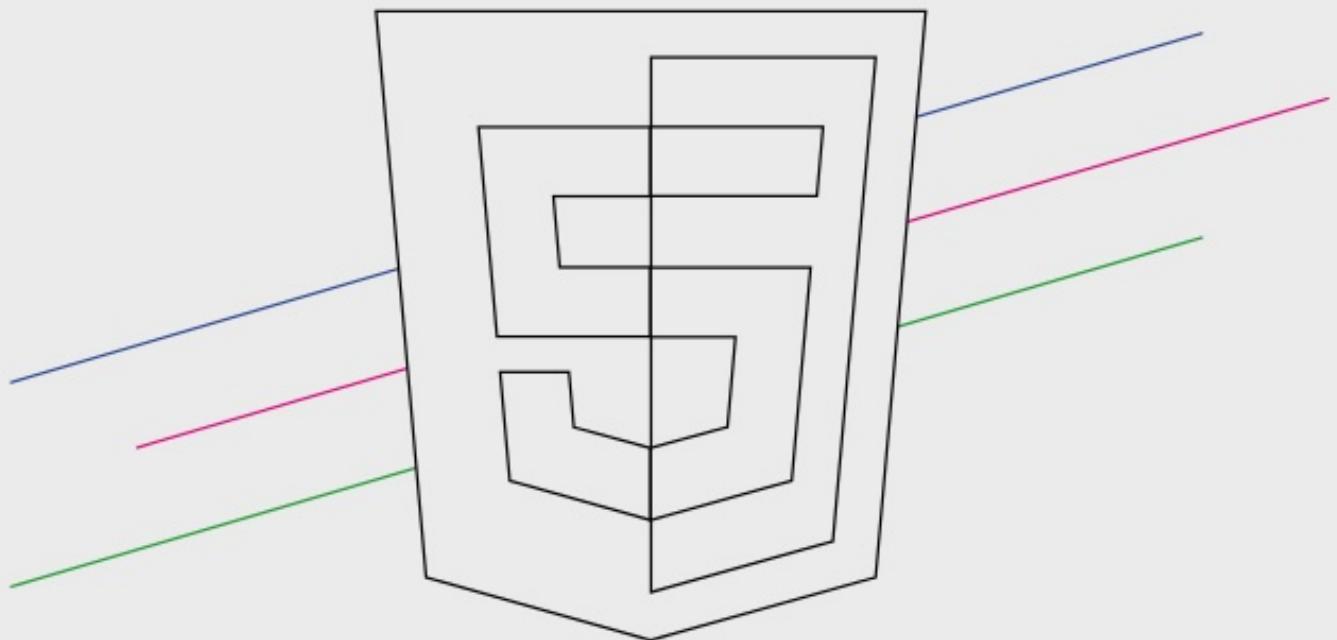


HTML



[**HTML5, how to rethink your web strategy**]

Table des matières

Introduction

Partie I

HTML5 : l'évolution du web

p4

Introduction au fonctionnement d'une page web

HTML, un langage en perpétuelle évolution

HTML5 : buzzword ou réelle disruption ?

HTML5 vs HTML4 : que faut-il retenir ?

L'état de l'art

- Quels sont les navigateurs et les OS les plus performants avec HTML5 ?
- Quels sont les navigateurs web et mobiles les plus utilisés en France ?

Partie II

Les applications natives : menacées par le web mobile ?

p20

Le modèle des applications, bien que dominant le marché du mobile...

...Est actuellement remis en question

HTML5 au service de la mobilité : forces et enjeux

- Les sites web en HTML5 peuvent faire beaucoup mieux que des applications natives
- La distribution des sites web sur mobile

Partie III

Applications mobiles vs applications natives

p25

Comparaison de leurs possibilités techniques et fonctionnelles

Deux modes de conception qui mènent à des expériences utilisateurs différentes

Grandes questions liées à la mise en œuvre d'un projet de développement

Matrice de décision : que faire dans quel cas ?

- Site Internet ou applications natives?
- Qu'en est-il des solutions hybrides ?

Partie IV

Nos recommandations

p.37

Finalement, quelle solution pour quel type d'applications ?

- Jeux et divertissements
- Médias, actualités et informations
- Réseaux sociaux
- Déplacements et cartes
- Vie pratique (réservation, e-shopping, transaction bancaire)
- Multimedia

Synthèse

Introduction

HTML 5 marque la renaissance du web : mobilité, interactivité et ouverture.

Inspirées du succès des applications natives sur appareils connectés, les nouvelles fonctionnalités d'HTML5 apportent des solutions aux développeurs pour créer plus facilement et plus rapidement des expériences plus riches et plus mobiles : tactile, mode hors-connecté, gestion de l'audio et de la vidéo, utilisation de la caméra ou du GPS de l'appareil, etc.

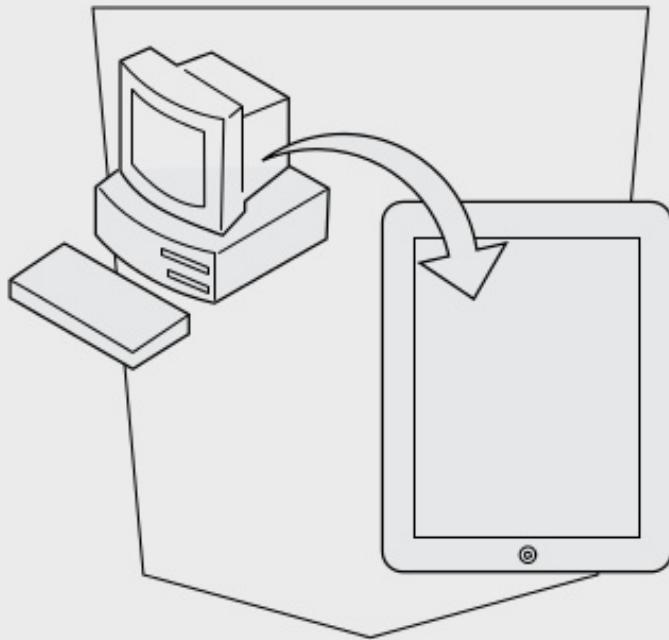
Mais la vocation d'HTML5 n'est pas seulement d'apporter une alternative aux applications natives et aux écosystèmes fermés du type iOS ou Android : il s'agit, plus largement, d'apporter des réponses à ceux qui souhaitent rationaliser leur stratégie numérique multi-plate-forme, sans pour autant déserter l'App Store ou le Google Play Store.

Le but de cette étude est de fournir aux entreprises souhaitant développer des produits digitaux :

- Un aperçu des nouvelles fonctionnalités offertes par HTML5 ;
- Une analyse simple des dernières tendances en matière de développement web, mobile, tablette et smart TV ;
- Une grille de décision qui permettra, pour chaque projet, qu'il soit mono-plate-forme ou multi-plates-formes, de déterminer la meilleure stratégie de développement possible.

NB : Les parties 1.1 et 1.2. de ce document présentent les éléments de pédagogie indispensables à la compréhension des enjeux du HTML5. Les lecteurs déjà familiers des technologies web en général, et de HTML5 en particulier, sont invités à passer directement à la partie 1.3.

[I]



HTML5 : l'évolution du web

- *Introduction au fonctionnement d'une page web*
- *HTML, un langage en perpétuelle évolution*
- *HTML5 : buzzword ou réelle disruption ?*
- *HTML5 vs HTML4 : que faut-il retenir ?*
- *L'état de l'art*

Introduction au fonctionnement d'une page web

Une **page web** est un assemblage de textes, d'images, d'objets multimédia, de fichiers et de programmes dispersés sur le web.

Les langages de programmation peuvent être séparés en 2 catégories :

Les standards du web

Le code HTML de la page définit tous les éléments de la page, indique où les trouver et décrit la façon dont ils s'agencent les uns par rapport aux autres. C'est le constituant de base qui détermine la structure de la plupart des pages web.

La feuille de style CSS de la page est un document qui définit le rendu visuel de chaque élément: police des textes, couleur des fenêtres, largeur des bordures, emplacement des éléments, etc.

JavaScript est le langage de script exécutable par un navigateur le plus répandu. C'est généralement JavaScript qui gère le comportement des différents "boutons" d'une page web, des champs de texte, etc.

Ces langages font partie des langages «de base» du web, supportés par tous les navigateurs. Ils n'appartiennent à aucune entreprise et évoluent donc au gré des usages et des besoins de la communauté de développeurs. Ce sont des organismes internationaux comme le W3C et l'ECMA qui sont chargés de fixer les normes d'écriture de ces langages et de les faire évoluer, sous forme de recommandations publiées régulièrement. **Chaque éditeur de site web et de navigateur web a la liberté de les adopter ou pas, et d'en créer d'autres ou pas.**

Souvent, on parle simplement de HTML pour désigner le trio HTML + CSS + JavaScript car ces langages sont très intimement liés. Dans la suite, nous adopterons aussi cet abus de langage.

Flash, Silverlight, etc. = les technologies propriétaires

Flash et Silverlight sont des langages de script souvent utilisés pour gérer le comportement d'éléments multimédia de types vidéo, animation, objet 3D, etc.

Ces technologies sont très normées car elles émanent d'un seul éditeur qui dicte les normes d'écriture et fournit les outils capables de les interpréter parfaitement.

Ex : **Flash** est édité par Adobe. Pour lire un script Flash, un navigateur doit forcément être équipé du **plug-in** (élément de logiciel qui se «branche» sur le navigateur pour augmenter ses capacités) **Flash Player** édité uniquement par Adobe.

Un navigateur web est un logiciel (qui peut être installé sur un ordinateur, un mobile, une smartTV, une console de jeux vidéos, etc.) capable de lire le code HTML et la feuille de style CSS pour afficher tous les éléments du site web à l'écran.

Vous utilisez certainement l'un de ces navigateurs :

- Internet Explorer
- Firefox
- Chrome
- Safari
- Opera

La plupart des sites web ne sont pas statiques : le comportement de certains éléments peut varier en fonction des actions de l'utilisateur. La façon dont ces éléments dynamiques se comportent est décrite par des **scripts**, c'est-à-dire des mini-programmes informatiques insérés dans le code HTML et exécutés directement dans le navigateur.

Une web app est un type particulier de site web qui fournit un service accessible directement dans le navigateur de l'utilisateur. Ex : webmail, moteur de recherche, wiki, blog, etc.

