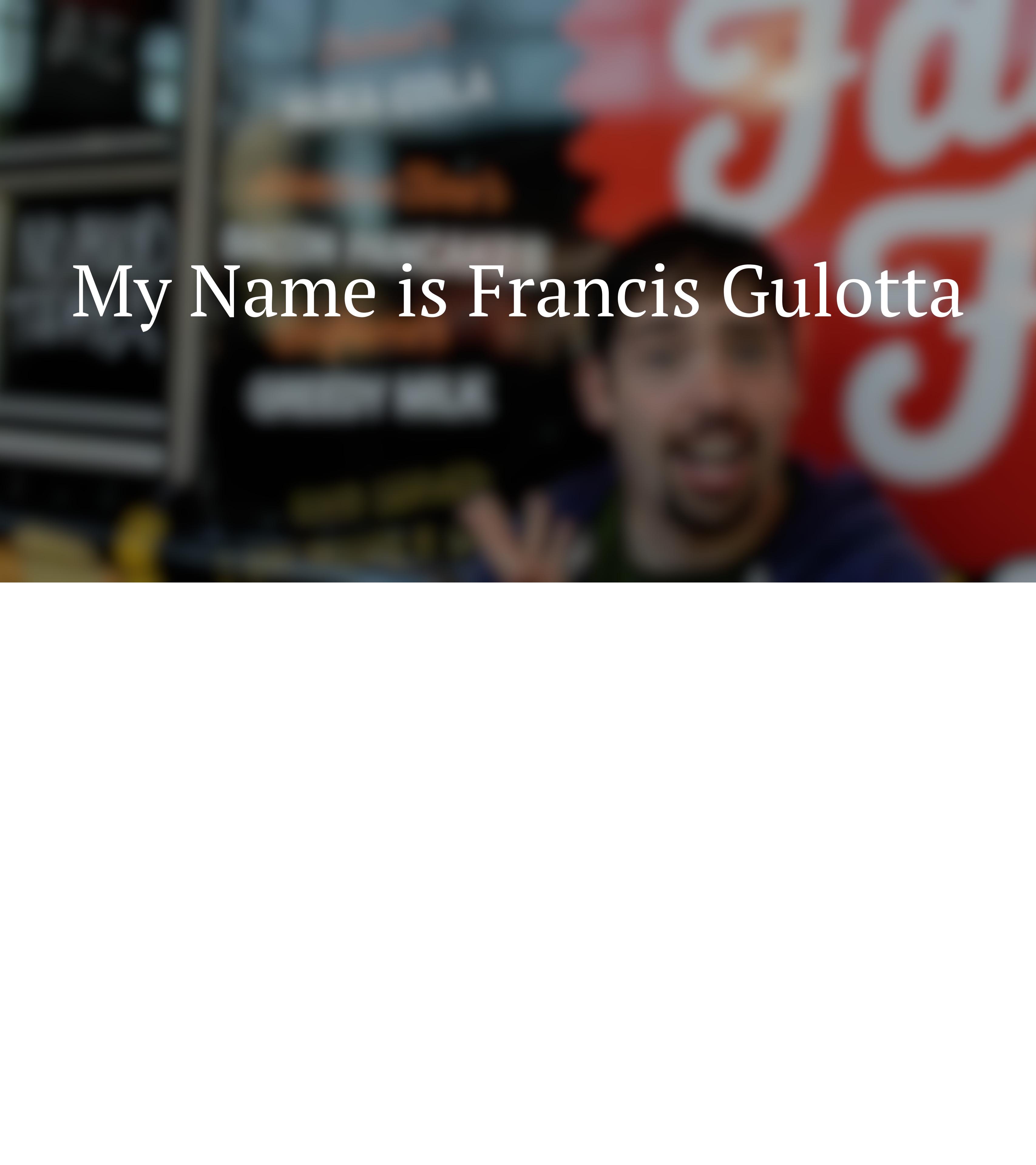


# Providing a good Developer Experience for JavaScript Hardware

---





My Name is Francis Gulotta





@reconbot

I work at Bocoup



# **Open Design & Technology Services**

## **for People Creating the Future**



# Open Design & Technology Services

by people who helped build and improve the open source libraries and tools you're already using like v8, grunt, ember, jquery, d3, johnny-five, jshint, wordpress and so much more.

