nodeJS

Evented I/O for V8 javascript.

Some parts taken from http://s3.amazonaws.com/four.livejournal/20091117/jsconf.pdf

~~Server-Side JavaScript~~

http://narwhaljs.org

http://www.mozilla.org/rhino/

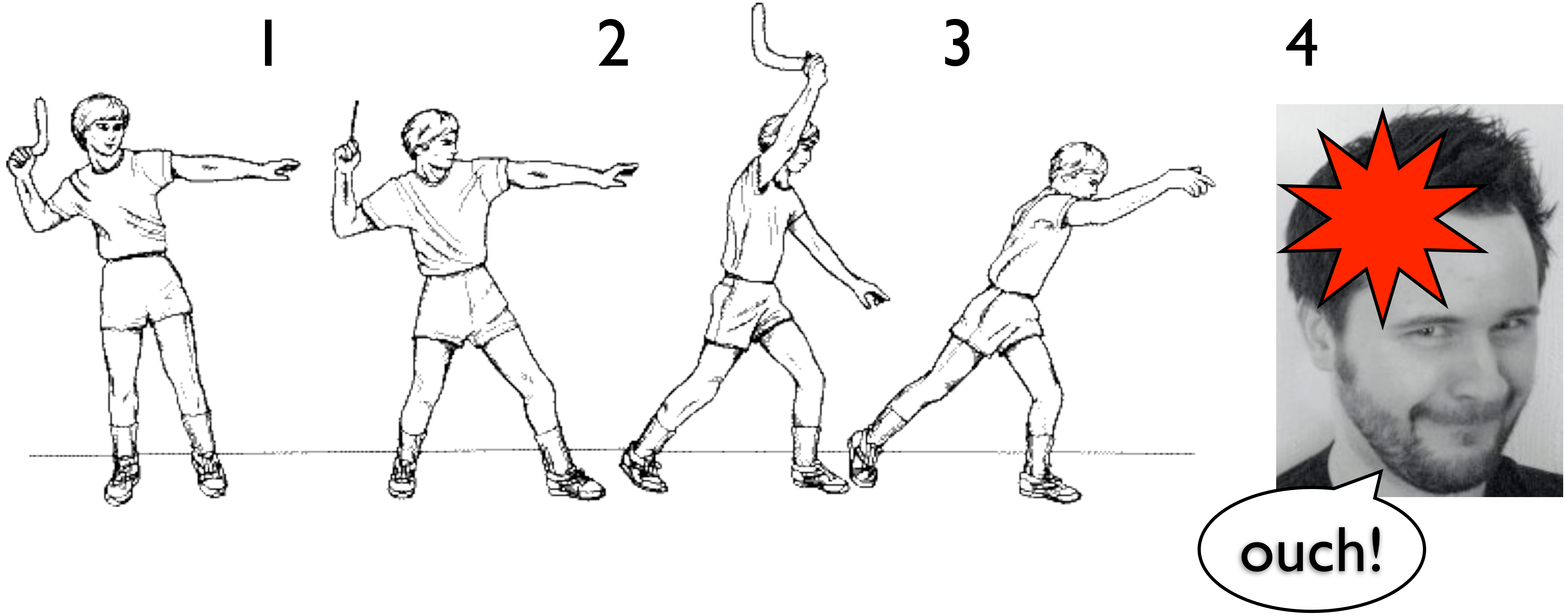
```
diff narwhal node | grep io
```

Image: http://www.greendisk.com/images/Computer.jpg

```
var results = db.query("select * from T");

var content = file.read();

var input = gets();
```

wrong

?



http://www.boomerang.org.au/articles/article-how-to-throw-a-boomerang.html

Marc-André
Cournoyer

Thin

Ruby

A Programmer's Best Friend

Ruby/EventMachine

Fast Network I/O and Event Management for Ruby Programmers heroku

TALKER

Fast, intuitive and extensible group chat

http://talkerapp.com

Scale to the Moon™

(results may vary)

Performance Expert?

Threads

Deadlocks

Threads

Race

conditions

Deadlocks Threads

Race

conditions

Deadlocks Threads

Synchronization

Race

conditions

Deadlocks Threads

Synchronization

Context

Switching

Race conditions

Deadlocks Threads

Synchronization

Context Switching

Mutex

Stack Race conditions

Deadlocks Threads

Synchronization

Context Switching Mutex

no

TOP SECRET

I/O Bound

not CPU bound

Waiting for disk / socket ...

Threads

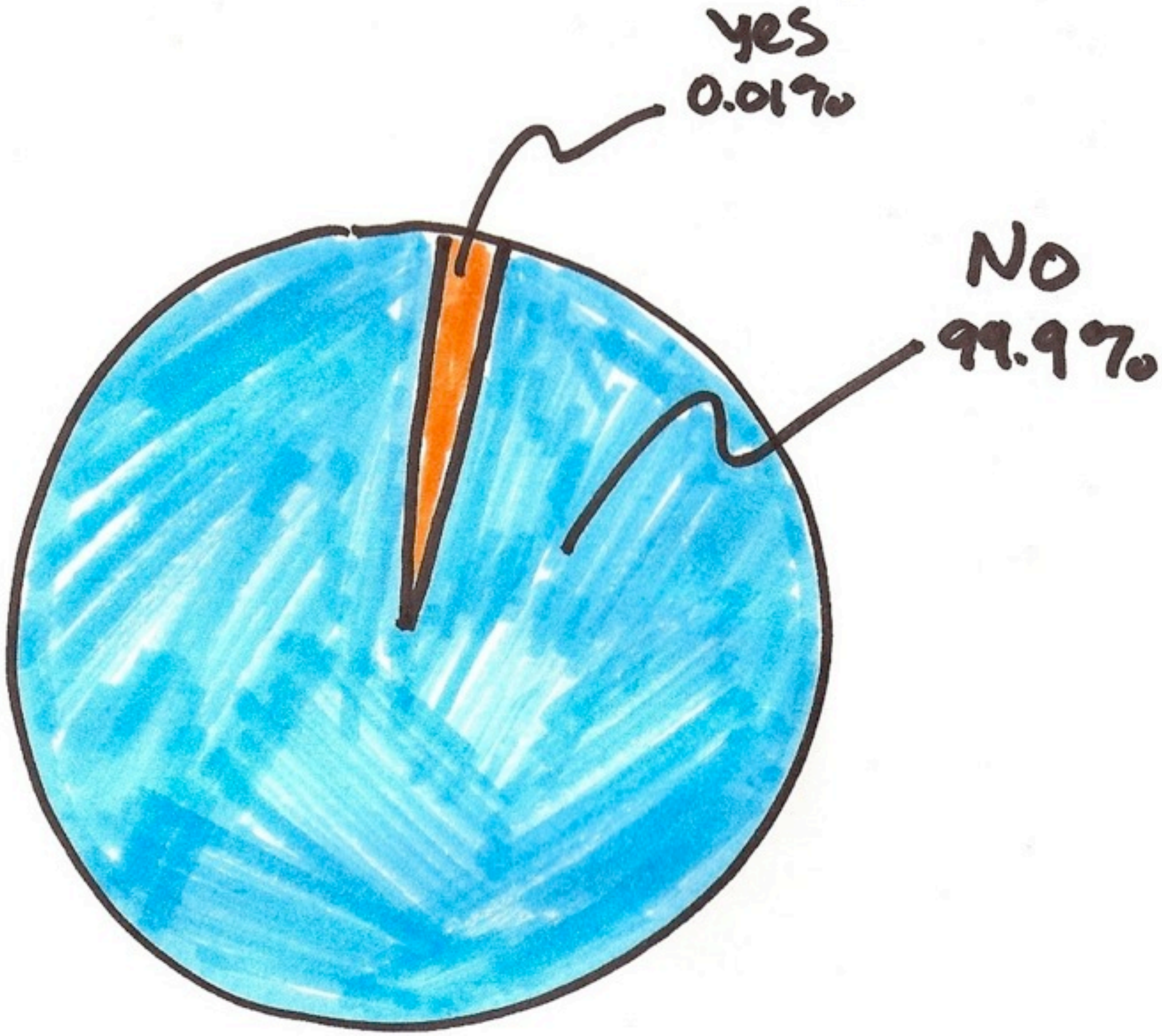
Threads

OVERKILL

Evented I/O

Fast Systems

Do you like slow websites?



Scienbific

(

Fast is relative

DemoCamp » TalkerApp

Gary: View code

```
Talker.plugins.push(new Talker.FoundedFormatter());
Talker.plugins.push(new Talker.PasteFormatter());
Talker.plugins.push(new Talker.ImageFormatter());
```

me: This code is awesome!

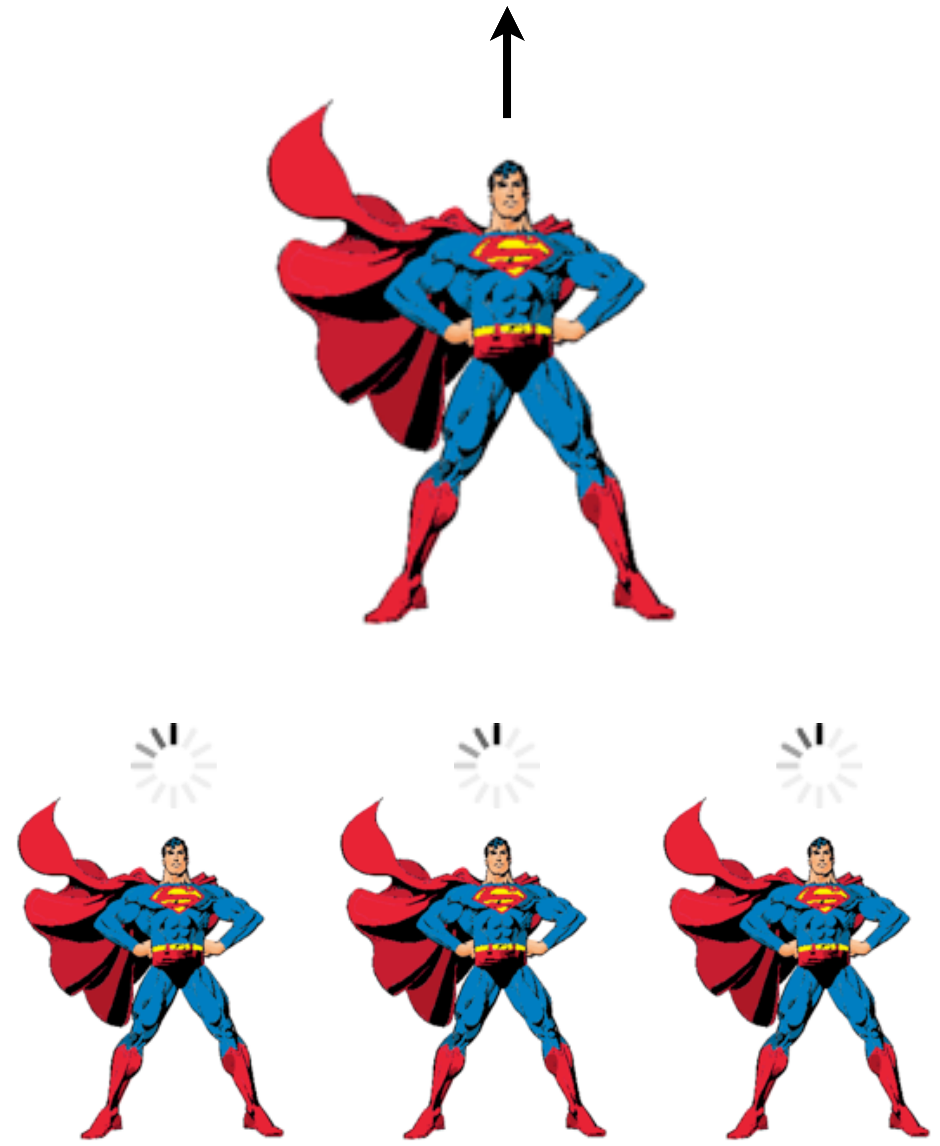
Gary:

Neu: am I supposed to say something smart here :P?

