Total messages 311035

nearForm

JS Everywhere

by **Matteo Collina**

Credits http://github.com/nodebots

This blink like it's 1998

what js is good for

```
if (el.style.visibility === 'hidden') {
  el.style.visibility = 'visible'
} else {
  el.style.visibility = 'hidden'
}
```

Flow control is built-in

If applications sends too much data, we can slow them down with backpressure!

```
function process(data, cb) {
  /* oh yeah, we are processing one
     thing at a time
     or we can do batches! */
  setTimeout(cb, 500)
}
```

Node.js has streams too!

```
source.pipe(dest).pipe(source)
```

Microservices

AWS Lambda

Netflix

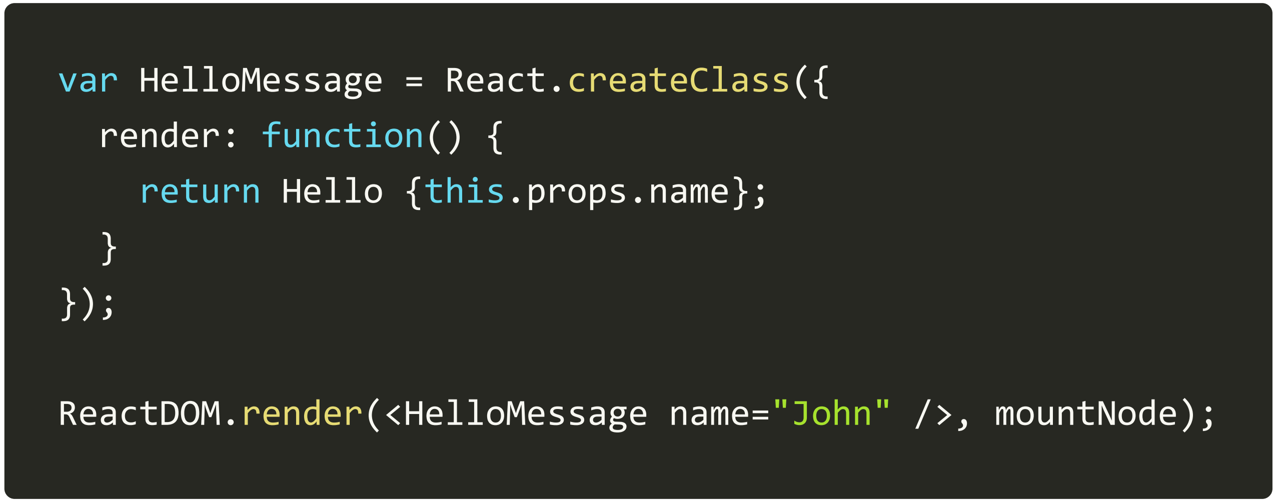
Uber

... and many others!

Seneca

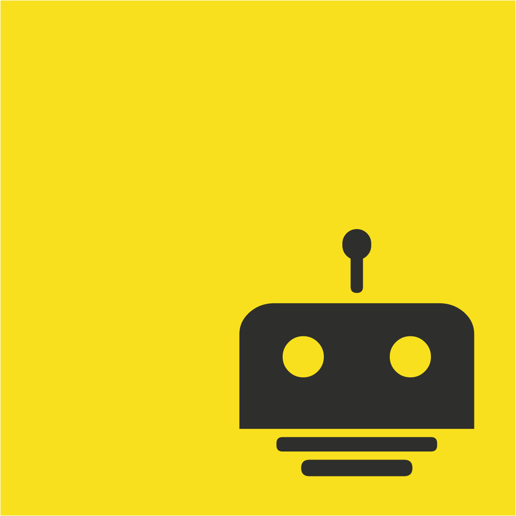
The best community ever

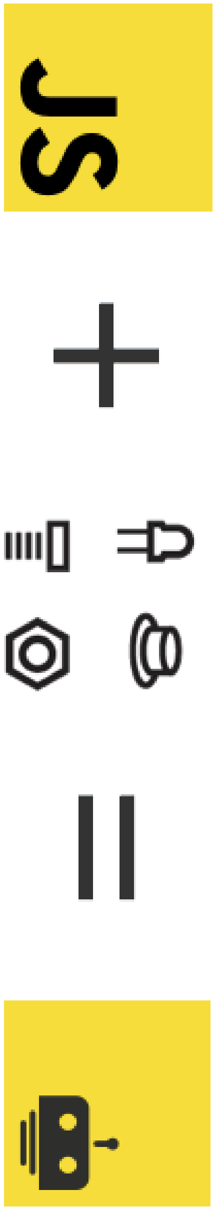
we love you too, npm

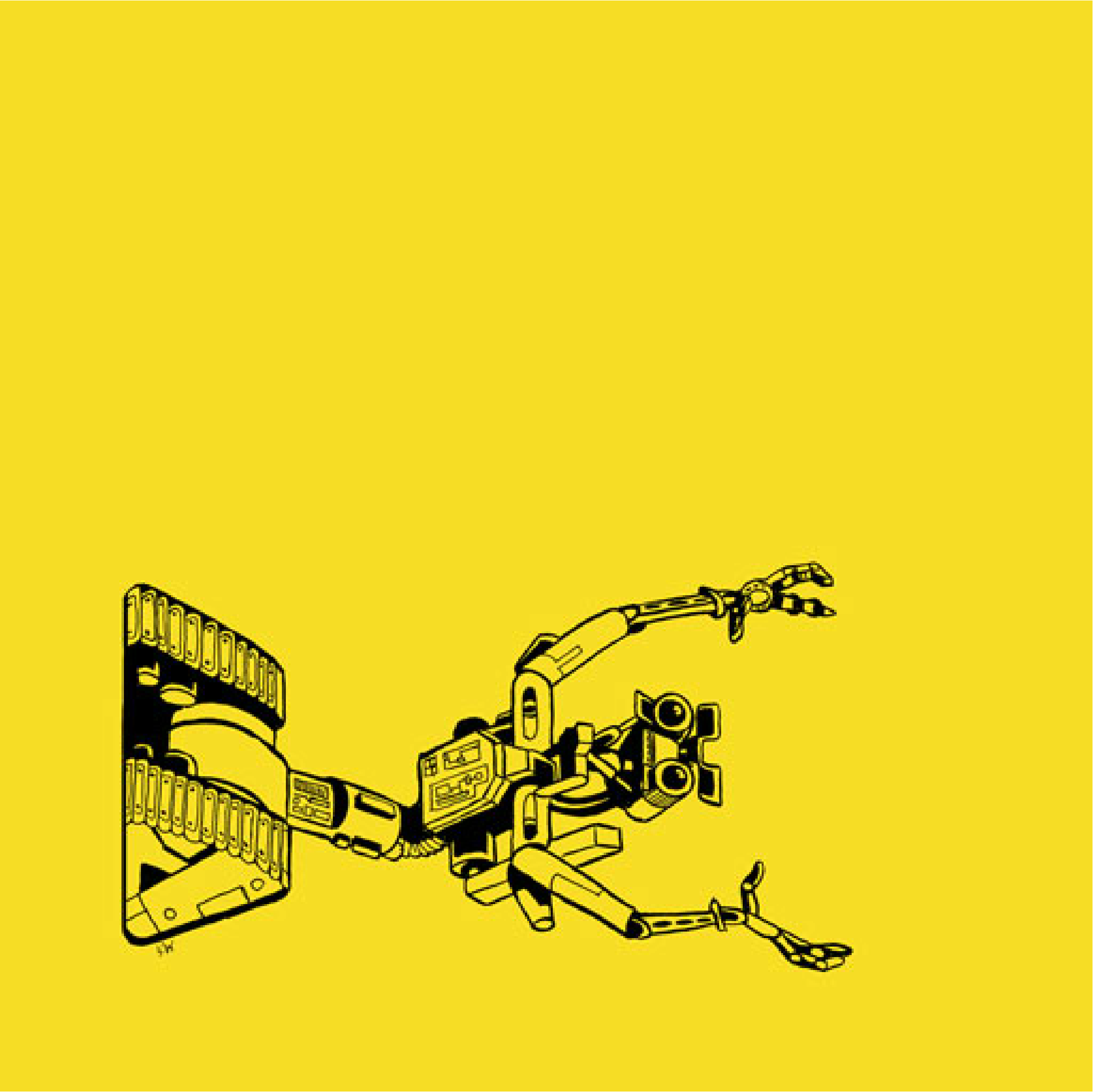


```
var HelloMessage = React.createClass({
  render: function() {
    return Hello {this.props.name};
  }
});

ReactDOM.render(<HelloMessage name="John" />, mountNode);
```