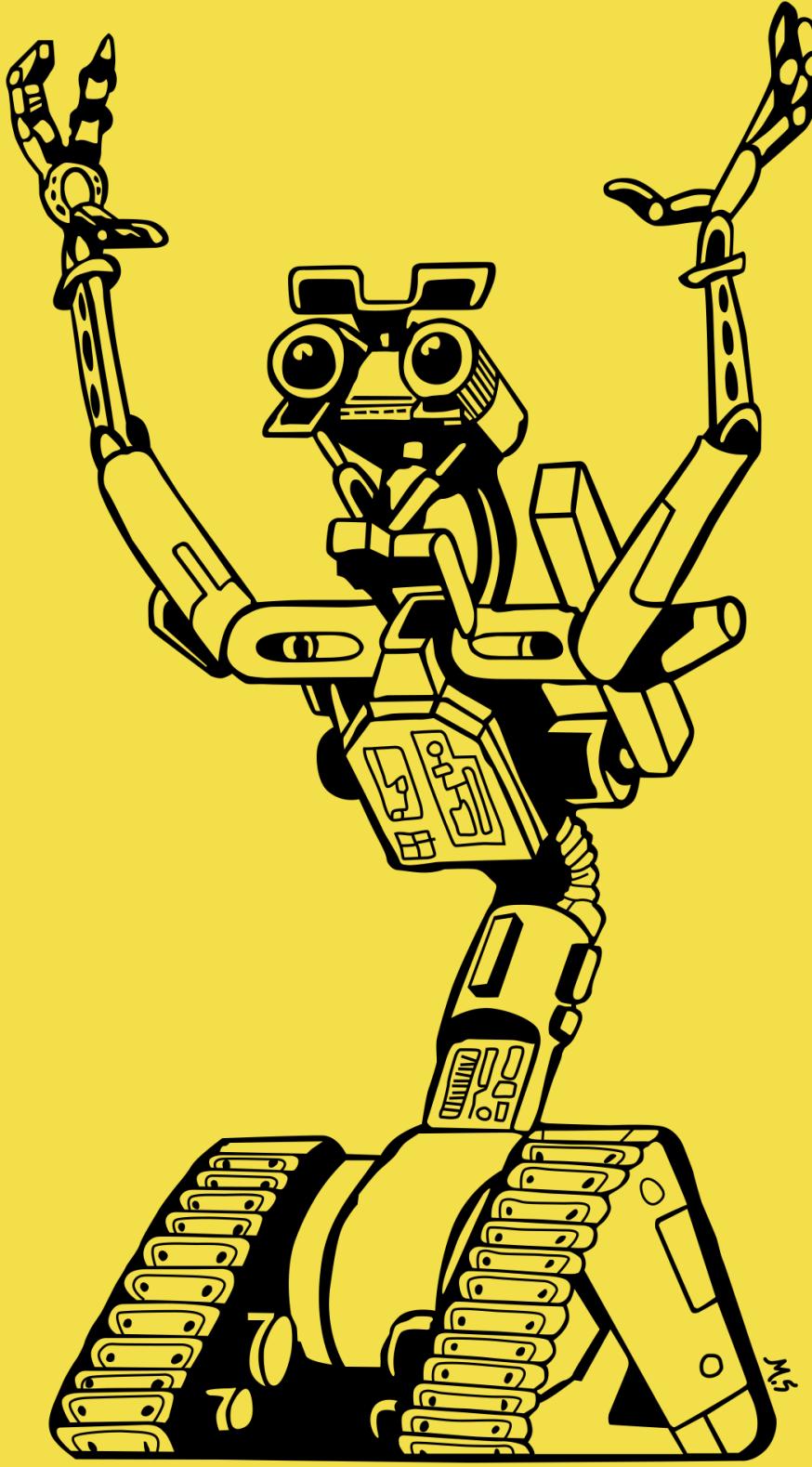


I work on

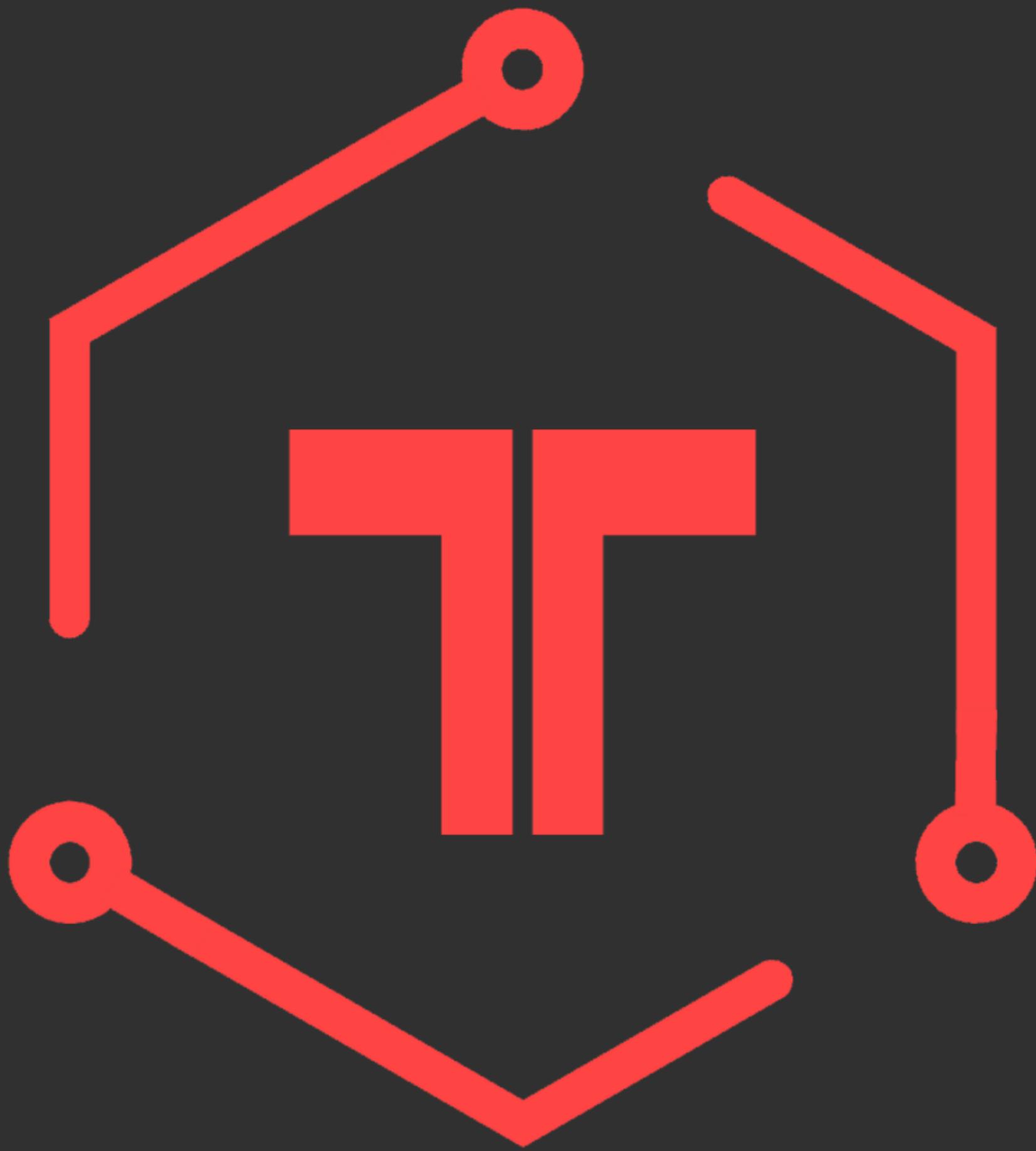
- Node SerialPort (and Web SerialPort W3C)
- Johnny-Five
- Tessel 2