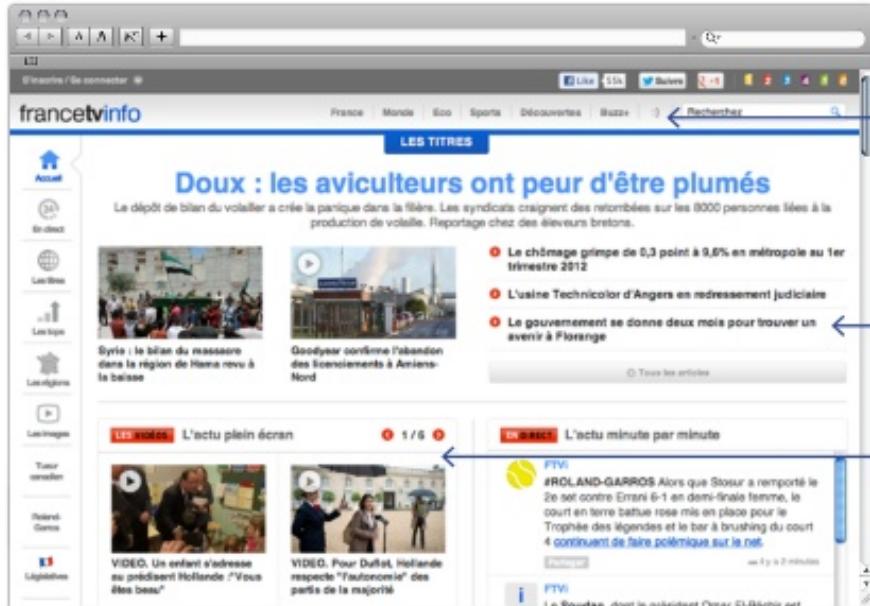


Figure 1
Structure web de la page d'accueil du site Internet de France TV Info.

La position des éléments, la couleur de fond du header, les polices des titres, etc. sont décrits dans la feuille de style CSS de la page

Les textes, menus, sections, etc. de la page sont décrits dans son code HTML.

Le comportement dynamique de la fenêtre « les vidéos » lorsque l'utilisateur clique sur les flèches de défilement est décrit par un script JavaScript.

HTML, un langage en perpétuelle évolution

La communauté de développeurs propose constamment de nouvelles fonctionnalités HTML. Avant de parvenir aux utilisateurs finaux, ces fonctionnalités passent par 4 niveaux d'adoption.



Figure 2
Principales étapes du processus d'intégration des nouvelles fonctionnalités de HTML sur le web.

Le W3C émet en permanence de nouvelles recommandations afin que les navigateurs s'harmonisent le mieux possible : il s'agit de les encourager à implémenter les nouvelles fonctionnalités et à se conformer aux mêmes normes pour faciliter le travail des développeurs de sites web.

Parfois, lorsque le W3C considère que les nouvelles fonctionnalités actuellement développées et testées par la communauté de développeurs constituent une avancée marquante pour le web, il décide d' « annoncer une nouvelle version de HTML ».

Il faut bien comprendre qu'il n'y a pas UNE version de HTML : chaque site web et chaque navigateur intègre à son rythme et selon ses propres modalités les nouvelles fonctionnalités HTML développées par la communauté de développeurs.

Aujourd'hui, la version officiellement recommandée par le W3C est HTML4. HTML5 est toujours « en cours de spécification », même si de nombreux sites web et navigateurs dans le monde supportent des fonctionnalités de HTML5.

HTML5 : buzzword ou réelle disruption ?



Les nouvelles fonctionnalités de HTML regroupées sous l'étiquette « HTML5 » (qu'elles soient déjà développées et implémentées sur les navigateurs principaux ou en cours de développement) suscitent l'engouement de la communauté de développeurs car **elles permettent de reproduire très simplement des fonctionnalités qui étaient jusqu'à présent assurées par des technologies propriétaires**. En particulier :

Figure 3
HTML5 : le futur du web ?

HTML5 vs Flash

- HTML5 permet d'intégrer des **vidéos** ou des **pistes audio** dans une page web sans passer par Flash. Les éléments multimédias peuvent ainsi être lus par un navigateur sans aucun plug-in particulier.
- HTML5 permet d'intégrer des animations et des éléments dynamiques en 2D (et bientôt en 3D) sans passer par Flash. En particulier, il est possible de créer des **jeux** entièrement en HTML5. C'est une révolution car les plupart des jeux en ligne reposent sur Flash ou d'autres technologies propriétaires.

HTML5 vs applications natives

Beaucoup de fonctionnalités d'HTML5 reproduisent le comportement d'applications natives d'appareils connectés :

- Utilisation dans la page web d'éléments de **hardware** de l'appareil: appareil photo, micro, touches de clavier, GPS, etc.
- Optimisation du comportement des éléments de la page web pour des **interfaces multitouch**
- Stockage de données sur l'appareil pour un fonctionnement **en mode hors connexion ou un enregistrement automatique des données / préférences de l'utilisateur**
- Optimisation de l'affichage de la page web en fonction des dimensions de l'écran.

C'est la raison pour laquelle de plus en plus de développeurs commencent à envisager de développer des sites web accessibles via des navigateurs mobiles plutôt que des applications natives.

HTML5 vs HTML4 : que faut-il retenir ?

Récemment, de nouvelles fonctionnalités sont apparues pour HTML : de nouvelles balises ont été disponibles, et d'autres, obsolètes, ont été supprimées par rapport à la version 4.0. Voici les principales fonctionnalités liées à la cinquième version de HTML :

La révolution Webdesign

- La mise en forme est désormais clairement le rôle de CSS dans une page web.
- **Simplification de la structure** d'une page web avec des balises toutes faites pour les titre, haut de page, pied de page, sections, etc. Cette nouvelle structure commune à tous les sites web favorise l'accessibilité et permet entre autre aux synthétiseurs vocaux de se développer et aux personnes malvoyantes de se repérer plus facilement.
- Amélioration des **possibilités graphiques** sur la page web : API « canvas », nouvelles polices de caractères plus compactes pour être utilisées sur le web, intégration d'images au format vectoriel.

Quelques exemples de nouvelles fonctionnalités disponibles depuis peu ou bientôt disponibles grâce à HTML5 :

Animations au sein d'une page

Aujourd'hui, HTML5 permet d'intégrer des animations en 2D sur une page web, sans avoir à recourir à Flash. Demain, HTML5 intégrera probablement des animations en 3D également.



Figure 4

Exemple d'animation : les traits lumineux se dessinent en suivant le trajet de la souris sur l'écran, ils bougent et s'effacent au fur et à mesure.

De nouveaux filtres ou animations CSS pour des mises en pages inédites



Figure 5

Exemple d'animation CSS : le texte s'aligne sur les contours de la voiture lorsque celle-ci est déplacée sur l'écran.

Une expérience utilisateur plus riche

- Plus d'interaction possible pour l'utilisateur : remplir un formulaire en ligne, application « drag'n'drop », chargement de fichiers depuis son propre ordinateur vers le web, géolocalisation, édition et modification de la page web ou de certaines zones de la page, surligner les mots trouvés dans la page lorsqu'on effectue une recherche.
- Intégration de rich medias : intégration de vidéo ou d'éléments audio.

Interaction avec le micro

Parlez et votre navigateur réagit. Une fonctionnalité qui permettra bientôt de faciliter la navigation sur le web pour les personnes aveugles.



Figure 6
Exemple d'interaction avec le micro : lorsqu'on fait du bruit devant l'ordinateur, les billes se mettent à sauter sur l'écran.

Interaction avec la webcam

Désormais vous pouvez prendre des photos directement au sein d'une page web. Demain, le chat vidéo pourra donc se faire en ligne, sans téléchargement d'une application supplémentaire.



Figure 7
Exemple d'interaction avec la webcam : on peut se prendre en photo et ajouter des filtres ou des effets.

Interaction avec la souris

En passant la souris sur votre écran, vous pouvez animer les objets. Une fonctionnalité qui enrichit l'expérience de navigation, la rend plus esthétique ou permet de mettre en place des interfaces de dessin sur internet.

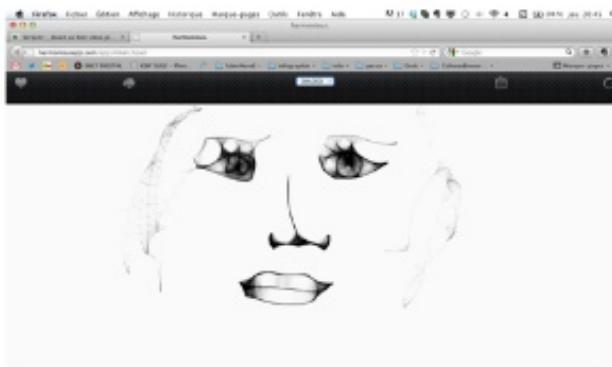


Figure 8
Dessin réalisé à l'aide de la souris au sein d'une page web.

Les jeux vidéos entièrement jouables sur le web

Il existe déjà des jeux relativement évolués accessibles directement depuis un navigateur. Bientôt une manette pourra commander directement l'action de la souris sur une page web.



Figure 9
Reprise du célèbre jeu *Snake* entièrement en HTML5

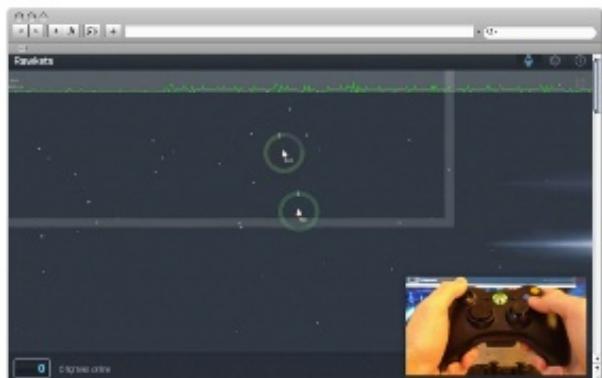


Figure 10
Illustration d'un jeu en HTML5, piloté grâce à une manette.

Une expérience mobile

Vers une utilisation mobile et performante : optimisation pour le développement responsive design, stockage de données lors de l'initialisation d'une page pour permettre de continuer la navigation même en mode non connecté.

