

Kernel

Document de Vision

Version *0.1*

Historique des révisions

Date	Version	Description	Auteur
3 Mars 2020	0.1	Initial Initial draft	David Rodriguez
11 Mars 2020	1.0	First release	La plupart des membres

1 Introduction

1.1 Objectifs du document

Ce document propose une base d'entente sur les divers aspects techniques et conceptuels pour le développement du service "Kernel", comme :

- Identifier le besoin auquel répond ce service, et les clients potentiels.
- Décrire les caractéristiques de la solution proposée.
- Identifier les personnes liés au développement et utilisation de la solution ainsi que leurs rôles.
- Proposer une base de connaissances sur les défis liés au développement de la solution.

1.2 Portée

Ce projet fait participer de nombreuses technologies :

- Connu : Java et IDE, HTML, Git
- Inconnus : Apache Maven, Docker, Kubernetes, NodeJS, Angular, TCP, SSL
- Différents réseaux sociaux et leur API (voir ci-dessous)

1.3 Définitions, Acronymes et Abréviations

Membres de l'équipe de développement : David, Julien, Kathleen, Loan, Mark, Svetlana

Sont utilisés de manière interchangeable : "Kernel", "service", "solution", etc

Technologies : Java, Apache Maven, Docker, Kubernetes, NodeJS, Angular, TCP, SSL, Git

Réseaux sociaux : Sont principalement visés **Facebook**, **Twitter**, **Instagram**

Concurrents : **SocialHub**, **Buffer**, **Hootsuite**, **IM+**

API : "Interface de programmation", moyen technique de permettre la communication entre deux logiciels.

RGPD : "Règlement général sur la protection des données", texte de loi récent implémenté au niveau européen, dicte des limitations strictes sur la manipulation de données personnelles.

2 Positionnement

2.1 Opportunité commerciale

Le projet est motivé par l'utilisation actuelle massive des réseaux sociaux et leur diversité. Beaucoup de personnes pourraient être intéressées par un service d'aggrégation, *y compris payant*.

2.2 Position du problème

Le problème	<i>Les publications et les fils d'actualité trop nombreux ou dupliqués, ainsi que la décentralisation de fils de discussion (sur chacune des plateformes) avec une même personne.</i>
Affecte	<i>les utilisateurs de multiples réseaux sociaux</i>
L'impact du problème est	<i>confusion et redondance des fils d'actualité, des publications, ainsi que des messages en provenance des différents réseaux sociaux, ce qui diminue l'efficacité de communication par la nécessité de changer de plateformes fréquemment, une plus grande probabilité de ne pas voir certains contenus par le filtrage manuel des doublons</i>
Une solution satisfaisante serait	<i>regrouper/unifier dans une même application les fils d'actualités, les publications, ainsi que les messages en provenance des différents réseaux sociaux, et éventuellement les contacts en indiquant quels réseaux sociaux sont utilisés par chacun.</i>