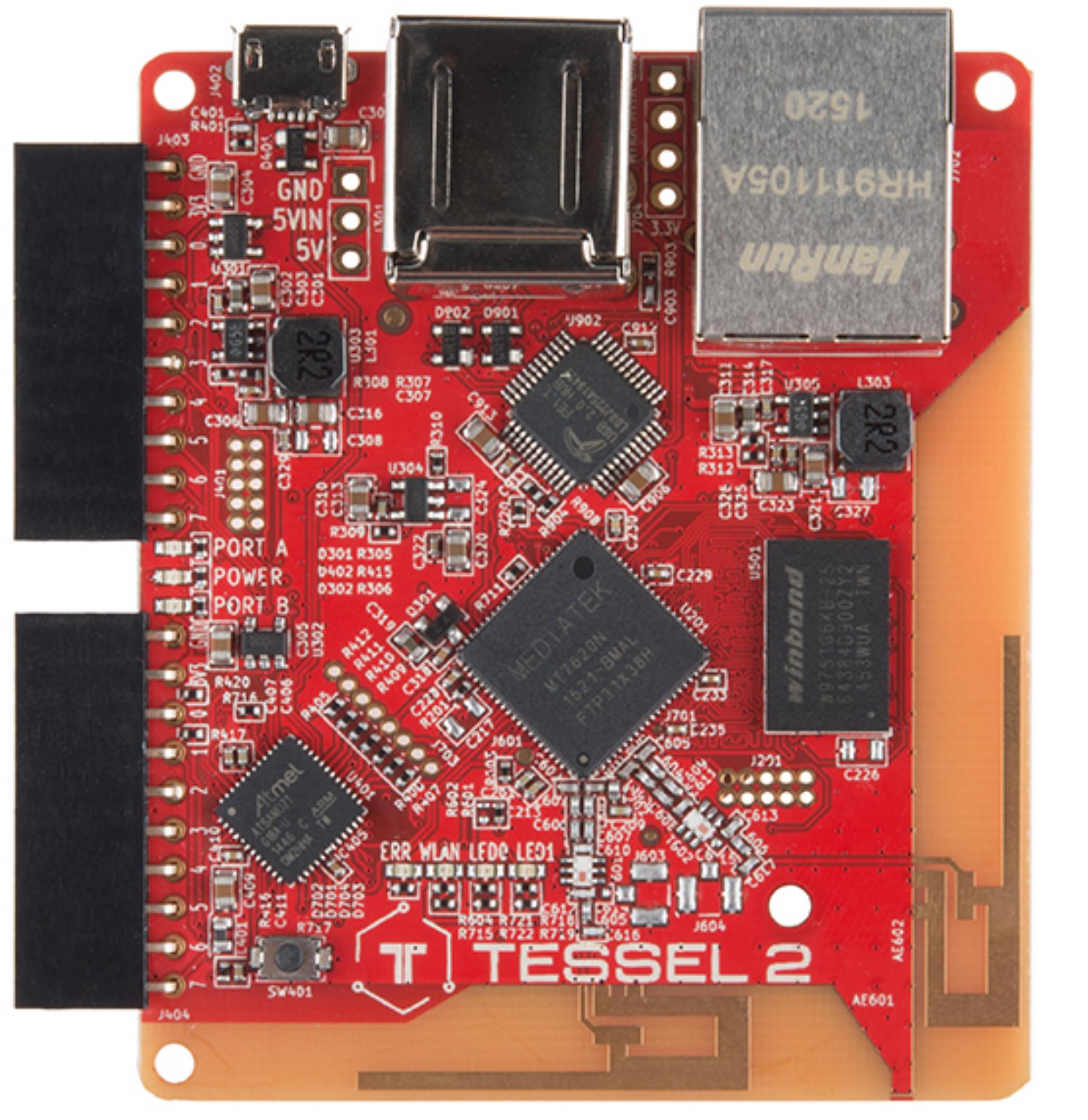
node sp provides serialport access to over 25 platforms across versions of node, cross platform and on all sorts of cpus, in the top 1-2% of all npm packages

# J5



Johnny Five provides a javascript hardware abstraction layer and device drivers. Meaning you can write an app once and use it anywhere.





Tessel provides the best user experience for deploying, managing and running your JS code on a single board computer. Soon Rust!

Sorry that's a lot about me, but you should know who I am because I'm going to ask you to believe me

We need a good developer  
experience for JavaScript  
Hardware.

---

because

# C++ is horrible

---

```
void EIO_Open(uv_work_t* req) {
    OpenBaton* data = static_cast<OpenBaton*>(req->data);

    char originalPath[1024];
    strncpy_s(originalPath, sizeof(originalPath), data->path, _TRUNCATE);
    // data->path is char[1024] but on Windows it has the form "COMx\0" or "COMxx\0"
    // We want to prepend "\\\\" to it before we call CreateFile
    strncpy(data->path + 20, data->path, 10);
    strncpy(data->path, "\\\\", 4);
    strncpy(data->path + 4, data->path + 20, 10);

    int shareMode = FILE_SHARE_READ | FILE_SHARE_WRITE;
    if (data->lock) {
        shareMode = 0;
    }

    HANDLE file = CreateFile(
        data->path,
        GENERIC_READ | GENERIC_WRITE,
        shareMode,
        // dwShareMode 0 Prevents other processes from opening if they request
        // delete, read, or write access
        NULL,
        OPEN_EXISTING,
        FILE_FLAG_OVERLAPPED,
        // allows for reading and writing at the same time and sets the handle for asynchronous I/O
        NULL
    );
}
```

