



HTML5APPS

DELIVERABLE D3.4 APPS STANDARDIZATION ROADMAP 2

30 September 2015

Project	
Grant Agreement number	611327
Project acronym:	HTML5Apps
Project title:	HTML5Apps: Closing the Gaps
Funding Scheme:	Coordination & Support Action
Date of latest version of Annex I against which the assessment will be made:	May 24, 2013
Document	
Deliverable number:	D3.4
Deliverable title	Apps Standardization roadmap 2
Contractual Date of Delivery	M24
Actual Data of Delivery:	M24
Editor(s):	Dominique Hazael-Massieux
Author(s):	
Reviewer(s):	Philipp Hoschka, Daniel Dardailler
Participant(s):	
Work package no.:	3
Work package title:	Standards Coordination
Work package leader:	Daniel Dardailler
Distribution:	PU
Version/Revision:	1.0
Draft/Final:	Final
Total number of pages (including cover):	69
Keywords:	

DISCLAIMER

This document contains description of the HTML5Apps 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 HTML5Apps 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/index_en.html)



HTML5Apps is a project funded in part by the European Union.

DOCUMENT HISTORY

Version	Date	Comments
0.1	3 September 2015	Submitted to internal review
0.2	28 September 2015	Second internal review with expanded introduction
1.0	30 September 2015	Submission to the Commission

TABLE OF CONTENTS

1. Introduction	6
2. Standards for Web Applications on Mobile: current state and roadmap	7
2.1. Status and changes	8
2.2. Document structure	10
2.3. Core Web Design and Development	13
2.3.1. Graphics and Layout	13
2.3.2. Device Adaptation	20
2.3.3. Forms	24
2.3.4. Data storage	27
2.4. Media and Real-Time Communications	31
2.5. Usability and Accessibility	38
2.6. Device interaction	44
2.7. Network Integration	48
2.8. Application Lifecycle	53
2.9. Payment and Services	58
2.10. Performance & Tuning	60
2.11. Security & Privacy	64
2.12. Acknowledgments	69

1. INTRODUCTION

To facilitate information, coordination and participation of research projects with respect to HTML5 app standardization, the HTML5Apps project has developed a standardization roadmap outlining the current state of standardization efforts relevant to HTML5 apps inside and outside of W3C.

Compared to earlier versions of the roadmap introduced in a previous activity, the HTML5Apps project has introduced a number of very significant changes:

- The report is now built along the categories defined in the [application foundations framework](#), making it thus easier to read and navigate for Web developers;
- The [online version of the report](#) loads at least 10 times faster on average, thanks to performance improvements made to how it is built and rendered in browsers;
- The infrastructure used to maintain and generate the report was moved from a ill-documented and hard-to-use set of one-off tools to a much more contributor-friendly github repository;
- The report highlights which specifications and test suites can be contributed to via github, which increases the opportunities for developers to participate in the evolution of the Web platform.

The HTML5Apps project has published seven editions of this roadmap (one every quarter from January 2014 to August 2015). Each of these editions tracked and documented the rapid pace of changes in the Web platform, in terms of new standardization groups, new specifications, new implementations, and new testing efforts.

This is the final App Standardization roadmap published by the HTML5Apps project.

2. STANDARDS FOR WEB APPLICATIONS ON MOBILE: CURRENT STATE AND ROADMAP

AUGUST 2015



Latest version

<http://www.w3.org/Mobile/mobile-web-app-state/>

This version

<http://www.w3.org/2015/08/mobile-web-app-state/> (PDF version)

Previous version

<http://www.w3.org/2015/05/mobile-web-app-state/>

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. A good subset of these technologies are described and explained in the [W3C on-line training on programming Web applications](#).

1. Core Web Design and Development
 - Graphics and Layout
 - Device Adaptation
 - Forms
 - Data storage
2. Media and Real-Time Communications
3. Usability and Accessibility
4. Device Interaction
5. Network Integration
6. Application Lifecycle
7. Payment and Services
8. Performance & Tuning
9. Security & Privacy

2.1. STATUS AND CHANGES

This document is the 18th edition of this overview of mobile Web applications technologies. The previous edition was released in [May 2015](#). A live version of this document accepts contributions on [the W3C Web and Mobile Interest Group Github repository](#).

This document is published by the [Web and Mobile Interest Group](#); feedback on every aspect of this document should be sent to public-web-mobile@w3.org, the [publicly archived mailing list](#) of the interest group, or raised as [issues on the Github repository](#), or alternatively to the author (dom@w3.org). It will serve as input for the next iteration of the document.

The following changes in the Web platform since May 2015 are documented in this update:

Emerging Work

- The early work on a [generic sensor API](#) is now embodied in an editors draft
- A proposal to enable [CSS transitions between Web pages](#) was brought to the CSS Working Group
- A proposal to detect the [input-capabilities of devices](#) (in particular their ability to react to touch) was brought to the Web Applications Working Group
- The Web Performance Working Group is considering the definition of a [callback mechanism for scheduling function calls](#) and has a proposed editors draft for it;
- A [charter for a Web Payments Working Group](#) has been proposed; the group would work on a browser API that would vastly simplify payments on the Web
- A [draft charter for a Hardware Security Working Group](#) is being developed; it would bring interaction with secure hardware modules to the Web platform

Published as First Public Working Draft

- The [Geofencing API](#), enabling developers to be notified when the user enters specific geographical areas, was published as a First Public Working Draft
- The [Entry Point Regulation specification](#), which provides another layer of protection against common attack vectors such as cross-site-script or cross-site request forgery, was published as First Public Working Draft
- The [Preload](#) specification, which offers a way to load stylesheets and scripts immediately, but defer their application, was published as a First Public Working Draft

Returned to Working Draft

- The [Proximity Events](#) and [Ambient Light Events](#) APIs went back to Working Draft status (respectively from Candidate Recommendation and Last Call status) based on the expectation of their significant rework to match the emerging [generic sensor API](#) work

Reached Candidate Recommendation

- The [Canvas API](#), a programmatic graphics API, re-entered Candidate Recommendation phase after it had gone back to Last Call following a substantive change
- The [Tracking Preference Expression \(DNT\) specification](#), which defines a mechanism for expressing the user's preference regarding tracking, reached Candidate Recommendation status

