



MobiWebApp

Mobile Web Applications for Future Internet Services

Deliverable D4.1

**Standardization Roadmap
Year 1**

August 2011 - Version 1

Project	
Grant Agreement number	257800
Project acronym:	MobiWebApp
Project title:	Mobile Web Applications for Future Internet Services
Funding Scheme:	Coordination & Support Action
Date of latest version of Annex I against which the assessment will be made:	March 17, 2010
Document	
Deliverable number:	D4.1
Deliverable title	
Contractual Date of Delivery:	Project month 12
Actual Date of Delivery:	August 31, 2011
Editor (s):	Dominique Hazaël-Massieux
Author (s):	Dominique Hazaël-Massieux
Reviewer (s):	Dr. Marie-Claire Forgue, Dr. Philipp Hoschka
Participant(s):	ERCIM/W3C
Work package no.:	4
Work package title:	Standardization
Work package leader:	Dominique Hazaël-Massieux
Distribution:	PU
Version/Revision:	1
Draft/Final:	
Total number of pages (including cover):	37
Keywords:	Standards, Device APIs, Mobile Web, Web Applications,

Disclaimer

This document contains description of the MobiWebApp project work and findings.

The authors of this document have taken any available measure in order for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated in the creation and publication of this document hold any responsibility for actions that might occur as a result of using its content.

This publication has been produced with the assistance of the European Union. The content of this publication is the sole responsibility of the MobiWebApp consortium and can in no way be taken to reflect the views of the European Union.

The European Union is established in accordance with the Treaty on European Union (Maastricht). There are currently 27 Member States of the Union. It is based on the European Communities and the member states cooperation in the fields of Common Foreign and Security Policy and Justice and Home Affairs. The five main institutions of the European Union are the European Parliament, the Council of Ministers, the European Commission, the Court of Justice and the Court of Auditors. (<http://europa.eu.int/>)



MobiWebApp is a project funded in part by the European Union.

Table of Contents

Introduction.....	5
1 Graphics.....	7
2 Multimedia	10
3 Device adaptation.....	13
Forms	15
4 User interactions	17
5 Data storage.....	19
6 Personal Information Management	21
7 Sensors and hardware integration	23
8 Network	25
9 Communication	28
10 Packaging	30
11 Performance & Optimization	32
Recapitulation of planned standardization work in upcoming year.....	35

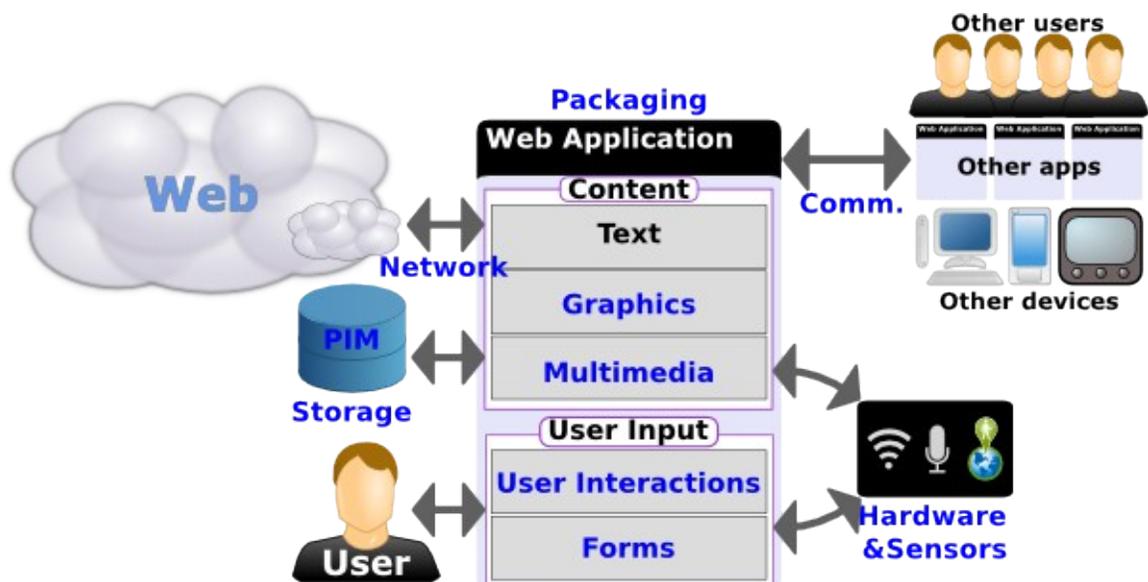
INTRODUCTION

Web technologies have become powerful enough that they are used to build full-featured applications; this has been true for many years in the desktop and laptop computer realm, but is increasingly so on mobile devices as well.

This document summarizes the various technologies developed in W3C that increase the power of Web applications, and how they apply more specifically to the mobile context.