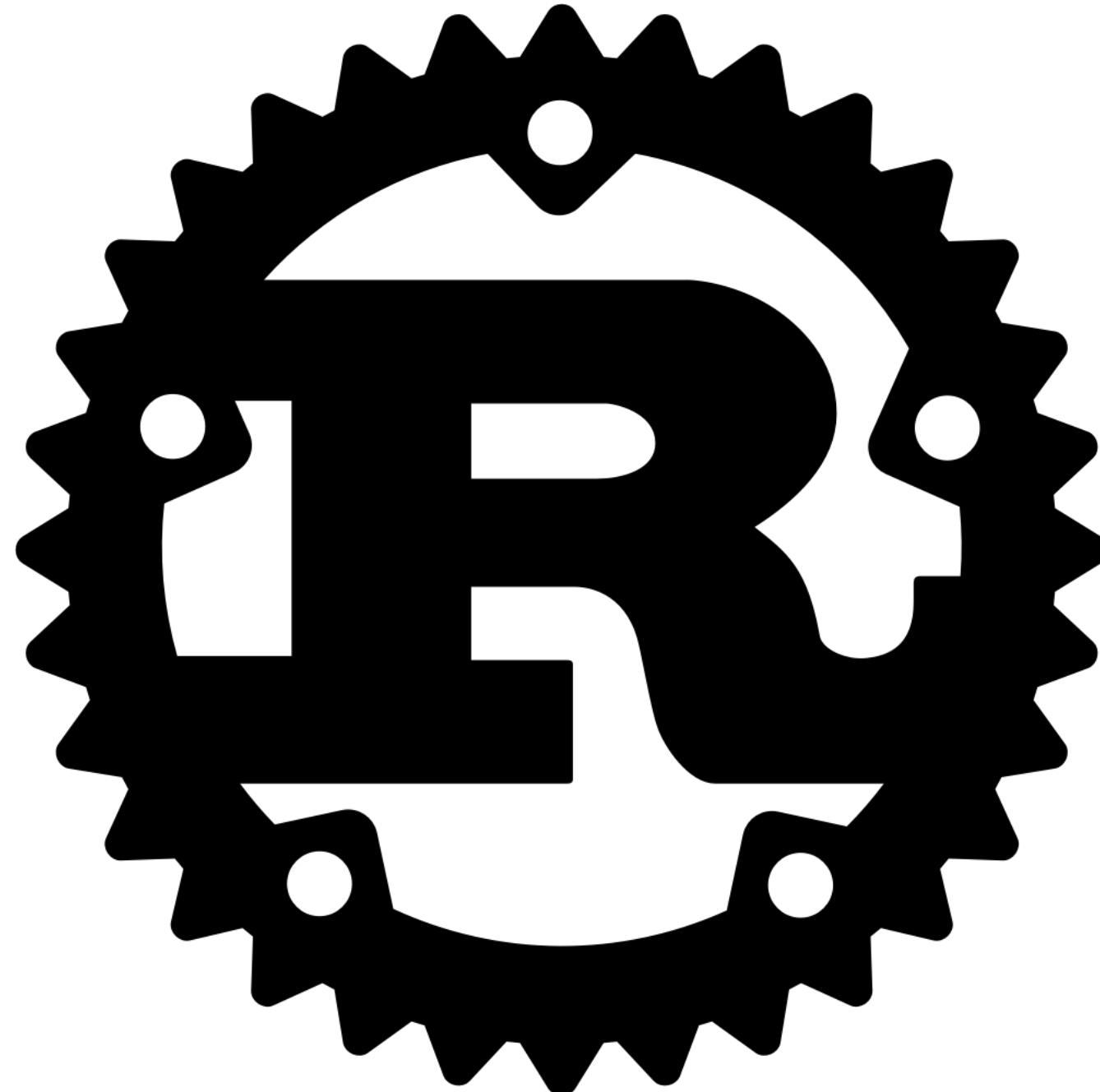
look at this horrible c++ I wrote

## C++

- hard to learn
- hard to write
- hard to not make mistakes
- really hard to work online

**remember heartbleed?**

Rust



I'm enjoying learning rust more than  
the 10 years of c++ I've failed  
c is 44 years old  
C++ is 33 years old

---

"C++ is a horrible language."

— Linus Torvald Out of Context

---

He actually likes C but that's horrible too

We need a good developer  
experience for JavaScript  
Hardware.

---

I didn't answer your question, lets try again

# Firmware is C++

---

actually it's assembly



and it's everywhere

Honeywell







In anything with wifi



in your car



We need a good developer  
experience for JavaScript  
Hardware.

