



Conditions Générales Techniques de Nearbee

Juin 2010

SOMMAIRE

<u>1.</u>MODELE TECHNIQUE.....	3
<u>2.</u>INFRASTRUCTURE D'HEBERGEMENT	5
2.1. Conditions techniques et moyens mis en œuvre pour l'infrastructure.....	5
2.2. Sécurité	6
2.3. Dispositif de Sauvegardes quotidiennes.....	7
2.4. Robots de monitoring et mécanisme d'alertes.....	8
2.5. Statistiques	8
2.6. Plan de reprise d'activité	9
<u>3.</u>ARCHITECTURE DU PROGICIEL	10
<u>4.</u>DISPOSITIF DE DEVELOPPEMENT ET D'EXPLOITATION DU LOGICIEL	11
4.1. Développement	11
4.2. Processus de validation de code	11
4.3. Processus de mise à jour	11
<u>5.</u>Authentification.....	12
5.1. Authentification à la plateforme Nearbee.....	12
5.2. Authentification Email.....	12
5.3. Identification des utilisateurs.....	13
5.4 Application d'un serveur Frontal certificats SSL.....	13
5.5 Authentification SSO	14
<u>6.</u>Services Web	16
6.1. Définition.....	16
6.2. <i>RESTful</i> API	16
6.3. XML-RPC	17
6.4. <i>WebDav</i>	18

Nearbee est un éditeur de solutions logicielles collaboratives destinées aux entreprises et administrations. Ces solutions logicielles sont commercialisées sous la forme de plateformes et adressent plusieurs types d'enjeux : collaboratifs, communautaires, éditoriaux, transactionnels. Il peut s'agir d'améliorer la circulation et le partage des informations et connaissances dans une organisation, d'améliorer la visibilité sur le web pour collecter de nouveaux clients, de promouvoir des communautés professionnelles ou encore d'accélérer les processus de commercialisation des offres d'une entreprise.

Pour mieux répondre aux enjeux de mobilité, de coûts et de déploiement, Nearbee a choisi le modèle SAAS en tant que solution de livraison de ses logiciels, privilégiant ainsi les atouts du réseau internet.

Description du modèle SaaS

Définition selon Wikipedia

"Le **logiciel en tant que service** ou en anglais le **Software as a Service (SaaS)** est un concept consistant à proposer un abonnement à un logiciel plutôt que l'achat d'une licence. Avec le développement des Technologies de l'information et de la communication, de plus en plus d'offres **SaaS** se font au travers du web. Il n'y a alors plus besoin d'installer une application de bureau ou client-serveur. Ce concept, apparu au début des années 2000, prend la suite de celui du fournisseur de service d'application (« *application service provider* » - ASP)."

Le modèle SaaS est donc un modèle de livraison de solution logicielle, où l'éditeur fournit à ses clients, les moyens d'infrastructure et de support en plus des fonctionnalités du logiciel. Les clients ne paient pas pour posséder le logiciel en lui-même mais plutôt pour l'utiliser. Ils l'utilisent soit directement via l'interface disponible, soit via des API fournies (souvent réalisées grâce aux *WebServices* ou à l'architecture REST (*Representational state transfer*)), permettant d'utiliser les fonctionnalités à l'intérieur d'autres solutions logicielles.

Une **interface de programmation** (*Application Programming Interface* ou **API**) est un ensemble de fonctions, procédures ou classes mises à disposition des programmes informatiques par une bibliothèque logicielle, un système d'exploitation ou un service. La connaissance des API est indispensable à l'interopérabilité entre les composants logiciels.

SaaS est donc la livraison conjointe d'applications logicielles, de moyens, de services et d'expertise qui permettent aux entreprises d'externaliser intégralement un aspect de leur système d'information (Collaboration, Gestion de fichiers ou de projets, messagerie électronique, relation client...) et de l'assimiler à un coût de fonctionnement plutôt qu'à un investissement.

C'est aussi, selon le contrat de services proposés, **la possibilité de mettre en œuvre un niveau de qualité de service élevé** afin de proposer à l'ensemble des utilisateurs une assistance afin d'accéder aux bénéfices des applications.

SaaS peut être vu comme le pendant "*business*" de l'architecture orientée service (SOA).

Par conséquent, le client peut accéder à distance à des applications hébergées, facturées à l'usage. Aujourd'hui, les utilisateurs sont conscients que la mutualisation des ressources permet aux éditeurs et hébergeurs de proposer un niveau de sécurité que peu d'entreprises peuvent s'offrir.

Les bénéfices :

L'utilisation de solutions SaaS en entreprise permet un meilleur contrôle des charges techniques. L'ensemble des solutions techniques étant confiées à l'éditeur, le coût devient fixe. Le prix englobe le coût des licences des logiciels, de la maintenance et de l'infrastructure.

Il est courant que le coût d'une solution SaaS soit moindre que l'acquisition des licences et un déploiement en interne, via la suppression des coûts de déploiements et la mutualisation des services réalisés par l'éditeur du logiciel.

Les avantages du SaaS présentent un impact budgétaire et financier. Si les coûts totaux d'acquisition et de maintenance de la solution (TCO, *total cost of ownership*) s'avèrent moindres, la dépense passera dans les charges de fonctionnement (OPEX), contrairement à une acquisition traditionnelle de licence qui est généralement passée en immobilisation (CAPEX, hors maintenance).