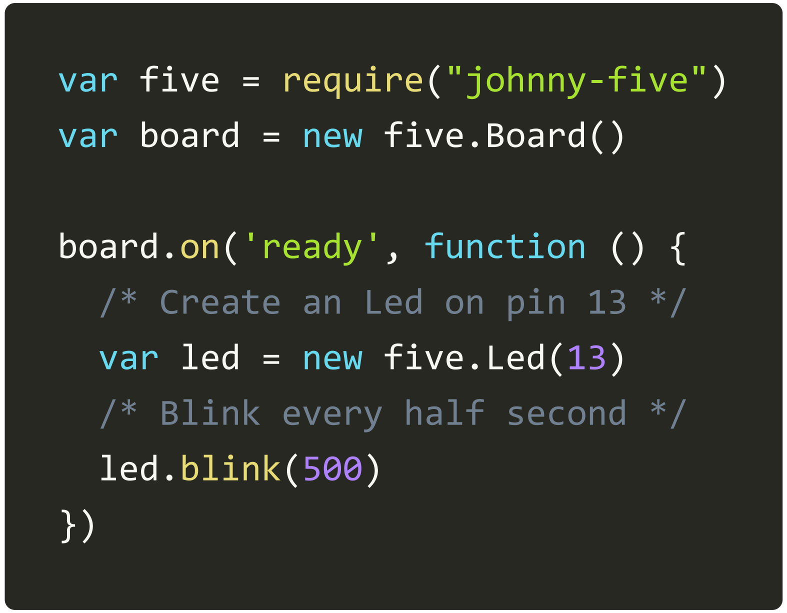
NodeBots




JS
+
=



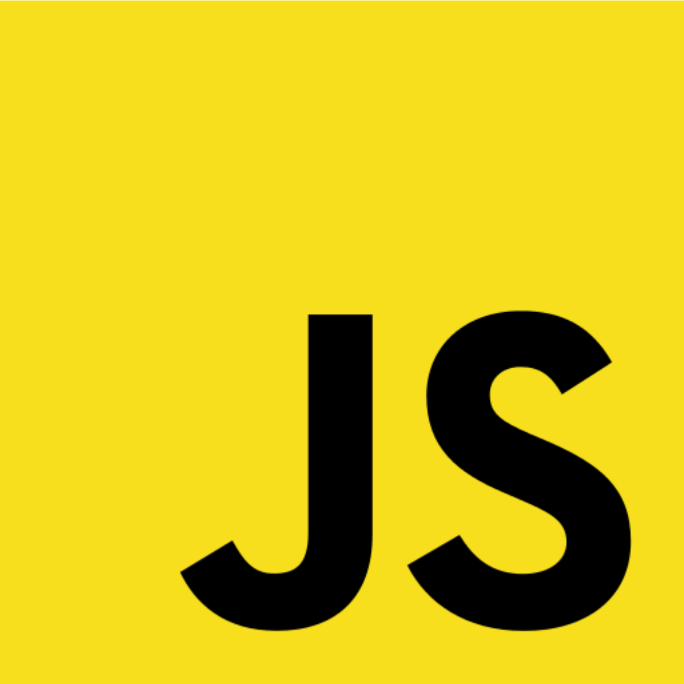
Johnny Five



```
var five = require("johnny-five")
var board = new five.Board()

board.on('ready', function () {
  /* Create an Led on pin 13 */
  var led = new five.Led(13)
  /* Blink every half second */
  led.blink(500)
})
```

You should not see this!

Universal JS



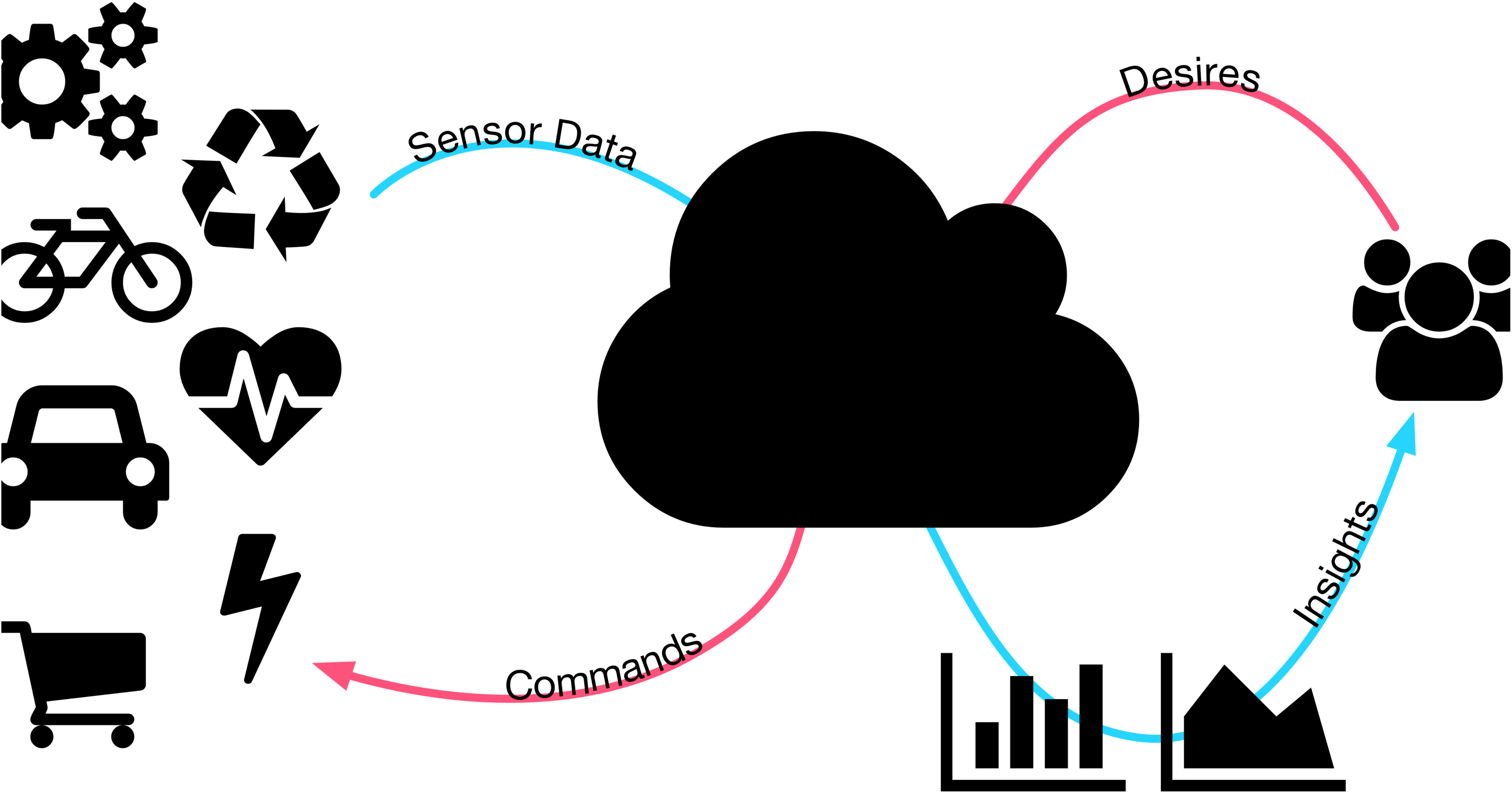
use the same libraries in browsers, servers, and devices