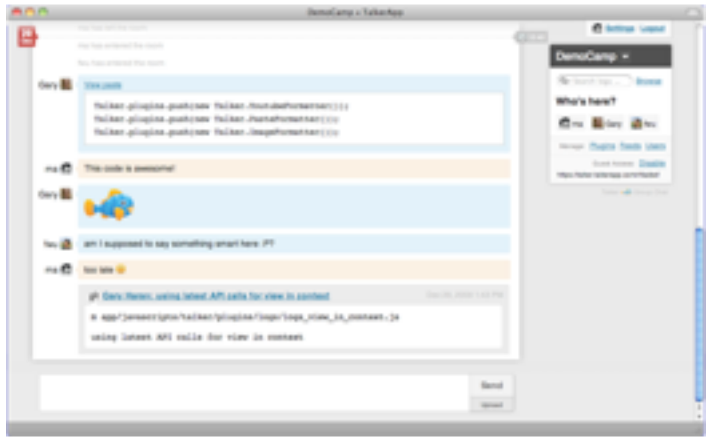
me: too late

Gary Haran: using latest API calls for view in context

```
M app/javascripts/talker/plugins/logs/logs_view_in_context.js
using latest API calls for view in context
```



Source: http://www.superherostuff.com/blog/images/superman-standing.gif


DemoCamp » TalkerApp

Settings Logout

DemoCamp

Browse

Who's here?

me Gary Nou

Gary: View code

```
Talker.plugins.push(new Talker.HoundedFormatter());
Talker.plugins.push(new Talker.PasteFormatter());
Talker.plugins.push(new Talker.ImageFormatter());
```

me: This code is awesome!

Nou: am I supposed to say something smart here :P?

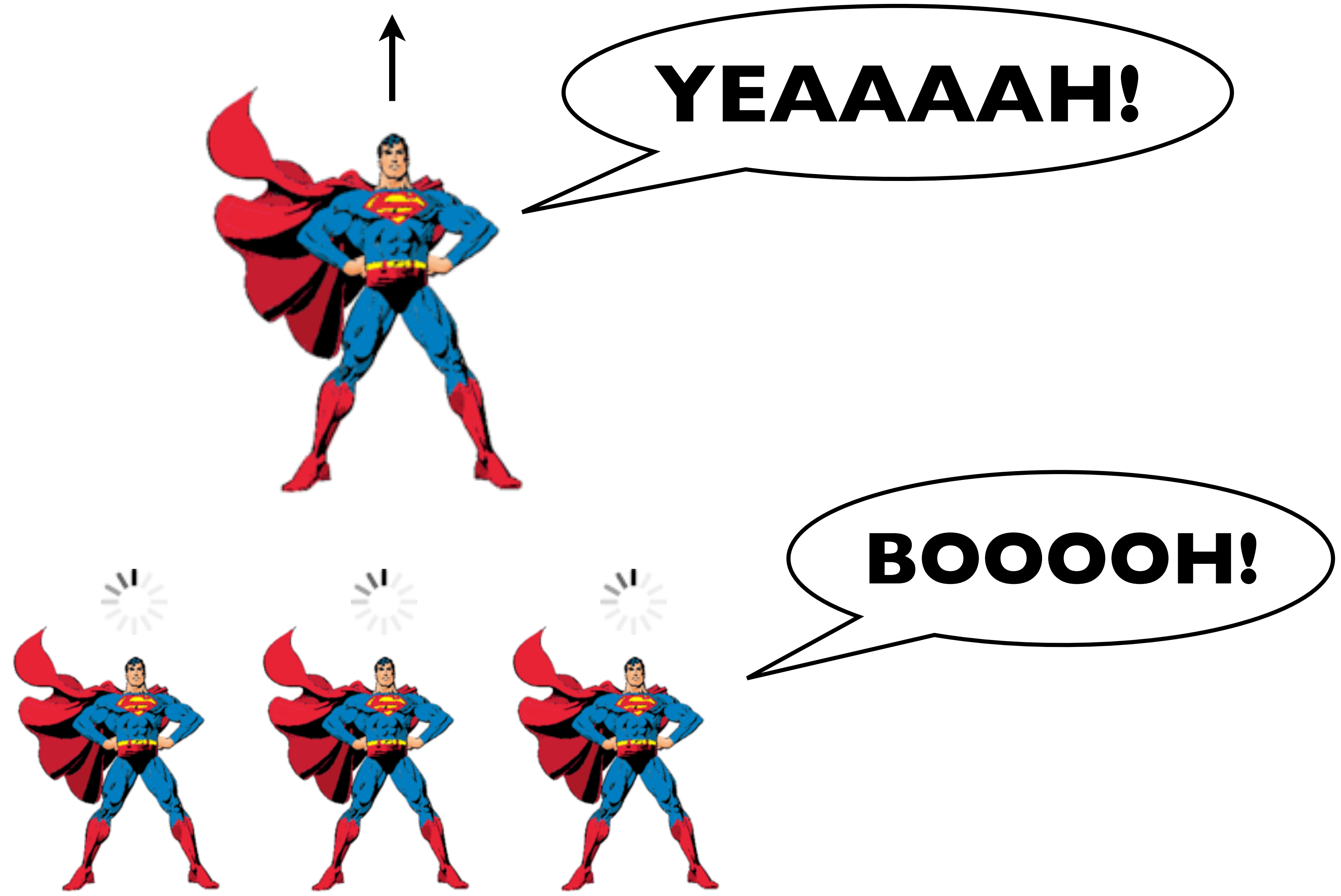
me: too late

Gary Haran: using latest API calls for view in context

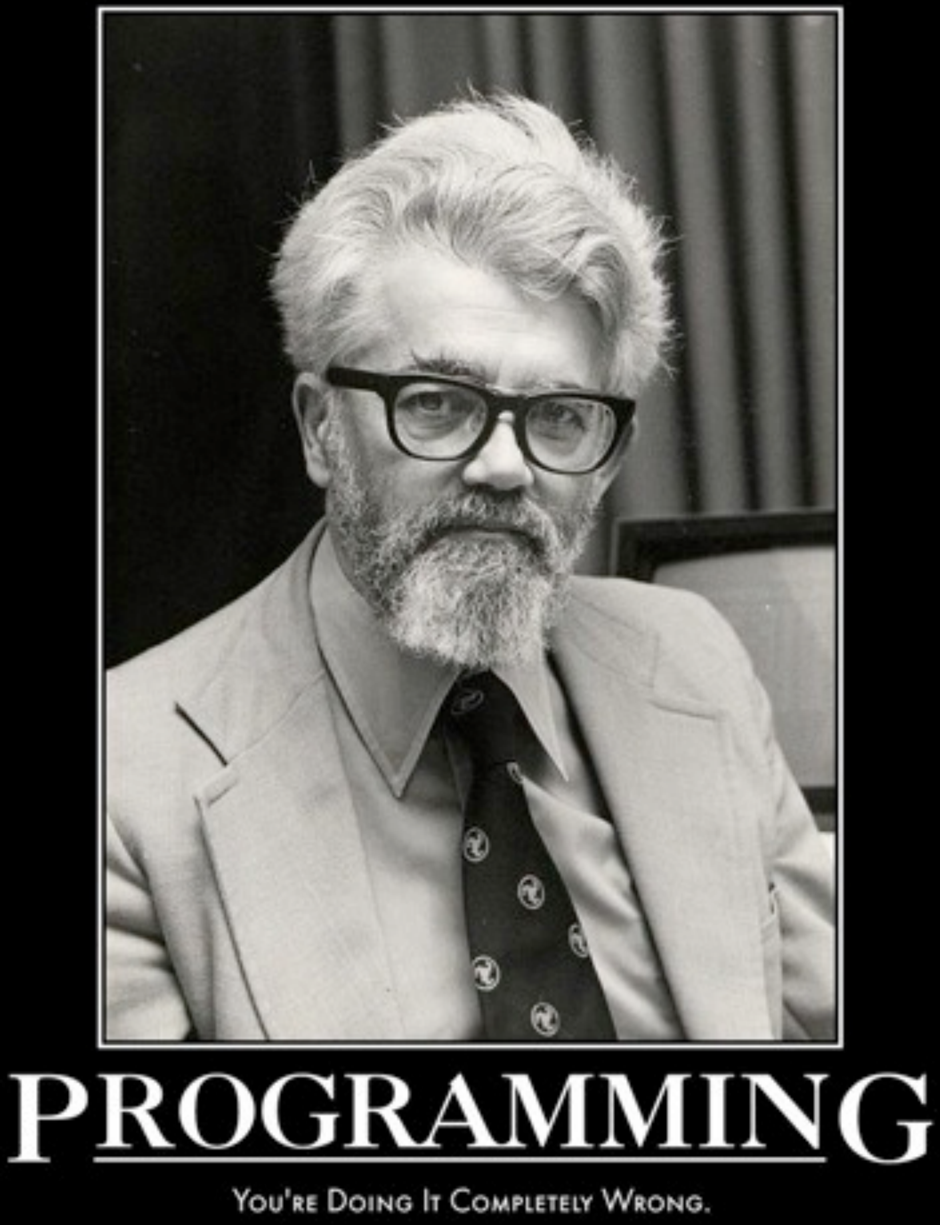
```
M app/javascripts/talker/plugins/logs/logs_view_in_context.js
using latest API calls for view in context
```

Send

Upload



Source: http://www.superherostuff.com/blog/images/superman-standing.gif



Source: http://www.lisp50.org/schedule/schedule/mccarthy_files/doing-it-wrong.jpg

Fast language

not important

efficient use

resources

is

Languages are not important,
paradigms are.

