

Cahier des charges



Cadre du projet	6
Résumé du projet	6
Contexte de création du projet	6
Présentation du groupe	6
Situation initiale du besoin & Benchmark	6
Première situation	6
Solutions possible sans Arkage	6
La solution Arkage	7
La différence avec la concurrence	7
Seconde situation	8
Solution sans Arkage	8
Solution avec Arkage	8
En résumé	8
Définition & objectifs du projet	8
Arkage Kernel	8
Arkage UI	9
Arkage Website	9
Considération marketing	9
Cibles	9
Les utilisateurs	9
Les développeurs	10
Les entreprises	10
Personas	10
Référencement	11
Charte éditoriale	11
Notre histoire	11
Nos valeurs	11
Présentation de l'organisation interne	12
Stratégie de communication	12
Les KPI à surveiller	12
Sur l'application	12
Sur le site web	12
Sur les réseaux sociaux	12
Rappel des (principaux) supports utilisés	13
Notre vision éditoriale	13
Mots-clés	13
Notre ton de communication	13
Bonnes pratiques	13
Types de contenus à produire	13
Vocabulaire	14

Licence logiciel	14
Introduction	14
Utilisation personnelle ou utilisation pour les associations à but non-lucratif	14
Utilisation professionnelle ou à but lucratif	15
Publication du code	15
Licence de version ou licence continue	15
Identité visuelle	15
Brief créatif	15
Charte graphique	16
Spécifications fonctionnelles	16
Arkage Kernel	16
Arkage Package Scenario	16
Les modules systèmes	17
Terminal	17
Shell Grammar	18
SettingsManager	18
PasswordManager	18
Mots de passes primaire et secondaire	18
SMSBridge	19
PermissionManager	19
Rapport d'utilisation	20
UX & UI	20
Données récoltées pour un rapport	20
Arkage UI	20
Arborescence	20
Interface "Welcome Page"	20
Interface "Lock Page"	21
Interface "Home Page"	21
Partie menu inférieur	21
Partie menu supérieur droite	21
Comportement gestuel	21
Interface "Options"	21
Liste des options	21
Interface "Autorisations"	22
Listes des autorisations	22
Interface "Documentation"	22
Interface "Terminal UI"	23
Interface "Exportation des logs"	24
Arkage Website	24
Arborescence	24
Page {nom de la page}	24

Reporting API & Databases	24
Spécifications techniques	24
Arkage Kernel	24
Arkage Package Scenario	24
Terminal	24
Shell Grammar	26
Exemples	26
Les options orphelines	26
Le découpage des commandes	27
SettingsManager	27
Nommage de la clé de setting	27
Liste des Settings	27
PasswordManager	28
Mots de passes primaire et secondaire	28
SMSBridge	29
PermissionManager	30
Rapport d'utilisation	30
Premier rapport d'utilisation	30
Données récoltées pour un rapport	31
Stockage sur les serveurs Arkage	31
Arkage UI	31
Arkage Website	31
Technologie & Framework	31
Documentation "Docusaurus"	31
Domaines, sous-domaines et hébergement	31
www.arkage.app	32
api.arkage.app	32
admin.arkage.app	32
doc.arkage.app	32
Statistique et utilisation du site	32
API rapport d'utilisation	32
Sécurité	32
Environnement de développement & maintenance	32
Conclusion	32
Le futur d'Arkage	33
Lexique & annexes	33

Cadre du projet

Résumé du projet

Arkage est une technologie Android dont le but est de permettre le contrôle à distance du téléphone et la création de programmes internes. Cette application est un moteur pour d'autres applications en surcouche. Ces types d'applications sont aussi appelées des "[middleware](#)".

Pour cela, Arkage embarque un terminal interne ainsi que d'autres outils.

Contexte de création du projet

Le projet Arkage est créé dans le cadre d'un projet de fin d'année pour l'école My Digital School Angers. Ce projet réunit les différentes filières de l'école dans un seul projet, à savoir :

- Marketing
- Communication
- Design
- Développement

Ce projet doit être présenté en fin d'année scolaire, durant un oral.

Présentation du groupe

Arkage va être initialement développé par 6 étudiants de 3ème année de Bachelor. Swann Lambard, au poste de chef de projet & marketing. Camille Esneault est à la charge des médias & communication. Emma Hervé directrice artistique. Amir Lachemet et Alexandre Pouivet développeur mobile. Et Téo Conan, leader technique, développeur mobile et à l'initiative du projet.

Situation initiale du besoin & Benchmark

Première situation

“Une jeune femme fait une longue promenade dans un parc. Arrivée aux trois quarts de la promenade, elle se rend compte qu'elle a perdu son smartphone. Elle sait qu'il est tombé sur le chemin ou dans l'herbe mais elle ne sait pas à quel moment elle l'a perdu. Elle n'a donc plus moyen de communiquer et peut seulement demander de l'aide aux personnes aux alentours pour appeler, faire une recherche internet, etc...”

Solutions possible sans Arkage

Dans ce cas de figure, plusieurs solutions sont possibles pour retrouver ce smartphone :

- Faire le trajet inverse et observer le sol.

- Inconvénients : Aucun indice sur la position du smartphone, le smartphone ne sera peut-être pas retrouvé.
- Demander à un passant d'appeler le smartphone pour espérer l'entendre sonner.
 - Inconvénients : Possible bruit ambiant, smartphone en mode silencieux etc...
- Certains portables ont au sein de l'OS un "smartphone finder"
 - Inconvénients : Mais dans certains cas, il fonctionne mal ou nécessite d'avoir un ordinateur ou d'être connecté.

La solution Arkage

Tout dépend du type d'utilisateur ou de portable, mais régulièrement les smartphones n'ont pas la localisation ou les données mobiles activées (forfait limité, manque de batterie etc...) Cependant, la plupart des smartphones sont tout le temps disponibles à la réception de SMS ou d'appel. Pourquoi ne pas utiliser cette ressource pour faire passer un message au smartphone perdu pour tenter de le retrouver ?

Si l'utilisateur a Arkage installé sur son smartphone, il suffit d'emprunter le portable d'un passant et d'envoyer un SMS au smartphone perdu.

La commande par SMS à envoyer peut ressembler par exemple à :

"Arkage je suis perdu 6519"

Arkage lit les messages entrant et reconnaît lorsqu'on lui envoie une commande, et ne voudra pas se lancer sans le mot de passe.

Une fois le SMS reçu, Arkage lancera son mode "Lost" et communiquera avec la personne qui a envoyé le SMS et exécutera un protocole pour permettre de le retrouver.

Pour donner un exemple :

- Réponse au numéro entrant disant qu'Arkage s'est bien lancé
- Surveillance de la batterie et réagir en fonction du niveau restant
- Activation des données mobiles et localisation
- Prise de photos et envoi au numéro entrant
- Enregistrement du son ambiant si les photos ne donnent pas d'indice concernant la localisation
- Etc...