simple architecture, fewer components

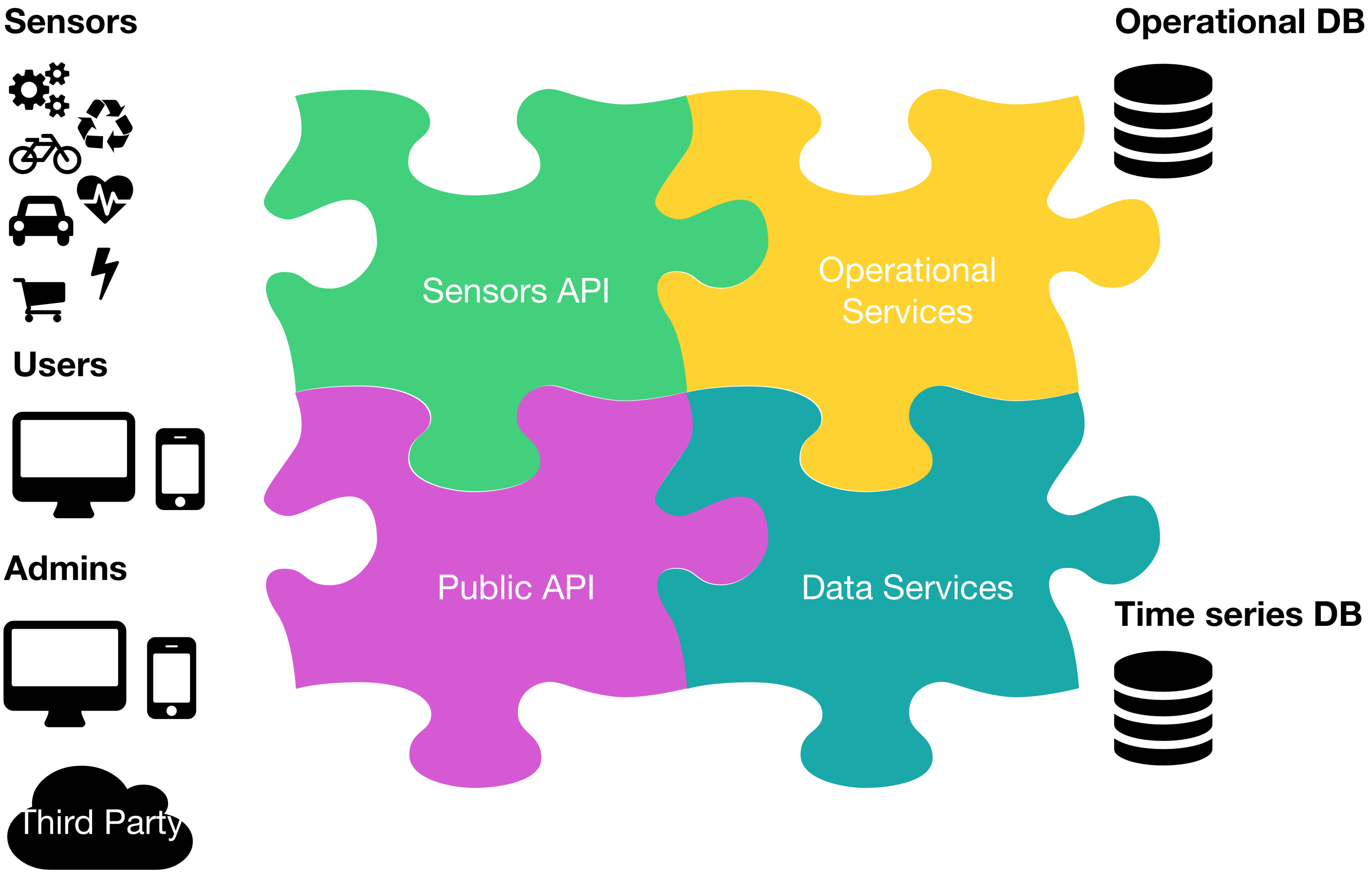
can we use all of this to build an IoT system?

An IoT system

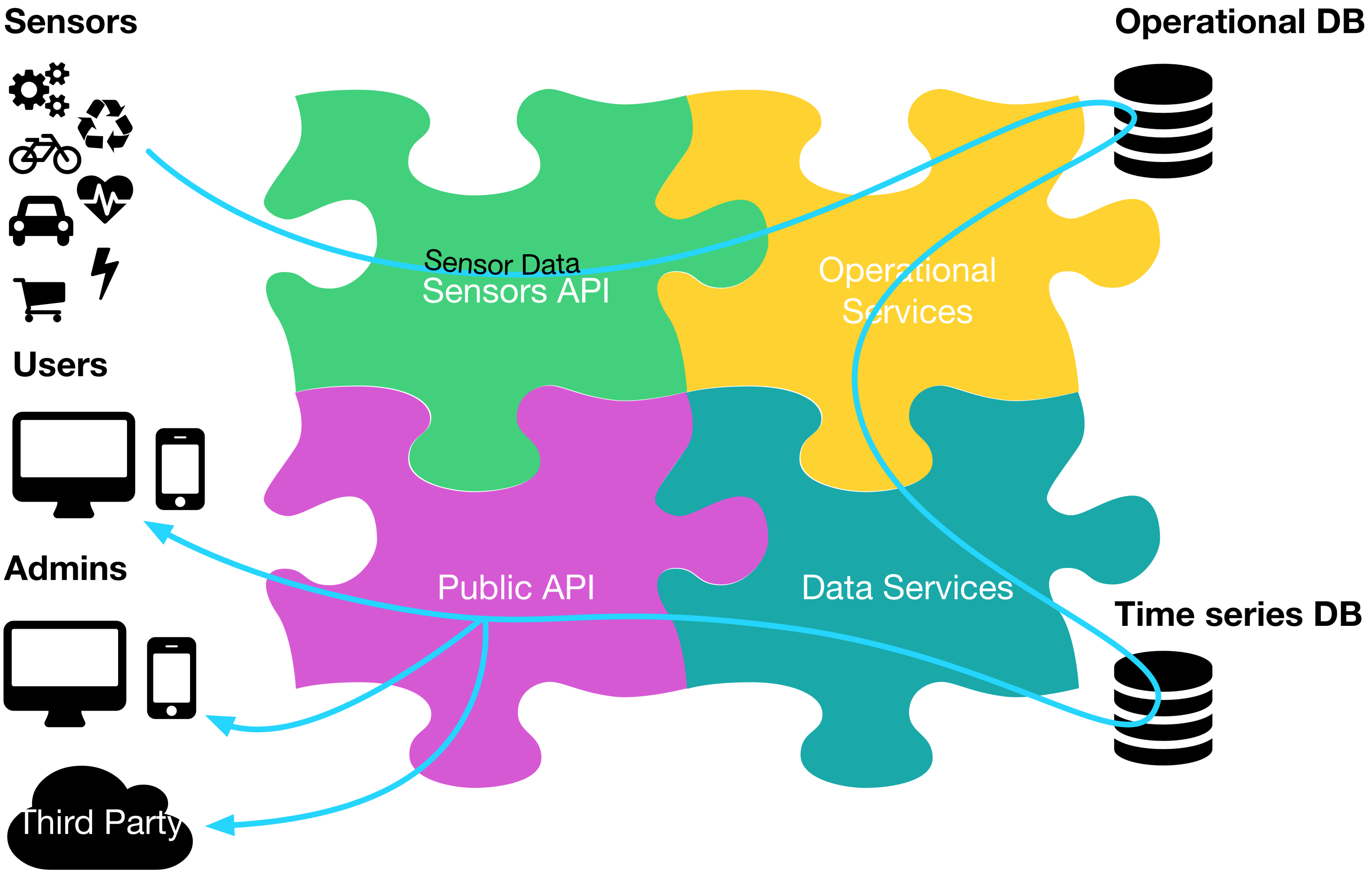
What is an IoT system?



