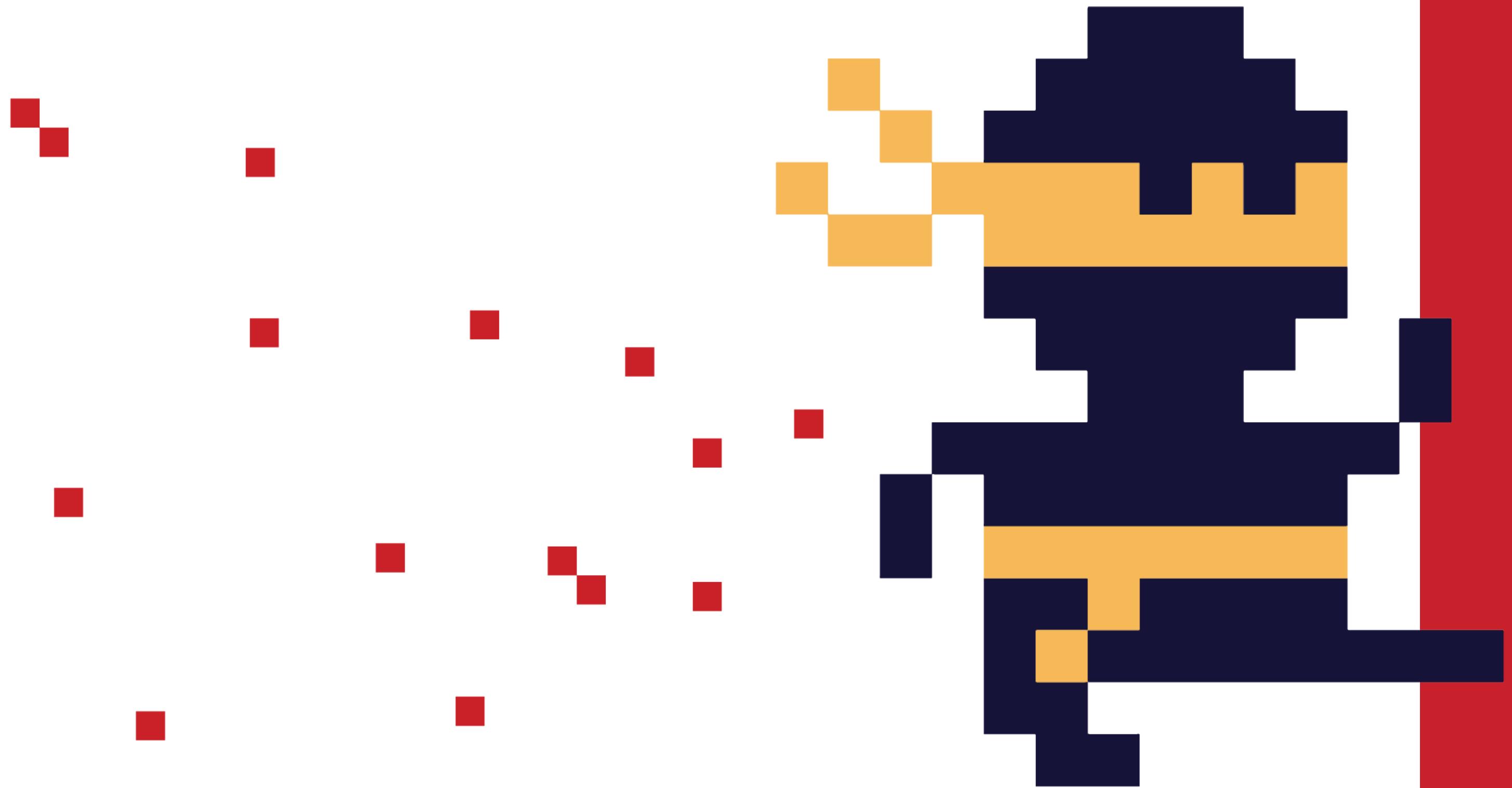


# ODESSAJS

July 20-21, Odessa



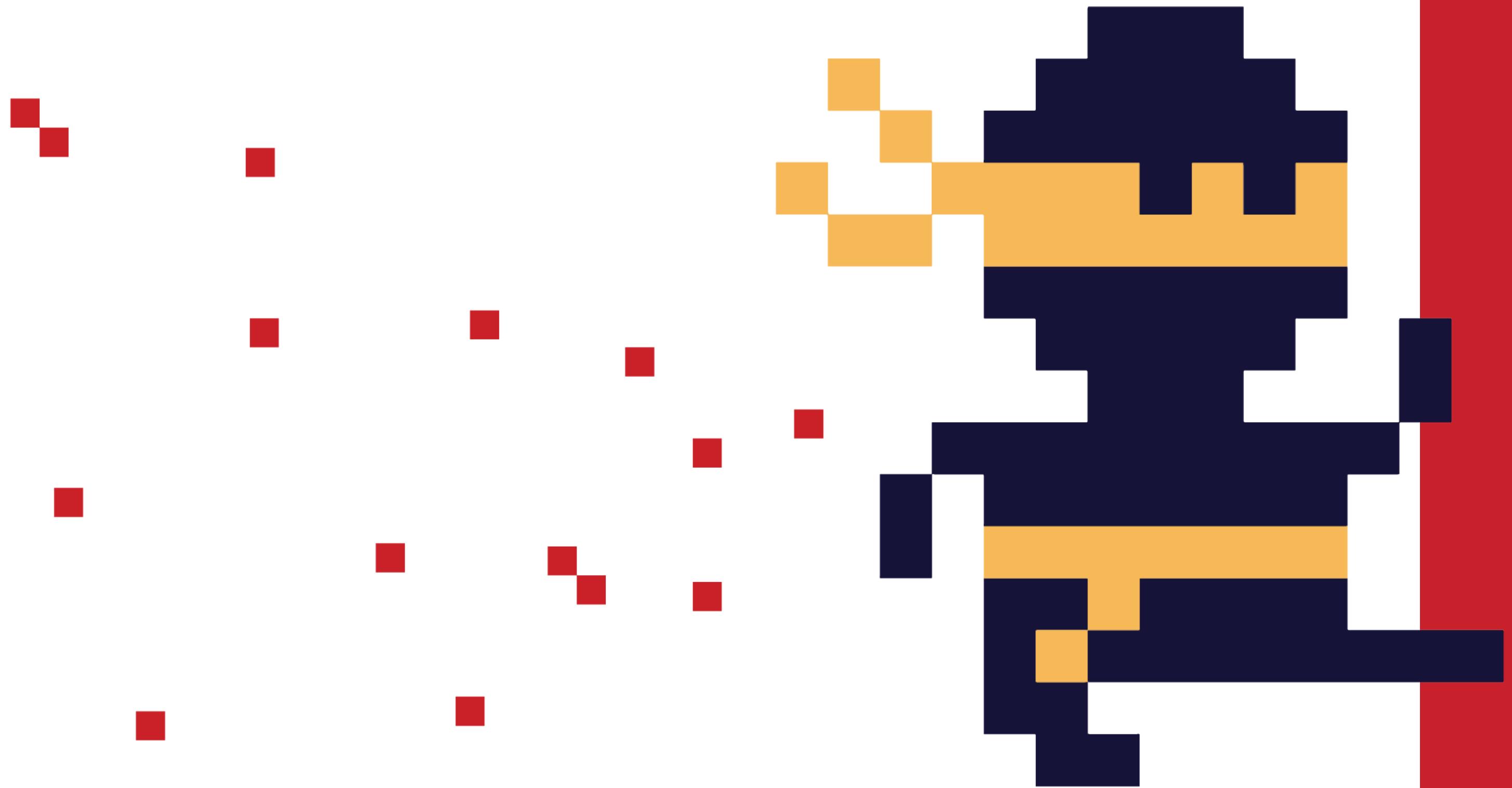
Summer. Sea. JavaScript.

Niels  
Leenheer

Fun with  
Bluetooth

# ODESSAJS

July 20-21, Odessa

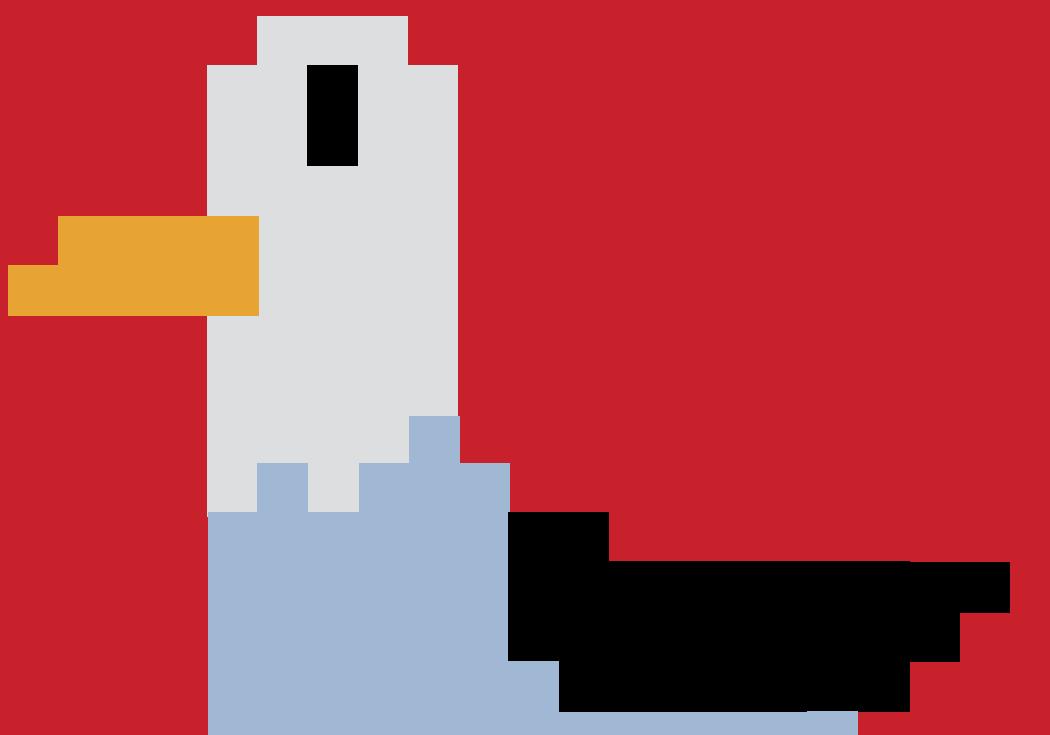


Summer. Sea. JavaScript.