The features that these technologies add to the Web platform are organized under the following categories:

- [Graphics](#)
- [Multimedia](#)
- [Device Adaptation](#)
- [Forms](#)
- [User interactions](#)
- [Data storage](#)
- [Personal Information Management](#)
- [Sensors and hardware integration](#)
- [Network](#)
- [Communication](#)
- [Packaging](#)
- [Performance & Optimization](#)



In each category, a table summarizes for each feature:

- which W3C specification defines the feature,
- which W3C group is responsible of the said specification,
- the stage of the specification in the W3C Recommendation track (see below),
- the estimated stability of the document, i.e. how widely the document is expected to change, as estimated by the author of this report, with three levels: low (the document is mostly stable), medium (some parts are stable, others are expected to change significantly), high (the document is expected to evolve significantly),

- some rough qualitative indication on availability of implementations on mobile devices,
- a link to the latest editors draft of the document,
- a link to the test suite for the said feature.

As a reminder, W3C creates Web standards by progressing documents through its [Recommendation track](#), with the following stages:

- “Editors drafts” represent the current view of the editors of the specification but have no standing in terms of standardization.
- “Working Drafts” are early milestones of the Working Group progress.
- “Last Call Working Drafts” signal that the Working Group has determined that the specification fulfils its requirements and all the known issues have been resolved, and thus requests feedback from the larger community.
- “Candidate Recommendations” trigger a call for implementations where implementers are invited to implement the specification and send feedback; Working Groups are expected to show the specification gets implemented by running test suites they have developed.
- “Proposed Recommendations” manifests that the group has gathered sufficient implementation experience, and triggers the final review by W3C Members
- “W3C Recommendations” are stable and completed Web standards; these documents only get updated rarely, through the Edited Recommendation process, as a results from errata collected by Working Groups.

Prior to starting standardization, a Working Group needs to be chartered, based on input from W3C Members, often through the organization of a [workshop](#), after the reception of a [W3C Member Submission](#). W3C has recently set up [Community Groups](#), a new mechanism that allows anyone to do experimental work within the W3C infrastructure, under IPR rules that are compatible to transition the work to the W3C standardization process.

1 GRAPHICS

Ongoing work

[SVG](#), Scalable Vector Graphics, provides an XML-based markup language to describe two-dimensional vector graphics. Since these graphics are described as a set of geometric shapes, they can be zoomed at the user request, which makes them well-suited to create graphics on mobile devices where screen space is limited. They can also be easily animated, enabling the creation of very advanced and slick user interfaces.

The integration of SVG in HTML5 opens up new possibilities, for instance applying advanced graphic filters (through SVG filters) to multimedia content, including videos.

In complement to the declarative approach provided by SVG, the `<canvas>` element added in HTML5 enables a [2D programmatic API](#) that is well-suited for processing graphics in a less memory intensive way. That API not only allows to render graphics, but can also be used to make image processing and analysis.

Both SVG and HTML can be styled using [CSS](#) (Cascading Style Sheets); in particular, CSS3 (the third level of the specification) is built as a collection of specifications set to offer a large number of new features that make it simple to create graphical effects, such as rounded corners, complex background images, shadow effects ([CSS Backgrounds and Borders](#)), rotated content ([CSS 2D Transforms](#)), animations ([CSS Animations](#), [CSS Transitions](#)), and even 3D effects ([CSS 3D Transforms](#)).

Animations can be resource intensive — the possibility offered by the [Timing control for script-based animations API](#) to manage the rate of updates to animations can help keep them under control.

Fonts play also an important role in building appealing graphical interfaces, but mobile devices are in general distributed with only a limited set of fonts. [WOFF](#) (*Web Open Font Format*) addresses that limitation by making it easy to use fonts that are automatically downloaded through style sheets, while keeping the size of the downloaded fonts limited to what is actually needed to render the interface.

NB: work on defining a [3D graphic API called WebGL](#) has started outside of W3C, as part of the [Khronos Group](#); this API has been built to be compatible with [OpenGL ES](#), i.e. for embedded systems, and is intended to work on mobile devices.

Expected progress and new work in upcoming year

Work on a new version of SVG (SVG 2.0) is expected to start, with the goal of providing better integration with other Web technologies (e.g. HTML5 and CSS3), making it possible to use some of the most advanced features of SVG directly from regular Web pages. High priority features include Filters, Web Animation, Compositing, Integration, Gradients, Color Management, and 2D Transforms. The increase of capabilities in mobile devices will make it unnecessary to build a separate version targeted specially at these devices.

Work on the canvas API is expected to bring the specification to Candidate Recommendation in the coming year.

The downloadable font specification is expected to reach Recommendation status in the first quarter of 2012.

The [Declarative 3D for the Web Architecture Community Group](#) was recently launched and will explore the opportunity of developing a declarative approach for 3D graphics, completing what WebGL provides as an API.