2.3 Position du produit

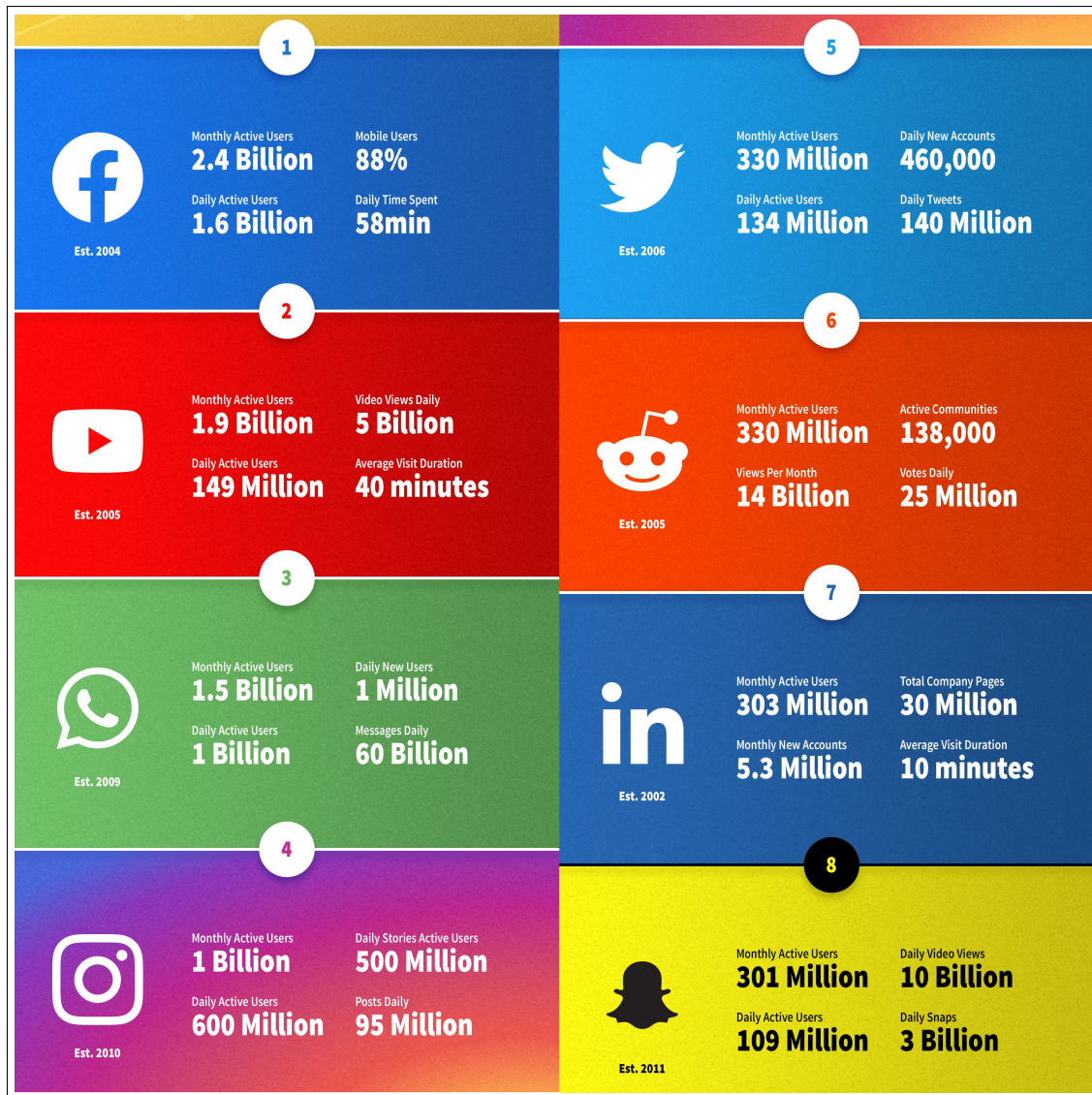
Pour	<i>un utilisateur avéré de réseaux sociaux</i>
Qui	<i>qui utilise différentes plateformes et voudrait avoir une expérience plus plaisante</i>
Kernel	<i>est un agrégateur</i>
Qui	<i>regrouper les fonctionnalités principales de ces différentes plateformes</i>
A la différence de	<i>SocialHub (et autres)</i>
Notre produit	<i>visé un large public (les utilisateurs lambda) plutôt que les entreprises ou individus importants</i>

3 Description des intervenants et des utilisateurs

3.1 Taille du marché

Le marché visé est **potentiellement énorme** : des centaines de millions d'utilisateurs se connectent quotidiennement sur leurs réseaux sociaux favoris.

SocialHub et Hootsuite ont des centaines de clients (entreprises). Le marché visé de notre projet est plus large, soit **des millions d'utilisateurs**. La taille du marché précis (personnes utilisant de multiples réseaux sociaux et souhaitant une solution) reste difficile à estimer précisément.



Source : <https://dustinstout.com/social-media-statistics/>

3.2 Les intervenants

On considère l'équipe de développement elle-même (chef de projet, programmeurs, administrateur système) puis l'équipe d'évaluation externe (professeur, assistant, moniteur).

