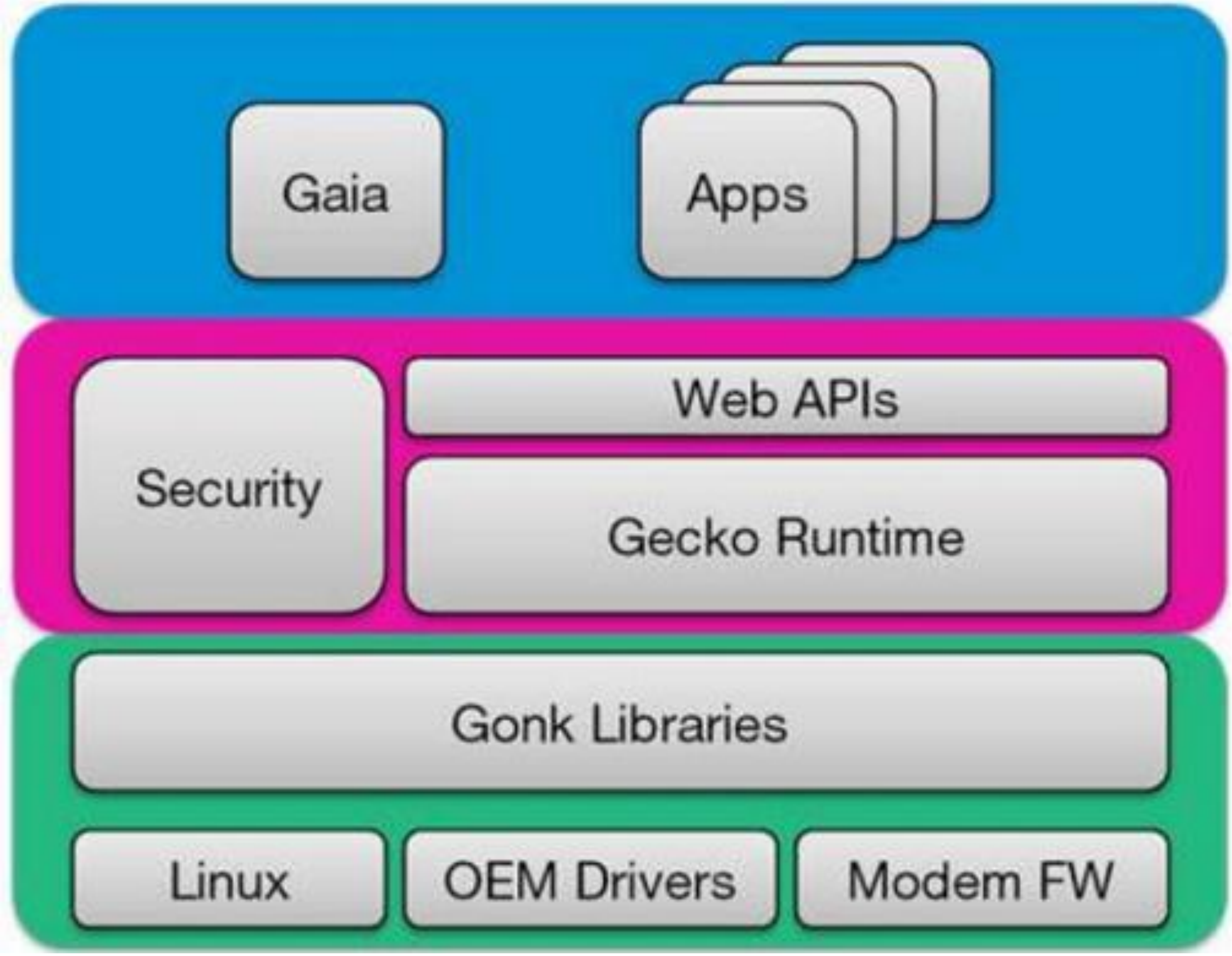
Introduction To Firefox OS

FxOS Architecture & Its Apps

Firefox OS Architecture



Gaia

Apps

Security

Web APIs

Gecko Runtime

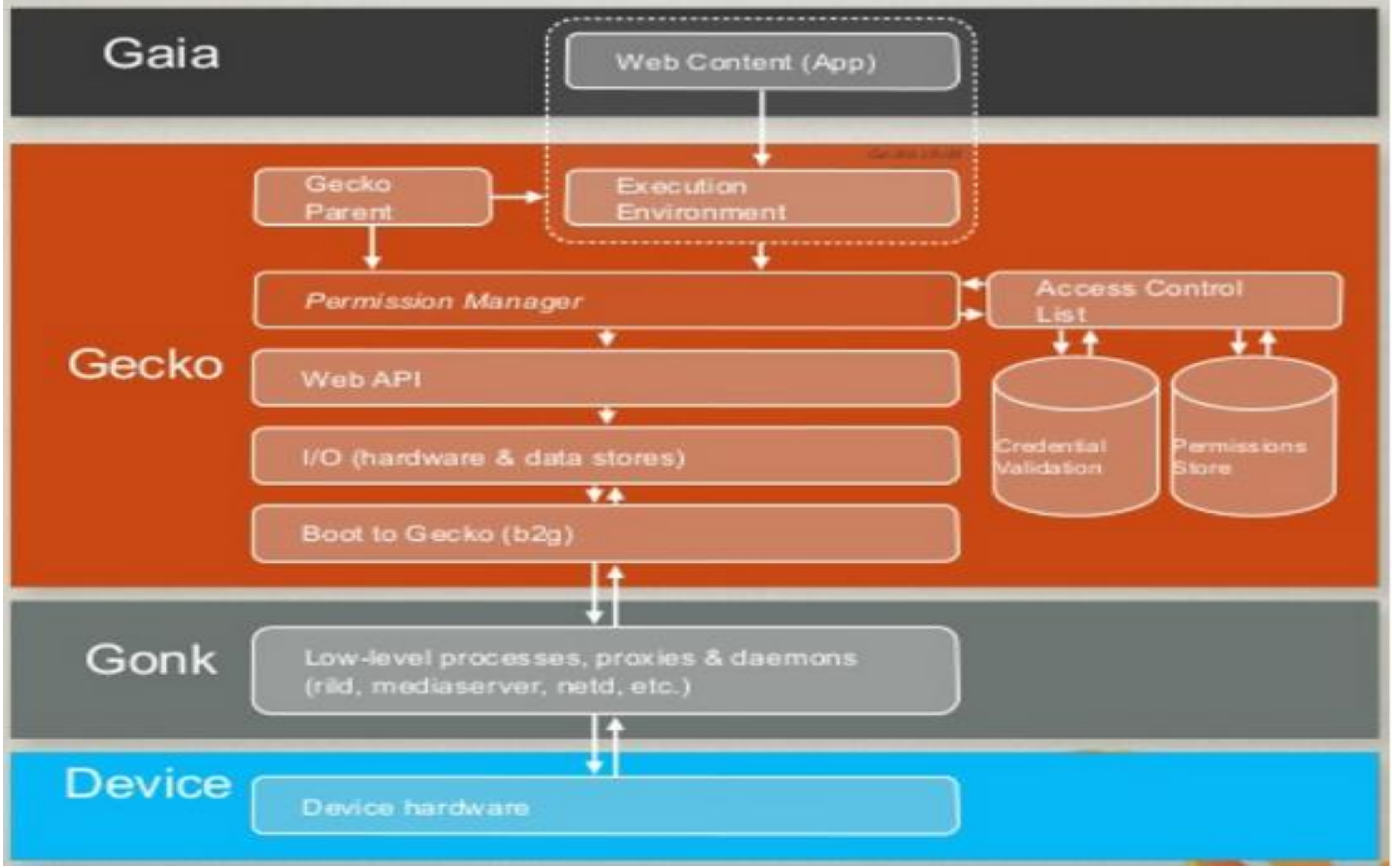
Gonk Libraries

Linux

OEM Drivers

Modem FW

Firefox OS Architecture



Gaia

Web Content (App)

Gecko

Gecko Parent

Execution Environment

Permission Manager

Access Control List

Web API

Credential Validation

Permissions Store

I/O (hardware & data stores)

Boot to Gecko (b2g)