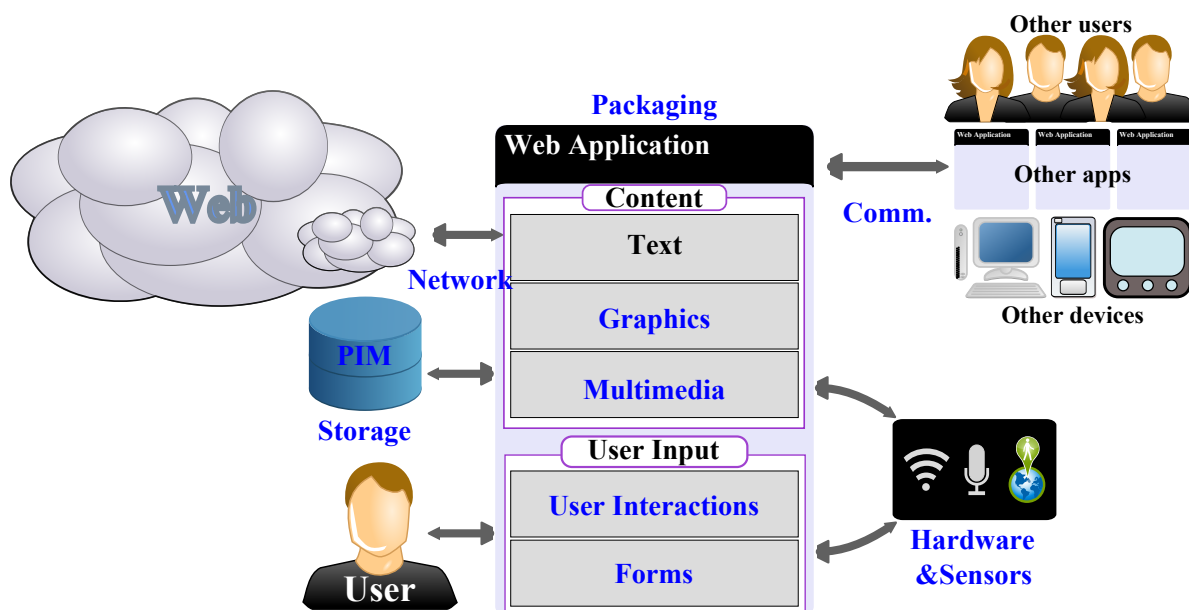
Specification merged, split or abandoned

- The [System Applications Working Group](#) closed, and work on the [Contacts Manager API](#) and the [Task Scheduler API](#) has stopped as for the other specifications from this group;

2.2. DOCUMENT STRUCTURE

The features that these technologies add to the Web platform are organized around the [application foundations for the Open Web Platform](#), a set of high-level components that application developers rely on to build their Web-based content and services.

The following application foundations are considered in this document: [core web design and development](#), [media and real-time communications](#), [usability and accessibility](#), [device interaction](#), [application lifecycle](#), [payment and services](#), [performance & tuning](#), and [security & privacy](#). In addition, it covers topics related to [network integration](#).



The Web as an application development platform

In each category of features, 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 feature, i.e. how little the author expects it to change, from an early draft that can still evolve a lot, to a finished document with only minor expected changes,
- a link to the latest editors draft of the document, and a representation of the recent editing activity; the example visualization (contra) shows a blue bar proportional to the number of commits for each of the past 12 months (here with commits in October and November 2014, then January, July and August 2015)
- some qualitative indication on availability of implementations on mobile devices, based on data collected primarily from [Can I Use...](#) and [mobile HTML5](#), completed with data from [Mozilla developer network](#), [QuirksMode](#), [JWPlayer's state of HTML5 video](#),



[Chromium Dashboard](#), [Internet Explorer Platform status](#), the [Device APIs Working Group Implementation status](#) as well as the author's understanding of the mobile devices market (see also the [code used to generate the support icons](#)). The example visualization (contra) shows that the said technology is supported in Safari from version 3.2, the Android Browser from version 4.4, Chrome on Android from version 44, Opera Mobile from version 10, Internet Explorer mobile from version 10, and Firefox on Android from version 40.

- When available, a link to a relevant tutorial on [WebPlatform Docs](#), and to relevant [on-line training courses on W3DevCampus](#)
- a link to the test suite for the said feature, and when relevant, a github ribbon to access the underlying git repository.

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” (WD) are early milestones of the Working Group progress.
-  “Last Call Working Drafts” signal that the Working Group has determined that the specification fulfills its requirements and all the known issues have been resolved, and thus requests feedback from the larger community.
-  “Candidate Recommendations” (CR) trigger a call for implementations where implementors 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” (PR) manifests that the group has gathered sufficient implementation experience, and triggers the final review by W3C Members
-  “W3C Recommendations” (Rec) 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.

For groups that have adopted it, the [2014 update of the W3C Process](#) simplifies a bit the progression by removing the Last Call stage — instead of a single global call for review addressed to the whole community, Working Groups are empowered with soliciting reviews from their various related

communities as long as they can demonstrate sufficient wide review of the specification before requesting transition to Candidate Recommendation.

Prior to starting standardization, a Working Group needs to be chartered, based on input from W3C Members, often through the organization of a [workshop](#), or after the reception of a [W3C Member Submission](#).

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.

2.3. CORE WEB DESIGN AND DEVELOPMENT

2.3.1. Graphics and Layout

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. **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 rendering graphics, but can also be used to do image processing and analysis — **HTML 5.1** adds the ability to do that **processing in a separate Web Worker**.

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 Transforms*, including with **3D effects**).

Animations can be described declaratively via *CSS Animations*, and *CSS Transitions*.

Animations can also be managed via scripting through the API exposed in **Web Animations**; as they 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.

To ensure optimal performances when animating parts of an app, authors can make use of the **CSS will-change** property to let browsers compute the animation ahead of its occurrence.

An early proposal for **Web Navigation Transitions** would enable animations that occur when navigating from one page to another.