La différence avec la concurrence

Les applications présentes sur le marché offrent les mêmes fonctionnalités de base : traquer et sécuriser son téléphone. Pour ce faire, elles mettent en place divers processus de tracking, à savoir la géolocalisation ou l'envoi d'un sms permettant de retrouver les coordonnées géographiques de son téléphone. Certaines applications se spécialisent dans le vol de téléphone, telles que Prey Anti Theft, qui avance une sécurisation avancée : indication de la localisation ou prise de photos de l'environnement. La plupart d'entre-elles mettent également à disposition la fonction verrouillage du téléphone. C'est cette fonction

qui semble poser problème chez certaines applications, puisque quelques utilisateurs rencontrent des difficultés au moment du déverrouillage du téléphone.

La majorité de ces applications sont étrangères et quelques unes sont indisponibles en langue française, ce qui pose problème pour la compréhension de l'application.

Seules les applications "Prey Anti Theft" et "Where's my Droid" semblent reposer sur une stratégie webmarketing solide. Elles affichent un site web vitrine complet, contrairement aux autres applications. Une partie blog vient s'ajouter à leur site, qui permet de mettre en avant leurs actualités, les mises à jour ou les nouveautés techniques de l'application. La plupart des applications sont également présentes sur les réseaux sociaux. Facebook et Twitter semblent privilégiés, seule "Prey Anti Theft" est présente sur Instagram et YouTube. Cette dernière utilise également la technique du livre blanc intitulé : "5 essential mobile security tips".

Pour se légitimer, ces applications se basent sur des accréditations officielles (awards) et affichent au premier plan les avis positifs de leurs clients. La plupart présentent également une vidéo démonstrative qui vient argumenter l'utilisation de l'application. Bien souvent, cette vidéo est intégrée à un site web vitrine, qui vient donner davantage d'explications sur l'application que la plateforme de diffusion. Le site web présente les différentes fonctionnalités de l'application, les formules disponibles (gratuites ou payantes) et offre la plupart du temps un support technique pour les utilisateurs.

On peut donc constater une vraie présence concurrentielle sur le marché, avec un leader (Prey) et d'autres suiveurs, qui ne restent pas indétrônables. Ces applications présentent des fonctionnalités poussées, mais parfois défailtantes, ce qui ressort dans les inconvénients à l'usage. Seule Prey également est vraiment présente en matière de stratégie digitale, ce qui laisse de la place pour penser à une application efficace et une stratégie digitale pertinente.

Seconde situation

"Un homme se fait voler son téléphone dans un restaurant, il s'en rend compte très rapidement mais a déjà perdu de vue les malfaiteurs.

Il doit bloquer au plus vite son portable avant que les voleurs utilisent le portable.

Cependant il n'a pas les données mobiles activées, ni la localisation."

Solution sans Arkage

Si par hasard vous avez une application dédiée pour ce type de situation et qu'elle fonctionne vous pourrez sceller votre smartphone pour que les voleurs n'y aient pas accès... Et / ou vous pourrez aller porter plainte dans un commissariat.

Solution avec Arkage

Même système que la première situation : depuis n'importe quel portable, il suffit d'envoyer un SMS pour passer le smartphone en mode "volé"

Suite à ça, Arkage sera capable de prendre en vidéo et photos les voleurs, enregistrer le son ambiant, lister les wifi ambiants...

En résumé

Arkage est donc une application permettant de demander à son téléphone d'effectuer des actions à distance (SMS, bluetooth etc...) au travers de différentes commandes faciles d'utilisation.

Arkage referme aussi, pour les plus doués, un terminal de commande. Comme on peut en trouver un sous linux.

Voici donc ce dont est capable Arkage en surface. Mais à présent rentrons un peu plus en détails sur ce projet.

Définition & objectifs du projet

Comme nous l'avons décrit ci-dessus, le pitch de base d'Arkage est d'offrir à l'utilisateur des outils pour retrouver son téléphone.

Nous pourrions créer une application qui fait tout simplement ce que nous listons ci-dessus, mais là n'est pas nos ambitions.

Comme décrit dans le résumé du projet, nous voulons créer un "middleware"

Nous voulons pas créer une application qui fera 1 / 2 tâches sans pouvoir la faire évoluer.

Nous souhaitons créer tout un système, offrant la possibilité de créer par dessus ce système d'autres applications (et entre autres une application pour retrouver son téléphone).

Arkage Kernel

Le "Arkage Kernel" aussi appelé "le noyau" est le coeur de la technologie. On peut le comparer au moteur d'une voiture, sans lui rien ne fonctionne.

C'est dans ce moteur que seront disponibles tous les outils et fonctionnalités du projet. Cependant petit détail : l'Arkage Kernel ne dispose d'aucune interface.

Arkage UI

L'application UI Arkage, aussi appelé "**Client Arkage**", est une application Android classique, avec ses interfaces, ses options, son tableau de bord etc... Le tout est relié au Kernel pour définir ses options, ses autorisations ainsi que pour voir l'activité globale de l'application.

Le but de cette application est de contrôler le Kernel au travers d'une interface.

C'est cette application qui fait la "jointure" entre le "front-end et le back-end" d'Arkage. Sans cette application, le Kernel est toujours contrôlable et modifiable, mais uniquement via des lignes de commande. Elle a donc pour but d'aider l'utilisateur et de lui apprendre à utiliser Arkage.

Arkage Website

Le site web d'Arkage à plusieurs objectifs :

- La mise à disposition du code source Arkage ainsi que des outils de développement pour créer des "Arkage Package Scenario". Nous reviendrons sur ce qu'est un APS plus tard dans le document.
- Une documentation en ligne expliquant ce qu'est Arkage et comment le noyau fonctionne.

Des landing pages adoptant un discours spécifique en fonction de la cible :

- Une landing page dédiée aux utilisateurs "normaux" de l'application, autrement dit des personnes qui souhaitent utiliser les scénarios d'Arkage et qui n'ont aucune connaissance en développement.
- Une landing page dédiée aux professionnels de l'assurance qui souhaitent intégrer Arkage comme service supplémentaires dans leurs solutions d'assurances.
- Une landing page dédiée aux développeurs, qui les inciterait à utiliser l'application dans son total potentiel, comme par exemple le terminal.
- Une landing page dédiée aux professionnels, pour les entreprises qui souhaitent utiliser Arkage pour l'intégrer dans leur projet et ainsi accélérer leur temps de développement

Considération marketing

Tout ce qui concerne la considération marketing a été réalisé par Swann Lambard, Camille Esneault et Emma Herve.

Leurs rôles était d'apporter un point de vue ainsi qu'une expertise détachée du monde du développement. C'est donc grâce à ce groupe que nous devons toute la conception de marketing du projet, la communication et la conception UI / UX.

Cibles

Les utilisateurs (particuliers)

Ce type d'utilisateur n'a pas particulièrement de compétences en informatique, ils utilisent uniquement l'application Arkage UI. Dans la vie de tous les jours, ces utilisateurs se serviront rarement d'Arkage, l'application dormira et le jour où la situation se présentera ils se serviront de l'application pour sauver leur téléphone.

