CAHIER DES CHARGES

Version finale

Révision du 03/03/2011
Projet n°22
Rédigé par: Julien GUERY, Hamed Brahane KY et Yann MAHE

SOMMAIRE

1INTRODUCTION.....3
1.1 Objet du document.....3
1.2 Portée du document.....3
1.3 Terminologie.....3
2 PRÉSENTATION GÉNÉRALE DU PROJET.....4
2.1 Analyse de l'existant.....4
2.2 Énoncé du besoin du client.....4
2.3 Définition du projet et des livrables.....4
2.4 Contexte du projet.....5
3 EXIGENCES SUR LE PRODUIT.....5
3.1 Capacités Fonctionnelles.....5
3.1.1 Acteurs mis en jeu.....5
3.1.2 Descriptions des fonctionnalités.....6
3.1.3 Interopérabilité.....8
3.2 Exigences non fonctionnelles.....8
3.2.1 Sécurité.....8
3.2.2 Rendement.....8
3.2.3 Maintenabilité.....8
3.2.4 Portabilité.....8
3.2.5 Facilité d'utilisation.....8
3.3 Exigences concernant le développement du produit.....8
3.3.1 Objectifs de délais.....8
3.3.2 Objectifs de coûts.....9
3.3.3 Exigences de réalisation.....9
4 SCÉNARIOS DES CAS D'UTILISATION.....9

1 INTRODUCTION

1.1 Objet du document

Ce document constitue le cahier de charges fonctionnel du projet de développement S2 intitulé *Développement d'un clone libre de DropBox*.
Le document présente le produit final attendu, les services qu'il doit offrir comme fonctionnalités, ainsi que toutes les exigences et contraintes qu'il doit satisfaire. On y trouvera également des cas et scénarios d'utilisation.

1.2 Portée du document

Ce document est destiné à expliciter et formaliser le besoin du client Matthieu Arzel dans le cadre du Projet Développement S2.

1.3 Terminologie

Vous rencontrerez les termes et abréviations suivants tout au long de la lecture. Il est donc nécessaire de comprendre leur signification et le contexte dans lequel nous les employons.

Terme	**Description**
Produit	Terme générique désignant la suite complète de logiciels à fournir à la fin du projet (serveur, client, outils de configurations, interface web, scripts, base de données ...).
Serveur SVN	Serveur utilisant un système de gestion de versions Subversion. C'est le serveur de stockage qui contiendra les fichiers synchronisés.
Client	Ensemble des applications à l'intention de l'utilisateur, destinés à interagir avec le serveur pour la synchronisation souhaitée des répertoires et fichiers. Il contient l'application cliente à proprement parlé et une interface graphique de configuration.
Serveur	Ensemble des applications à l'intention du super administrateur, destinés à interagir avec le SVN et les clients pour permettre la synchronisation souhaitée des répertoires et fichiers. Il se compose d'une application serveur à proprement parlé, d'une interface graphique de configuration, d'une interface web et d'une base de données.
Interface web	Portail web permettant à l'utilisateur selon ses droits d'ajouter/supprimer/modifier un administrateur/dépôt/client et de consulter l'historique des ajouts/suppressions/modifications de ses fichiers synchronisés.
Base de données	Support sur lequel seront stockés des informations utiles au bon fonctionnement du serveur tel que les identifiants et mots de passe des utilisateurs ou encore l'adresse du serveur SVN.
Super administrateur	Utilisateur hébergeant le serveur. Il est l'utilisateur possédant le plus de droits (notamment l'ajout/suppression/modification de dépôts et administrateurs) et chargé de créer les dépôts et leurs administrateurs.
Administrateur	Utilisateur chargé d'administrer un dépôt (ajout/suppression/modification des droits d'un client). Son compte est créé par le super-administrateur.

2 PRÉSENTATION GÉNÉRALE DU PROJET

2.1 Analyse de l'existant

Dropbox est un service de stockage et de partage de fichiers en ligne proposé par l'entreprise *Dropbox, Inc*.
Il permet de synchroniser automatiquement avec un serveur distant des répertoires choisis sur plusieurs machines utilisateurs. Le serveur distant qui stocke les fichiers est un serveur appartenant à *DropBox Inc*. Les fichiers deviennent donc accessibles par le web via n'importe quel navigateur web, mais aussi en utilisant un client multi-système d'exploitation (Linux, Mac OS, Microsoft Windows, iOS ...).
Les intérêts d'un tel logiciel sont évidents : la synchronisation automatique protège les données quasiment en temps réel, et elle permet de travailler à partir de plusieurs machines sur les mêmes fichiers de manière complètement transparente (sans avoir besoin de se connecter à un serveur pour accéder à des documents).

