Another billion

WEB TECHNOLOGIES FOR NEXT GENERATION OF CONNECTED DEVICES

Overview

We have a new generation of low power low performance connected devices.

It is called Internet of Things

Can we learn from mobiles?

How can we develop apps for them?

Can we reuse knowledge and tools?

Can we learn anything from mobile industry?

Background

C++ developer, embedded systems and mobile OS development

Web – Mobiles – Next Gen

We have made a long journey with mobile internet and web.

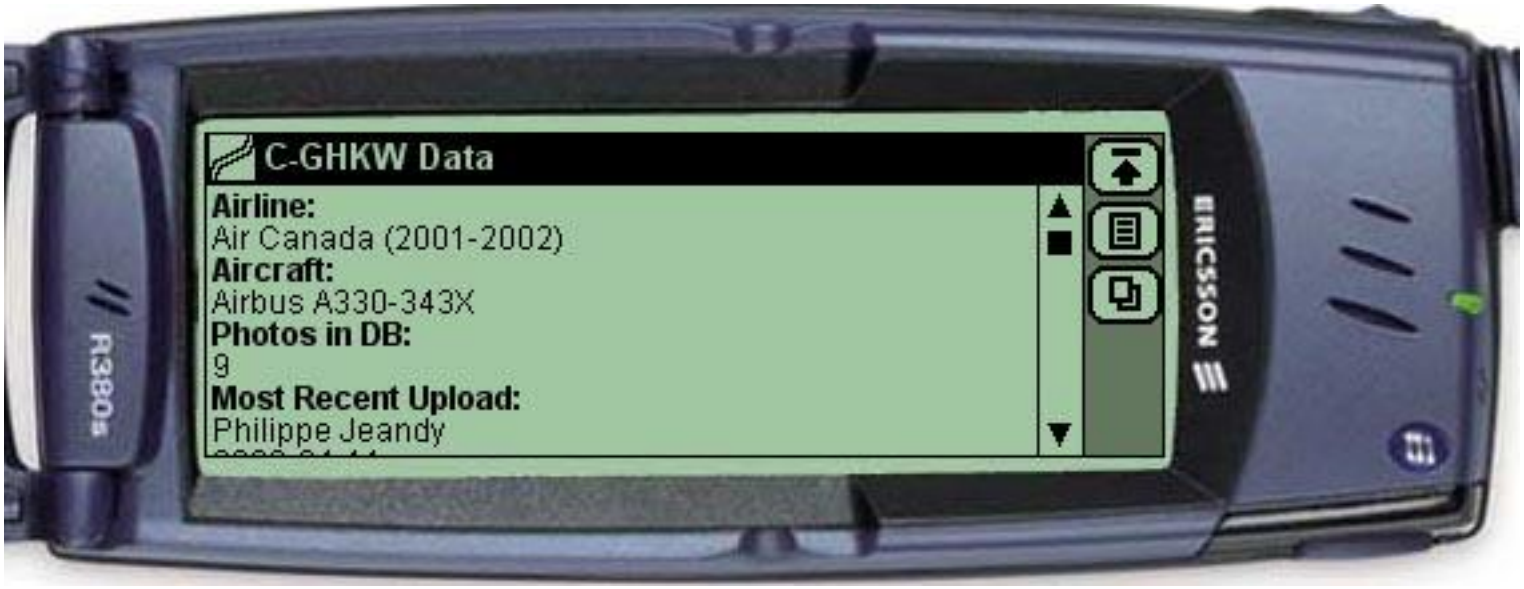
It took many steps to get where we are today – working mobile web

Smartphones

Smartphones are running version of desktop browsers.

Will we repeat the same mistakes with next generation devices?

This was going to be future of mobile web



History – Wireless Application Protocol

Driven by mobile operators driven

Mobile equivalent of HTML + HTTP

Huge investment by entire industry for technology nobody really used

History – WAC 2.0

Wholesale Applications Community – platform for mobile software developers

History – WAC 2.0

- Driven by mobile operators
- Based on HTML, JavaScript, and CSS

Almost like HTML5, but not quite. By the time it was ready, HTML5 was picking up.

Mobile Browsers

Initially driven (and limited) by hardware and software (mobile phones).

This usually caused more issues than in solved.

Versions of desktop browsers are available on mobiles now.

Next Gen - IoT

Internet of Things is emerging technology – smart sensors, wearable tech, connected house, connected car... You name it

Reasonably established in some categories like wearable tech and smart watches.

Internet of things by numbers

Range of connected devices and already counting in billions

- Many trivial devices – smart sensors
- More complex devices with dedicated OS – Google **Project Brillo**
- Devices supporting web technologies – perhaps smallest group of IoT devices, but given the scope of market still significant

Next Gen - IoT

Why web technologies ?

Existing dev tools and existing developers

Raspberry Pi

Epiphany -Fast, Modern Browser for the Raspberry Pi

Much-improved HTML5
JavaScript JIT
Hardware-accelerated video decoding

Samsung Gear 2

Runs WebKit

IoT issues

As with any emerging technology, there are issues to be sorted out

Ease of development

Security

Privacy

Less powerful hardware

Lets use WebKit

Using WebKit as development environment for IoT has many advantages

- Existing development tools
- Developers
- Well understood Security implementation

Security

We get a lot from WebKit technologies by default

https:// - based on TLS and SSL using OpenSSL libraries

Privacy

Privacy in WebKit – for embedded devices, we can disable or configure some features

- Cookies
- web storage
- CORS

Running Out Of Memory

If you run with memory budget of 200MB you are likely to hit OOM

- The Web App really needs that much memory – not much to do apart from redesigning

Running Out Of Memory

Web App is not optimised

Optimise the App itself and improve WebKit OOM handling and Garbage Collection algorithm.

Generic optimisation

```
for(var i=0; i<10; i++){
    var obj = {key:'val'};
    console.log(obj);
}
```

OOM stress test

Initialise variables

```
var numberOfImages = 0;

var memoryDiv = document.getElementById('memory');


//get references to the 6 static DOM elements up
front.

var imageDivs = [];

for (var i = 0; i < 6; i++) {

    imageDivs.push(document.getElementById('image' +
i));

}
```

WebKit OOM & GC

```
function interval() {
    var currentImageDiv = imageDivs[numberOfImages % 6];
    currentImageDiv.innerHTML =
    '<img src="testimage.png?cachebuster=' + numberOfImages + '" />';

    numberOfImages++;
}
```

Existing GC issue in WebKit

In certain situations clearing the WebKit's MemoryCache doesn't releases all CachedResource to system.

https://bugs.webkit.org/show_bug.cgi?id=111094

Garbage Collect to release the references of CachedResource

Out Of Memory in WebKit

Existing functionality in WebKit is proven for desktop.

Managing memory budget involves cache, font storage, Garbage Collection

Full Memory Pressure Handler is needed in WebKit.

Going forward

Web developers and designers should consider memory optimisation on Next Gen devices.

Better memory management in WebKit is needed.

Specific embedded WebKit port for IoT devices is needed.

Conclusion

Performance as such is not generally an issue

Hardware acceleration is not an issue

Memory usage is the issue