Java or Ruby

~~Java or Ruby~~

Batch or Event-driven

http://www.sfgate.com/blogs/images/sfgate/whalen/2009/04/13/arnoldhamlet.png

)

nodeJS

Evented I/O

Evented I/O

Asynchronous I/O

Evented I/O

Asynchronous I/O

Event-Driven Programming

Evented I/O

Asynchronous I/O

Event-Driven Programming

Async

Evented I/O

Asynchronous I/O

Event-Driven Programming

Async

less-then-expert programmers

Like me

Highly concurrent programs

```
var results = db.query("select * from T");
```

```
db.query("select * from T", function(result) {
  // use results
});
// do other stuff here ...
```

Evented I/O

Evented I/O

Evented I/O

$$\begin{tikzcd}
\underline{\text{Event}}\text{ed} \arrow[r, bend left] & \text{I/O} \arrow[dl] \\
& \text{callback} &
\end{tikzcd}$$

callbacks

```
db.query("select * from T", function(result) {
  // use results
});
```

```
db.query("select * from T", function(result) {
  // use results
});
```

simple

no threads

high concurrency

EXTREME

high concurrency

why?

2

1st

```
puts("Enter your name: ");
var name = gets();
puts("Name: " + name);
puts("Enter your name: ");
var name = gets(function(name) {
  puts("Name: " + name);
});
```

2nd

Sphinx

amazon webservices™

MySQL®

memcached

redis

Sphinx

amazon web services™

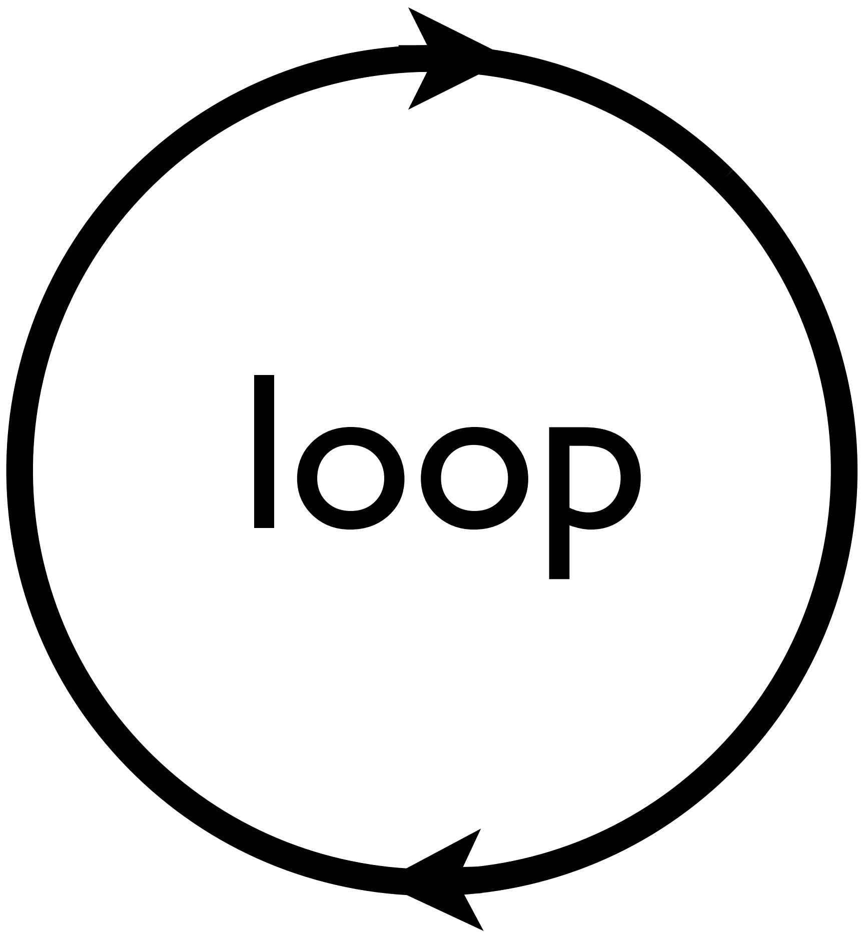
MySQL

BLOCKING

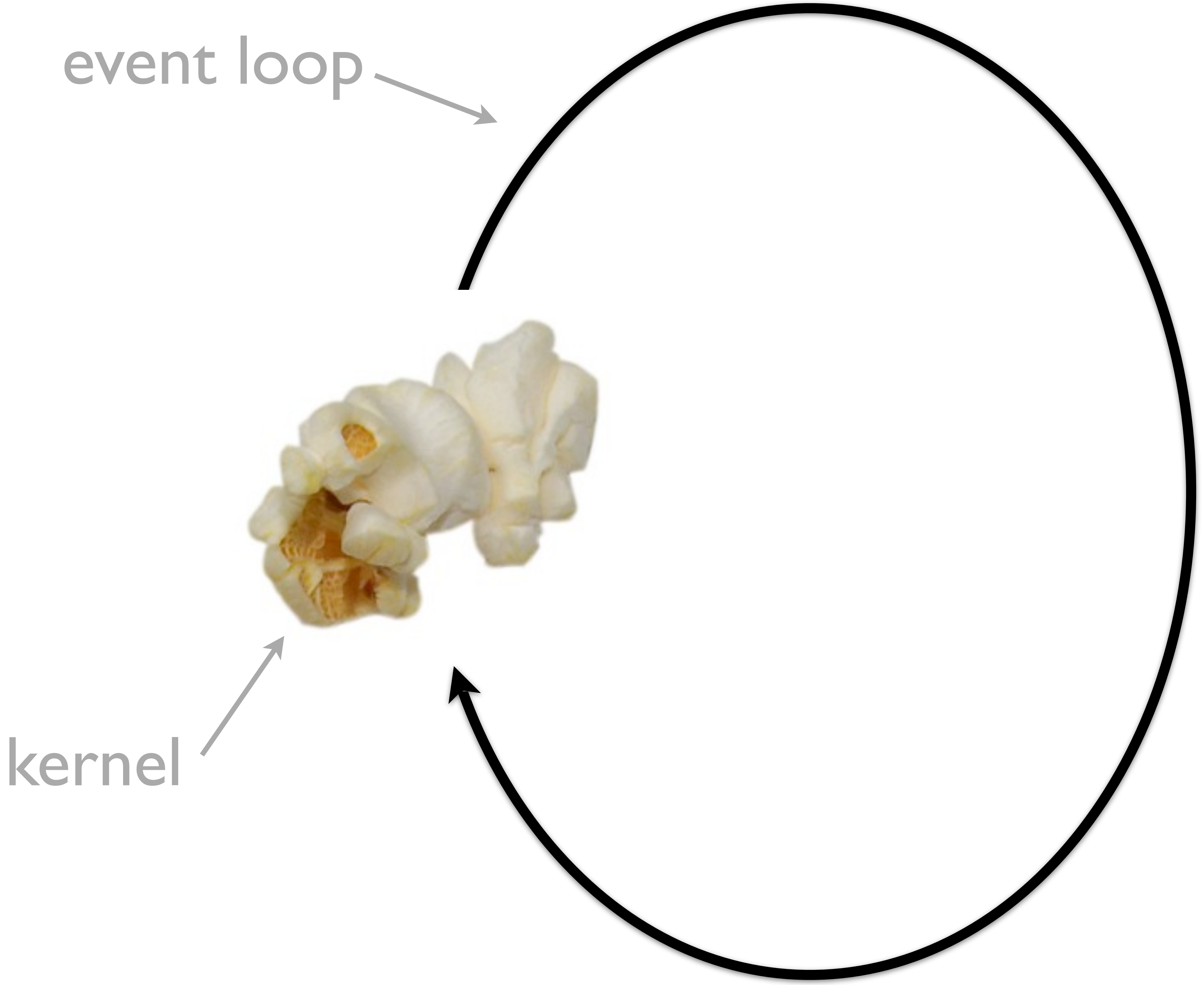
memcached

redis

event

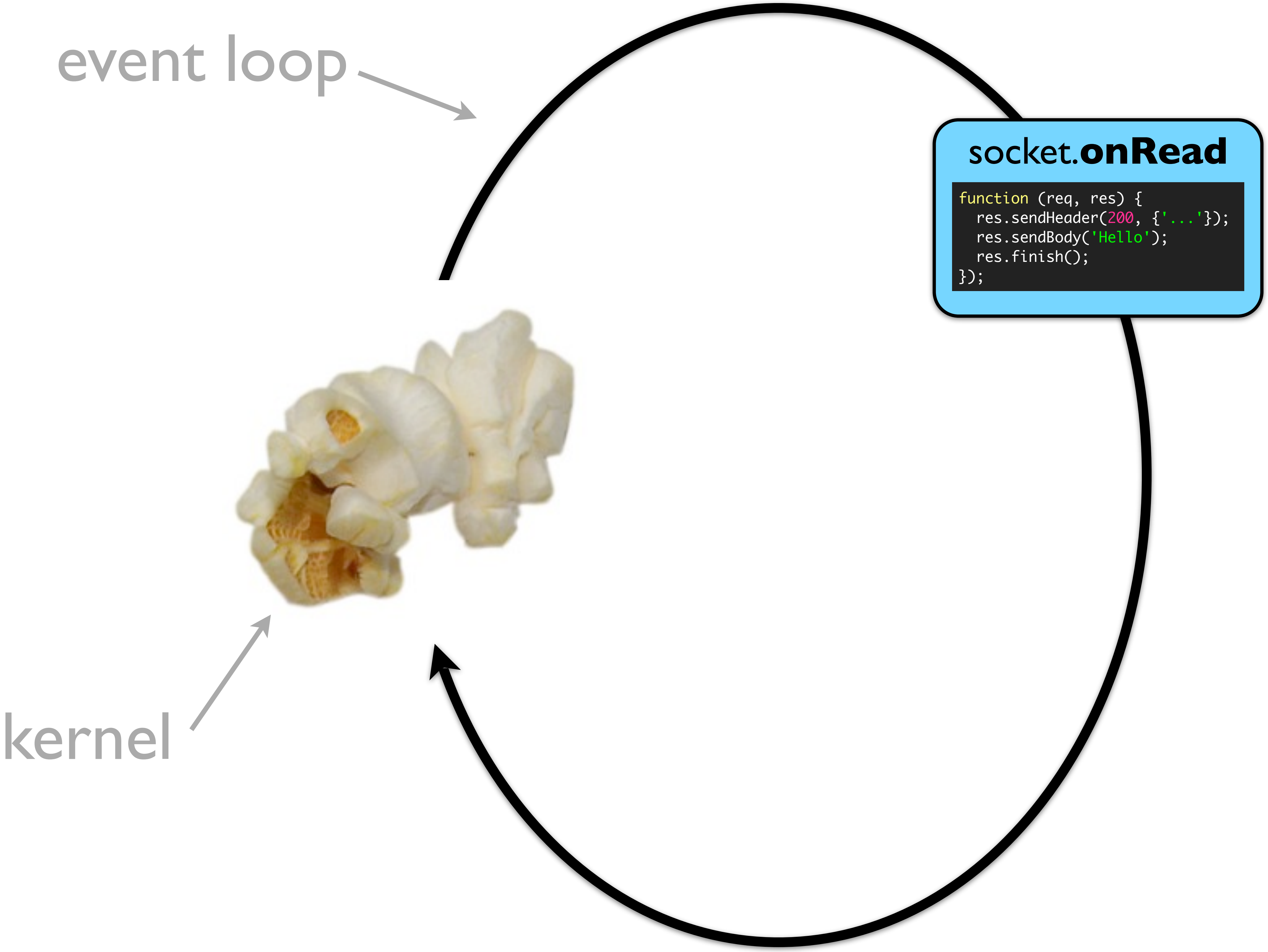

loop

?