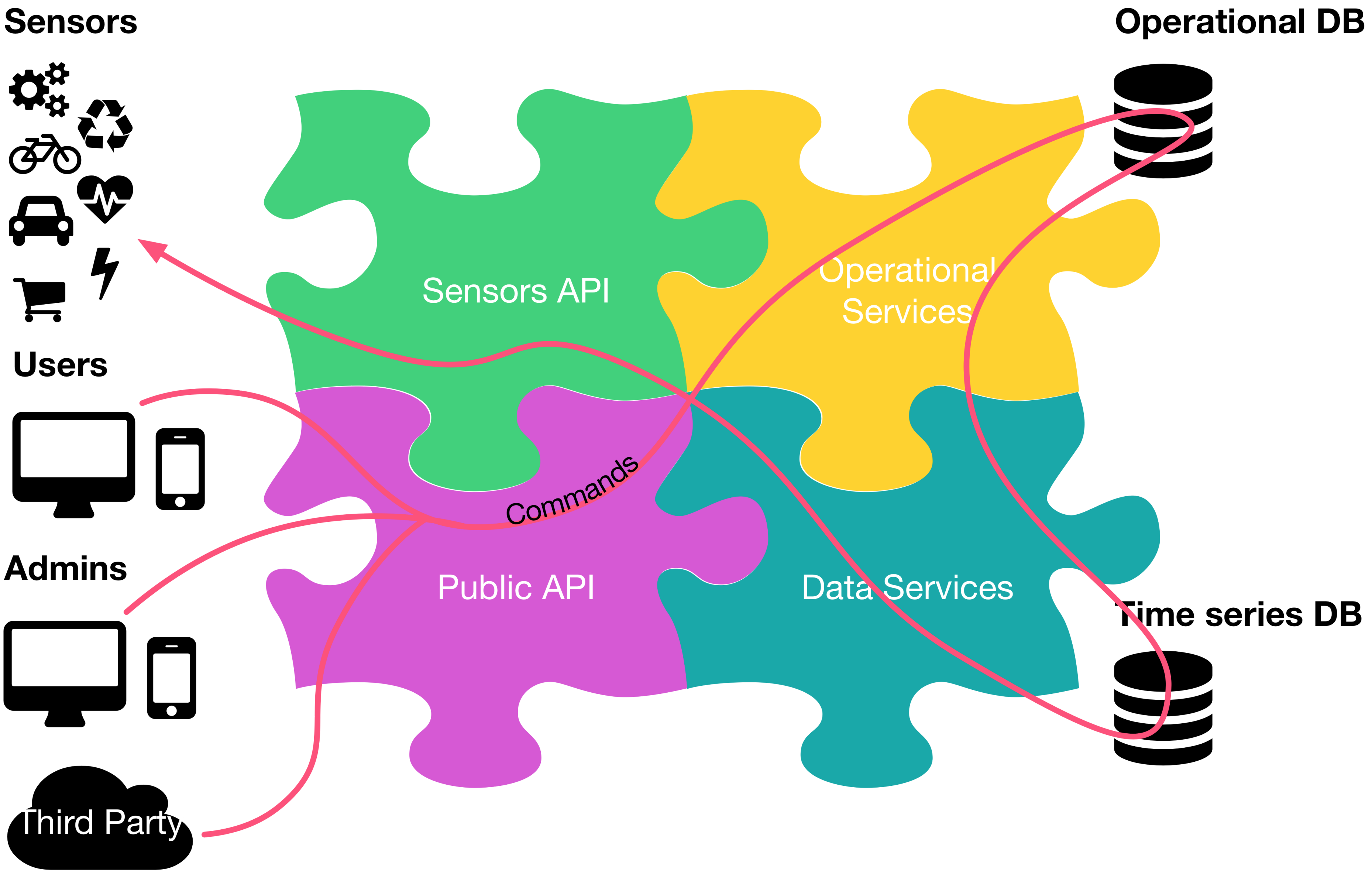
Architecture

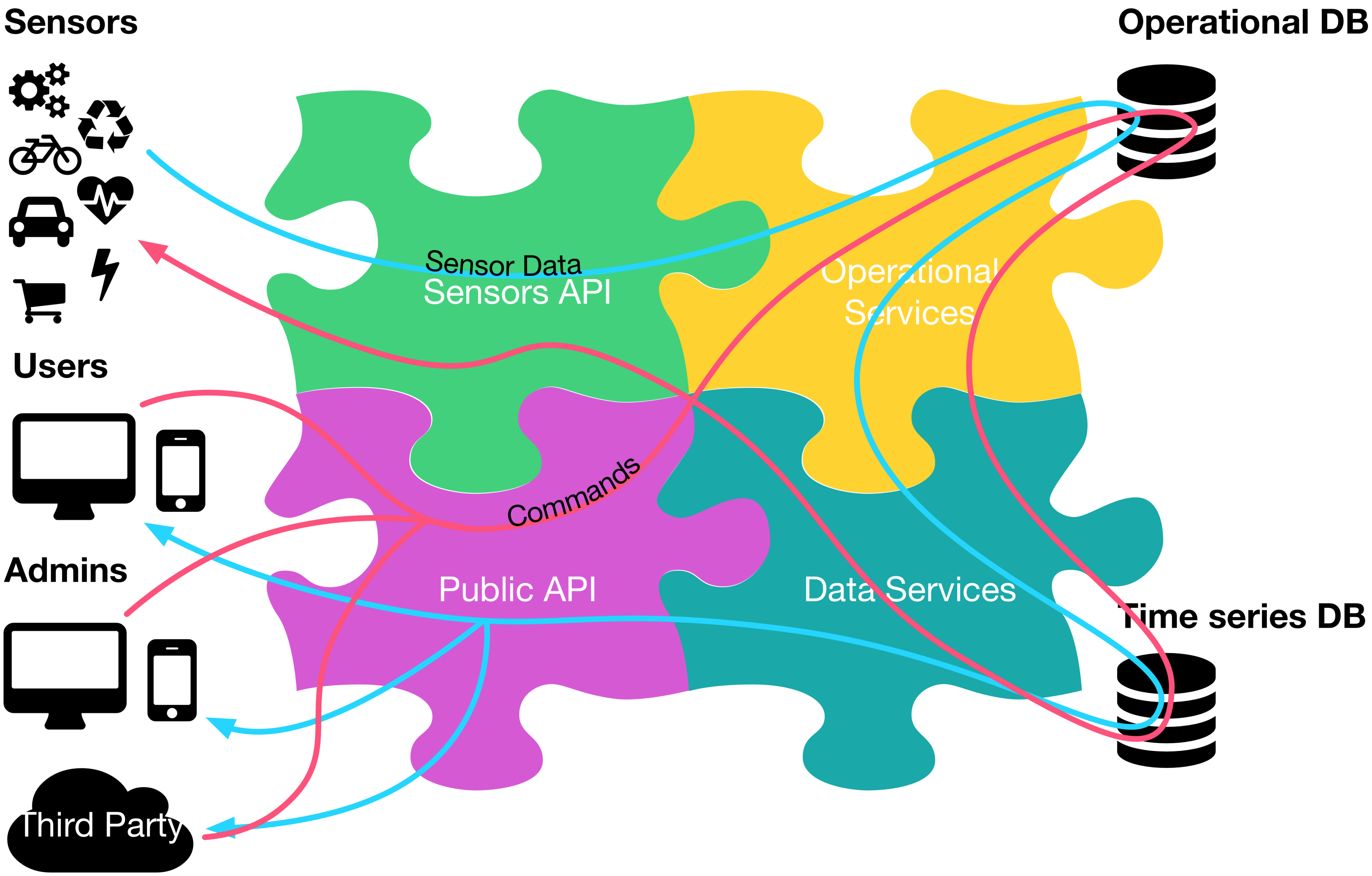


Sensor Data Flow



Commands Flow




Sensors
Users
Admins
Third Party
Sensor Data
Sensors API
Commands
Public API
Operational Services
Data Services
Operational DB
Time series DB

