



MobiWebApp

Mobile Web Applications for Future Internet Services

Deliverable D4.2

**Standardization Roadmap
Year 2**

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:	D4.2
Deliverable title	
Contractual Date of Delivery:	Project month 24
Actual Date of Delivery:	August 31, 2012
Editor (s):	Dominique Hazaël-Massieux
Author (s):	Dominique Hazaël-Massieux
Reviewer (s):	Dr. Marie-Claire Forgue
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:	Final
Total number of pages (including cover):	41
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.....	11
3 Device adaptation	16
4 Forms	18
5 User interactions.....	20
6 Data storage.....	23
7 Personal Information Management	26
8 Sensors and hardware integration.....	28
9 Network	31
10 Communication	34
11 Packaging.....	36
12 Performance & Optimization.....	38

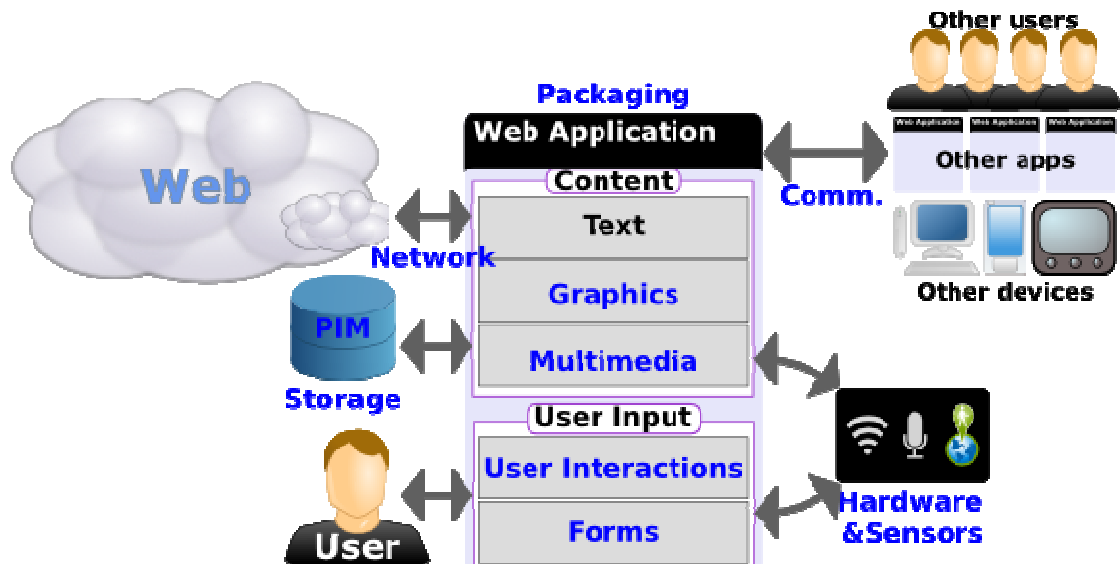
INTRODUCTION

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 capabilities 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 and Discovery](#)
- ⤴ [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, based on data collected primarily from Can I Use¹ and mobile HTML5², completed with data from Mozilla developer network³, QuirksMode⁴, as well as the author’s understanding of the mobile devices market (see also the code used to generate the support icons⁵)
- ^ 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](#).

In 2011, W3C has set up [Community Groups](#), a 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 <http://caniuse.com/>

² <http://mobilehtml5.org/>

³ <https://developer.mozilla.org/>

⁴ <http://quirksmode.org/>

⁵ <https://github.com/dontcallmedom/canmymobilebrowser>

1 GRAPHICS

[SVG](#), Scalable Vector Graphics, provides an XML-based markup language to describe two-dimensions 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. SVG 2.0 is set to facilitate that integration and complete the set of features in SVG.

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 do 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.





Another important aspect in graphics-intensive applications (e.g. games) is the possibility to use the entire screen to display the said graphics; the [Fullscreen API](#) lets a Web application requests and detects full screen display.