2.2 Énoncé du besoin du client

Un des gros inconvénients de *DropBox* est qu'il impose d'avoir une totale confiance dans le gestionnaire du serveur central qui stocke les fichiers. Ce serveur, qui est hébergé par *DropBox Inc.*, pourrait à tout moment accéder à vos données confidentielles.
Un autre inconvénient est que le logiciel est propriétaire, et devient payant pour plus de 2 Go d'espace de stockage.
De plus, le logiciel souffre d'autres défauts, en particulier le fait qu'il n'est pas possible de configurer et de personnaliser très proprement les répertoires à partager.
Par conséquent, concevoir et réaliser un logiciel capable de mimer les fonctionnalités de *DropBox* mais avec un serveur local permettrait de mettre à disposition des entreprises et universités qui le souhaitent un logiciel équivalent, avec l'avantage de la confidentialité des données synchronisés et d'un déploiement sur réseau local.

2.3 Définition du projet et des livrables

Le projet est intitulé *Développement d'un clone libre de DropBox*, et consiste en la mise en place d'un système de stockage, de partage et de synchronisation automatique de fichiers, tel que *DropBox*, mais avec le gros avantage de la confidentialité des données assurée grâce au déploiement du serveur de stockage dans un réseau local.
Ce système de synchronisation se constituera de deux applications (client,serveur). Le serveur se servira du système SVN pour le stockage des fichiers (il ne s'agit pas de développer un nouvel outil).
Le principe consiste à ce que le serveur interagisse avec les clients pour synchroniser leurs fichiers. Ces fichiers sont stockés sur un système SVN qui n'est accessible que par le serveur.
Un manuel utilisateur, un manuel administrateur, ainsi que tous les livrables conçus lors de la phase de développement devront être fournis avec le produit.

2.4 Contexte du projet

Le projet se situe dans un cadre universitaire. Tous les livrables seront conçus à cet effet, et le seront sous une licence libre.

3 EXIGENCES SUR LE PRODUIT

3.1 Capacités Fonctionnelles

3.1.1 Acteurs mis en jeu

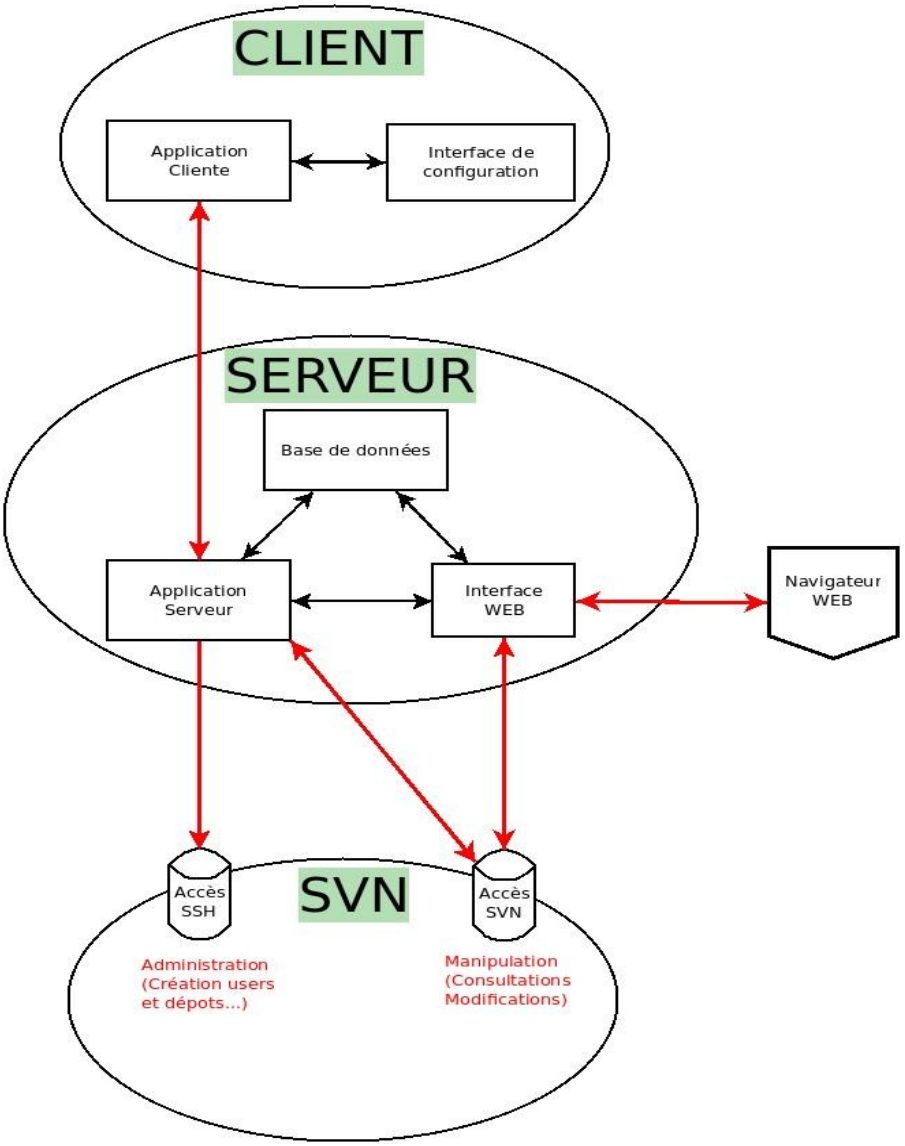
Le produit à concevoir se décompose en deux parties. Il est destiné à interagir avec un serveur SVN existant ou pas.

