Easy does it

Simpler App development using JavaScript and Bluetooth

Peter Svensson, Lead developer Evothings AB
@psvensson

Sure, but why?

Why do web-based Apps in JS instead

of platform-specific Apps?

- Cross-platform
- Leveraging Web-based UI libs
- Less LoCs
- Quicker reloads when developing

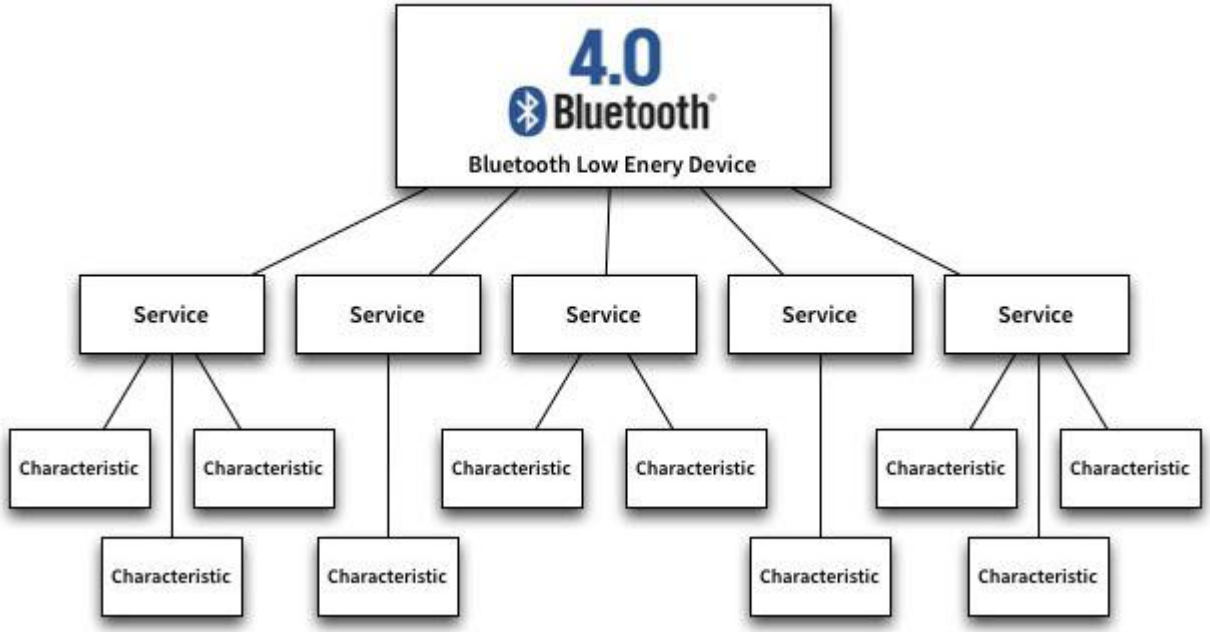
IoT: WAT (lightning recap)

IoT -> Lots of very small, ‘headless’ computers that send and receive data in ways normal computers generally doesn’t.