Les développeurs

Ce type d'utilisateur sera soit des développeurs, ou bien des personnes avec de grandes connaissances en informatique. Ils seront capables de se servir de toute les fonctionnalité d'Arkage sans forcément passer par l'application UI.

Ce type d'utilisateur sera susceptible de créer eux-même des commandes communautaires.

Les professionnels

Les compagnies d'assurance

Elles sont susceptibles d'être intéressées par l'application afin d'enrichir leur offre auprès de leur client. L'application serait incluse dans un contrat d'assurance (habitation par exemple) et constituerait un atout marketing différenciant majeur.

Les entreprises

Certaines entreprises pourraient se voir intéressées par Arkage dans le but d'accélérer leur temps de développement. L'intérêt d'avoir un noyau prêt et fonctionnel, gérant lui-même ses permissions pourrait intéresser certaines compagnies.

Personas

Aline Moreau



"J'adore être avec des personnes qui me font oublier de regarder mon téléphone"

Age: 35 ans
Situation: En activité, vendeuse dans un magasin de vêtements
Lieu de vie: Rennes
Téléphone: Huawei p30
Nombre d'applications: Entre 15 et 20

Aisance (smartphone)

80%

Habitude

Mode son Mode silencieux

Décontractée

Réfléchie

Patiente

Informations confidentielles sur son smartphone

- Application bancaire.
- Mails
- Pièces d'identité

Bio

Aline vit à Rennes depuis 3 ans. Elle travaille dans un magasin de vêtements. Elle est mariée et a deux enfants de 4 et 8 ans.

Aline a toujours son téléphone sur elle, au cas où elle recevrait un message ou un appel urgent. Elle adore prendre des photos de ses moments en famille et avec ses amis. Elle a déjà perdu son téléphone il y a une dizaine d'années. Elle n'a jamais pu le retrouver car elle n'avait pas les moyens pour !

Aline espère ne plus jamais perdre son téléphone car on y trouve de nombreuses informations confidentielles.

Marques préférées



Freins

- Peur d'une application compliquée à utiliser
- Ne pas se rappeler du mot de passe utilisé

Pierre Dupont



"Je n'imagine même pas une journée sans mon téléphone..."

Age: 21
Situation: Étudiant
Lieu de vie: Angers
Téléphone: Samsung
Nombre d'application: + de 30

Aisance (smartphone)



Habitude (smartphone)



Stressé Étourdi Dynamique

Informations confidentielles sur son smartphone :

- Application bancaire
- Application de transaction (Pumpkin...)
- Pièces d'identité
- Documents confidentiels
- Carte bancaire

Bio

Pierre est un étudiant angevin. Très à l'aise lorsqu'il s'agit d'utiliser son smartphone, sa plus grande crainte est de se faire voler ou perdre celui-ci.

En effet, il s'agit du premier téléphone que Pierre s'est acheté avec ses propres sous, après avoir perdu celui qu'il avait.

Pierre a également perdu son téléphone une autre fois, mais heureusement pour lui, en rebroussant chemin, il l'a retrouvé.

Marques préférées



Frustrations :

- Oublier que l'application est installée sur son téléphone
- Peur qu'on le localise à cause de cette application
- Peur que l'on récupère ses informations de localisation via l'application

Référencement

//TODO

Charte éditoriale

Les textes ci-dessous ont été reformulé et extrait de la rédaction de la charte éditoriale par Camille Esneault.

Notre histoire

L'application Arkage est une idée de Téo Conan. Ayant pour habitude de toujours perdre son smartphone, il a imaginé un terminal Android via SMS permettant de trouver facilement son téléphone.

L'équipe chargée de la création d' Arkage est réunie en janvier 2019. Elle imagine une application ergonomique, simple dans son utilisation et surtout, utile au quotidien.

Spécialiste du digital, l'équipe d'Arkage entend bien proposer une application sécurisant l'utilisation du téléphone au quotidien.

Nos valeurs

Transparence, Simplicité, Innovation

L'application Arkage doit être facile d'utilisation. En quelques minutes, l'utilisateur doit pouvoir sécuriser son téléphone en cas de perte ou de vol. Pour ce faire, il a été imaginé un système performant, qui protège le téléphone et permet de le géolocaliser sans collecter des données.

Présentation de l'organisation interne

L'équipe chargée de la rédaction du contenu est composée de Swann Lambard et Camille Esneault. Ils produisent le contenu rédactionnel, soumis à la validation du responsable de projet, Téo Conan. Emma Hervé est chargée de la production du contenu visuel : illustrations, choix des photographies...

Stratégie de communication

Le Content Marketing a pour objectifs :

- faire connaître l'application Arkage et ses différentes fonctionnalités
- Expliquer le fonctionnement d'Arkage
- Rassurer l'utilisateur
- Informer sur l'évolution de la cybersécurité
- Recruter de nouveaux utilisateurs
- Fidéliser les utilisateurs actifs

Les KPI à surveiller

Sur l'application

- Nombre de téléchargements

Sur le site web

- Temps de lecture
- Taux de rebond
- Parcours de l'internaute
- Nombre de clics sur la page
- Taux de conversion vers une landing page

Sur les réseaux sociaux

- Engagement
- Portée
- Nombre d'abonnés
- Vues de la page
- Recommandations

Rappel des (principaux) supports utilisés

- Site web / landing page
- Réseaux sociaux : publications (photos, infographies, vidéos)
- Publicités payantes sur les réseaux sociaux

Notre vision éditoriale

Thème 1 : Actualités d'Arkage

Thème 2 : Histoires de téléphone retrouvé

Thème 3 : Cybersécurité

Mots-clés

- sécurité
- vol téléphone
- perte téléphone
- sms
- simple
- géolocalisation
- téléphone
- smartphone
- android

Hashtags courants : [#smartphones](#) [#android](#) [#samsung](#) [#samsunggalaxyflip](#)
[#samsunggalaxys](#)

Hashtags personnalisés : [#arkage](#) [#teamarkage](#) [#simplyarkage](#)

Notre ton de communication

- Pédagogique / explicatif
- Détendu

Bonnes pratiques

1. Penser avant tout à l'utilisateur final afin de le garder en ligne et de le fidéliser. Informer mais ne pas ennuyer.
2. Respecter les règles de référencement naturel : titres, paragraphes (h1, h2, h3...),etc.
3. Adopter un style fluide, compréhensible avec des phrases courtes.
4. Illustrer les propos avec une image, illustration ou vidéo.

Types de contenus à produire

- Articles de blog : 1x /semaine
- Publications sur les réseaux sociaux : 2x/semaine
- Textes explicatifs

- Infographies
- Vidéos / motion design

Vocabulaire

Eviter le vocabulaire technique

Eviter de dire trop de fois “collecte de données”

Ne jamais être négatif

Jamais de “je”

Pas d’anglicisme

Licence logiciel

Les textes ci-dessous doivent être étudiés par des professionnels pour savoir si ce qui suit est réalisable et légal.