Autres fonctionnalités

- **Détection de la situation de l'utilisateur** : est-il connecté au Wifi, à la 3G, à la 4G ? suivant ce qui est détecté, le contenu de la page s'adapte et vous offre la meilleure expérience.
- **Détection de l'action de l'utilisateur** : si je regarde une vidéo dans un onglet, la vidéo se met automatiquement sur pause si je consulte un autre onglet, et se relance lorsque je reviens sur l'onglet.
- **Full Screen** : le navigateur se met en mode « plein écran » en tappant simplement sur la touche « f ».

Ces fonctionnalités existent toutes déjà sur les navigateurs de tests (tels que Chrome Canary) ou sur les versions les plus récentes des navigateurs web. Bientôt, la majorité de la population y aura accès, ce qui permettra d'imaginer de nouvelles expériences et un nouvel usage du web.

À retenir

HTML5 n'est pas une technologie complètement nouvelle : il s'agit simplement de l'ensemble des dernières fonctionnalités de HTML, un langage qui évolue constamment. Ces dernières fonctionnalités constituent une réelle avancée pour HTML et le web en général car elles permettent de faire des sites plus beaux, plus complexes et mieux adaptés aux nouveaux supports connectés.

L'état de l'art

Quels sont les navigateurs et les OS les plus performants avec HTML5 ?

HTML 5 n'est pas également supporté par tous les navigateurs : tandis que certains intègrent les dernières innovations proposées par la communauté de développeurs, d'autres sont « en retard ».

Les chiffres qui suivent présentent un classement des différents navigateurs et de leur compatibilité avec les fonctionnalités HTML5. Le Score est obtenu sur un total 500 établi à partir des 500 fonctionnalités les plus récentes de HTML.

Navigateurs web

Les navigateurs listés ici sont des navigateurs stables couramment utilisés. HTML5 peut obtenir un score supérieur sur des versions de navigateurs instables en cours de développement, mais non utilisées par le grand public. Il faut noter la mauvaise position d'Internet Explorer, loin derrière les autres navigateurs qui intègrent plutôt bien HTML5.

	Score
Maxthon 3.4.1	422
Chrome 20	414
Opera 12.00	385
Firefox 13	345
Safari 5.1	317
Internet Explorer 9	138

Figure 11
Score (sur 500) des six navigateurs les plus compatibles avec HTML5.

Source
<http://html5test.com/>

Que se passe-t-il lorsqu'une fonctionnalité HTML (ex : une vidéo en HTML5) n'est pas reconnue par le navigateur ? 3 possibilités :

- Il ne se passe rien : la vidéo ne se lance pas.
- L'animation est dégradée : le son et l'image de la vidéo sont décalés.
- On ne s'en rend pas compte : pour contourner le problème, les développeurs ont recours à une version « fall back » de leur site. Sur les navigateurs qui liront la fonctionnalité HTML5, tout sera en HTML5 ; sur les autres, ils contourneront le problème en intégrant un morceau de Flash ou Silverlight par exemple.

OS de Smartphones

Sur un mobile, la navigation web se fait sur un navigateur qui est associé à la version de l'OS utilisé par le mobile : Safari pour iOS, Android pour Android, etc. D'autres navigateurs mobiles se sont également développés indépendamment, et s'installent petit à petit sur le marché : Opera, Firefox, etc.

Sur le mobile, les scores sont globalement moins bons qu'avec les navigateurs web classiques, car ils n'intègrent pas encore les dernières fonctionnalités de HTML5.

Opera, Firefox et Safari sont très bons, mais les autres sont encore à la traîne, notamment Android, BlackBerry et Nokia.

		Score
RIM Tablet OS 2.0	Blackberry Playbook	373
Chrome	All Android 4 devices	371
Opera Mobile 12.00	Multiple platforms	369
Firefox Mobile 10	Multiple platforms	325
iOS 5.1	Apple iPhone, iPad and iPod Touch	324
Android 4.0	Asus Transformer Prime and others	280
WebOS 3.0	HP TouchPad	217
Silk 1.0	Amazon Kindle Fire	174

Figure 12
Score (sur 500) des huit OS de smartphones qui permettent la meilleure compatibilité avec les fonctionnalités de HTML5.

Source
<http://html5test.com/>

OS de tablettes

Le marché des tablettes étant plus récent, les navigateurs proposés sont moins nombreux. On constate comme dans le mobile qu'Opera, Firefox et iOS sont plutôt en avance vis-à-vis de HTML5, tandis qu'Android a du mal à suivre les évolutions. En revanche, on remarque l'assez bonne performance de RIM Tablet de BlackBerry.

		Score
Chrome	All Android 4 devices	371
Opera Mobile 12.00	Multiple platforms	369
Firefox Mobile 10	Multiple platforms	325
iOS 5.1	Apple iPhone, iPad and iPod Touch	324
MeeGo/Harmattan	Nokia N9 and N950	284
Android 4.0	Samsung Galaxy Nexus	280
Blackberry OS7	Blackberry Bold 9900 and others	273
Bada 2.0	Samsung Wave and others	268
Nokia Belle FP1	S60 5.4 Nokia 603, 700 and 701	226
webOS 2.2	Palm Pre 2 and HP Pre 3	210
Android 2.3	Google Nexus S and others	189
Windows Phone 7.5	Samsung Omnia, W, LG E906 and others	138

Figure 13
Score (sur 500) des douze OS de tablettes qui permettent la meilleure compatibilité avec les fonctionnalités de HTML5.

Source
<http://html5test.com/>

Comme nous l'avons vu, HTML évolue constamment et les différents navigateurs tentent de s'adapter à ces changements.

Si tous ne sont pas aujourd'hui au même point, tous font des efforts pour suivre la norme du W3C : les navigateurs libres tels que Chrome ou Firefox ont attiré plus de développeurs HTML, ils sont aussi ceux qui mettent à disposition des chercheurs des versions instables de leur navigateur (Firefox 14, Chrome Canary) pour leur permettre de tester de nouvelles fonctionnalités HTML. Ils sont donc mieux placés qu'Internet Explorer, mais tout cela évolue très rapidement.

Smart TV

Enfin, sur le marché des Smart TV, les navigateurs sont encore un peu moins bons que sur mobile ou tablettes, car c'est un marché émergeant, qui touche encore peu d'utilisateurs. Les scores sont donc plus serrés, avec 4 leaders au-dessus de 300 (60% de fonctionnalités de HTML5 compatibles) : Sharp Aquos, Google TV, Philips NetTV et Toshiba. Samsung, LG et Sony suivent de peu.

			Score
Sharp Aquos	Espial 6.05	Sharp Aquos televisions	342
Google TV		Sony Internet TV, LOGitech Revue and others	328
Philips NetTV	Opera Devices 3.2	Philips televisions	325
Toshiba	Espial 6.04	Toshiba L7200 televisions	310
Samsung Smart TV 2012		Samsung televisions	286
LG NetCast 2012		LG televisions	282
Sony Internet TV	Opera Devices 3.1	Sony televisions and Bluray players	275
Sharp Aquos	Opera Devices 3.0	Sharp Aquos televisions	236
Boxee		Boxee Box by D-Link, Iomega TV wit...	214
Panasonic Smart Viera		Panasonic Viera televisions	214

Figure 14
Score (sur 500) des 10 OS de Smart TV qui permettent la meilleure compatibilité avec les fonctionnalités de HTML5.

Source
<http://html5test.com>

Note : Une liste exhaustive des navigateurs web et mobile est disponible en suivant le lien : http://fr.wikipedia.org/wiki/Liste_des_navigateurs_Webdes

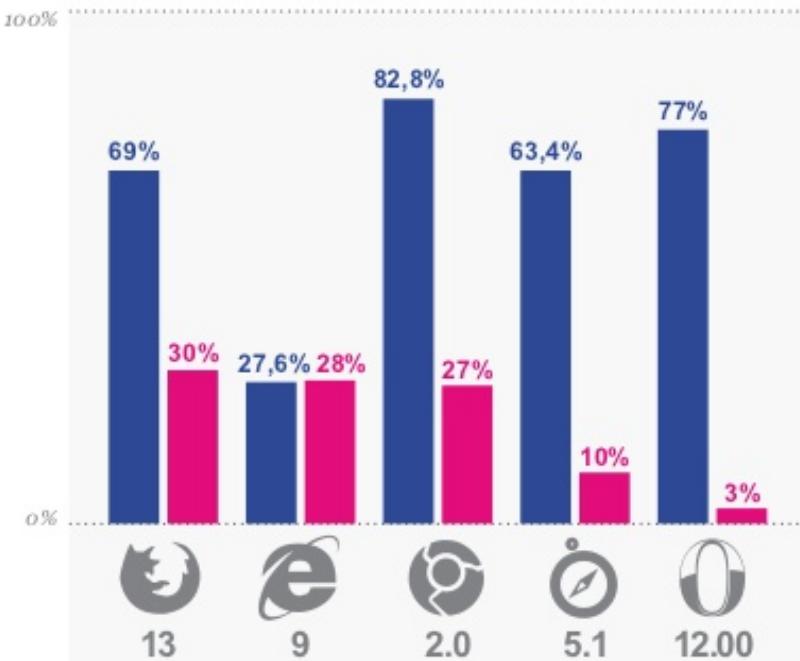
Quels sont les navigateurs web et mobiles les plus utilisés en France ?

L'accès des français aux dernières fonctionnalités de HTML5 est encore faible.

Navigateurs web

En France, seuls 50,6% des internautes ont un navigateur pour ordinateur de bureau qui intègre à plus de 80% les fonctionnalités actuelles d'HTML5. Cela représente la part d'utilisateurs de Chrome et Firefox. Cependant, encore 27% des utilisateurs Français et Européens utilisent Internet Explorer 8 ou 9, qui n'intègre qu'à 22% et 47% les fonctionnalités actuelles d'HTML5.

Pourcentage de compatibilité des navigateurs avec HTML5
Pourcentage d'utilisation des navigateurs en France en Juillet 2012



Trois géants des navigateurs web : Firefox, Chrome et Internet Explorer.

Internet Explorer est encore largement utilisé dans la version 8.0 qui n'intègre que 22% des fonctionnalités actuelles de HTML5.

Safari (le navigateur web installé par défaut sur les ordinateurs Apple) peine à décoller mais se situe tout de même juste derrière les trois géants.

Figure 15
Navigateurs web les plus utilisés en France et leur pourcentage de compatibilité avec HTML5.