CSS Flexbox allows to build complex layouts as required for interactive applications on small screens.

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 1.0** (*Web Open Font Format*) addresses that limitation by making it










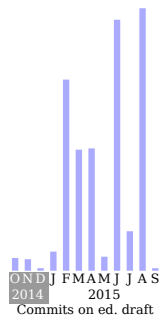
























easy to use fonts that are automatically downloaded through style sheets, while keeping the size of the downloaded fonts limited to what is actually needed to render the interface. The upcoming [WOFF 2.0](#) update to that format promises 25%-smaller download sizes; on mobile, a 35% reduction in the time needed to download and display these fonts has been measured.


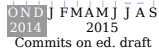



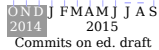
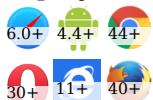


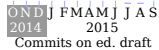



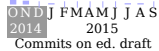


Given the time required for downloading fonts over mobile networks, authors need to adapt their content to the progressive availability of fonts; [CSS Font Loading](#) gives the necessary events to developers to enable that adaptation.

Another important aspect in graphics-intensive applications (e.g. games) is the possibility to use the entire screen to display the said graphics; the work on a [Fullscreen API](#) to request and detect full screen display, previously co-developed by the Web Applications and CSS Working Groups, has now fully moved to the [WHATWG](#).



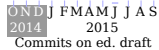




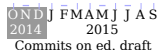


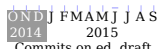




Likewise, in these scenarios, it is often useful to be able to **lock the orientation of the screen**; the [Screen Orientation API](#) allows not only to detect orientation change, but also to lock the orientation in a specific state.











NB: a [3D graphic API for HTML5 canvas](#), called [WebGL](#), has been developed 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	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
2D Vector Graphics	Scalable Vector Graphics (SVG) 1.1 Specification	SVG		Finished		Widely deployed      	 WebPlatform.org	High coverage
	Scalable Vector Graphics (SVG) 2			Early draft	Last updated September 2015  Commits on ed. draft	N/A      		N/A
2D Program... API	HTML Canvas 2D Context	HTML		Stable	Last updated December 2014  Commits on ed. draft	Widely deployed      	 WebPlatform.org 	 Good coverage
	Canvas Proxy for Web Workers in HTML 5.1			Early draft	Last updated July 2015	None      		None

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Rounded Corners	CSS Backgrounds and Borders Module Level 3	CSS		Mostly finished	Last updated August 2015 	Well deployed 	 WebPlatform.org	Good coverage
Complex background images	CSS Backgrounds and Borders Module Level 3			Mostly finished	Last updated August 2015 	Well deployed 	 WebPlatform.org	Good coverage
Box shadow effects	CSS Backgrounds and Borders Module Level 3			Mostly finished	Last updated August 2015 	Well deployed 	 WebPlatform.org	Good coverage
2D Effects	CSS Transforms Module Level 1	SVG and CSS		Mostly stable	Last updated August 2015 	Well deployed 	 WebPlatform.org	Good coverage

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
3D Effects	CSS Transforms Module Level 1			Stabilizing	Last updated August 2015 	Well deployed 		Good coverage
Animations	CSS Animations	CSS		Early draft	Last updated August 2015 	Well deployed 		well started
	CSS Transitions			Early draft	Last updated August 2015 	Well deployed 		well started
	Web Animations	SVG and CSS		Early draft	Last updated August 2015 	Limited 		Well started

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
	Timing control for script-based animations	Web Performance		Stable	Last updated June 2015  	Well deployed 		 Well started
	CSS Will Change Module Level 1	CSS		Early draft	Last updated July 2015  	Growing deployment 		N/A
	Web Navigation Transitions		Not on standards track yet	Early proposal	Last updated June 2015  	None		N/A
Complex layouts	CSS Flexible Box Layout Module Level 1			Mostly finished	Last updated September 2015  	Well deployed 		Good coverage

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Download... fonts	WOFF File Format 1.0	WebFonts		Finished		Good deployment 5.0+ 4.4+ 44+ 11+ 10+ 40+	 WebPlatform.org	Good coverage
	WOFF File Format 2.0			Early draft	Last updated September 2015 	Growing X 44+ 44+ 30+ X 40+		N/A
	CSS Font Loading Module Level 3	CSS		Stabilizing	Last updated September 2015 	Growing X 44+ 44+ 30+ X X		N/A
Orientation Lock	The Screen Orientation API	Web Applications		Early draft	Last updated June 2015  	Growing deployment X X 44+ 30+ 11+ 40+		 Well started

2.3.2. 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 and Streams](#) API exposes some specific information on capabilities of 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 — to which [Media Queries Level 4](#) proposes to add the availability and type of a pointing device, the ability to hover over elements, and the ambient luminosity. It also lets developers the ability to define media queries that [react to script-defined variables](#), making it easier to integrate these adaptation rules in the overall logic of the app.











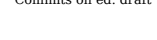

[CSS Device Adaptation](#) defines a set of CSS directives to define the size on which this layout should be based, relatively to the size of the underlying device — specifying what has been implemented using the `<meta name="viewport">` element so far.


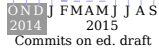



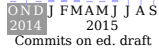



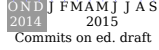







The [viewport-relative CSS units `vw` and `vh`](#) let design layouts that adapt to the dimensions of the viewport, while [CSS Mobile Text Size Adjustment](#) lets text adapt to zoomed parts of a page.




The [Responsive Images Community Group \(RICG\)](#) developed an extension to HTML, known as the [picture element](#), that allows authors define what image to load depending on device capabilities and/or other media features.

As a complementary approach, the [srcset attribute](#), specified by the [WHATWG](#) and also published as an extension to HTML, let Web developers define the various device pixel ratios of an image, letting the browser pick the best choice for the pixel density of the screen. As of January 2014, there is general agreement amongst browser vendors to implement both `picture` and `srcset`.

[SVG](#), which lets define images that can be scaled up and down without any loss of quality, is another critical tool to the development of Web applications that adapt to the resolution of the underlying device.