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
2D Vector Graphics	SVG Tiny 1.2	SVG Working Group	 Standard	Finished	New version of SVG (SVG 2.0) in preparation	 Widely deployed (iOS, BlackBerry, WebKit on Nokia, webOS, Opera, Firefox, announced for Android, announced for Windows Phone)	High coverage
2D Programmatic API	HTML Canvas 2D Context	HTML Working Group	 Last Call Working Draft	Mostly stable	Updated regularly	 Widely deployed (iOS, BlackBerry, Android, webOS, Opera, Firefox, announced for Windows Phone)	Good coverage
Rounded Corners			 Candidate Recommendation			 Deployed as an extension in many mobile browsers	
Complex background images	CSS Backgrounds and Borders	CSS Working Group		Mostly finished	Updated regularly	 Limited	None
Box shadow effects						 Limited	

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
CSS 2D Transforms	CSS 2D Transforms Module Level 3		 Working Draft	Stabilizing	Last update May 2010	 Limited	None
3D Effects	CSS 3D Transforms Module Level 3		 Working Draft	First draft	Last update Dec 2010	 Very limited	None
Animations	CSS Animations Module Level 3		 Working Draft	First draft	Updated regularly	 Limited	None
	CSS Transitions Module Level 3		 Working Draft	Early draft	Last update Aug 2010	 Limited	None
	Timing control for script-based animations API	Web Performance Working Group	 Working Draft	Early	Regularly updated	 None	None
Downloadable fonts	WOFF File Format 1.0	WebFonts Working Group	 Candidate Recommendation	Mostly stable	Last update Aug 2011	 Good deployment	Good coverage

2 MULTIMEDIA

Ongoing Work

HTML5 adds two tags that dramatically improve the integration of multimedia content on the Web: the [<video>](#) and [<audio>](#) tags. Respectively, these tags allow to embed video and audio content, and make it possible for Web developers to interact much more freely with that content than they would through plug-ins. They make multimedia content first-class citizens of the Web, the same way images have been for the past 15 years.

While these tags allow to play multimedia content, the [HTML Media Capture](#) and the [Media Capture API](#) define **mechanisms to capture and record multimedia content** using attached camera and microphones, a very common feature on mobile devices. The newly chartered [Web Real-Time Communications Working Group](#) will also provide an API to directly manipulate **streams from camera and microphones**.

Beyond recording, two additional APIs add multimedia manipulation capabilities to the Web platform. We have already mentioned the [Canvas 2D Context](#) API: it enables modifying images, which in turn opens up the possibility of **video editing**. In a similar vein, a [W3C Incubator Group](#) has been working on an **Audio API** ([Mozilla's proposal draft](#)) that makes it possible to modify audio content, as well as analyse and synthesize sounds — this work serves as a basis to the newly chartered [Audio Working Group](#).

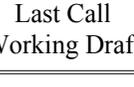
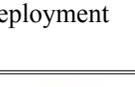
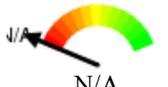
Finally, the new charter of the Device APIs Working Group includes an API for reading the current audio volume of a device, allowing to adapt the type of interactions with the user depending on that setting.

The combination of all these features mark the starting point of the Web as a comprehensive platform for multimedia, both for consuming and producing. The rising interest around bridging the Web and TV worlds (manifested through the [W3C Web and TV Interest Group](#)) should strengthen that trend in the coming months. Mobile devices are expected to take a growing role in many users TV experience, providing a “second screen” experience, where users can find more information on or interact with a TV program they're watching via their mobile devices.

Expected progress and new work in upcoming year

The Web Real Time Communications Working Group is expected to publish the first drafts of its APIs to access and handle audio/video streams in the last quarter of 2011.

Likewise, the first drafts of the API to analyse and synthesize Audio data should be published before the end of 2011.

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Video playback	HTML5 video element	HTML Working Group	 Last Call Working Draft	Stabilizing	Updated regularly	 Good deployment	Just started
Audio playback	HTML5 audio element		 Last Call Working Draft			 Good deployment	Barely started
Capturing audio/video	HTML Media Capture	Device APIs Working Group	 Working Draft	Early draft	Last update Apr 2011	 Very limited	None
	The Media Capture API		 Working Draft	Early Draft	Last update Dec 2010	 None	None
	API to get Video/Audio streams from camera/mike	Web Real-Time Communications Working Group	 N/A	None	getUserMedia proposal from WhatWG	 A few experimental ones	None
Image & Video analysis, modification	HTML Canvas 2D Context	HTML Working Group	 Last Call Working Draft	Mostly stable	Updated regularly	 Widely deployed (iOS, BlackBerry, Android, webOS, Opera, Firefox, announced for Windows Phone)	Good coverage
Audio analysis, modification	None	Audio Working Group	 N/A	Not started	Mozilla Data Audio API , Google Web Audio API	 None	None

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Audio volume reading	N/A	Device APIs	 N/A	Not started	N/A	 None	None

3 DEVICE ADAPTATION

Ongoing Work

Mobile devices not only differ widely from traditional computers, but they also have a lot of variations among themselves, in term of screen size, resolution, type of keyboard, media recording capabilities, etc.

The [Device Description Repository API](#) is a unified server-side API that allows Web developers to retrieve data on the devices that are accessing their pages on a variety of device information database.

A client-side equivalent has been proposed as part of the [Systems Information API](#); while that specification is being vastly revisited, the Device APIs Working Group expects to do further work on a device information API.

To take advantage of the large variety of media capturing devices provided on mobile phones, the [The Media Capture API](#) offers some detailed indication on the features and capabilities these devices.

[CSS Media Queries](#) offer a mechanism that allows to adapt the layout and behavior of a Web page based on some of the characteristics of the device, including the screen resolution.