Problems

Latency

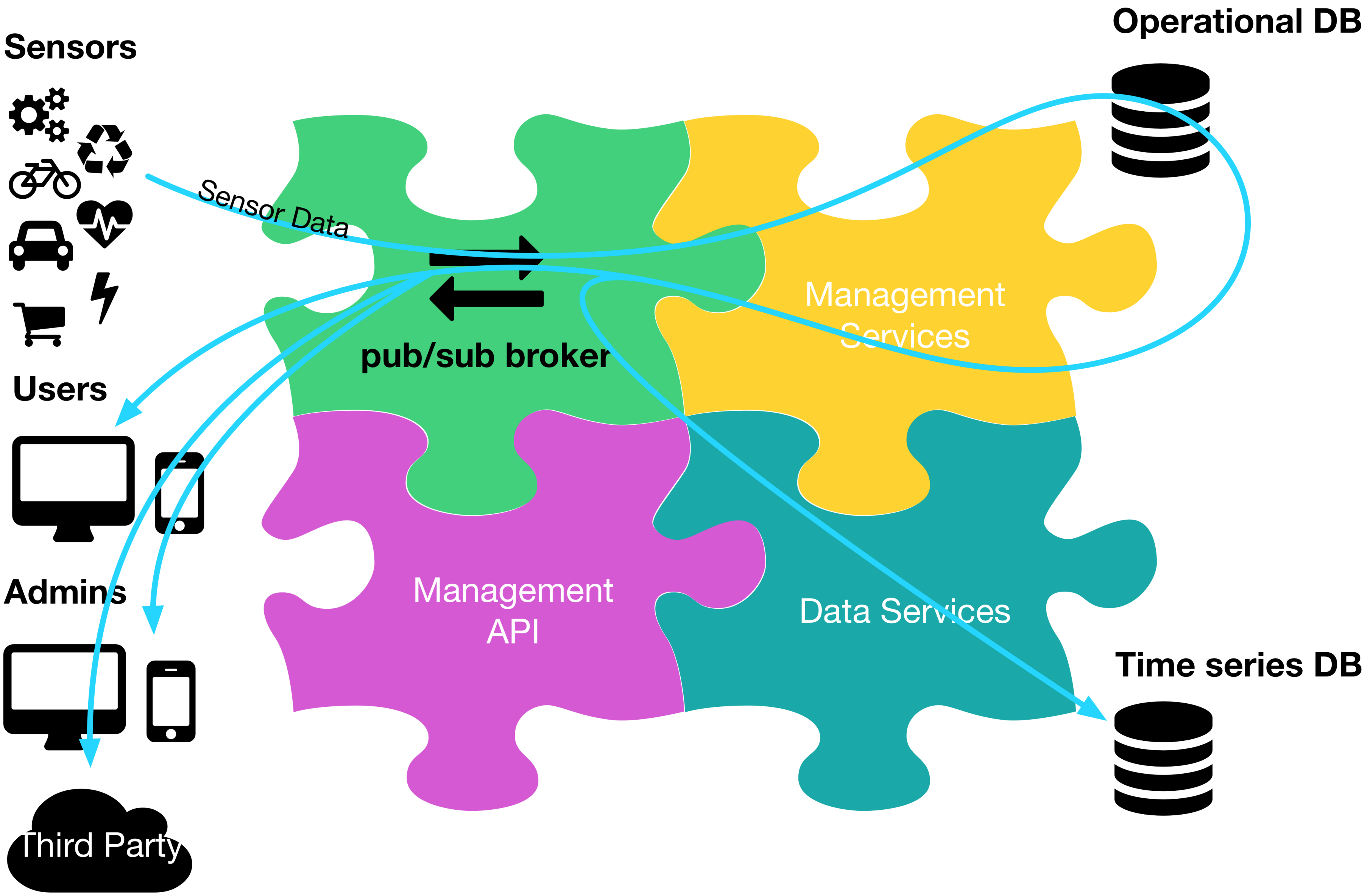
no reuse

complex

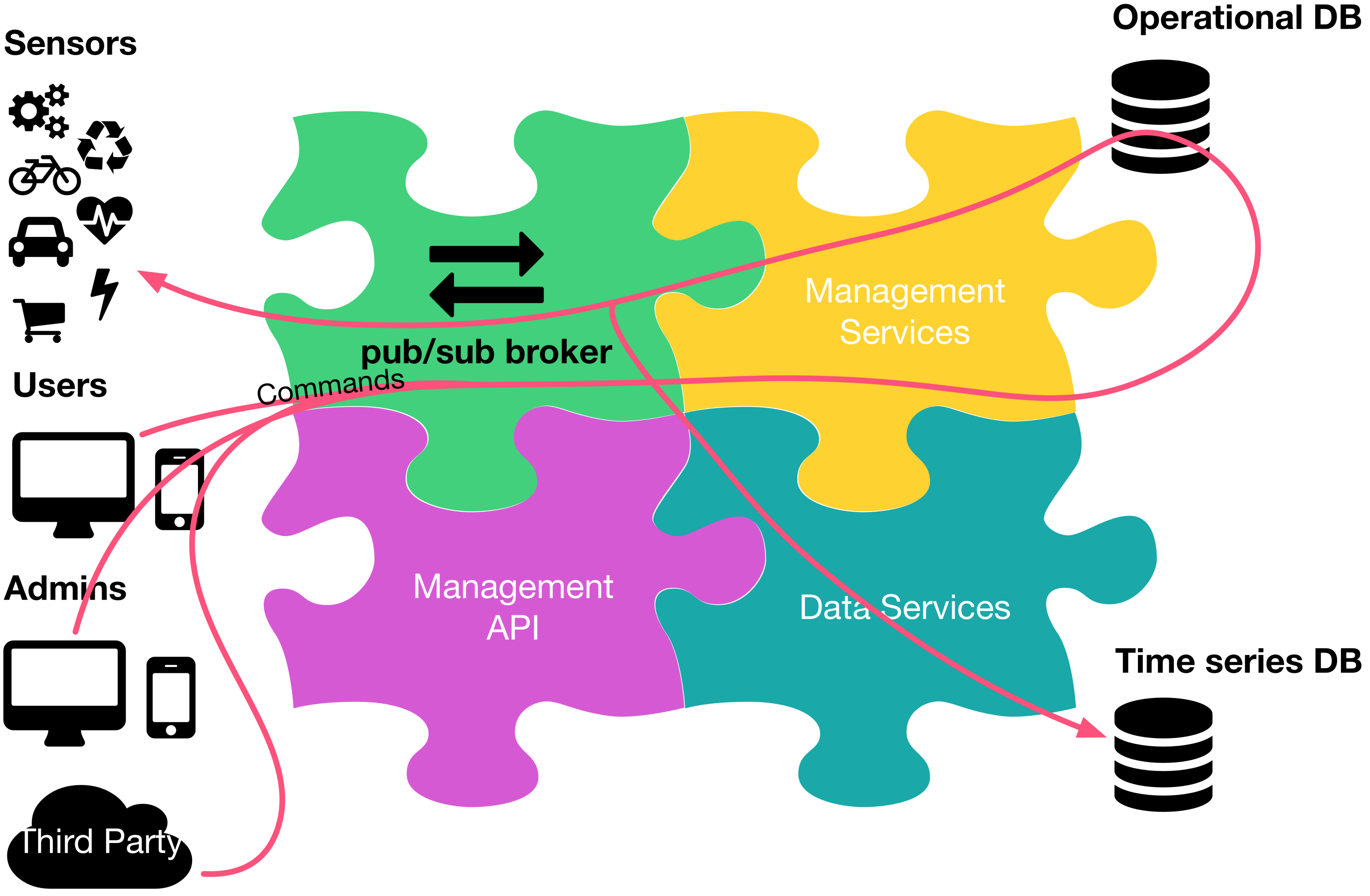
lots of moving parts

Using a pub/sub broker

Sensor Data Flow



Commands Flow



MQTT.ORG

MQTT.js

MQTT.js works on

browsers

servers

devices

MQTT

publish/subscribe protocol

multiple quality of service level..

..with at-least-once and exactly-once semantics

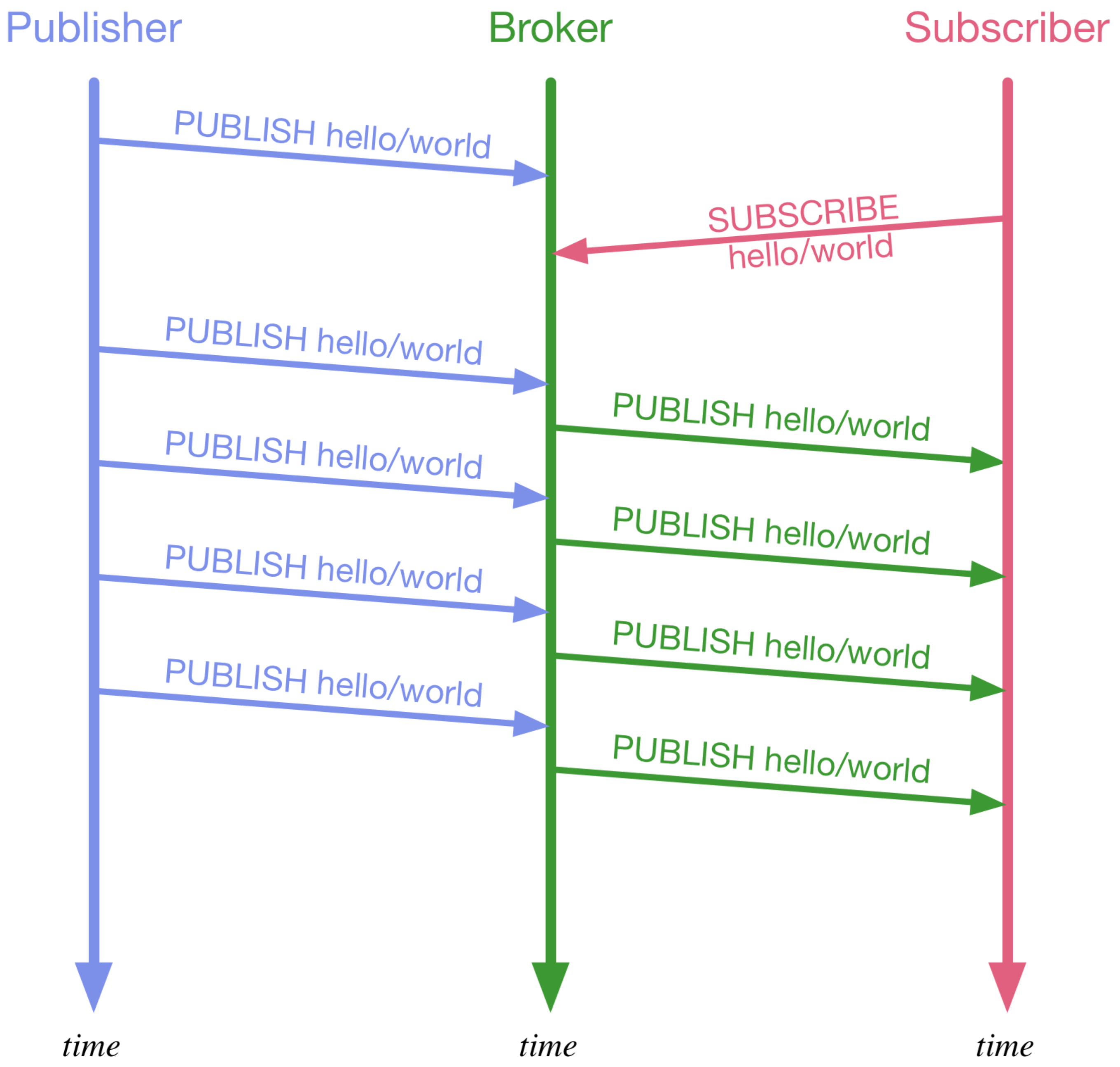
low overhead (2 bytes at minimum)

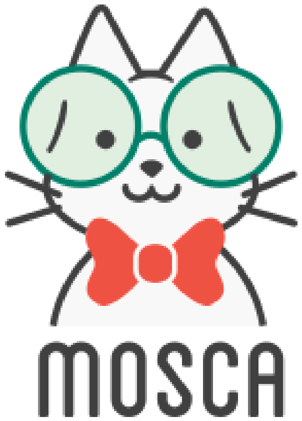
offline messaging

retained messages, like a key/value store

can run over WebSocket

Publish-Subscribe





MQTT broker in Node.js

Mosca

http://npm.im/mosca

Standalone usage, through `$ mosca`

Embeddable in your app, so you can do X for every message

Authentication APIs

Supports Mongo, Redis, AMQP, and MQTT as pub/sub backends

Needs a DB, such as LevelDB, Mongo, or Redis

Support websockets

Test instance available at test.mosca.io

New version coming with 10x throughput improvement

Authentication in Mosca

```
var server = new mosca.Server(settings);
/* Accepts the connection if the username
   and password are valid */
function authClient(client, user, pass, cb) {
  var authorized = (
    user=== 'alice' &&
    pass.toString() == 'secret' );
  if (authorized) client.user = user;
  callback(null, authorized);
}
server.authenticate = authClient;
```

Publish Authorization in Mosca

```
var server = new mosca.Server(settings);
function authPub(client, topic, payload, cb) {
  var ok = client.user == topic.split('/')[1];
  /* we can alter the message here */
  if (ok) callback(null, payload);
  else callback(null, false);
}
server.authorizePublish = authPub;
```