Likewise, in these scenarios, it is often useful to be able to **lock the orientation of the screen**; the [Screen Orientation API](#) allows not only to detect orientation change, but also to lock the orientation in a specific state.

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.

Feature	Specification	Working Group	Maturity	Stability	Latest draft	editors	Current implementations	Test suite
2D Vector Graphics	SVG Tiny 1.2	SVG Working Group	Recommendation	Finished	New version of SVG (SVG 2.0) in preparation		Widely deployed 	High coverage
	SVG 2.0		N/A	N/A	Editors draft		N/A	N/A
2D Programmatic API	HTML Canvas 2D Context	HTML Working Group	Working Draft	Mostly stable	Updated regularly		Widely deployed 	Good coverage
Rounded Corners	CSS Backgrounds and Borders	CSS Working Group	Candidate Recommendation	Mostly finished	Updated regularly		Deployed as an extension in many mobile browsers 	None
Complex background images							Growing deployment 	

Feature	Specification	Working Group	Maturity	Stability	Latest draft	editors	Current implementations	Test suite
Box shadow effects							Widely deployed 	
2D Effects	<u>CSS Transforms</u>		Working Draft	Mostly stable	<u>Latest update Aug 2012</u>		Well deployed 	<u>Good coverage</u>
3D Effects				Stabilizing			Growing deployment 	
Animations	<u>CSS Animations Module Level 3</u>		Working Draft	Early draft	<u>Updated regularly</u>		Growing deployment 	None
	<u>CSS Transitions Module Level 3</u>		Working Draft	Early draft	<u>Latest update July 2012</u>		Well deployed 	None

Feature	Specification	Working Group	Maturity	Stability	Latest draft	editors	Current implementations	Test suite
Downloadable fonts	<i>Timing control for script-based animations API</i>	Web Performance Working Group	Last Call Working Draft	Stabilizing	Regularly updated		Very limited 	Started
	WOFF File Format 1.0	WebFonts Working Group	Candidate Recommendation	Mostly stable	Latest update Jan 2012		Good deployment 	Good coverage
Fullscreen display	Fullscreen API	Web Apps and CSS Working Groups	Working Draft	Early draft	Regularly updated		None 	None
Orientation Lock	The Screen Orientation API	Web Apps Working Groups	Working Draft	Early draft	Regularly updated		None 	None

2 MULTIMEDIA

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.

To cater for the needs of some content providers, a proposal to enable **playback of protected content**, [Encrypted Media Extensions](#) is an API that is under consideration in the [HTML Working Group](#).

The [Pick Media Intent](#) offers a Web-intent based approach to search and retrieve **locally or remotely stored media content**, while the [Networked Service Discovery and Messaging API](#) opens the door for integrating DLNA-hosted content into Web applications.

While the new HTML5 tags allow playing multimedia content, the [HTML Media Capture](#) defines a **markup-based mechanism to access captured multimedia content** using attached camera and microphones, a very common feature on mobile devices. The [Web Real-Time Communications Working Group](#) and the [Device APIs Working Group](#) are building together an [API \(getUserMedia\)](#) 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, the [Audio Working Group](#) is working on an API that that makes it possible to modify audio content, as well as **analyze, modify and synthesize sounds**, the [Web Audio API](#), in favor of which the competing proposal called [MediaStream Processing API](#) has been abandoned.

The combination of all these features marks 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.

HTML5 adds two tags that dramatically improve the integration of multimedia content on the Web: the [<video>](#) and [<audio>](#) tags. Respectively, these tags allow embedding video and audio content, and making 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 playing 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 marks 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.