Niels  
Leenheer  
@html5test

Fun with  
Bluetooth

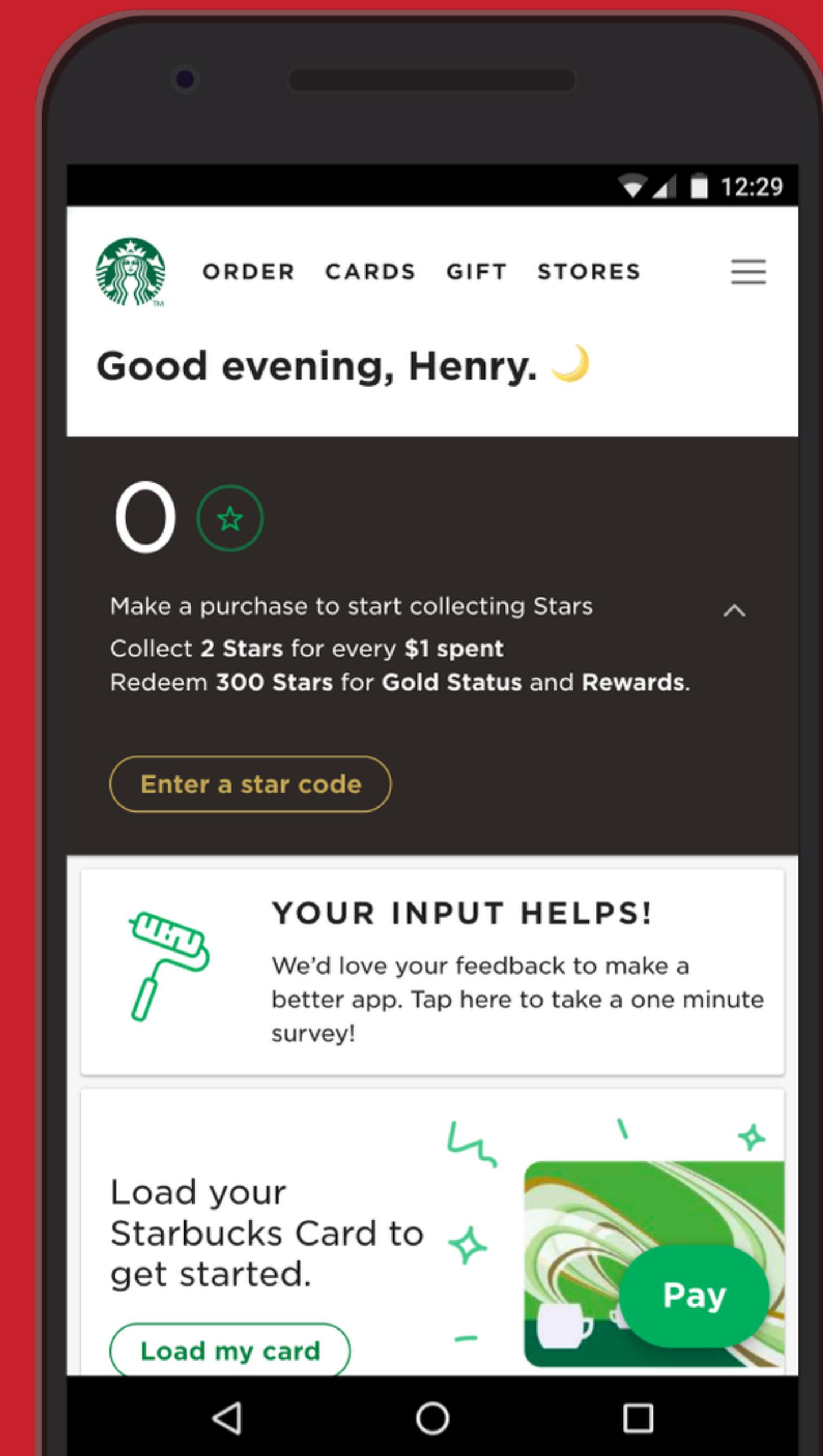
why?



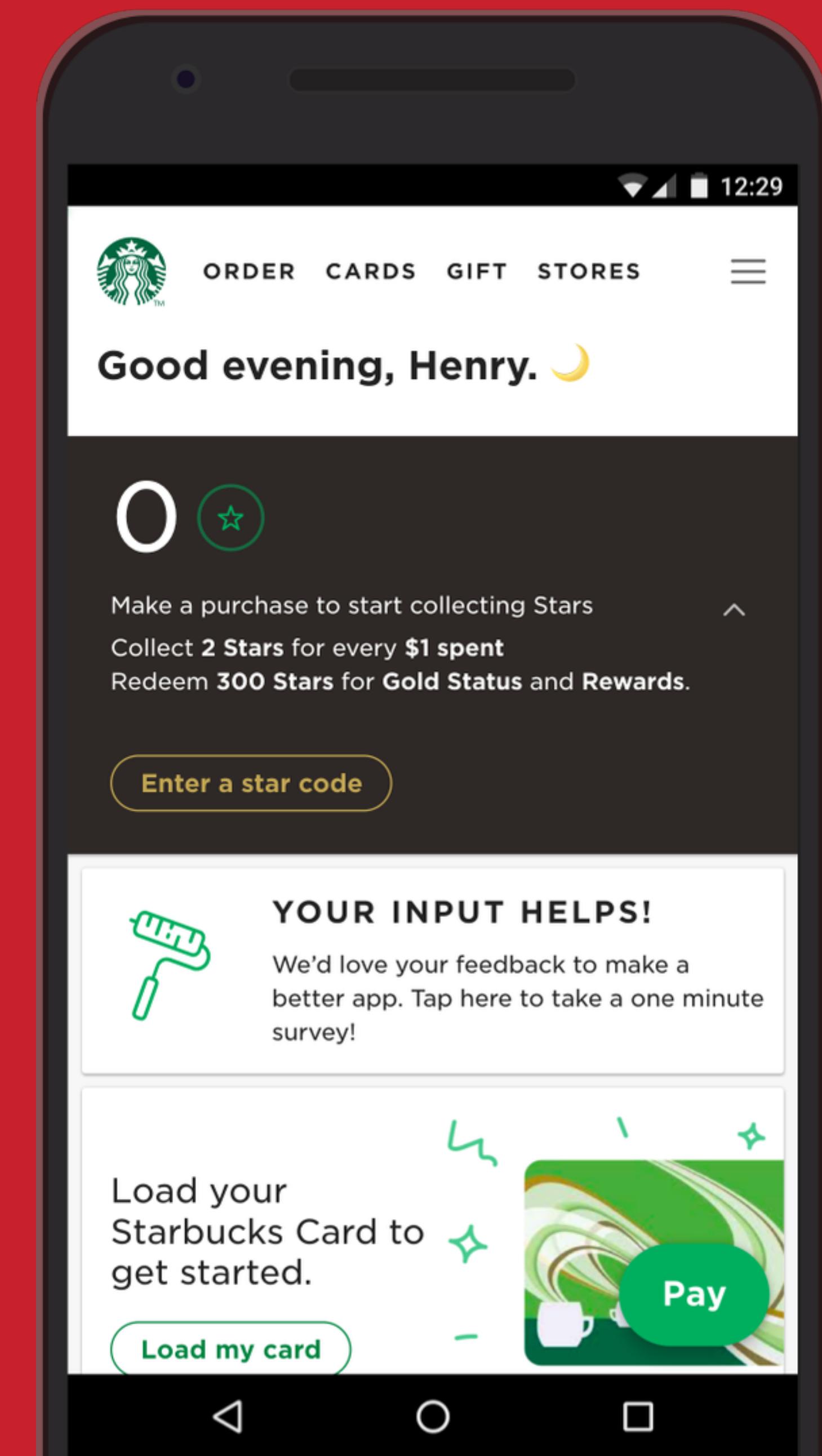
PWA

progressive  
web apps

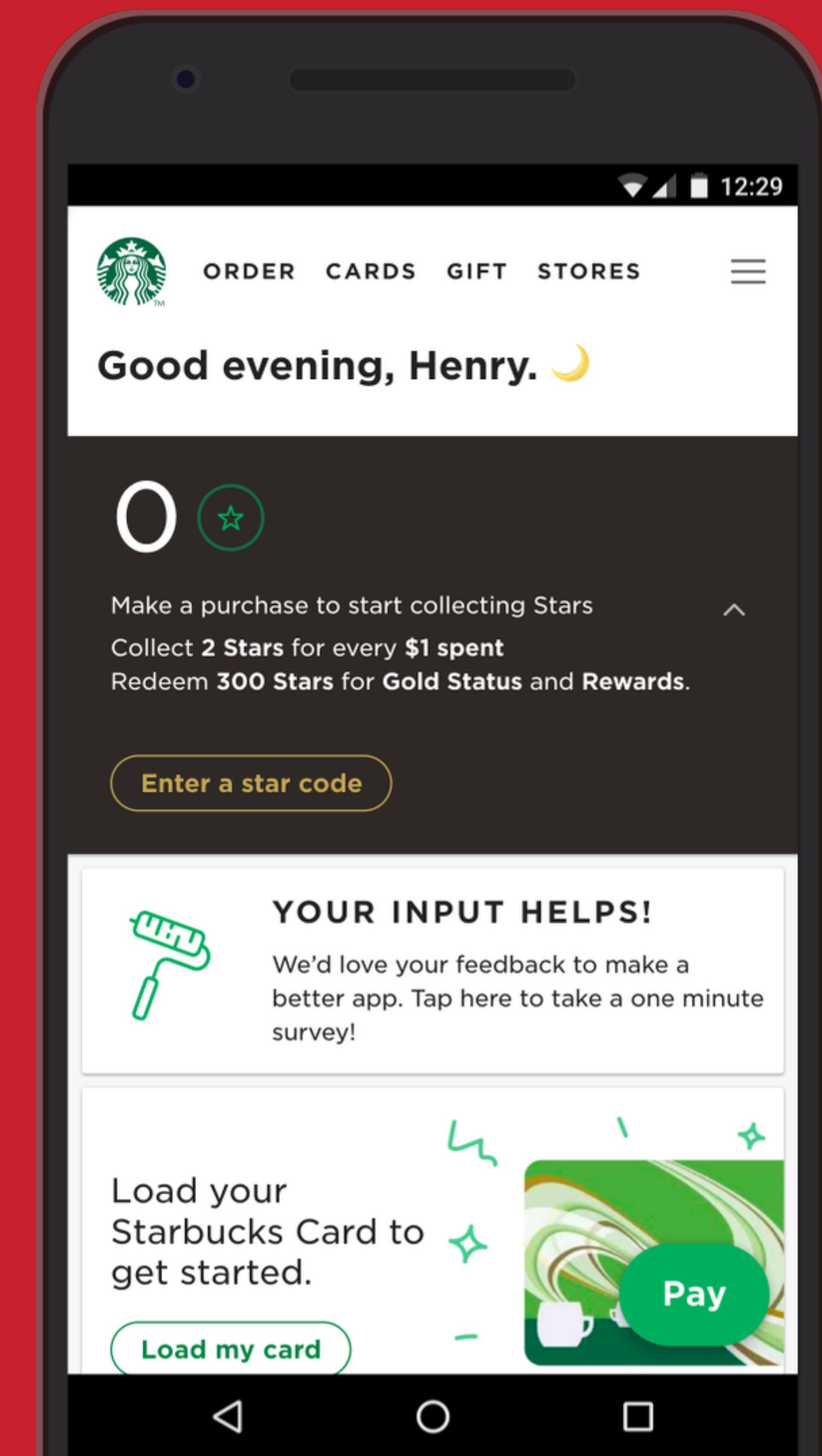
# pwa's are great!