Source
<http://gs.statecounter.com/>

Navigateurs mobiles

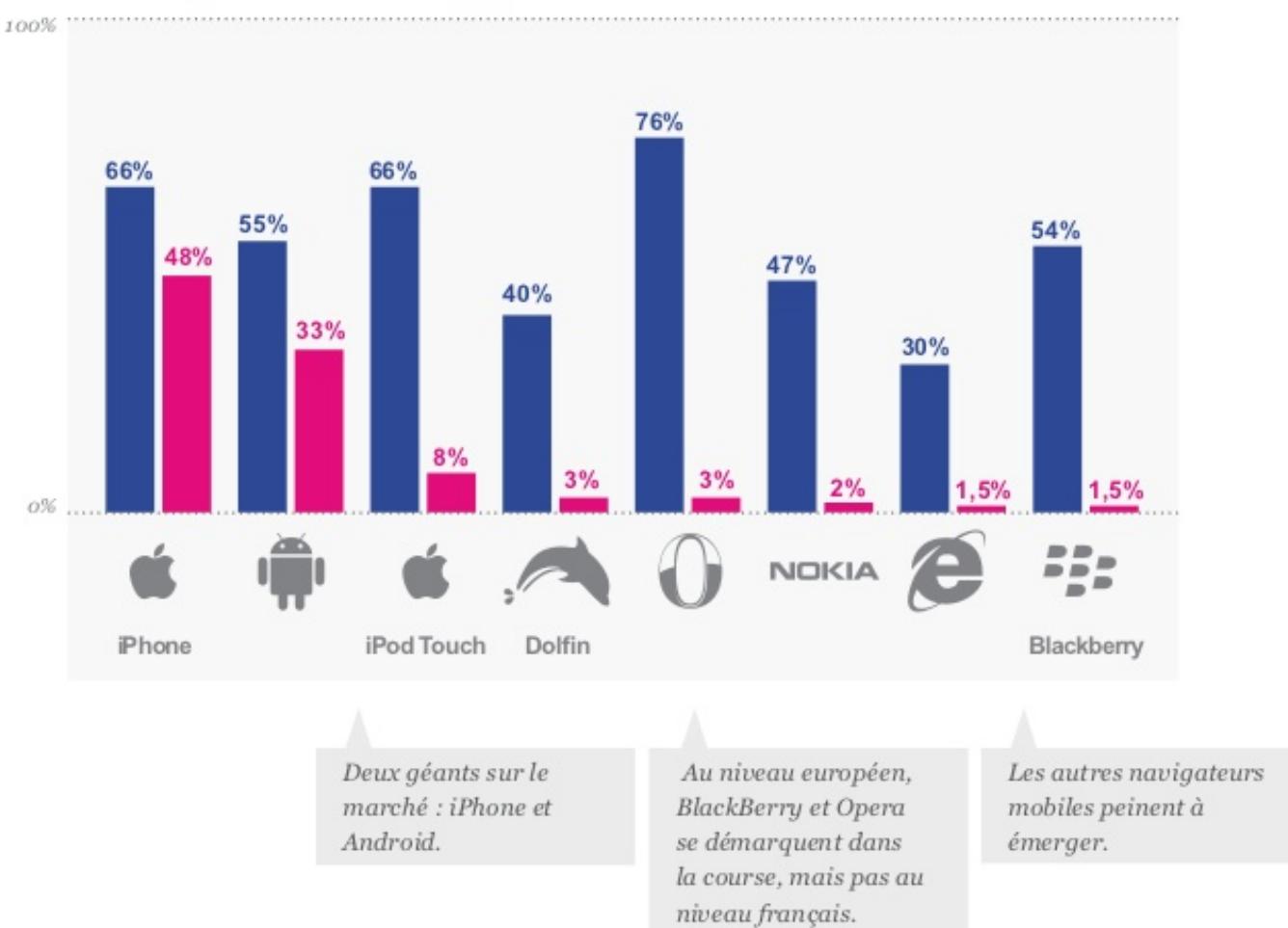
En France, seuls 56,9% des utilisateurs ont un navigateur mobile qui intègre à plus de 60% les fonctionnalités actuelles de HTML5.

Et seulement 2,7% des utilisateurs ont un navigateur mobile qui intègre à plus de 70% les fonctionnalités actuelles de HTML5. Cette proportion correspond en fait aux utilisateurs d'Opera Mobile, le seul à dépasser les 70% de compatibilité avec les fonctionnalités actuelles de HTML5.

Safari sur iPhone et iPod Touch et Android arrivent un peu au-dessus des 50% de compatibilité, et concernent la majorité des utilisateurs Français et Européens.

Figure 16
Navigateurs mobiles les plus utilisés en France et leur pourcentage de compatibilité avec HTML5.

Source
<http://gs.statcounter.com/>

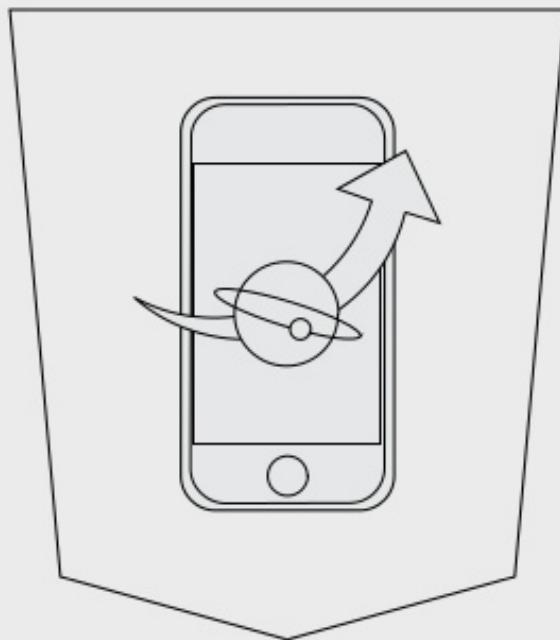


À retenir

Les Français (et plus généralement les européens) sont encore mal équipés pour utiliser des sites ayant recours aux fonctionnalités les plus avancées de HTML5. Un éditeur qui souhaite rendre son site accessible au plus grand nombre doit donc veiller aujourd'hui à ne pas utiliser les fonctionnalités les plus avancées de HTML5, ou alors à proposer systématiquement un fall back pour ces fonctionnalités.*

*Rappel : Un fallback est une « copie » d'une fonctionnalité de HTML5 développée avec d'autres technologies plus répandues (JavaScript, Flash, Silverlight, etc.) pour que les utilisateurs qui utilisent des navigateurs faiblement compatibles avec HTML5 puissent quand même profiter de cette fonctionnalité (parfois dans une version dégradée).

[II]



Les applications natives : menacées par le web mobile ?

- *Le modèle des applications, bien que dominant le marché du mobile...*
- *...Est actuellement remis en question*
- *HTML5 au service de la mobilité : forces et enjeux*

... Le modèle des applications, bien que dominant le marché du mobile...

Le modèle popularisé par Apple fonctionne tellement bien que toutes les plates-formes mobiles l'ont adopté.

Pourquoi le modèle des applications s'est-il imposé sur les smartphones ?

Côté utilisateurs :

- **Gain de temps pour l'accès aux contenus et services** : La connectivité mobile (aujourd'hui en France : EDGE, 3G) n'étant généralement pas aussi rapide que la connectivité fixe (Wi Fi, ADSL), la recherche sur le web peut prendre du temps. Les applications mobiles évitent ce problème puisqu'elles sont un point d'accès immédiat aux services préférés de l'utilisateur.
- **Amélioration des performances** : Tandis qu'un site web doit être intégralement chargé via Internet dans le navigateur à chaque visite, une application permet de charger une partie des fonctionnalités et des contenus « en dur », directement sur le téléphone. Cela diminue les besoins de bande passante pour utiliser le service correctement.
- **Meilleure intégration des fonctionnalités du hardware** : Parce que l'appli est développée spécifiquement pour une plate-forme et un type d'appareil, elle peut mieux tirer parti de l'équipement de cet appareil : recours à la caméra, au GPS intégré, optimisation des ressources de stockage ou de l'utilisation du processeur, etc.
- **Effets graphiques et interactions plus riches** : Ces éléments sont gérés en local, au niveau du téléphone, et pas en ligne comme dans un site web.

Côté éditeurs :

- **Un moyen de distribution efficace de leur service** : Sur les stores d'applications de type App Store, toutes les applications sont référencées, présentées de la même façon et téléchargeables très simplement. C'est un changement radical par rapport au web où les sites dépendent de leur référencement par les moteurs de recherche (en particulier Google).
- **Des perspectives de monétisation plus intéressantes que sur le web** : Le système de distribution imaginé par Apple favorise la création de contenus et services à haute valeur ajoutée, pour lesquels il est plus facile de faire payer l'utilisateur.

*Côté plates-formes
(iOS, Android, Blackberry, Windows Phone, etc.) :*

- **Une façon de garder le contrôle sur l'écosystème** : Apple, Google, etc. fixent les règles sur leurs plates-formes pour garantir à leurs utilisateurs un certain niveau de qualité et une certaine cohérence de l'expérience lorsqu'ils utilisent les applications.
- **Une nouvelle source de revenus** : Grâce aux stores d'applications, Apple, Google etc. maîtrisent complètement la chaîne de distribution des applications et peuvent se permettre de prendre une commission sur chaque vente d'application et chaque transaction effectuée au sein des applications.

...Est actuellement remis en question

Les écosystèmes d'applications, qui ont permis de redonner de la valeur à des services et des contenus qui avaient du mal à trouver un modèle économique viable sur le web (dominé par la gratuité), présentent aussi des inconvénients pour les éditeurs :

- **Une perte de contrôle** sur leur service au profit des plates-formes mobiles.
- **Une perte d'argent** liée à la commission prise par les plates-formes à chaque vente d'application et à chaque transaction à l'intérieur des applications.
- **Une perte d'efficacité** (qui se traduit en perte d'argent), car chaque plate-forme nécessite de développer une application spécifique, dans un langage spécifique, qui peut nécessiter l'intervention d'un prestataire spécifique.
- **Aucun apport de la communauté Open Source**
- **Très peu de lien** entre les applications qui restent des objets fermés
- **Une mauvaise gestion de la publicité display**, globalement moins contextualisée et personnalisée que sur le web

Malgré ces inconvénients, les éditeurs n'ont pas vraiment d'alternative :

- Dans son état actuel, HTML5 ne permet pas de rivaliser avec les applications natives.
- Le recours aux applications natives est devenu un réflexe pour les utilisateurs. Sur les appareils connectés, l'utilisation de navigateurs pour accéder à des services et consulter des informations reste marginale.