Expected progress and new work in upcoming year

The evolution of the Systems Information API and of the Media Capture API is expected to lead to the development of a new approach that would allow Web developers to use a single API to get access to this device information. One remaining concern with these types of APIs is that the collection of device-specific information allows to identify a single device and a single user (through fingerprinting) without her knowledge, which creates risk for privacy.

Work on allowing to load different versions of an image in HTML based on the device screen resolution has been proposed as part of the next generation of HTML, which should see a first draft published in the coming year.

The work on CSS Media Queries will focus on the building of a test suite to ensure as much interoperability as possible among browsers.

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Device information	Device Description Repository Simple API	Device Description Working Group (now closed)	 Recommendation	Finished	N/A	 Limited	Good Coverage
	Systems Information API	Device APIs Working Group	 Working Draft	Likely to be vastly reworked	Last update March 2011	 None	N/A
	The Media Capture API		 Working Draft	Likely to be vastly reworked	Last update December 2010	 Very limited	N/A
CSS-based Adaptation	Media Queries	CSS Working Group	 Candidate Recommendation	Mostly finished	Last update Aug 2010	 Widely deployed	Good coverage

FORMS

Ongoing Work

The ability to build rich forms with HTML is the basis for user input in most Web-based applications. Due to their limited keyboards, text input on mobile devices remains a difficult task for most users; [HTML5](#) address parts of this problem by offering new type of form controls that optimize the way users will enter data:

- [date and time entries](#) can take advantage of a number of dedicated form controls (e.g. `<input type="date">`) where the user can use a native calendar control;
- the `<input type="email">`, `<input type="tel">` and `<input type="url">` can be used to optimize the ways user enter these often-difficult to type data, e.g. through dedicated virtual keyboards, or by accessing on-device records for these data (from the address book, bookmarks, etc.);
- the [pattern](#) attribute allows both to guide user input as well as to avoid server-side validation (which requires a network round-trip) or JavaScript-based validation (which takes up more resources);
- the [placeholder](#) attribute allows to guide user input by inserting hints as to what type of content is expected in a text-entry control;
- the new `<datalist>` element allows to create free-text input controls coming with **pre-defined values** the user can select from.

Expected progress and new work in upcoming year

The work on the new generation of forms is now stabilizing and getting increasingly widely deployed on mobile devices.

Two new efforts are in the early stage of standardization in this space:

- the development of a component-based API that would make it easier for developers to create and share new form controls, independently of the browser release schedule; discussions on this topic are active in the Web Applications Working Group public mailing list.
- work on HTML Editing APIs that would streamline the creation of content from within the browser; this work is being explored in the [HTML Editing APIs Community Group](#).

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Date and time entries	HTML5 Date and Time state of input element	HTML Working Group	 <p>LC Last Call Working Draft</p>	Stabilizing	Updated regularly	 <p>Limited</p>	None
Customized text entries (tel, email, url)	HTML5 telephone, email and URL state of input element			Stabilizing	Updated regularly	 <p>Limited</p>	None
Input pattern	HTML5 pattern attribute			Stabilizing	Updated regularly	 <p>Very limited</p>	None
Input hint	HTML5 placeholder attribute			Stabilizing	Updated regularly	 <p>Limited</p>	None
Pre-defined values for text entries	HTML5 datalist element			Stabilizing	Updated regularly	 <p>Very limited</p>	None

4 USER INTERACTIONS

Ongoing Work

An increasing share of mobile devices relies on touch-based interactions. While the traditional interactions recognized in the Web platform (keyboard, mouse input) can still be applied in this context, a more specific handling of touch-based input is a critical aspect of creating well-adapted user experiences. As a result, work has started on defining [Touch Events in the DOM](#) (Document Object Model).

Conversely, many mobile devices use haptic feedback (such as vibration) to create new form of interactions (e.g. in games); work on a **vibration API** is under consideration in the [Device APIs Working Group](#) (proposed addition to the [draft new charter of the group](#).)

But as the Web reaches new devices, and as devices gain new user interactions mechanisms, it also becomes important to allow Web developers to react to a more abstract set of user interactions: instead of having to work in terms of “click”, “key press”, or “touch event”, being able to react to an “undo” command, or a “next page” command independently of how the user instructed it to the device will prove beneficial to the development of device-independent Web applications. Work on **abstract DOM events** that would address this need is planned as part of the [Web Events Working Group](#).

Mobile devices follow their users everywhere, and many mobile users rely on them to remind them or notify them of events, such as new messages: the [Web Notifications](#) specification proposes to add that feature to the Web environment. Similarly, the proposed addition to the [draft new charter of the Device APIs Working Group](#) of an API to generate system beeps (rather than app-provided sounds) would facilitate the integration of the underlying system mechanisms to notify the user.

Mobile devices, and mobile phones in particular, are also in many cases well-suited to be used through voice-interactions; the [HTML Speech Incubator Group](#) is exploring the opportunity of starting standardization work around a framework that would make it possible for users to interact with a Web page through spoken commands (see their [use cases and requirements](#).)

Expected progress and new work in upcoming year

The work on touch-based events is expected to reach Recommendation status in 2012. Early work on a next generation of such events has started and is expected to be published as a first draft in the coming months.

The Web Notifications specification is expected to reach Candidate Recommendation status in 2012.