Subscribe Authorization in Mosca

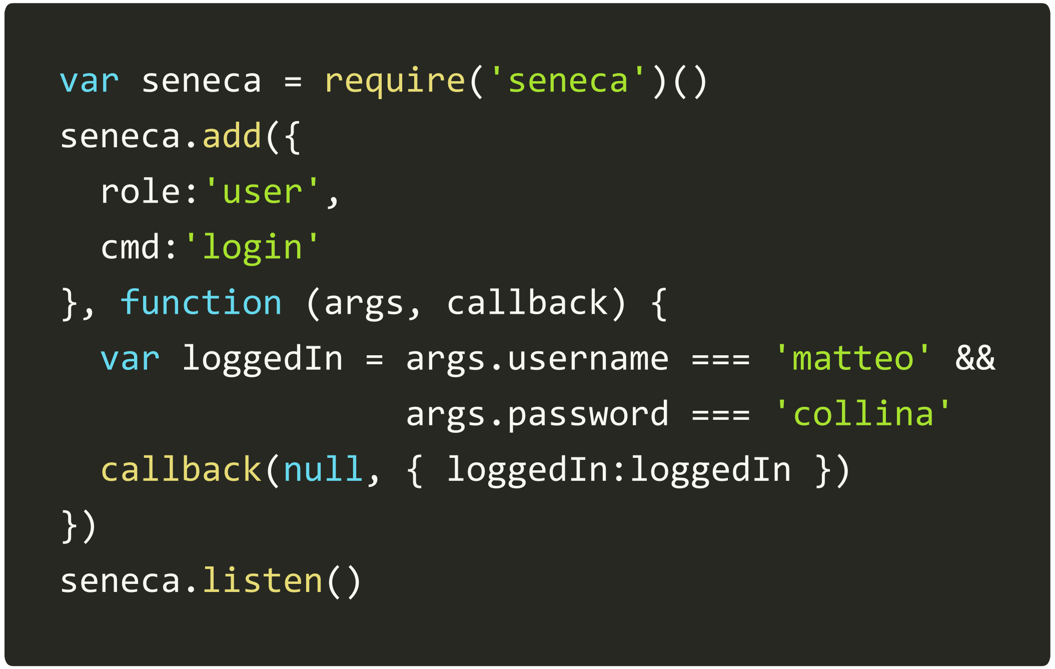
```
var server = new mosca.Server(settings);
function authSub(client, topic, cb) {
  var ok = client.user === topic.split('/')[1];
  cb(null, ok);
}
server.authorizeSubscribe = authSub;
```

Storing data in a timeseries

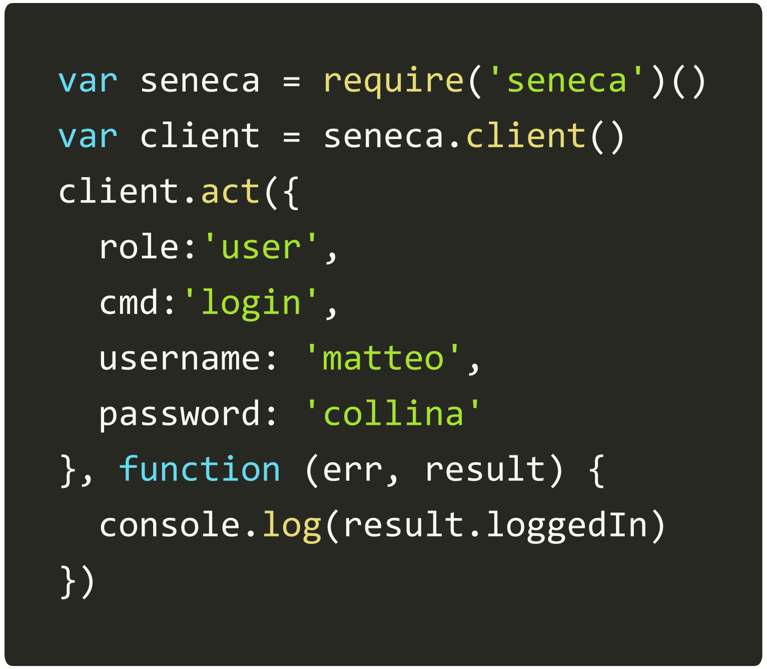
```javascript
var server = new mosca.Server(settings);

function published(packet, client, callback) {
  timeseries.store(packet, callback);
}
server.published = published;
```

Seneca

http://senecajs.org



```
var seneca = require('seneca')()
seneca.add({
  role:'user',
  cmd:'login'
}, function (args, callback) {
  var loggedIn = args.username === 'matteo' &&
                 args.password === 'collina'
  callback(null, { loggedIn:loggedIn })
})
seneca.listen()
```



```
var seneca = require('seneca')()
var client = seneca.client()
client.act({
  role:'user',
  cmd:'login',
  username: 'matteo',
  password: 'collina'
}, function (err, result) {
  console.log(result.loggedIn)
})
```