Le premier point est en passe d'être résolu grâce aux nouvelles possibilités d'HTML 5 : bientôt, il sera possible, pour certains types de services, de faire des sites web qui fonctionnent aussi bien (ou presque aussi bien) sur navigateur mobile qu'une application native. Ce n'est qu'une question de mois ou d'années (cf partie *III* de la note).

Le second point est plus délicat : les applications proposent une expérience tellement plus fluide et déjà tellement ancrée dans les usages qu'il faudra, pour que les utilisateurs se mettent à utiliser des sites web sur leurs appareils connectés, que ces sites web réussissent à faire mieux que les applications natives dans certains domaines.

... HTML5 au service de la mobilité : forces et enjeux

Les sites web en HTML5 peuvent faire beaucoup mieux que des applications natives

Remettre du lien entre les services

Contrairement aux applications natives, les sites web sont bien plus ouverts et interconnectés. Il est plus facile de créer des liens complexes entre deux sites Internet qu'entre deux applications.

S'inscrire dans un processus d'amélioration constante

Le process de mise à jour d'une application native est long, en particulier parce qu'il implique un contrôle de l'application par l'opérateur de la plate-forme (Apple pour l'App Store par exemple). La fréquence de leurs mises à jour est donc limitée. À l'inverse, un site Internet peut être mis à jour aussi souvent que son éditeur le souhaite.

La distribution des sites web sur mobile

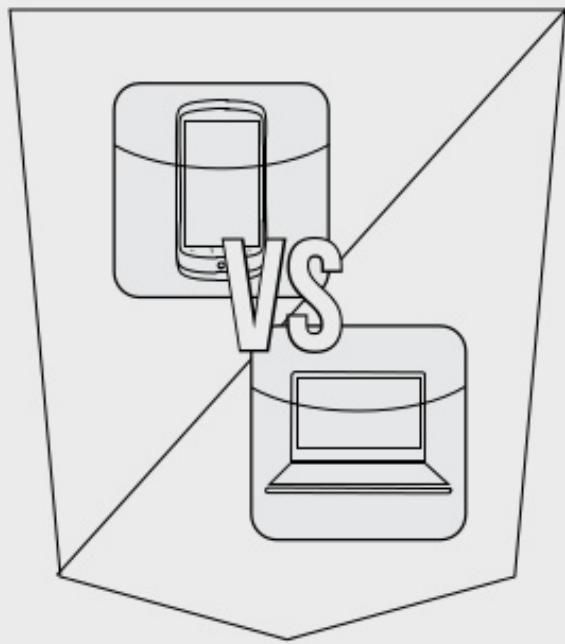
Une des clés de la réussite des applications mobiles est leur modèle de référencement et de distribution sur des **stores d'applications**. Les utilisateurs de mobiles trouvent immédiatement les applications dont ils ont besoin sur leur écran d'accueil, sans passer par un moteur de recherche web.

L'enjeu pour les éditeurs de sites en HTML 5 est de proposer un mode d'accès facile pour les utilisateurs de mobiles et autres appareils connectés.

Deux pistes sont à envisager :

- Communiquer et encourager les utilisateurs à créer des raccourcis vers les sites HTML sur l'écran d'accueil de leur appareil.
- Profiter de l'apparition de « stores universels » recensant des applications natives mais aussi des sites en HTML. Le store universel le plus prometteur est certainement le Facebook App Center, qui va profiter de la force de frappe de Facebook avec des 900 millions d'utilisateurs et sa puissance virale.

[III]



Applications mobiles vs applications natives

- Comparaison de leurs possibilités techniques et fonctionnelles
- Deux modes de conception qui mènent à des expériences utilisateurs différentes
- Grandes questions liées à la mise en œuvre d'un projet de développement
- Matrice de décision : que faire dans quel cas ?

... Comparaison de leurs possibilités techniques et fonctionnelles

Aujourd'hui, les applications natives permettent d'offrir une expérience de meilleure qualité pour les utilisateurs : performance graphique, ergonomie, interactions, etc. Cependant, nous avons vu qu'il commence à exister en HTML les outils nécessaires pour palier ces défauts. Il suffit donc que les navigateurs mobiles se mettent au niveau de la technologie HTML5.

	Site web (= web app)	Application native
Rendu graphique	Des progrès sont en cours grâce aux avancées de CSS et HTML mais elles sont encore peu intégrées sur les navigateurs de mobiles.	Rendu graphique de très bonne qualité, notamment pour la 3D.
Animations	Possibilité de faire des animations en 2D dans certains navigateurs récents, mais pas sur mobile. Impossible pour le moment de faire des animations en 3D sans le plug-in flash sur le web.	Meilleure puissance de calcul pour les animations en 2D et 3D, ce qui permet de créer des jeux, des expériences immersives, etc.
Accès à la caméra de l'appareil	Possible sur les navigateurs les plus récents, mais ceux-ci ne sont pas encore disponibles sur mobile.	Oui
Accès au micro de l'appareil	Possible sur les navigateurs les plus récents, mais ceux-ci ne sont pas encore disponibles sur mobile.	Oui
Accès à la géolocalisation de l'appareil	Oui	Oui
Utilisation de notifications depuis l'application vers l'écran d'accueil de l'appareil	Pas de notifications possibles d'un site web en HTML vers l'écran d'accueil. Cependant, les développeurs travaillent à combler ce manque.	Oui
Interactions utilisateur/machine avancées	Possibilité de naviguer comme sur le web en scrollant, en cliquant, etc. Avec HTML5, l'interaction via le micro arrivera bientôt sur les navigateurs mobiles.	Possibilité d'utiliser l'accéléromètre pour interagir avec l'appareil : secouer, renverser, incliner, etc.
Fonctionnement offline	Possible avec HTML5 mais pas forcément sur les navigateurs les plus utilisés.	Oui
Compte utilisateur (e-commerce, réseaux sociaux...)	Oui.	Oui

Suite du tableau page suivante >

	Site web (= web app)		Application native
Intégration de vidéos	Possible avec HTML5	✓	Oui, avec le lecteur natif de l'appareil. ✓
Jeux	Limité à des expériences graphiques en 2D, et à des actions de type « mouvement et clic de souris ».	≈	Design en 3D et interactions innovantes : incliner, secouer... ✓
Accès à l'application / au site	Au sein du navigateur web, ou depuis une icône sur l'écran d'accueil de l'appareil (raccourci vers le site web)	≈	Depuis une icône sur l'écran d'accueil. ✓
Téléchargement de l'application	Depuis le site web correspondant ou depuis une plateforme d'applications web.	≈	Depuis le Store correspondant : AppStore, Android Market, etc. ≈
Personnalisation de l'application	Uniquement si une inscription à un compte et la différenciation de différents profils sont prévues.	≈	L'application se personnalise facilement depuis les « réglages » : accès aux données de géolocalisation, notifications, etc. ✓
E-commerce	Possible via des sites comme PayPal facilement intégrable dans une webapps.	✓	Possible directement avec le compte de l'utilisateur sur la plate-forme. ✓
Liens vers d'autres services	Navigation facilitée d'une page à l'autre au sein du navigateur.	✓	Les applications sont des environnements fermés qui communiquent très peu entre eux. ✗
Partage sur les réseaux sociaux	En un clic si l'API du réseau social ou le widget sont intégrés sur le site.	✓	Possible si construit dans l'application. ✓

Figure 17
Comparaison des possibilités techniques et fonctionnelles entre un site web et une application native.

... Deux modes de conception qui mènent à des expériences utilisateurs différentes

Cas ebay

eBay est un site de vente aux enchères en ligne créé à la fin des années 90 : depuis, il s'est imposé comme référence dans le secteur de l'e-commerce. Plus de 270 millions d'utilisateurs vendent et achètent toute sorte d'objets par cette plateforme.

Pourtant le site internet de eBay se révèle particulièrement difficile d'utilisation : les informations sont difficiles à trouver, l'ergonomie laisse à désirer, etc. Au début, ce mauvais design n'était pas flagrant pour les utilisateurs, mais il l'est devenu en 2007 avec le lancement de l'application eBay pour iPhone, suivi un peu plus tard par le lancement de l'application eBay pour iPad. Les versions mobile et tablette sont en effet bien plus simples et rapides à utiliser.

Pourquoi cette différence ?

- Le développement d'un site web permet de plus grandes libertés de conception, parfois même - comme c'est le cas pour eBay - au détriment de la qualité de l'expérience utilisateur.
- A l'inverse, une application iPhone se développe à partir de briques ergonomiques préexistantes, et est destinée à un écran d'une taille restreinte qui force à aller à l'essentiel.

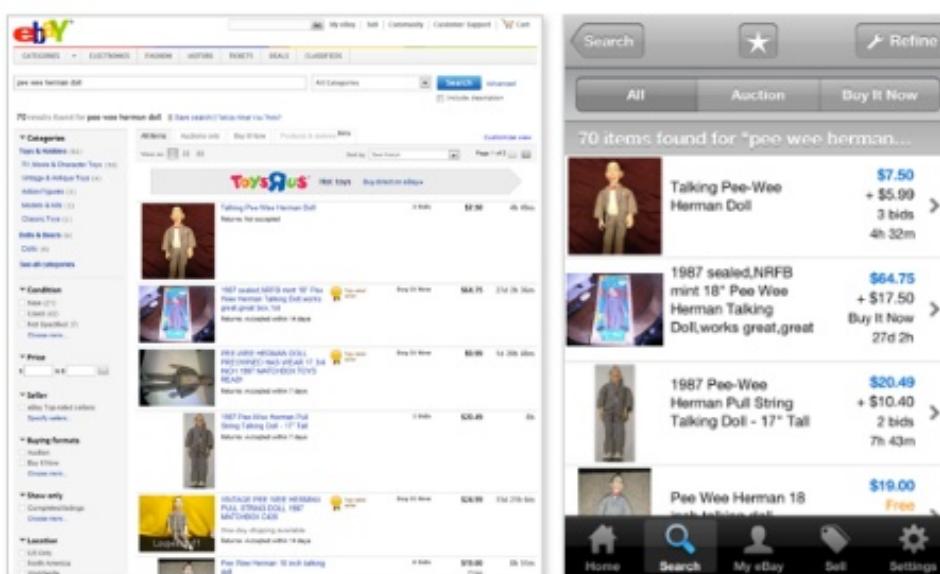


Figure 18
Comparaison entre le site web de ebay et leur application pour smartphone.