(Artistic representation)

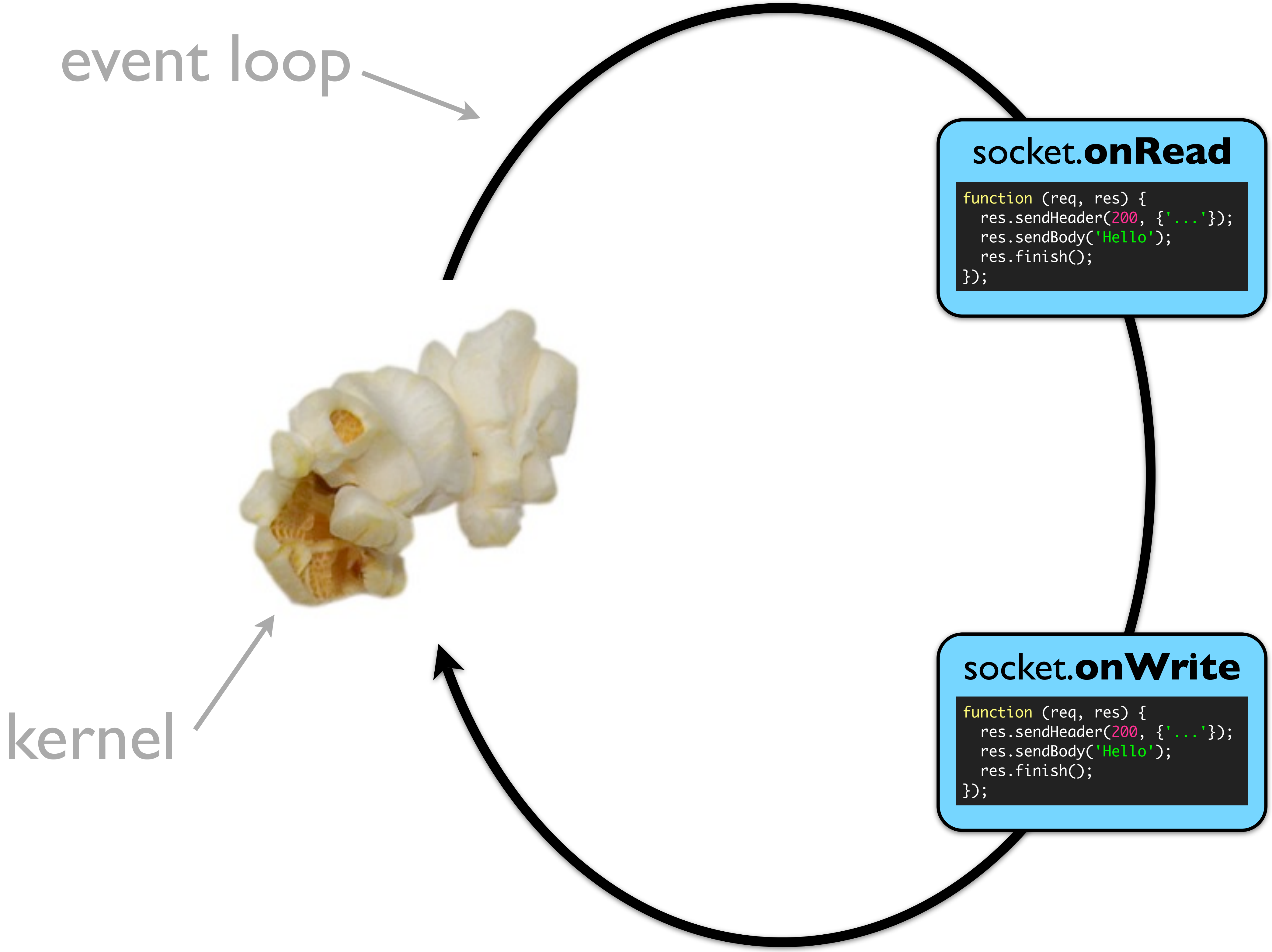
Source: http://pack283.chieftarhe.org/wp-content/uploads/2008/08/butterfly_kernel.jpg



```
function (req, res) {
  res.sendHeader(200, {'...'});
  res.sendBody('Hello');
  res.finish();
});
```

(Artistic representation)

Source: http://pack283.chieftarhe.org/wp-content/uploads/2008/08/butterfly_kernel.jpg

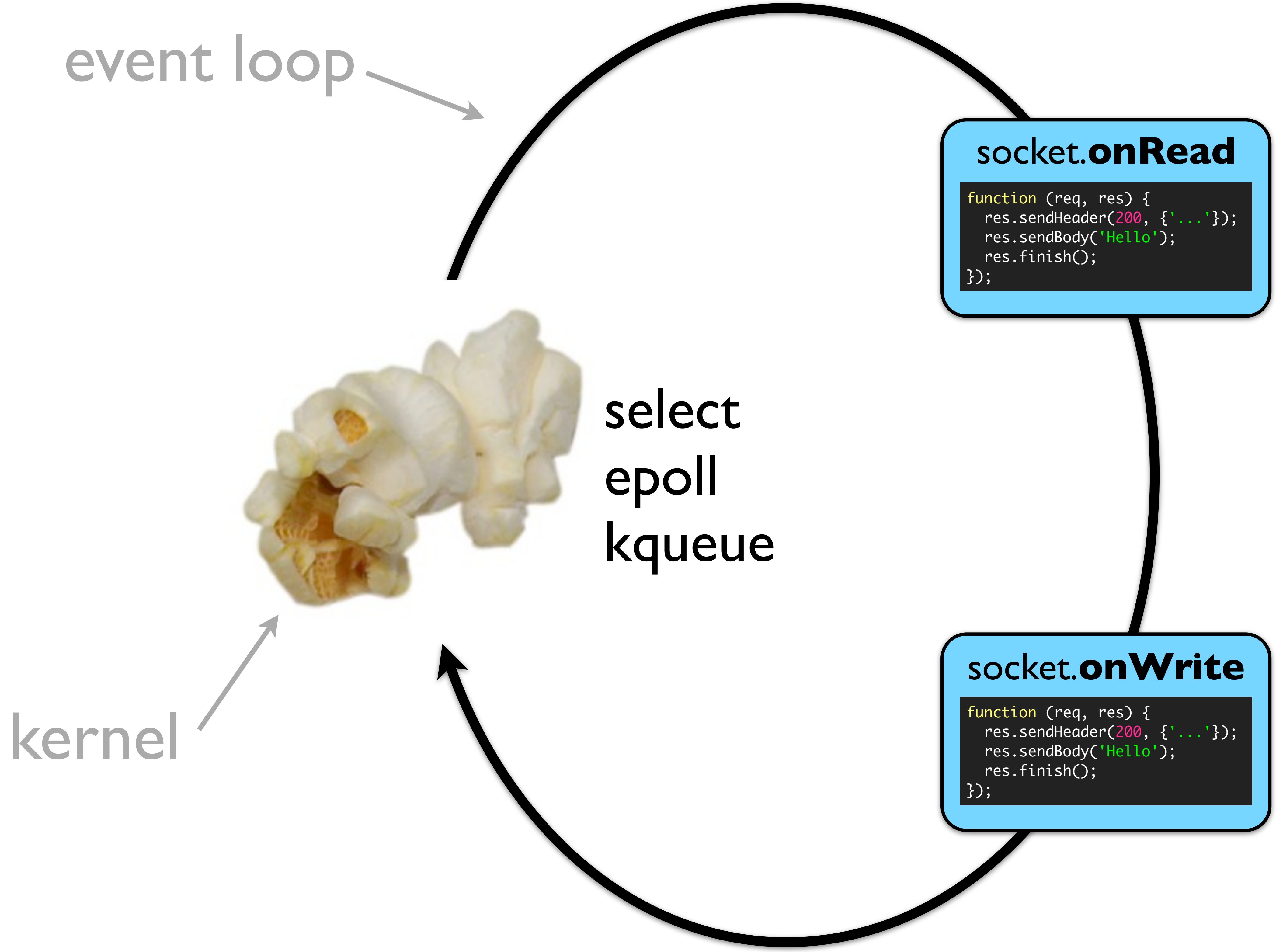


```
function (req, res) {
  res.sendHeader(200, {'...'});
  res.sendBody('Hello');
  res.finish();
});
```

```
function (req, res) {
  res.sendHeader(200, {'...'});
  res.sendBody('Hello');
  res.finish();
});
```

(Artistic representation)

Source: http://pack283.chieftarhe.org/wp-content/uploads/2008/08/butterfly_kernel.jpg