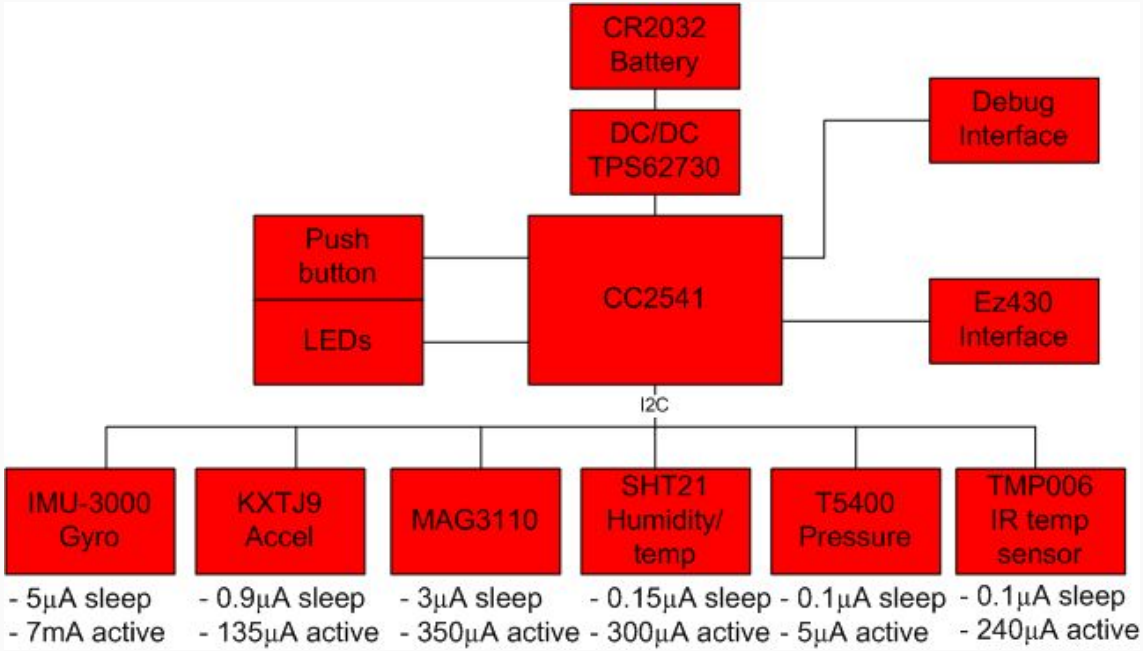
This talk focus on one of these ways - Bluetooth (or more specifically BLE)

BLE

Stuff inside a BLE device is organized into services (like an accelerometer) which has characteristics (like Z-axis)