Un autre avantage pour les entreprises est la rapidité de déploiement. Les solutions SaaS étant déjà préexistantes, le temps de déploiement est extrêmement faible. Dans la même lignée, les solutions SaaS sont **généralement très flexibles et permettent de mettre en place des solutions « On-Demand ».**

Un autre avantage est de réduire la consommation électrique en permettant la mutualisation des ressources sur des serveurs partagés par plusieurs entreprises ainsi que l'usage de PC à faible consommation ou Netbook équipés de systèmes d'exploitation moins gourmands en ressources tels que Linux et d'un simple Navigateur Web.

Les évolutions fonctionnelles

L'hébergement en un point unique des applications permet à l'éditeur **de procéder plus rapidement et régulièrement à des mises à jour du code logiciel** et à l'insertion de nouveautés fonctionnelles, qui pourront être activées ou non par le client.

En conclusion, le développement des solutions SaaS a accompagné l'évolution des entreprises, qui se concentrent sur leur cœur de métier en externalisant les fonctions support. Le mode SaaS répond également aux exigences des directions financières qui contrôlent leur budget selon l'usage et disposent d'une solution plus flexible qu'en mode achat de licence.

Enfin, le mode SaaS est également une bonne réponse aux exigences de mobilité croissante des utilisateurs, au développement du télé-travail et à l'éclatement des structures, qui impliquent de faire communiquer et collaborer des sites distants.

Nearbee a confié à la société IntermédiaSud, SAEM basée à Castres Mazamet, l'hébergement de ses serveurs abritant les logiciels et applications de ses solutions de plateformes.

Pour plus d'informations : <http://www.e-teleport.net>

L'hébergement de nos services se fait chez IntermédiaSud, et se conforme aux exigences de fiabilité, sécurité et haute disponibilité.

Nearbee loue chez IntermédiaSud les prestations d'hébergement pour ses serveurs, d'infrastructures d'accès et de sécurité, et d'exploitation des systèmes d'infogérance.

2.1. Conditions techniques et moyens mis en œuvre pour l'infrastructure

Cette infrastructure réseau garantit des services Internet de haut niveau grâce à :

- Une connectivité fibre optique sur deux cheminements géographiquement distincts.
- Capacité de bande passante Internet 200 Mbps (1Giga si nécessaire).
- Accès au réseau Internet multi-opérateurs.
- Protocole de routage BGP4.
- Optimisation des routes sur les infrastructures internationales.
- Niveau de disponibilité de 99,99%.
- Assignation et gestion d'adresses IP fixes en Ipv4 et Ipv6.
- Sécurisation des flux IP redondée (pare-feu, antivirus, antispam).
- Gestion de DNS primaires et secondaires.
- Gestion de la qualité de service, monitoring, métrologie des flux.
- Maintenance matérielle de premier niveau sur les équipements hébergés.

Les locaux du *data-center* consistent en :

- 200 m² de salles blanches aux normes *Carrier Class*
- Alimentation électrique sécurisée par deux sources EDF indépendantes et un groupe électrogène.
- Onduleur permettant l'alimentation des baies dédiées à l'hébergement.
- Puissance électrique de 16A par baie (capacité totale de 80 KVA) ou 32A en option.
- Régulation thermique redondante de 19°C (+/-2).
- Contrôle hygrométrique des salles - humidité relative 55%.
- Protection anti intrusion.
- Télésurveillance de l'accès général aux locaux.
- Badge d'accès physique 24h/24, 7j/7.

2.2. Sécurité

La sécurité des données est l'un des aspects les plus importants de notre organisation. Nous veillons dès lors à l'optimiser à chaque niveau de notre infrastructure.

Entre autres, nous pouvons énumérer les mesures suivantes :

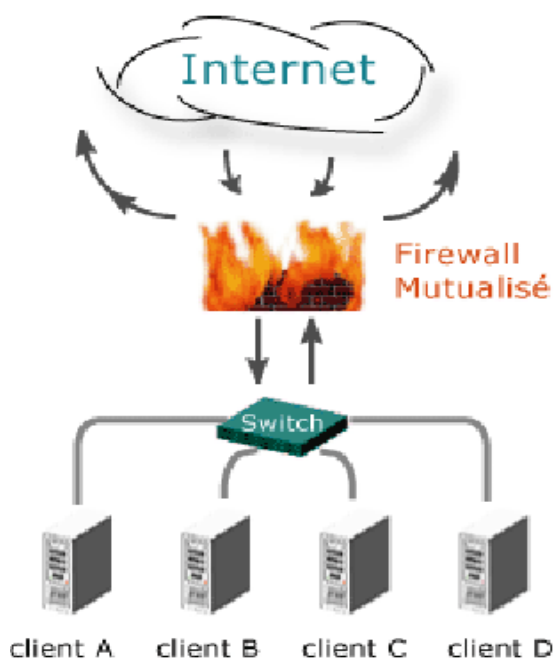
- Mises à jour régulières des systèmes et codes applicatifs pour éliminer tout code connu pour sa vulnérabilité.
- Utilisation de systèmes de type Unix, reconnus pour leur sécurité.
- **Deux niveaux de pare-feu.**
- Systèmes de détection d'intrusions et journaux de pare-feu.
- Cryptage des données sensibles et mots de passe.