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Device information	Device Description Repository Simple API	Device Description		finished		Limited		Good Coverage
Media Capture Capabilities	Source capabilities in Media Capture and Streams	Device APIs and Web Real-Time Communic...		Stabilizing	Last updated August 2015  	None 		None
CSS-based adaptation	Media Queries	CSS		Finished		Widely deployed 		Good coverage
	Media Queries Level 4			Early draft	Last updated May 2015  	None 		None





















Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
	CSS Device Adaptation			Early draft	Last updated May 2015  Commits on ed. draft	Limited 		N/A
	viewport relative units in CSS Values and Units Module Level 3			Mostly finished	Last updated June 2015  Commits on ed. draft	Well deployed 	 Good coverage	
	CSS Mobile Text Size Adjustment Module Level 1			Early draft	Last updated July 2015  Commits on ed. draft	Growing deployment 	N/A	
Responsive images	picture element in HTML 5.1	HTML		stabilizing	Last updated July 2015	Growing 		 Started
	srcset attribute in HTML 5.1			Stabilizing	Last updated July 2015	Well deployed 		 Well started


























Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
	Scalable Vector Graphics (SVG) 1.1 Specification	SVG		Finished		Widely deployed 	 WebPlatform.org	High coverage

2.3.3. 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 **inputmode** attribute (proposed in HTML 5.1) defines the type of textual input expected in a text entry;
- 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 `<datalist>` element allows creating free-text input controls coming with **pre-defined values** the user can select from; HTML 5.1 defines a mechanism for the **autocomplete attribute** to automatically fill input fields based on **well-known data** for the user.

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Date and time entries	Data and Time input element in HTML5	HTML		Stable	Finished	Well deployed 	 	 Just started
Customized text entries (tel, email, url)	telephone, email and URL input element in HTML5			Stable	Finished	Well deployed 	 	 Just started
Input modality	inputmode attribute in HTML 5.1			Early draft	Last updated July 2015	None 	 	 None
Input pattern	pattern attribute in HTML5			Stable	Finished	Limited but growing 	 	 Just started

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Input hint	placeholder attribute in HTML5			Stable	Finished	Well deployed      	 WebPlatform.org 	 Started
Autocomp... for text entries	datalist element in HTML5			Stable	Finished	Growing      		 Started
	autocompl... attribute in HTML 5.1			Early draft	Last updated July 2015	Very limited      		None

2.3.4. 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 provides the **File Reader API** makes it possible to load the content of a file.

Discussions have started on a new proposal for a [sandboxed filesystem API](#).

Meanwhile, the [HTML5 download attribute](#) provides a simple mechanism to trigger a file download (rather than a page navigation), with the possibility of setting a user-friendly filename.

On top of this file-based access, the **Indexed Database API** (IndexedDB) defines a database of values and hierarchical objects that integrates naturally with JavaScript, and can be queried and updated very efficiently - a [new second edition of the specification](#) is under development. 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 Management API** will offer to Web applications.











Communication applications can benefit from integrating with their users' existing data records; on mobile devices, the address book is a particularly useful source of information.


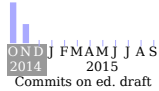




For Web apps outside of the browser, a purely programmatic approach was part of the [System Applications Working Group](#); since this group has now closed, no further work on the [Contacts Manager API](#) is expected for the time being.

In the browser, HTML 5.1 provides [autocompleted fields for contacts information](#) that would let browsers re-use data from addressbooks.

Some of this data need to be encrypted, the [Web Cryptography API](#) from the [Web Cryptography Working Group](#) exposes strong cryptography primitives to Web applications, and can be bound to pre-provisioned keys via the [WebCrypto Key Discovery API](#).

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Simple data storage	Web Storage	Web Applications		Finished	Finished	Well deployed 3.2+ 2.1+ 44+ 11+ 10+ 40+	 WebPlatform.org	 Complete
File operations	File API			Stabilizing	Last updated April 2015  O N D J F M A M J J A S 2014 2015 Commits on ed. draft	Getting well deployed 6.0+ 3+ 44+ 11.1+ 10+ 40+		 Started
	FileSystem API		N/A	Early proposal	Last updated July 2015  O N D J F M A M J J A S 2014 2015 Commits on ed. draft	N/A		None
	download attribute in HTML5	HTML		Stable	Finished	Growing 4.4+ 44+ 30+ X 40+		 None
Database query/update	Indexed Database API	Web Applications		Stable	Finished	Well deployed 8+ 4.4+ 44+ 30+ 10+ 40+	 WebPlatform.org 	 Good coverage

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
	Web SQL Database		Retired	Abandoned		Somewhat deployed, but won't be further deployed 	 	N/A
Quota for storage	Quota Management API			Early work	Last updated April 2015   Commits on ed. draft	Very limited 		None
Addressb... data	Contacts Manager API	System Applications	Retired	Abandoned	Last updated April 2014	None 		None
	autocompl... attribute in HTML 5.1	HTML		Early draft	Last updated July 2015	Very limited 		None

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Encrypted storage	Web Cryptogra... API	Web Cryptogra...		Stable	Last updated November 2014 	Well deployed 		 Early start
	WebCrypto Key Discovery			Early work	Last updated May 2014	None 		None

2.4. MEDIA AND REAL-TIME COMMUNICATIONS

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 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 20 years.

The playback content can be streamed, augmented and completed via [Media Source Extensions](#) that lets developers buffer and generate media content in JavaScript.

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](#).

Mobile devices often expose shortcuts to handle the audio output of a main application (e.g. a music player) from a lock screen or the notification areas. The WHATWG [Media Session](#) specification proposes deeper integration with these features in Web applications.

Because mobile devices often come with a variety of configurable audio output (phone speaker, loudspeaker, headset, Bluetooth speakers), the [Audio Output Devices API](#) let developers set on which output devices a given audio resource should play.

While the new HTML5 tags allow to play 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**, as well as an API to record these streams into files, and another API to use access to cameras to [take photos programatically](#).

Beyond capturing and 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](#).

The [Network Service Discovery](#) 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.

An alternative proposal to the Network Service Discovery API has emerged: [Named Web Sockets](#) offers to provide well-known sockets to existing and approved local network services.


