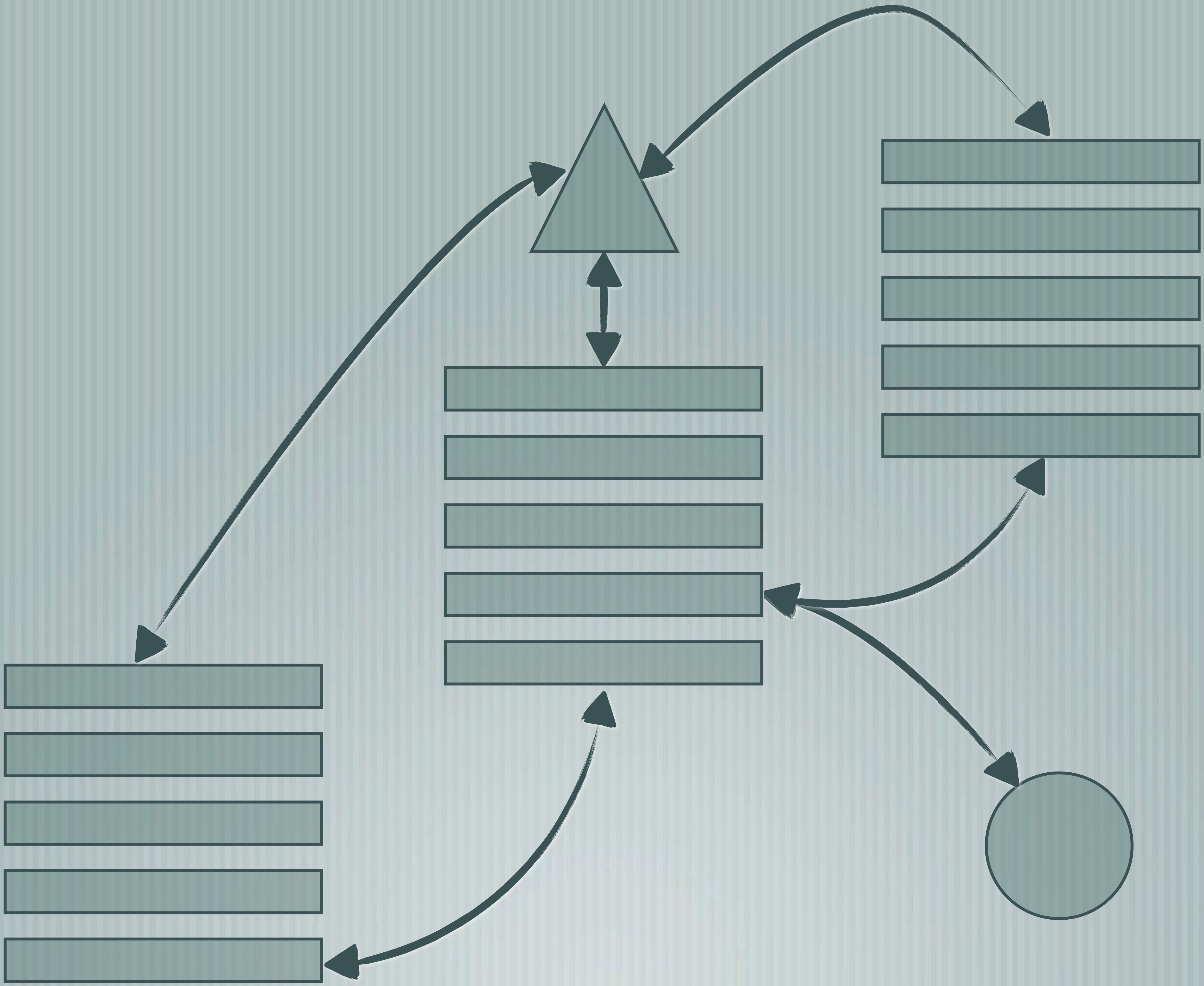
ØMQ

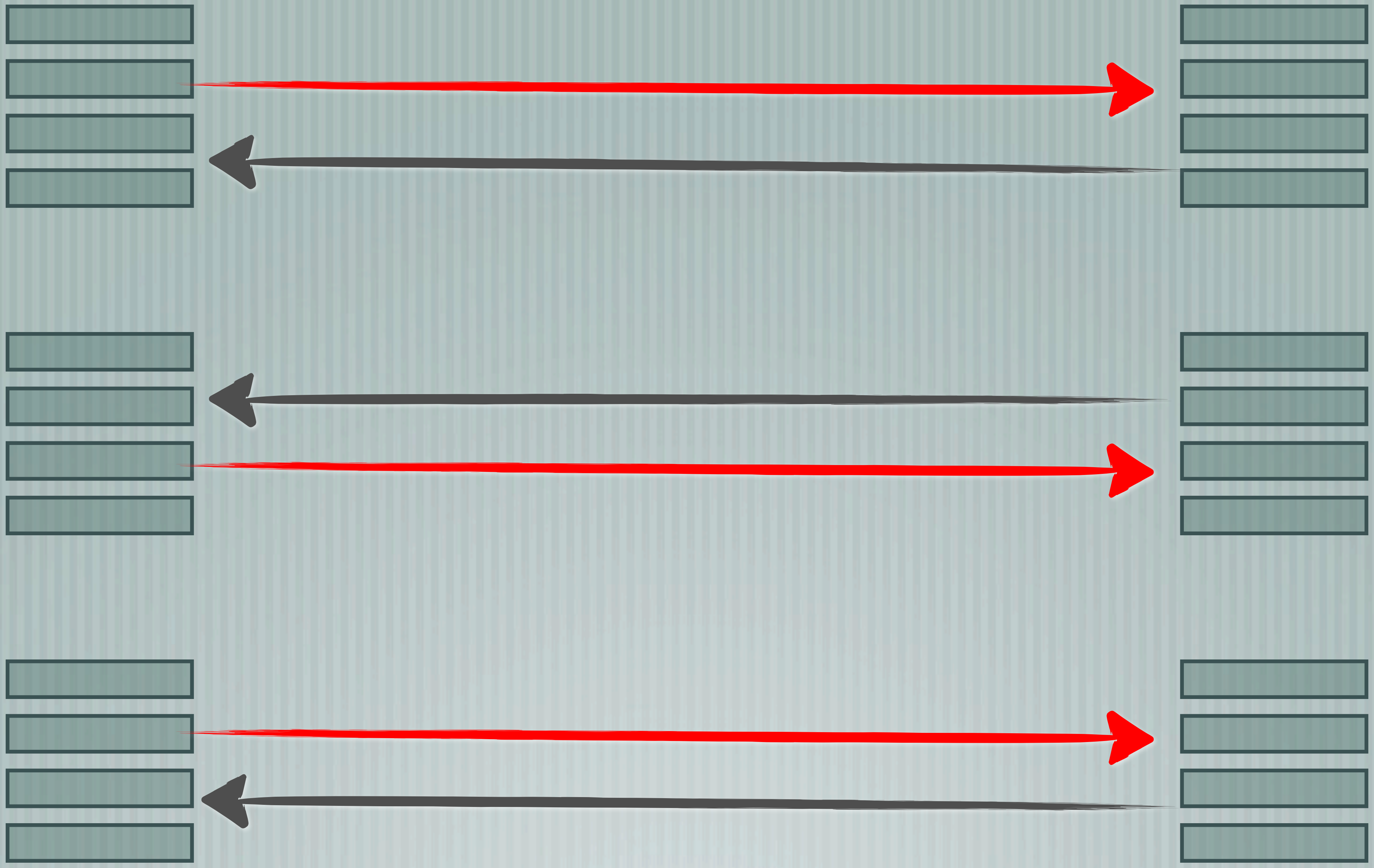
messaging made simple

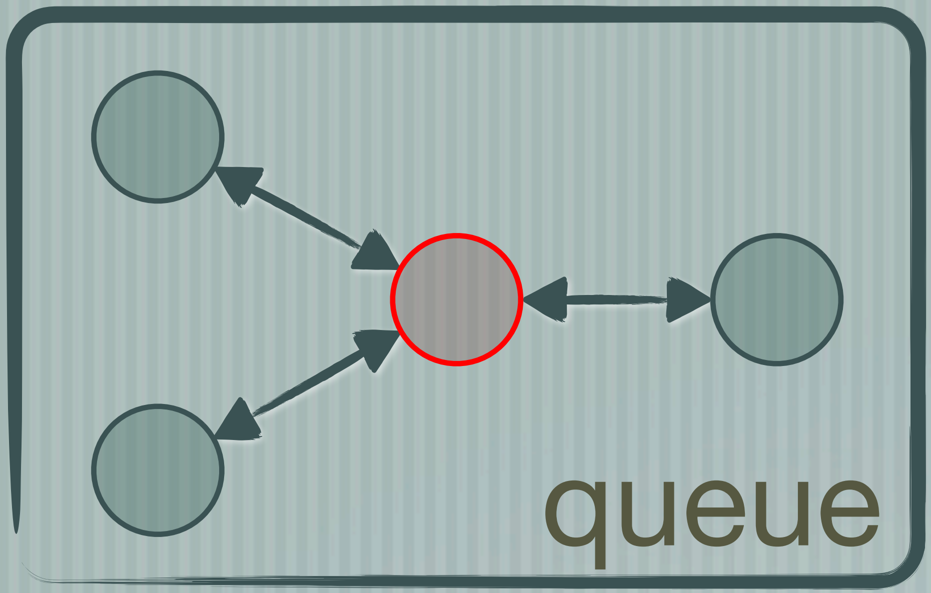
Ian Barber
http://phpir.com - http://zero.mq/ib
ianb@php.net - @ianbarber

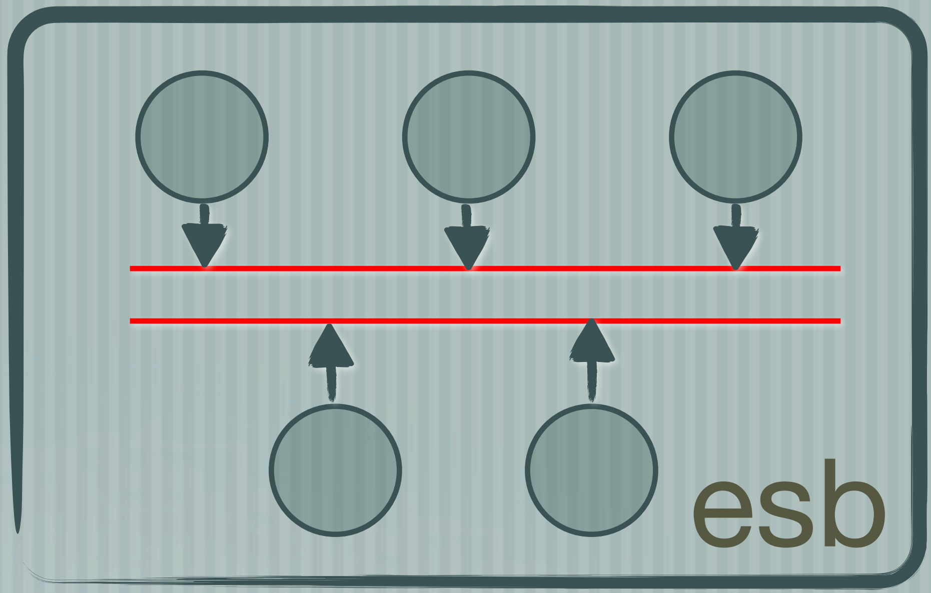
“0MQ is unbelievably cool – if you haven’t got a project that needs it, make one up”