Nom	Description	Rôle
Chef de projet	Il coordonne l'équipe de développement, tout en développant lui-même	Coordonner l'équipe, régler les bugs, gérer les disponibilités des membres de l'équipe
Programmeur	Il développe la plateforme	Maintenir le code, implémenter les fonctionnalités de la plateforme
Administrateur système	Il assure que le système (VM) soit fiable et le code de la plateforme soit fonctionnel. Il intègre le code en vérifiant la qualité du code (unit testing + system testing)	Assurer le bon fonctionnement du système, gérer les problèmes techniques, intégrer le code, exécuter les tests
Équipe d'évaluation	Professeur, assistant, moniteur (à définir)	Évalue la plateforme et ses fonctionnalités : Évaluer le code, utiliser l'interface de l'application, rédiger des rapports d'évaluation
Législateur	Propre à chaque pays, il possède le pouvoir de décider et faire appliquer les lois	Définit le cadre légal dans lequel Kernel doit opérer
Régie publicitaire	Offre des services publicitaires	Fournit la rémunération nécessaire à faire vivre Kernel

3.3 Les utilisateurs

On différencie les utilisateurs standards, premium et ceux souhaitant sponsoriser leurs publications.

Nom	Description	Rôle	Représentant
Utilisateurs standards	Utiliser la plateforme de façon basique	Utiliser la plateforme avec des publicités et publications sponsorisées	Equipe d'évaluation, législateur
Utilisateurs premium	Bénéficie en plus de fonctionnalités avancées	Utiliser la plateforme sans publicités avec un fil d'actualité personnalisé	Equipe d'évaluation, législateur
Utilisateur (premium) sponsor	Souhaite que ses publications soient visibles par d'autres utilisateurs	Paie pour des publications sponsorisées	Equipe d'évaluation, législateur

3.4 Environnement utilisateur

Typiquement, les utilisateurs finaux utilisent des applications mobiles, ou parfois un navigateur internet, afin d'accéder à leurs réseaux sociaux.

Kernel doit impérativement proposer une expérience similaire aux utilisateurs, donc permettre d'utiliser les fonctionnalités de base des réseaux sociaux dont ils ont l'habitude.

3.5 Profils des intervenants

Décrire chaque intervenant (non utilisateur) en remplissant les rubriques de la table ci-dessous

3.5.1 Chef de projet

Représentant	Svetlana
Description	Membre de l'équipe de développement
Type	Novice-connaisseur en programmation, sans expertise en direction de projet
Responsabilités	Le chef de projet est particulièrement responsable de la performance de l'équipe et donc la réussite du projet
Critère de succès	Le projet arrive à un stade auquel il est utilisable et répond aux exigences listées dans ce document. La récompense pour le succès est la fierté d'accomplissement et l'obtention d'une bonne note.
Implications	Coordonne l'équipe de développement tout en développant, gère les disponibilités des membres de l'équipe.
Livrables	L'équipe de développement est bien organisée et peut travailler efficacement.
Comments/Issues	Si ce membre n'est pas efficace à son rôle, le reste de l'équipe et le projet en souffrira.

3.5.2 Programmeur

Représentant	Tous les membres de l'équipe
Description	Membre de l'équipe de développement
Type	Novice-connaisseur en programmation, sans expertise dans ce type de projet
Responsabilités	Doit travailler en équipe et fournir du code de qualité
Critère de succès	Le projet arrive à un stade auquel il est utilisable et répond aux exigences listées dans ce document. La récompense pour le succès est la fierté d'accomplissement et l'obtention d'une bonne note.
Implications	Fait partie de l'équipe de développement
Livrables	Le code qui compose le projet (y compris les tests) et la documentation.
Comments/Issues	Si un membre n'est pas efficace à son rôle, le reste de l'équipe et le projet en souffrira.

3.5.3 Administrateur Système

Représentant	Mark
Description	Membre de l'équipe de développement
Type	Novice-connaisseur en programmation, sans expertise dans ce type de projet
Responsabilités	Assurer le bon fonctionnement du système (serveur et environnement de développement), gérer les problèmes techniques, intégrer le code, exécuter les tests
Critère de succès	Le projet arrive à un stade auquel il est utilisable et répond aux exigences listées dans ce document. La récompense pour le succès est la fierté d'accomplissement et l'obtention d'une bonne note.
Implications	Fait partie de l'équipe de développement et assume le rôle supplémentaire d'administrateur système.
Livrables	Les membres de l'équipe de développement ont les outils nécessaires
Comments/Issues	Si un membre n'est pas efficace à son rôle, le reste de l'équipe et le projet en souffrira.