Certain des éléments exprimé ci-dessous ne seront disponibles que dans des versions d’Arkage ultérieures. Ceci est une première ébauche de la notion de licence Arkage

Introduction

Arkage doit être soumis à certaines restrictions. Le but n’est pas qu’Arkage soit totalement libre.

Il doit être libre pour l’usage personnel, de manière à encourager une communauté à utiliser Arkage, mais cependant si une entreprise souhaite utiliser le middleware Arkage elle devra payer une licence pour pouvoir l’utiliser et l’intégrer dans son projet.

Ce qu’il faut retenir, c’est que si vous gagnez de l’argent grâce / au travers d’Arkage ou que vous êtes une entreprise à but non-lucratif, vous êtes dans l’obligation d’acheter une licence Arkage.

Utilisation personnelle ou utilisation pour les associations à but non-lucratif

Si vous êtes un utilisateur utilisant et / ou modifiant Arkage vous avez certains droits et certaines interdictions.

Vous êtes autorisé à corriger des bugs présents dans le Kernel.

Pour proposer vos modifications, vous pourrez faire une “pull request” sur Github.

Après ceci, si votre modification est acceptée, vous serez crédités sur une page dédiée du site d’Arkage.

Vous ne pourrez cependant pas proposer vos modifications ailleurs, le but est d’éviter la propagation de programmes malveillants.

Vous êtes aussi autorisé à développer vos propres modules, commandes terminaux ou encore librairies. Nous en reparlerons plus tard.

Utilisation professionnelle ou à but lucratif

Si vous êtes un utilisateur générant du profit grâce ou au travers d'Arkage vous serez soumis aux mêmes règles qu'une entreprise.

Si vous êtes une entreprise à but lucratif vous serez soumis à certaines règles différentes des utilisateurs.

Vous serez dans l'obligation d'acheter une licence logiciel pour pouvoir utiliser Arkage.

Publication du code

Si vous souhaitez partager votre code sans l'intégrer de manière permanente à Arkage, vous pourrez publier votre code où vous le souhaitez (si possible GitHub).

Suite à ça, il vous faudra faire une demande de vérification auprès d'Arkage

Nous vérifierons si votre code est inoffensif et publierons le lien du dépôt sur le site d'Arkage dans un espace dédié.

Le but est de créer une liste de dépôts "sécurisé".

Licence de version ou licence continue

*"Une **licence de logiciel** est un **contrat** par lequel le titulaire des droits d'auteur sur un **programme informatique** définit avec son cocontractant (exploitant ou utilisateur) les conditions dans lesquelles ce programme peut être utilisé, diffusé ou modifié."* - Wikipédia

Il existe 2 types de licence pour les entreprises.

Les licences dites "de version" signifie qu'une entreprise achète à un prix fixe Arkage en l'état actuel du développement. Ces entreprises auront donc de la version "stable" d'Arkage mais ne bénéficieront pas d'améliorations.

Les licences dites "continue" sont des licences qui se payent tous les mois et permettent aux entreprises de bénéficier des améliorations du code et d'aides ou de conseils de la part d'Arkage (limité à un nombre d'heures par mois)

Identité visuelle

Brief créatif

Le texte ci-dessous a été rédigé par Emma Herve.

Arkage est une solution permettant aux utilisateurs de retrouver leur téléphone en cas de perte ou de vol. Aucune des données des utilisateurs ne vont être stockées.

Les valeurs de l'entreprise vont alors tourner autour de : la sécurité, la fiabilité, l'innovation et surtout de la transparence. Arkage est une application technologique poussée. C'est pour cela que le logo doit être à la fois moderne et épuré.

Nous avons choisi de représenter le logo symboliquement. Par la couleur, la forme nous pouvons lui attribuer un sens et cultiver une émotion autour de notre marque. Il nous semblait également important de combiner le logo au nom de la marque. Le texte et le symbole travaillent ensemble pour renforcer notre image. Les gens vont pouvoir commencer à associer notre nom à notre marque picturale. Dans le futur il sera ainsi beaucoup plus facile d'exclure le nom du symbole.

Notre entreprise se base sur la transparence. Pour cette raison, nous avons décidé que notre logo sera représenté par une goutte d'eau. Aussi, une fois retourné, nous pouvons deviner le symbole de géolocalisation.

La couleur choisie nous semblait évidente : du bleu. Le bleu symbolise la fiabilité et la maturité. Cela va nous aider à ce que notre marque soit prise au sérieux et inspire la confiance.

Charte graphique

La totalité de la charte graphique a été réalisé par Emma Herve, le document complet de la charte graphique se trouve en annexes, cependant voici quelques extraits :

//TODO

Spécifications fonctionnelles

Arkage Kernel

L'intérêt du noyau Arkage est de créer un système totalement indépendant et autonome, modulable, capable de s'adapter au matériel sur lequel il est installé.

Le noyau gère les modules du système, il les charge, les exécute etc...

Le système en question doit être capable d'accueillir des améliorations / des programmes en provenance de personnes extérieures à Arkage.

Le système doit aussi gérer les "Arkage Package Scenarios".

Arkage Package Scenario

Pour définir ce que sont les APS, il est nécessaire de parler un peu de Linux. Dans un système Linux, le système a ses propres commandes, mais l'utilisateur peut aussi installer des paquets qui permettent d'ajouter des commandes au terminal, par exemple "git" sous Linux, qu'il faut installer pour l'avoir en console.

Arkage fonctionne en partie de la même façon. Le système a ses propre commandes mais il est aussi possible de créer des "custom command".

Si nous reprenons l'exemple de la situation initiale, avec l'utilisateur qui perd son smartphone durant sa promenade, nous parlions d'une commande qui ressemblait à ceci : **“Arkage je suis perdu 6519”**, cette “custom command” n'est évidemment pas une vraie commande : c'est une commande “user-friendly”, en vérité “Arkage je suis perdu {password}” n'est ni plus ni moins que le raccourci de : **“arkage app stolen-mode”**. Le but des “custom command” est uniquement de permettre aux utilisateurs de déclencher des programmes d'une manière plus simple que de devoir taper une vraie commande Terminal. Tous les utilisateurs ne sont pas capable de se servir d'un terminal.

Les APS pourront être créer en interne par notre équipe ou bien par des personnes extérieures à Arkage dans le but de créer leurs propres commandes ou applications.

Les modules systèmes

Comme nous l'avons dit précédemment, le Kernel gère les modules. Mais qu'est-ce qu'un module ?

Un module est comme un composant hardware d'un ordinateur. Il est modulable, interchangeable mais aussi indispensable.

Sans module, le Kernel ne sert absolument à rien. Il sera comme une armature vide.

Les modules interagissent entre eux, et ont besoin des uns et des autres pour pouvoir fonctionner et offrir des fonctionnalités.

