Mozilla Persona : Enfin le bon système d'identification sur le Web ?

Avec combien de login et de mots de passe devons-nous jongler dans la journée ?

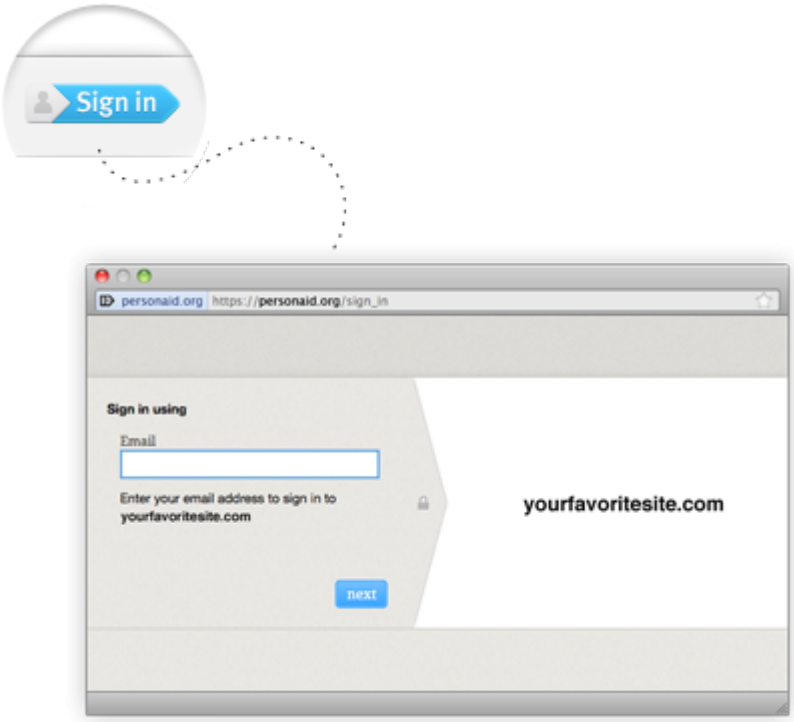
Beaucoup en effet…

En même encore plus si nous ne les enregistrions pas automatiquement par défaut dans notre machine (avec un gros risque de sécurité). Ou, pire encore, si nous n'avions pas décidé de nous enregistrer sur des services tiers avec notre compte Facebook, Twitter ou Google (avec un énorme risque de se retrouver prisonnier de leurs cages dorées).

À l'heure et à l'ère de la multiplication exponentielle des services Web, la question de l'identification devient cruciale et problématique.

Nous avons besoin d'un système global pratique, sécurisé et auquel nous pourrions faire confiance.. C'est ce que propose **Persona**, le nouveau système d'authentification de Mozilla.

Il n'est pas forcément aisé de le comprendre de l'intérieur mais c'est pour que ce soit le plus simple possible du côté de l'utilisateur



Pourquoi Persona de Mozilla apporte la bonne réponse à la question de l'identification

Why Mozilla Persona Is the Right Answer to the Question of Identity

David Somers – 1er octobre 2012 – Blog personnel
(Traduction : greygjhart, ZeHiro, Yho, Evpok, aKa, FredB, Simounet, Mounou, Coyau, Isdf, peupleLa, thot, HgO, salelodenouye)

Le 27 septembre 2012, Mozilla a lancé la première version bêta de **Persona**. Persona est un système d'authentification d'apparence similaire à OpenID et oAuth, mais qui s'en distingue d'un point de vue technique et fonctionnel.

Nous avons eu la chance de nous associer à Mozilla pour créer la version bêta de The Times Crosswords lors du lancement du Mozilla App Store. Nous sommes ainsi l'un des premiers services tiers à avoir intégré Persona de Mozilla (qui s'appelait encore BrowserID à l'époque), Mozilla en a même fait une vidéo.

Pourquoi un nouveau système d'identification ?

Passons en revue quelques-uns des problèmes d'OpenID et oAuth :

- OpenID utilise des URLs en tant qu'identifiants.
    - Même si fondamentalement l'idée est bonne, cela peut être

déroutant pour les utilisateurs. Le risque ? Se voir demander de « choisir un service d'identification » comme Google, LiveJournal, etc. alors qu'en fait vous n'êtes pas du tout en train de vous identifier avec eux.
    - La plupart des sites voudraient que vous leur fournissiez au moins une adresse email pour pouvoir vous contacter. Il faudra donc presque toujours en passer par une étape supplémentaire lors de la première connexion.
- OpenID est un système de connexion incohérent : il faut que vous quittiez complètement le site où vous étiez puis y retourner après vous être authentifié avec un outil tiers. On peut en dire autant d'oAuth (même si certaines de ses implémentations permettent une connexion en un seul clic, comme Twitter).
- oAuth est complexe à implémenter pour les développeurs : cela nécessite le stockage et la gestion de jetons . Il y a également plusieurs versions du protocole, et parfois une authentification supplémentaire (les jetons de rafraîchissement de Google par exemple).
- Tant OpenId qu'oAuth permettent à votre service d'identification (qu'il s'agisse de Google, Facebook, Twitter) de pister chaque site sur lequel vous vous inscrivez.