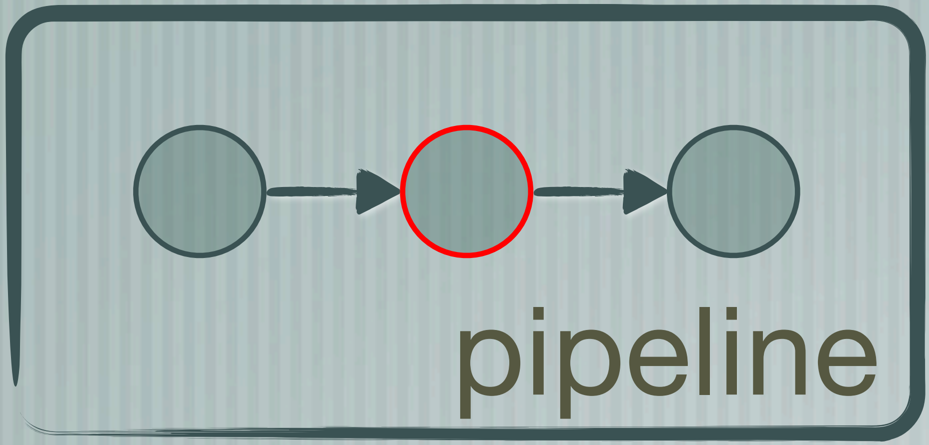
jon gifford - loggly

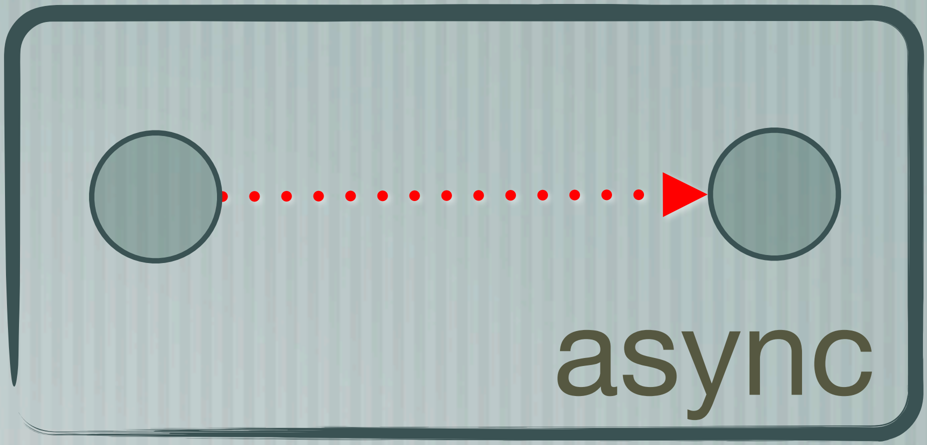


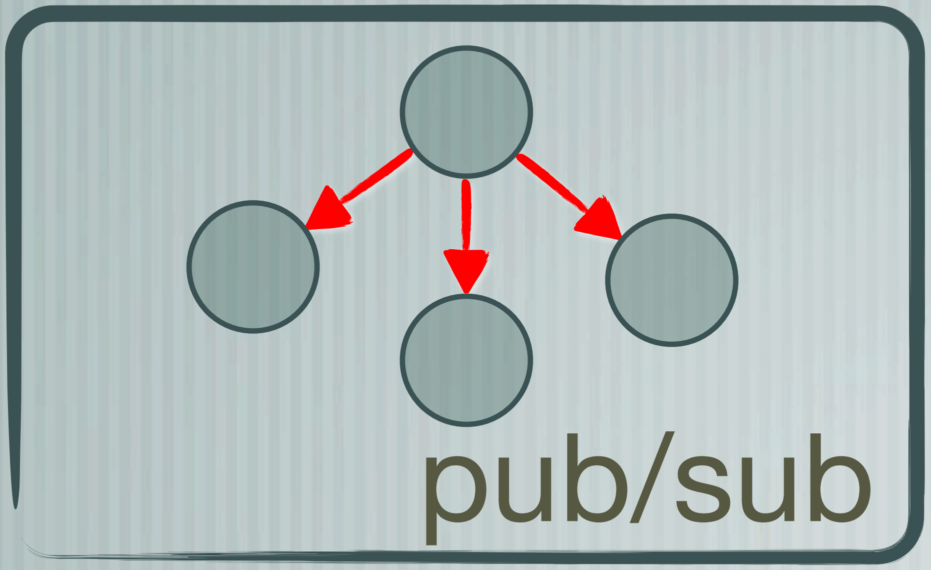


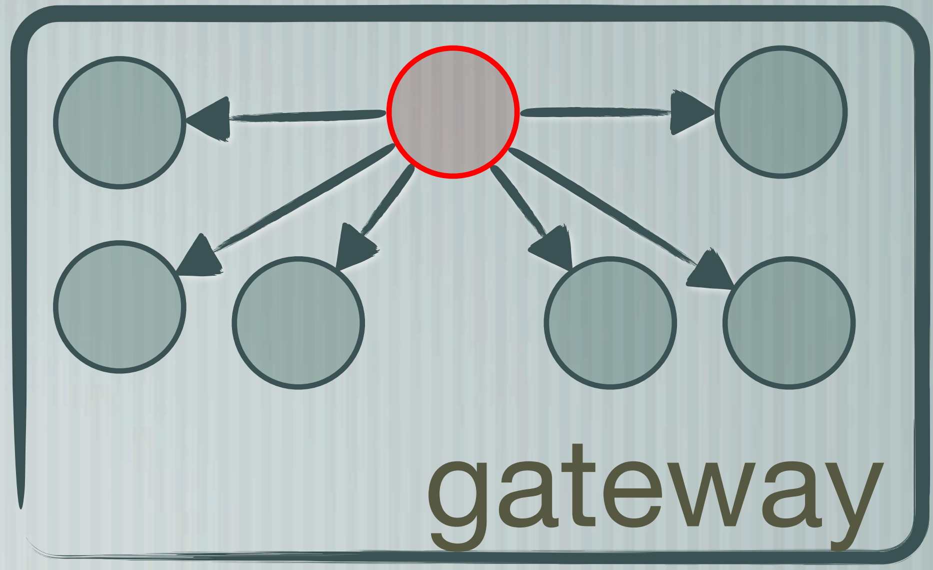

queue


esb


pipeline


async


pub/sub


gateway

request/response

erlrep.erl

```
run() ->
 {ok, Ctx} = erlzmq:context(),
 {ok, Sock} = erlzmq:socket(Ctx, rep),
 ok = erlzmq:bind(Sock,"tcp://*:5454"),
 loop(Sock).

loop(Sock) ->
 {ok, Msg, _F} = erlzmq:recv(Sock),
 Rep = binary_to_list(Msg) ++ " World",
 io:format("Sending ~s~n", [Rep]),
 ok = erlzmq:send(Sock,
              list_to_binary(Rep)),
 loop(Sock).
```

rep.py

```python
import zmq

context = zmq.Context()
server = context.socket(zmq.REP)
server.bind("tcp://*:5454")

while True:
    message = server.recv()
    print "Sending", message, "World\n"
    server.send(message + " World")
```

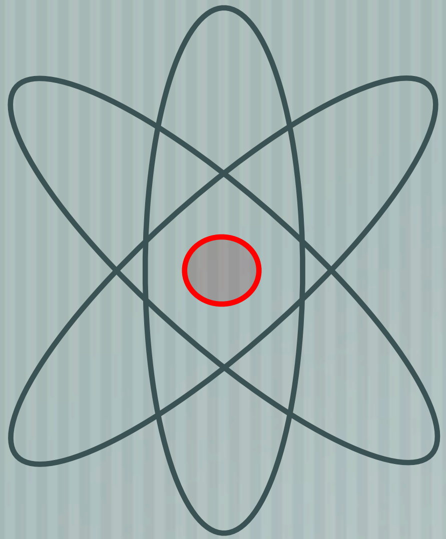
request/response

req.php

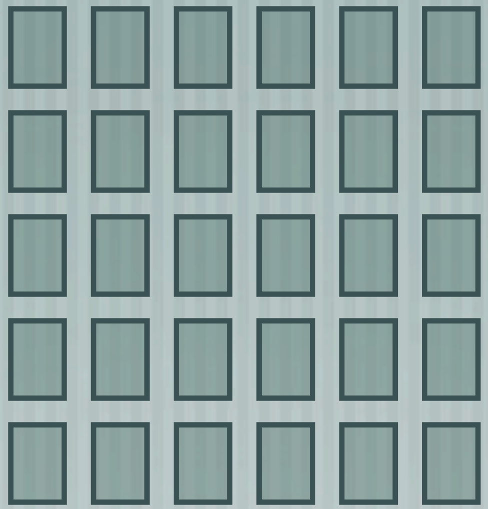
```
$ctx = new ZMQContext();
$req =
  new ZMQSocket($ctx, ZMQ::SOCKET_REQ);
$req->connect("tcp://localhost:5454");

$req->send("Hello");
echo $req->recv();
```

```
git clone http://github.com/zeromq/erlzmq2.git

make
make test
```

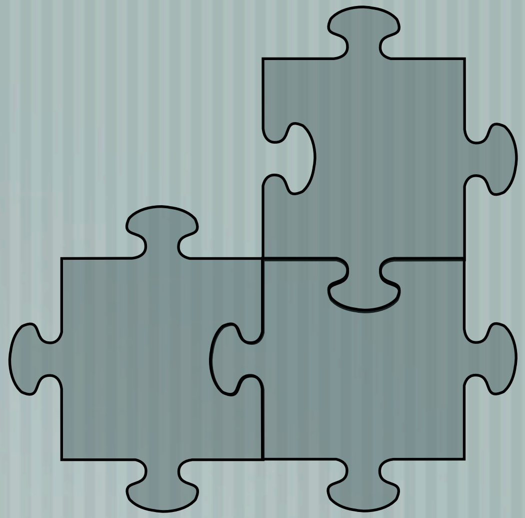
http://download.zeromq.org/
http://github.com/zeromq/zeromq2-1
http://github.com/zeromq/libzmq