Ce qu'il faut retenir :

Le manque de liberté et les contraintes liées aux OS poussent à optimiser l'interface utilisateur et l'ergonomie de l'application, qui sera par ailleurs contrôlée et validée par le Store correspondant avant d'être mise en ligne. Cette exigence favorise la qualité.

Désormais, certains préconisent une «mobile first design strategy».

Le mobile étant par essence un support plus contraignant pour le développement (petit écran, connectivité souvent faible, utilisation en déplacement, etc.), il est un très bon point de départ pour concevoir un service simple, clair et efficace.

Le site web conçu après l'application mobile bénéficiera donc des grands principes de design et d'ergonomie liés à la conception de l'application native.

Pour un site développé en HTML 5, il est conseillé de commencer par concevoir l'expérience mobile: quelles sont les informations et fonctionnalités essentielles à mettre en avant ? Comment simplifier le parcours utilisateur ? Etc.

... Grandes questions liées à la mise en œuvre d'un projet de développement

Le tableau présenté ci-dessous montre que le développement en HTML est plus simple à mettre en œuvre que le développement d'une application native.

HTML étant plus ancien, on trouve plus de développeurs compétents, et la technologie est assez stable, simple à diffuser et à mettre à jour.

	Site web (= web app)	Application native
Ressources humaines nécessaires au développement	Un prestataire ou en interne pour le développement à proprement parler, souvent en interne pour la maintenance et mise à jour car HTML est un langage connu depuis longtemps et de nombreuses entreprises ont une division web intégrée.	Il faut la plupart du temps faire appel à un prestataire spécialiste d'un OS, même pour la maintenance. ✓ ✗
Disponibilité de développeurs compétents	Le développement en HTML5 requiert de très bonnes compétences web : connaissance des dernières tendances, maîtrise du responsive design, des APIs, des bonnes pratiques de CSS, etc. Les développeurs HTML5 sont donc plus rares et plus chers que la moyenne des développeurs web.	Les développeurs compétents sont relativement nombreux pour les plates-formes les plus répandues (iOS et Android) mais plus rares et plus chers pour les autres plates-formes. ≈ ≈
Coût global de développement	Un seul site (en responsive design) à développer pour toutes les plates-formes.	Un développeur spécialisé de chaque plate-forme pour chaque application. ✓ ✗
Mise à jour et maintenance	Réalisée par le développeur, synchronisée automatiquement sur le web.	Réalisée par le développeur, il faut manuellement télécharger la nouvelle version de l'application : c'est plus pénible pour l'utilisateur. ✓ ✗
Confidentialité du code	Le code est ouvert (HTML). ≈	Le code est tenu confidentiel. ≈
Délais de développement	Uniquement le temps nécessaire au travail du développeur. Possibilité de faire du prototypage rapide.	Plus long car l'application doit être contrôlée et validée par le Store concerné. ✓ ✗
Résolution de l'écran	Utilisation possible du responsive design qui adapte le contenu à l'orientation de l'écran. Le responsive design permet ensuite aussi d'adapter une même application à toutes sortes d'écrans.	A chaque taille d'écran correspond une application différente, développée spécifiquement. ✓ ✗
Design et apparence	Entièrement libre.	Des standards sont imposés par l'OS correspondant. ✓

Figure 19
Comparaison du développement d'un site web et d'une application native.

Concernant **les business models** employés, les politiques sont différentes :

- Le modèle HTML se veut un modèle ouvert : pas de contrôle de qualité, gratuité du service (la plupart du temps du moins), universalité des supports de lecture, distribution et référencement sur le web.
- Le modèle natif est un modèle d'exigence et de qualité : chaque support a sa propre application, les applications sont contrôlées et validées avant d'être référencée sur un Store qui facilite leur distribution mais qui a un coût pour l'éditeur. Les utilisateurs sont prêts à payer pour ce niveau de qualité.

	Site web (= web app)	Application native
Référencement	Comme tout site web, les web apps sont référencées sur les moteurs de recherche, comme Google. Elles peuvent aussi se trouver sur des Stores de web applications.	≈ Pas de référencement sur le web, mais seulement sur le Store correspondant au support utilisé : on ne trouve que des applications compatibles avec le support utilisé, ce qui simplifie le parcours utilisateur. ✓
Monétisation, publicité	Contrôle de la publicité et de la monétisation de la web app en fonction de la page web consultée.	✓ Il faut passer par les Stores correspondants pour intégrer une publicité sur une application. Elle ≈ n'est pas forcément liée à l'application consultée.
Interopérabilité	Un site HTML peut être lu depuis tous type d'appareils : smartphone, tablettes, ... et toucher le maximum d'utilisateurs.	✓ Une application développée pour une plate-forme ne peut pas être lue sur un autre, elle ne vise qu'une catégorie d'utilisateurs. ✗
Vente	Une application web est accessible généralement gratuitement, il s'agit d'un site web.	✗ Une application native est disponible sur un Store, gratuite ou payante, ≈ mais le distributeur prend environ 30% du prix.
Distribution	Pas de mode de distribution, à part le référencement sur les moteurs de recherche.	≈ Via un Store tel que l'App Store ou l'Android Market. Cela permet de bénéficier de l'image du distributeur. ✓ Cependant, c'est lui qui décide de la visibilité de l'application (seules les 50 premières ont vraiment du succès).
Evolution par rapport au contexte	Le web sera toujours là dans quelques années, et au-delà, en utilisant les normes HTML promues par le W3C. Développer en HTML est une solution durable.	✓ On ne sait pas quelle plate-forme dominera le marché dans les années à venir : montée d'Android et Samsung face à Apple... Développer en natif pour la plate-forme dominante d'aujourd'hui ne garantit pas le succès pour l'avenir. ✗

Figure 20
Comparaison des business models liés au développement d'un site web et d'une application native.

Etude de Cas : Financial Times

En juin 2011, le Financial Times a lancé sa web app qui a considérablement accéléré la consultation sur mobile du FT (smartphone et tablettes) : 12% des inscriptions se font sur mobile, et 19% du trafic du site est générée par la consultation sur mobile, ce qui représente 2M d'utilisateurs.

En août 2011, la web app avait déjà attiré plus d'utilisateurs que l'application native qui existait alors, et qui a été retirée de l'AppStore depuis.

HTML5 leur a en effet permis de toucher plus d'utilisateurs, notamment tous ceux qui avaient un smartphone Androïd, ou d'autres navigateurs. Par ailleurs, FT souhaitait également se libérer du contrôle d'Apple et des 30% de revenus prélevés par la firme de Cupertino.



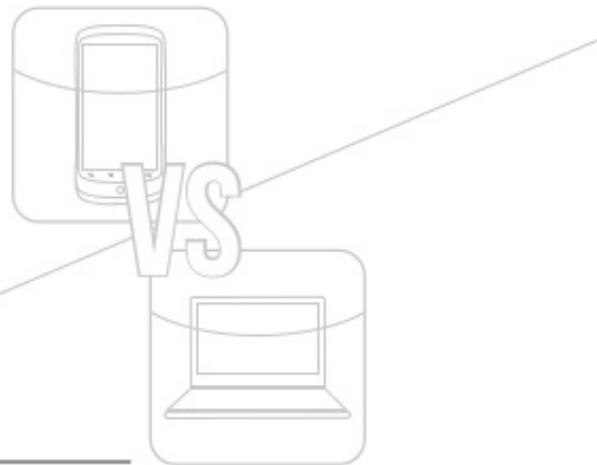
Figure 21
Le site web du Financial Times lu depuis une tablette.

Matrice de décision : que faire dans quel cas ?

Site Internet ou applications natives ?

Application Native si le service...

- Utilise des fonctionnalités du téléphone : appareil photo, gyroscope, boussole...
- Permet plusieurs utilisations possibles, réglables dans les « préférences »
- Est accessible simplement depuis n'importe quel autre application ou depuis la home
- Fait partie d'un business model cohérent avec ceux des Store
- Ne vise qu'une catégorie de personnes avec un seul Smartphone
- Doit être très performant
- N'a pas besoin de se connecter à internet
- Est développé avec un code qui doit rester confidentiel
- Utilise des données qui doivent être sécurisées



Site Internet (= Web App) si le service...

- Doit toucher toute la population, quel que soit le support utilisé, quel que soit le marché futur des Smartphones ou tablettes ou smart TV...
- ...Ou au contraire, touchera très peu de personnes (inutile d'investir dans un langage moins connu que HTML)
- Est accessible par une simple recherche web
- Sera souvent mis à jour et rapidement
- Ne coûte rien pour être mis en ligne
- Propose des liens vers d'autres sites web
- Sera maintenu par quelqu'un d'autre que le développeur
- Doit avoir un design complètement original
- Doit permettre un contrôle complet de la monétisation et de la publicité (analytics)
- Doit être rapidement développé/pourra servir à développer d'autres services (réutilisation de widgets existants, etc.)
- Sera disponible rapidement
- Fait aussi référence à un site web à consulter depuis un ordinateur

Qu'en est-il des solutions hybrides ?

Il est impossible de convertir facilement une application native d'un support vers un autre. En effet :

- Le langage de développement diffère ;
Exemples : Objective-C pour Apple, Java pour Android
- L'ergonomie varie d'une plateforme à une autre ;
- Les OS qui supportent les applications sont différents.

La seule solution possible pour développer une application native qui fonctionne sur différentes plateformes est de développer une application native différente pour chaque plateforme, avec une ergonomie propre, et un développeur spécialiste de l'OS et du langage.

A l'inverse, une application web développée en HTML peut être lu par tout téléphone muni d'un navigateur Internet : Iphone, Androïd, ou encore BlackBerry.

Pourtant, certaines applications recherchent à la fois la possibilité :

- D'être utilisée sur tout type d'OS : Androïd, iOS, etc. ;
- D'accéder aux fonctionnalités natives du Smartphone ou de la tablette : micro, caméra, géolocalisation précise, liste de contacts, calendriers.

Dans ces cas particuliers, il est possible d'opter pour une solution intermédiaire : les applications hybrides.

