



Web Runtimes – evolving beyond the browser

David Pollington
Vodafone Group R&D
Jan 2008

Open Ajax Alliance



Bringing the Web experience to the mobile phone

Early attempts at bringing the Internet experience to the mobile phone relied on new microbrowser technologies such as WAP, NTTDoCoMo's i-mode and Openwave's HDML platform instead of using the desktop internet technologies available at the time

Whilst these introduced the mobile phone to the concept of accessing connected content, the ultimate goal was always to be able to deliver the full Web experience

This is now becoming a reality with the introduction of advanced mobile browsers from the likes of S60, Opera, Access, Apple and others, all of which are bringing the full Web 2.0 experience to mobile phones:

- Preservation of the original page layout
- Pointer navigation
- Easy navigation of Web pages through page miniatures or sophisticated zooming
- Visual History (an easy-to-use back function, showing miniature views of previous pages)
- Text Search (which works as you type, taking you directly to the interesting part of the page)
- Landscape mode
- Web Feeds (ability to subscribe to RSS feeds from Web sites, blogs, and news services)
- Etc.



Evolving Web services beyond the browser

But it doesn't stop there; the mobile browser is evolving to become as much an application platform as an information delivery tool by enabling Web services to be increasingly executed in a client mode on the handset through supporting techniques such as XHR

Such techniques allow Web services to be much more responsive and are particularly applicable in the mobile space which typically suffers from a higher round-trip latency than the wired Internet

Enabling a client-based approach also allows much of the Web service graphical assets and layout information to be cached on the handset speeding service start-up, enabling offline access (e.g., in flight mode) and minimising the amount of data that needs to be fetched over the mobile network (which in itself reduces the impact on battery life)

Evolve the browser as a new Web Runtime for hosting mobile apps and services

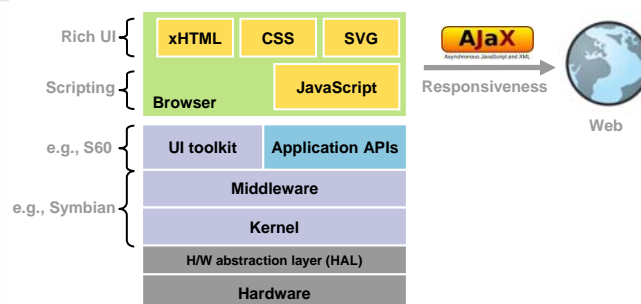


What do we mean by a Web runtime?

In simple terms, a Web Runtime combines the core technologies of a browser:

- *xHTML, CSS and SVG to provide a Rich UI layer*
- *JavaScript to provide simple scripting for the application logic*
- *XHR Method to provide asynchronous update to improve responsiveness*

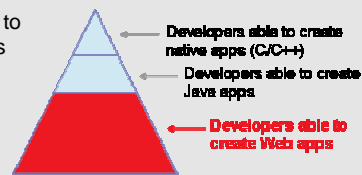
...with an AMS (Application Management System) for app installation, management and revocation



Web Runtime benefits

By leveraging Web technologies, a Web Runtime approach offers a number of additional benefits:

- Ability to match the rich and responsive nature of native and Java applications
- Lowers the entry barriers for developers hence growing a much larger base of developers able to innovate on the mobile phone
- Reduces development time to a matter of days rather than months and by doing so enables new ideas to be easily prototyped and deployed
- Decouples application development from any differences in the underlying platform hence reducing the requirements for porting the application across a number of different devices and in doing so speeding deployment of new services across the portfolio of devices
- Allows dynamic update of the application to changes in the environment & user needs
- Enables an evolution from a monolithic approach for developing services to one which is more component based



GROUP R&D



Enhancing the Web Runtime with device APIs

A Web Runtime is fine for simple Web apps such as fetching a weather forecast or displaying a map but for the Web Runtime to be truly useful it needs to be able to access the services and user data on the underlying handset

Opening up device APIs to the Web Runtime enables Web apps to start mixing up both local data and information from the Web to generate a range of interesting 'mashup' services or simply to make services more personal

Device APIs:

- File system access to pictures, music and video clips
- PIM data (contacts, calendar)
- Messaging (SMS, MMS, email)
- Call handling
- System data (battery, coverage, roaming, location)
- Camera (get new picture)

Example use cases:

- Displaying a map based on a contact's address
- Fetching a weather forecast based on location
- Setting up a call by clicking on a Web advert etc.