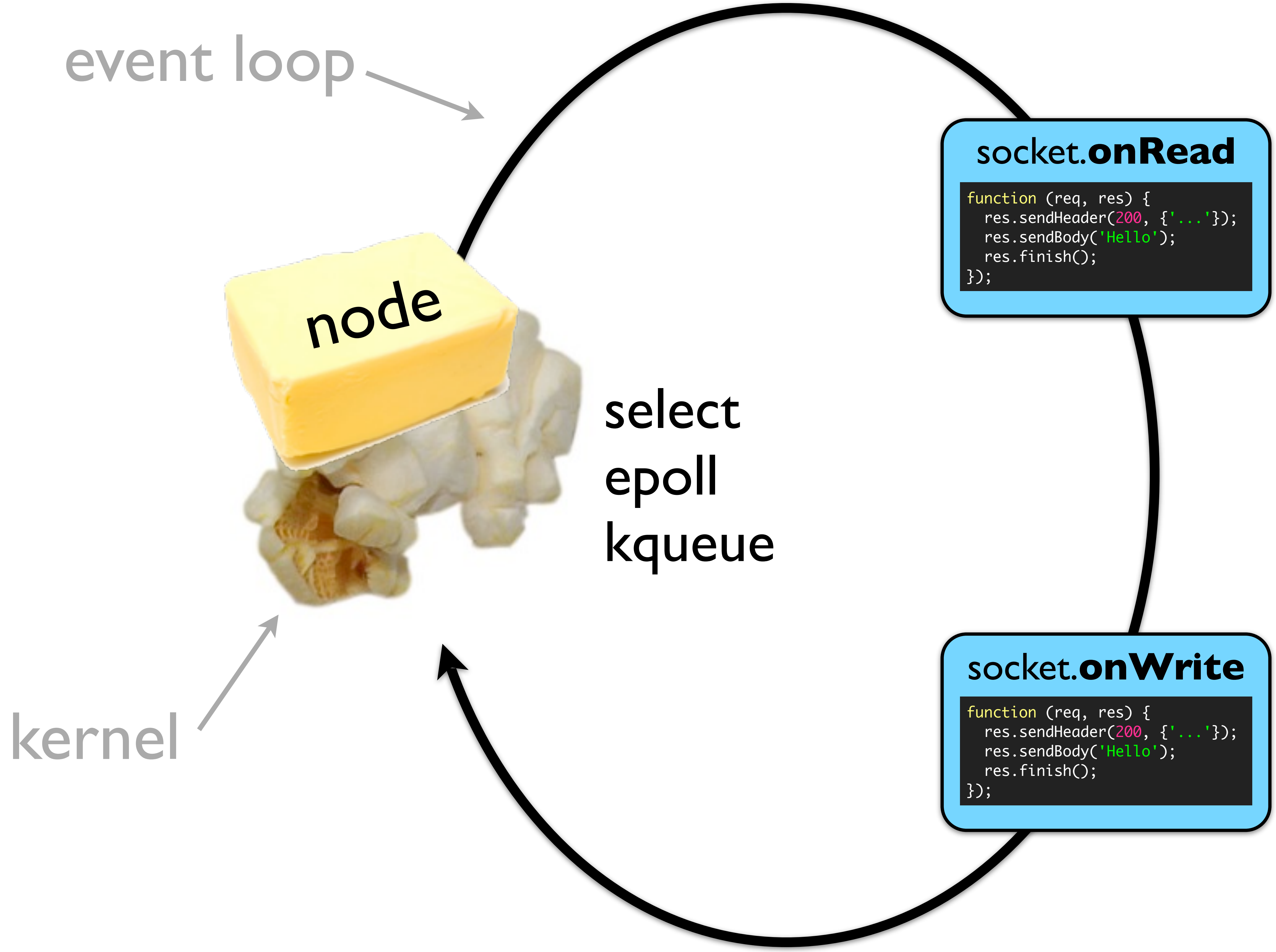


```
function (req, res) {
  res.sendHeader(200, {'...'});
  res.sendBody('Hello');
  res.finish();
});
```

```
function (req, res) {
  res.sendHeader(200, {'...'});
  res.sendBody('Hello');
  res.finish();
});
```

(Artistic representation)

Source: http://pack283.chieftarhe.org/wp-content/uploads/2008/08/butterfly_kernel.jpg

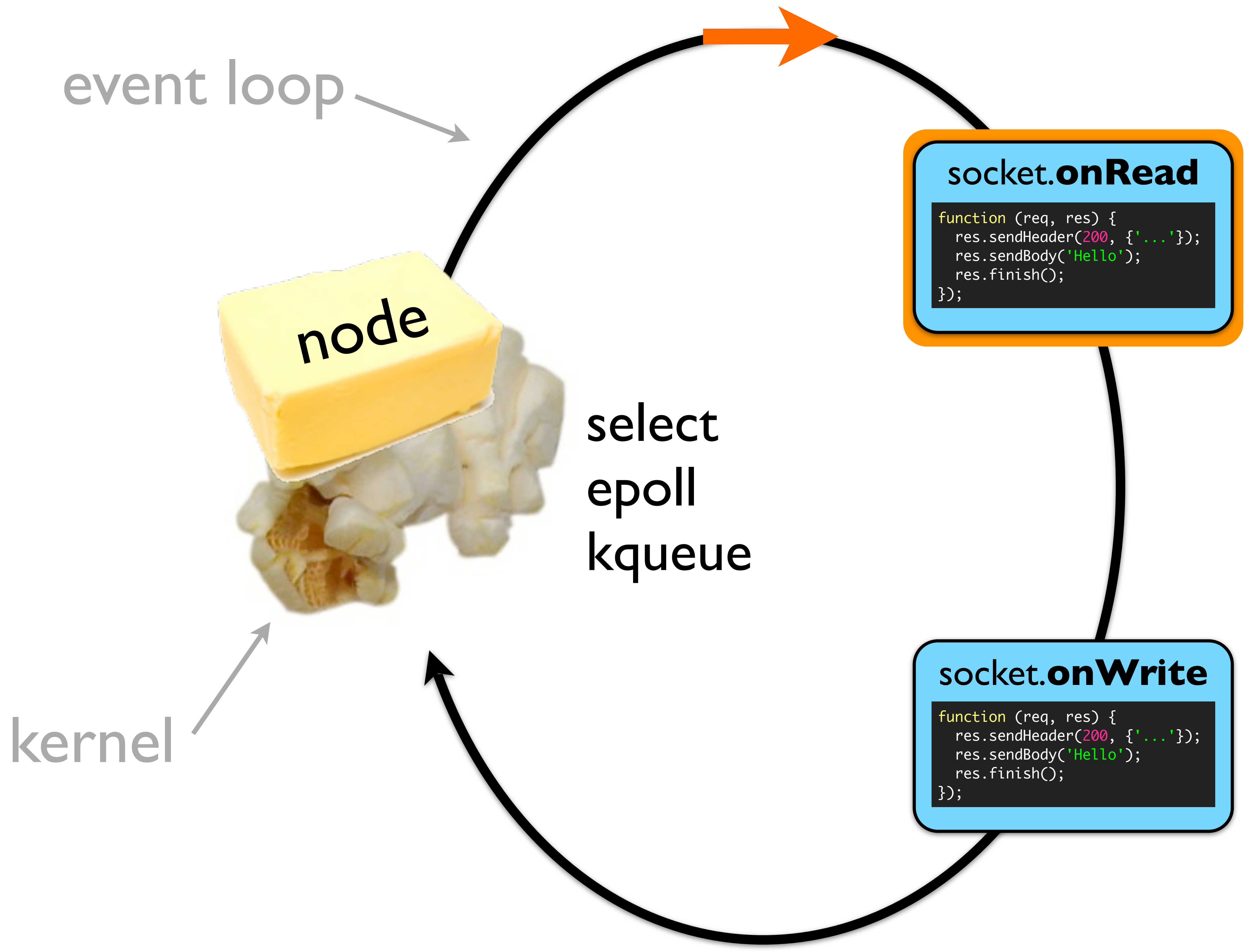


```
function (req, res) {
  res.sendHeader(200, {'...'});
  res.sendBody('Hello');
  res.finish();
});
```

```
function (req, res) {
  res.sendHeader(200, {'...'});
  res.sendBody('Hello');
  res.finish();
});
```

(Artistic representation)

Source: http://pack283.chieftarhe.org/wp-content/uploads/2008/08/butterfly_kernel.jpg

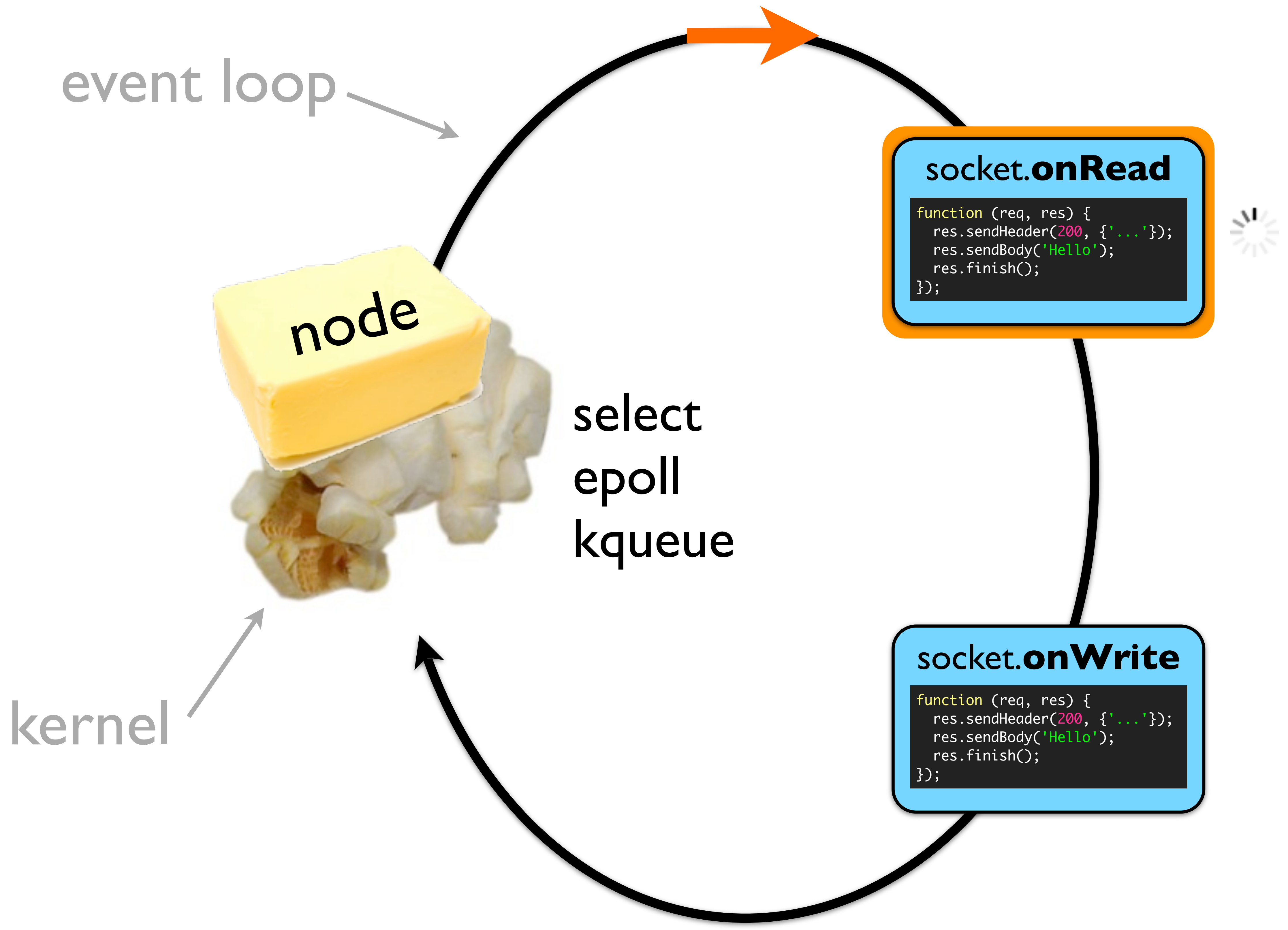


```
function (req, res) {
  res.sendHeader(200, {'...'});
  res.sendBody('Hello');
  res.finish();
});
```

```
function (req, res) {
  res.sendHeader(200, {'...'});
  res.sendBody('Hello');
  res.finish();
});
```