Filtrage des flux

Nos *firewalls* sur les espaces d'hébergement sont basés sur *Netfilter* avec une gestion par *IPtables*. *Netfilter* est la solution complète de firewall la plus proche des couches basses du réseau ce qui permet une granularité très fine avec des temps de traitement extrêmement faibles.

Elle permet de faire du filtrage évolué sur les adresses source et destination, port sources et destination, du *firewalling statefull* (gestion des sessions mère/filles, par état. ex: ftp actif/passif) de la translation de port, d'adresse (NAT), du filtrage niveau 2 (MAC), la modification de paquets à la volée (typiquement la correction de mss sur les connexions ppp).

Netfilter est reconnue comme étant une solution fiable disposant de très nombreuses options permettant des règles d'une grande précision en IPV4 et IPV6, y compris dans la gestion des VPNs.



La sécurité IP est faite par les routeurs en limite de zone d'hébergement. La sécurité est faite au plus proche du client et non pas de manière globale.

La sécurité des flux s'effectue donc par le biais d'un ensemble de modules basés sur du *Netfilter* sur le routeur le plus proche du client.

La règle par défaut est de jeter tous les paquets transitant par le routeur. Une règle est généralement appliquée concernant les flux icmp qui sont souvent ouverts. Elle consiste à n'autoriser que 10 paquets icmp par seconde afin d'éviter des dénis de services.

L'ouverture de flux est fait au cas par cas suivant les demandes du client mais en général on autorise les flux sortants (ie : initié par le Lan du client).

Il y a possibilité de loguer tous les flux ou des flux ciblés (mais cela risque de rajouter une latence, normalement fait par le serveur lui même).

Détection d'intrusion

Le réseau et les serveurs sont munis de sondes logicielles opérant une analyse systématique du trafic réseau à la recherche de motifs connus d'attaques (vers , virus, scan, paquets visant des *buffer overflow...*) , ainsi que des comportements suspects ou contraires à la politique de sécurité en vigueur (ex: tentative de connexion SSH à partir d'un réseau non autorisé ...)

2.3. Dispositif de Sauvegardes quotidiennes

Notre architecture implique systématiquement un mécanisme de sauvegardes quotidienne set sécurisées de toutes les données.

Ces sauvegardes s'effectuent selon deux mécanismes:

- 1) Réplication base de données sur un deuxième serveur sur le même site géographique, chez IntermediaSud.
- 2) Sauvegarde quotidienne avec transfert (sécurisé et crypté) des données vers un deuxième site géographique (Telehouse 2)

Explications:

La réplication base de données sur un deuxième serveur (sur le même site géographique) constitue un degré de plus dans la disponibilité des applications et des données.

Elle permet d'avoir en permanence une deuxième copie de la base de données répliquée sur un deuxième serveur en temps-réel.

Les sauvegardes quotidiennes s'effectuent sur cette réplique très tôt le matin (entre 2h et 4h du matin), pour être complètement transparentes à l'utilisateur final en ne causant aucun arrêt ou ralentissement de la plateforme en production, et restent disponibles en cas de panne matérielle sur le premier serveur.

La sauvegarde quotidienne avec transfert des données vers un deuxième site géographique est opérée systématiquement sur toutes les plateformes Nearbee.

Elle consiste en une sauvegarde totale des données, et leur transfert sécurisé et crypté sur deux autres serveurs, l'un se trouvant sur le même site géographique et l'autre se trouvant sur un autre site géographique distant (ces deux sites sont sur Paris et Castres), constituant ainsi une protection maximale des données en cas de défaillance matérielle, logicielle, ou même de catastrophe naturelle.

Remarque:

Un mécanisme d'alerte par mail signale quotidiennement le bon déroulement de la sauvegarde.

2.4. Robots de monitoring et mécanisme d'alertes

Afin de garantir un excellent niveau de qualité de service aux utilisateurs, Nearbee a déployé un dispositif redondé d'outils de supervision composé de plusieurs mesures.

Ces informations sont centralisées et qualifiées en temps réel.

Ensuite, elles sont analysées, traitées puis restituées sous forme d'indicateurs graphiques, tickets d'incident, tableau de synthèse, etc.

Pour superviser de manière exhaustive et très fine la qualité de service, Nearbee a développé et déployé plusieurs robots, permettant de simuler les scénarios d'usages des utilisateurs.

Il existe différents types de robots de monitoring développés sur mesure pour nos besoins.

En effet, en plus des logiciels "classiques" de monitoring, nous avons développé plusieurs programmes surveillant la disponibilité de nos plateformes.

Le premier type est dit: **Test serveur**

Il s'agit de tests de ping assurant une alerte au personnel technique dans le cas d'une défaillance de serveur ou défaillance réseau.

Le second type est dit : **Test applicatif**

Il teste certains contenus types dans une plateforme, et signale à l'équipe technique un éventuel dysfonctionnement sur nos serveurs d'application ou base de données.