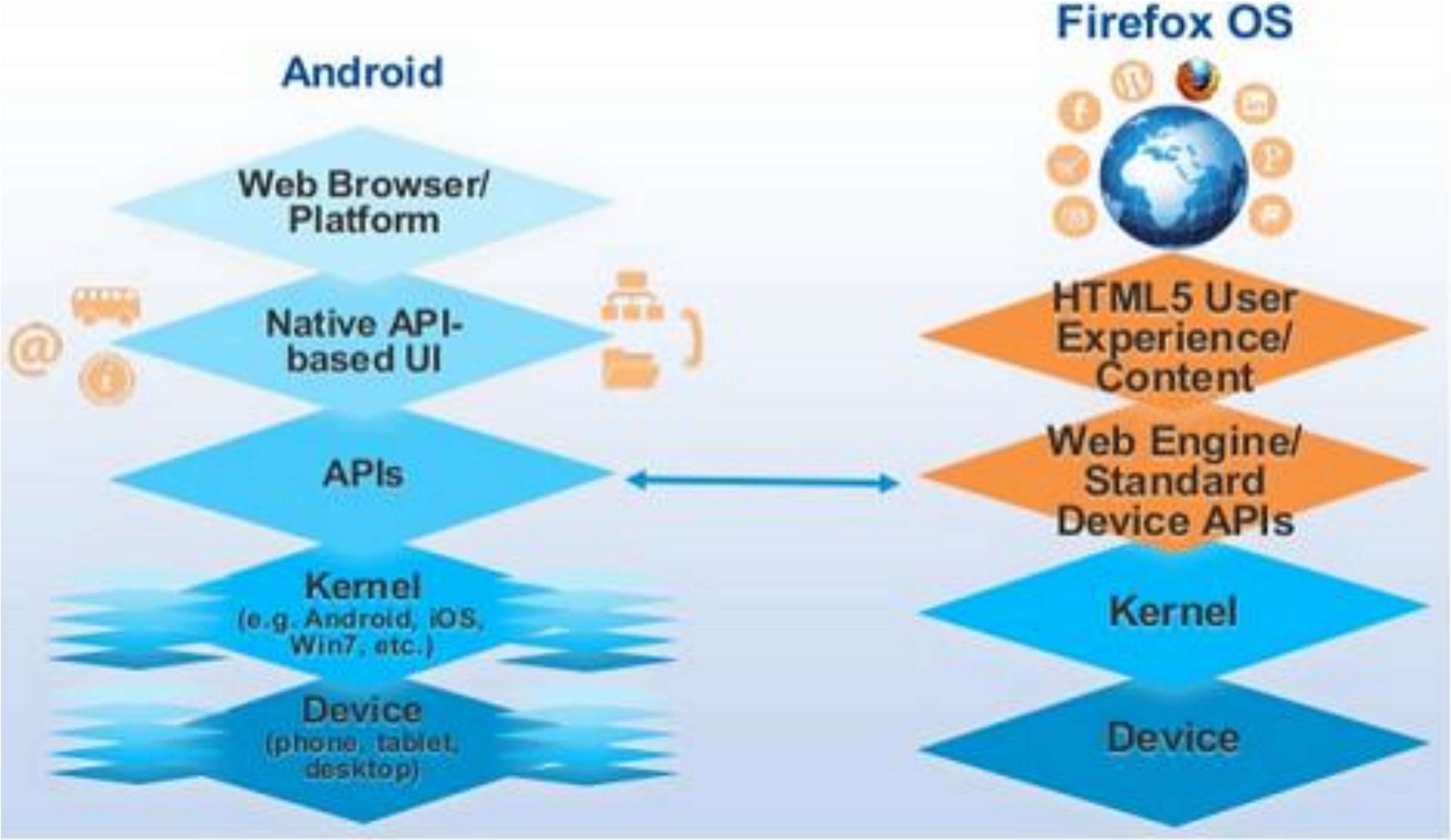
Gonk

Low-level processes, proxies & daemons (rild, mediaserver, netd, etc.)

Device

Device hardware

Firefox OS v.s Android, etc



Firefox OS App Development

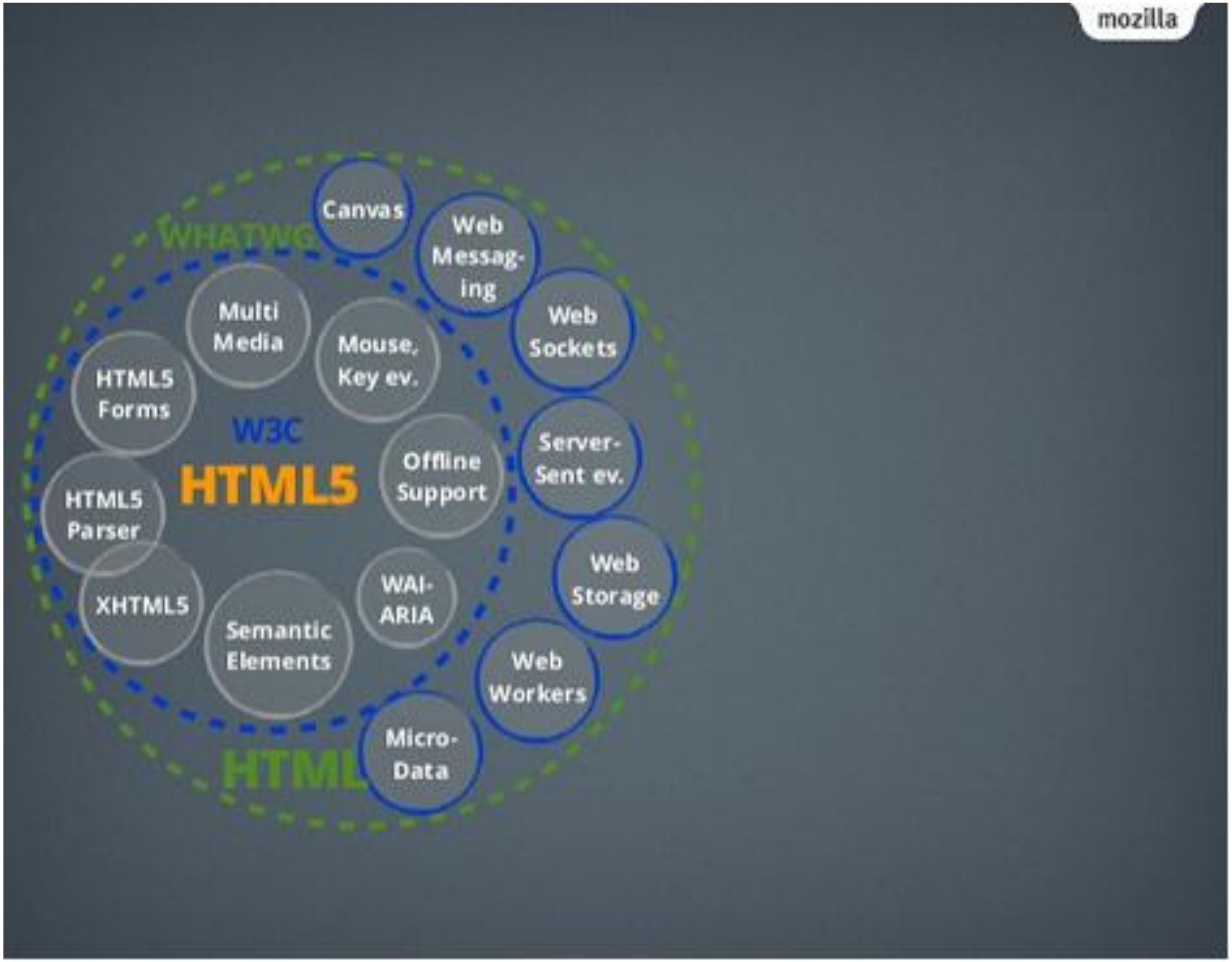
- Based on Open Web Standard:
  - HTML 5
    - Layout, Canvas
    - Web Socket, WebGL, WebCL, …
    - + Device (Sensors) API, File API
    - SVG, …
  - CSS 3
    - Animation, Media Query (Responsiveness),
  - JavaScript
    - XHR, Local Storage, …