(Artistic representation)

Source: http://pack283.chieftarhe.org/wp-content/uploads/2008/08/butterfly_kernel.jpg

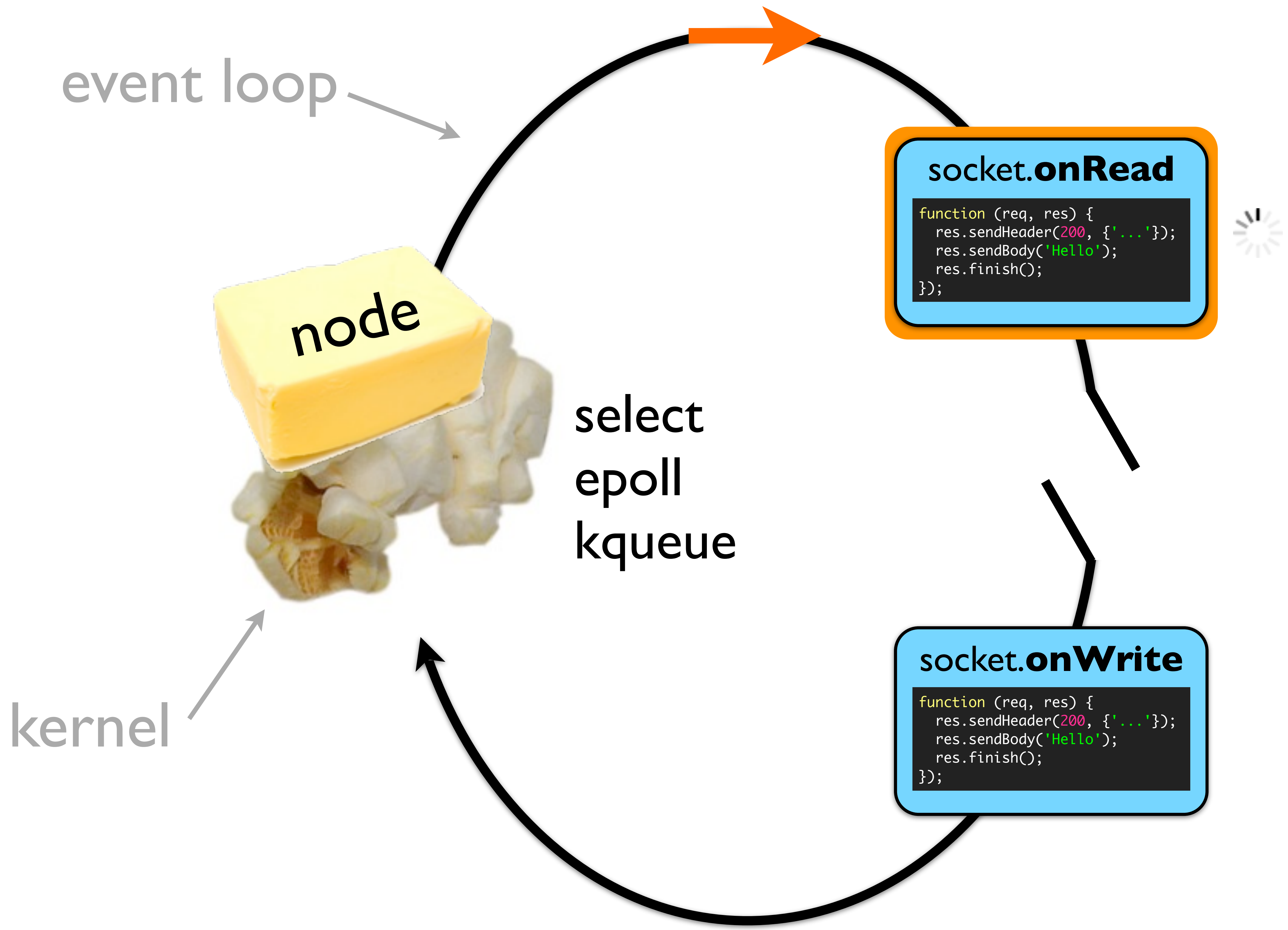


```
function (req, res) {
  res.sendHeader(200, {'...'});
  res.sendBody('Hello');
  res.finish();
});
```

```
function (req, res) {
  res.sendHeader(200, {'...'});
  res.sendBody('Hello');
  res.finish();
});
```

(Artistic representation)

Source: http://pack283.chieftarhe.org/wp-content/uploads/2008/08/butterfly_kernel.jpg



```
function (req, res) {
  res.sendHeader(200, {'...'});
  res.sendBody('Hello');
  res.finish();
});
```

```
function (req, res) {
  res.sendHeader(200, {'...'});
  res.sendBody('Hello');
  res.finish();
});
```

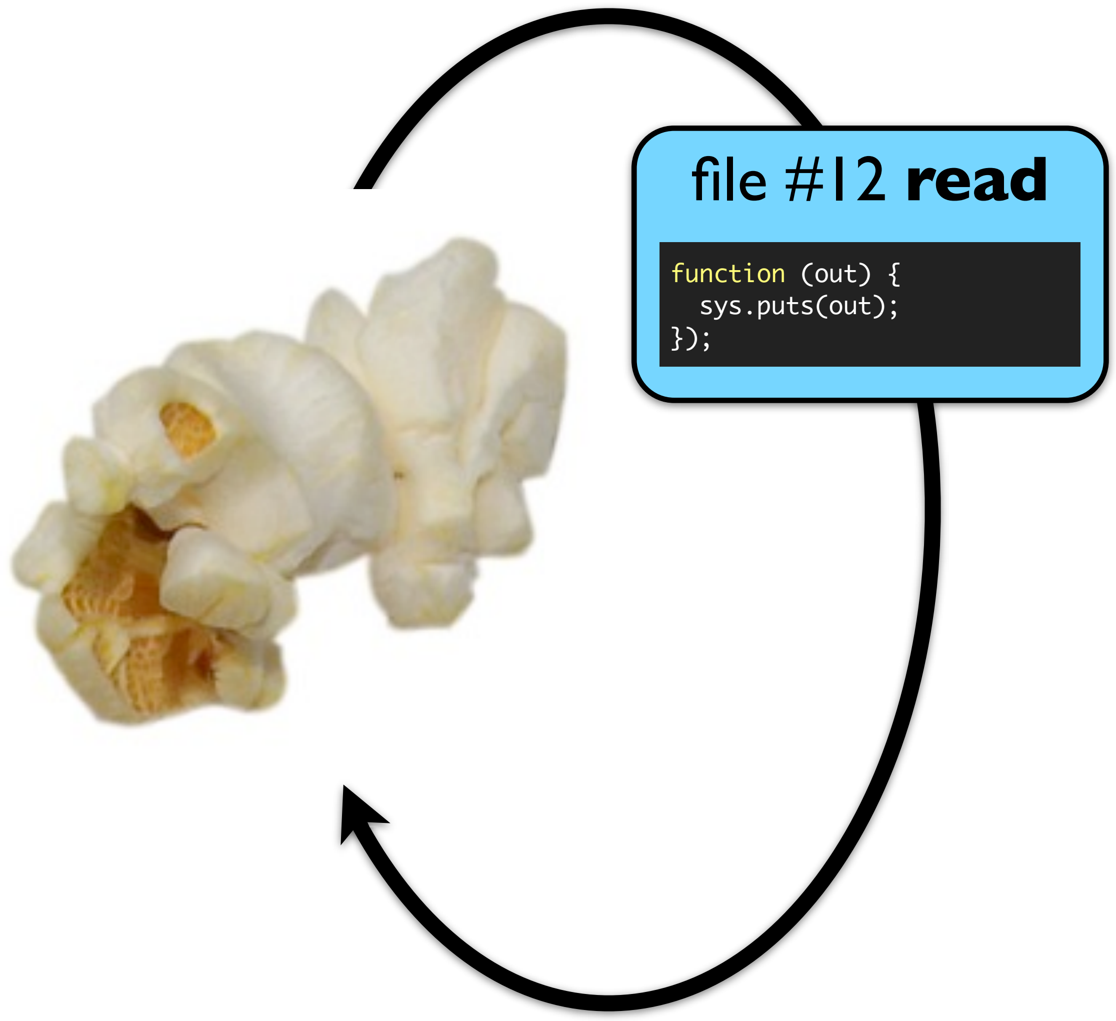
(Artistic representation)

