



La notion de Web 2.0 fait beaucoup parler d'elle mais est difficile à appréhender du fait qu'elle ne représente pas une nouvelle technologie mais plutôt un concept. L'objectif est de proposer une nouvelle expérience utilisateur et d'axer vers un meilleur service client.

Je vous propose ici un petit dossier qui permet de introduire ce nouveau terme (nébuleux !?).

La présentation fut découpée en quatre parties :

- Présentation et définition du terme Web 2.0
- Paradigm shift : transition des modèles et des langages Web
- Intégration : présentation des solutions
- Développement : présentation des solutions

Ces résultats sont principalement basés sur l'article "What is Web 2.0?" (Tim O'Reilly 2006) et sur la présentation faite par Etienne LAVERDIERE de Bull SA.

{mospagebreak heading=Introduction&title=Web 2.0} **Web 2.0**

Une très bonne manière d'appréhender ce que représente le Web 2.0 est la comparaison suivante entre un site représentant le Web 1.0 et un autre représentant le Web 2.0.

Mon exemple préféré est celui de Wikipedia et de Britannica Online. Ce sont tous deux des sites dédiés à la diffusion en ligne d'une encyclopédie. Mais là où Wikipedia a écrasé son concurrent est sur l'interaction qu'il propose avec les utilisateurs. Alors que Britannica Online se contente de faire de la publication, Wikipedia propose un système participatif. Le résultat est une expérience plus riche des utilisateurs et donc un plus grand intérêt/nombre de clients. Il permet de plus d'enrichir le contenu de manière flagrante.

On peut résumer les grandes idées du Web 2.0 par:

- Une expérience utilisateur riche
- Une orientation service
- Une orientation data: importance des données sur la présentation

On comprends alors le potentiel inexploité pour les sociétés possédant un grands nombre de données, mais peu utilisé (base de données utilisateur).

---



## Paradigm Shift

Pour appliquer le modèle Web 2.0, de nouvelles contraintes sont imposées, et que les anciens langages de programmation ne pouvaient satisfaire.

**AJAX** (Asynchronous JavaScript and XML) permet alors d'éviter la synchronisation en client et serveur, et d'éviter un rafraichissement obligatoire de la page web pour l'affichage d'un nouveau contenu. L'idée est d'utiliser une connection HTTP long-live et de ne pas la fermer. Cela permet alors au serveur de fournir l'information à la demande, et dans des délais très

courts.

**Comet** (Reverse AJAX) permet en plus la création de Push côté serveur et d'application Event-driven. Ce sont les événements du serveur qui initialisent le transfert d'information vers le navigateur.

---

## Intégration

**Encapsulation de Données** : XML a certes ses avantages mais un gros inconvénient: l'utilisation d'un parseur XML afin d'extraire les données. La charge et donc le temps que ce travail demande reste très pénalisant, surtout dans un environnement web. L'utilisation d'autres notations est donc recommandée, comme JSON qui a l'avantage d'être directement interprété comme du JavaScript.

**ReST**: c'est en fait un architecture que permet la fabrication d'applications pour les systèmes distribués. Le protocole HTTP fournit déjà suffisamment de verbes (GET, POST, PUT et DELETE) afin d'effectuer une communication complète entre machine et non pas seulement homme/machine. C'est un moyen plus simple que SOAP ou RPC à mettre en œuvre et plus rapide (6x plus rapide que SOA). Enfin une grande compatibilité est à signaler car nombreux Toolkits et Framework sont implémentés pour utiliser ReST dans un processus AJAX.

**MashUp**: Certaines APIs, comme Google Maps, sont mise à disposition et peuvent être utilisées dans un cadre très large. Réutiliser ces APIs à son propre bénéfice est l'objectif.

---

## Développement

De très nombreux Toolkits et Framework existent. Il nous a surtout été présenté le toolkit JavaScript Dojo et le Framework DWR pour les raisons suivantes.

**Dojo Toolkit:** Dojo fournit un système de paquetage conçu pour faciliter la séparation des fonctionnalités en paquetages et sous paquetages individuels. Une grande bibliothèque de composants est proposée, et aidant à la l'utilisation d'objet complexes. Une autre de ses forces est le Layout Widgets: un système de conteneurs qui permet de simuler des boîtes / fenêtres redimensionnables et repositionnables à la souris. Mais sa faiblesse reste le manque de documentation.

**DWR FrameWork:** Son principal avantage sur les autres est sa stabilité. Léger et simple à utiliser, il est aussi facile à intégrer. Enfin sa licence Apache le classe parmi les meilleurs choix.

---

## Conclusion

Même si cette obscure notion qu'est le Web 2.0 peut sembler un écran de fumée, j'espère que ce dossier vous aura aidé à comprendre ce qu'elle représente.

Pas de nouvelles technologies, ni rien de révolutionnaire d'ailleurs mais un concept, une

philosophie d'un Web toujours plus interactif, offrant toujours plus de service.

Et n'est ce pas ce que l'on demande au final, profiter d'un contenu interactif, gratuit, puissant et disponible. On en viendrait presque à trouver cela plaisant ;)