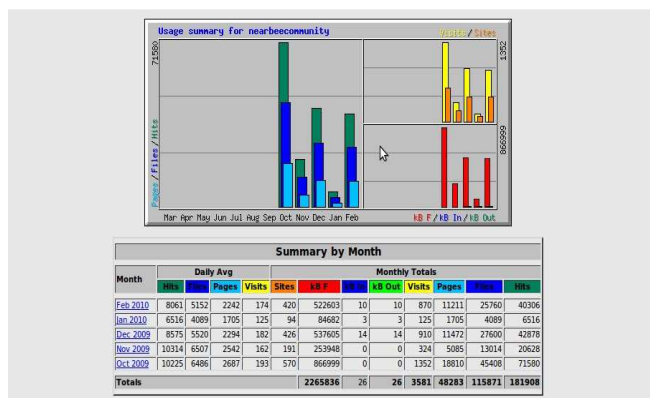
Le troisième type de robots est dit: **Test de latence**

Il vérifie les temps de réponses de pages témoins, et permet de signaler une éventuelle lenteur d'affichage de page sur une plateforme publique ou privée.

→ L'équipe technique est alertée de façon automatique 24h/24 et 7 jours sur 7 en cas de panne technique sur une plateforme production, par email et SMS.

2.5. Statistiques

Les plateformes publiques ou privées, sont systématiquement fournies avec une interface de statistiques permettant d'évaluer l'audience, selon une multitude de critères.



exemple d'un écran de statistiques -

Les statistiques sont aussi faites sur certaines plateformes avec *Google analytics*, selon le besoin du client.

2.6. Plan de reprise d'activité

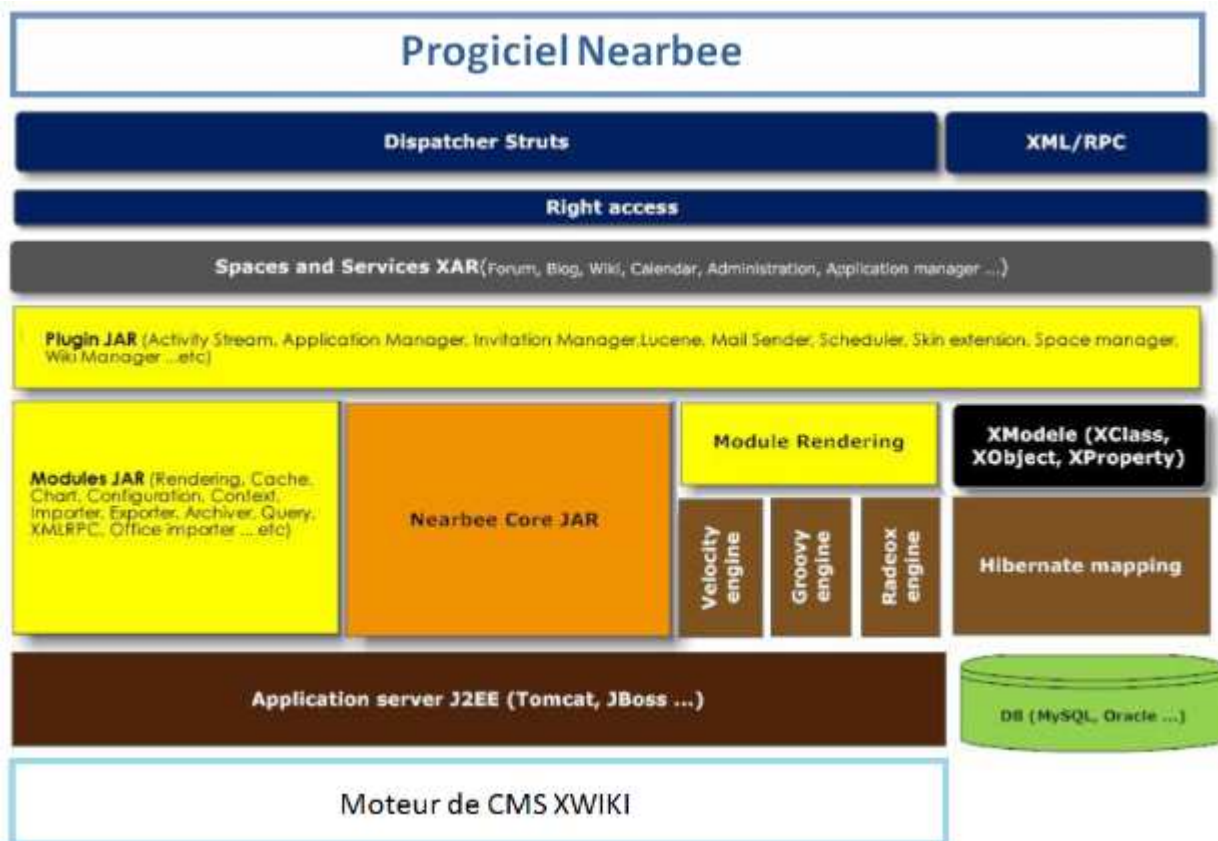
Nous avons établi un plan de reprise d'activité pour parer à toute éventualité, et pouvoir reprendre le service en cas de force majeure, ou catastrophe naturelle sur notre premier site.