Une application hybride est basée sur les mêmes technologies que les applications web : le code est construit avec les standards HTML, CSS et JS. Elle a donc les mêmes performances et les mêmes qualités que les applications du web.

Ce code est ensuite intégré dans des « frameworks » existants lesquels lui permettent d'accéder aux fonctionnalités natives d'un OS.

Qu'est-ce qu'un Framework ?

Il s'agit d'une application native vide, qui permet simplement de lire le contenu de la page web développée en code HTML. Le Framework a des propriétés très similaires à celles d'un navigateur.

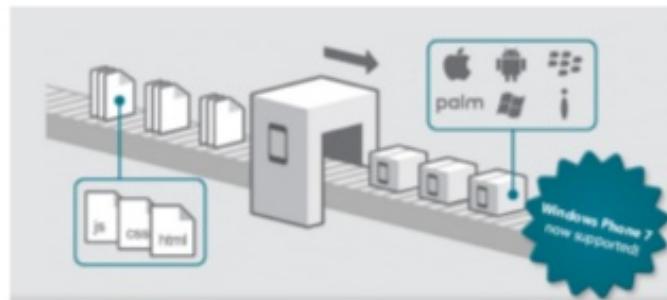


Figure 22
Schéma de l'entreprise
PhoneGap montrant le rôle
des frameworks.

Ainsi, comme toute application native, l'application hybride a accès aux fonctionnalités du Smartphone (appareil photo, GPS, etc.) et sera référencée sur le Store correspondant.

L'application hybride profite en plus des avantages de son code HTML : simplicité de mise à jour et de développement, possibilité de faire du responsive design, stabilité durable du langage employé.

Quels acteurs proposent ces frameworks ?

Voici deux acteurs qui développent des frameworks et intègrent des pages HTML pour en faire des applications hybrides sur différentes plateformes.

PhoneGap

soutenu par l'incubateur de la fondation Apache

Les fonctionnalités natives supportées par les différents frameworks de PhoneGap sont les suivantes :

	iOS iPhone / iPhone 3G	iOS iPhone 3GS and newer	Android	BB OS 5.x	BB OS 6.0+	hp WebOS	WP7	Symbian	COCO Bada
ACCELEROMETER	✓	✓	✓	✓	✓	✓	✓	✓	✓
CAMERA	✓	✓	✓	✓	✓	✓	✓	✓	✓
COMPASS	✗	✓	✓	✗	✗	✗	✓	✗	✓
CONTACTS	✓	✓	✓	✓	✓	✗	✓	✓	✓
FILE	✓	✓	✓	✓	✓	✗	✓	✗	✗
GEOLOCATION	✓	✓	✓	✓	✓	✓	✓	✓	✓
MEDIA	✓	✓	✓	✗	✗	✗	✓	✗	✗
NETWORK	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (ALERT)	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (SOUND)	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (VIBRATION)	✓	✓	✓	✓	✓	✓	✓	✓	✓
STORAGE	✓	✓	✓	✓	✓	✓	✓	✓	✗

Figure 23
Tableau des fonctionnalités natives de différentes plateformes, supportées par les frameworks de l'entreprise PhoneGap.

Sencha Touch

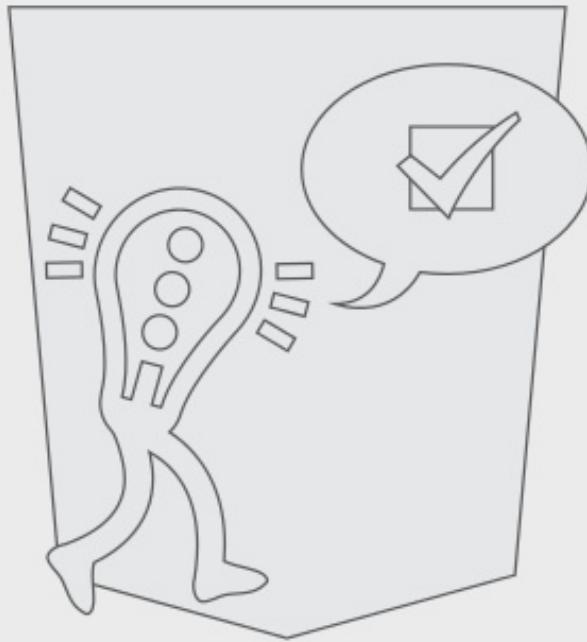
Sencha propose également des frameworks pour que les applications web aient accès à la plupart des fonctionnalités natives des appareils : géolocalisation, boussole, accéléromètre, etc. Pour donner accès à l'appareil photo, ils utilisent les développements de PhoneGap.

Ci-dessous la liste des appareils compatibles avec la solution Sencha :



Figure 24
Les différentes plateformes sur lesquelles Sencha Touch peut diffuser une application hybride, à partir du même code HTML5.

[IV]



Nos recommandations

- *Finalement, quelle solution pour quel type d'applications ?*
 - Jeux et divertissements*
 - Médias, actualités et informations*
 - Réseaux sociaux*
 - Déplacements et cartes*
 - Vie pratique (réservation, e-shopping, transaction bancaire)*
 - Multimedia*
- *Synthèse*

Nos recommandations

Finalement, quelle solution pour quel type d'applications ?

Jeux et divertissements

[Application native]

Pourquoi ? De telles applications exigent réactivité et graphisme de qualité pour évoquer un nouvel univers, ainsi qu'éventuellement des interactions innovantes avec les éléments du hardware (accéléromètre par exemple).

Exemple 1 : Infinity Blade

Ce jeu est très proche des jeux vidéos pour console. Son graphisme 3D, de grande qualité, demande une puissance de calcul élevée pour permettre à cet univers de réagir aux actions des joueurs. Aujourd'hui impossible à obtenir en HTML5, le développement natif s'impose.



Figure 25
Image du jeu "Infinity Blade", une application native qui utilise des animations en 3D.

Exemple 2 : Angry Birds

L'un des jeux les plus célèbres sur téléphone mobile est désormais disponible entièrement en HTML5... mais uniquement sur Chrome 19, qui n'est pas encore disponible pour un Smartphone ou une tablette. Précisons qu'il s'agit d'un jeu encore relativement simple : un seul type d'interaction, graphisme en 2D.



Figure 26
Le jeu "Angry Birds" désormais disponible en HTML5 depuis le navigateur Chrome 19.

Médias, actualités et informations

[Hybride aujourd'hui ... web app pour bientôt !]

Pourquoi ? Les informations doivent être **sans cesse consultées, actualisées...** et tous les utilisateurs en ont besoin, quel que soit leur mobile. Par ailleurs, **il n'y a pas d'interactions particulières** à imaginer sur une application d'informations (lecture, partage, commentaire). L'avantage des applications natives est de permettre les notifications : cette fonctionnalité devrait arriver d'ici peu sur HTML5.

Exemples : Financial Times & Le Monde

Ce jeu est très proche des jeux vidéos pour console.
Son graphisme 3D, de grande qualité, demande
une puissance de calcul élevée pour permettre
à cet univers de réagir aux actions des joueurs.
Aujourd'hui impossible à obtenir en HTML5, le
développement natif s'impose.

Financial Times est une
web application



Le Monde est une
application native



Figure 27

Les applications mobiles du Financial Times (site web) et du Monde (application native).

Réseaux sociaux

[Natives ou hybrides]

Pourquoi ? Les applications de réseaux sociaux ont des caractéristiques assez proches de celles d'une application d'actualité. Elles sont par ailleurs très souvent mises à jour et modifiées, et visent un public large, qui utilise des supports variés. Enfin, elles ont besoin d'avoir accès à l'appareil photo et à la géolocalisation pour que les utilisateurs partagent également ce type de données.

Exemple 1 : LinkedIn

LinkedIn est une application hybride.



Figure 28
L'application hybride du réseau LinkedIn sur Smartphone.

Exemple 2 : Facebook

Récemment, Facebook a abandonné HTML5 pour une nouvelle version de son application 100% native. L'objectif de Facebook est de rendre l'application plus réactive lors de son ouverture et de l'affichage des contenus multimédia notamment. Facebook a donc fait le choix de favoriser l'efficacité et la rapidité d'une application native face à l'option d'un développement HTML5 standard lisible sur n'importe quelle plate-forme mobile.



Figure 29
La nouvelle application native Facebook lancée en Aout 2012.

Déplacements et cartes

[Application native et web application]

Pourquoi ? Pour ce type d'application, la géolocalisation est indispensable. La géolocalisation est aussi précise avec une web application qu'avec une application native. La géolocalisation fait partie des premières fonctionnalités compatibles avec les navigateurs les plus courants.

Exemple : Foursquare

Foursquare est une application native disponible sur Android et sur iOS.

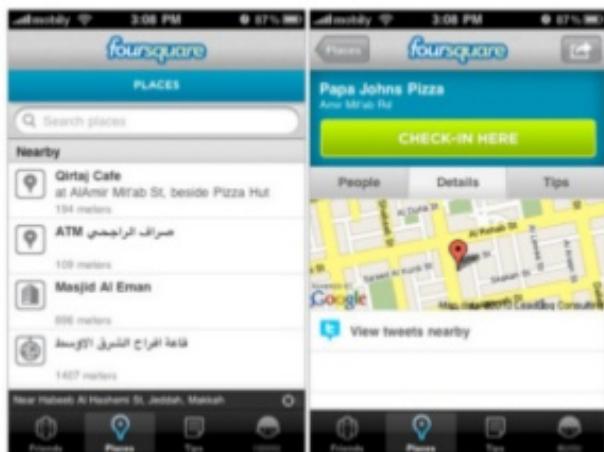


Figure 30
L'application Foursquare, une application native qui utilise la géolocalisation native des appareils.

Vie pratique (réservation, e-shopping, transaction bancaire)

[Application native]

Pourquoi ? Pour ce genre d'applications, il faut pouvoir s'identifier et effectuer des transactions en toute sécurité. Dans les applications natives, l'utilisateur est la plupart du temps déjà enregistré et identifié, donc il n'a pas besoin d'entrer ses identifiants à chaque transaction.

Exemple : eBay

Le groupe eBay a fait le choix d'une application native, en plus d'un site web optimisé pour mobile. En effet, les fonctionnalités disponibles sur iPhone permettent d'enrichir encore le service proposé par eBay sur le web : scan de code bar avec la caméra, notifications lorsqu'un objet est vendu ou acheté, etc. Ces fonctionnalités ne peuvent pas encore exister dans HTML5.