3.6 Profils des utilisateurs

3.6.1 Utilisateur standard

Représentant	Équipe d'évaluation, législateur
Description	Utilisateur de réseaux sociaux, utilise gratuitement Kernel
Responsabilités	Respecter les règles d'utilisation, noter le service
Critère de succès	Le service répond à leur besoin, Kernel est plus agréable à utiliser que l'alternative. Possibilité de faire connaître sa satisfaction en note et commentaire.
Implications	Sans utilisateurs, Kernel n'est pas utile
Livrables	Retour (commentaire, note, etc)
Comments/Issues	Le large public peut avoir des attentes/goûts non envisagés, ou injustement mal noter le service

3.6.2 Utilisateur premium

Représentant	Équipe d'évaluation, législateur
Description	Utilisateur de réseaux sociaux, utilise Kernel de manière payante, souhaite une expérience avancée, sans publicité et avec accès au service de publications sponsorisées.
Responsabilités	Respecter les règles d'utilisation, noter le service
Critère de succès	Le service répond à leur besoin, Kernel est plus agréable à utiliser que l'alternative. Possibilité de faire connaître sa satisfaction en note et commentaire.
Implications	Sans ces utilisateurs, Kernel ne peut pas être rentable
Livrables	Retour (commentaire, note, etc)
Comments/Issues	Ce public peut avoir des attentes/goûts non envisagés, ou injustement mal noter le service

3.7 Alternatives et concurrence

SocialHub, Buffer, Hootsuite, IM+

Ces services sont similaires et partagent le fait qu'ils sont axés vers les entreprises et les individus importants.

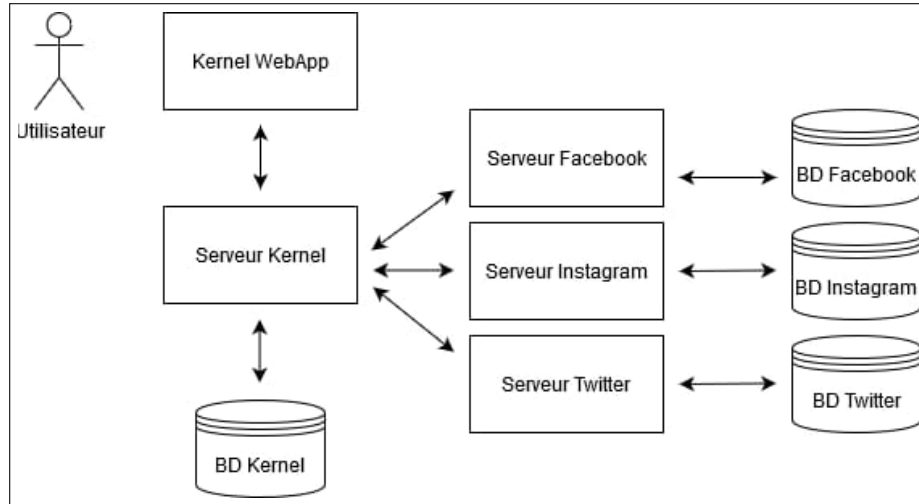
- Forces : Produit professionnel, clients bien définis, ressources, etc
- Faiblesse : Inaccessibles ou inintéressants pour le large public

4 Vue d'ensemble du produit

4.1 Perspective du produit

L'architecture de Kernel nécessite de s'appuyer sur les services externes (API) des réseaux sociaux, agissant comme *intermédiaire* entre ceux-ci et l'utilisateur.

Le service de Kernel nécessite les composants basiques que sont : un client (ici une webapp, fonctionne sur le terminal de l'utilisateur), qui communique avec se(s) serveur(s) afin d'obtenir le contenu, et ce(s) serveur(s) s'appuie(nt) sur une base de données (pour les information permanentes comme les informations sur l'utilisateur).