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Video playback	HTML5 video element		Working Draft			Good deployment 	Just started
					Stabilizing	Updated regularly	
Audio playback	HTML5 audio element	HTML Working Group	Last Call Working Draft			Good deployment 	Barely started
Protected content playback	<i>Encrypted Media Extensions</i>		N/A	Early proposal	Latest update Aug 2012	None 	None
Multimedia Gallery access	Pick Media Intent	Device APIs Working Group	Working Draft	Early Web-intents based approach	Last updated Aug 2012	None 	N/A
	Networked Service Discovery and Messaging		Working Draft	Early draft	Last updated Aug 2012	None 	None

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Capturing audio/video	HTML Media Capture	Device APIs Working Group	Last Call Working Draft	Stabilizing	Latest update July 2012	Limited, but growing 	None
	Media Capture and Streams	Joint work between Web Real-Time Communications Working Group and Device APIs Working Group	Working Draft	Stabilizing, but still subject to large changes	last updated Aug 2012	A few experimental ones 	Started
Image & Video analysis, modification	HTML Canvas 2D Context	HTML Working Group	Working Draft	Mostly stable	Updated regularly	Widely deployed 	Good coverage
Audio analysis, modification	Web Audio API	Audio Working Group	Working Draft	Early work	Regularly updated	A couple 	None

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
	<u>MediaStream Processing API</u>		Abandoned	Abandoned	N/A	None 	None

3 DEVICE ADAPTATION

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.

The [Media Capture task force](#) is currently evaluating if and how to expose capabilities from camera and microphones to make it possible to take advantage of the large variety of media capturing devices provided on mobile phones.

[CSS Media Queries](#) offer a mechanism that allows adapting the layout and behavior of a Web page based on some of the characteristics of the device, including the screen resolution. [CSS Device Adaptation](#) defines a set of CSS directives to define the size on which this layout should be based, relatively to the size of the underlying device — specifying what has been implemented using the `<meta name="viewport">` element so far.

The [Responsive Images Community Group](#) has been exploring proposals to make it possible to load and display images in HTML that are best adapted to the resolution of the device, and is now [looking into incorporating a <picture> element in HTML5](#).






Complementarily, the [WHATWG](#) is discussing a [proposal for a sreset attribute](#) that would let Web developers define the various existing resolutions of an image, letting the browser pick the best choice for the resolution of the screen.

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Device information	Device Description Repository Simple API (server-side)	Device Description Working Group (now closed)	Recommendation	finished	N/A	Limited	Good Coverage
	Media Capture Capabilities API	WebRTC and Device APIs Working Group	N/A	Not started	N/A	N/A	N/A
CSS-based adaptation	Media Queries	CSS Working Group	Recommendation	Finished	Latest update Apr 2012	Widely deployed 	Good coverage
	CSS Device Adaptation		Working Draft	Early draft	Latest update Apr 2012	Very limited 	N/A
Adaptive images	picture element	Responsive Images Community Group	N/A	Proposal	Last updated Jul 2012	None 	N/A
	srcset attribute	WHATWG	N/A	Proposal	Regularly updated	None 	None

4 FORMS

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.

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		Working Draft	Stabilizing	Updated regularly	Limited 	Just started
Customized text entries (tel, email, url)	HTML5 telephone, email and URL state of input element		Working Draft	Stabilizing	Updated regularly	Limited, but growing 	None
Input pattern	HTML5 pattern attribute	HTML Working Group	Working Draft	Stabilizing	Updated regularly	Very limited 	Just started
Input hint	HTML5 placeholder attribute		Working Draft	Stabilizing	Updated regularly	Well deployed 	None
Pre-defined values for text entries	HTML5 datalist element		Working Draft	Stabilizing	Updated regularly	Very limited 	None

5 USER INTERACTIONS

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, which [Touch Events in the DOM](#) (Document Object Model) enable. The work on that specification is now nearly finished, as the [patents that had been disclosed](#) have been [determined to not apply](#).

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](#) in the [Device APIs Working Group](#) is making good progress.






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 a joint task force between the [Web Events Working Group](#) and the [Indie UI 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 messages: the [Web Notifications](#) specification proposes to add that feature to the Web environment.

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

The hardware constraints of mobile devices, and their different usage context can make [mobile users experience similar barriers to people with disabilities](#). These similarities in barriers mean that similar solutions can be used to cater for them, [making a Web site accessible both for people with disabilities and mobile devices](#) a natural goal. The [Relationship between Mobile Web Best Practices and WCAG](#) document looks into these similarities.