Firefox OS SDK?!

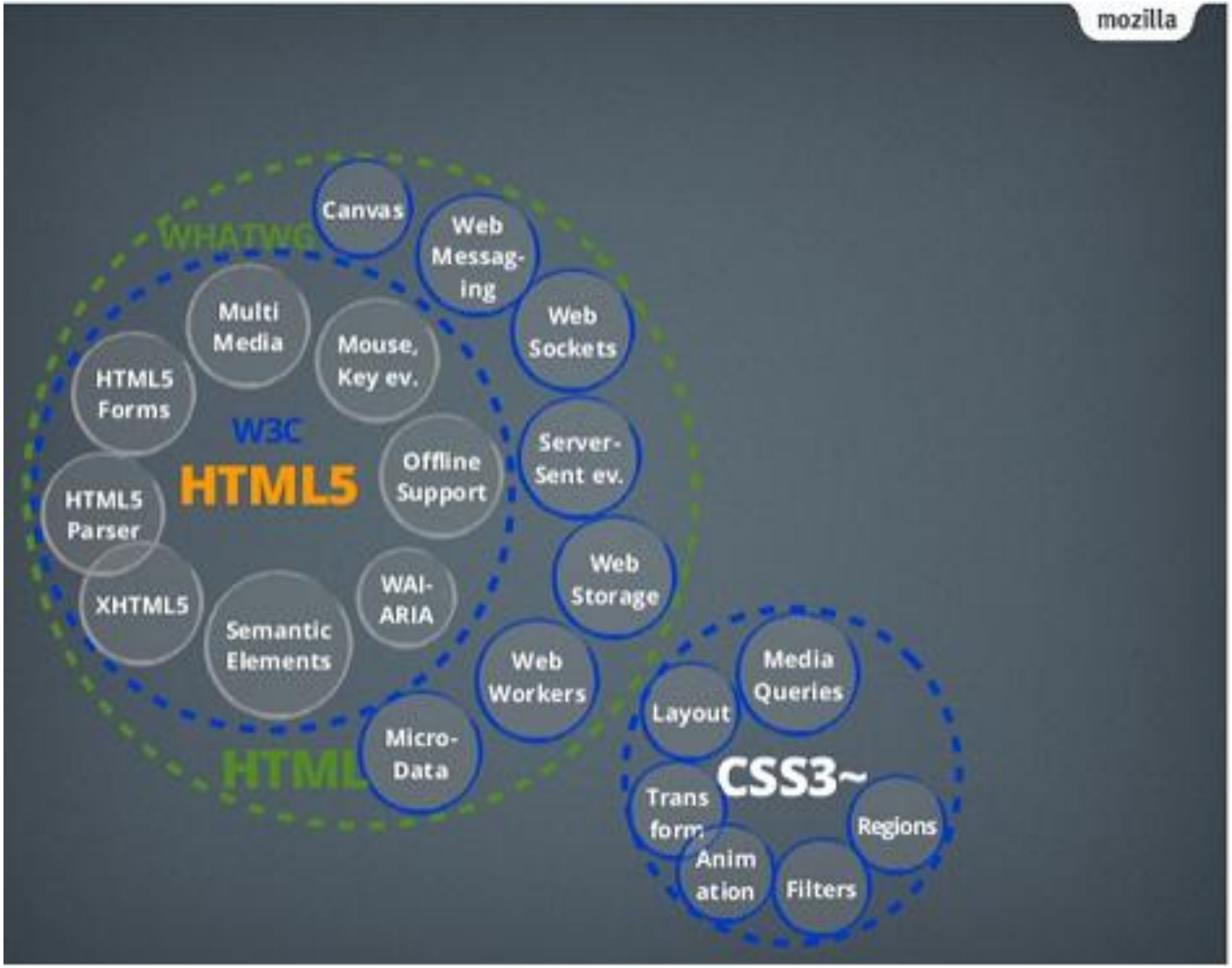
- It Does not require any Special SDK
- Uses Open Web Standard