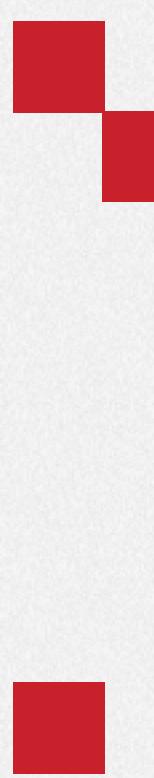


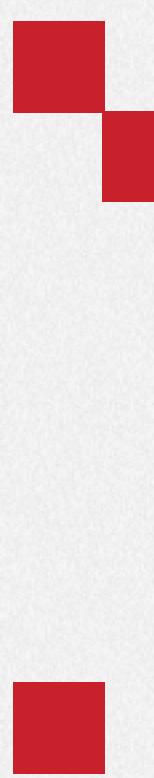
# but...

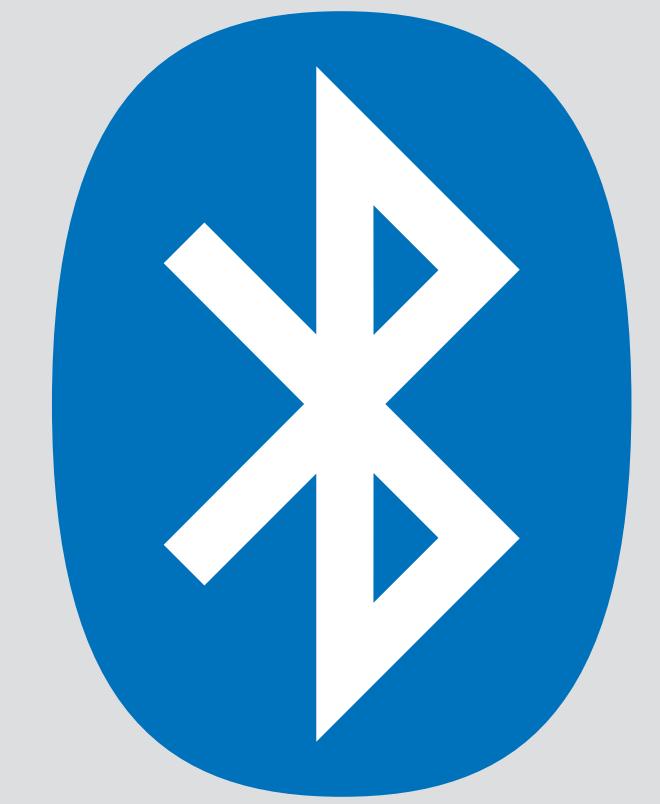


# but...

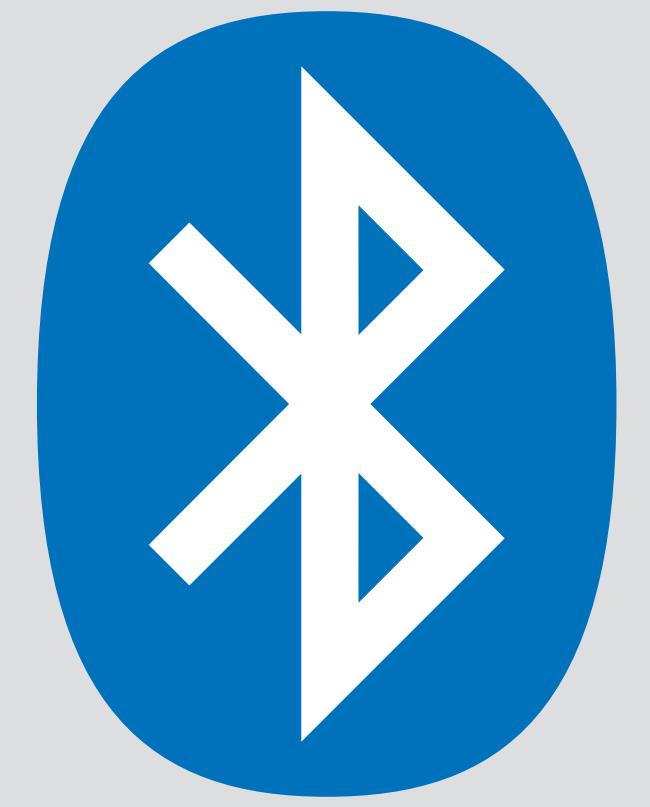




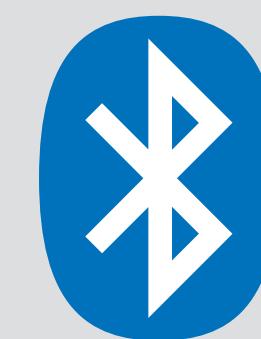




bluetooth

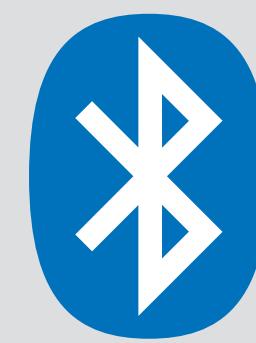


bluetooth  
sucks



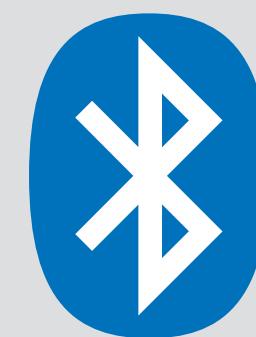
# classic bluetooth

the reason everybody  
hates bluetooth

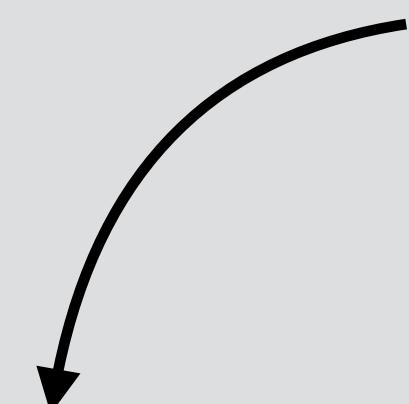


# classic bluetooth

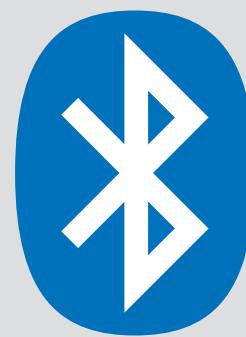
vs.