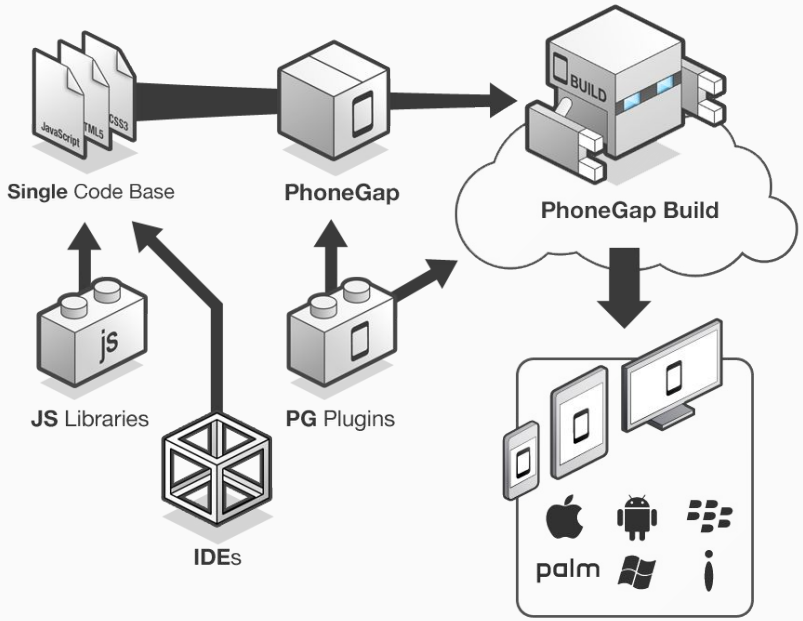


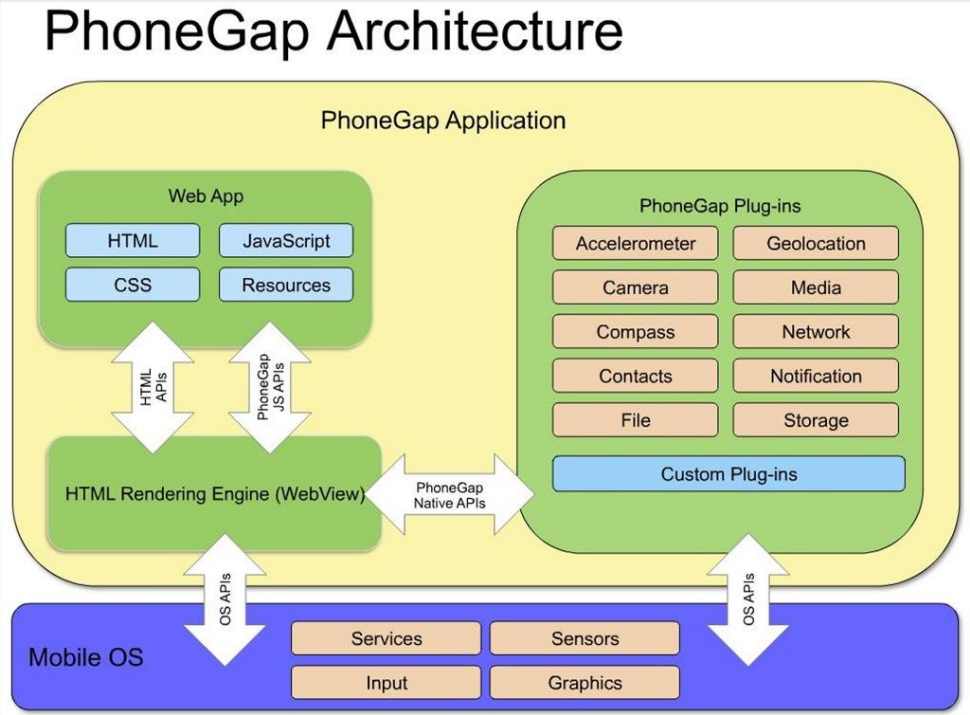
MOAR BLE


CR2032 Battery
DC/DC TPS62730
Debug Interface
Push button
LEDs
CC2541
Ez430 Interface
I2C
IMU-3000 Gyro
KXTJ9 Accel
MAG3110
SHT21 Humidity/ temp
T5400 Pressure
TMP006 IR temp sensor
- 5µA sleep
- 7mA active
- 0.9µA sleep
- 135µA active
- 3µA sleep
- 350µA active
- 0.15µA sleep
- 300µA active
- 0.1µA sleep
- 5µA active
- 0.1µA sleep
- 240µA active

Web Apps - Mobile phones - Devices

Web apps on mobile? (even quicker recap)





Cordova BLE plugin

Two years ago it was hard to find good BLE plugins for Cordova

So we made one (https://github.com/evothings/cordova-ble)

- Scan for BLE devices
- Establish connections
- List services, characteristics and descriptors
- Read and write the values of characteristics and descriptors
- Request notification of changes to the values of characteristics
- Poll RSSI (signal strength) of a device (Android and iOS only)

Cordova BLE plugin

```
getServices: function(deviceHandle)
{
        app.displayStatus('Reading services...');

        evothings.ble.readAllServiceData(deviceHandle, function(services)
        {
                // Find handles for characteristics and descriptor needed.
                for (var si in services)
                {
                        var service = services[si];

                        for (var ci in service.characteristics)
                        {
                                var characteristic = service.characteristics[ci];

                                if (characteristic.uuid == '713d0002-503e-4c75-ba94-3148f18d941e')
                                {
                                        app.characteristicRead = characteristic.handle;
                                }
                                else if (characteristic.uuid == '713d0003-503e-4c75-ba94-3148f18d941e')
                                {
                                        app.characteristicWrite = characteristic.handle;
                                }
```

EasyBLE - removing (some) boilerplate

```
app.readServices = function(device)
{
        device.readServices(
                [
                app.sensortag.MOVEMENT_SERVICE // Movement service UUID.
                ],
                // Function that monitors accelerometer data.
                app.startAccelerometerNotification,
                function(errorCode)
                {
                        console.log('Error: Failed to read services: ' + errorCode + '.');
                });
};
```

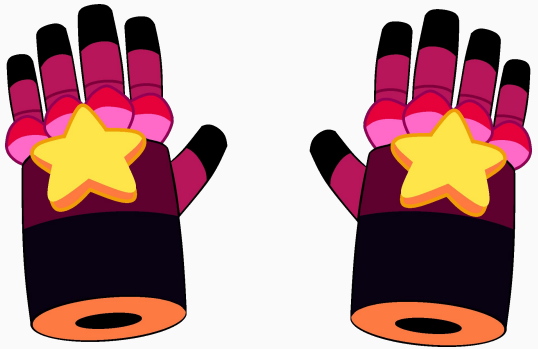
Developer Problems

Hard to stay in the ‘zone’ when developing on/for hardware

Packaging and deployment breaks concentration

When developing for IoT you’re on (at least) two devices

Hard to find good examples



Debugging on device (mobile phone)

Emulators are all fine but…

Solving BLE problems really do have to be done on the device singing BLE

But how do we get the logs back to the developer?