C'est en fait un système client-serveur qui se définit comme suit :

- **Une partie cliente** :
  Elle est composée d'une application cliente à proprement parlé et d'une interface de configuration. Cette dernière servira à renseigner l'adresse et le port de l'application serveur. Elle servira aussi à choisir les emplacements où seront chargées les fichiers synchronisés.

- **Une partie serveur** :
  Elle est composée d'une application serveur à proprement parlé, d'une interface web et d'une base de données. Elle stocke les fichiers synchronisés sur un serveur SVN.
  - L'interface web servira à la création de dépôts/administrateurs/clients, à la consultation des fichiers et de l'historique.
  - La base de données servira à stocker les identifiants/mots de passe des utilisateurs et leur adresse ip.

Le serveur permet aux clients de synchroniser des fichiers qui sont stockés sur le SVN. Le scénario général d'utilisation est le suivant : Dès qu'un client détecte une modification, il envoi le fichier au serveur. Ce dernier le sauvegarde sur le SVN, et envoi une copie à tous les autres clients pour qu'ils soient à jour.
Nous expliciterons cela tout au long du document.
D'abord, le schéma suivant met en jeu toutes les parties de l'application, et les interactions qu'il y a entre elles.



3.1.2 Descriptions des fonctionnalités

Fonctionnalités offertes par la partie cliente :

Nom de la fonctionnalité	***Module concerné***	***Description***
S'identifier au près du serveur.	Application	Le client doit être capable de s'identifier auprès du serveur afin que celui-ci connaisse son identité
Détecter les changements dans les répertoires partagés.	Application	Le client doit posséder la liste des répertoires partagés sur l'ordinateur, et doit être capable de détecter les changements dans ces répertoires.
Prévenir le serveur d'une modification dans le(s) répertoire(s) surveillé(s) et lui envoyer la liste du/des fichier(s) modifié(s).	Application	Lorsqu'il y a détection d'un(e) ajout/mise à jour/suppression d'un fichier, le client doit être capable de prévenir le serveur et de lui transférer le(s) fichiers concernés.

Recevoir une liste de fichiers modifiés correspondant à une mise à jour.	Application	Le client doit être prêt à recevoir une mise à jour du serveur et donc correctement actualiser les répertoires synchronisés.
Permettre la configuration et la personnalisation des répertoires à synchroniser	Interface de configuration	L'interface de configuration doit permettre l'accès graphique à plusieurs options tel que le paramétrage de adresse serveur et la personnalisation du répertoire synchronisé.

Fonctionnalités offertes par la partie serveur :

Nom de la fonctionnalité	***Module concerné***	***Description***
Recevoir une liste de fichiers modifiés correspondant à une mise à jour d'un client, et les importer sur le serveur SVN.	Application	Le serveur doit être prêt à recevoir de la part d'un client une liste de fichiers modifiés. Il doit ensuite être capable de transférer cette modification sur le serveur SVN.
Envoyer la liste des fichiers modifiés reçue aux autres clients.	Application	Le serveur doit être capable d'avertir et de mettre à jour tous les clients lorsqu'une modification est reçue.
Création (modification ou suppression) d'un nouvel administrateur et d'un nouveau dépôt.	Interface Web Base de données Application	L'interface web doit permettre au Super Administrateur de créer un nouveau dépôt et son nouvel administrateur. Les informations relatives aux nouveaux comptes crées devront être stockées dans la base de données. L'application devra être capable de relayer la requête jusqu'au serveur SVN, pour que ces modifications soient prises en compte.
Création (modification ou suppression) d'un nouveau client.	Interface Web Base de données Application	L'interface web doit permettre à l'administrateur de créer un nouveau client pour son dépôt. Les informations relatives aux nouveaux comptes crées devront être stockées dans la base de données. L'application devra être capable de relayer la requête jusqu'au serveur SVN, pour que ces modifications soient prises en compte.
Permettre la configuration.	Interface de configuration	L'interface de configuration doit permettre l'accès graphique à plusieurs options tel que le paramétrage du serveur SVN.