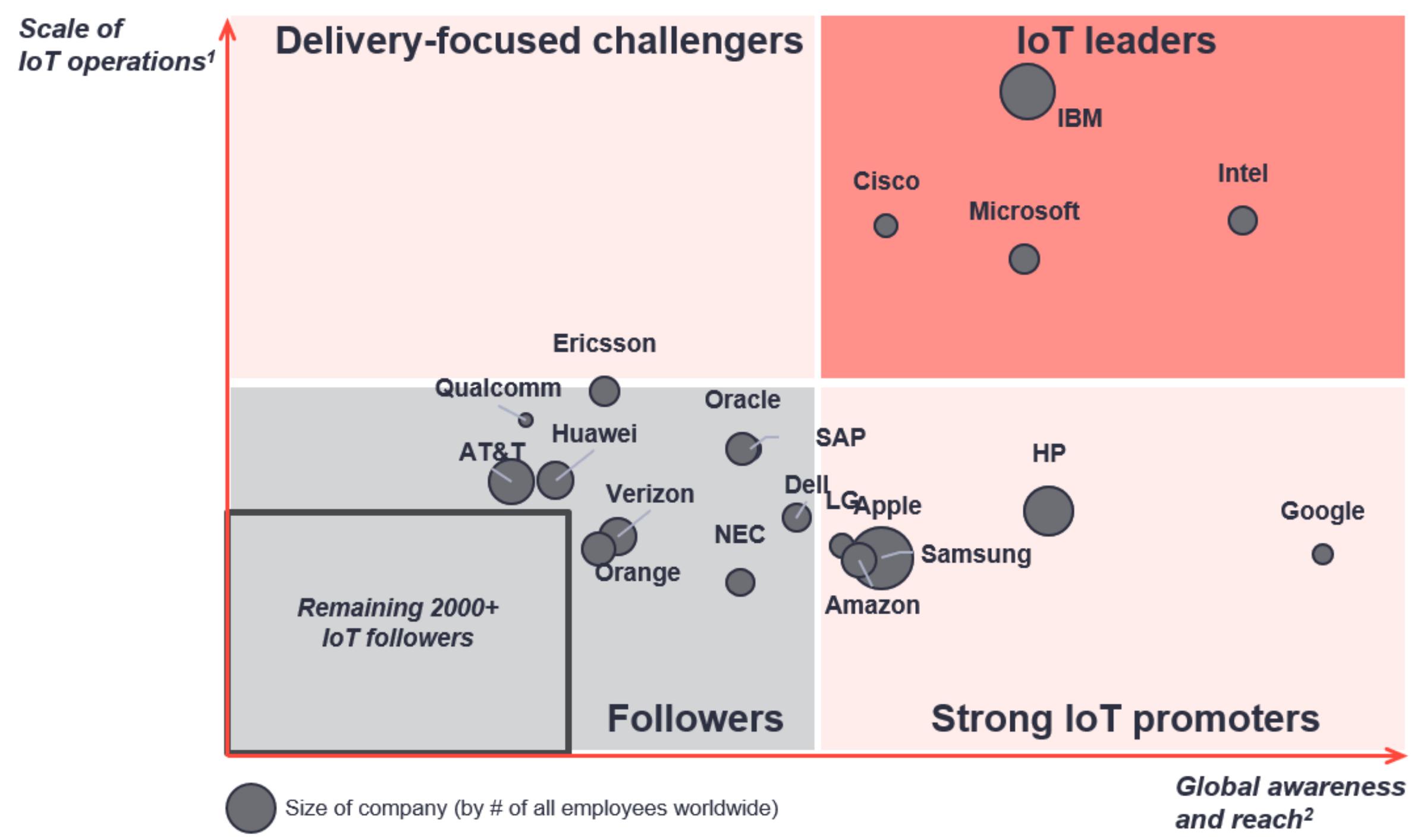
---

Still don't know why

Every device company  
wants to be an IoT  
company.

---

## Matrix of leading IoT companies



1. Scale of IoT operations measured by estimated number of employees performing IoT-related operations 2. Global awareness and reach measured by number of news appearances and related search engine inquiries. Source: Google, LinkedIn, IoT Analytics

# for one reason

# A device is one sale

---

# A web service is infinity sales

---

# Devices + Web = IoT

---

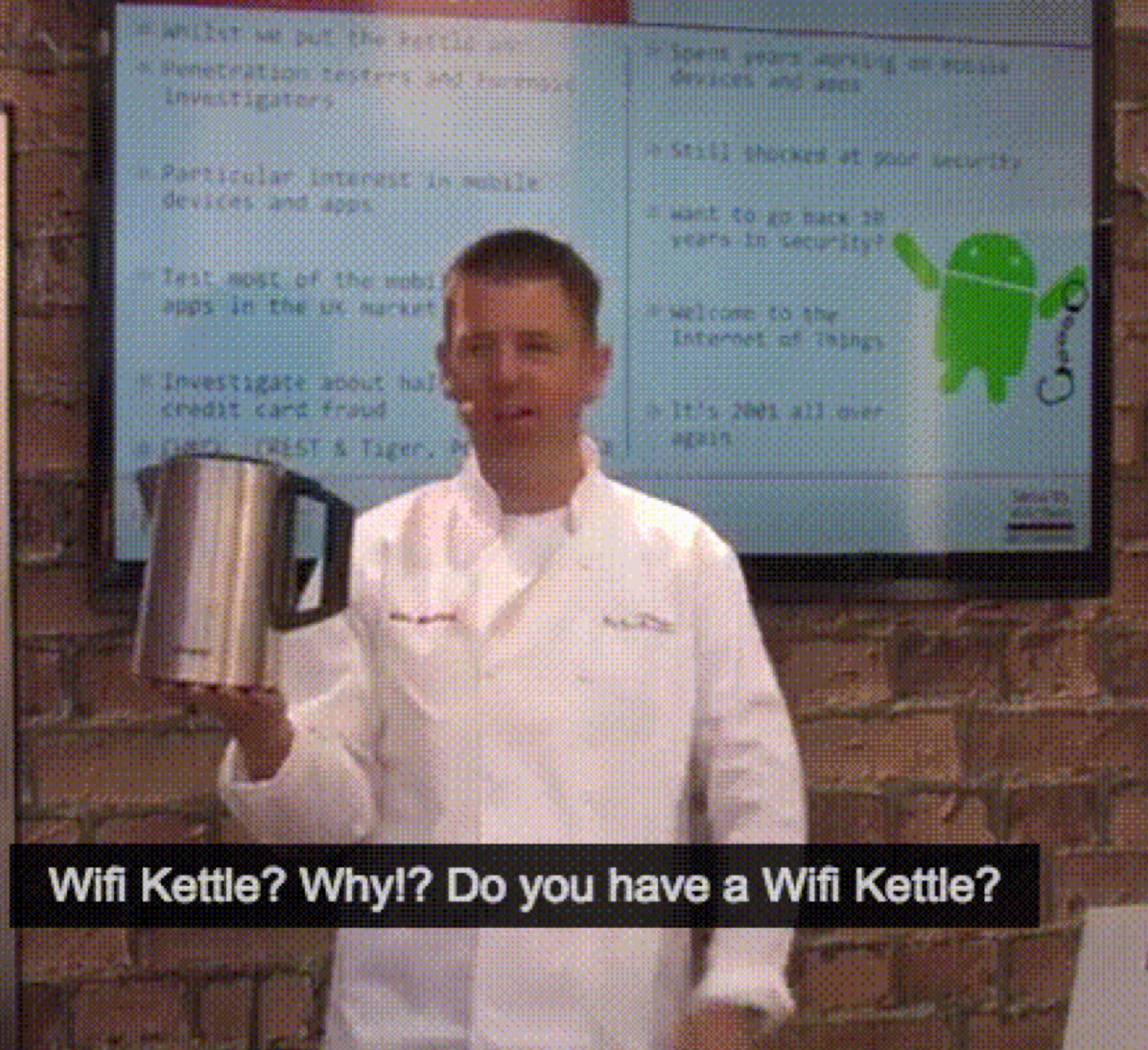
Every device manufacturer wants to be "web"

# C++ + WEB = IoT

---

# C++ is horrible

Yeah but I'll need some evidence



Researchers at an infosec conference discover simple already solved bugs in IoT devices

---

"It's like going back to the year  
2000 ... Everything is the wild  
west of security"

— "Pen Test Partners" @ Infosecurity Europe 2015

---

---

"Manufactures of these *things*  
don't get security."