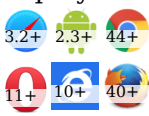




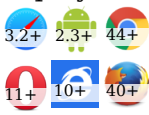





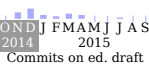
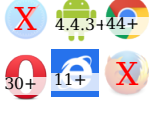






The [Second Screen Presentation Working Group](#) is building an [API to request display of content on an external screen](#), including through network-based protocols, forming the basis for second-screen scenarios.

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








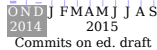





- **Peer-to-peer connection** across devices,
- **P2P Audio and video streams** allowing for real-time communications between users.

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.



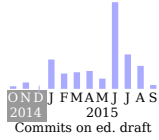


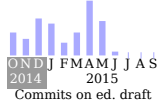







Likewise, the opportunity offered to deploy real-time communication services on the Web is another step towards the unification of the telecommunication and Web ecosystems.

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Video playback	video element in HTML5	HTML		Stable	Finished	Good deployment 	 	 Well started
Audio playback	audio element in HTML5			Stable	Finished	Good deployment 	 	 Started
Generation of media content	Media Source Extensions			Stable	Last updated July 2015  	Well deployed 		 Well started
Protected content playback	Encrypted Media Extensions			Early draft	Last updated August 2015  	Limited 		None

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Media focus	Media Session	WHATWG	N/A	Early draft	Last updated September 2015 	None 		N/A
Audio output selection	Audio Output Devices API	Device APIs and Web Real-Time Communic...		early draft	Last updated February 2015 	None 		None
Capturing audio/video	HTML Media Capture	Device APIs		Stable	Last updated October 2014 	Growing deployment 		 Good coverage

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
	Media Capture and Streams	Device APIs and Web Real-Time Communic...		Stabilizing	Last updated August 2015  	Growing 		 started
	MediaStre... Recording			Early draft	Last updated December 2014  	Very limited 		None
	Mediastream Image Capture			Early draft	Last updated January 2015  	None 		None

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Image & Video analysis, modificati...	HTML Canvas 2D Context	HTML		Stable	Last updated December 2014 	Widely deployed 	 	Good coverage
Audio analysis, modificati...	Web Audio API	Audio		Starting to stabilize	Last updated August 2015 	Good deployment 		Started
Local network services	Network Service Discovery	Device APIs		Early draft, unsure future	Last updated February 2014	None 		None
	Named Web Sockets		N/A	Early proposal	Last updated December 2014 	None 		None

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
	Presentation API	Second Screen Presentation		Early draft	Last updated September 2015  	None		N/A
P2P connectio... and audio/video streams	WebRTC 1.0: Real-time Communic... Between Browsers	Web Real-Time Communic...		Early draft	Last updated June 2015  	Growing     	 WebPlatform.org	 Early start

2.5. USABILITY AND ACCESSIBILITY

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.

Meanwhile, the [Pointer Events Working Group](#) has made good progress on an alternative approach to handle user input, [Pointer Events](#), that allows to handle mouse, touch and pen events under a single model. It provides a complementary and more unified approach to the currently more widely deployed Touch Events.

In particular, the [CSS property `touch-action`](#) that lets filter gesture events on elements is gaining traction beyond implementations of Pointer Events.

The early proposal for an [InputDevice capabilities API](#) would provide information about a given “mouse” event comes from a touch-capable device.

As more and more content gets rendered as long scrollable lists, more and more logic is attached to scrolling events, and the quality of the user experience of these actions is highly dependent on their performances. The [CSSOM View Module](#) determines when scrolling events get fired, and let developers specify the type of scrolling behavior they want.

The proposed work on [CSS Scroll Snap Points](#) adds greater ability to control the behavior of panning and scrolling by defining points to which an app view would snap when the user moves through the page.

The [CSS `will-change`](#) property is also available to indicate to browsers that a given part of the page will be soon scrolled to and should be pre-rendered.

Many mobile devices use on-screen keyboards to let users type; the [Input Method Editor \(IME\) API](#) makes it possible to coordinate the interactions between that on-screen keyboard and Web applications, but the [future of that API is under discussion](#) given its relative lack of adoption.

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. The [IndieUI Events](#)

specification, developed by the [Indie UI Working Group](#), aims at addressing this need.

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 enables that feature in the Web environment, while the [Push API](#) makes it possible for server-side notifications to alert the user, even if the browser is not running.










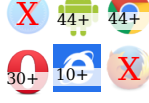



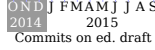

Mobile devices, and mobile phones in particular, are also in many cases well-suited to be used through voice-interactions; the [Speech API Community Group](#) is exploring the opportunity of starting standardization work around a [JavaScript API](#) that would make it possible for users to interact with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





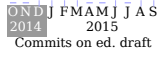


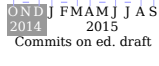



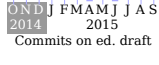




Whether users are speaking commands to their apps or working with them through non-haptic interactions, they risk seeing the screens turned off automatically by their devices screensaver. An early proposal for a [Wake Lock API](#) would let developers signal the needs to keep the screen up in these circumstances.

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 (as detailed in [Relationship between Mobile Web Best Practices and WCAG](#)).











The WCAG and UAWG Working Group provide guidance on mobile accessibility in [how Web Content Accessibility Guidelines \(WCAG\) and other WAI guidelines apply to mobile](#) — that is, making websites and applications more accessible to people with disabilities when they are using mobile phones and a broad range of other devices.