atomic



bytes

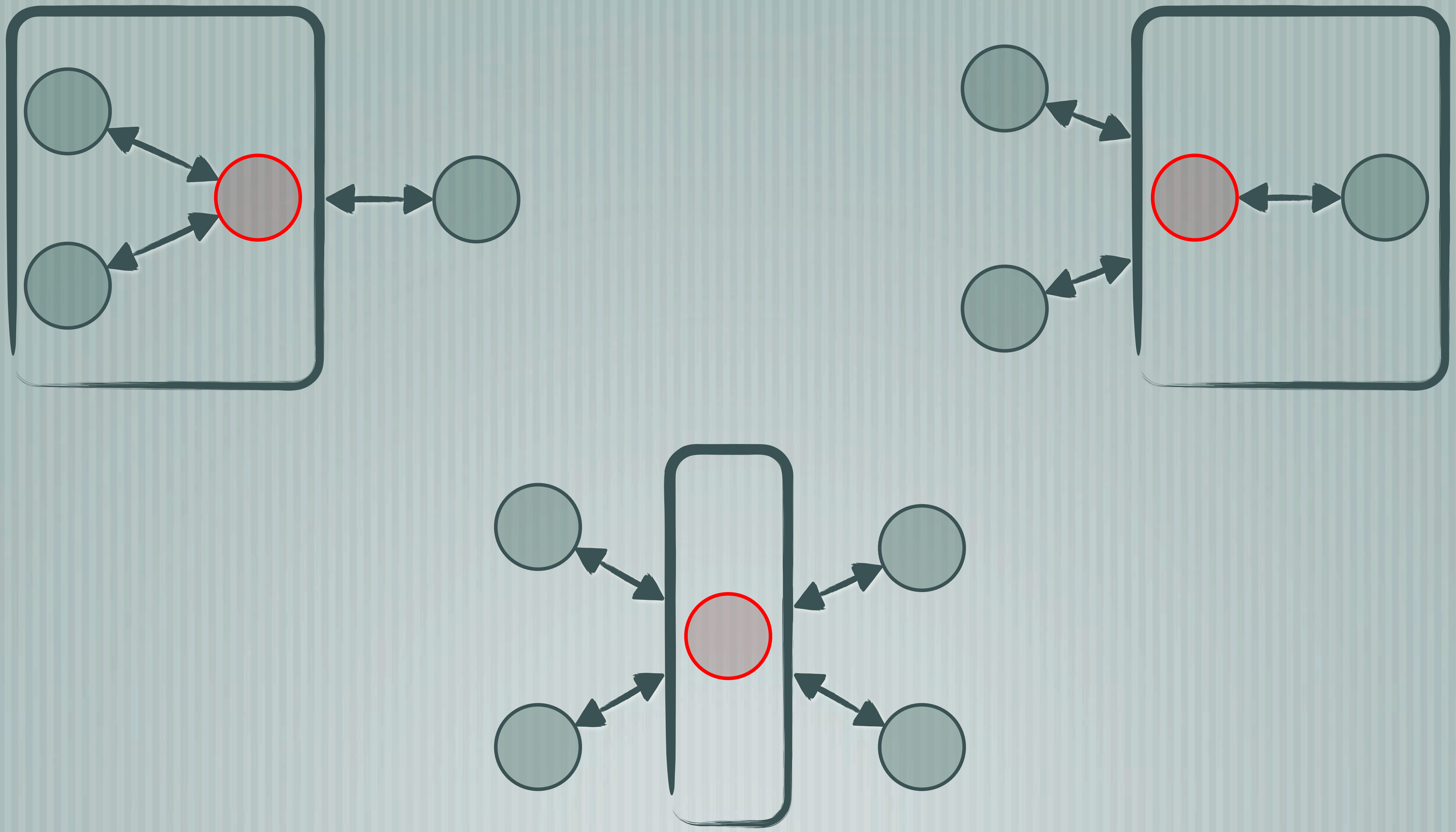


multipart

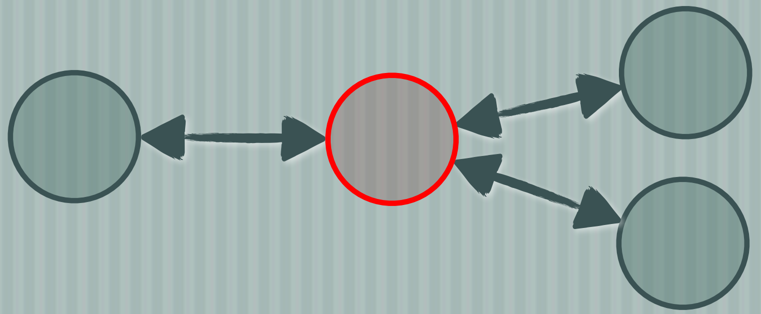
messaging

Post Office Image: http://www.flickr.com/photos/10804218@N00/4315282973
Post Box Image: http://www.flickr.com/photos/kenjonbro/3027166169

queue



queue



```
run() ->
 {ok, Ctx} = erlzmq:context(),
 {ok, Front} = erlzmq:socket(Ctx,
                 [xrep, {active,true}]),
 {ok, Back} = erlzmq:socket(Ctx,
                 [xreq, {active,true}]),
 ok = erlzmq:bind(Front, "tcp://*:5454"),
 ok = erlzmq:bind(Back, "tcp://*:5455"),

 loop(Front, Back),
 ok = erlzmq:close(Front),
 ok = erlzmq:close(Back),
 ok = erlzmq:term(Ctx).
```