- Developer can either
  - Write its Web App using plain HTML5/CSS/JS, or
  - Adopt Any General Client Side Web Framework

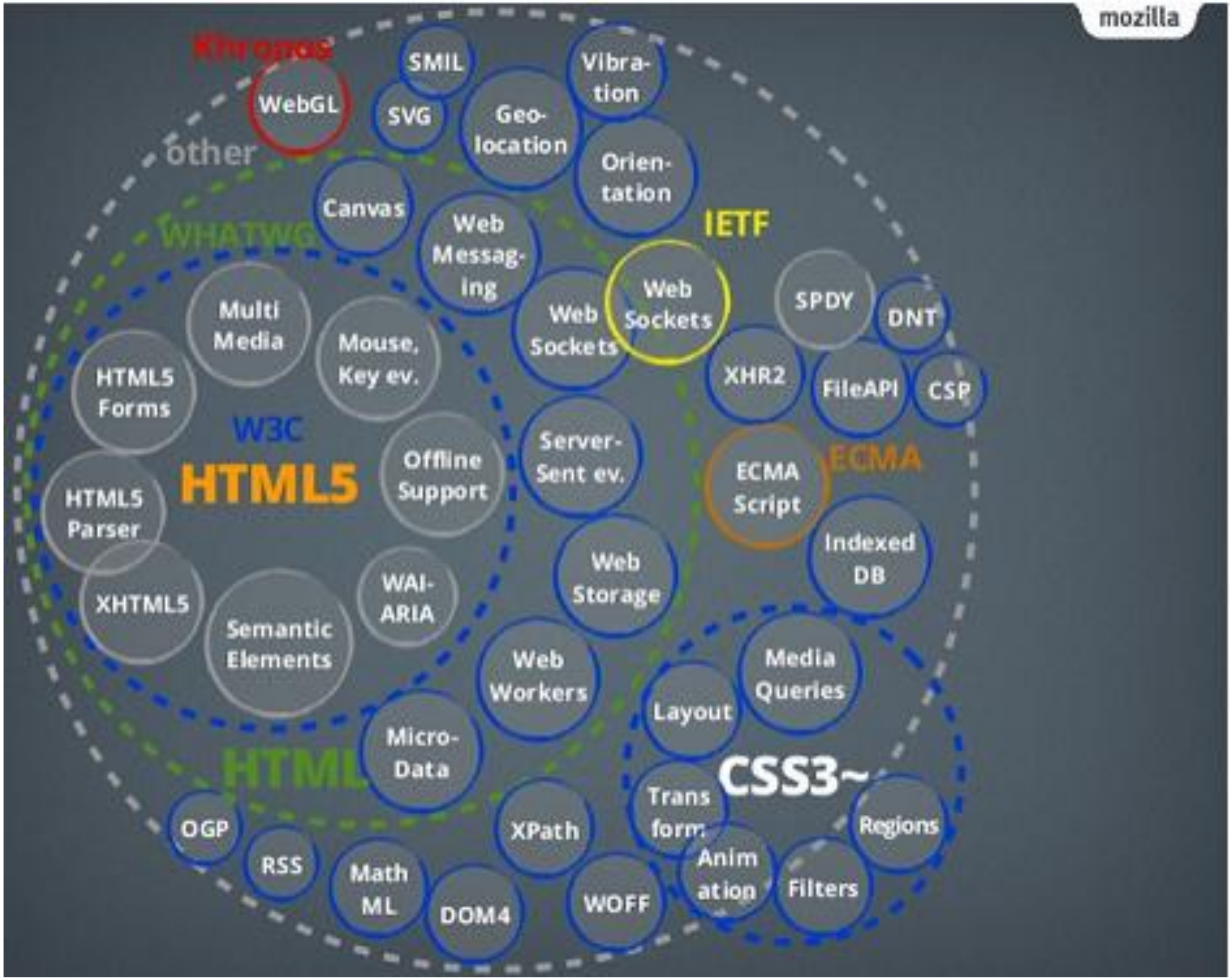
Web API is Already Well Developed HTML



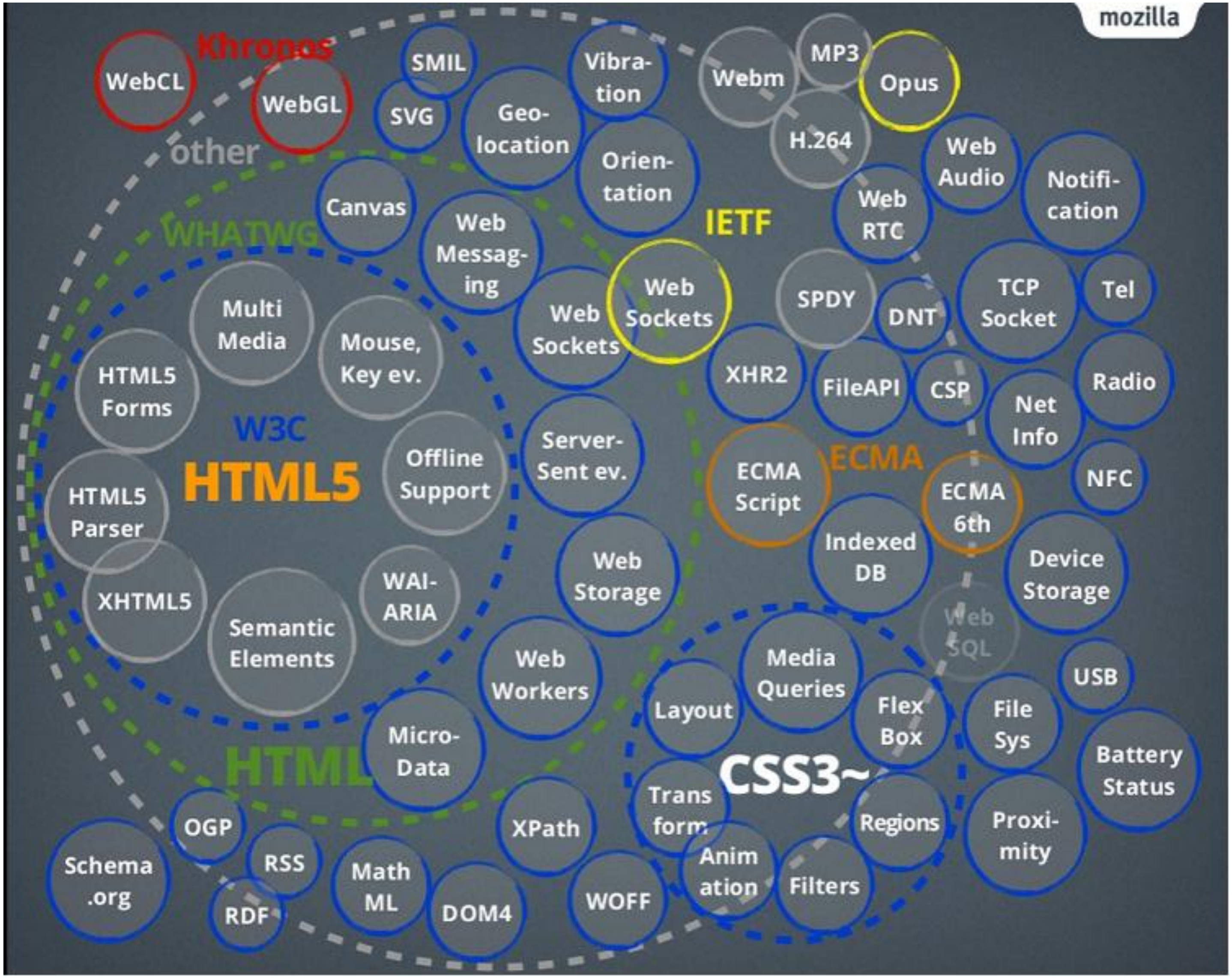
Web API is Already Well Developed CSS



Web API is Already Well Developed JavaScript (ECMAScript)