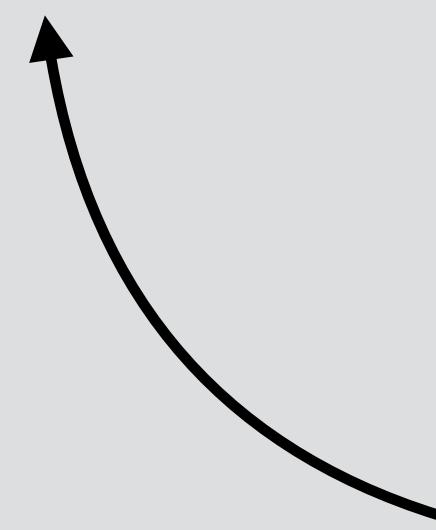
# bluetooth low energy



*control drones and other cool shit*



# bluetooth low energy



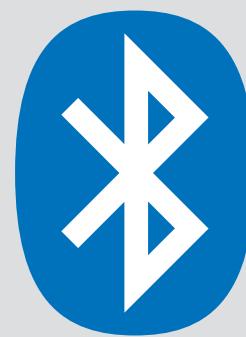
also known as

Bluetooth Smart

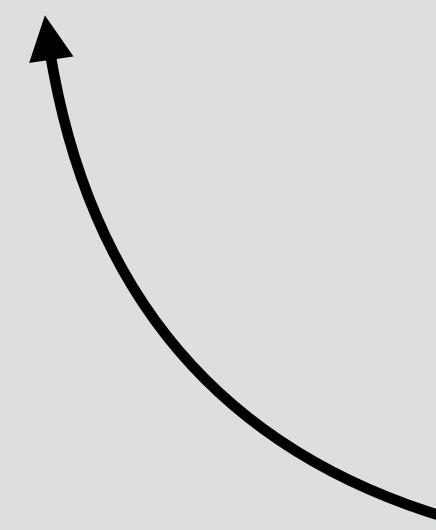
BLE

Bluetooth LE

Bluetooth 4



# bluetooth low energy



also known as

Bluetooth Smart

BLE

Bluetooth LE

Bluetooth 4 and 5

10 million  
bluetooth devices  
shipping every day



mobile phone

computer



glucose monitor

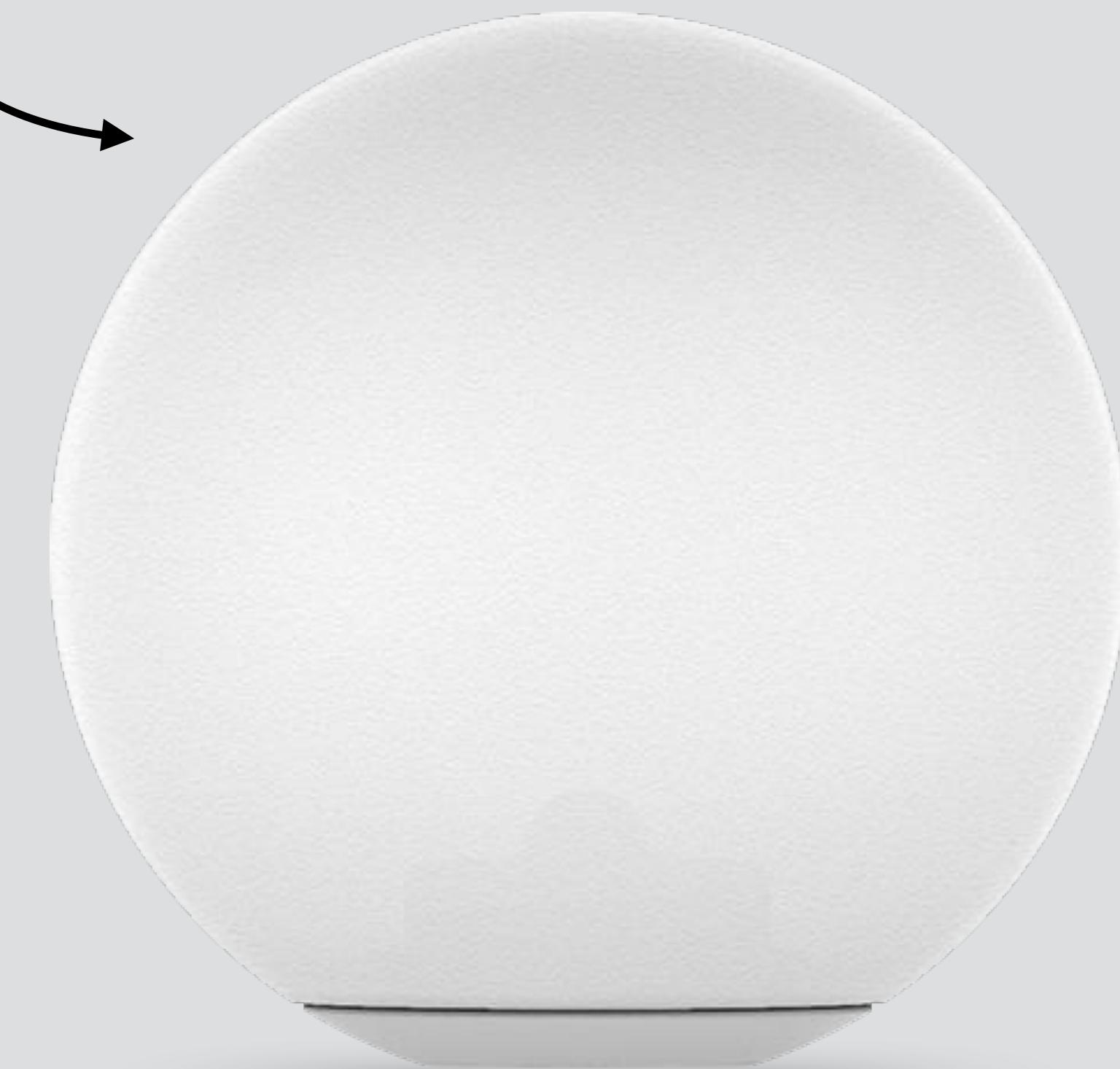


somebody's hand

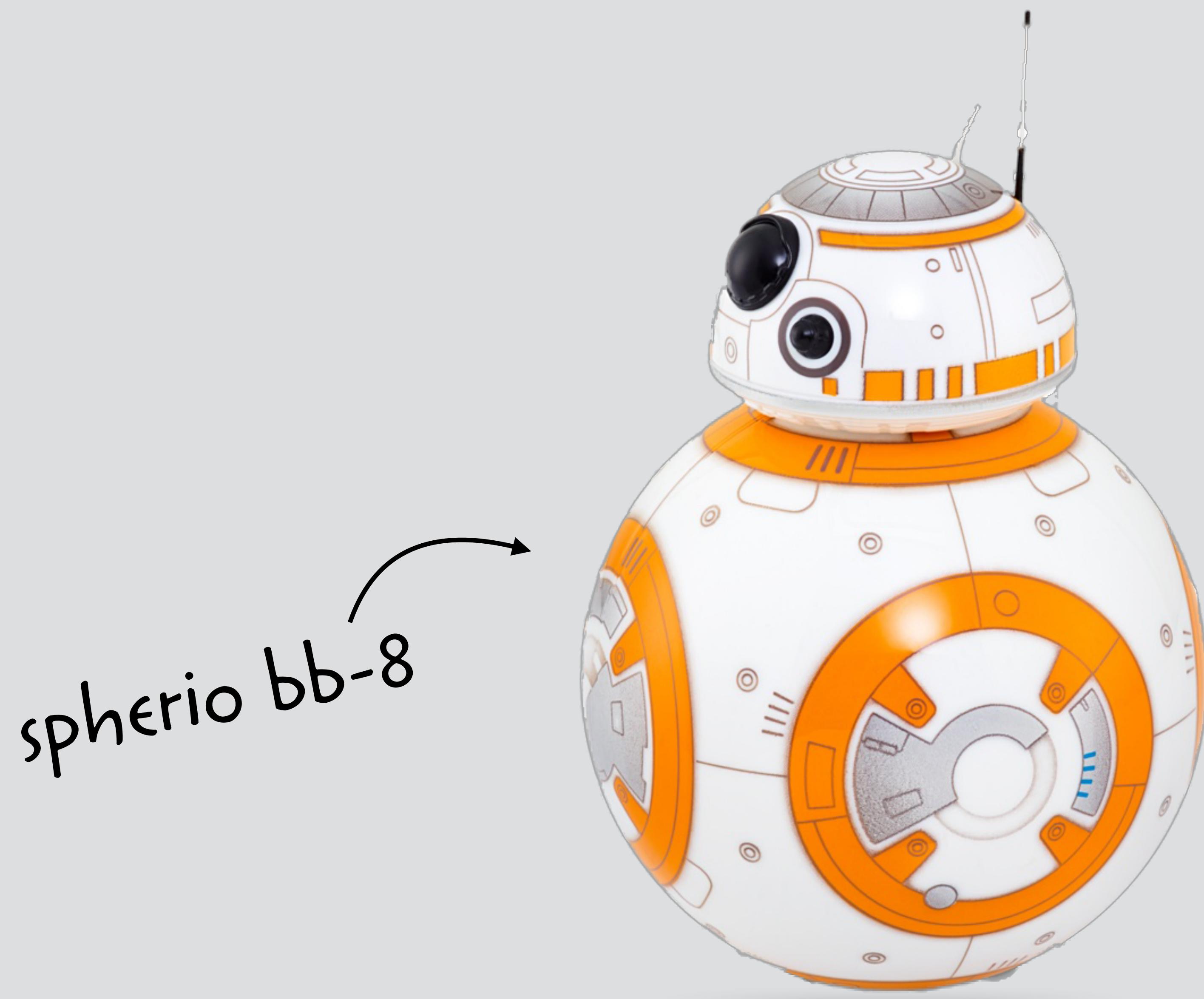
activity tracker



Playbulb sphere



playbulb





parrot mini drone



fidget spinner

the boring theoretical stuff



central



peripheral



central



# generic attribute profile

generic attribute profile ?

# generic attribute profile

gatt, because GAP was already taken

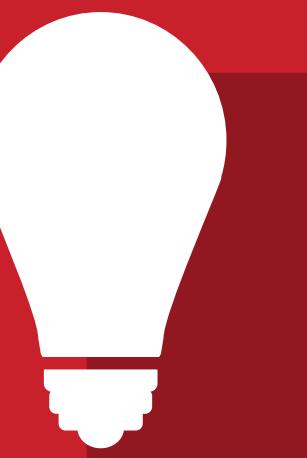


~~central~~  
~~client~~

~~peripheral~~  
~~server~~



device information



light

multiple services per device



device information



battery



flight control



## device information



battery



steering control



device information



battery



heart rate





device information



battery



heart rate



device information



battery



heart rate





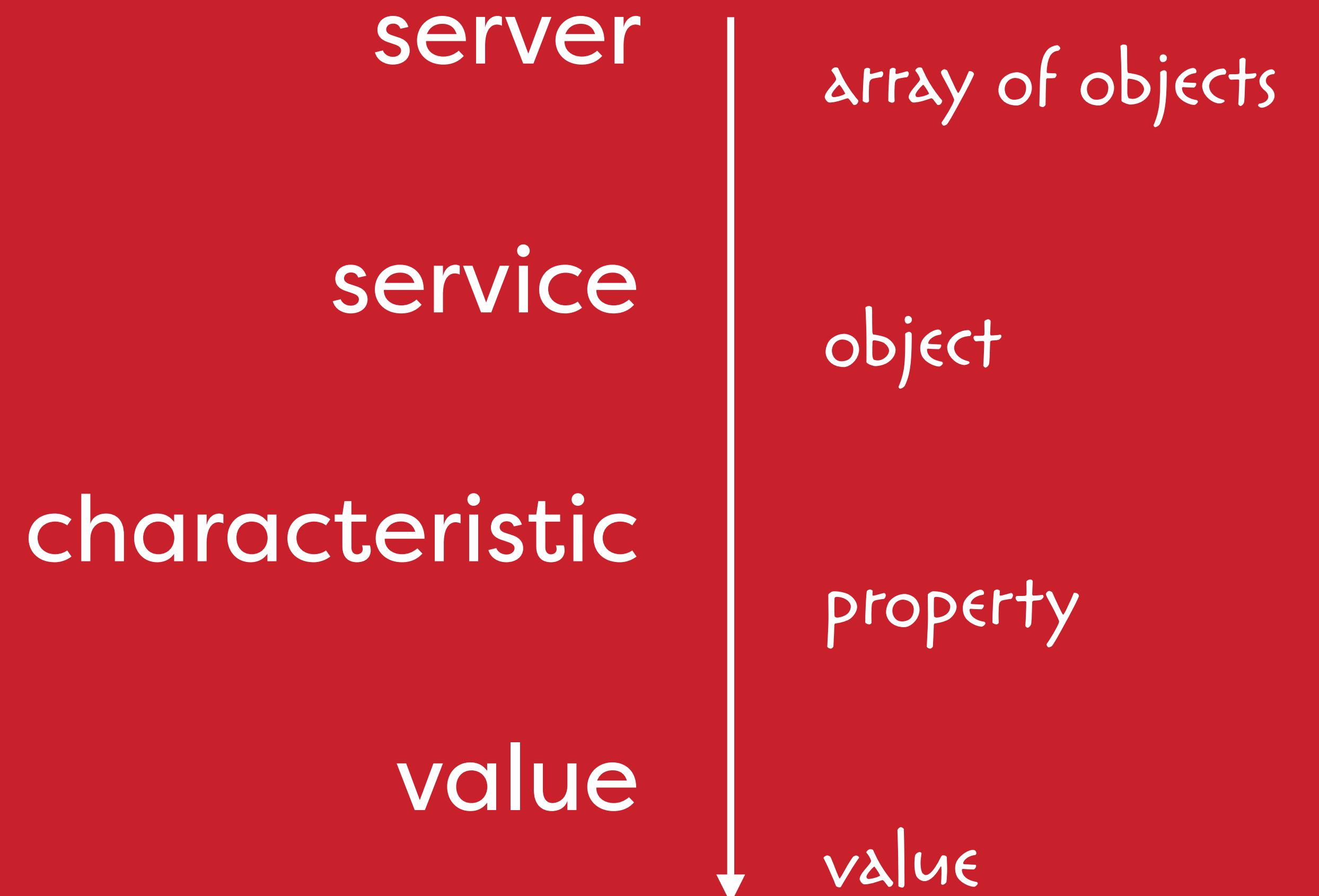
## device information

- manufacturer
- model number
- serial number
- hardware revision
- firmware revision
- software revision

...