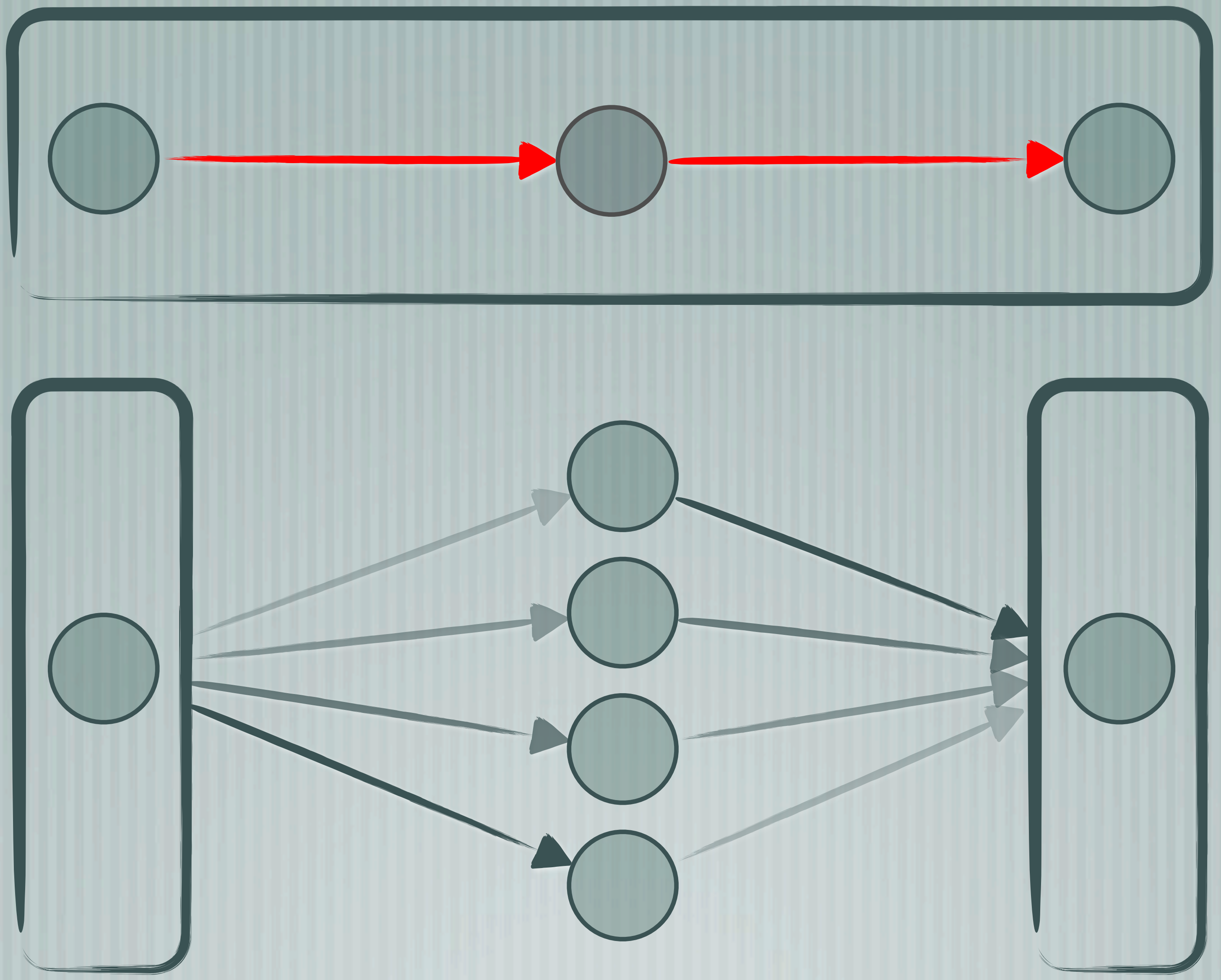
queue.erl

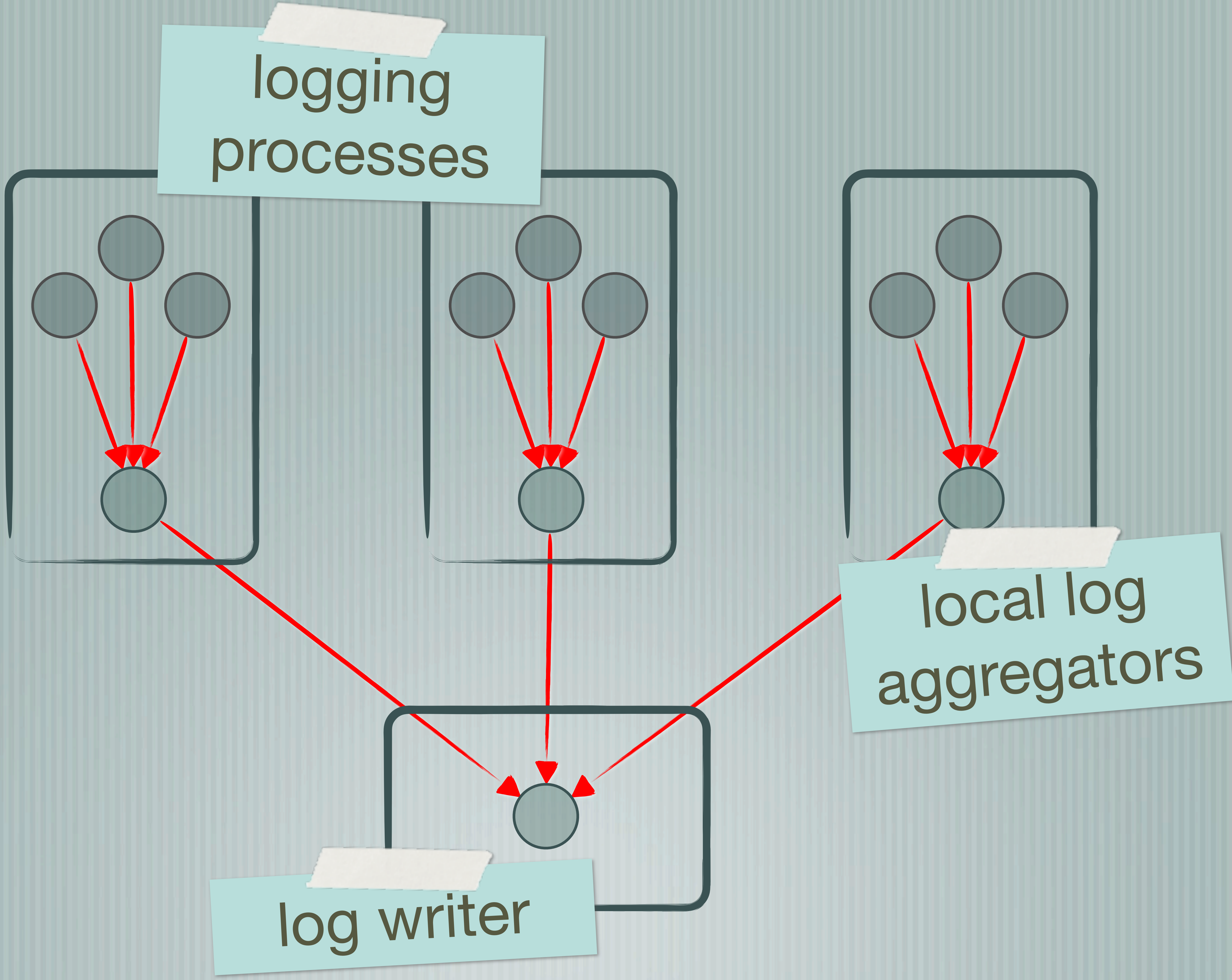
```
loop(Front, Back) ->
 receive
  {zmq, Front, Msg, Flags} ->
    io:format("Sending Back: ~p~n",[Msg]),
    sendall(Back, Msg, Flags),
    loop(Front, Back);
  {zmq, Back, Msg, Flags} ->
    io:format("Sending Front: ~p~n",[Msg]),
    sendall(Front, Msg, Flags),
    loop(Front, Back)
 end.

sendall(To, Part, [rcvmore|_Flags]) ->
  erlzmq:send(To, Part, [sndmore]);
sendall(To, Part, _Flags) ->
  erlzmq:send(To, Part).
```

stable / unstable

Image: http://www.flickr.com/photos/pelican/235461339/

pipeline



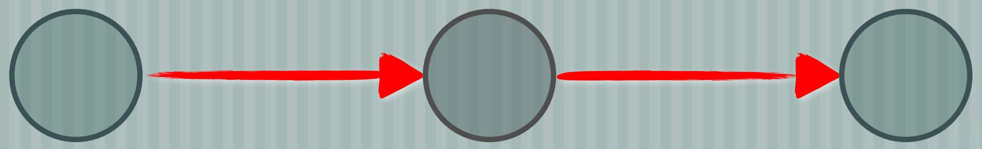


logging processes

local log aggregators

log writer