3.1.3 Interopérabilité

Le système interagira avec 3 acteurs externes qui sont : Un serveur SVN, un système de gestion de base de données, et un serveur web.
Il sera développé de façon à ce qu'il soit conforme aux versions actuelles et précédentes.
Pour les versions futures, aucune conformité n'est garantie, étant donné qu'on ne peux prédire le protocole à adopter pour la communication.

3.2 Exigences non fonctionnelles

3.2.1 Sécurité

Le logiciel devra fournir les moyens nécessaires pour rendre l'accès aux fichiers, sécurisé. Si besoin est, des protocoles sécurisés seront utilisés pour la communication entre le serveur et les clients. Il devra donc fournir des fonctionnalités d'authentification pour les clients vers le serveur.

3.2.2 Rendement

Le client et le serveur s'exécutant en tâche de fond, ils devront être assez fluide d'exécution, pour que l'utilisateur final ne perçoit pas de ralentissement sur sa machine.

3.2.3 Maintenabilité

Le logiciel est livré avec un dossier d'étude complet contenant ses documents descriptifs, sa structure et dynamique. De plus, les sources du code commentées seront également livrées.

3.2.4 Portabilité

Le logiciel fonctionnera à priori sur les trois systèmes d'exploitations les plus répandus à savoir : Microsoft Windows, Linux, et Mac OS.

3.2.5 Facilité d'utilisation

Le logiciel devra aussi assurer un niveau acceptable d'ergonomie et de facilité d'utilisation. Pour cela, des interfaces de configuration seront disponibles pour le client et le serveur.
Pour plus de détails, les utilisateurs pourront toujours se référer au manuel d'utilisation.

3.3 Exigences concernant le développement du produit

3.3.1 Objectifs de délais

Une première version sera livrée le 18 Mai 2011, et la version finale est à rendre avant le 15 Juin 2011.

3.3.2 Objectifs de coûts

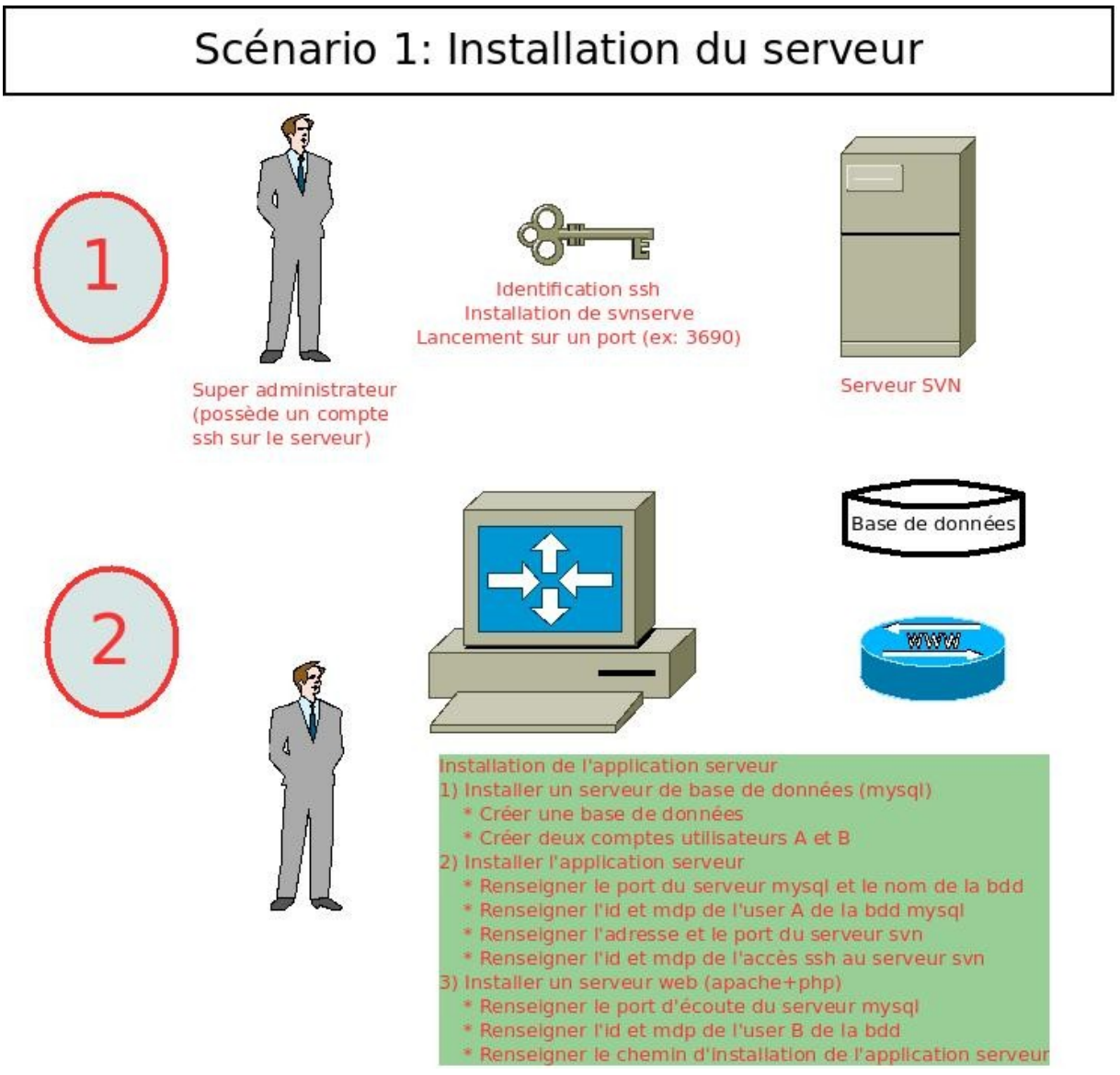
Le projet s'inscrit dans un cadre de logiciel libre. Aussi, tous les outils utilisés lors du développement devront être de licence libre.