— "Pen Test Partners" @ Infosecurity Europe 2015

---

<https://www.youtube.com/watch?v=GDy9Nvcw4O4>

---

They're shoehorning "WEB" into  
existing device firmware

— Francis

---



Tal Klein

@VirtualTal



Follow

My Nest thermostat has been locked by ransomware.. It's demanding \$300 in 24 hours or it'll lock the temp at 99.

#complaintsfromthefuture

RETWEETS

95

LIKES

80



2:16 AM - 17 Jan 2014



95



80

...

<https://twitter.com/VirtualTal/status/424077709598146560>



Defcon 2016 <http://motherboard.vice.com/read/internet-of-things-ransomware-smart-thermostat>

also the nest has been hacked! Takes 15 seconds <http://www.forbes.com/sites/aarontilley/2015/03/06/nest-thermostat-hack-home-network/#5a23729e5cb0>  
<http://www.computerworld.com/article/2476599/cybercrime-hacking/black-hat-nest-thermostat-turned-into-a-smart-spy-in-15-seconds.html>

## SHARE



SHARE



TWEET



PIN



COMMENT



EMAIL

ANDY GREENBERG SECURITY 05.02.16 7:00 AM

# FLAWS IN SAMSUNG'S 'SMART' HOME LET HACKERS UNLOCK DOORS AND SET OFF FIRE ALARMS



GET WIRED

Don't Let The Future Leave You Behind. Get 6 Issues For Just \$5.

SUBSCRIBE NOW

## LATEST NEWS



## MOVIES

*Manchester By The Sea*  
Trailer: Amazon's First Oscar Bait

10 MINS



## ELECTRIC VEHICLES

Tesla's Model S Now Does an Even Loonier 60 MPH in 2.5 Seconds  
22 HOURS

## GAMING

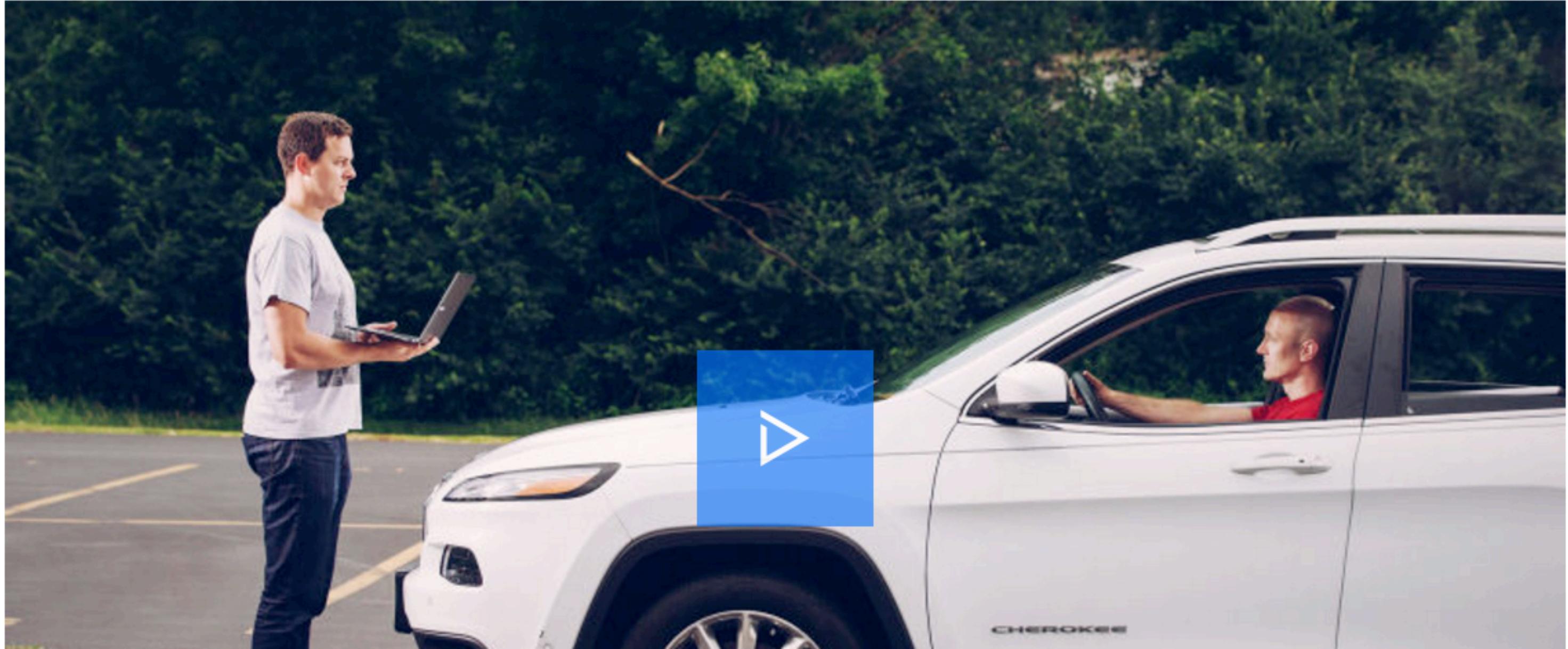
PlayStation Now, On-Demand Game Streaming, Is Coming to PCs  
23 HOURS

<https://www.wired.com/2016/05/flaws-samsungs-smart-home-let-hackers-unlock-doors-set-off-fire-alarms/>

Wired magazine keeps  
calling it like it is 

---

# HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY—WITH ME IN IT



public ip address on the Internet, with easy to exploit c++ services

^ <https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/>



Paul Ford

@ftrain



Following

i'm sitting here like an ass using immutable data structures and reading about TLA+ and people at Toyota are like `#define MAKE_CAR_GO 1`

RETWEETS

345

LIKES

521



1:23 PM - 30 Nov 2015

📍 Brooklyn, NY



345



521

...