multiple characteristics  
per service





services and characteristics  
are identified by uuid's

16 bit or 128 bit



## device information



16 bit    0x180A

128 bit    0000180A-0000-1000-8000-00805F9B34FB



battery



16 bit    0x180F

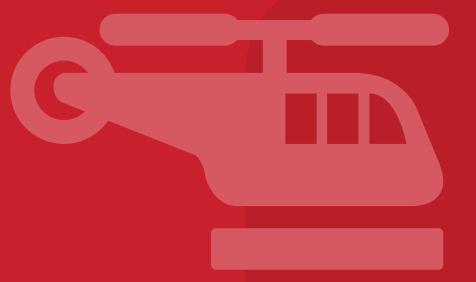
128 bit    0000180F-0000-1000-8000-00805F9B34FB



light



steering control



flight



*still, everybody does this*

16 bit    not recommended

128 bit    any UUID outside of the range

xxxxxxxx-0000-1000-8000-00805F9B34FB



## device information

manufacturer

model number

serial number

hardware revision

firmware revision

software revision

...



i

0x180A

0x2A29  
0x2A24  
0x2A25  
0x2A27  
0x2A26  
0x2A28  
...



good for bad for saving readability,  
bandwidth

each characteristic supports  
one or more of these



read

write

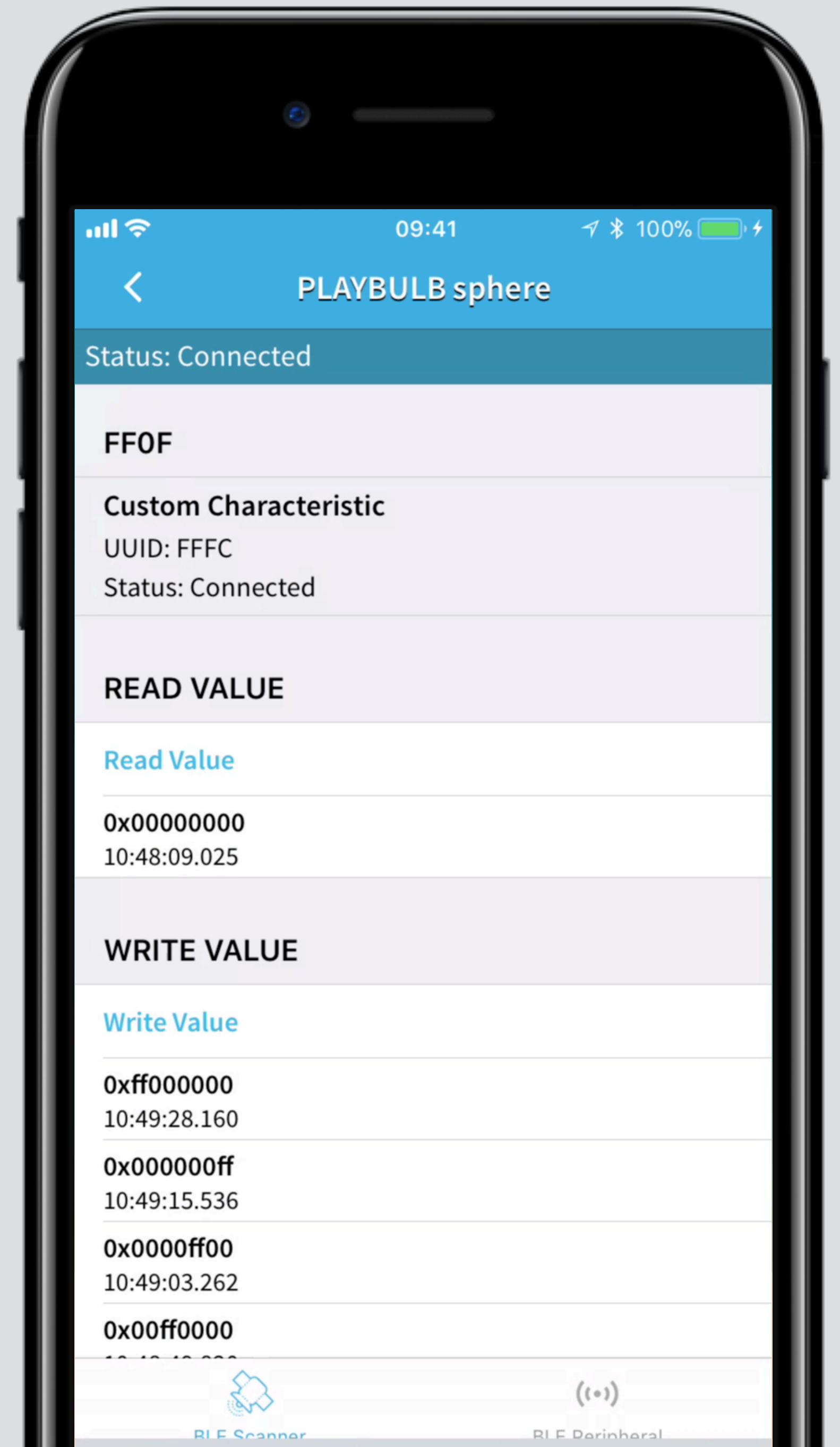
write without response

notify

every value is an array of bytes

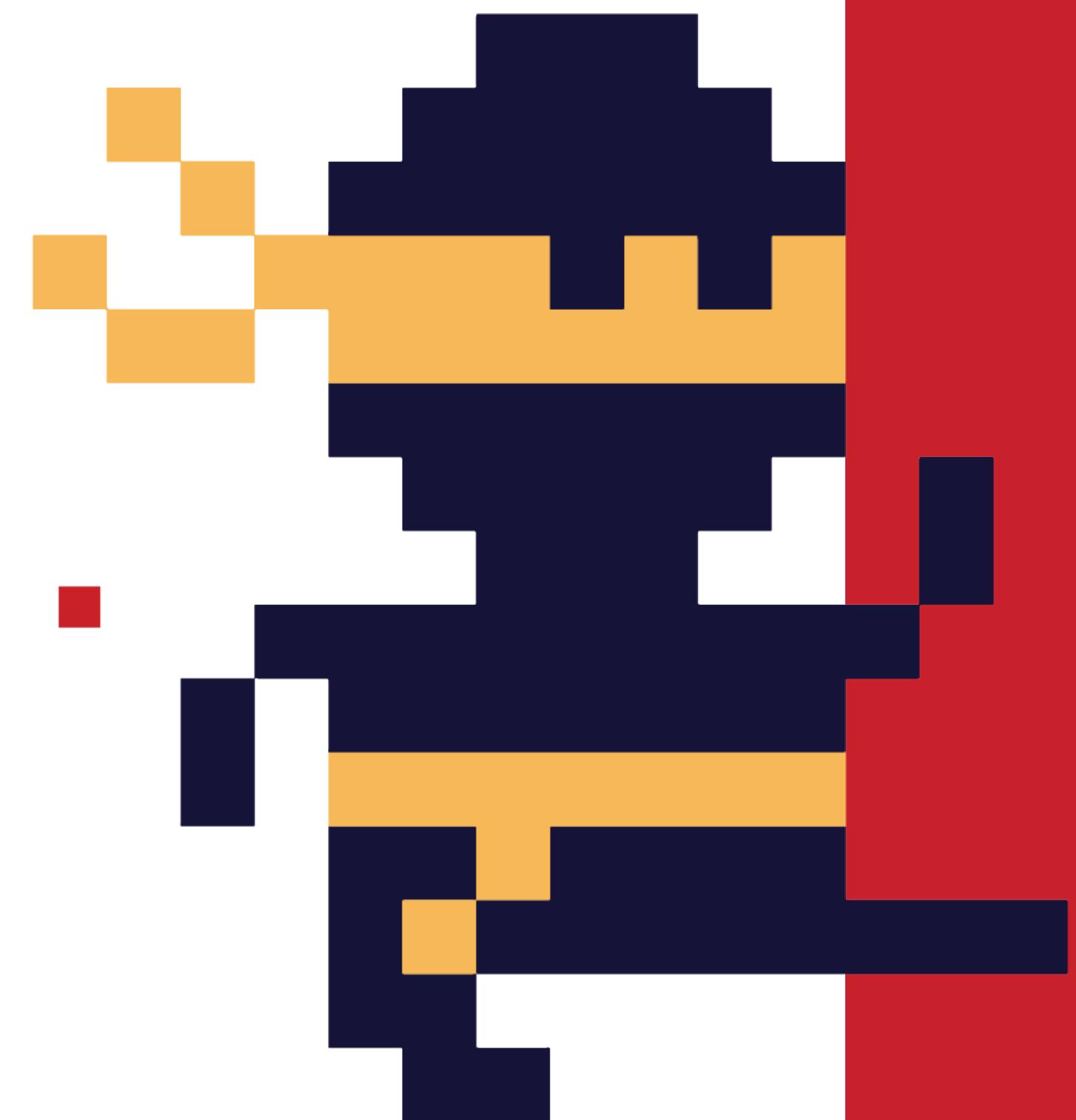
no fancy datatypes, just bytes

pfew...