The work on vibration and beep APIs in the Device APIs Working Group is dependent on submissions of technical proposals to the group, which haven't been made yet.

The HTML Speech Incubator Group is planning to request transition of its exploratory work to the W3C Recommendation track, most likely through the creation of a new Working Group in the last quarter of 2011 or first quarter of 2012.

A charter for a new Working Group, the [Model-Based UI Working Group](#), supported by the SERENOA EU project, is under discussion by the W3C Members. The goal of this group is to simplify the development of Web applications that work across a variety of user interaction modes.

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Touch-based interactions	Touch Events Specification	Web Events Working Group	 Working Draft	Early draft	Updated regularly	 Good deployment	None
Vibration	N/A	Device APIs	 N/A	Not started	Not started	 None	None
Intent-based events	N/A	Web Events Working Group	 N/A	Not started	Not started	 None	None
Notification	Web Notifications	Web Notifications Working Group	 Working Draft	Early draft	Regularly updated	 None	None
System beeps	N/A	Device APIs	 N/A	Not started	Not started	 None	None
Speech-based interactions	N/A	HTML Speech Incubator Group ,	 N/A	N/A	Use cases and requirements	 N/A	N/A
Model-based user interfaces	N/A	Proposed Model-Based User Interfaces Working Group	 N/A	N/A	Model-Based UI Incubator Group report	 N/A	None

5 DATA STORAGE

Ongoing Work

A critical component of many applications is the ability to save state, export content, as well as integrate data from other files and services on the system.

For simple data storage, the [Web Storage](#) specification offers two basic mechanisms, `localStorage` and `sessionStorage`, that can preserve data respectively indefinitely, or on a browser-session basis.

For richer interactions, the Web platform has a growing number of APIs to interact with a device filesystem: the [File Reader API](#) makes it possible to load the content of a file, the [File Writer API](#) allows to save or modify a file, while the nascent [FileSystems API](#) give access to more general file operations, including directory management.

On top of this file-based access, the [Indexed Database API](#) defines a database of values and hierarchical objects that integrates naturally with JavaScript, and can be queried and updated very efficiently. Note that the work around a client-side SQL-based database, which had been started in 2009, has been abandoned in favour of this new system.

Expected progress and new work in upcoming year

The Web Storage API is mostly finished and widely implemented; the work in this space will focus on the development of a test suite that will allow the completion of its standardization process.

The work on the various file-access APIs is expected to reach Candidate Recommendation in coming year.

The IndexedDB API should be published as a Last Call Working Draft in the coming months.

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Simple data storage	Web Storage	Web Applications Working Group	 Working Draft	Stabilizing	Updated regularly	 Well deployed	None
File reading	File API	Web Applications Working Group	 Working Draft	Stabilizing toward LC	Regular updates	 Limited	None
File writing	File API: Writer	Web Applications Working Group	 Working Draft	Early draft (but starting to stabilize)	Last update May 2011	 Limited	None
Filesystems operations	File API: Directories and System	Web Applications Working Group	 Working Draft	Early draft	Last update May 2011	 None	None
Database query/update	Indexed Database API	Web Applications Working Group	 Working Draft	Still changing, but starting to stabilize	Regularly updated	 Very limited	None
	Web SQL API	Web Applications Working Group	 Working Group Note	Abandoned	N/A	 Limited	None

6 PERSONAL INFORMATION MANAGEMENT

Ongoing Work

Applications can benefit from integrating with existing data records; on mobile devices, the address book and calendar are particularly useful source of information, which the [Contacts API](#) and the [Calendar API](#) bring access to.

Expected progress and new work in upcoming year

The Contacts API is expected to reach Candidate Recommendation status in the coming year, and is getting increasing attention from browser vendors.

The Calendar API is expected to reach Last Call in 2012.

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Address book data	Contacts API	Device APIs Working Group	 Last Call Working Draft	Stabilizing	Regularly updated	 Very limited	early draft
Calendar data	Calendar API	Device APIs Working Group	 Working Draft	Still changing	Regularly updated	 Very limited	None

7 SENSORS AND HARDWARE INTEGRATION

Ongoing Work

Mobile devices are packed with sensors, making them a great bridge between the real and virtual worlds: GPS, accelerometer, ambient light detector, microphone, camera, thermometer, etc.

To take full advantage of these sensors in mobile Web applications, Web developers need to be provided with hooks to interact with them.

The [Geolocation API](#) provides a common interface for locating the device, independently of the underlying technology (GPS, WIFI networks identification, triangulation in cellular networks, etc.)

Work has also started on providing access to orientation and acceleration data via the [DeviceOrientation Event Specification](#).

The [System Information API](#) proposes a generic API to get and monitor data from sensors, although the Working Group producing it is evaluating whether that generic approach is the most practical way forward.

As already mentioned in the section on [Multimedia](#), there is ongoing work on APIs to open up access to camera and microphone streams.

Expected progress and new work in upcoming year

The first version of the Geolocation API is expected to become a recommendation in the next few months, while a second version of this API that includes access to civic addresses should be published as first public draft.

The Orientation API is expected to stabilize into a Last Call Working Draft in the coming year.

The first drafts of the Web RTC Working Group are expected before the end of 2011.