Comment Persona et BrowserID résolvent-ils ces problèmes ?

- Ils utilisent les adresses email au lieu des URLs. Non seulement les adresses email sont plus faciles à mémoriser, mais vous pouvez aussi utiliser une adresse email par identité ; par exemple votre adresse email de travail ou personnelle.
- Mozilla Persona utilise une « popup », ainsi vous ne quittez pas le site internet sur lequel vous êtes. Mieux encore, si le navigateur supporte le protocole BrowserID, vous n'aurez rien à faire.
- Persona/BrowserID est en général géré par les navigateurs (que ce soit en JavaScript ou en natif) et en tant que développeur vous n'aurez qu'à vérifier que l'utilisateur est bien celui qu'il dit être. Ceci doit être fait sur votre propre serveur, mais peut être implémenté avec à peine plus qu'une requête cURL.
- En plaçant le navigateur au centre du processus d'authentification, les services d'identification ne peuvent pas tracer les visites des utilisateurs,

mais ils permettent tout de même aux sites visités de vérifier leur identité. On y parvient en incorporant une cryptographie à clé publique dans le protocole.

Bien entendu, Persona ne vous sera d'aucune utilité quand vous aurez besoin d'accéder à des ressources tierces authentifiées comme vos données Twitter, mais ce n'est pas son rôle. Et c'est la ligne de démarcation qu'il faut entre votre identité et vos données.

Comment ça marche ?

Ce qu'il y a de bien avec la solution Persona de Mozilla, c'est qu'elle se décompose en deux niveaux. Le premier est le service BrowserID amorcé : Persona. Le second est le protocole d'identification en lui-même : BrowserID. En concevant ainsi l'amorçage du protocole, Mozilla évite les problèmes de prise en main et le rend attractif pour les développeurs.

BrowserID

Dans un monde idéal où BrowserID serait massivement adopté, voici ce qui se passerait au moment où vous voulez vous connecter à un site Web :

1. Vous cliquez sur « Connexion ».
2. Votre navigateur vous demande avec quelle adresse email vous voulez vous identifier.
3. Vous êtes connecté.

Et voici ce qui se passerait dans le détail :

1. Vous cliquez sur « Connexion » sur un site désigné ci-dessous par Service Tiers.
2. Votre navigateur vous demande avec quelle adresse email vous voulez vous identifier.
3. Votre navigateur contacte votre Service d'Identification (désigné dans la suite par S.I, par exemple Gmail) en utilisant vos identifiants (adresse email et clé publique) et demande un certificat signé.
4. Optionnel : Votre S.I. vous demande de vous inscrire (avec les habituels identifiant/mot de passe pour cela).
5. Votre S.I. envoie à votre navigateur un certificat signé qui dure 24 heures.
6. Votre navigateur génère une « Assertion ». Elle fait preuve que vous êtes

le légitime détenteur de votre adresse email : générée à partir de votre clé privée (stockée dans le navigateur), elle contient le domaine du site pour lequel vous vous authentifiez ainsi qu'une date d'expiration.
7. L'assertion et votre certificat signé sont tous deux envoyés au Service Tiers.
8. À l'aide de votre clé publique (fournie par votre navigateur), le Service Tiers vérifie que votre Assertion semble correcte.
9. Le Service Tiers demande la clé publique de votre S.I. (mais n'envoie aucune information sur l'utilisateur) et s'en sert pour vérifier que le certificat, envoyé par votre navigateur, est également correct.
10. Vous êtes connecté.

Plutôt long le processus ! Mais au moins, il est décentralisé, sécurisé et respecte votre vie privée (votre S.I. ne peut pas savoir quels sites vous êtes en train de visiter).

Si c'était la seule façon d'implémenter BrowserID, son déploiement serait compromis. C'est là que Persona entre en piste.

Mozilla Persona

Mozilla Persona est une application tierce qui fournit une interface de programmation (API) REST plutôt cool pour cacher toute cette cryptographie à clé publique.

- Persona gère la vérification via les services d'identification et agit également comme service d'identification à part entière pour les services tiers qui n'en possèdent pas (ce qui est la cas de tous les fournisseurs d'adresses email actuellement).
- Persona fournit une interface de programmation de type REST pour valider chaque assertion.

En traitant toute la partie cryptographie, implémenter BrowserID ne requiert que quelques lignes de JavaScript (le bouton de login et les callbacks en POST pour envoyer l'assertion au serveur) et une requête cURL (pour valider l'assertion).

Envie de l'utiliser ?

Commencez par jeter un œil sur le Persona Quick Setup *(NdT : Installer Rapidement Persona)* qui vous fournira les instructions pour ajouter Persona à

votre site Web, avec des exemples en JavaScript et une implémentation en Python de la vérification de l'assertion (c'est vraiment très simple). Le tout en une soixantaine de lignes de code.

Nous vous suggérons ensuite de consulter le guide des bonnes pratiques pour vous assurer que vous ne faites rien de travers.

NdT : En annexe deux liens de mozilliens francophones :

- *BrowserID par Pierre Rudloff*
- *BrowserID, implémentation en Java côté serveur par Flaburgan*