ODESSAJS

July 20-21, Odessa



boring facts  
Fun about  
with  
**Bluetooth**

ODESSAJS

July 20-21, Odessa



Fun with  
Bluetooth

# web bluetooth. api

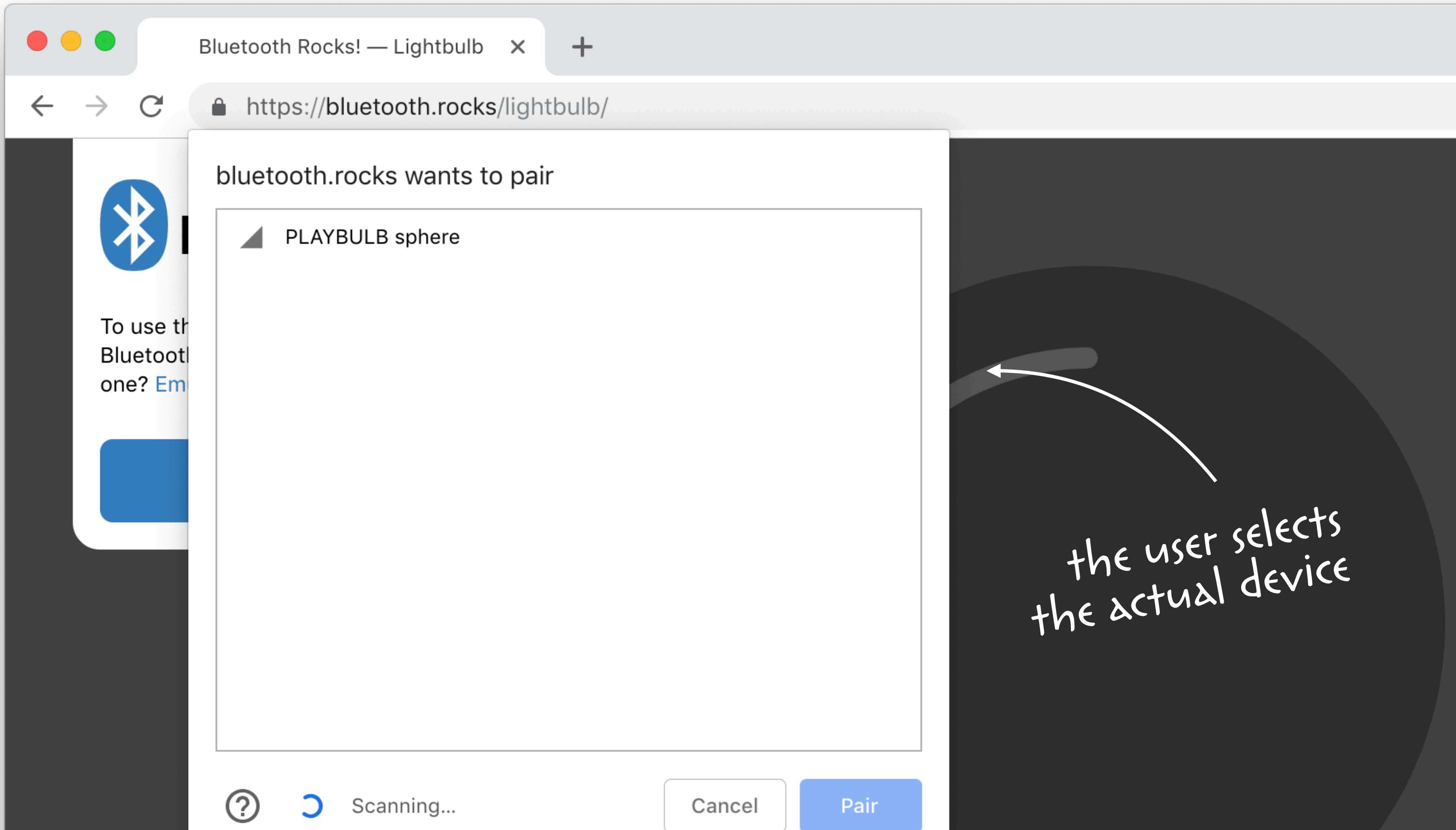
*still not the fun part*  
:-(  


# connecting to a device

```
navigator.bluetooth.requestDevice({  
  filters: [  
    { namePrefix: 'PLAYBULB' }  
,  
  optionalServices: [ 0xff0f ]  
})
```



we tell the browser what  
kind of device we want



```
navigator.bluetooth.requestDevice({  
  filters: [  
    { namePrefix: 'PLAYBULB' }  
,  
  optionalServices: [ 0xff0f ]  
})  
.then(device => {  
  ....  
})
```

promises are so 2017

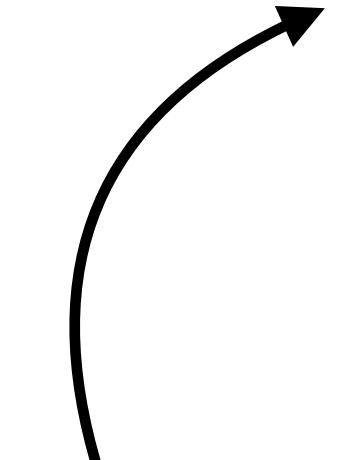
```
let device = await navigator.bluetooth.requestDevice({  
  filters: [  
    { namePrefix: 'PLAYBULB' }  
,  
  optionalServices: [ 0xff0f ]  
} );
```

```
let server = await device.gatt.connect());
```



connect to the server

```
let service = await server.getPrimaryService(0xff0f));
```



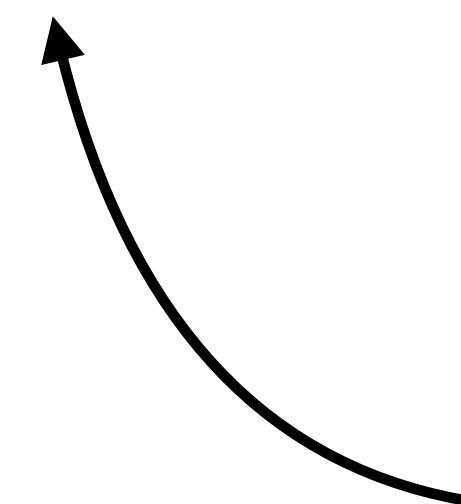
get the service

```
let characteristic =  
    await service.getCharacteristic(0xffffc));
```

get the characteristic

# writing data

```
characteristic.writeValue(  
    new Uint8Array([ 0x00, r, g, b ])  
) ;
```

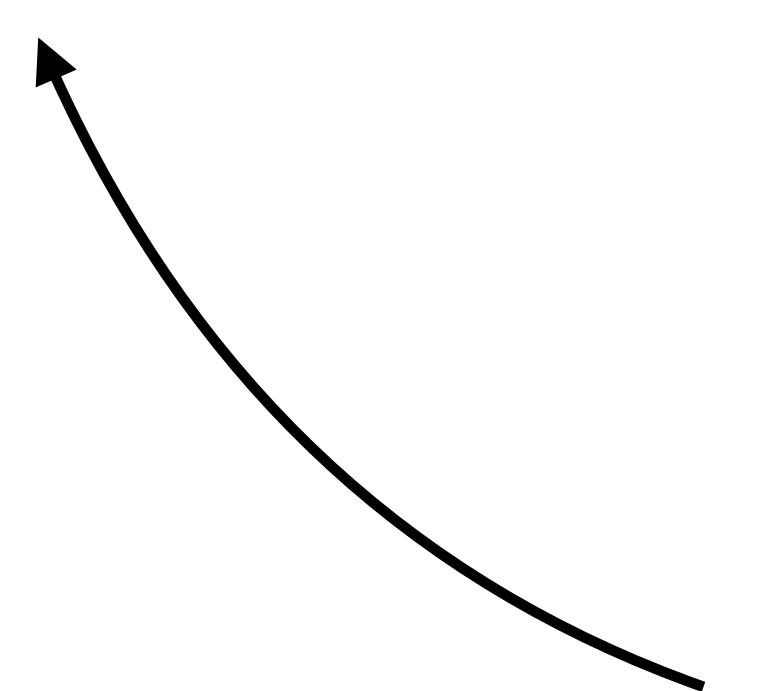


write some bytes

# reading data

```
let value = await characteristic.readValue();
```

```
let r = value.getUint8(1);  
let g = value.getUint8(2);  
let b = value.getUint8(3);
```



read some bytes

get notified of changes

```
characteristic.addEventListener(  
    'characteristicvaluechanged', e => {  
        let r = e.target.value.getUint8(1);  
        let g = e.target.value.getUint8(2);  
        let b = e.target.value.getUint8(3);  
    }  
);
```

```
characteristic.startNotifications();
```

add event listener  
don't forget to start listening

# things you need to know:

- the webbluetooth api
- promises (or async await)
- typed arrays

duh!

# browser support



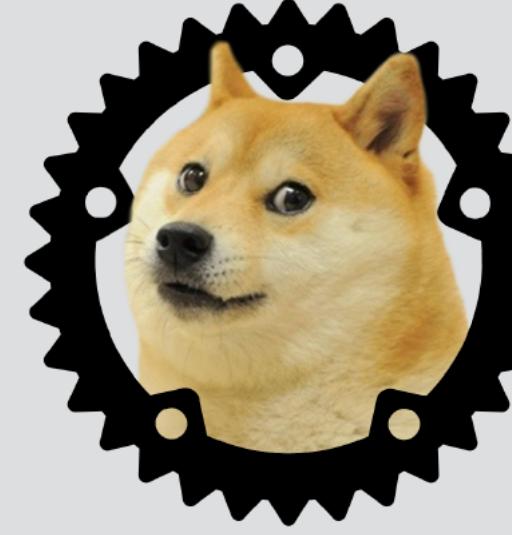
Chrome



Edge  
(Chromium)



Samsung



Servo  
(soon)

# browser support



Chrome



Edge  
(Chromium)



Samsung



Servo  
(soon)



iOS

# browser support



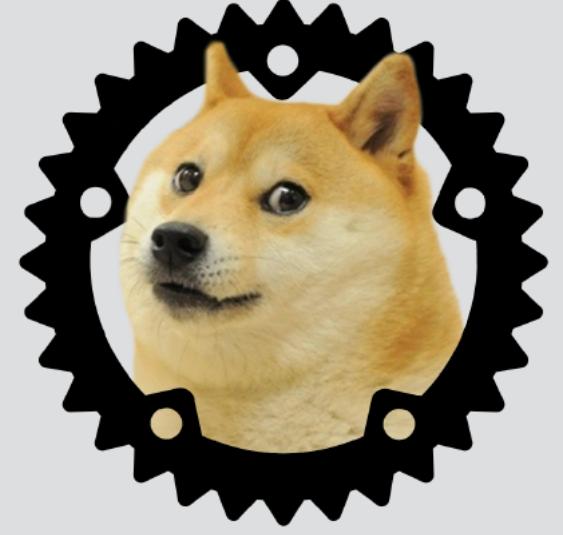
Chrome



Edge  
(Chromium)



Samsung



Servo  
(soon)



iOS

# browser support



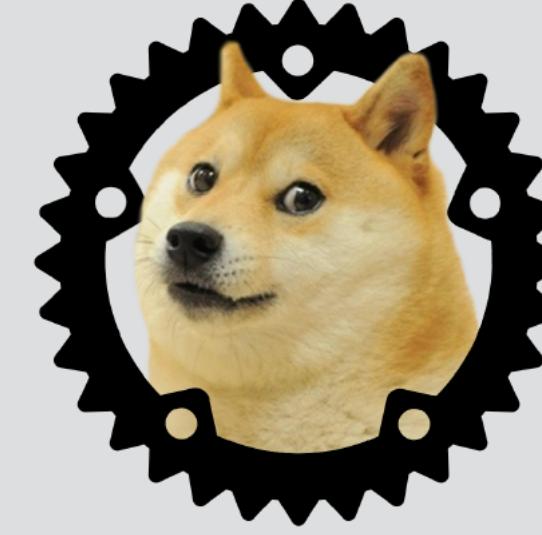
Chrome



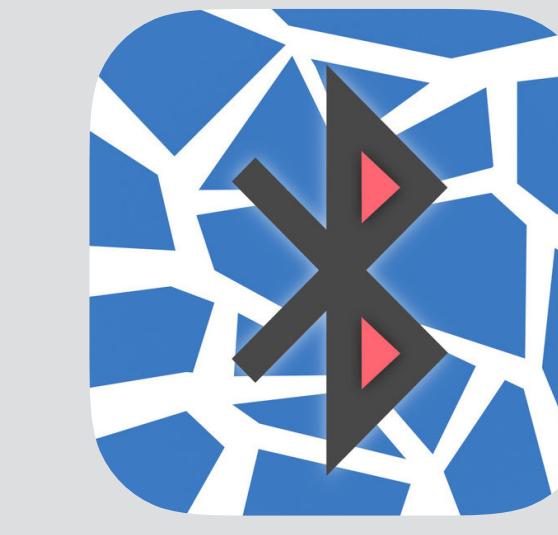
Edge  
(Chromium)



Samsung



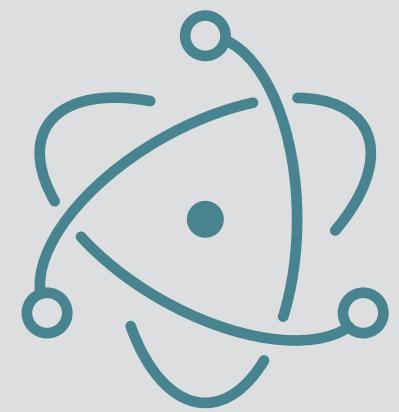
Servo  
(soon)



WebBLE  
for iOS

kinda works

and...