4.2 Résumé des caractéristiques

Avantage pour l'utilisateur	Caractéristiques correspondantes
Avoir un fil d'actualité unique	L'application propose de regrouper les fils d'actualité en provenance des différents réseaux sociaux utilisés par l'utilisateur.
Présenter un fil d'actualité personnalisé	L'application propose un fil d'actualité personnalisé à l'utilisateur.
Gestion des publications	L'application propose de réaliser une même publication sur un ou plusieurs réseaux sociaux (au choix) parmi ceux utilisés.
Gestion des posts	L'application propose un filtrage des posts en provenance des différents réseaux sociaux selon les centre d'intérêts ou préférences des différents utilisateurs.
Présenter un seul fil de discussion par contact	L'application propose de regrouper les messages des différents contacts en provenance des différents réseaux sociaux en une messagerie unique.
Gérer un seul fil de discussion	L'application permet d'envoyer un message à un contact indépendamment du réseau social utilisé
Regrouper les contacts	L'application propose de regrouper les contacts en indiquant quels réseaux sociaux sont utilisés par chacun d'entre d'eux.
Etendre la notoriété	L'application permet de sponsoriser les contenus pour atteindre plus de personnes.
Etendre la notoriété	L'application facilite la présence sur de multiples réseaux sociaux.

4.3 Hypothèses

Les caractéristiques du service Kernel dépend des éléments suivante :

- Accès aux APIs : Sans accès aux réseaux sociaux avec lesquels il doit interagir, Kernel sera incapable de fonctionner. Il faudra se contenter des réseaux sociaux dont l'API répond à nos besoins.
- Temps : La contrainte de temps et la taille de l'équipe de développement borne sérieusement la quantité de fonctionnalités qui pourra être implémentée.
- Connaissance technique : Les domaines technique inconnus identifiés sont vastes et nombreux (et pas forcément aisés à apprendre), et notre expérience est très limitée, ce qui devrait grandement nous limiter en termes de sophistication.
- Technologies empruntées : Nous utilisons des technologies comme Docker, Kubernetes, Angular, etc, sans vraiment les comprendre entièrement, ce qui signifie que Kernel dépendra du bon fonctionnement de ces composantes externes. Nous dépendrons aussi du serveur et la connexion Internet de l'Université

4.4 Coût et politique de prix

Bien qu'il fut déterminé que les comptes premium et les publications sponsorisées seraient payantes, nous n'avons pas d'idée fixe sur les coûts qui devraient y être associés pour le moment.

5 Caractéristiques essentielles du produit

En blue : **Must**, en noir : **Should**, en gris : **Could**

Avantage pour l'utilisateur	Caractéristiques correspondantes
Avoir un fil d'actualité unique	L'application propose de regrouper les fils d'actualité en provenance des différents réseaux sociaux utilisés par l'utilisateur.
Présenter un fil d'actualité personnalisé	L'application propose un fil d'actualité personnalisé à l'utilisateur.
Gestion des publications	L'application propose de réaliser une même publication sur un ou plusieurs réseaux sociaux (au choix) parmi ceux utilisés.
Gestion des posts	L'application propose un filtrage des posts en provenance des différents réseaux sociaux selon les centre d'intérêts ou préférences des différents utilisateurs.
Présenter un seul fil de discussion par contact	L'application propose de regrouper les messages des différents contacts en provenance des différents réseaux sociaux en une messagerie unique.
Gérer un seul fil de discussion	L'application permet d'envoyer un message à un contact indépendamment du réseau social utilisé
Regrouper les contacts	L'application propose de regrouper les contacts en indiquant quels réseaux sociaux sont utilisés par chacun d'entre d'eux.
Etendre la notoriété	L'application permet de sponsoriser les contenus pour atteindre plus de personnes.
Etendre la notoriété	L'application facilite la présence sur de multiples réseaux sociaux.

6 Contraintes sur le produit

En plus des contraintes énoncées jusqu'ici (principalement en section 4.3), on peut mentionner :

- Cadre légal : La législation de chaque pays doit être prise en compte car cela impose des limites sur ce que Kernel peut faire. En particulier, le **RGPD** devra être étudié.
- Capacité de test : La plupart des membres de l'équipe estiment ne pas faire partie de la population cible de Kernel. Il faudra donc se familiariser avec les réseaux sociaux et créer des comptes.

7 Tolérances de qualité non fonctionnelles

Voir le document joint `Kernel_NFR`.

8 Autres exigences sur le produit

8.1 Besoins système

Un serveur avec suffisamment de ressources (inclut mémoire de travail et de stockage, puissance de calcul et accès à internet), une plateforme de développement satisfaisante, des services externes fonctionnels, la non-annihilation de la population et des infrastructures (informez nous si vous avez lu ça).

9 Exigence de documentation

Nous suivrons les directives quand aux exigences de documentation imposées. Si non applicable, nous adopterons une approche "just enough doc".