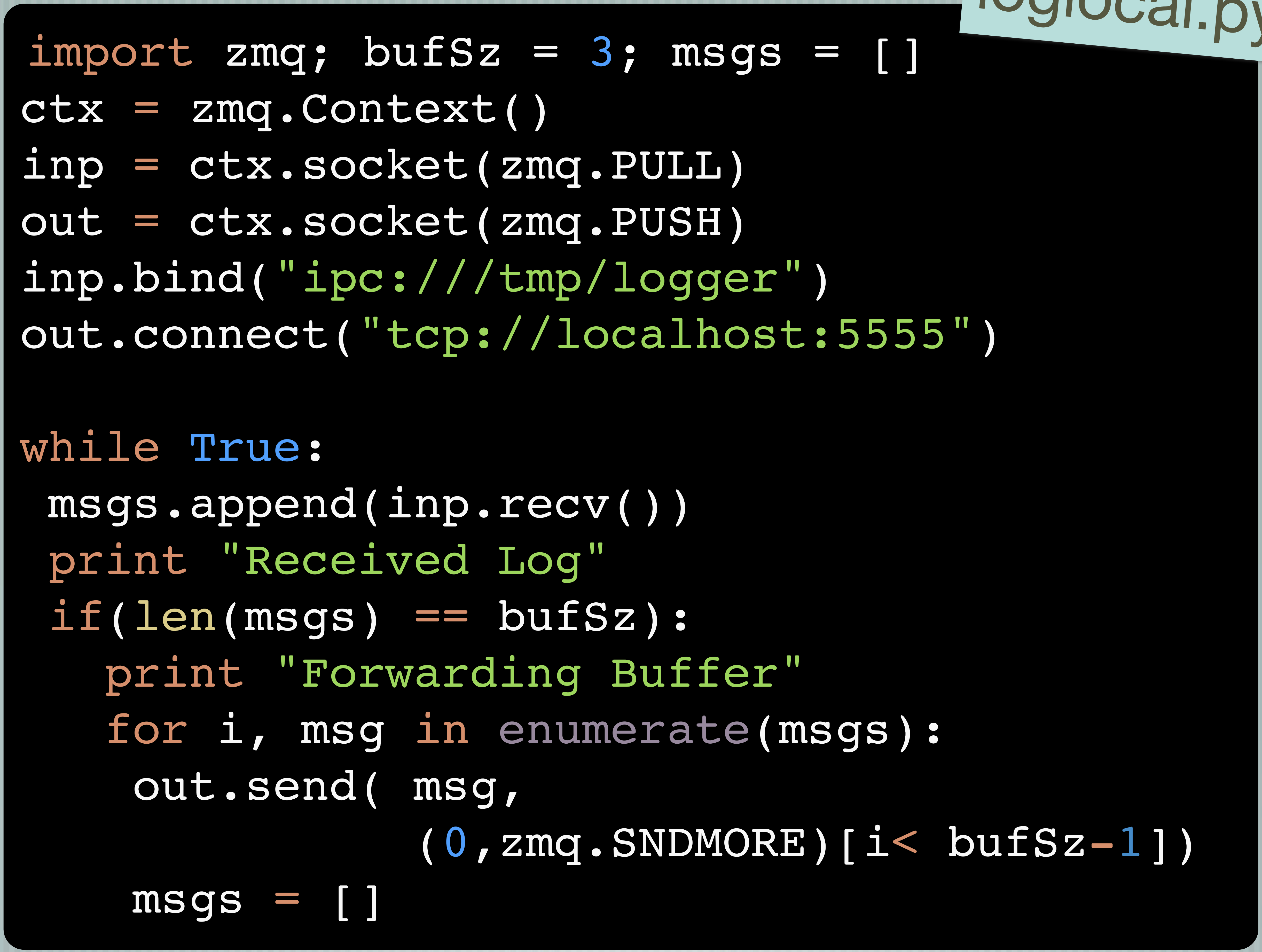
pipeline



logger.php

```
<?php

$ctx = new ZMQContext();
$out = $ctx->getSocket(ZMQ::SOCKET_PUSH);
$out->connect("ipc:///tmp/logger");
$msg = array("time" => time());
$msg['msg'] = $_SERVER['argv'][1];
$out->send(json_encode($msg));
```

loglocal.py



```python
import zmq; bufSz = 3; msgs = []
ctx = zmq.Context()
inp = ctx.socket(zmq.PULL)
out = ctx.socket(zmq.PUSH)
inp.bind("ipc:///tmp/logger")
out.connect("tcp://localhost:5555")

while True:
 msgs.append(inp.recv())
 print "Received Log"
 if(len(msgs) == bufSz):
   print "Forwarding Buffer"
   for i, msg in enumerate(msgs):
    out.send( msg,
              (0,zmq.SNDMORE)[i< bufSz-1])
    msgs = []
```