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

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Touch-based interactio...	Touch Events	Web Events		Finished	Finished	Largely deployed 	 	Complete
	Pointer Events	Pointer Events		Stable	Finished	Limited deployment 		 Well started
	touch-action property in Pointer Events			Stable	Finished	Growing 		 Good coverage
	InputDevic...		Not on standards track yet	Early proposal	Last updated August 2015   Commits on ed. draft	None		 Started

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Smooth scrolling	smooth scrolling in CSSOM View Module	CSS		Still changing	Last updated August 2015 	None 		N/A
	CSS Scroll Snap Points Module Level 1			Early draft	Last updated July 2015 	Limited 		None
	CSS Will Change Module Level 1			Early draft	Last updated July 2015 	Growing deployment 		N/A
On-screen keyboard interactio...	Input Method Editor API	Web Applications		Uncertainty about future	Last updated June 2015  	Very limited 		N/A
Vibration	Vibration API	Device APIs		Finished	Finished	Growing 		 Good coverage

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Intent-based events	IndieUI: Events 1.0	Independent User Interface (Indie UI)		Early draft	Last updated June 2014	None 		None
Notification	Web Notifications	Web Notification		Stable	Last updated January 2015 ONDJ FMAMJJAS 2014 2015 Commits on ed. draft	Growing deployment X 4.4+ 44+ 30+ X 40+		Good coverage
	Push API	Web Applications		Early draft, now with Service Workers	Last updated August 2015 ONDJ FMAMJJAS 2014 2015 Commits on ed. draft	Limited X X 42+ X X X		N/A
Speech-based interactio...	Web Speech API	Speech API Community Group	N/A	N/A	Last updated January 2013	Limited X X 44+ X X X		N/A
Screen wake	Wake Lock API	Device APIs		Just started	Last updated July 2015 ONDJ FMAMJJAS 2014 2015 Commits on ed. draft	None X X X X X X		N/A

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Accessibil...	Relationship between Mobile Web Best Practices (MWBP) and Web Content Accessibility Guidelines (WCAG)	Mobile Web Best Practices and Education and Outreach	NOTE	Finished		N/A		N/A
	Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to Mobile	Web Content Accessibility Guidelines and User Agent Accessibility Guidelines		Early draft	Last updated June 2015   O N D J F M A M J J A S 2014 2015 Commits on ed. draft	N/A		N/A
	Accessible Rich Internet Applications (WAI-ARIA) 1.0	Protocols and Formats		Stable		Well deployed  3.2+  4.4+  4.4+  10+  10+  40+		Good coverage

2.6. DEVICE INTERACTION

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 towards a new **geofencing API** (i.e. an API to detect when a device enters a given geographical area) has started.


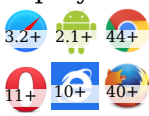














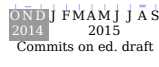


Web applications can also now access **orientation and acceleration** data via the *DeviceOrientation Event Specification*.
















A number of APIs for other sensors are under development: the **Battery Status API**, the *Proximity Events API*, the *Ambient Light Events API* or the proposed *Ambient Humidity Events API*. The Device APIs Working Group has started an effort to propose a **unification pattern for these various sensors**.










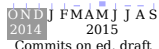














As already mentioned in the section on **multimedia**, there is ongoing work on **APIs to open up access to camera and microphone streams**.

The **NFC Working Group** had been chartered to develop APIs for accessing from **Near-Field Communications** systems to Web runtimes, but outside the traditional Web security model; as the need and requirements to bring NFC interactions under that model have become clearer, the existing NFC Working Group has been closed, and work on **browser-compatible NFC API** has started in the **Web NFC Community Group**.

A **Web Bluetooth Community Group** was started to develop a **Bluetooth API for browsers** with a particular goal of supporting Bluetooth Low Energy devices.

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Geolocation	Geolocation API Specification	Geolocation		Finished	Finished	Widely deployed 	 	 Good coverage
	Geofencing API			Just started	Last updated June 2015   Commits on ed. draft	None 		None
Motion sensors	DeviceOri... Event Specification	Device APIs		Stabilizing, but with planned updates	Last updated August 2014 	Well deployed 	 	Started
Battery Status	Battery Status API			Stable	Last updated August 2015   Commits on ed. draft	Growing 		 Good coverage

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Proximity sensors	Proximity Events			Likely to evolve substantially	Last updated September 2015  	Very limited 		 Started
Ambient Light sensor	Ambient Light Events			Likely to evolve significantly	Last updated September 2015  	Very limited 		 Started
Humidity sensor	Ambient Humidity Events		N/A	Unofficial draft	Last updated October 2013 	None 		N/A
Generic Sensors	Generic Sensor API			Early draft	Last updated June 2015  	N/A		N/A

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Camera & Micropho... streams	Media Capture and Streams	Device APIs and Web Real-Time Communic...		Stabilizing	Last updated August 2015  	Growing  44+  44+  12+  40+		 started
NFC	Web NFC API	Web NFC Community Group	Not on standards track	Early draft	Last updated August 2015  	None      		N/A
Bluetooth	Web Bluetooth	Web Bluetooth Community Group	Not on standards track	Early draft	Last updated August 2015  	Experimen...      		N/A

2.7. NETWORK INTEGRATION

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 basis for Ajax development) 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*) was meant to document 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), but that [work is now likely to be done only in the WHATWG](#). The WHATWG [fetch API](#) also provides a more powerful Promise-based alternative.

The **Beacon** API aims at letting developers queue unsupervised HTTP requests, leaving it to the browser to execute them when appropriate, opening the door for better network optimizations.

Early work on a **Web Background Synchronization API** would provide a robust Service Worker-based mechanism to enable Web applications to download and upload content in the background, even in the absence of a running browser.

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 triggering 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. An [IETF Working Group charter](#) is under discussion to standardize the protocol aspects of the mechanism.




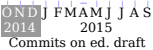







The **WebSocket API**, built on top of the IETF [WebSocket protocol](#), offers a bidirectional, more flexible, and less resource intensive network connectivity than XMLHttpRequest.












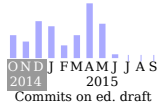






The work on [Web Real-Time Communications](#) will also provide direct **peer-to-peer data connections** between browsers with real-time characteristics, opening the way to collaborative multi-devices Web applications.



















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](#), which was supposed to address discovery of the network characteristics, has been abandoned for the time being due to lack of clear supporting [use cases](#).

The [Resource Timing API](#) offers to measure precisely the impact of the network on the time needed to load various resources, offering another approach to adapt a Web app to its network environment.