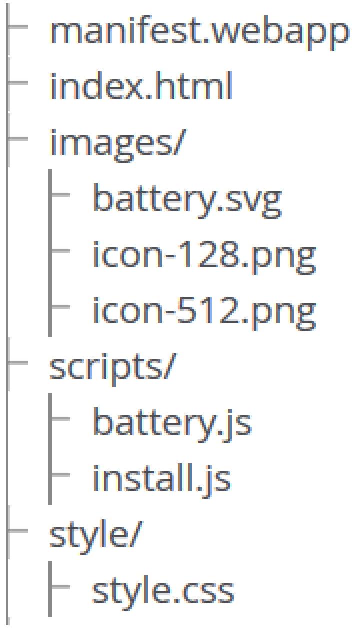


Web API is Already Well Developed Much More …



Firefox OS Apps

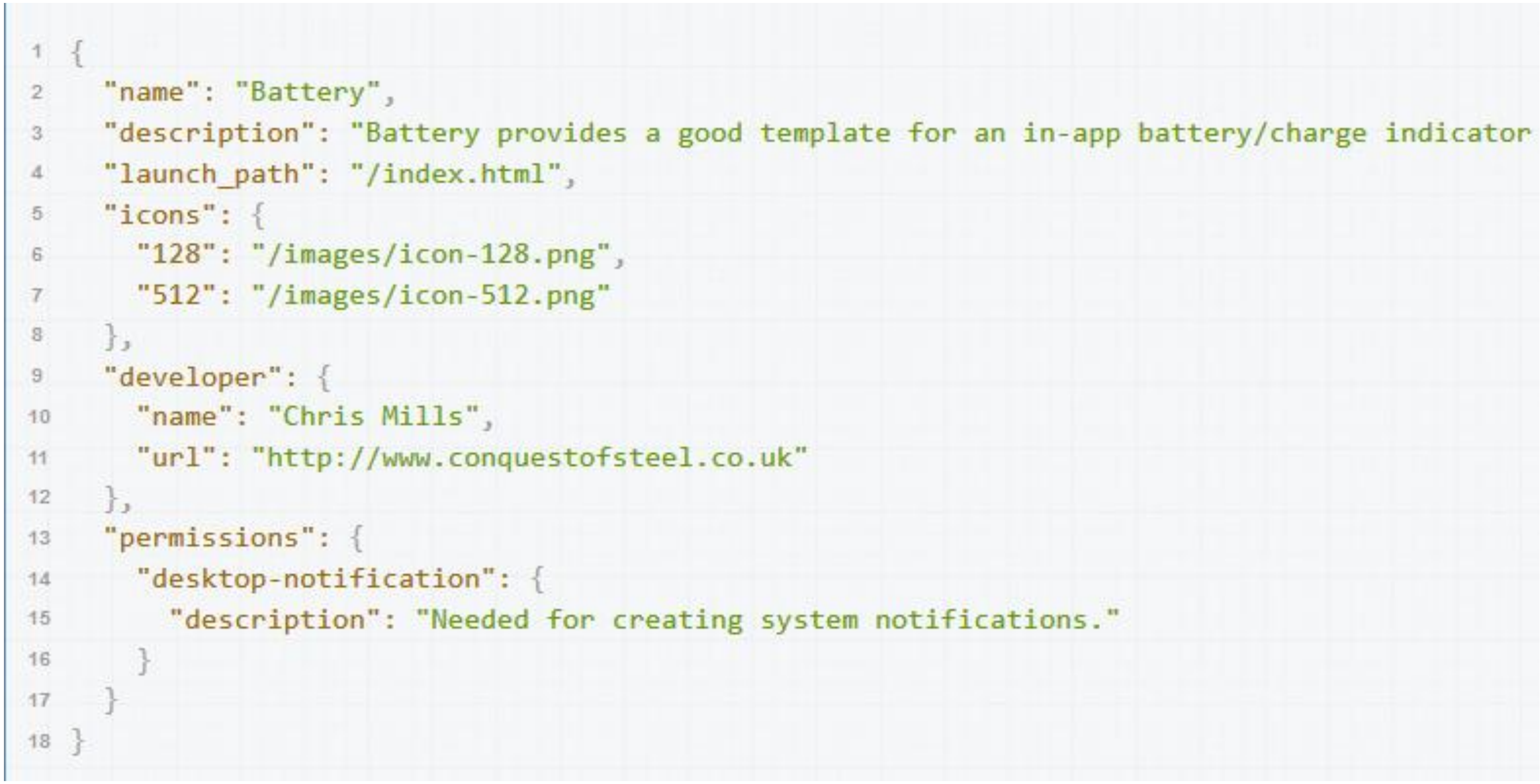
- There are three types of Apps
  - Hosted Apps
  - Packaged Apps
  - Privileged Apps
- Sample App Anatomy



```
├ manifest.webapp
├ index.html
├ images/
│ ├ battery.svg
│ ├ icon-128.png
│ ├ icon-512.png
├ scripts/
│ ├ battery.js
│ ├ install.js
├ style/
│ ├ style.css
```

Adding a Manifest File to App

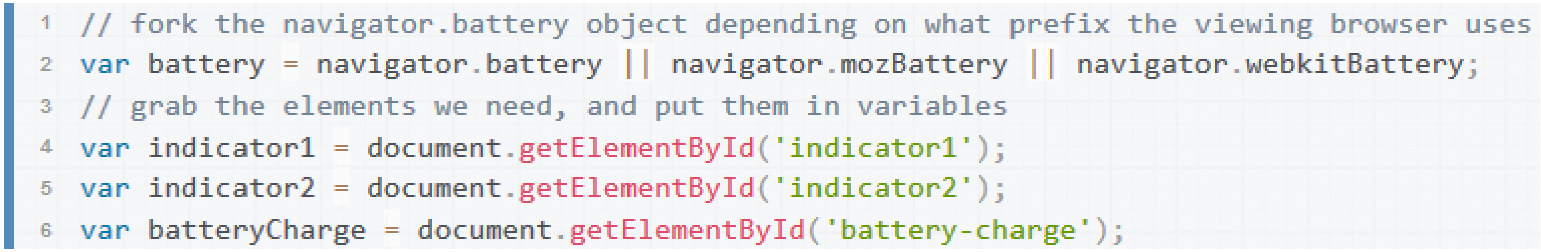
- It is written in JSON (JavaScript Object Notation) format
- If available Lets provides app installation functionality