Ce qui nous permet de récupérer les sauvegardes quotidiennes des plateformes, faites sur le deuxième site géographique et pouvoir rétablir le service dans un délai raisonnable. (48 heures)

3

ARCHITECTURE DU PROGIciel

En respectant les règles et bonnes pratiques de développements, cette architecture technique permet une grande liberté d'intégration, un haut niveau de sécurité et de transparence, une pérennité dans la solution mise en place ainsi qu'une évolution importante suite aux nombreuses contributions de la communauté des partenaires techniques.



Développé principalement sous Java et sur un socle open source, le progiciel Nearbee permet de construire des dispositifs d'intégration de données et d'applications, puissants et sécurisés.



DISPOSITIF DE DEVELOPPEMENT ET D'EXPLOITATION DU LOGICIEL

4.1. Développement

Nos équipes de développement travaillent sur des serveurs dédiés aux développements. Cela permet une totale séparation physique des environnements "développement" de ceux de "production" permettant ainsi d'effectuer les développements de nouvelles fonctionnalités en toute sérénité.

Nous avons aussi mis en œuvre des environnements de pré-production permettant de finaliser les tests de migration et confirmer les attentes du client avant de mettre à jour une plateforme en production.

4.2. Processus de validation de code

Les développeurs en collaboration avec "l'administrateur systèmes et réseaux" disposent d'une multitude d'outils permettant d'auditer la qualité du code logiciel et d'identifier d'éventuels goulots d'étranglement, de définir les requêtes base de donnée, qui pourraient provoquer des lenteurs ou des blocages.

Les serveurs de "développements" sont aussi munis de sondes affichant toutes les informations (mémoire, threads, utilisation CPU..).

Les serveurs de "développements" intègrent un mécanisme de *stress-testing* pour afficher les performances potentielles d'une plateforme donnée.

Enfin, nous avons mis en place des techniques de *dump* mémoire et d'analyse de *dump* pour détecter les erreurs de type fuite mémoire (*memory leak*) ainsi que d'autres formes d'excès d'utilisation mémoire.

4.3. Processus de mise à jour

Le processus de mise à jour du code logiciel, est programmé pour minimiser au maximum l'interruption de service pour un client.

Un serveur de pré-production permet de migrer les données du client sur une nouvelle version.

Cet environnement de pré-production est vérifié par les testeurs puis validé par le client (qui continue d'utiliser sa plateforme production avec l'ancienne version).

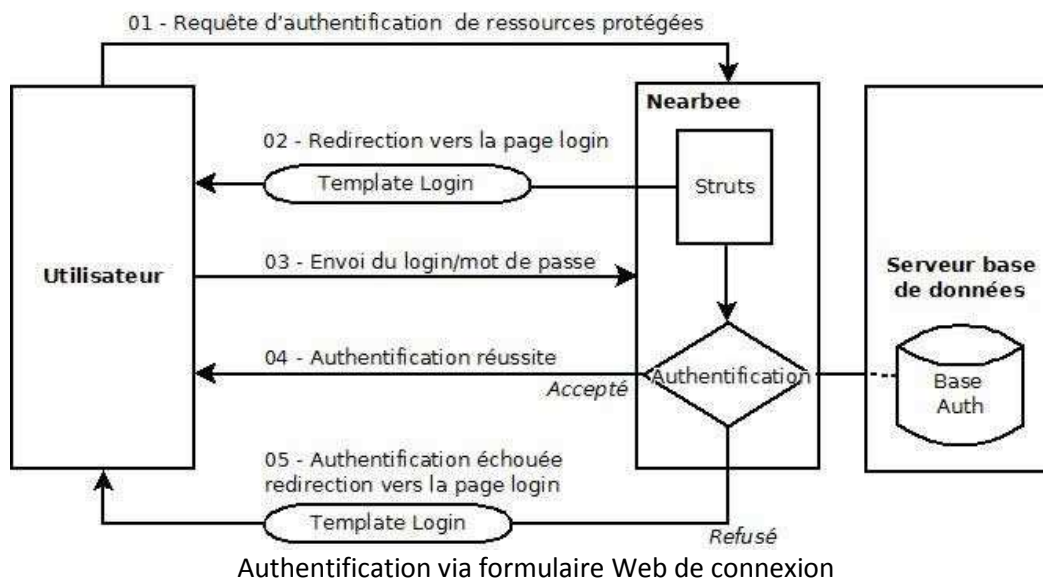
Après validation, l'administrateur peut engager la procédure de migration sur le serveur de production, le soir, ou le week-end (ou autre), selon la migration.

5.1. Authentification à la plateforme Nearbee

Lorsqu'on accède à des ressources protégées de la plateforme Nearbee, le conteneur active un mécanisme d'authentification lié à chaque ressource sécurisée.

Ce mécanisme est appelé *Form-based login authentication* traduit par : Authentification via un formulaire de connexion illustré ci dessous.

Il s'agit d'une possibilité puissante et pratique de la plateforme Nearbee, qui lui permet de personnaliser et d'adapter facilement l'authentification pour se brancher à des annuaires externes appartenant à d'autres plateformes en développant le plugin adéquat.



L'authentification via formulaire se déroule comme suit:

- 1) Un client demande l'accès à une ressource sécurisée.
- 2) Si le client est non authentifié, Struts le redirige vers une page de connexion.
- 3) Le client envoie le formulaire de connexion à la plateforme avec un email et un mot de passe.
- 4) Si la connexion réussit, la plateforme redirige le client vers la ressource demandée.
- 5) Si la connexion échoue, le client est redirigé vers une page d'erreur d'authentification.