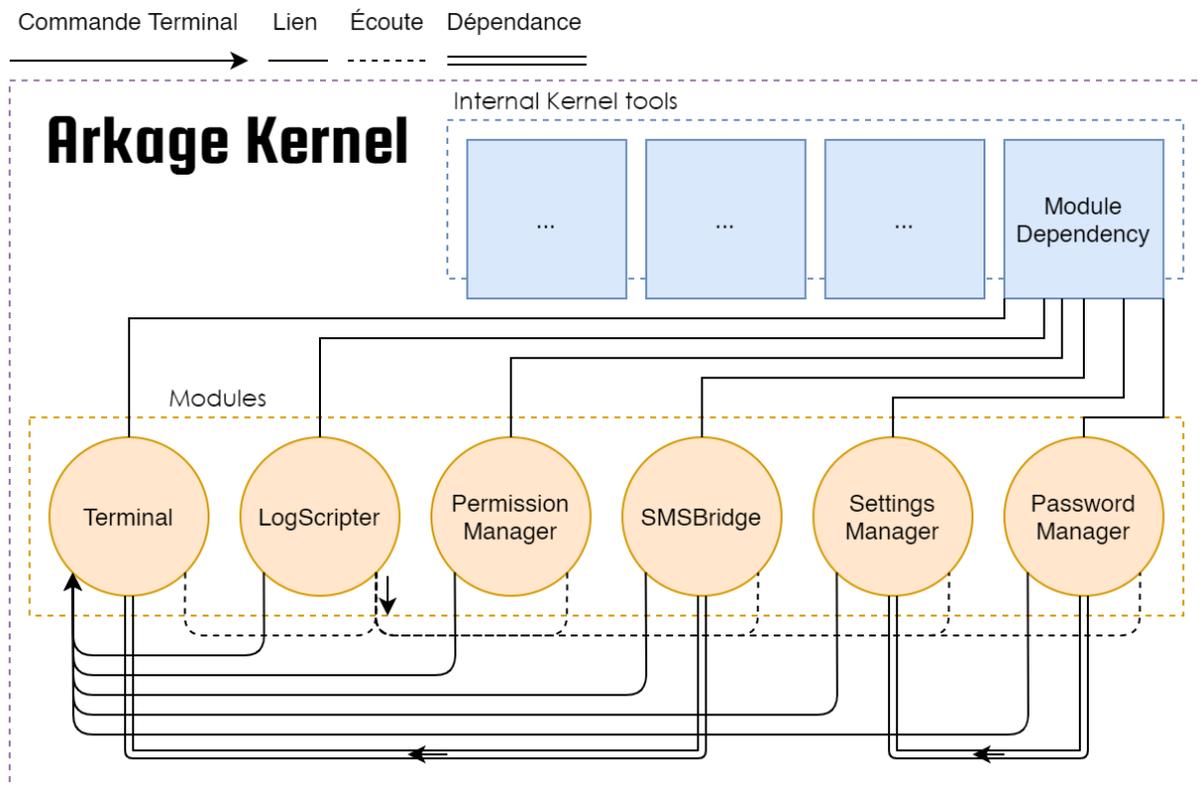


Illustration d'une partie du Kernel avec ses échanges inter-modules et les outils du Kernel

Terminal

Le module du Terminal est au coeur de l'interaction avec l'extérieur de l'application. Le Terminal est capable de modifier des paramètres du Kernel, d'exécuter des commandes, d'exécuter des commandes Linux, d'exécuter des "custom command" et de renvoyer le résultat de toutes ces commandes. Le Terminal est aussi en mesure de retourner des erreurs dans la formation de la commande et les erreurs retournées durant l'exécution de la commande.

Shell Grammar

Le Shell Grammar, autrement dit la façon dont sont écrites les commandes et comment elles sont interprétées, est basé sur celui de Linux. Nous reverrons comment il doit fonctionner techniquement dans les spécifications techniques.

SettingsManager

Le module SettingsManager est le module qui gère tous les paramètres et les préférences du Kernel à enregistrer en base de données.

Son utilisation au sein du code doit être simple et automatisée, de manière à ce que les développeur ne s'en soucie pas.

PasswordManager

Le PasswordManager est un module assez simple qui a pour but de gérer les mots de passe du Kernel (primaire et secondaire)

Ce module est capable de confirmer si un mot de passe est bon, de définir les différents mots de passe et de sélectionner quel mot de passe est à utiliser entre le mot de passe primaire et secondaire.

Mots de passe primaire et secondaire

Reprenons notre situation initiale avec le téléphone perdu dans un parc.

Comme nous disions l'utilisateur emprunte le téléphone "A" d'une personne (qu'il ne connaît probablement pas) dans la rue pour pouvoir envoyer une commande à son téléphone perdu "B".

Pour qu'une commande soit envoyée par SMS, il faut impérativement le mot de passe de l'application dans le SMS pour pouvoir l'exécuter. Sans quoi aucune réponse ni signe de vie ne sera renvoyé par le Kernel.

Ceci permet d'éviter des tentatives d'intrusion de la part de personnes qui tenteraient d'envoyer les commandes Arkage à des numéros au hasard.

Cependant le problème réside dans le fait que le mot de passe est écrit dans le SMS en clair ! Voici un exemple :

"Arkage je suis perdu 6519"

Une fois le téléphone "A" rendu, son propriétaire peut relire les SMS envoyés par l'utilisateur. Et par ailleurs avoir le contrôle d'Arkage en utilisant le mot de passe du SMS.

L'idée d'avoir un mot de passe primaire et secondaire au sein de l'application.

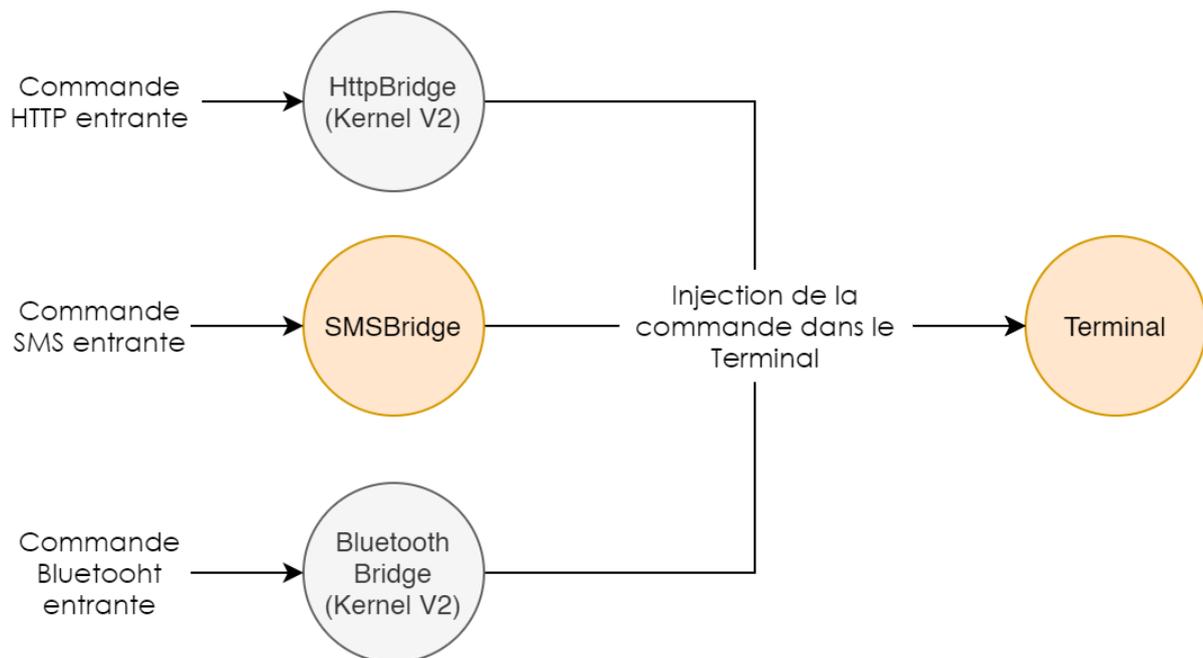
Une fois le SMS avec la commande "**Arkage je suis perdu 6519**", l'utilisateur passera l'application sur le mot de passe secondaire, de manière à ce que le mot de passe "6519" soit erroné mais que l'utilisateur puisse toujours avoir accès à Arkage