The current proposal for a generic sensors API in the Device APIs Working Group is likely going to be replaced by new approaches based on recent input from Working Group Members, including based on the work done in the Webinos EU project which was recently brought to the group's attention.

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Geolocation	Geolocation API	Geolocation	 Candidate Recommendation	Mostly finished	Regularly updated	 Widely deployed	Good coverage
Accelerometer / Orientation	DeviceOrientation Event Specification	Geolocation	 Working Draft	stabilizing; last call planned before November	Regularly updated	 Limited, but growing	None
Battery Status	Battery Status Event	Device APIs	 Working Draft	Stabilizing	Updated regularly	 None	None
Generic sensors	The System Information API	Device APIs	 Working Draft	Need a lot of changes	Last update Jan 2011	 None	None
Camera & Microphone streams	API to get Video/Audio streams from camera/mike	Web Real-Time Communications Working Group	 N/A	None	getUserMedia proposal from WhatWG	 A few experimental ones	None

8 NETWORK

Ongoing Work

Network connectivity represents a major asset for mobile devices: the Web is an immense store of content, as well as an almost endless source of processing power, overcoming two of the limitations of mobile devices.

The Web platform is growing a number of APIs that facilitate establishing network connectivity in different contexts.

[XMLHttpRequest](#) (the “X” in AJAX) is a widely deployed API to load content from Web servers using the HTTP and HTTPs protocol.

A second version of that specification, [XMLHttpRequest Level 2](#) completes the existing API with the ability to make requests on servers in a different domain, programmatic feedback on the progress of the network operations, and more efficient handling of binary content.

By default, browsers do not allow to make request across different domains (or more specifically, across different *origins*, a combination of the protocol, domain and port) from a single Web page; this rule protects the user from having a Web site abusing their credentials and stealing their data on another Web site. Sites can opt-out of that rule by making use of the [Cross-Origin Resource Sharing](#) mechanism, opening up much wider cooperation across Web applications and services.

XMLHttpRequest is useful for client-initiated network requests, but mobile devices with their limited network capabilities and the cost that network requests induce on their battery (and sometimes on their users bill) can often make better use of server-initiated requests. The [Server-Sent Events](#) API allows to trigger DOM events based on push notifications (via HTTP and other protocols.)

The [WebSocket API](#), built on top of the IETF [WebSocket protocol](#), offers a bidirectional, more flexible, and less resource intensive network connectivity than XMLHttpRequest.

Of course, an important part of using network connectivity relies on being able to determine if such connectivity exists, and the type of network available. The [HTML5 onLine DOM flag](#) (and its associated change event, `ononline`) signals when network connectivity is available to the Web environment.

The [network-information API](#) addresses discovery of the network characteristics, allowing to determine for instance if the user is on a WIFI or a 3G connection.

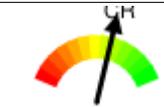
Expected progress and new work in upcoming year

The work on the first version of the XMLHttpRequest API will focus on the development of a more complete test suite, and its integration in the testing framework being set up in W3C.

The work on cross-origin requests (CORS) is expected to become a joint deliverable with the recently proposed [Web Application Security Working Group](#), which will also propose new specifications to reduce the risk of some script-based attacks (such as cross-site scripting).

Publication of Server-sent events as a Last Call Working Draft is expected in the coming weeks.

The Network information API I and the Web Socket API are expected to reach Last Call status in the coming year.

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
HTTP(s) network API	XMLHttpRequest	Web Applications WG	 Candidate Recommendation	Mostly finished	Regularly updated	 Widely deployed	Good coverage
XHR2	XMLHttpRequest Level 2	Web Applications WG	 Working Draft	Still changing, but starting to stabilize	Updated regularly	 Very limited	None
Cross-domain requests	Cross-Origin Resource Sharing	Web Applications WG	 Working Draft	Close to stabilizing	Last update November 2010	 Widely deployed (Implementation data)	None
Server-pushed requests	Server-Sent Events	Web Applications WG	 Last Call	Still changing but stabilizing	Regularly updated	 Growing	None
Bidirectional connections	The WebSocket API	Web Applications WG	 Working Draft	Still changing but stabilizing	Regularly updated	 Limited, but growing	None
on-line state	HTML5 onLine DOM state	HTML WG	 Last Call Working Draft	Mostly stable	Regularly updated	 Getting deployed	None

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Network information	The Network Information API	Device APIs WG	 <p>Working Draft</p>	Early draft	Regularly updated	 <p>Very limited</p>	N/A

9 COMMUNICATION

Ongoing Work

Beyond connection to on-line services, allowing communications between users, but also between devices and between applications is an important aspect of a good mobile development platform.

The [Messaging API](#) completes the existing ability to create and send message through links (with `sms:`, `mms:` and `mailto:` URI schemes) with more control on adding attachments and the success of the message sending.

The `postMessage` API of [HTML5 Web Messaging](#) allows for Web Applications to communicate between each other.

Exploratory work in the Device APIs Working Group, inspired by the [Web Introducer](#) and similar works, would also open up possibilities of closer integration of Web applications.

The recent launch of the [Web Real-Time Communications Working Group](#) is the starting point of a much wider set of communication opportunity:

- **Peer-to-peer connection** across devices,
- **P2P Audio and video streams** allowing for real-time communications between users.