5.2. Authentification Email

Contrairement au login ou pseudo, une adresse email est plus pratique à l'usage et plus pertinente. De ce fait, le mécanisme d'authentification avec le couple email/ password est aujourd'hui le plus utilisé dans les systèmes d'information ainsi que les plateformes collaboratives qu'elles soient de nature professionnelles comme :

- Viadeo > <http://www.Viadeo.com/>
- LinkedIn > <http://www.Linkedin.com/>
- xing > <http://www.xing.com/>

ou sociales comme :

- facebook > <http://www.facebook.com>
- Twitter > <http://www.twitter.com/>
- Friendster > <http://www.friendster.com/>
- MySpace > <http://fr.myspace.com/>
- Trombi > <http://www.trombi.com/>

Pour inscrire un utilisateur sur une plateforme Nearbee, différents champs nécessaires à son identification sont saisis et enregistrés, notamment son adresse email. Ensuite, vient l'étape indispensable de validation de l'email par la plateforme afin qu'elle puisse vérifier et reconnaître l'identité unique de chaque utilisateur inscrit. En partant du principe que chaque usagé d'un système de messagerie électronique possède une adresse email unique quelque soit le système qu'il utilise, le couple email/password est ainsi utilisé comme mécanisme d'authentification dans la plateforme Nearbee.

5.3. Identification des utilisateurs

Comme dans la plupart des applications web, la plateforme Nearbee vérifie les sessions ouvertes via des cookies sécurisés. A chaque demande d'accès aux ressources protégées (par exemple une page wiki), une validation du cookie est exécutée par un module de sécurité qui examine l'authentification et l'autorisation de l'utilisateur. Si le cookie est expiré ou corrompu, la demande d'accès sera refusée et l'utilisateur sera redirigé vers la page d'authentification.

La plateforme Nearbee offre deux moyens afin de sécuriser les cookies :

1 - Clés de chiffrement Cookie

Quand un utilisateur se connecte à la plateforme, des données cryptées sont enregistrées sur sa machine, notamment : son nom d'utilisateur et son mot de passe. Les cookies sont cryptées en utilisant deux paramètres de configuration spécifiques :

- validationKey
- encryptionKey

Ses paramètres permettent donc de crypter l'information stockée dans les cookies, et ils ne sont accessibles que par l'administrateur système de la plateforme Nearbee.

2 - Crypter les cookies via l'adresse IP

Bien que les informations d'authentification ne puissent être extraites à partir d'un cookie crypté, ce dernier peut être volé et exploité afin de simuler des accès dans une autres machine. La plateforme Nearbee bloque ce genre d'attaque en cryptant les cookies via l'adresse IP de l'utilisateur, ainsi les cookies volés et exploités dans une machine tiers sont systématiquement rejetés par le module de sécurité.

5.4 Application d'un serveur Frontal certificats SSL

SSL est l'acronyme de "Secure Socket Layer". Il s'agit d'un protocole qui permet aux navigateurs et aux serveurs Web d'échanger des données via un canal de communication sécurisé. Les données échangées sont cryptées avant d'être envoyées et ensuite décryptées à la réception avant le traitement. Le navigateur et le serveur cryptent donc tout le trafic avant transmission.

L'application d'un serveur frontal de certificat SSL permet de renforcer la sécurité d'accès aux ressources de la plateforme Nearbee et assure :

L'authentification :

Lors d'une première tentative de communication avec le serveur Nearbee, une négociation SSL se déclenche entre le navigateur et le serveur.

Le navigateur s'assure en premier lieu de l'identité du serveur Nearbee en examinant si le serveur avec lequel il communique est bien celui qu'il prétend être.

Cette vérification s'effectue au niveau du certificat émit par le serveur au navigateur, qui contient les paramètres **PKI** (Public Key Infrastructure). Ces paramètres font référence aux sociétés externes auxquelles le navigateur fait implicitement confiance comme : VeriSign, Thawte ... etc.

Une fois l'authenticité du certificat validée, l'échange des données entre le navigateur et le serveur Nearbee est ouvertement autorisé.

La confidentialité :

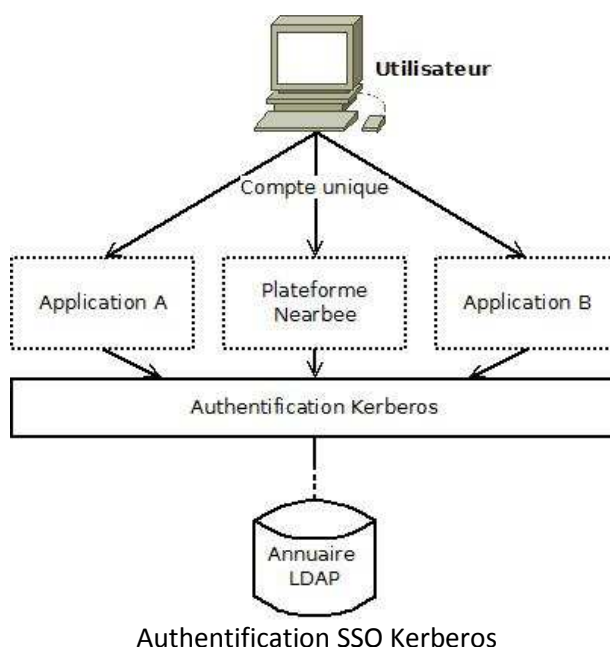
Lorsque les données sont transmises entre le navigateur et le serveur Nearbee, des tiers peuvent les visualiser en interceptant les données cryptées, dans ce cas, ces tiers parties ne peuvent en aucun cas déchiffrer les données cryptées qui restent donc strictement confidentielles.