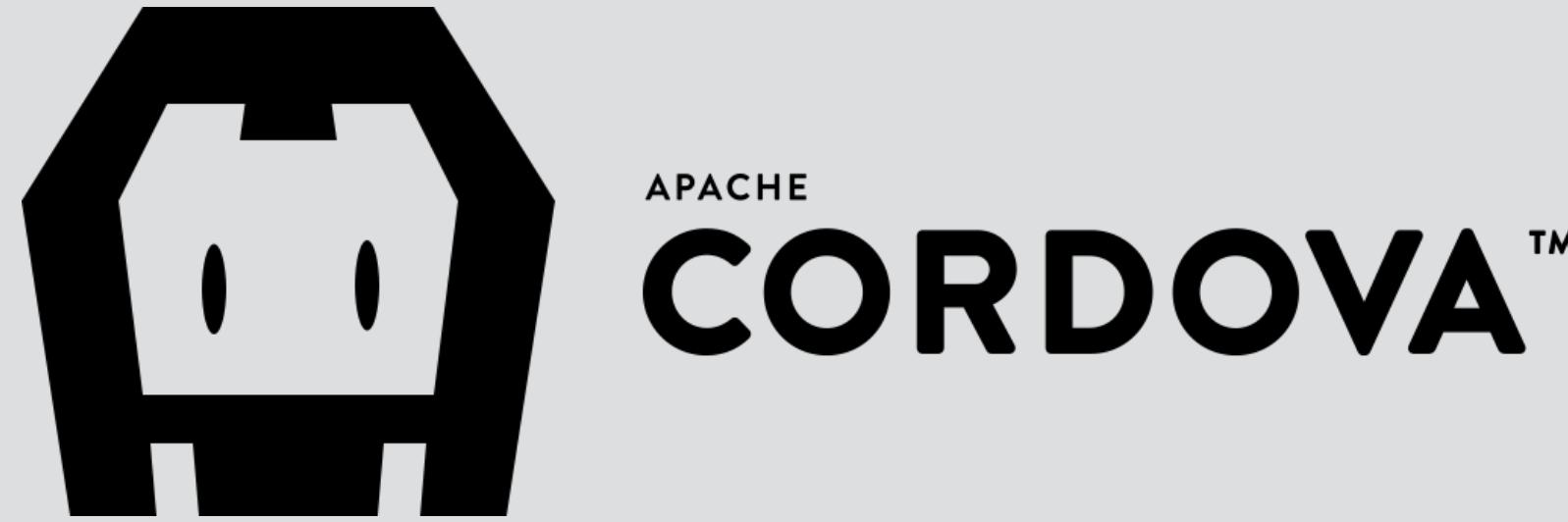
ELECTRON

and...



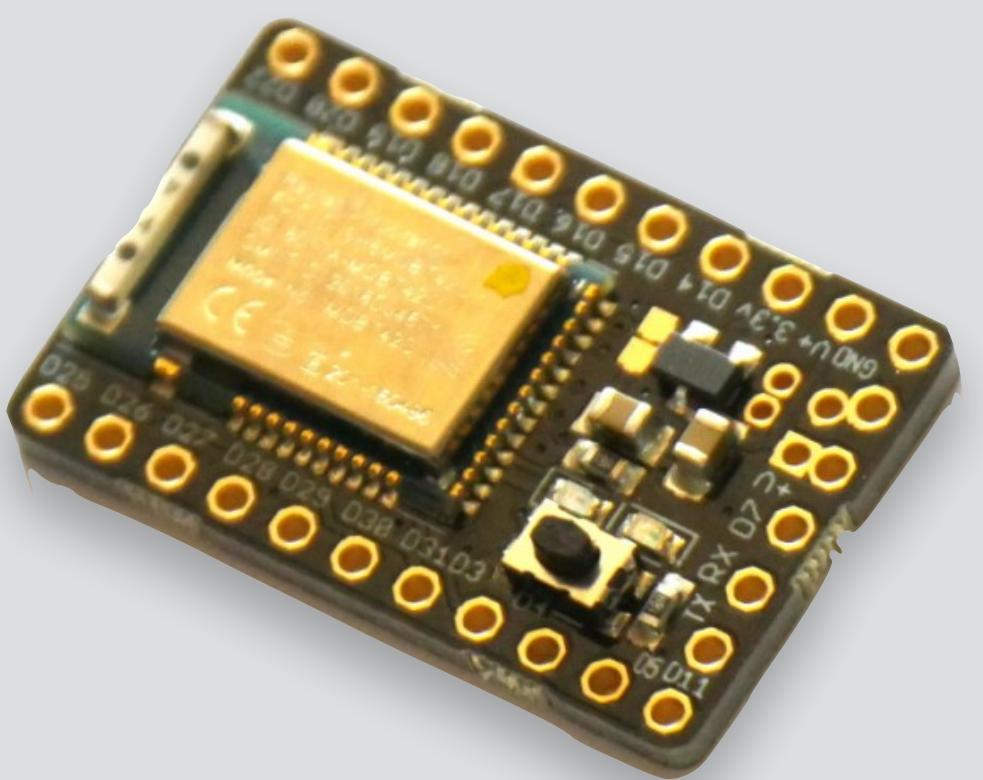
npm install node-web-bluetooth

and...



```
npm install cordova-plugin-webbluetooth
```

# and... espruino



18 × 28 mm

custom  
characteristics wtf!

```
function(r, g, b) {  
    return new Uint8Array([ 0x00, r, g, b ]);  
}  
  
function(buffer) {  
    return {  
        r: buffer.getUint8(1),  
        g: buffer.getUint8(2),  
        b: buffer.getUint8(3)  
    }  
}
```



```
function(r, g, b) {  
    return new Uint8Array([  
        0x01, g, 0x01, 0x00, 0x01,  
        b, 0x01, r, 0x01, 0x00  
    ]);  
}  
}
```



```
function(r, g, b) {  
    var buffer = new Uint8Array([  
        0xaa, 0x0a, 0xfc, 0x3a, 0x86, 0x01, 0x0d,  
        0x06, 0x01, r, g, b, 0x00, 0x00,  
        (Math.random() * 1000) & 0xff, 0x55, 0x0d  
    ]);  
  
    for (var i = 1; i < buffer.length - 2; i++) {  
        buffer[15] += buffer[i];  
    }  
  
    return buffer;  
}
```

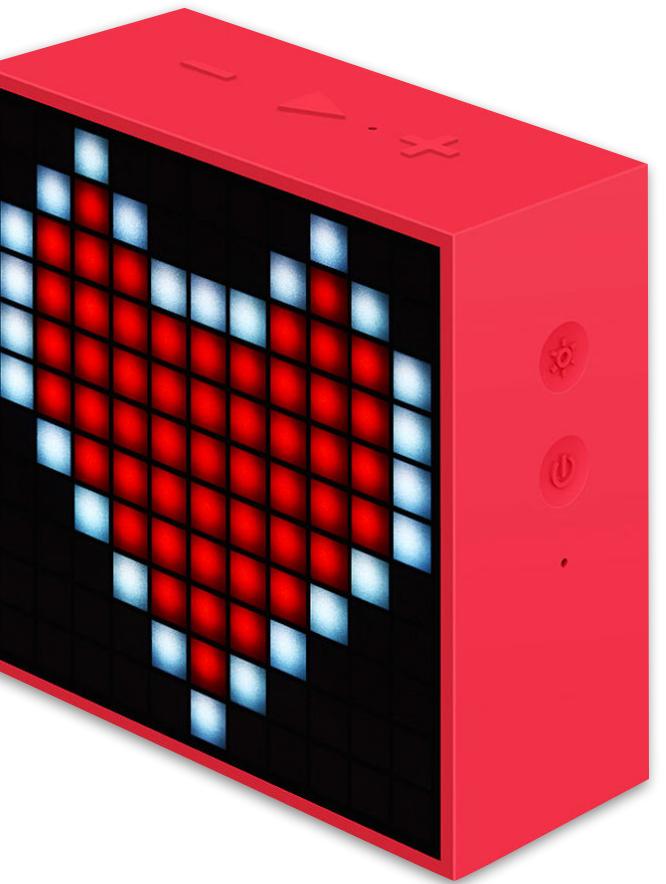


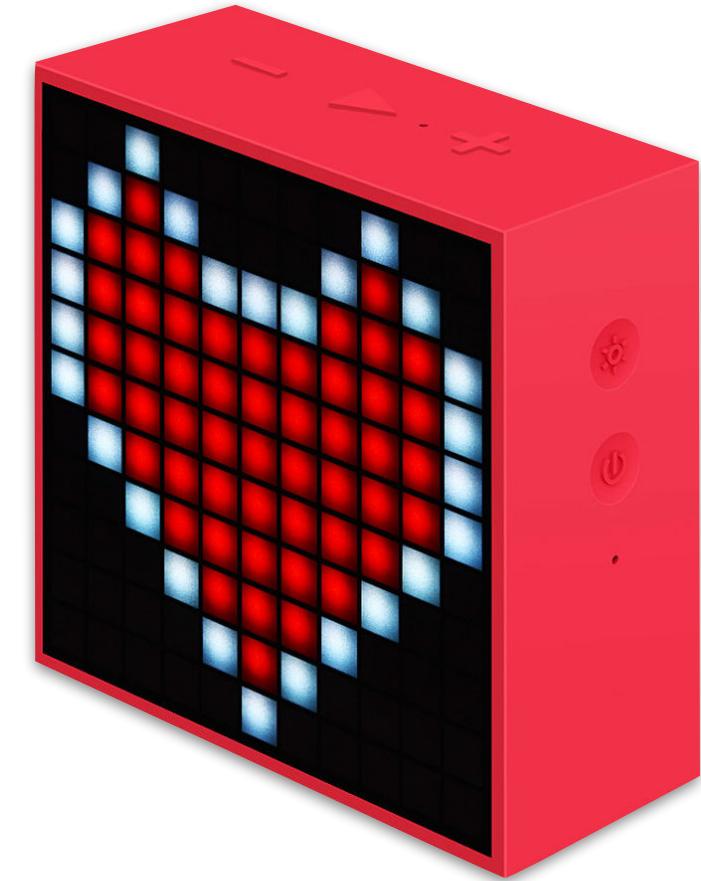
```
function(r, g, b, position) {  
    let buffer = new Uint8Array([  
        0x07, 0x02, position + 1, r, g, b  
    ]);  
  
    return buffer;  
}
```



```
function(r, g, b, position) {  
  let buffer = new Uint8Array([  
    0x58, r, g, b, 0x01, position  
  ]);
```

...