Nous illustrerons et reviendrons dans les détails lors des spécifications techniques.

SMSBridge

Le module SMSBridge a un rôle très simple : la redirection et la transformation des SMS vers le module Terminal.

Dans notre situation initiale, il était question d'un SMS qui faisait office de commande. En temps normal ce n'est pas possible car le Terminal n'est pas relié à la réception de SMS. C'est donc là le rôle du SMS Bridge.



PermissionManager

Le PermissionManager est un véritable atout au sein du Kernel. Le rôle de ce module est d'obtenir les différentes autorisations auprès d'Android.

Ce module est capable de :

- Donner l'état d'une permission spécifique (refusée, acceptée etc...)
- Demander à l'utilisateur d'accepter la permission (au travers d'une pop-up)
- Désactiver l'utilisation d'une permission auprès du Kernel

Rapport d'utilisation

Arkage est une technologie qui a besoin d'une optimisation très poussée, si l'application consomme trop de batterie sur le téléphone ou trop de CPU, l'application ne serait pas gardée par les utilisateur

Pour comprendre au mieux comment les utilisateurs se servent de l'application et comment l'optimiser en fonction de leurs besoins, nous allons avoir besoin de collecter des données anonymes sur l'utilisation d'Arkage

Un cahier des charges dédié aux API de Reporting sera disponible à la suite de ce document.

Arkage UI

Arborescence

Interface "Welcome Page"

Lors de la première utilisation, l'utilisateur sera confronté à une page de configuration Dans cette page sera demandé les informations suivantes :

1. Demander le nom du téléphone
2. Configuration du mot de passe primaire
3. Configuration du texte affiché en mode perdu (+ bouton preview)
4. Configurer le numéro de secours (opt)
5. Fin de la configuration ou afficher les options dev

Sur chacunes de ces étapes, une image est affichée, ci-dessous sera présent un indicateur de progression, un texte explicatif, un formulaire de configuration (inputbox, checkbox etc...) puis enfin un bouton "suivant"

Interface "Lock Page"

Interface "Home Page"

L'interface home page de l'application est l'interface principale de l'application. C'est depuis cette page que l'utilisateur aura accès à toutes les fonctionnalités de l'application.

L'ergonomie de la home page se décline en plusieurs parties :

Partie menu inférieur

Le menu inférieur comporte 3 boutons avec un icône (sans texte), un de chaque côté et un central surélevé

Les boutons sont les suivants :

1. Page d'autorisation (côté gauche)

2. Bouton retour home page
3. Page d'option (côté droit)

Partie menu supérieur droite

Un menu d'icônes est présent en haut à droite de l'écran
Il comportera un bouton redirigeant vers la page de documentation

Comportement gestuel

Swip de haut en bas : lancement de la page de documentation
Swip de gauche à droite : lancement de la page des options
Swip de droite à gauche : lancement de la page des autorisations

//TODO schéma comportement gestuel

Interface "Options"

La page des options regroupe toutes les options de l'application et des applications Arkage
L'affichage est sous forme de liste, similaire aux paramètres Android ou IOS

Liste des options

1. Options des modes (section)
 - a. Lost Mode (section)
 - b. Stolen Mode (section)
2. Mon téléphone (section)
 - a. Configuration des numéros de secours (pop up input box)
 - b. Configuration des coordonnées de secours (pop up input box)
3. Mots de passes (section)
 - a. Mot de passe primaire (pop up input box)
 - b. Mot de passe secondaire (pop up input box)
4. Afficher les logs (tableau)
5. Rapports d'utilisation (section)
 - a. Envoyer les rapports d'utilisation (switch)
 - b. Voir le dernier rapport d'utilisation (bouton)
6. Mode dev (switch) (**options du mode dev ci-dessous**)
7. Terminal Arkage (activité Terminal UI)
8. Documentation du terminal (section)
 - a. Chapitre 1...
9. Options Logs Scriptor (section)
 - a. Désactiver les logs (switch)
 - b. Activer l'encryption (switch)
 - c. Exporter les logs (interface Exportation des logs)

Interface "Autorisations"

Cette page est composée d'une liste des autorisations possible d'activer
L'état de l'autorisation est affichée (autorisé / non-autorisé)

Un clique sur une autorisation non-acceptée lance la demande d'autorisation

Listes des autorisations

- Device Administrator
- android.permission.CHANGE_WIFI_STATE
- android.permission.ACCESS_WIFI_STATE
- android.permission.CAMERA
- android.permission.DISABLE_KEYGUARD
- android.permission.WRITE_EXTERNAL_STORAGE
- android.permission.READ_EXTERNAL_STORAGE
- android.permission.ACCESS_COARSE_LOCATION
- android.permission.ACCESS_FINE_LOCATION
- android.permission.RECEIVE_SMS
- android.permission.WAKE_LOCK

Interface "Documentation"

L'interface de documentation a pour but d'aider l'utilisateur avec une documentation rédigée pour expliquer le contexte, le fonctionnement et le principe d'Arkage. Un utilisateur peut facilement réussir à se servir de l'application sans pour autant comprendre à quoi elle sert.

Le but est d'éviter que l'utilisateur finisse par se poser comme question :

"D'accord je comprend comment fonctionne l'application et ses interfaces... Mais à quoi sert cette application ?? C'est quoi Arkage ?"