Building Cordova Apps

Takes some time

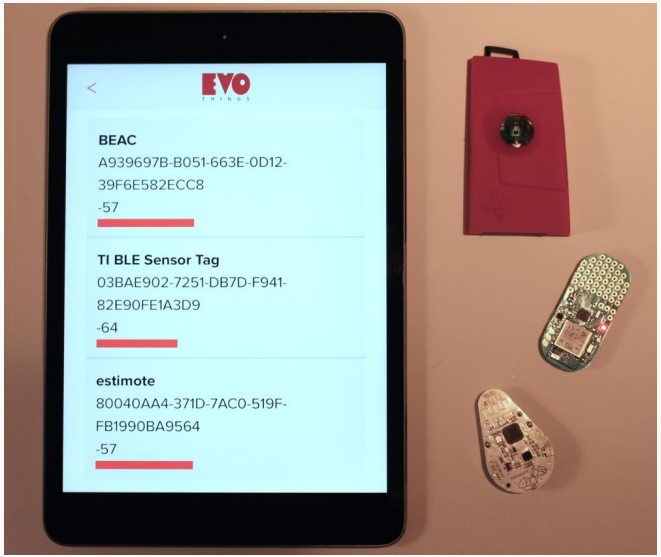
New apps has to be uploaded to the phone (obviously)

Wouldn’t it be nice to just quickly run and test your code without building?

Evothings Studio + Viewer

- The viewer is a small app for iOS and Android
- Contain a lot of Cordova plugins (BLE, for example)
- Connects to Studio

- The studio is a Linux, Windows or Mac application
- It manages one or more viewers
- It pushes out JS file changes in projects to the viewers



Yes, it’s open source :)

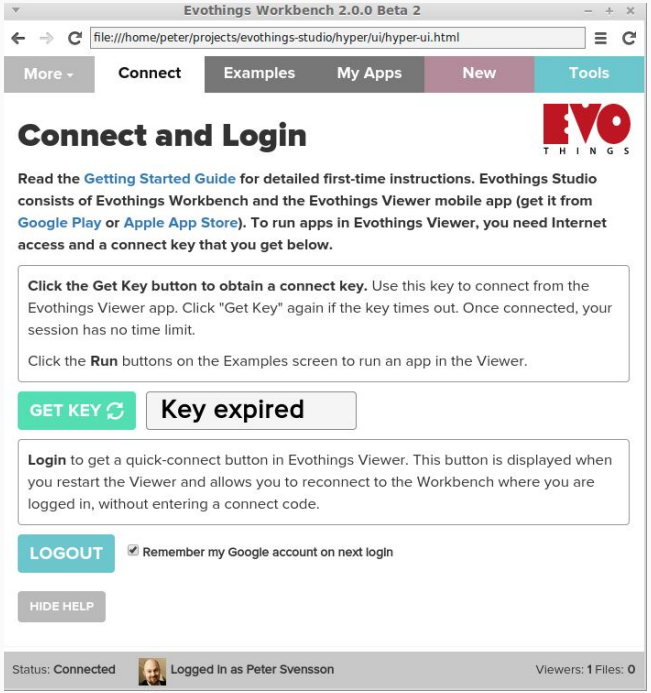
https://github.com/evothings/evothings-studio