[WAI-ARIA](#) provides semantic information on widgets, structures and behaviors hooks to make Web applications more accessible, including on mobile devices.

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Touch-based interactions	Touch Events Specification	Web Events Working Group	Candidate Recommendation	Mostly finished	Updated regularly	Largely deployed 	Just started
Vibration	Vibration API	Device API	Candidate Recommendation	Stable	Updated regularly	Experimental implementations 	started
Intent-based events	N/A	Indie UI Working Group and 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	A few experimental ones 	None
Speech-based interactions	N/A	Speech API Community Group	N/A	N/A	Regularly updated	N/A 	N/A

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Model-based user interfaces	N/A	Model-Based User Interfaces Working Group	N/A	N/A	Model-based UI Incubator Group report	N/A	N/A
Accessibility	Relationship between Mobile Web Best Practices (MWBP) and Web Content Accessibility Guidelines (WCAG)	Mobile Web Best Practices Working Group & Education and Outreach Working Group	Working Group Note	Finished	N/A	N/A	N/A
	Accessible Rich Internet Applications (WAI-ARIA) 1.0	Protocols & Formats Working Group	Candidate Recommendation	Stable	Latest update May 2012	Growing deployment 	None

6 DATA STORAGE






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 favor of this new system.

As more and more data need to be stored by the browser (e.g. for offline usage), it becomes critical for developers to get reliable storage space, which the proposed [Quota Mangement API](#) will offer to Web applications.



Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Simple storage	data Web Storage	Web Applications Working Group	Candidate Recommendation	Stable	Updated regularly	Well deployed 	Well started
File reading	File API		Working Draft	Stabilizing toward LC	Regular updates	Growing 	None
File writing	File API: Writer		Working Draft	Early draft (but starting to stabilize)	Latest update Mar 2012	None 	None
Filesystems operations	File API: Directories and System		Working Draft	Early draft	Latest update Mar 2012	None 	None
Database query/update	Indexed Database API		Last Call Working Draft	Stabilizing	Regularly updated	Growing 	Started

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Quota Storage	Web SQL API		Working Group Note	Abandoned	N/A	Somewhat deployed, but won't be further deployed 	N/A
	<i>Quota Management API</i>		Working Draft	Early work	Last updated June 2012	N/A	N/A

7 PERSONAL INFORMATION MANAGEMENT

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.

The current JavaScript APIs are being replaced with an approach based on [Web Intents](#). A purely programmatic approach is also part of the proposed new [System Applications Working Group](#).

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Address book data	Pick Contacts Intent	Device APIs Working Group	Working Draft	Early Web-intents based approach; a more complete API will be developed in the proposed new SysApps Working Group	Last updated Aug 2012	None 	Early draft based on previous API
Calendar data	Calendar API		Working Draft	Will likely change significantly	Regularly updated	None 	None

8 SENSORS AND HARDWARE INTEGRATION

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.) The proposed [second version of that API](#) which would have added the ability to retrieve a civic address matching the user's current location, has been [abandoned](#) due to lack of demand.





Web applications can also now access **orientation and acceleration** data via the [DeviceOrientation Event Specification](#).




The work on a generic [Sensor API](#) has been put on hold in favor to designing APIs for specific sensors, such as the [Proximity Events API](#), the [Ambient Light Events API](#) or the proposed [Ambient Humidity Events API](#).

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

[Discussions on enabling Web applications to use Near-Field Communications \(NFC\)](#) mechanisms have led to a [proposed charter for a dedicated Working Group](#), and would likely lead to the creation of a dedicated Working Group, currently under [review by W3C Members](#).

A more global access to sensors and hardware (including USB and bluetooth) would be in scope for the proposed **new** [System Applications Working Group](#), currently under [review by W3C Members](#).