L'intégrité :

Lorsque les données sont transmises entre le navigateur et le serveur Nearbee, des tiers peuvent les intercepter. Dans ce cas, SSL garantit que les données qui transitent ne peuvent être modifiés par une tierce partie.

5.5 Authentification SSO

Basé sur le SSO Kerberos, ce modèle fédère à la plateforme Nearbee différentes applications web qui utilisent un annuaire LDAP comme mécanisme d'authentification. Il offre la possibilité d'utiliser un seul couple *email/password* pour accéder à l'ensemble de l'écosystème applicatif associé, en simplifiant les procédures d'authentification pour les utilisateurs, comme illustré ci-dessous :



Le schéma indique que l'utilisateur se connecte avec le même compte à toutes les applications. Le SSO "Single Sign On" ou authentification unique, est une technique informatique qui permet à un utilisateur de s'authentifier une première fois et de pouvoir ainsi accéder à toutes les applications et ressources dont il a le droit, sans nouvelle authentification.

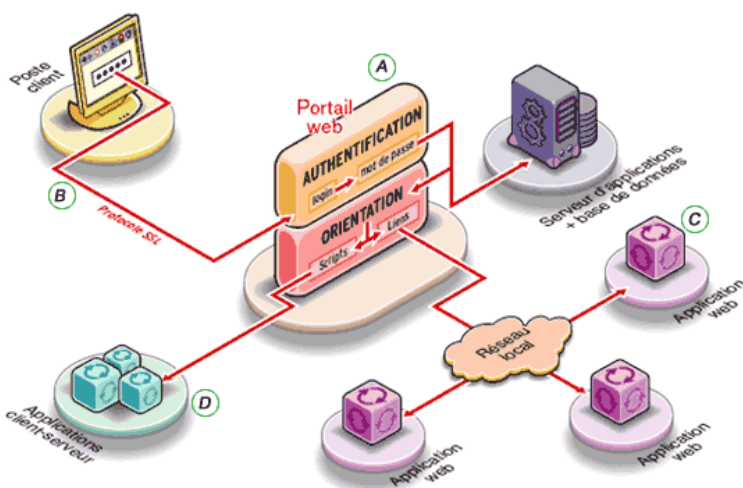
Ce modèle propose une architecture particulière qui repose sur un référentiel d'authentification commun (Annuaire LDAP) à l'ensemble de l'écosystème applicatif fédéré dont voici quelques avantages :

- Conformité et intégrité des différentes logiques d'authentifications des différentes applications connectées à la plateforme Nearbee.
- La mise à disposition des utilisateurs des applications hétérogènes de façon homogène, cohérente et conviviale.
- L'agrégation active des comptes propres aux applications tiers dans la plateforme Nearbee.
- Simplicité d'administration en termes de déploiement et de gestion des comptes utilisateurs.
- Offrir aux utilisateurs un bouquet de services pour un seul mot de passe à retenir.

Il faut savoir que :

- La technique de *single sign on*, SSO, ou authentification unique est utile si on dispose de nombreuses applications et/ou ressources nécessitant chacune une authentification.
- Les SSO sont rarement déployés simplement pour faciliter la vie des utilisateurs. Ils s'intègrent généralement à un projet de sécurité plus large, dans lequel ils ne constituent qu'un élément secondaire, voire un simple bonus.
- La technique *single sign on*, SSO, permet de ne pas avoir à s'authentifier pour chaque application ou ressource. Cette technique apporte donc un plus grand confort à l'utilisateur.
- En général, il s'agit de un portail unique grâce auquel, une fois identifié sur son serveur LDAP, l'utilisateur découvre une page Web construite dynamiquement en fonction de son profil. Elle lui présente les liens vers les différentes applications auxquelles il a le droit d'accéder, et le portail se charge de propager l'authentification initiale.

Exemple :



6.1. Définition

Un service web (ou service de la toile) est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur Internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, et en temps réel.

Parmi les API et interfaces qui permettent de fournir des services web, sur les plateformes Nearbee, on trouve :

- *RESTful Api*
- *XML-RPC Api*
- *WebDAV*

La plateforme Nearbee utilise trois technologies afin d'exposer ses fonctionnalités à d'autres applications via les *Web Services* dont : *RESTful*, *XML-RPC* et *WebDAV*

Dans ce cas, Nearbee partage ses contenus (Wikis, Blogs, Actualités, Evénements, etc.) et les expose comme référentiels pour d'autres applications, à titre d'exemple le CMS *Joomla* ou *OSCommerce* tous deux basées sur PHP ou encore SharePoint de Microsoft basé sur DOTNet.

Dans la mesure où la plateforme intègre des technologies *Web Services*, on peut afficher, modifier, gérer et interagir avec des contenus Nearbee depuis d'autres environnement et applications tiers.