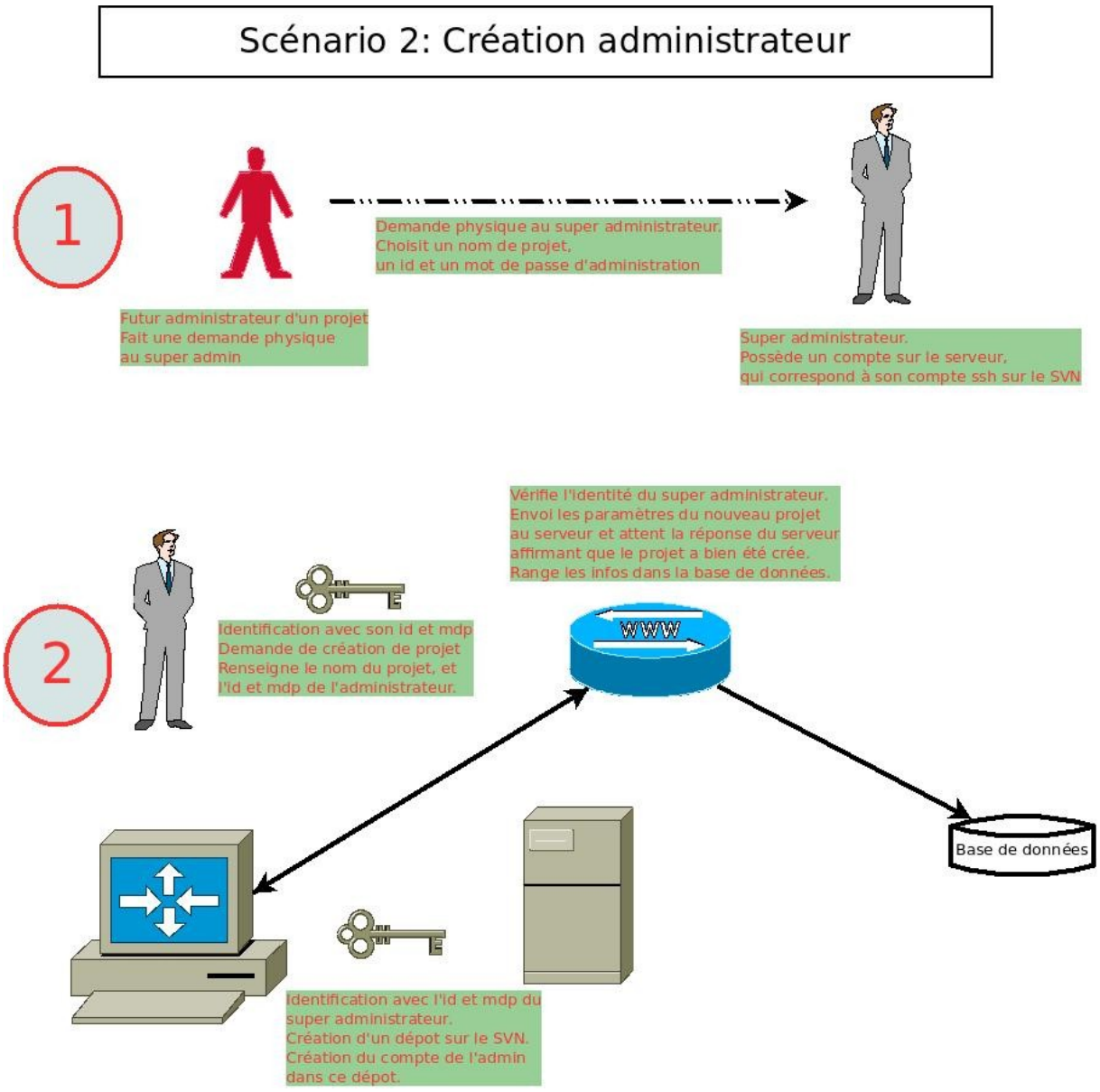
3.3.3 Exigences de réalisation

Le langage de programmation à utiliser sera le C++, et la bibliothèque Qt sera utilisée pour l'interface graphique et la gestion du réseau.
L'environnement de développement est Linux/Ubuntu. Tous les documents rédigés le seront au format odt. La plateforme de partage et de travail collaboratif utilisée est G-Forge.