<https://twitter.com/ftrain/status/671394222218592256>

BUSINESS

CULTURE

DESIGN

GEAR

SCIENCE

SECURITY

TRANSPORTATION

## SHARE



SHARE



TWEET



PIN



COMMENT



EMAIL

KIM ZETTER SECURITY 05.26.15 11:47 AM

# IS IT POSSIBLE FOR PASSENGERS TO HACK COMMERCIAL AIRCRAFT?



GET WIRED

**Don't Let The Future Leave You Behind. Get 6 Issues For Just \$5.**

SUBSCRIBE NOW

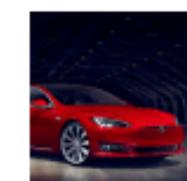
## LATEST NEWS



## MOVIES

*Manchester By The Sea* Trailer: Amazon's First Oscar Bait

44 MINS



## ELECTRIC VEHICLES

Tesla's Model S Now Does an Even Loonier 60 MPH in 2.5 Seconds

23 HOURS



## GAMING

PlayStation Now, On-

## SHARE



SHARE



TWEET



PIN



COMMENT



EMAIL

KIM ZETTER SECURITY 05.15.15 10:14 PM

# FEDS SAY THAT BANNED RESEARCHER COMMANDEERED A PLANE



GET WIRED

**Don't Let The Future Leave You Behind. Get 6 Issues For Just \$5.**

SUBSCRIBE NOW

## LATEST NEWS



## MOVIES

*Manchester By The Sea*  
Trailer: Amazon's First Oscar Bait

36 MINS



## ELECTRIC VEHICLES

Tesla's Model S Now Does an Even Loonier 60 MPH in 2.5 Seconds

23 HOURS



## GAMING

PlayStation Now, On-Demand Game Streaming, Is Coming to PCs

23 HOURS

# WHY?

---

# C++

---

# C++ and ...

---

A development  
environment that works  
great in isolation doesn't  
work great on the web.

---

"If copying a string is a dangerous operation in C++, then complaining on twitter using C++ is programmer negligence. ""

— Francis Gulotta

---

We need a good developer  
experience for JavaScript  
Hardware.

---

## A Short History of JavaScript

---

JavaScript, not to be confused with [Java](#), was created in 10 days in May 1995 by [Brendan Eich](#), then working at [Netscape](#) and now of [Mozilla](#). JavaScript was not always known as JavaScript: the original name was Mocha, a name chosen by [Marc Andreessen](#), founder of Netscape. In September of 1995 the name was changed to LiveScript, then in December of the same year, upon receiving a trademark license from Sun, the name JavaScript was adopted. This was somewhat of a marketing move at the time, with Java being very popular around then.

In 1996 - 1997 JavaScript was taken to [ECMA](#) to carve out a standard specification, which other browser vendors could then implement based on the work done at Netscape. The work done over this period of time eventually led to the official release of ECMA-262 Ed.1: ECMAScript is the name of the official standard, with JavaScript being the most well known of the implementations. ActionScript 3 is another well-known implementation of ECMAScript, with extensions (see below).

The standards process continued in cycles, with releases of ECMAScript 2 in 1998 and ECMAScript 3 in 1999, which is the baseline for modern day JavaScript. The "JS2" or "original ES4" work led by Waldemar Horwat (then of Netscape, now at Google) started in 2000 and at first, Microsoft seemed to participate and even implemented some of the proposals in their JScript.net language.

Over time it was clear though that Microsoft had no intention of cooperating or implementing proper JS in IE, even though they had no competing proposal and they had a partial (and diverged at this point) implementation on the .NET server side. So by 2003 the JS2/original-ES4 work was mothballed.

The next major event was in 2005, with two major happenings in JavaScript's history. First, Brendan Eich and Mozilla rejoined Ecma as a not-for-profit member and work started on E4X, ECMA-357, which came from ex-Microsoft employees at BEA (originally acquired as Crossgain). This led to working jointly with Macromedia, who were implementing E4X in ActionScript 3 (ActionScript 3 was a fork of Waldemar's JS2/original-ES4 work).

So, along with Macromedia (later acquired by Adobe), work restarted on ECMAScript 4 with the goal of standardizing what was in AS3 and implementing it in SpiderMonkey. To this end, Adobe released the "AVM2", code named Tamarin, as an open source project. But Tamarin and AS3 were too different from web JavaScript to converge, as was realized by the parties in 2007 and 2008.

Alas, there was still turmoil between the various players; Doug Crockford — then at Yahoo! — joined forces with Microsoft in 2007 to oppose ECMAScript 4, which led to the ECMAScript 3.1 effort.

While all of this was happening the open source and developer communities set to work to revolutionize what could be done with JavaScript. This community effort was sparked in 2005 when [Jesse James Garrett](#) released a white paper in which he coined the term Ajax, and described a set of technologies, of which JavaScript was the backbone, used to create web applications where data can be loaded in the background, avoiding the need for full page reloads and resulting in more dynamic applications. This resulted in a renaissance period of JavaScript usage spearheaded by open source libraries and the communities that formed around them, with libraries such as [Prototype](#), [jQuery](#), [Dojo](#) and [Mootools](#) and others being released.

In July of 2008 the disparate parties on either side came together in Oslo. This led to the eventual agreement in early 2009 to rename ECMAScript 3.1 to ECMAScript 5 and drive the language forward using an agenda that is known as [Harmony](#).

All of this then brings us to today, with JavaScript entering a completely new and exciting cycle of evolution, innovation and standardisation, with new developments such as the [Nodejs](#) platform, allowing us to use JavaScript on the server-side, and HTML5 APIs to control user media, open up web sockets for always-on communication, get data on geographical location and device features such as accelerometer, and more. It is an exciting time to learn

# JavaScript was born on the web

# JavaScript

- Easy to learn
- Easy to write
- Easy to work online
- Easy to share the work of others

# Easy to run on hardware

---

# Easy to control your hardware

---

# Intel gets it

---

<https://twitter.com/intel/status/734509277977530368>

<https://ostroproject.org/documentation/howtos/app-dev-nodejs.html>

## Intel

Worked with the V8 team to ensure NodeJS runs on the Intel Edison and Joule platforms

Worked with the Johnny-Five team to ensure device compatibility with all their new devices and distributes Johnny-Five in their getting started guides.