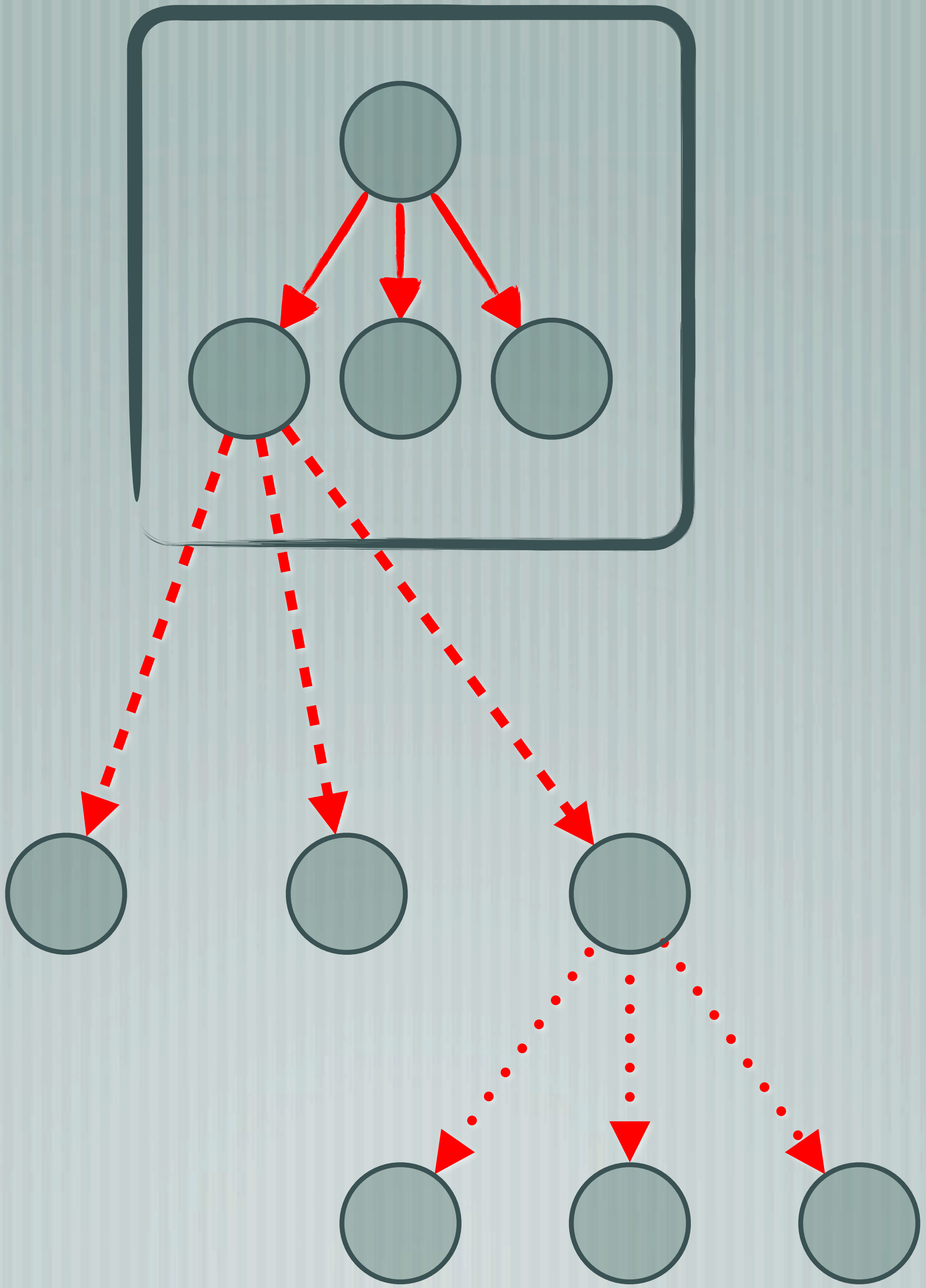
logserv.erl

```erlang
run() ->
 {ok, Ctx} = erlzmq:context(),
 {ok, In} = erlzmq:socket(Ctx,
                         [pull, {active,true}]),
 ok = erlzmq:bind(In, "tcp://*:5555"),
 loop(In).

loop(In) ->
 receive
  {zmq, In, Msg, _F} ->
  {ok,{obj, [{_, Time}, {_, Log}] }, [] }
   = rfc4627:decode(Msg),
  io:format("~B ~s~n", [Time, Log]),
  loop(In)
 end.
```