La documentation est donc "grand public", avec pour arborescence :

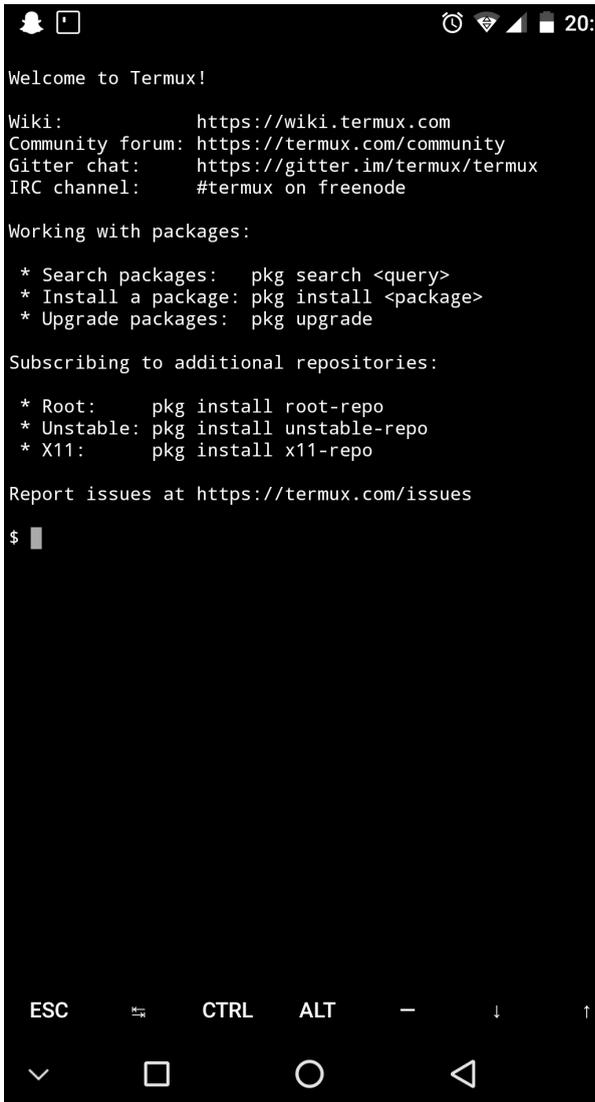
1. Comment fonctionne...
 - a. Arkage ? (raccourcis vers "explication")
 - b. Le mode "Lost" ?
 - c. Le mode "Stole" ?
2. Explications (fonctionnement et principe d'Arkage)
3. Aide et questions fréquentes
4. CGU
5. Mentions légales
6. Documentation développeurs (lien internet)

Interface “Terminal UI”

L'interface “Terminal UI” d'Arkage est un émulateur de terminal connecté au Kernel. Grâce à cette interface, un utilisateur / développeur peut tester les commandes du terminal.

Cette interface n'a pas d'autre réel but que de laisser l'utilisateur essayer le terminal du Kernel.

L'image ci-contre est une capture d'écran de l'application Android “Termux”



```
Welcome to Termux!

Wiki:           https://wiki.termux.com
Community forum: https://termux.com/community
Gitter chat:    https://gitter.im/termux/termux
IRC channel:    #termux on freenode

Working with packages:

* Search packages:  pkg search <query>
* Install a package: pkg install <package>
* Upgrade packages: pkg upgrade

Subscribing to additional repositories:

* Root:           pkg install root-repo
* Unstable:       pkg install unstable-repo
* X11:            pkg install x11-repo

Report issues at https://termux.com/issues

$ █
```

Interface “Exportation des logs”

Cette interface a pour but d'offrir quelques outils pour exporter les logs du téléphone. Les options disponibles sont :

- Destination, ou où enregistrer le fichier une fois exporté, par défaut dans le dossier téléchargement.
- Sélectionner l'intervalle d'exportation des logs, par exemple du 23/04 à 18h00 au 29/04 12h00

Arkage Website

Arborescence

Page {nom de la page}

Reporting API & Databases

Des APIs sont nécessaires pour permettre l'envoi des rapports d'utilisation entre le téléphone et les serveurs.

Une base de données sera aussi hébergée sur le serveur pour stocker les rapports.

Tests unitaires

L'application d'Arkage sera un système complexe qui pourra tomber en erreur de multiples façons. Il n'est pas pensable de ne pas intégrer de tests unitaires dans ce projet. C'est grâce à ces tests que nous pourrions prouver la fiabilité de notre application.

Spécifications techniques

Arkage Kernel

//TODO schéma du fonctionnement du Kernel

Service du Kernel

Arkage est une application qui a besoin d'être utilisée dans tous les cas possibles et surtout en arrière plan, pour ce faire android nous propose plusieurs options pour mettre en oeuvre une application en arrière plan et ce, via l'utilisation de service android.

Ces dits services vont permettre d'encapsuler l'application et de la faire tourner dans un plan où elle sera accessible de toute part. Nous avons beaucoup de choix pour créer des applications en arrière plan néanmoins avec l'arrivée de l'API 26 beaucoup d'inconvénient s'ajoute à ces derniers, de ce fait pour arkage nous avons décidé d'utiliser un **Foreground Service**, qui va permettre de garder le service actif dès lors de sa création, sans arrêt brut d'android ou de lui-même, couplé avec l'utilisation de **Runnable** où nous allons créer un nouveau thread pour exécuter nos opérations et ainsi ne pas surcharger le thread principal (UI thread). Cette solution est parfaite pour notre produit elle oblige par contre l'utilisation du système de notification.

***Foreground service** : Le Foreground service est un service dit de “premier plan” c’est à dire qu’il s’exécute au premier plan et y reste jusqu’à une interruption de se dernier par lui même (une fonction stopService) ou par l’utilisateur

***Runnable**: L’interface android Runnable représente les objets qui sont capables de faire exécuter du code au processeur. Elle ne possède qu’une méthode, `void run()`, dans laquelle il faut écrire le code à exécuter.

Arkage Package Scenario

Terminal

La fonction principale du Terminal est la méthode “shell”. C’est par cette méthode que sont exécutées toutes les commandes.

Lorsqu’une commande est entrée, disons par exemple “arkage help”, le programme va créer un objet “Command” en fournissant le texte en entré.

Le Terminal va ensuite fournir les commandes par défaut d’une commande.

Pour donner un exemple d’option par défaut, il y a l’option “text” qui spécifie que le texte de sortie attendu doit être sous format texte, ou “user-friendly”. Cette option est l’opposé de l’option “dev” qui permet d’obtenir un texte de sortie au format “dev” (ou plutôt au format JSON en l’occurrence)

Prenons quelques exemples :

Précisons que les exemples ci-dessous ne sont que des exemples et ne représentent pas la réalité

Voici à quoi ressemble une commande en mode “text”

```
$ arkage help
> usage: arkage [--version] [-h | --help] [-d | --destination] [-p
| --password] [-x | --text] [-v | --dev] [--list-actions]

You can find below basics Arkage commands:
[...]
To list all options, please run 'arkage --list-actions'
```

Et voici la même commande mais en mode “dev”, qui sera exploitable par un programme

```
$ arkage help --dev
> {
  "input": "arkage help",
  "availableOptions": [
    {
```

```

        "shortcut": "x",
        "name": "text",
        "isOrphan": true
    },
    {
        "shortcut": "v",
        "name": "dev",
        "isOrphan": true
    }
    [...]
],
"availableActions": [
    [...]
],
"text": [
    "To list all options, please run 'arkage
--list-actions'",
]
}