Figure 31
L'application eBay, une application de e-commerce native, qui utilise les notifications et la caméra pour scanner un code barre.

Multimedia

[Application native hier ... web application demain !]

Pourquoi ? Pendant très longtemps, les éditeurs de sites qui offrent la consultation de films ou musiques n'avaient d'autres alternatives que le recours aux applications natives. Désormais, **HTML5 permet d'intégrer du contenu multimédias sur les mobiles ou les tablettes**, avec une meilleure réactivité, si la connexion à internet est sûre.

Exemple : You Tube

Lors de la sortie des premiers Smartphones, YouTube a au préalable développé une application native qui utilise le lecteur de vidéo du Smartphone. En effet, seul l'ajout d'un plug-in Flash permettait alors de lire des vidéos en ligne. Or, Apple refusait de l'intégrer dans iOS mobile.

Plus récemment, les progrès de HTML5 ont permis de créer une application web de YouTube qui fonctionne sur toutes les plateformes.



Figure 32
L'application YouTube, en HTML5, identique sur toutes les plateformes.

Synthèse

Le débat application native contre application web n'est finalement pas de savoir quelle est la meilleure technologie, mais quelle est la plus adaptée aux besoins de l'éditeur en termes de qualité et de rapidité de l'application.

Lorsque l'application doit être très réactive, qu'elle doit fonctionner en mode déconnecté, ou utilise des fonctionnalités avancées du terminal (caméra, gyroscope, voix, push, envois par e-mail, carnet d'adresses, etc.), une application native est incontournable. De fait, chaque application sera optimisée pour sa propre plateforme.

Si au contraire, le développement de l'application est conduit par une logique de coûts et ne nécessite pas de fonctions aussi avancées que celles d'une application native, HTML5 offre le meilleur compromis possible. L'intérêt est de n'avoir alors qu'une seule application à développer et à maintenir. Les économies peuvent être considérables dès que le nombre d'applications est important.

Bien qu'HTML5 doive encore gagner en maturité...

Les fonctionnalités les plus intéressantes de HTML5 ne sont pas encore supportées par les navigateurs les plus courants. Elles le seront dans les prochains mois pour les navigateurs d'ordinateurs de bureau, mais pas forcément pour les navigateurs mobiles, tablettes et smartTV. Les acteurs qui contrôlent ces supports (Apple, Google, Samsung, etc.) n'ont en effet pas intérêt à encourager le développement des web apps.

Le développement des web apps sur les supports connectés nécessite deux révolutions difficiles à prévoir :

- Un changement de cap dans la stratégie d'Apple, Google, Samsung, etc. qui les amènerait à améliorer le support d'HTML5 sur leurs appareils connectés.
- Une révolution dans les habitudes des usagers qui se mettraient à passer par leur navigateur plutôt que par une application native pour accéder à leurs services préférés. Compte-tenu du succès actuel des applications natives, ce revirement dans les usages n'aura pas lieu sans une disruption majeure du marché.

...il est temps d'intégrer HTML5 à sa stratégie web multiplateformes...

Les navigateurs pour ordinateurs de bureau intègrent de plus en plus vite les nouvelles fonctionnalités d'HTML5. D'ici la fin de l'année 2012 ou le début de l'année 2013 HTML5 pourra être considéré comme la norme pour tous les développements web.

Même si toutes ne sont pas encore adoptées par les différents navigateurs, globalement, les nouvelles fonctionnalités de HTML5 permettent d'aller vers des expériences plus riches et plus interactives sur le web. Au-delà de cet aspect technologique, HTML5 apporte aussi une nouvelle façon de réfléchir et de concevoir :

- Stratégie multi-plates-formes : comment adapter une même expérience à différents supports (mobile, tablette, TV) en mutualisant les coûts de développement ?
- Facilité de mise à jour : les évolutions et les différentes versions sur le web se succèdent de façon fluide et transparente pour l'utilisateur. Cela permet également de concevoir de nouvelles expériences à travers des méthodologies de prototypage rapide, de façon incrémentale et itérative.
- Vers une logique de flux : on ne télécharge plus, on ne stocke plus. Tout est accessible sur le web, via une url.

Cela veut dire que les développements de nouveaux sites ou les mises à jour de sites existants devront se faire en considérant l'utilisation d'HTML5 et des standards du web pour remplacer des technologies propriétaires ou pour proposer une expérience utilisateur plus riche. Quelques exemples pour pluzz.fr :

- Pour les vidéos, passer de Flash à HTML5
- Proposer un mode plein écran pour plus d'immersion
- Etc.

...et d'explorer le terrain des applications hybrides.

Un service comme Pluzz doit pouvoir être déployé sur le plus grand nombre d'appareils connectés possible : smartphones, tablettes, smartTV. Pour limiter les coûts liés au développement d'une application native différente pour chaque plate-forme, un éditeur a tout intérêt à utiliser un cœur commun en HTML5. C'est par exemple la stratégie choisie par Netflix.

Les avantages :

- Le cœur HTML5 peut être mis à jour à tout moment, sans avoir à passer par Apple, Google, Samsung, etc.
- Il est possible de faire des tests dynamiques (A/B testing) pour améliorer rapidement l'interface
- Le coût de développement pour chaque application est largement réduit
- Cette stratégie constitue une opportunité pour penser de manière globale l'ensemble des services et ce pour une gestion plus simple, plus efficace et plus rationnelle (systématiser l'utilisation d'APIs, mieux structurer ses données, mieux faire « parler entre eux » les différents services, etc.).

Sources

Figure 3

HTML5 : le futur du web ?
<http://www.w3.org/html/logo/>

Figure 4

Exemple d'animation : les traits lumineux se dessinent en suivant le trajet de la souris sur l'écran, ils bougent et s'effacent au fur et à mesure.
<http://neave.com/fr/imagination/>

Figure 5

Exemple d'animation CSS : le texte s'aligne sur les contours de la voiture lorsque celle-ci est déplacée sur l'écran
<http://www.adobe.com/fr/devnet/html5/articles/css3-regions.html>

Figure 6

Exemple d'interaction avec le micro : lorsqu'on fait du bruit devant l'ordinateur, les billes se mettent à sauter sur l'écran.
<http://neave.com/fr/balles-sauteuses/>

Figure 7

Exemple d'interaction avec la webcam : on peut se prendre en photo et ajouter des filtres ou des effets.
<http://neave.com/fr/webcam/>

Figure 8

Dessin réalisé à l'aide de la souris au sein d'une page web
<http://harmoniousapp.com/>

Figure 9

Reprise du célèbre jeu Snake entièrement en HTML5
<http://neave.com/fr/serpent/>

Figure 10

Illustration d'un jeu en HTML5, piloté grâce à une manette
<http://hacks.mozilla.org/2011/12/gaming-and-the-mozilla-labs-apps-project>

Figure 11

Score (sur 500) des six navigateurs les plus compatibles avec HTML5.
<http://html5test.com/results/desktop.html>

Figure 12

Score (sur 500) des huit OS de smartphones qui permettent la meilleure compatibilité avec les fonctionnalités de HTML5.
<http://html5test.com/results/mobile.html>

Figure 13

Score (sur 500) des douze OS de tablettes qui permettent la meilleure compatibilité avec les fonctionnalités de HTML5
<http://html5test.com/results/tablet.html>

Figure 14

Score (sur 500) des 10 OS de Smart TV qui permettent la meilleure compatibilité avec les fonctionnalités de HTML5
<http://html5test.com/results/television.html>

Figure 15

Navigateurs web les plus utilisés en France et leur pourcentage de compatibilité avec HTML5.

<http://gs.statcounter.com/>

Figure 16

Navigateurs mobiles les plus utilisés en France et leur pourcentage de compatibilité avec HTML5.

<http://gs.statcounter.com/>

Figure 22

Schéma de l'entreprise PhoneGap montrant le rôle des frameworks.

<http://gs.statcounter.com/>

Figure 23

Tableau des fonctionnalités natives de différentes plateformes, supportées par les frameworks de l'entreprise PhoneGap

<https://build.phonegap.com/>

Figure 24

Les différentes plateformes sur lesquelles Sencha Touch peut diffuser une application hybride, à partir du même code HTML5

www.phonegap.com

Figure 29

La nouvelle application native Facebook lancée en Aout 2012.

www.igeneration.fr

Crédits et remerciements

francetélévisions

Bruno Patino

Directeur général délégué au développement
numérique et à la stratégie,
Directeur de France 5

Eric Scherer

Directeur de la prospective

Philippe Deloeuvre

Directeur de la stratégie



Aurélien Fache

Directeur Technique



Romain Goyet

Co-fondateur et CTO

Contact



17 rue du Faubourg du Temple
75010 Paris
+33 1 42 72 20 04

www.fabernovel.com

Marie-Caroline Lanfranchi

Project Director
marie-caroline.lanfranchi@fabernovel.com

Maxime Coupez

Senior Project Manager
maxime.coupez@fabernovel.com

Marguerite Meunier

Junior Project Manager
marguerite.meunier@fabernovel.com

Nous sommes faberNovel

Quelques références

SFR PLAYER

SFR Player, l'expérience en ligne ludique et immersive.

Dans le cadre de la session 2011 de l'événement SFR PLAYER, faberNovel a pris en charge la conception et l'animation d'une expérience en ligne complète, autour d'innovations technologiques.

<http://sfrplayer.sfr.com/>

INSIDE ENGRENAGES

En collaboration avec Canal+, faberNovel a pris en charge la réalisation de l'expérience interactive « Inside Engrenages », websérie INEDITE de la création originale « Engrenages ».

“Glissez-vous dans la peau d'un webreporter autorisé à suivre le groupe de la capitaine Laure Berthaud, le temps d'une enquête sur un réseau de trafic de faux papiers et de clandestins. »

Durant sept jours d'investigation, l'internaute suit en temps réel l'évolution de l'enquête et est averti par mail ou via les réseaux sociaux des rebondissements ou de la découverte de nouveaux indices.

Synopsis : Carole Becker, 24 ans, est retrouvée morte à son domicile, à demi nue, la tête entourée de scotch. La 2ème DPJ de Paris est saisie de l'affaire et arrive sur les lieux...

<http://engrenages.canalplus.fr/>



facebook.com/fabernovel



@fabernovel



www.fabernovel.com