https://github.com/evothings/evothings-viewer

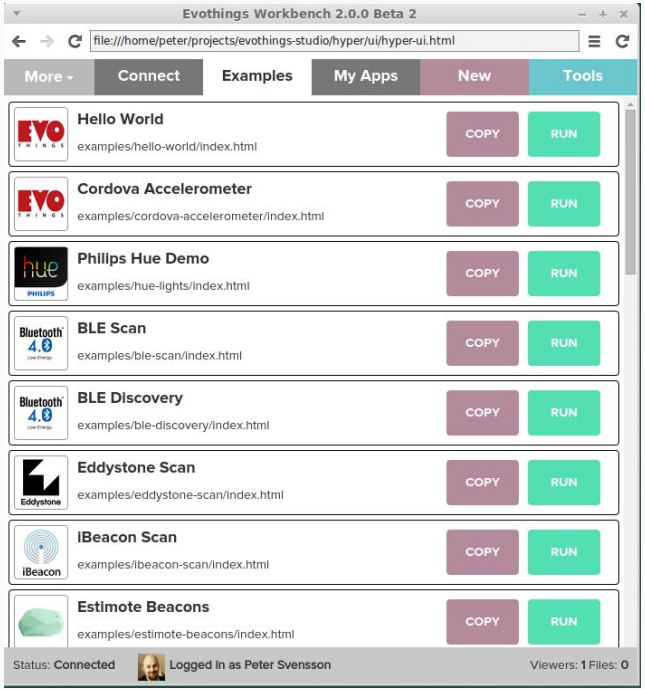
https://evothings.com/doc/starter-guides/evothings-studio-starter-guide.html