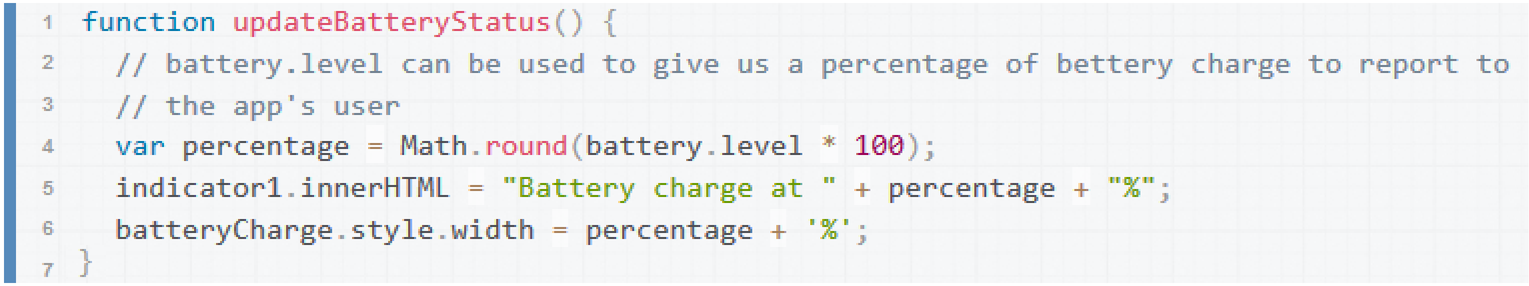
```
{
  "name": "Battery",
  "description": "Battery provides a good template for an in-app battery/charge indicator
  "launch_path": "/index.html",
  "icons": {
    "128": "/images/icon-128.png",
    "512": "/images/icon-512.png"
  },
  "developer": {
    "name": "Chris Mills",
    "url": "http://www.conquestofsteel.co.uk"
  },
  "permissions": {
    "desktop-notification": {
      "description": "Needed for creating system notifications."
    }
  }
}
```

Example Device Access Using JS

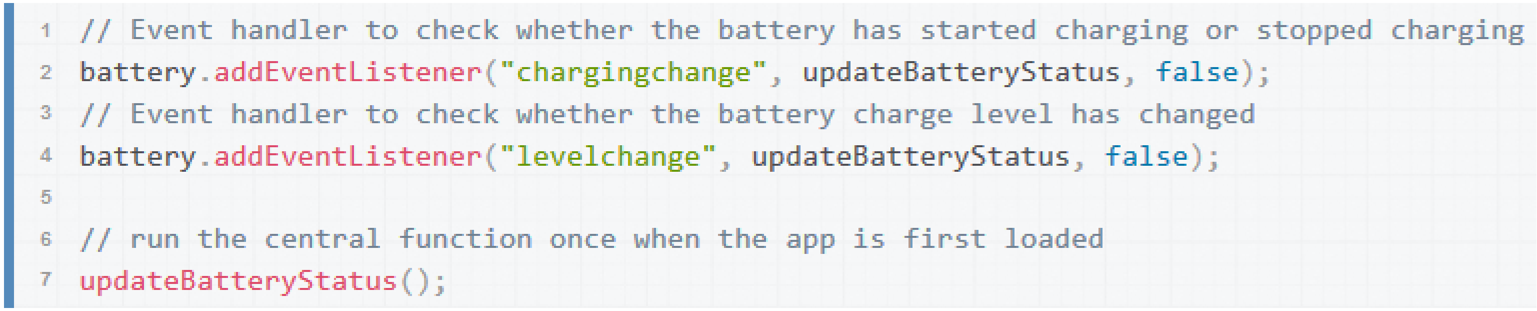
- Showing battery status in an HTML element



```
// fork the navigator.battery object depending on what prefix the viewing browser uses
var battery = navigator.battery || navigator.mozBattery || navigator.webkitBattery;
// grab the elements we need, and put them in variables
var indicator1 = document.getElementById('indicator1');
var indicator2 = document.getElementById('indicator2');
var batteryCharge = document.getElementById('battery-charge');
```



```
function updateBatteryStatus() {
  // battery.level can be used to give us a percentage of bettery charge to report to
  // the app's user
  var percentage = Math.round(battery.level * 100);
  indicator1.innerHTML = "Battery charge at " + percentage + "%";
  batteryCharge.style.width = percentage + '%';
}
```



```
// Event handler to check whether the battery has started charging or stopped charging
battery.addEventListener("chargingchange", updateBatteryStatus, false);
// Event handler to check whether the battery charge level has changed
battery.addEventListener("levelchange", updateBatteryStatus, false);

// run the central function once when the app is first loaded
updateBatteryStatus();
```