Expected progress and new work in upcoming year

The Web Real Time Communications Working Group is expected to publish the first drafts of its P2P connection APIs in the last quarter of 2011.

The work on HTML5 Web messaging is expected to reach Candidate Recommendation status in the coming months and will thus focus on the development of a test suite for the specification.

The work on email, SMS and MMS messaging in the Device APIs Working Group is still in flux, but based on renewed interest from browser vendors is expected to reach Last Call status in the coming year.

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Emails, SMS and MMS with generated attachments	The Messaging API	Device APIs Working Group	 Working Draft	Still subject to large changes	Last update July 2011	 None	None
Inter-app communications	HTML5 Web Messaging	Web Applications Working Group	 Last Call	Stabilizing	Regularly updated	 Limited	None
Inter-app triggers	None	Possibly Device APIs Working Group	 N/A	Not started	Possible inspiration from Web Introducer	 None	None
P2P connections	None	Web Real-Time Communications Working Group	 N/A	None	N/A	 None	None
P2P Video/Audio streams	None	Web Real-Time Communications Working Group	 N/A	None	N/A	 None	None

10 PACKAGING

Ongoing Work

An important aspect of the user experience of applications is linked to how the user perceives the said application is available permanently (even when off-line, which is particularly important on mobile devices), as well as how it can be shared and distributed, typically through purchases via applications stores — this is adequately addressed by packaging the application.

The Web platform offers two complementary approaches to packaging Web applications:

- HTML5's [ApplicationCache](#) enables access to Web applications off-line through the definition of a manifest of files that the browser is expected to keep in its cache;
- the [W3C Widgets](#) family of specifications define a framework for distributing Web applications as Zip files completed by a configuration file (see [Widget Packaging and Configuration](#)); this configuration file is the basis for additional features such as [signature of applications](#) , controlled access to advanced APIs, [restricted network usage](#), etc.

Expected progress and new work in upcoming year

The family of Widgets specifications are expected to reach Recommendation status in the coming year. Discussions around starting a second generation of these specifications have started, some of them being explored in the newly formed [Native Web Apps Community Group](#).

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Application Cache	HTML5 Application Cache	HTML Working Group	 Last Call Working Draft	Still changing but stabilizing	Regularly updated	 Getting deployed	None
Widgets	Widgets Packaging & Configuration	Web Applications Working Group	 Proposed Recommendation	Mostly finished	Last update Aug 2011	 Getting deployed	Full coverage
	Digital Signatures for Widgets		 Proposed Recommendation	Mostly finished	Last update Aug 2011	 Getting deployed	Full coverage
	Widget Access Request Policy		 Proposed Recommendation	Mostly finished	Last update Aug 2011	 Getting deployed	Full coverage

11 PERFORMANCE & OPTIMIZATION

Ongoing Work

Due to their limited CPU, and more importantly to their limited battery, mobile devices require a lot of attention in terms of performance.

The work started by the [Web Performance Working Group](#) on [Navigation Timing](#), [Resource Timing](#), and more recently [Performance Timeline](#) and [User Timing](#), gives tools to Web developers for optimizing their Web applications.

The proposed work on [Efficient Script Yielding](#) offers the opportunity to Web developers to use more efficiently asynchronous programming.

The [API to determine whether a Web page is being displayed](#) (**Page Visibility API**) can also be used to adapt the usage of resources to the need of the Web application, for instance by reducing network activity when the page is minimized. Likewise, the [Timing control for script-based animations API](#) can help reduce the usage of resources needed for playing animations.

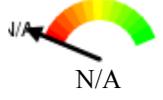
Beyond optimization of resources, the perceived reactivity of an application is also a critical aspect of the mobile user experience. The **thread-like mechanism** made possible via [Web Workers](#) allows keeping the user interface responsive by offloading the most resource-intensive operations into a background process.

The [Mobile Web Application Best Practices](#) provide general advice on how to build Web applications that work well on mobile devices, taking into account in particular the needs for optimization.

Expected progress and new work in upcoming year

Most of the APIs from the Web Performance Working Group are expected to reach Recommendation status in the coming year.

The Web Workers specification should become a Candidate Recommendation in 2012.

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Timing hooks	Navigation Timing	Web Performance Working Group	 Candidate Recommendation	Mostly finished	Regularly updated	 Getting deployed	Started
	Resource timing		 Last Call Working Draft	Stabilizing	Regularly updated	 None	N/A
	Performance Timeline		 Working Draft	Early draft	Regularly updated	 None	N/A
	User timing		 Working Draft	Early draft	Regularly updated	 None	N/A
Priority handling	Efficient Script Yielding		 N/A	Early draft	Regularly updated	 None	N/A
Page Visibility detection	Page visibility API		 Last Call Working Draft	Stabilizing	Regularly updated	 None	N/A

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Animation optimization	Timing control for script-based animations		Working Draft	Early draft	Regularly updated	None 	N/A
Threading	Web Workers	Web Applications Working Group	 Last Call	Stabilizing	Regularly updated	Limited 	None
Optimization Best Practices	Mobile Web Application Best Practices	Mobile Web Best Practices Working Group (now closed)	 Standard	Finished	N/A	N/A 	N/A