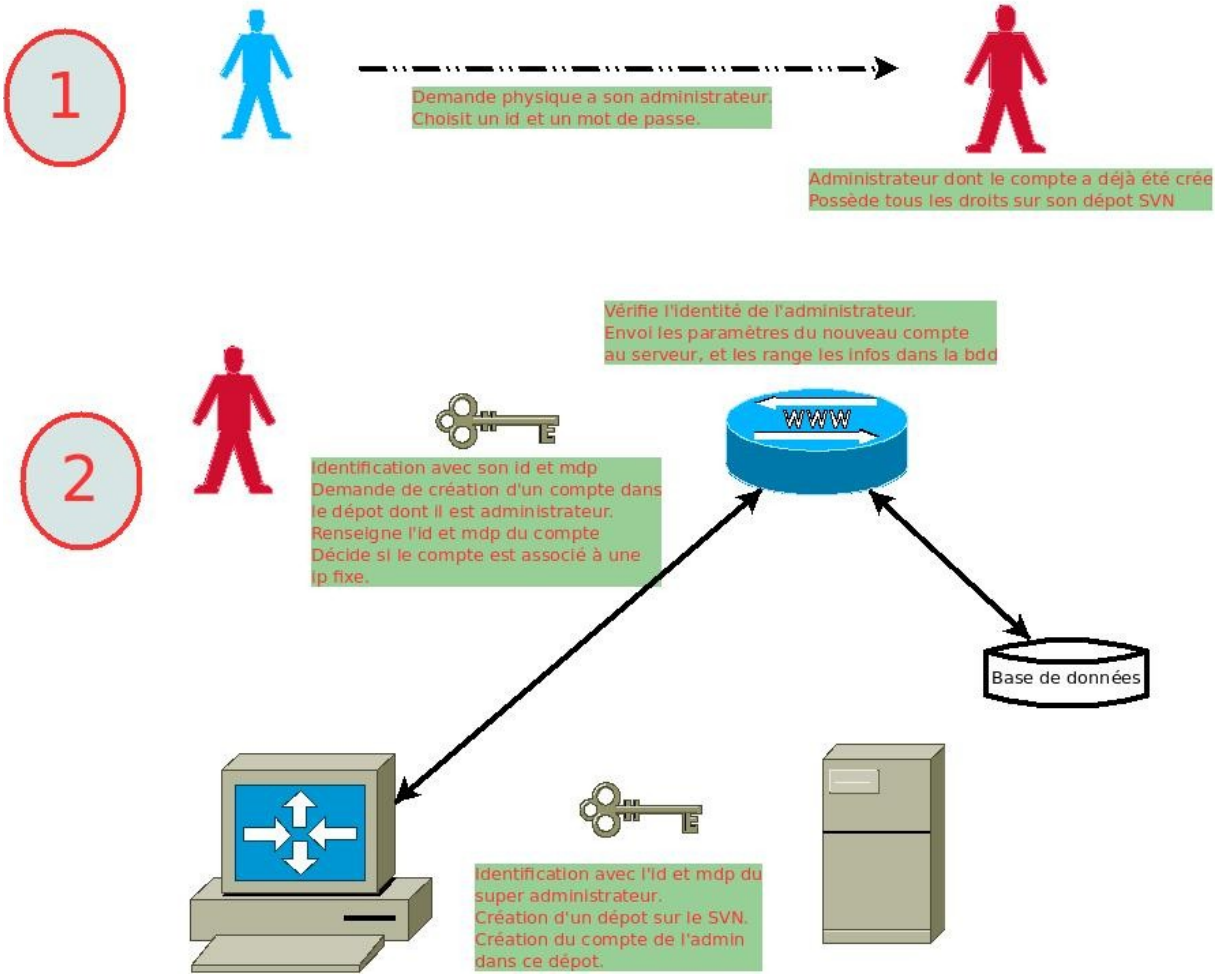
4 SCÉNARIOS DES CAS D'UTILISATION

Les schémas suivants ont été réalisés sur l'outil dia. Ils ont ensuite été importés dans ce document. Une version en format jpeg est disponible dans le fichier zip. Vous pourrez vous y reporter dans le cas où ces schémas ne sont pas assez visibles.