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
HTTP(s) network API	XMLHttpRequest Level 1	Web Applications		Likely to be abandoned in favor of WHATWG specification	Last updated May 2014	Well deployed 8+ 4.4.3+44+ 12+ 10+ 40+		 Good coverage
	Beacon	Web Performance		Stabilizing	Last updated April 2015  Commits on ed. draft	Limited but growing X 44+ 44+ 30+ X 40+		N/A
	Web Background Synchroniz...	Web Applications	N/A	Early draft	Last updated April 2015  Commits on ed. draft	None X X X X X X		None
Cross-domain requests	Cross-Origin Resource Sharing	Web Applications and Web Application Security		Stable		Well-deployed 6.0+ 4.4+ 44+ 12+ 11+ 40+	 WebPlatform.org	 Well started
Server-pushed requests	Server-Sent Events	Web Applications		Stable	Finished	Getting well-deployed 4.0+ 4.4+ 44+ 11.1+ X 40+	 WebPlatform.org	 Good coverage

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
	Push API			Early draft, now with Service Workers	Last updated August 2015  	Limited 		N/A
Bidirectio... connectio...	The WebSocket API			Stable	Last updated June 2014	Good deployment 	 	 Good coverage
P2P data connectio...	WebRTC 1.0: Real-time Communic... Between Browsers	Web Real-Time Communic...		Early draft	Last updated June 2015  	Growing 		 Early start
on-line state	onLine state in HTML5	HTML		Stable	Finished	Limited 		 Well started

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Network character...	The Network Information API	Device APIs	Retired	Abandoned for now, but might be restarted	Last updated November 2014   Commits on ed. draft	Limited      		None
	Resource Timing	Web Performance		Stable	Last updated August 2015   Commits on ed. draft	Growing      		 Well started

2.8. APPLICATION LIFECYCLE

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 its ability to get started based on external notifications.

These notions are part of the overall *application lifecycle*: how applications get installed, shown to the user in applications list, started, stopped, woken up from remote notifications, synced up when the device goes on-line.

These various capabilities are brought the Web platform through different mechanisms.

Although the notion of installed Web applications is still not well-defined, there are several components to the notion of installation that are under development.

[Packaging on the Web](#) describes a Web-adapted format to make Web content available in a single file for ease of download, sharing or archiving.

Whether packaged or not, users rely on a variety of metadata (name, icons) to identify the apps they want to use among their list of regularly used applications. The [JSON-based manifest format](#) lets developers group all these metadata in a single JSON file.

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.

While relatively well deployed, the current approach has shown some strong limitations in terms of how much developers can control what gets cached when. The Web Applications Working Group has thus been developing a more powerful approach, [ServiceWorker](#).

Not only does Service Worker enables Web applications to work seamlessly off-line or in poor network conditions, it also creates a model for Web applications to operate when they have not been opened in a browser window, or even if the browser itself is not running.

That ability opens the door for Web applications that run in the background and can react to remotely triggered events.
















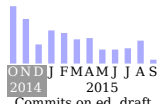


The [Task Scheduler API](#) makes it possible to trigger a task at a specified time via the Web app service worker. While the System Applications Working Group in which this API was developed has closed, the ServiceWorker-based approach taken in the specifications may make it an interesting starting point for further work in this space.




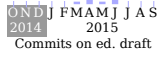






Similarly, the new [geofencing API](#) enables to wake up a Web app when a device enters a specified geographical area.




The [Push API](#) enables Web applications to subscribe to remote notifications that, upon reception, wake them up. Native applications have long enjoyed the benefits of greater user engagement that these notifications bring, and soon Web applications will share that ability.

Likewise, the [Web Background Synchronization](#) specification will enable Web applications to keep their user data up to date seamlessly, by running network operations in the background.

The [Page Visibility](#) specification lets developers detect when their application is in the foreground, and thus adapt their operations and resource consumption accordingly.

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Packaging	Packaging on the Web	TAG and Web Applications		Early draft	Last updated February 2015  	None 		N/A
	Manifest for a web application	Web Applications		Early draft	Last updated August 2015  	Limited 		N/A
Offline Web Apps	Applicatio... in HTML5	HTML		Stable (but Service Workers will be the preferred approach when available)	Finished	Well deployed 	 	 None
	Service Workers	Web Applications		Early draft	Last updated September 2015  	Limited 		 Well started

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Scheduled tasks	Task Scheduler API Specification	System Applications	Retired	Early draft	Last updated October 2014 	None 		None
Geofencing	Geofencing API	Geolocation		Just started	Last updated June 2015 	None 		None
Remote Notificati...	Push API	Web Applications		Early draft, now with Service Workers	Last updated August 2015 	Limited 		N/A
Backgrou... Sync	Web Background Synchroniz...		N/A	Early draft	Last updated April 2015 	None 		None

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Foreground detection	Page Visibility	Web Performance		Finished		Well deployed 		 Good coverage

2.9. PAYMENT AND SERVICES



Native mobile application stores have made it much easier for developers to monetize some of their applications, either by selling the application itself to users, or by providing in-app purchases.

While Web applications can use well-known on-line payment solutions, these solutions have so far proved much harder to use on mobile devices.

In March 2014, W3C organized a [workshop on Web payments](#) to identify ways in which standards could help make that payment experience much simpler, in particular on mobile devices. In October 2014, W3C [chartered](#) the [Web Payment Interest Group](#) to drive work in this space, and that group released [its sense of priority use cases for Web payments](#).

A [charter for a Web Payments Working Group](#) is now under review by the W3C Advisory Committee: this group, if chartered, would develop a browser API to facilitate payment operations in Web apps.

Meanwhile, HTML5.1 provides specific help for [autocomplete of credit card details](#), making it easier to pay via credit cards once these details have been entered once.

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Integrated payment	Credit card details autocompl... in HTML 5.1	HTML		Early draft	Last updated July 2015	Very limited 		None

2.10. PERFORMANCE & TUNING

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](#), [Performance Timeline](#) and [User Timing](#), gives tools to Web developers for optimizing their Web applications. The work on the [Frame Timing](#) API aims at providing detailed information on the frame-per-second obtained when an application is running on the user device.

The [Resource Hints](#) and [Preload](#) specifications let developers optimize the download of resources by enabling to delay either the download or the execution of the downloaded resource.