Source: http://pack283.chieftarhe.org/wp-content/uploads/2008/08/butterfly_kernel.jpg

nodeJS

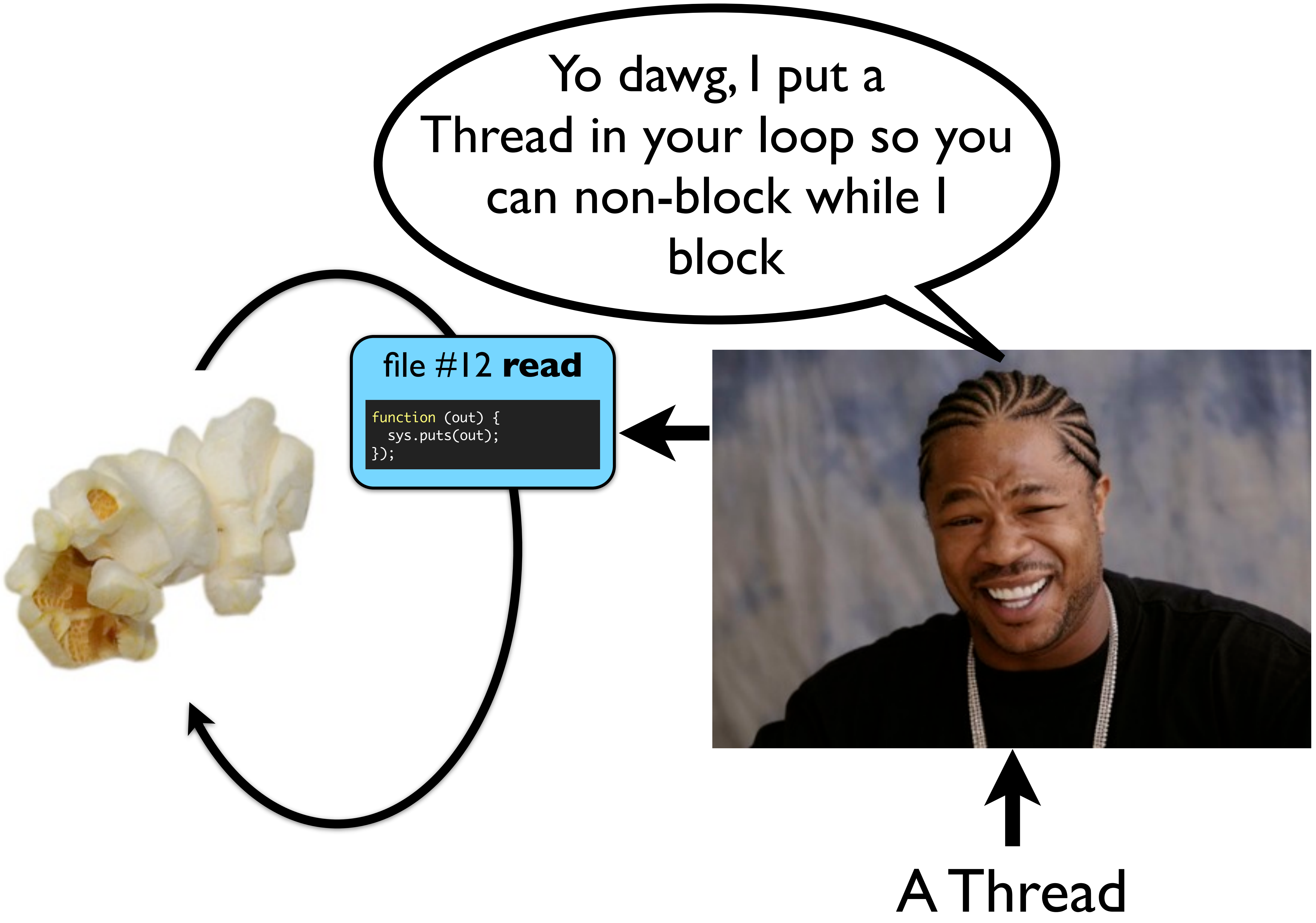
To provide a **purely evented, non-blocking** infrastructure to script **highly concurrent** programs.

```
sys.exec("ls -l /").addCallback(function(out) {
  sys.puts(out);
});
```



```
function (out) {
  sys.puts(out);
});
```

http://static.stuff.co.nz/1251812102/685/2824685.jpg


Yo dawg, I put a Thread in your loop so you can non-block while I block
file #12 **read**
A Thread

```
function (out) {
  sys.puts(out);
});
```

http://static.stuff.co.nz/1251812102/685/2824685.jpg

But why JavaScript ?

```
$(document).ready(function () {
  $("p").text("DOM is loaded");
});



$.getJSON("/image.json", function(data){
  $("<img/>").attr("src", data.url)
             .appendTo("#images");
  });
});
```

```
$(document).ready(function () {
  $("p").text("DOM is loaded");
});
```

← **Callback**

```
$.getJSON("/image.json", function(data){
  $("<img/>").attr("src", data.url)
             .appendTo("#images");
  });
});
```

← **Callback**

```
http.createServer(function (req, res) {
  res.sendHeader(200,
                {'Content-Type': 'text/plain'});
  res.sendBody('Hello');
  res.finish();
}).listen(8000);
```

... so what?

>10 000

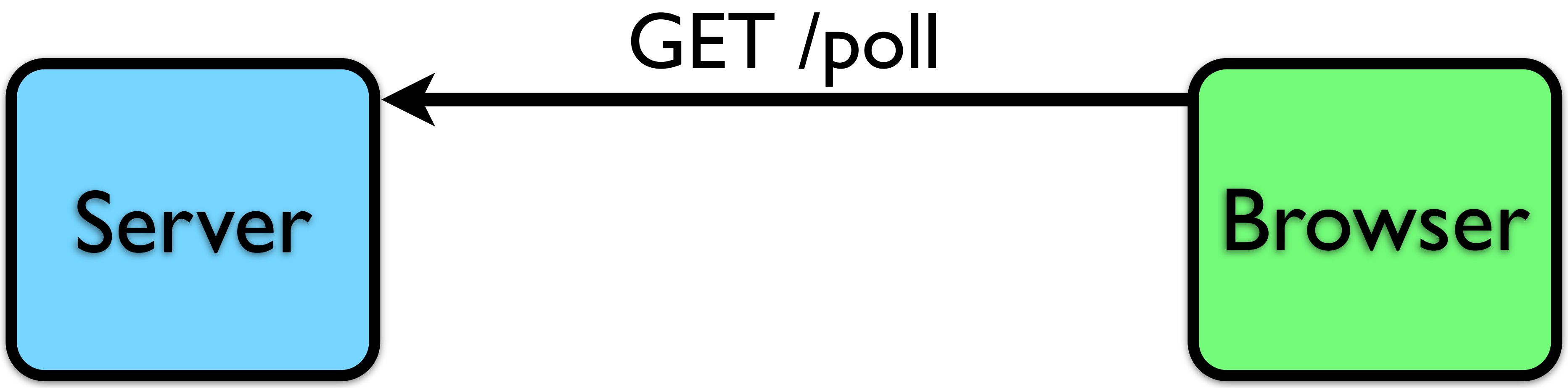
concurrent connections

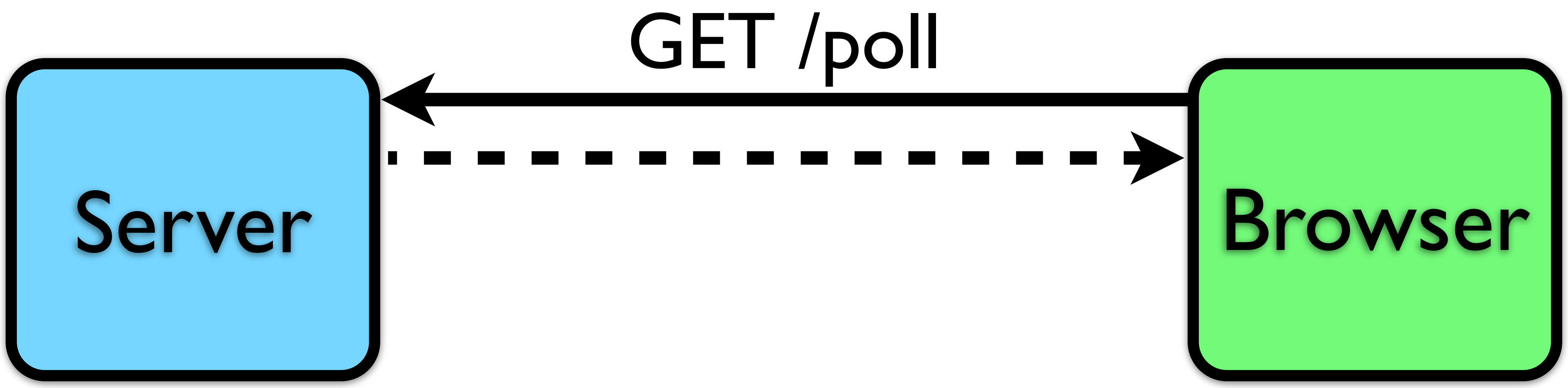
Source: http://en.wikipedia.org/wiki/C10k_problem

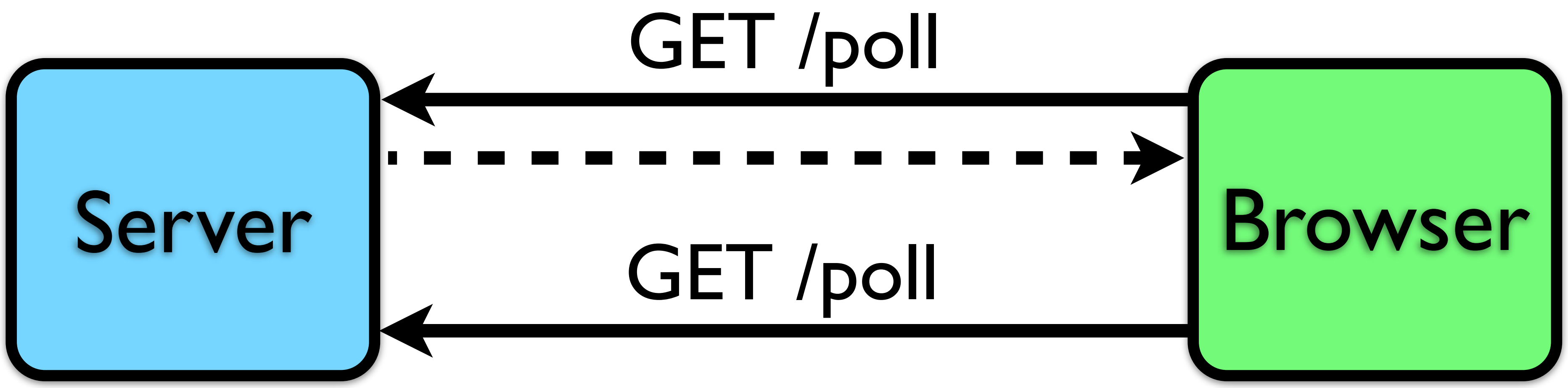
Disinfects
Comet
with Bleach
DISINFECTANT CLEANSER
Bleaches Tough Stains
NET WT 21 OZ
(1.31 LB) 595g
KEEP OUT OF REACH OF CHILDREN
CAUTION: MAY CAUSE EYE AND SKIN IRRITATION.
READ PRECAUTIONS ON BACK.