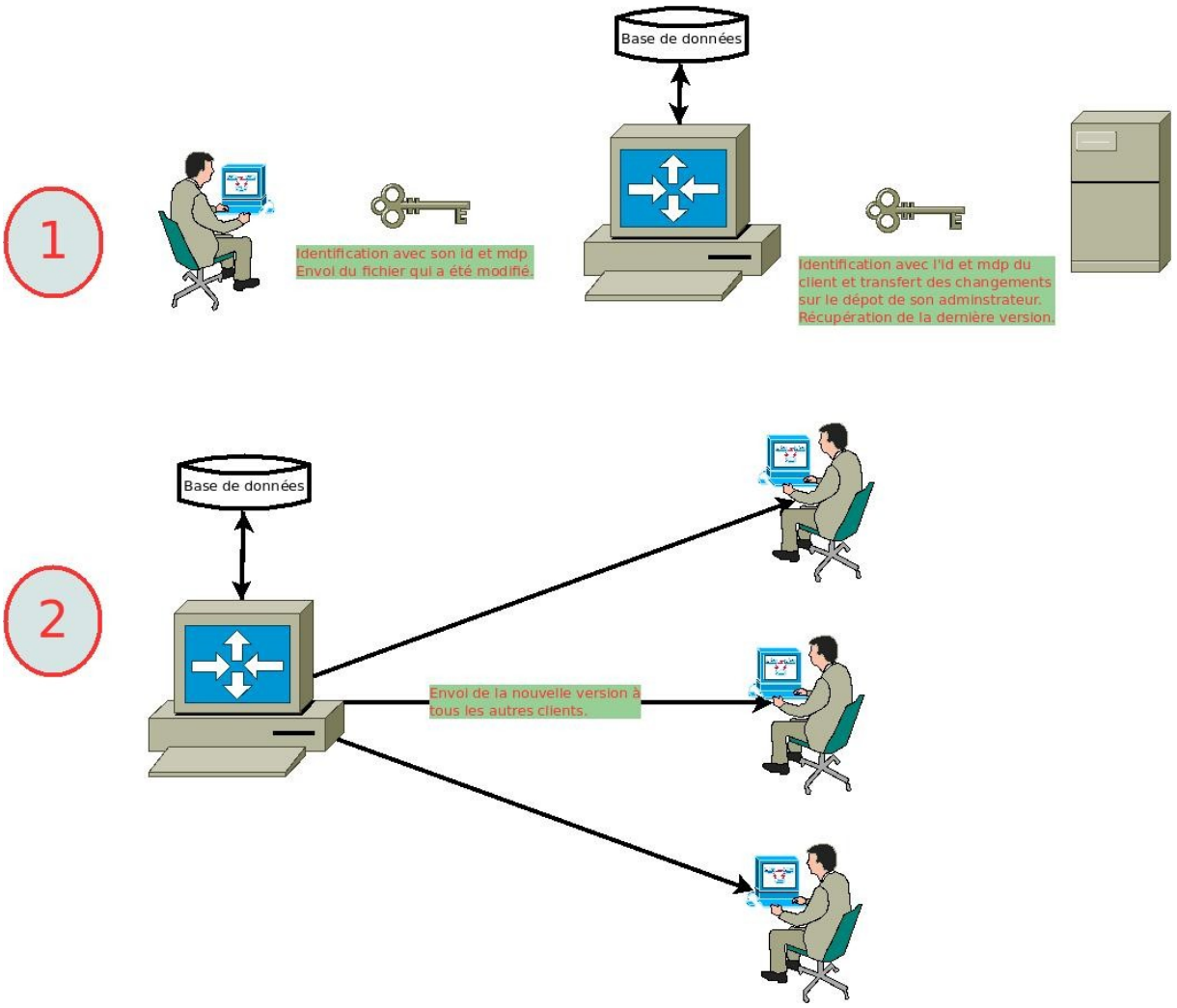



Scénario 2: Création administrateur
1
Demande physique au super administrateur.
Choisit un nom de projet,
un id et un mot de passe d'administration
Futur administrateur d'un projet
Fait une demande physique
au super admin
Super administrateur.
Possède un compte sur le serveur,
qui correspond à son compte ssh sur le SVN
Vérifie l'identité du super administrateur.
Envoi les paramètres du nouveau projet
au serveur et attent la réponse du serveur
affirmant que le projet a bien été crée.