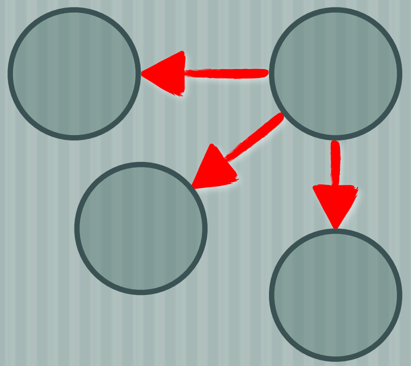
pub/sub

Image: http://www.flickr.com/photos/nikonvscanon/4519133003/

pub/sub



pub/sub

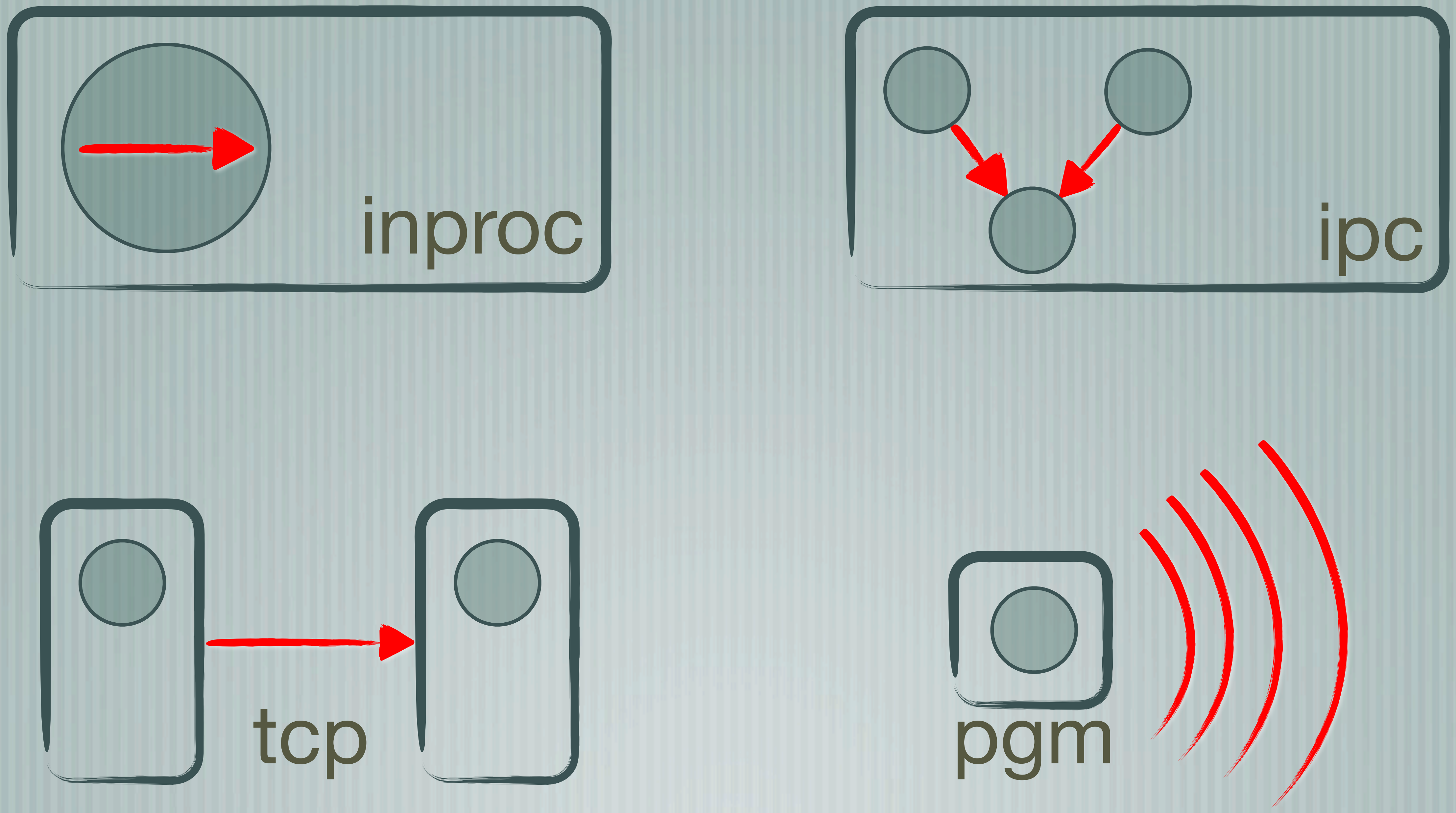


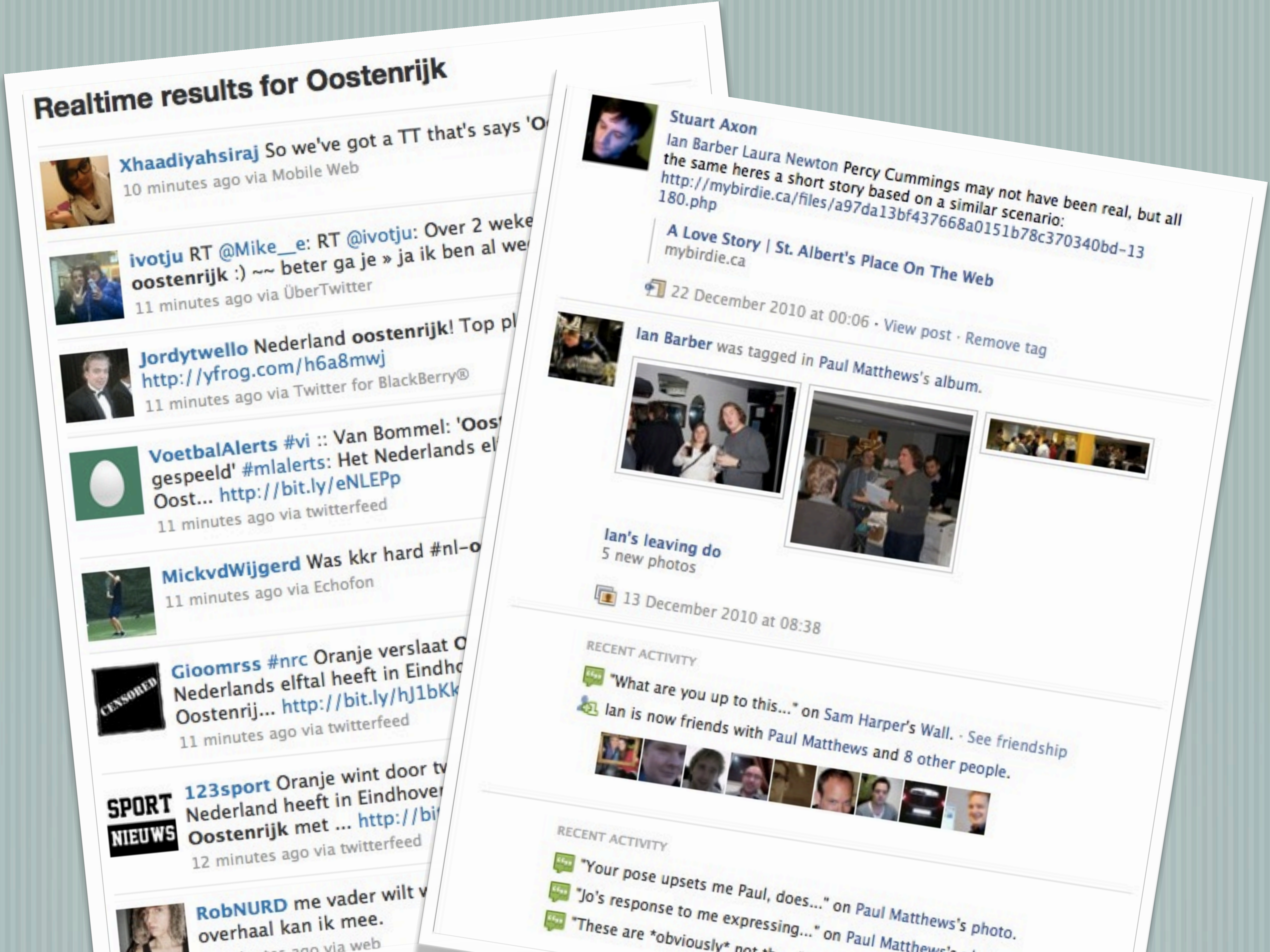
repeater.erl

```
init(_) ->
  spawn_link(?MODULE, run, []).

terminate(Pid) ->
  Pid ! {cmd, kill}.

handle_event(Event, Pid) ->
  Pid ! Event.
```

```erlang
run() ->
  {ok, Ctx} = erlzmq:context(),
  {ok, Sock} = erlzmq:socket(Ctx, [pub]),
  ok = erlzmq:bind(Sock, "tcp://*:5656"),
  loop(Sock).

loop(Sock) ->
  receive
    {cmd, kill} ->
      ok;
    {Action, Id, Event} ->
      Json = rfc4627:encode({obj,
             [{action, Action},
              {id, Id}, {event, Event}]}),
      erlzmq:send(Sock,
                    list_to_binary(Json)),
      loop(Sock)
  end.
```

types of transport





Realtime results for Oostenrijk

Xhaadiyahsiraj So we've got a TT that's says 'O
10 minutes ago via Mobile Web

ivotju RT @Mike__e: RT @ivotju: Over 2 weke
oostenrijk :) ~~ beter ga je » ja ik ben al we
11 minutes ago via ÜberTwitter

Jordytwello Nederland **oostenrijk**! Top pl
http://yfrog.com/h6a8mwj
11 minutes ago via Twitter for BlackBerry®

VoetbalAlerts #vi :: Van Bommel: 'Oos
gespeeld' #mlalerts: Het Nederlands el
Oost... http://bit.ly/eNLEPp
11 minutes ago via twitterfeed

MickvdWijgerd Was kkr hard #nl-o
11 minutes ago via Echofon

Gioomrss #nrc Oranje verslaat O
Nederlands elftal heeft in Eindho
Oostenrij... http://bit.ly/hJ1bKk
11 minutes ago via twitterfeed

123sport Oranje wint door tw
Nederland heeft in Eindhove
Oostenrijk met ... http://bi
12 minutes ago via twitterfeed

RobNURD me vader wilt v
overhaal kan ik mee.
... ago via web

Stuart Axon
Ian Barber Laura Newton Percy Cummings may not have been real, but all the same heres a short story based on a similar scenario:
http://mybirdie.ca/files/a97da13bf437668a0151b78c370340bd-13180.php

A Love Story | St. Albert's Place On The Web
mybirdie.ca

22 December 2010 at 00:06 · View post · Remove tag

Ian Barber was tagged in Paul Matthews's album.