# Jibo gets it

```
        }
    } else {
        root.update();
    }
}, 30));
}

let requireBehaviors = (dir) => {
    let files = fs.readdirSync(dir);
    files.forEach((file) => {
        let parsed = path.parse(file);
        let absolutePath = path.resolve(dir, file);
        if(fs.lstatSync(absolutePath).isDirectory()) {
            requireBehaviors(absolutePath);
        }
        if(parsed.ext === '.js') {
            let behavior = path.resolve(dir, parsed.name);
            console.log(behavior);
            require(behavior);
        }
    });
};

jibo.init().then(() => {
    require('./behaviors/text-to-speech');
    require('./behaviors/succeed-on-listen');
    require('./behaviors/subtree-js');

    animate();
});

let eyeElement = document.getElementById('eye');
jibo.visualize.createRobotRenderer(eyeElement, jibo.visualize.DisplayType.EYE);
start();
}).catch((e) => {
    console.error(e);
});
```

**Jibo**

Integrates with the atom editor to let you produce  
animations and develop "skills" in JavaScript for use  
through the Jibo robot interface.

**Jibo.com**

# Silk Labs gets it

---

# SENSE

by Silk Labs



Take a smart phone, remove the screen, keep the camera  
and run nodejs and you get the sense

# Silk Labs

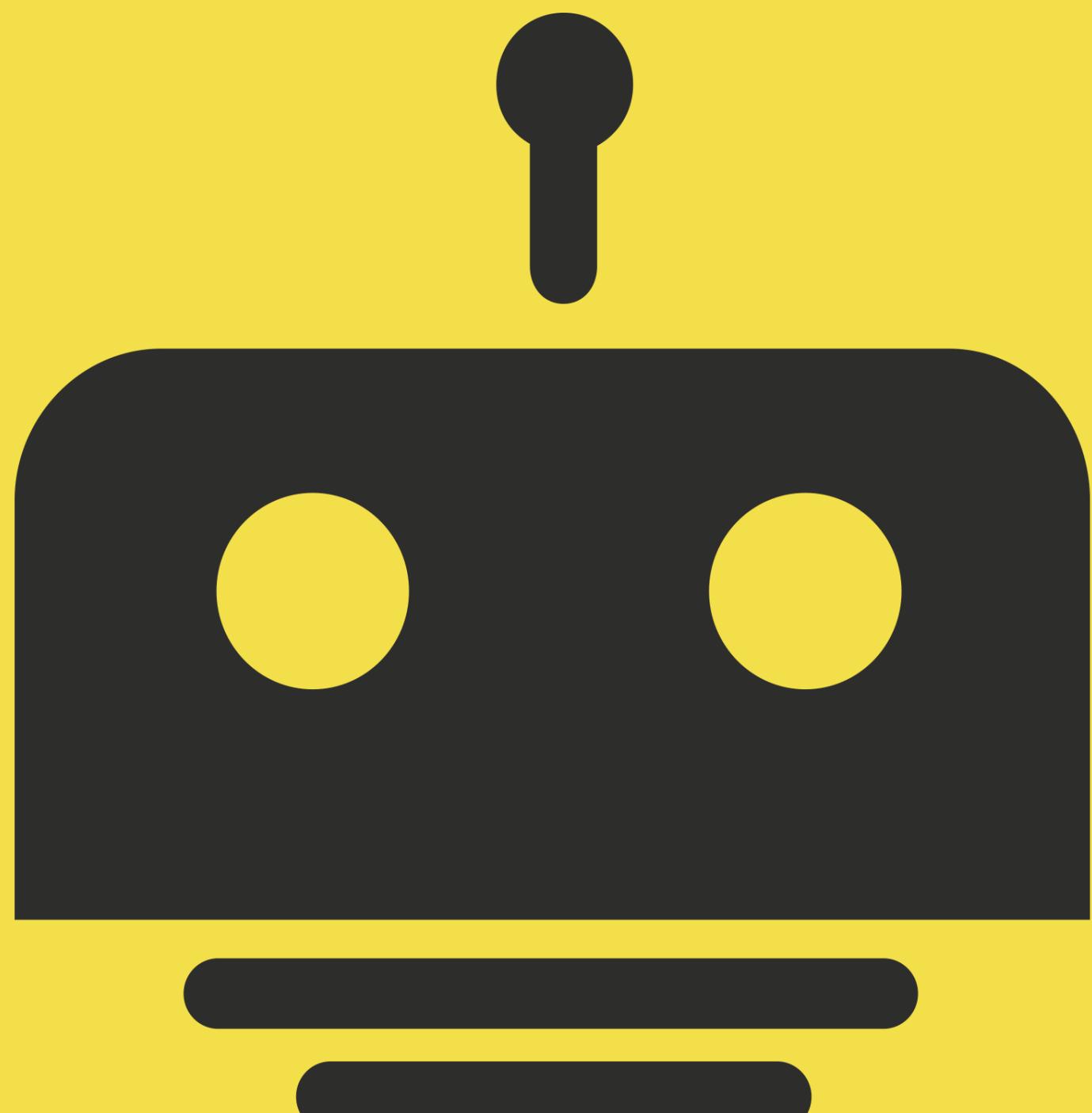
The "Sense camera and hub" provides JavaScript APIs for computer vision, advanced hearing, and deep learning libraries to mix with your standard NPM modules and run them to control your house from your house.

Not from the cloud.

<http://www.silklabs.com/>

# You can get it too

---



A world wide educational community around teaching  
JavaScript and electronics

# Johnny Five Inventors Kit (J5IK)

<https://www.sparkfun.com/products/13847>



# What's next?

---

# Most hacks are due to Poor Network and API access control

---

Most hacks are concerning  
pairing a device with a  
phone

---

# Help me start J5-Connect

---

Want to know which  
hardware has the best user  
experience?

## WCD Review

My team at Bocoup reviewed popular Single Board Computers

And rated them on ease of setup, quality of documentation, and flexibility of development workflow.

# Check out the Bocoup Web Connected Devices Review



<http://wcd-review.bocoup.com>

**Let me know what you think**

Francis@bocoup.com

@reconbot

**Thank you!**