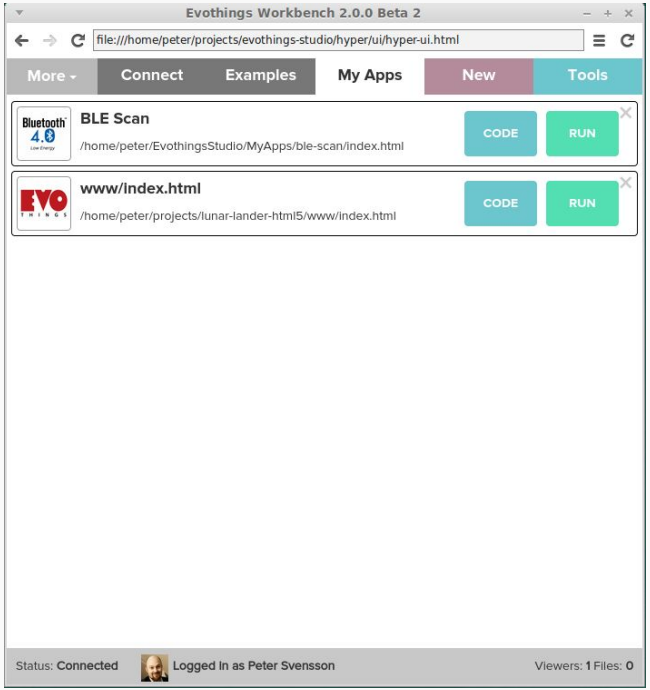
So, anyway

Demo time

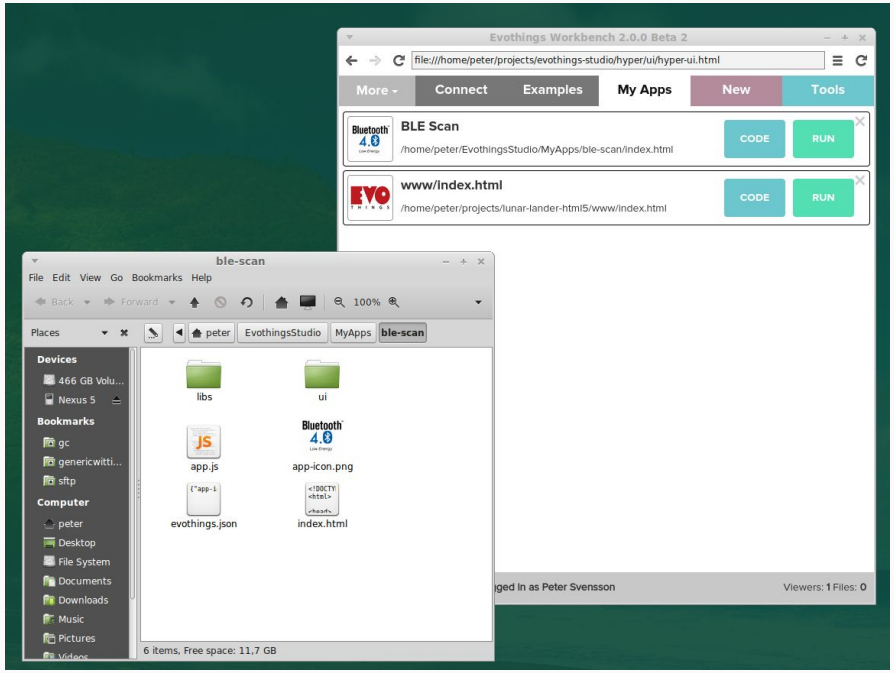


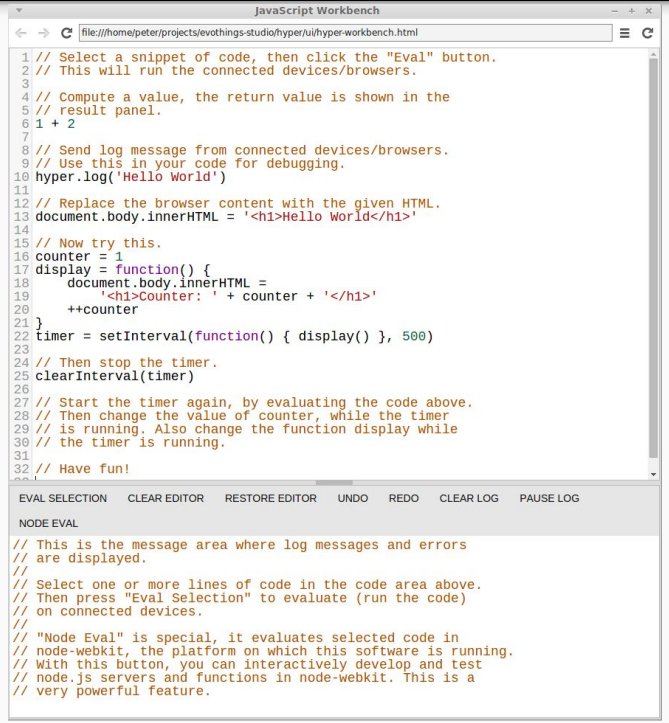
In case of borketh network





Bork bork ...


Evothings Workbench 2.0.0 Beta 2
file:///home/peter/projects/evothings-studio/hyper/ui/hyper-ui.html
More
Connect
Examples
My Apps
New
Tools
BLE Scan
/home/peter/EvothingsStudio/MyApps/ble-scan/index.html
CODE
RUN
www/index.html
/home/peter/projects/lunar-lander-html5/www/index.html
CODE
RUN
ble-scan
File Edit View Go Bookmarks Help
Back
Forward
100%
Places
peter
EvothingsStudio
MyApps
ble-scan
Devices
466 GB Volu...
Nexus 5
Bookmarks
gc
genericwitti...
sftp
Computer
peter
Desktop
File System
Documents
Downloads
Music
Pictures
Videos
libs
ui
app.js
app-icon.png
evothings.json
index.html
6 items, Free space: 11,7 GB
ged In as Peter Svensson
Viewers: 1 Files: 0


JavaScript Workbench
file:///home/peter/projects/evothings-studio/hyper/ui/hyper-workbench.html
// Select a snippet of code, then click the "Eval" button.
// This will run the connected devices/browsers.
// Compute a value, the return value is shown in the
// result panel.
1 + 2
// Send log message from connected devices/browsers.
// Use this in your code for debugging.
hyper.log('Hello World')
// Replace the browser content with the given HTML.
document.body.innerHTML = '<h1>Hello World</h1>'
// Now try this.
counter = 1
display = function() {
document.body.innerHTML =
'<h1>Counter: ' + counter + '</h1>'
++counter
}
timer = setInterval(function() { display() }, 500)
// Then stop the timer.
clearInterval(timer)
// Start the timer again, by evaluating the code above.
// Then change the value of counter, while the timer
// is running. Also change the function display while
// the timer is running.
// Have fun!
EVAL SELECTION
CLEAR EDITOR
RESTORE EDITOR
UNDO
REDO
CLEAR LOG
PAUSE LOG
NODE EVAL
// This is the message area where log messages and errors
// are displayed.
//
// Select one or more lines of code in the code area above.
// Then press "Eval Selection" to evaluate (run the code)
// on connected devices.
//
// "Node Eval" is special, it evaluates selected code in
// node-webkit, the platform on which this software is running.
// With this button, you can interactively develop and test
// node.js servers and functions in node-webkit. This is a
// very powerful feature.

Even More Demo!

IoT Hacking!!

http://www.meetup.com/IoTStockholm/events/228342296/