6.2. RESTful API

REST (*Representational State Transfer*) est une manière de construire une application pour les systèmes distribués comme le *World Wide Web*.

REST n'est pas un protocole ou un format, c'est un type d'architecture. Le type architectural original du Web. Dans cette architecture, un composant lie ou modifie une ressource en utilisant une représentation de cette ressource.

L'application de cette architecture au Web se comprend sur quelques principes simples :

- l'URI est important : *connaître l'URI doit suffire pour nommer et identifier une ressource*
- HTTP fournit toutes les opérations nécessaires : GET, POST, PUT et DELETE, essentiellement
- Chaque opération est auto-suffisante : *il n'y a pas d'état*
- Utilisation des standards hypermédia : HTML ou XML

Dans le cas de nos plateformes :

Les ressources dans Nearbee sont :

- 1) Via RESFull Nearbee partage les ressources suivantes avec d'autres applications externes:
 - Pages (Wiki, Blog)
 - Attachements (Fichiers attachés)
 - Propriétés (nom prénom d'un utilisateur, titre d'un document, etc.)
 - Espaces (Evénement, Actualités, Espaces Wiki, etc.)...

- 2) La représentation se fait via XML en utilisant la balise <link>. Cette balise a deux paramètres importants qui sont REL et HREF :
 - REL spécifie la sémantique de la ressource
 - HREF est l'URI de la ressource
- 3) Nearbee fournit une API qui permet d'exposer / utiliser des services RESTful permettant entre autre :
 - L'authentification
 - L'envoi de représentations
 - De dépasser les limitations du navigateur (possibilité d'utiliser la méthode PUT en http)

6.3. XML-RPC

XML-RPC est un protocole **RPC (Remote procedure call)** : une spécification simple et un ensemble de codes, qui permettent à des processus s'exécutant dans des environnements différents de faire des appels de méthodes à travers un réseau.

XML-RPC permet d'appeler une fonction sur un serveur distant à partir de n'importe quel système (Windows, Mac OS X, GNU/Linux) et avec n'importe quel langage de programmation. Le serveur est lui même sur n'importe quel système et est programmé dans n'importe quel langage. Cela permet de fournir un Service web utilisable par tout le monde sans restriction de système ou de langage.

Les processus d'invocation à distance utilisent le protocole HTTP pour le transport des données et la norme XML pour le codage des données. XML-RPC est conçu pour permettre à des structures de données complexes d'être transmises, exécutées et renvoyées très facilement.

Dans le cas de nos plateformes :

- 1) XML-RPC facilite la construction d'applications tierces, qui peuvent se connecter aux plateformes Nearbee. Par exemple cette fonctionnalité permet d'intégrer Nearbee à d'autres applications Java existantes. En d'autres termes XML-RPC permet de doter d'autres applications de fonctionnalités Nearbee.
- 2) Nearbee fournit une Api qui permet :
 - Accès Authentifié / Accès anonyme
 - Espaces : *récupération, création, suppression*
 - Pages: *récupération, affichage, création, historique, modification, recherche et suppression*
 - Attachements: *récupération (téléchargement), création (upload), déplacement/renommage et suppression*
 - Commentaires: *récupération, création et suppression*
- 3) Plusieurs Clients RPC sont disponibles permettant ainsi, l'utilisation de l'API à partir de différents environnements :
 - *Apache XML-RPC Client (JAVA)*
 - *Groovy Client*
 - *Perl Client*
 - *Python Client*
 - *Ruby Client*

6.4. WebDav

WebDAV (Web-based Distributed Authoring and Versioning) est un protocole (plus précisément, une extension du protocole [HTTP](#)) défini par le groupe de travail [IETF](#) éponyme. *WebDAV* permet de simplifier la gestion de fichiers avec des serveurs distants. Il permet de récupérer, déposer, synchroniser et publier des fichiers (et dossiers) rapidement et facilement. L'objectif principal de *WebDAV* est de rendre possible l'écriture à travers le web et pas seulement la lecture de données.

Dans le cas de nos plateformes :

La fonctionnalité *WebDAV* permet d'exposer du contenu Nearbee (attachements, contenu de pages) à travers le protocole *WebDAV protocol*.

Ceci rend possible l'utilisation de clients *WebDAV* comme DAVExplorer. Les explorateurs de fichiers tels que Windows Explorer, le Finder (Mac) ou encore Nautilus (Linux), permettant ainsi d'explorer et d'éditer directement du contenu Nearbee comme s'il s'agissait de fichiers du système de fichiers local sur la machine.

Quelques bénéfices :

- Monter du contenu wiki sur le système de fichier en local.
- L'utilisation d'applications locales pour manipuler le contenu des pages wiki & attachements
- Effectuer facilement des opérations en Batch sur les documents (ex ajouter 100 attachements à une page wiki)

Nearbee est compatible avec les explorateurs suivants :

- [DAVExplorer](#) - toutes plateformes
- [WebFolders](#) - Windows XP
- [WebDAV Redirector](#) - Windows XP
- [NetDrive / WebDrive](#) - Windows XP
- [Nautilus - Gnome](#) (Linux)
- [Konqueror - KDE](#) (Linux)
- [Davfs2](#) - Linux

FIN