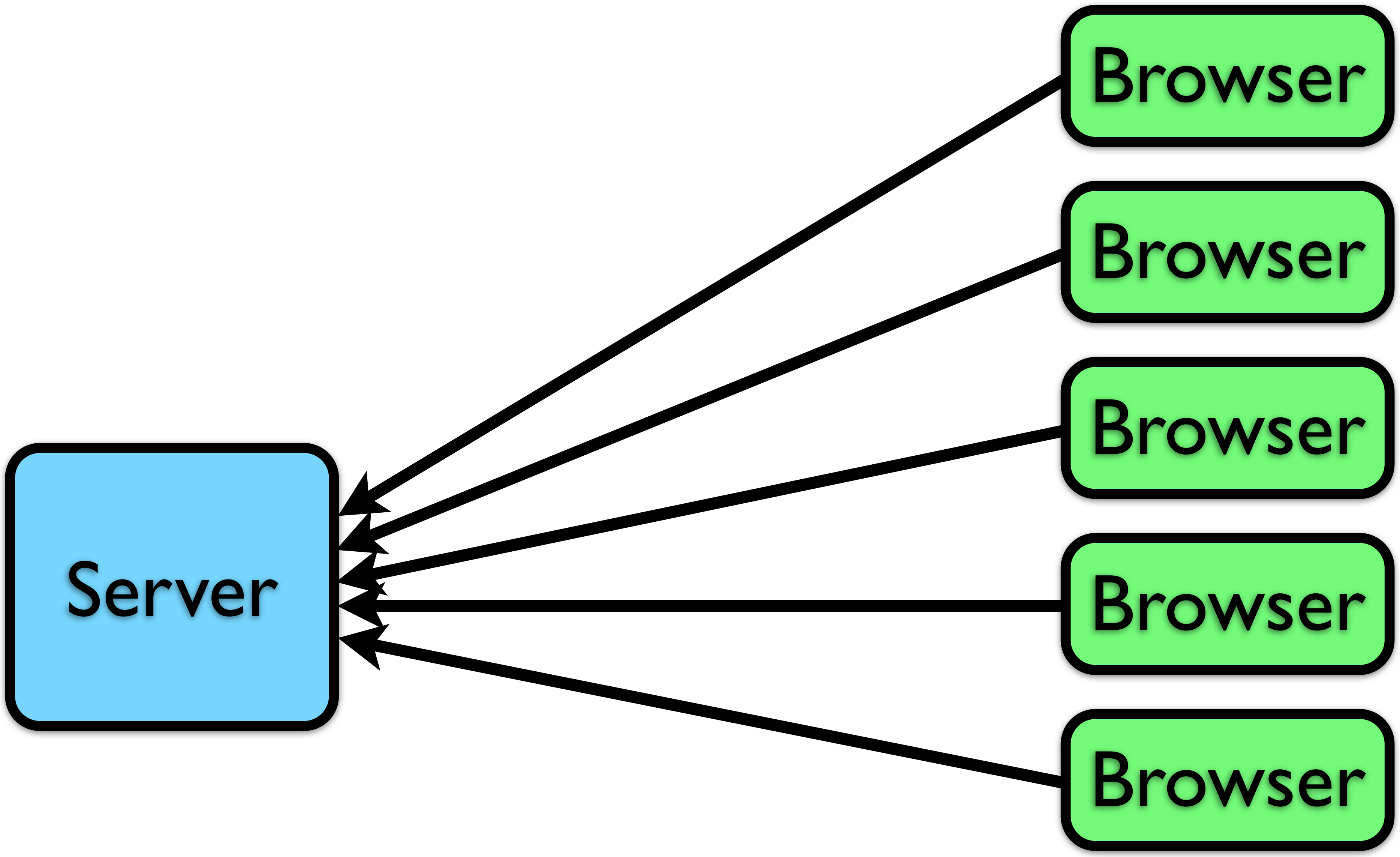
Chat app!

Long Poll


Server
GET /poll
Browser


GET /poll
Server
Browser


GET /poll
Server
Browser
GET /poll


Server
Browser
Browser
Browser
Browser
Browser

Demo

HTTP

HTTP Server

```
http.createServer(function (req, res) {
  res.sendHeader(200,
                {'Content-Type': 'text/plain'});
  res.sendBody('Hello');
  res.finish();
}).listen(8000);
```

HTTP Client

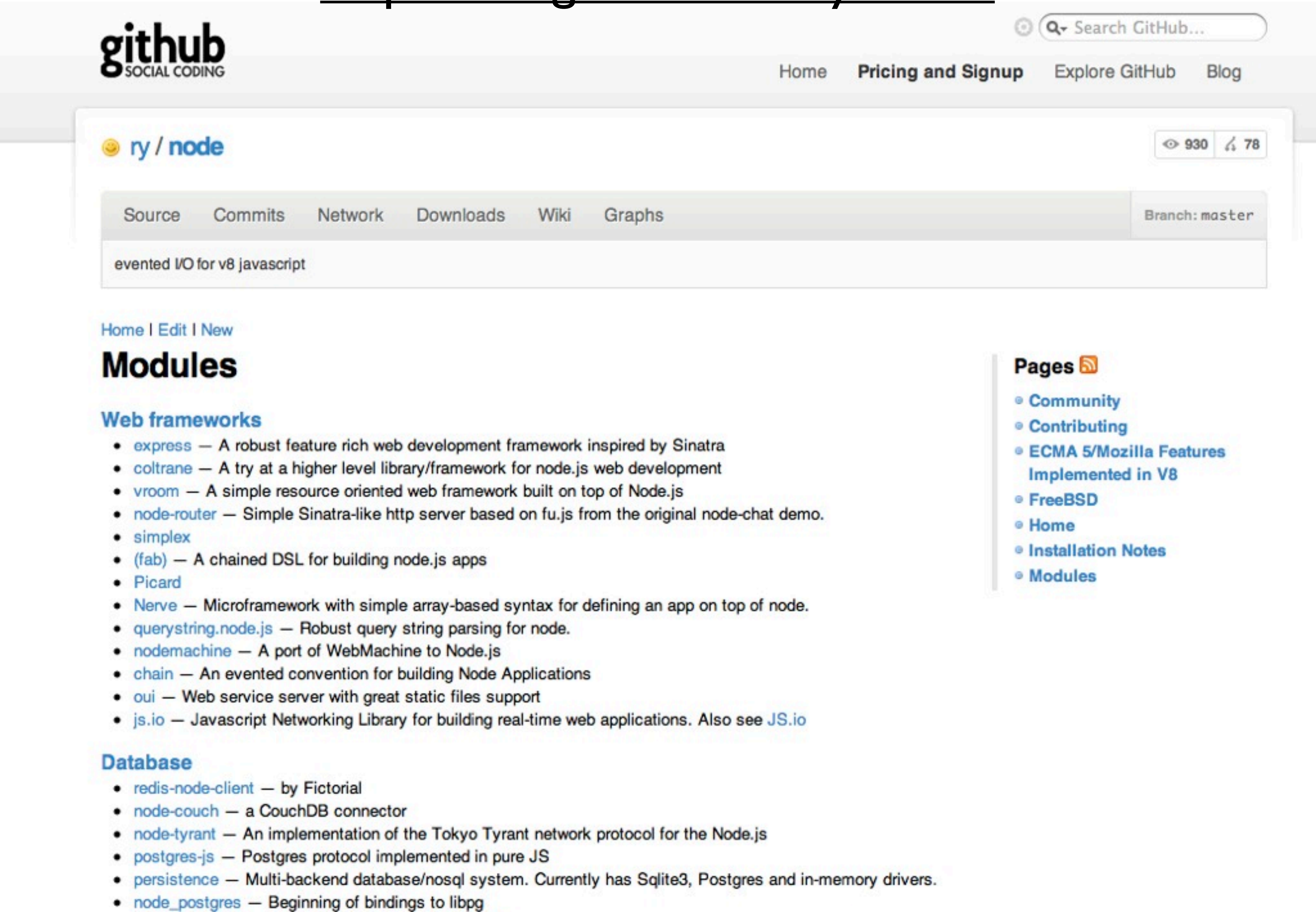
```
var sys = require("sys"),
    http = require("http");

var google = http.createClient(80, "www.google.com");
var request = google.request("GET", "/",
                             {"host": "www.google.com"});

request.finish(function (response) {
  sys.puts("STATUS: " + response.statusCode);
  sys.puts("HEADERS: " + JSON.stringify(response.headers));
  response.setBodyEncoding("utf8");
  response.addListener("body", function (chunk) {
    sys.puts("BODY: " + chunk);
  });
});
```

Streaming

Multipart Streaming

```
var multipart = require("multipart");
var stream = new multipart.Stream(options);
var parts = {};

stream.addListener("part", function (part) {
  var buffer = "";

  part.addListener("body", function (chunk) {
    buffer = buffer + chunk;
  });

  part.addListener("complete", function () {
    parts[part.name] = buffer;
  });
});

stream.addListener("complete", function () {
  // The parts object now contains all parts and data
});
```

http://wiki.github.com/ry/node/



github SOCIAL CODING

Search GitHub...

Home | Pricing and Signup | Explore GitHub | Blog

ry / node

930 | 78

Source | Commits | Network | Downloads | Wiki | Graphs | Branch: master

evented I/O for v8 javascript

Home | Edit | New

Modules

Web frameworks

- express — A robust feature rich web development framework inspired by Sinatra
- coltrane — A try at a higher level library/framework for node.js web development
- vroom — A simple resource oriented web framework built on top of Node.js
- node-router — Simple Sinatra-like http server based on fu.js from the original node-chat demo.
- simplex
- (fab) — A chained DSL for building node.js apps
- Picard
- Nerve — Microframework with simple array-based syntax for defining an app on top of node.
- querystring.node.js — Robust query string parsing for node.
- nodemachine — A port of WebMachine to Node.js
- chain — An evented convention for building Node Applications
- oui — Web service server with great static files support
- js.io — Javascript Networking Library for building real-time web applications. Also see JS.io

Database

- redis-node-client — by Fictorial
- node-couch — a CouchDB connector
- node-tyrant — An implementation of the Tokyo Tyrant network protocol for the Node.js
- postgres-js — Postgres protocol implemented in pure JS
- persistence — Multi-backend database/nosql system. Currently has Sqlite3, Postgres and in-memory drivers.
- node_postgres — Beginning of bindings to libpg

Pages

- Community
- Contributing
- ECMA 5/Mozilla Features Implemented in V8
- FreeBSD
- Home
- Installation Notes
- Modules

Twisted Matrix Labs
Building the engine of your internet

http://twistedmatrix.com

Ruby/EventMachine
Fast Network I/O and Event Management for Ruby Programmers

http://rubyeventmachine.com

Apache MINA

libev

http://mina.apache.org

http://software.schmorp.de/pkg/libev.html

nodeJS

http://nodejs.org/

The C10K problem

http://www.kegel.com/c10k.html

Questions ?

http://macournoyer.com

http://talkerapp.com