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Geolocation	<u>Geolocation API</u>		Proposed Recommendation	Mostly finished	<u>Regularly updated</u>	Widely deployed 	<u>Good coverage</u>
	<u>Geolocation API v2</u>	<u>Geolocation Working Group</u>	Abandoned	Abandoned	N/A	None	N/A
Motion sensors	<u>DeviceOrientation Event Specification</u>		Last Call Working Draft	Stabilizing	<u>Regularly updated</u>	Growing 	<u>Started</u>
Battery Status	<u>Battery Status API</u>	<u>Device APIs Working Group</u>	Candidate Recommendation	Stable	<u>Updated regularly</u>	Experimental implementations 	None
Proximity sensors	<u>Proximity Events</u>		Working Draft	Early draft	<u>Regularly updated</u>	A couple of experimental ones 	<u>Started</u>

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Ambient Light sensor	<i>Ambient Light Events</i>		Working Draft	Early draft	Regularly updated	None 	Started
Humidity sensor	<i>Ambient Humidity Events</i>		N/A	Unofficial draft	Last updated Aug 2012	None 	N/A
Camera & Microphone streams	<i>Media Capture Streams</i>	Web Real-Time Communications Working Group and Device APIs Working Group	Working Draft	Stabilizing, but still subject to large changes	Latest update Aug 2012	A few experimental ones 	Started

9 NETWORK

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: the W3C specification (formerly known as *XMLHttpRequest Level 2*) completes the existing deployed 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. The work on documenting the currently deployed API (XMLHttpRequest Level 1) has been abandoned in favor of getting the new API developed more quickly.

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.)





Early work on a [Push API](#) would allow Web applications to receive server-sent messages whether or not the said Web app is active in a browser window.




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

The work on [Web Real-Time Communications](#) will also provide direct **peer-to-peer data connections** between browsers with real-time characteristics, opening the way to collaborative multi-devices Web applications.

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 the rough bandwidth of the current connection.

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
HTTP(s) network API	<u>XMLHttpRequest</u>		Working Draft	Still changing, but starting to stabilize	<u>Regularly updated</u>	Very broad for level 1 features, growing for level 2 	<u>Coverage of XMLHttpRequest "level 1"</u>
Cross-domain requests	<u>Cross-Origin Resource Sharing</u>	<u>Web Applications Working Group</u>	Last Call Working Draft	Stable	<u>Latest update May 2012</u>	Growing deployment 	<u>Started</u>
Server-pushed requests	<u>Server-Sent Event</u>		Last Call Working Draft	Stabilizing	<u>Regularly updated</u>	Growing 	None (?)
	<i>Push API</i>		N/A	Very early draft	<u>Last updated Aug 2012</u>	None	N/A
Bidirectional connections	<u>The WebSocket API</u>		Last Call Working Draft	Stable	<u>Regularly updated</u>	Growing 	<u>Started</u>

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
P2P data connections	<u>WebRTC 1.0: Real-time Communication Between Browsers</u>	<u>Web Real-Time Communications Working Group</u>	Working Draft	Early draft	<u>Regularly updated</u>	None 	None
on-line state	<u>HTML5 onLine DOM state</u>	<u>HTML Working Group</u>	Working Draft	Mostly stable	<u>regularly updated</u>	Limited 	None
Network characteristics	<u>The Network Information API</u>	<u>Device APIs Working Group</u>	Working Draft	Early draft	<u>Regularly updated</u>	Very limited 	None

10 COMMUNICATION

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. To communicate with unknown devices or pre-existing services, a discovery component is critical.

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. At this time, this API is likely to be entirely replaced by an approach based on [Web Intents](#).






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

Work has started in a joint task force of the Device APIs and Web Apps Working Groups that open up possibilities of closer integration of Web applications, as well as of Web applications with native applications via a mechanism called [Web Intents](#).

The [Networked Service Discovery and Messaging](#) API offers to discover services on the local network (such as the ones offered via DLNA), enabling mobile Web applications to integrate seamlessly with these services.