GROUP R&D

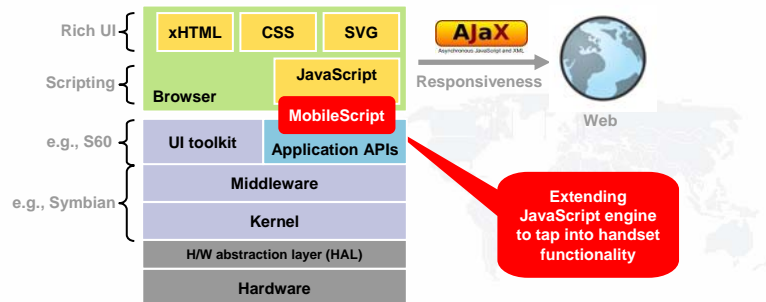


Enhancing the Web Runtime with device APIs

An example is MobileScript developed by VF Group R&D which extends the scripting capabilities of a Web Runtime to bind to the APIs provided by the underlying platform; by modelling a mobile handset by means of a Document Object Model it provides developers with:

- access to Device properties
- ability to invoke services and Handle events

GROUP R&D



7 Web Runtimes
Jan 2008



Code in action

Simple app that provides a basic Web page that runs in the browser and enables the user to make calls; also captures incoming call events and automatically rejects them and returns an SMS 'Sorry, busy at the moment'... and all in only 6 lines of code

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE = "JavaScript">
```

```
function newcall(){
var caller = new OutgoingCallRegistry();
caller.NewCall(destAddressTB.value);
```

Starts a new call any time the call button is clicked

Puts down incoming calls and sends the caller an SMS

```
function incomingcallhandler(){
Phone.IncomingCallRegistry.IncomingCall.PutDownAndReply("Sorry, busy at the moment");}
```

```
document.addeventlistener('incomingcall', incomingcallhandler, false);
```

```
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="Form1">
<INPUT type="TEXT" name="destAddressTB" size="15" />
<INPUT TYPE="BUTTON" NAME="newcallB" VALUE="Call" onClick="newcall(), />
</FORM>
</BODY>
</HTML>
```

The web page subscribes to the incoming call event



GROUP R&D

8 Web Runtimes
Jan 2008



Security risks

Naturally, opening up device APIs to a Web Runtime will introduce a number of security risks and potentially expose the user to a range of malicious attacks:

- Automatically invoking premium rated services or calls...
- Bookmark modification to enable phishing
- Theft or corruption of personal data
- Spam injection
- Modification of settings etc.



Such attacks have an obvious negative impact on users, potentially discouraging them from using their phones in a web environment. However security issues could also impact operators due to service disruption, loss of revenues and increased customer care costs

Defining an appropriate security framework will therefore be of utmost importance, although it is recognised that these security requirements will need to be balanced to ensure that the resulting framework does not choke developer creativity and innovation



Making a case for standardisation

Web Runtimes have the opportunity of enabling rapid development and deployment of new applications and services on mobile handsets and in doing so fostering an environment for innovation

However, there is a risk that much of this potential will be squandered if Web Runtimes are left to evolve on a per company basis and result in a fragmented marketplace and an execution environment which is insecure and exposes the user to unacceptable levels of malicious attack

A number of core aspects of Web Runtimes therefore need to be discussed, agreed upon and potentially standardised. At the very least this should include:

1. Packaging format for the distribution and installation of Web apps (already being developed within the W3C WAF WG)
2. Extended JavaScript API spec to provide a common method of accessing underlying device capabilities
3. Security framework and associated policies for managing the use of these APIs (incl. privacy of user data)



David Pollington

Vodafone GROUP R&D.UK
Vodafone House
The Connection
Newbury, Berkshire RG14 2FN



T + 44 1635 685504
david.pollington@vodafone.com

<http://www.vodafonebetavine.net/web/MobileScripting/>

<https://forge.vodafonebetavine.net/projects/mobilescript/>