Range les infos dans la base de données.
2
Identification avec son id et mdp
Demande de création de projet
Renseigne le nom du projet, et
l'id et mdp de l'administrateur.
WWW
Base de données
Identification avec l'id et mdp du
super administrateur.
Création d'un dépot sur le SVN.
Création du compte de l'admin
dans ce dépot.

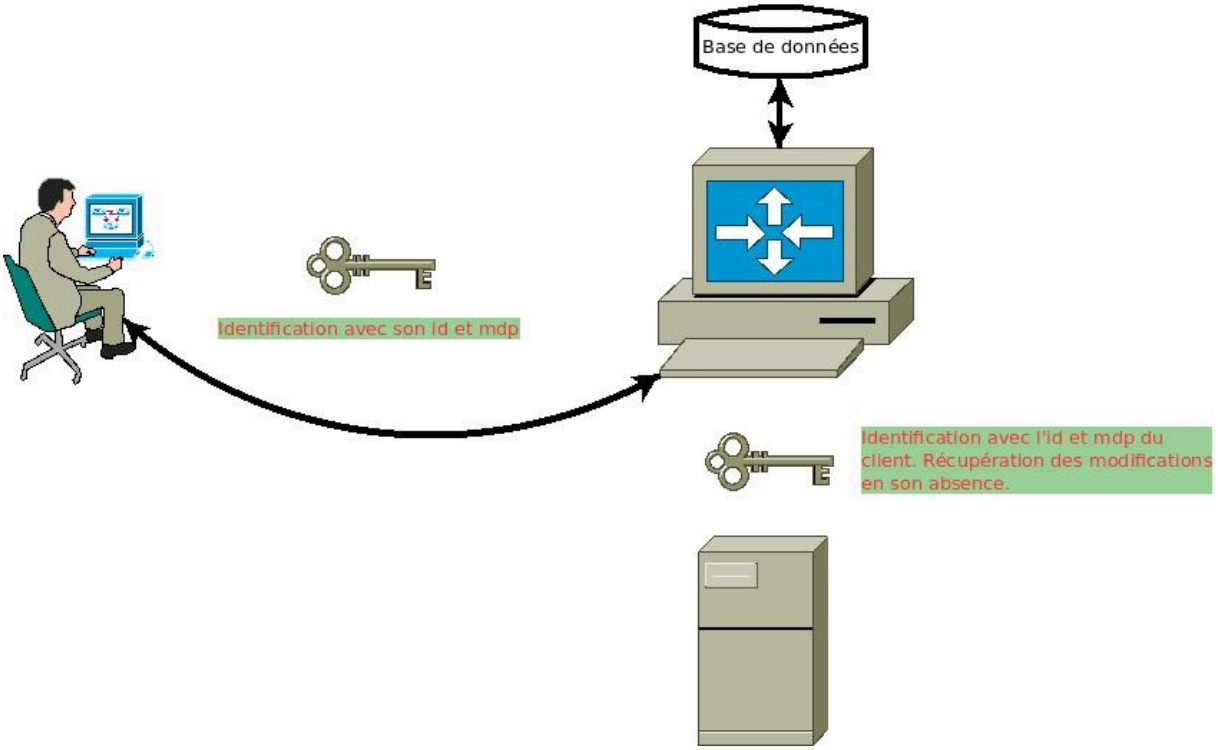
Scénario 3: Création client



Scénario 4: Détection d'une modification



Scénario 5: Démarrage d'un client



Scénario 6: Accès par l'interface web