The [Web Real-Time Communications Working Group](#) is the host of specifications for a wider set of communication opportunities:

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

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	Candidate for replacement by a Web Intents -based approach	Latest update July 2011	None 	None
Inter-app communications	HTML5 Web Messaging	Web Applications Working Group	Candidate Recommendation	Stable	Regularly updated	Well deployed 	Started
Inter-app triggers	Web Intents	Device APIs Working Group and Web Apps Working Group	Working Draft	Early draft	regularly updated	Experimental 	None
Networked services discovery	Networked Service Discovery and Messaging	Device APIs Working Group	Working Draft	Early draft	Last updated Aug 2012	None 	None
P2P connections	WebRTC 1.0: Real-time	Web Real-Time Communications Working Group	Working Draft	Early draft	Regularly updated	None 	None
P2P Video/Audio streams	Communication Between Browsers						



11 PACKAGING

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 technologies for distributing Web applications as Zip files which include a configuration file (see [Widget Packaging and Configuration](#)); this configuration file enables the inclusion of additional features such as [signature of applications](#), controlled access to advanced APIs, [restricted network usage](#), etc. In addition to aiding in the development of client-side Web applications for mobile devices, W3C Widgets have been used as server side-applications, standalone applications, daemons, starting point for hybrid Web/native applications, and as a Browser extension format.

As part of its [new charter](#), the Web Apps Working Group is considering to work on an evolution of the Widgets configuration file based on a [JSON format](#), although the [latest discussions on it](#) have established a likely dependency with the [proposed new System Applications Working Group](#).

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Application Cache	HTML5 Application Cache	HTML Working Group	Working Draft	Still changing but stabilizing	Regularly updated	Well deployed 	None
	Widgets Packaging & Configuration		Recommendation	Finished	Latest update Aug 2011	4 complete implementations; 1 impl 99% 	Full coverage
Widgets	Digital Signatures for Widgets	Web Applications Working Group	Proposed Recommendation	Finished	Latest update Aug 2011	2 or more implementations pass each test	Full coverage
	Widget Access Request Policy (WARP)		Recommendation	Finished	Latest update Dec 2011	3 complete implementations; 1 impl 98%	Full coverage
Packaged Web App	<i>Web Application Manifest Format and Management APIs</i>		N/A	N/A	Last updated June 2012	N/A	N/A

12 PERFORMANCE & OPTIMIZATION

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.

The [battery API](#) allows to adjust the use of resources to the current level of power available in the battery of a mobile device.

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.

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Timing hooks	Navigation Timing	Web Performance Working Group	Proposed Recommendation	Mostly finished	Regularly updated	Getting deployed 	Medium coverage
	Resource timing		Candidate Recommendation	Stable	Regularly updated	None 	None
	Performance Timeline		Candidate Recommendation	Stabilizing	Regularly updated	None 	None
	User timing		Candidate Recommendation	Stable	Regularly updated	None 	None
Priority handling	Efficient Script Yielding		N/A	Early draft	Regularly updated	Very limited 	None

Feature	Specification	Working Group	Maturity	Stability	Latest editors draft	Current implementations	Test suite
Page Visibility detection	<u>Page visibility API</u>		Candidate Recommendation	Stable	<u>Regularly updated</u>	Very limited 	<u>Started</u>
Animation optimization	<u>Timing control for script-based animations</u>		Last Call Working Draft	Stabilizing	<u>Regularly updated</u>	Very limited 	<u>Started</u>
Threading	<u>Web Workers</u>	<u>Web Applications Working Group</u>	Candidate Recommendation	Stable	<u>Regularly updated</u>	Growing 	<u>Started</u>
Battery Status	<u>Battery Status Events</u>	<u>Device APIs Working Group</u>	Candidate Recommendation	Stable	<u>Updated regularly</u>	Very limited 	None
Optimization Best Practices	<u>Mobile Web Application Best Practices</u>	<u>Mobile Web Best Practices Working Group</u> (now closed)	Recommendation	Finished	N/A	N/A	N/A