RECAPITULATION OF PLANNED STANDARDIZATION WORK IN UPCOMING YEAR

Graphics

- A new version of SVG (SVG 2.0) is expected to start, with the goal of providing better integration with other Web technologies (e.g. HTML5 and CSS3), making it possible to use some of the most advanced features of SVG directly from regular Web pages. High priority features include Filters, Web Animation, Compositing, Integration, Gradients, Color Management, and 2D Transforms. The increase of capabilities in mobile devices will make it unnecessary to build a separate version targeted specially at these devices.
- The canvas API is expected to bring the specification to Candidate Recommendation in the coming year.
- The downloadable font specification is expected to reach Recommendation status in the first quarter of 2012.
- The [Declarative 3D for the Web Architecture Community Group](#) was recently launched and will explore the opportunity of developing a declarative approach for 3D graphics, completing what WebGL provides as an API.

Multimedia

- The Web RTC Working Group is expected to publish the first drafts of its APIs to access and handle audio/video streams in the last quarter of 2011.
- Likewise, the first drafts of the API to analyse and synthesize Audio data should be published before the end of 2011.

Device Adaptation

- The evolution of the Systems Information API and of the Media Capture API is expected to lead to the development of a new approach that would allow Web developers to use a single API to get access to this device information.
- Work on allowing to load different versions of an image in HTML based on the device screen resolution has been proposed as part of the next generation of HTML, which should see a first draft published in the coming year.
- CSS Media Queries will focus on the building of a test suite to ensure as much interoperability as possible among browsers.

Forms

The work on the new generation of forms is now stabilizing and getting increasingly widely deployed on mobile devices.

Two new efforts are in the early stage of standardization in this space:

- The development of a component-based that would make it easier for developers to create and share new form controls, independently of the browser release schedule; discussions on this topic are active in the Web Applications Working Group public mailing list.
- Work on HTML Editing APIs that would streamline the creation of content from within the browser; this work is being explored in the [HTML Editing APIs Community Group](#).

User interactions

- The work on touch-based events is expected to reach Recommendation status in 2012. Early work on a next generation of such events has started and is expected to be published as a first draft in the coming months.
- The Web Notifications specification is expected to reach Candidate Recommendation status in 2012.
- The work on vibration and beep APIs in the Device APIs Working Group is dependent on submissions of technical proposals to the group, which haven't been made yet.
- The HTML Speech Incubator Group is planning to request transition of its exploratory work to the W3C Recommendation track, most likely through the creation of a new Working Group in the last quarter of 2011 or first quarter of 2012.

- A charter for a new Working Group, the [Model-Based UI Working Group](#), supported by the SERENOA EU project, is under discussion by the W3C Members. The goal of this group is to simplify the development of Web applications that work across a variety of user interaction modes.

Data storage

- The Web Storage API is mostly finished and widely implemented; the work in this space will focus on the development of a test suite that will allow the completion of its standardization process.
- The work on the various file-access APIs is expected to reach Candidate Recommendation in coming year.
- The IndexedDB API should be published as a Last Call Working Draft in the coming months.

Personal Information Management

- The Contacts API is expected to reach Candidate Recommendation status in the coming year, and is getting increasing attention from browser vendors.
- The Calendar API is expected to reach Last Call in 2012.

Sensors and hardware integration

- The first version of the Geolocation API is expected to become a recommendation in the next few months, while a second version of this API that includes access to civic addresses should be published as first public draft.
- The Orientation API is expected to stabilize into a Last Call Working Draft in the coming year.
- The first drafts of the Web RTC Working Group are expected before the end of 2011.
- The current proposal for a generic sensors API in the Device APIs Working Group is likely going to be replaced by new approaches based on recent input from Working Group Members, including based on the work done in the Webinos EU project which was recently brought to the group's attention.

Network

- The work on the first version of the XMLHttpRequest API will focus on the development of a more complete test suite, and its integration in the testing framework being set up in W3C.
- The work on cross-origin requests (CORS) is expected to become a joint deliverable with the recently proposed [Web Application Security Working Group](#), which will also propose new specifications to reduce the risk of some script-based attacks (such as cross-site scripting).
- Publication of Server-sent events as a Last Call Working Draft is expected in the coming weeks.
- The Network information API I and the Web Socket API are expected to reach Last Call status in the coming year.

Communication

- The Web Real Time Communications Working Group is expected to publish the first drafts of its P2P connection APIs in the last quarter of 2011.
- The work on HTML5 Web messaging is expected to reach Candidate Recommendation status in the coming months and will thus focus on the development of a test suite for the specification.
- The work on email, SMS and MMS messaging in the Device APIs Working Group is still in flux, but based on renewed interest from browser vendors is expected to reach Last Call status in the coming year.

Packaging

- The family of Widgets specifications are expected to reach Recommendation status in the coming year. Discussions around starting a second generation of these specifications have started, some of them being explored in the newly formed [Native Web Apps Community Group](#)

Performance & Optimization

- Most of the APIs from the Web Performance Working Group are expected to reach Recommendation status in the coming year.

- The Web Workers specification should become a Candidate Recommendation in 2012.