The proposed work on [Efficient Script Yielding](#) offers the opportunity to Web developers to use more efficiently asynchronous programming, but has so far gained very limited traction.












































The [requestIdleCallback](#) API similarly proposes a way for scheduling an operation at the next opportunity when the app is not processing another operation.

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.











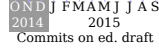


















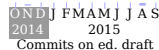









Beyond optimization of resources, the perceived reactivity of an application is also a critical aspect of the mobile user experience. The **thread-like mechanism** made possible via [Web Workers](#) allows keeping the user interface responsive by offloading the most resource-intensive operations into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

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

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. The opportunity to update these best practices is [under discussion](#) in the [Web and Mobile Interest Group](#).

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Timing hooks	Navigation Timing	Web Performance		Finished		Well deployed      		 Good coverage
	Resource Timing			Stable	Last updated August 2015  	Growing      		 Well started
	Performance Timeline			Finished		Limited      		 Started
	User Timing			Finished		Growing      		 Well started
	Frame Timing			Early draft	Last updated June 2015  	None      		None

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Network prioritizat...	Resource Hints			Early draft	Last updated August 2015 	Growing deployment 		None
	Preload			Early draft	Last updated September 2015 	None?		None
Priority handling	Efficient Script Yielding			Early draft	Last updated April 2014 	Very limited 		None
	Cooperative Scheduling of Background Tasks			Early draft	Last updated August 2015 	Experimen...		None

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Page Visibility detection	Page Visibility			Finished		Well deployed      		 Good coverage
Animation optimizati...	Timing control for script-based animations			Stable	Last updated June 2015   Commits on ed. draft	Well deployed      		 Well started
Threading	Web Workers	Web Applications		Stable	Last updated May 2014	Well deployed      	 WebPlatform.org 	 Good coverage
Battery Status	Battery Status API	Device APIs		Stable	Last updated August 2015  Commits on ed. draft	Growing      		 Good coverage
Optimizat... Best Practices	Mobile Web Application Best Practices	Mobile Web Best Practices		Finished		N/A		N/A

2.11. SECURITY & PRIVACY

Mobile devices follow their users everywhere, and hold some of their most private or confidential data (contacts, pictures, calendar, etc.) As a result, it is critical for users to be able to rely on their phones to keep that data safe from attackers.

W3C specifications are reviewed for their security and privacy impact as part of their progress through the Recommendation track; the [Privacy Interest Group](#) and the [Web Security Interest Group](#) in particular are coordinating reviews on their respective fields.

Many sensitive APIs are gated by a request for user consent; while these requests give control to the user, they can be sometimes hard to integrate in the overall user experience without visibility on which permission has been granted or denied. The [Permissions API](#) aims at fixing this.

But beyond these cross-technology considerations, a number of ongoing work items address needs for additional protection.

The first line of defense for users, and the unit of isolation for Web apps is the same-origin policy that roughly limits what a Web application can access to content and data hosted on the same origin, i.e. the combination of URL scheme, domain name and port.

For legacy reasons, this policy is not as stringent on some parts of the Web platform, exposing users to greater attack surface via cross-site scripting or cross-site request forgery. To enable Web application authors to reduce the attack surface beyond what legacy requires, the [Content Security Policy \(level 2\)](#) offers hooks that severely limits damages that an attacker could hope to achieve.

To further strengthen the integrity of their applications, Web developers can make use of the proposed [Subresource integrity](#) mechanism, that makes it possible to block man-in-the-middle attacks or compromised third-parties providers.

[Entry Point Regulation](#) provides another layer of strengthening and offers to filter the type of HTTP requests that can be made from external sites, reducing risks of cross-site script and cross-site request forgery.

In applications that aggregate content from multiple (possibly untrusted) sources, the [HTML5 iframe sandbox](#) makes it possible to restrict what kind of interactions third-party embedded content can make use of.

As described earlier, the [Web Cryptography API](#) provides the necessary tools to encrypt data for storage and transmission from within Web applications, with access pre-provisioned keys via the [WebCrypto Key Discovery API](#).






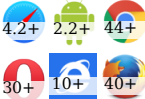



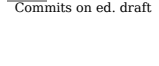






There are discussions to bring the capabilities of hardware-security modules to the Web, to enable access to high-security operations for encryption,





payment, identity proof, etc., embodied in a [draft charter for a Hardware Security Working Group](#).

For users that wish to indicate their preferences not to be tracked across Web applications and sites, the [Tracking Preference Expression \(also known as Do No Track\)](#) enables browsers to communicate explicitly their wish to content providers, and to determine whether a given content provider asserts fulfilling that wish.

To facilitate the authentication of users to on-line services, the Web Application Security Working Group is proposing a [credential management API](#) that lets developers interact more seamless with user-agent-managed credentials.

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Permission managem...	The Permissions API	Web Application Security		Early draft	Last updated August 2015 	Limited 		N/A
Strengthen security	Content Security Policy Level 2			Stable	Last updated August 2015 	Well-deployed 		Well started
	Subresource Integrity			Just started	Last updated August 2015 	Limited 		None

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
	Entry Point Regulation			Just started	Last updated June 2015  	None 		None
	Sandboxed iframe in HTML5	HTML		Stable	Finished	Widely deployed 		 None
Encryption	Web Cryptogra... API	Web Cryptogra...		Stable	Last updated November 2014  	Well deployed 		 Early start
	WebCrypto Key Discovery			Early work	Last updated May 2014	None 		None
Tracking protection	Tracking Preference Expression (DNT)	Tracking Protection		Stabilizing	Last updated June 2015	Good deployment 		None

Feature	Specificat...	Working Group	Maturity	Stability	Latest editors draft	Current implemen...	Developers doc	Test suite
Identity managem...	Credential Management Level 1	Web Application Security		Early draft	Last updated September 2015  	None 		N/A

2.12. ACKNOWLEDGMENTS

Thanks to Art Barstow, Anssi Kostianen, Jo Rabin, J. Manrique López, Mounir Lamouri, Marcos Caceres, François Daoust and Ronan Cremin for their contributions to this document.

This document is produced through the [HTML5Apps project](#), funded by the European Union through the Seventh Framework Programme (FP7/2013-2015) under grant agreement n°611327 - HTML5 Apps.