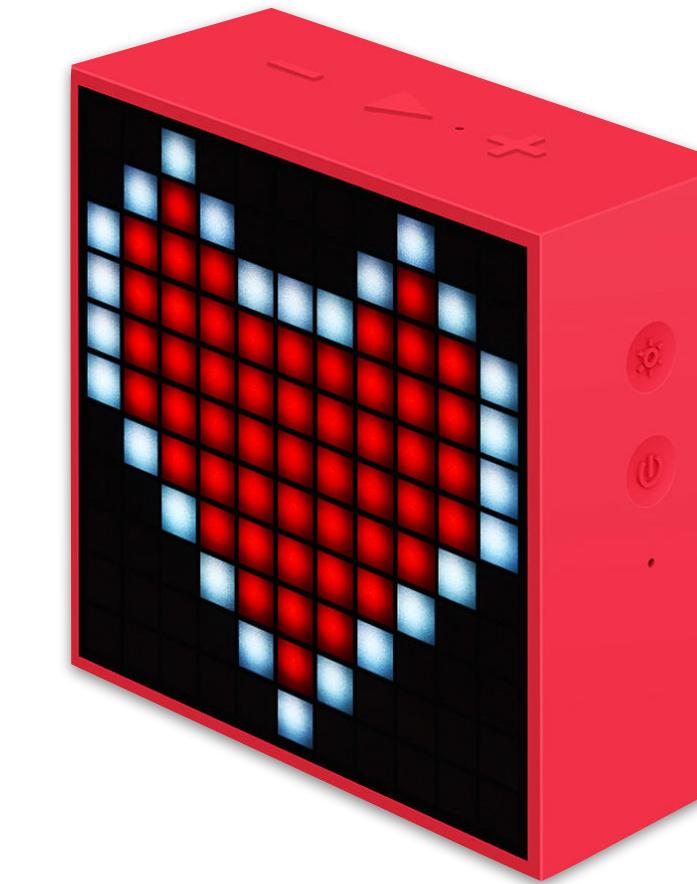


```
function(r, g, b, position) {  
    let buffer = new Uint8Array([  
        0x58, r, g, b, 0x01, position  
    ]);  
  
    let payload = new Uint8Array(buffer.length + 4);  
    payload[0] = payload.length - 2;  
    payload[1] = payload.length - 2 >>> 8;  
    payload.set(buffer, 2);  
  
    let checksum = payload.reduce((a, b) => a + b, 0);  
    payload[payload.length - 2] = checksum;  
    payload[payload.length - 1] = checksum >>> 8;  
  
    let extra = payload.filter(value => {
```

```
        message[m] = 0x05;
        message[m + 1] = 0x06;
        m += 2;
    }
    else {
        message[m] = payload[i];
        m++;
    }
}

message[0] = 0x01;
message[message.length - 1] = 0x02;

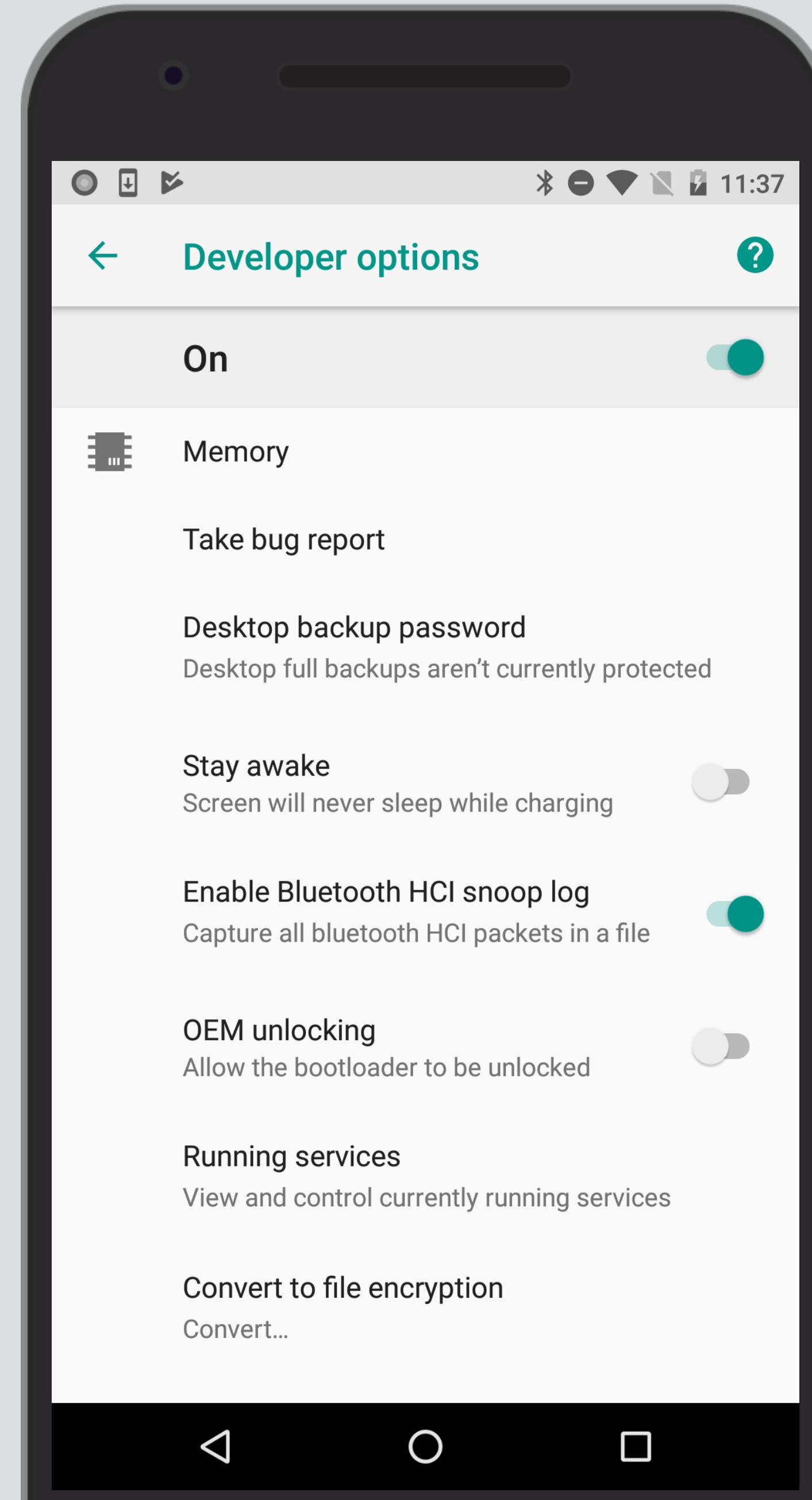
return message;
}
```





adafruit  
bluetooth  
sniffer

# log all bluetooth packets on your phone



and use `adb logcat`  
to transfer the log

btsnoop\_hci.log

Apply a display filter ... <⌘>/>

No.	Time	Source	Destination	Protocol	Length	Value
3620	118.078099	controller	host	HCI_EVT	8	
3621	118.137911	LgElectr_6c:eb:b9 ...	remote ()	ATT	32	fa030039010006000000000000000000000000003c300
3622	118.140070	LgElectr_6c:eb:b9 ...	remote ()	ATT	32	00000003c30000000003c30000000000fc000000
3623	118.142233	LgElectr_6c:eb:b9 ...	remote ()	ATT	32	000030000000000003000000000003000000000000
3624	118.144795	LgElectr_6c:eb:b9 ...	remote ()	ATT	16	000b55a9
3625	118.166811	controller	host	HCI_EVT	8	
3626	118.166955	controller	host	HCI_EVT	8	
3627	118.167380	controller	host	HCI_EVT	8	
3628	118.168192	controller	host	HCI_EVT	8	
3629	118.230426	LgElectr_6c:eb:b9 ...	remote ()	ATT	32	fa030039010006000000000000000000000000f0c000
3630	118.232516	LgElectr_6c:eb:b9 ...	remote ()	ATT	32	000000f0c000000000f0c0000000003f0f000000
3631	118.234390	LgElectr_6c:eb:b9 ...	remote ()	ATT	32	000c0f000000000c0f00000000c0000000000000
3632	118.236671	LgElectr_6c:eb:b9 ...	remote ()	ATT	16	000b55a9
3633	118.256972	controller	host	HCI_EVT	8	
3634	118.257299	controller	host	HCI_EVT	8	
3635	118.258136	controller	host	HCI_EVT	8	
3636	118.258542	controller	host	HCI_EVT	8	
3637	118.322899	LgElectr_6c:eb:b9 ...	remote ()	ATT	32	fa0300390100060000000000000000000000003c300000

▶ Frame 3629: 32 bytes on wire (256 bits), 32 bytes captured (256 bits)  
 ▶ Bluetooth  
 ▶ Bluetooth HCI H4  
 ▶ Bluetooth HCI ACL Packet  
 ▶ Bluetooth L2CAP Protocol  
 ▶ Bluetooth Attribute Protocol  
 ▶ Opcode: Write Command (0x52)  
 ▶ Handle: 0x0011 (9ecadc240ee5a9e093f3a3b50200406e)  
 Value: fa030039010006000000000000000000000000f0c000

0000	02 02 00 1b 00 17 00 04 00 52 11 00 fa 03 00 39	..... R....9
0010	01 00 06 00 00 00 00 00 00 00 00 f0 c0 00	..... .....

Value (btatt.value), 20 bytes

Packets: 4148 · Displayed: 4148 (100.0%) · Load time: 0:0.7

use wireshark to  
look at the data

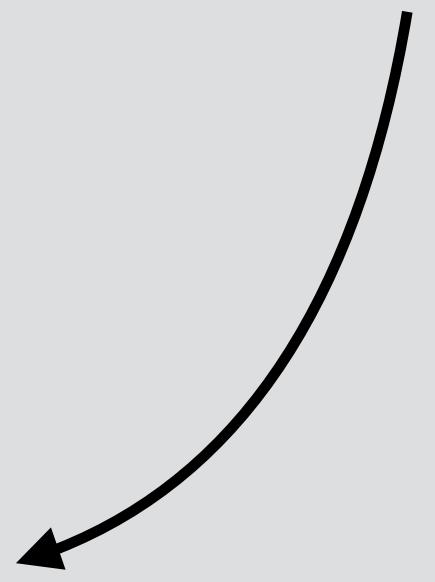
# decompiling the apk



don't tell anyone!

# demo

*finally the fun part*





**warning**  
experimental technology

setting low expectations



# warning

## wifi interference

lowering them even further



ODESSAJS

July 20-21, Odessa



# Fun with Bluetooth

# questions?

@html5test