SENECA

Features

multiple transports, from bare TCP to busses

build a monolith, and then split it away

deep inspection & debugging tools

authentication/user managament system

control panel

integration with HAPI

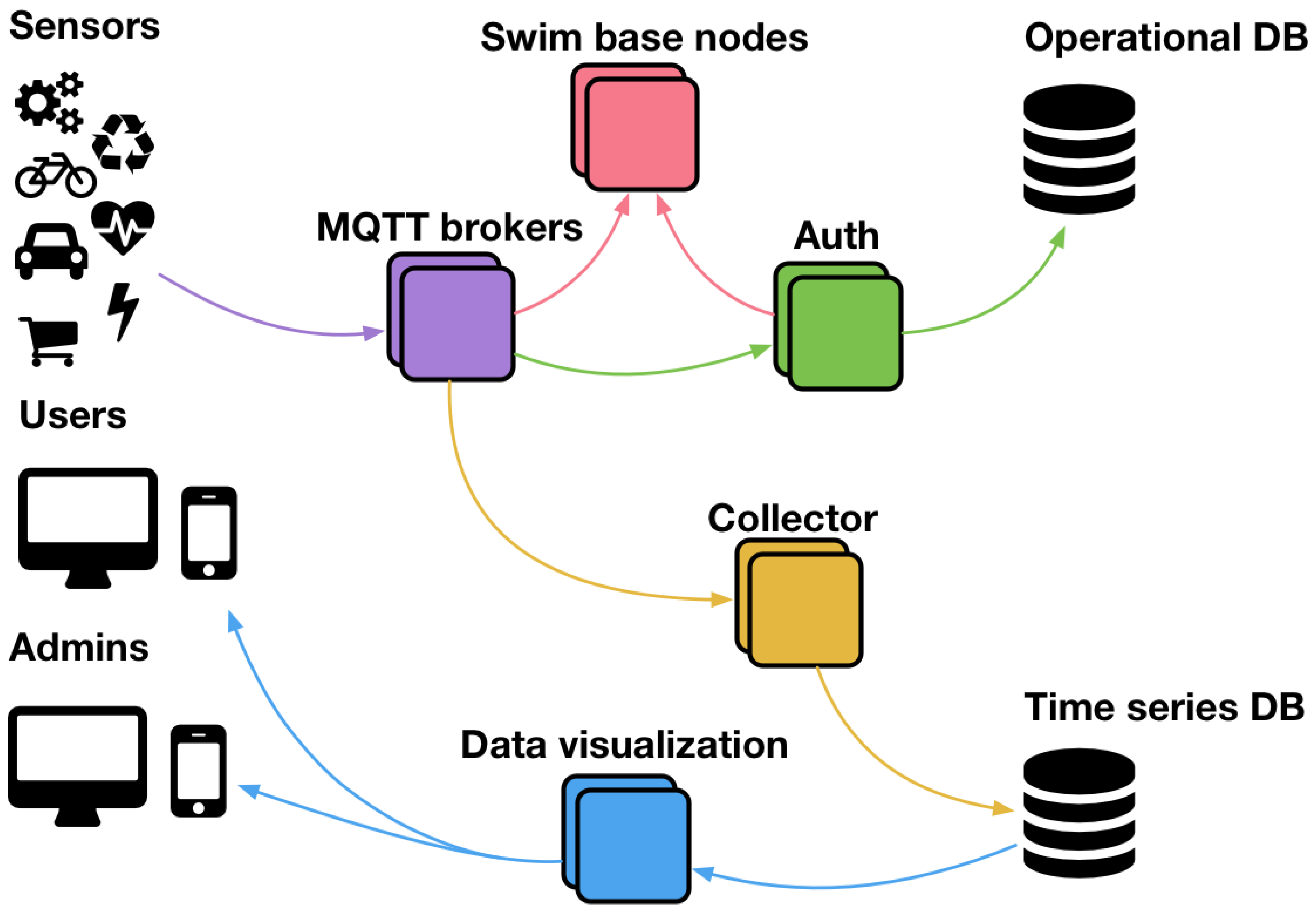
gossip-based service discovery

Demo: an IoT system

microservice based

on top of MQTT

service discovery based on gossip


Sensors
Swim base nodes
Operational DB
MQTT brokers
Auth
Users
Collector
Admins
Time series DB
Data visualization

fuge

Demo!

https://github.com/nearform/iot-system

Links

MQTT.js: http://npm.im/mqtt

Mosca: http://mosca.io

Seneca: http://senecajs.org/

Fuge: http://fuge.io

Vidi: https://github.com/vidi-insights/vidi-dashboard

seneca-mesh: http://npm.im/seneca-mesh

toolbag: http://npm.im/toolbag

swim-js: http://npm.im/swim-js

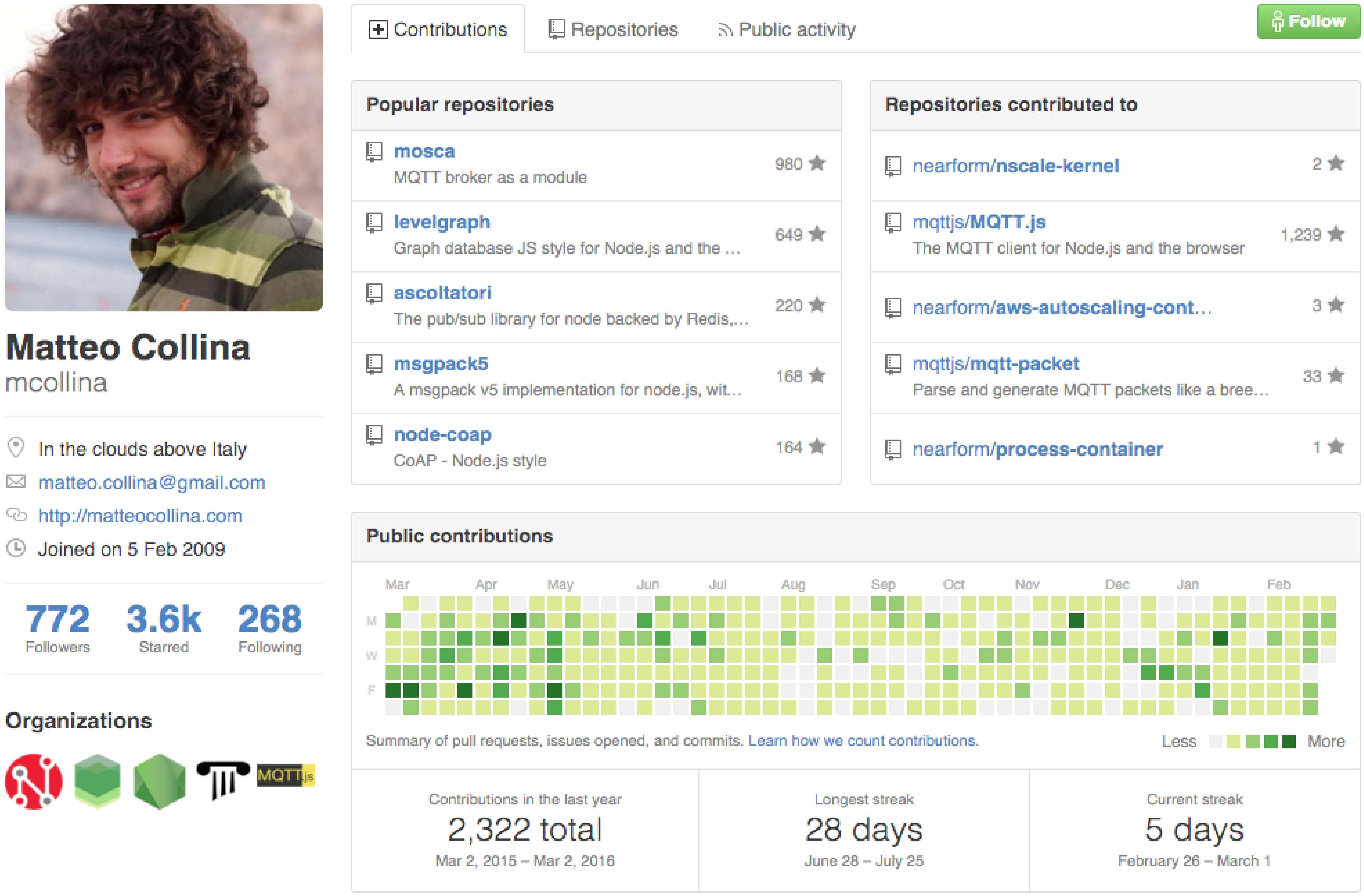
baseswim: http://npm.im/baseswim

This presentation

https://github.com/mcollina/js-everywhere

https://mcollina.github.io/js-everywhere

https://github.com/nearform/iot-system



Matteo Collina
mcollina

In the clouds above Italy
matteo.collina@gmail.com
http://matteocollina.com
Joined on 5 Feb 2009

772 Followers
3.6k Starred
268 Following

Organizations

Popular repositories

mosca MQTT broker as a module	980 ★
levelgraph Graph database JS style for Node.js and the …	649 ★
ascoltatori The pub/sub library for node backed by Redis,…	220 ★
msgpack5 A msgpack v5 implementation for node.js, wit…	168 ★
node-coap CoAP - Node.js style	164 ★

Repositories contributed to

nearform/**nscale-kernel**	2 ★
mqttjs/**MQTT.js** The MQTT client for Node.js and the browser	1,239 ★
nearform/**aws-autoscaling-cont…**	3 ★
mqttjs/**mqtt-packet** Parse and generate MQTT packets like a bree…	33 ★
nearform/**process-container**	1 ★

Public contributions

Summary of pull requests, issues opened, and commits. Learn how we count contributions.

Contributions in the last year	Longest streak	Current streak
2,322 total	28 days	5 days
Mar 2, 2015 – Mar 2, 2016	June 28 – July 25	February 26 – March 1

nearForm

Thanks!

If you need help with Node

or the Internet of Things

matteo.collina@nearform.com

@matteocollina on Twitter

www.nearform.com