Ian's leaving do
5 new photos

13 December 2010 at 08:38

RECENT ACTIVITY

"What are you up to this..." on Sam Harper's Wall. · See friendship

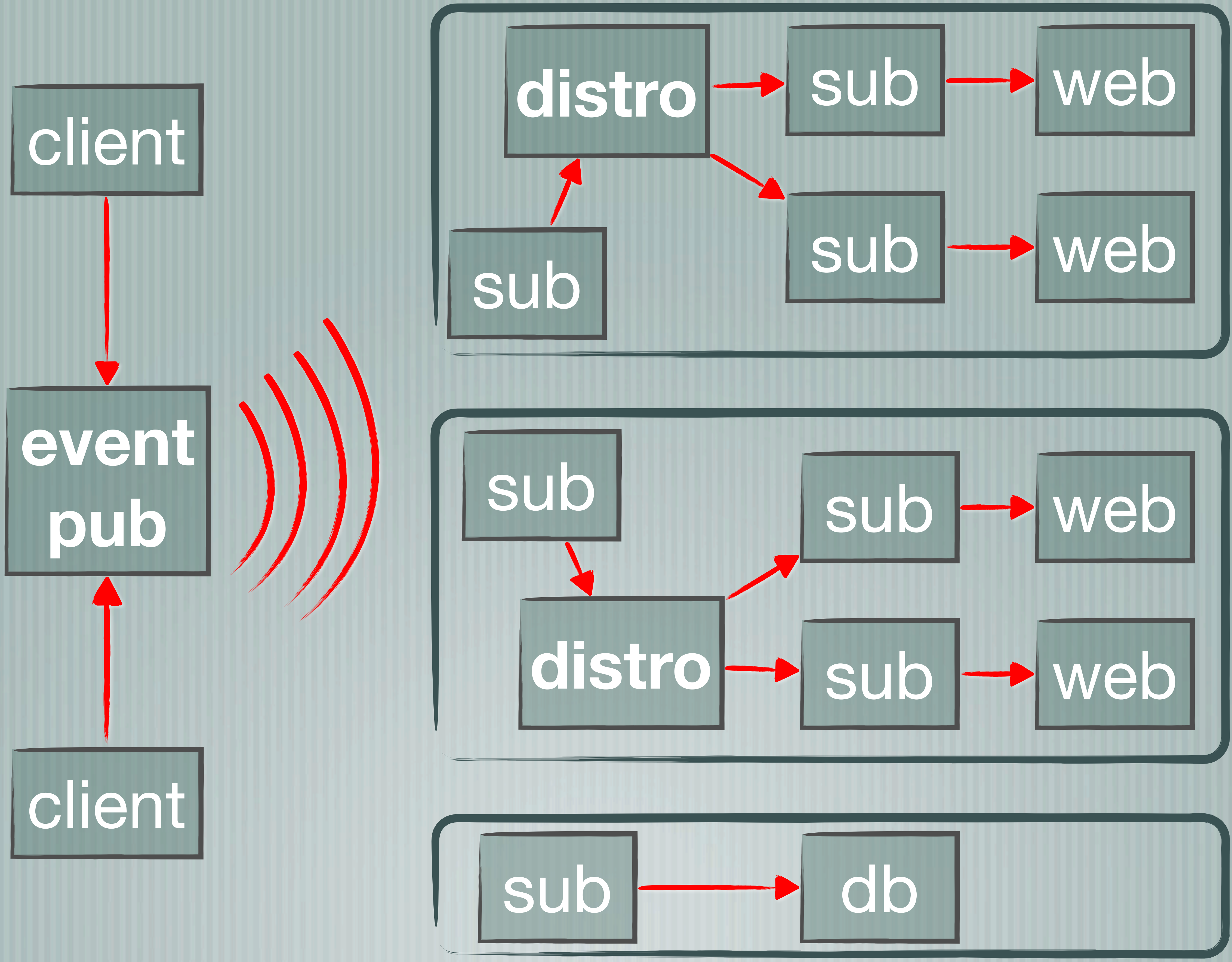
Ian is now friends with Paul Matthews and 8 other people.

RECENT ACTIVITY

"Your pose upsets me Paul, does..." on Paul Matthews's photo.

"Jo's response to me expressing..." on Paul Matthews's photo.

"These are *obviously* not th... on Paul Matthews's ...



client

event pub

client

distro

sub

web

sub

sub

web

sub

sub

web

distro

sub

web

sub

db

```
import zmq
from zmq.devices.basedevice
     import ProcessDevice

pd = ProcessDevice(zmq.STREAMER,
                   zmq.PULL, zmq.PUB)
pd.bind_in("tcp://*:6767")
pd.connect_out("epgm://eth0;239.192.0.1:7601")
pd.setsockopt_out(zmq.RATE, 10000)
pd.start()

# Do other things here
```

eventhub.php

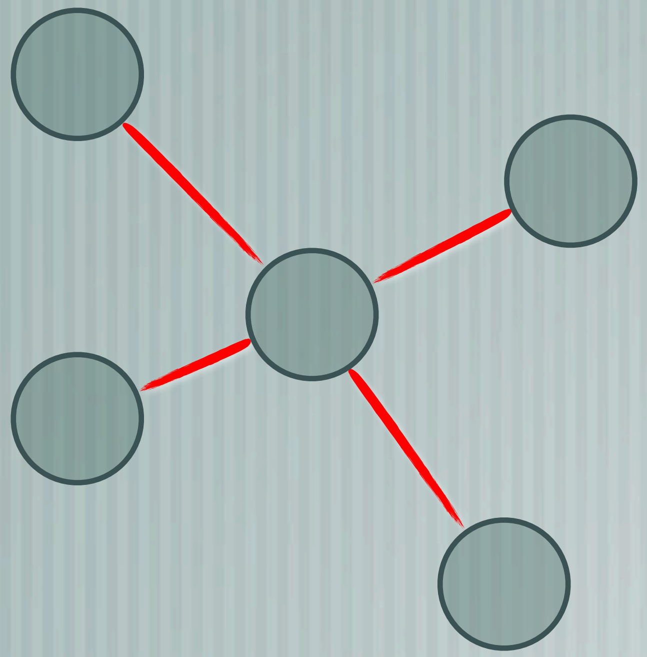
```
int main(void) {
  void *ctx = zmq_init(1);
  void *in =  zmq_socket(ctx, ZMQ_SUB);
  void *out = zmq_socket(ctx, ZMQ_PUB);
  zmq_setsockopt(in, ZMQ_SUBSCRIBE, "", 0);
  int rcin =  zmq_connect(in,
              "epgm://;239.192.0.1:7601");
  int rcout = zmq_bind(out,
              "ipc:///tmp/events");
  int rcd =   zmq_device(ZMQ_FORWARDER,
              in, out);

  zmq_close(in); zmq_close(out);
  zmq_term(ctx);
  return 0;
}
```

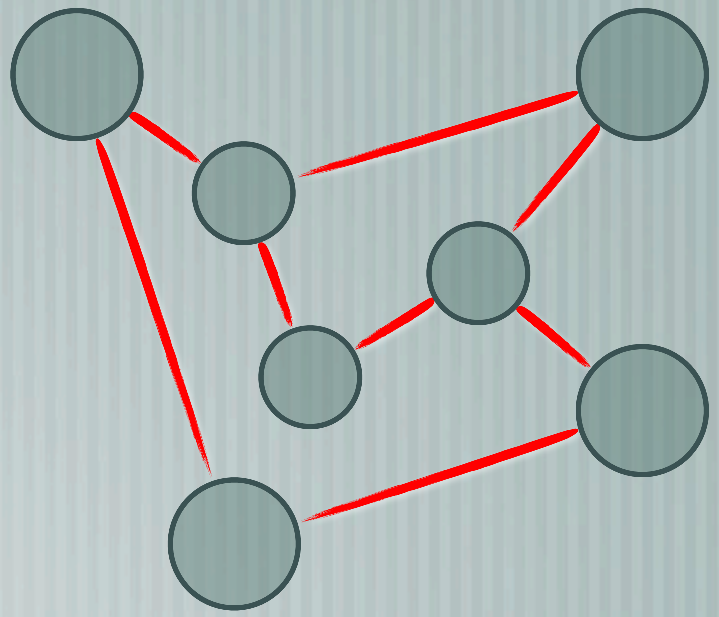
distro.c

```php
$ctx = new ZMQContext();
$in = $ctx->getSocket(ZMQ::SOCKET_SUB);
for($i = 0; $i<100; $i++) {
   $in->setSockOpt(
           ZMQ::SOCKOPT_SUBSCRIBE,
           rand(100000, 999999));
}
$in->connect("ipc:///tmp/events");
$i = 0;
while($i++ < 1000) {
  $who = $in->recv();
  $msg = $in->recv();
  printf("%s %s %s", $who, $msg, PHP_EOL);
}
```

client.php

amqp /= ømq





http://zero.mq/amqp

```
[{r0mq,
  [{services,
    [
{<<"PIPELINE">>, pull, "tcp://127.0.0.1:5557"},
{<<"PIPELINE">>, push, "tcp://127.0.0.1:5558"},
{<<"PUBSUB">>, pub, "tcp://127.0.0.1:5555"},
{<<"PUBSUB">>, sub, "tcp://127.0.0.1:5556"}
    ]
  }]
}].
```

rabbitmq.config

Helpful Links

http://zero.mq

http://zguide.zero.mq

http://zero.mq/ib

thanks!