```

Après avoir donné les options par défauts, le Terminal va découper la commande selon le **Shell Grammar**

Shell Grammar

Pour concevoir le Shell Grammar du Terminal, nous avons étudié la façon dont Linux lit ses commandes et nous en sommes arrivé à ce schéma :

[action] [[options] [optionsValues]] [settings]

Il est à savoir que ce schéma est interchangeable. Il n'est pas impossible de faire passer [settings] avant [options], ce qui donnerait : [action] [settings] [[options] [optionsValues]]

Exemples

[/Préfix/] [Action] [[Tableaux d'options] [Tableau de valeur]]
 [Tableau de settings]

Commande Linux :

- tar -cvf Arkage.tar Arkage/
- git checkout arkage

Commandes d'exemples :

- exec -rsp wifi.toggle true 0505
- exec -t gps.pos -p 4432 true

- `exec -trp gps.pos 1598`
- `switch-password -p 5168`
- `very-difficult-command setting1 -h setting2 -p 5174`
`--first-setup false setting3`

La dernière commande des exemples vous à peut-être paru étrange ou malformée à cause de l'option "-h". En effet si on suis le schéma, cette option ne semble pas avoir de valeur, puisqu'à en croire la syntax, ce qui suit -h est setting2. Si on suit le premier pattern pourtant il aurait été correct d'écrire setting1 -h valueOfH setting2
 La syntax de l'exemple est bonne, et ceci grâce aux **options orpheline**

Les options orphelines

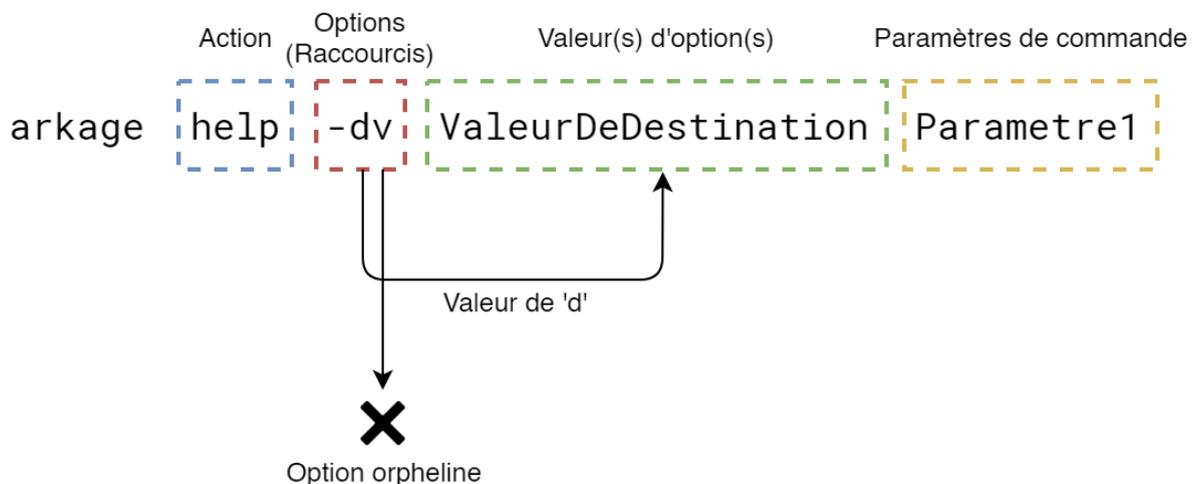
Parfois dans les commandes, la présence de la clé de l'option suffit, prenons un exemple :
`arkage start --first-setup`

Dans cet exemple, nous avons l'option "--first-setup" qui n'a pas de valeur. En effet il aurait été plus logique d'un point de vue du programme d'écrire comme suit :

```
arkage start --first-setup true
```

Dans le but d'attribuer "true" à l'option "--first-setup", nous tombons sur une **"option orpheline"**

Le programme comporte une liste d'option dite "orpheline", par ceci on entendra qu'elle n'ont pas besoin de valeur.



Le découpage des commandes

SettingsManager

Le SettingManager fonctionnera d'une manière assez simple : il stocke les paramètres de l'application dans le "SharedPreferences" d'Android.

<https://developer.android.com/training/data-storage/shared-preferences>

Les settings sont le plus souvent composé d'une clé-valeurs. La clé doit respecter un nommage normalisé.

Nommage de la clé de setting

[[MODULE_NAME] | [APP_NAME] | USER | ARKAGE]_[ACTION]_[TARGET]
[[MODULE_NAME] | [APP_NAME] | USER | ARKAGE]_[VARIABLE_NAME]

Liste des Settings

Clé et description	Valeur par défaut
LOG_SCRIPTER_ENABLE_LOGS	true
LOG_SCRIPTER_ENCRYPT_LOGS Encrypter les logs, n'est pas toujours possible selon l'OS	false
LOG_SCRIPTER_LOG_PATH Chemin de fichier des logs	logs
ARKAGE_SYSTEM_STORAGE Dossier d'Arkage dans le stockage interne	/Android/data/com.arkage.system/
PASSWORD_MANAGER_CURRENT_PASSWORD_ID Utilisation courante du mot de passe (primaire (1) ou secondaire (2))	1
USER_RECOVERY_NUMBER Numéro de secours de l'utilisateur	null
USER_RECOVERY_SECONDARY_NUMBER Numéro secondaire de secours de l'utilisateur	null
USER_RECOVERY_CONTACT_INFORMATIONS Texte affiché à l'écran lorsque perdu	null
ARKAGE_NEED_CONFIGURATION Défini si l'application a besoin d'être configurée	true
ARKAGE_IS_WELCOME Si l'utilisateur ouvre l'application pour la première fois	true
SMS_BRIDGE_PREFIX_COMMAND Préfix des commande pour le SMSBridge	arkage
PERMISSION_MANAGER_ENABLE_ALERTS Activer les alertes du PermissionManager	true

PasswordManager

Mots de passes primaire et secondaire

Voici donc techniquement comment fonctionne l'interchangement des mots de passes primaire et secondaire :

Imaginons que l'utilisateur a défini :

- Mot de passe primaire : 6519 (actif)
- Mot de passe secondaire : 4886

1 - L'utilisateur lance le mode "lost" depuis le téléphone d'un inconnu

```
$ arkage app lost --password 6519  
> [...]
```

Le mot de passe fonctionne, le Kernel exécute le mode "lost"

2 - L'utilisateur se protège et interchange les mot de passes pour que le propriétaire ne puisse pas se resservir du mot de passe

```
$ arkage switch-password --password 6519  
> Le mot de passe d'Arkage est maintenant le mot de passe  
secondaire
```

Ce qui nous donne :

- *Mot de passe primaire : 6519 (ce mot de passe ne fonctionne plus)*
- *Mot de passe secondaire : 4886 (actif)*

3 - L'utilisateur rend au propriétaire son téléphone

4 - Le propriétaire relis les SMS et tente d'exécuter une commande avec le mot de passe écrit en clair :

```
$ arkage destroy-smartphone --password 6519  
>
```

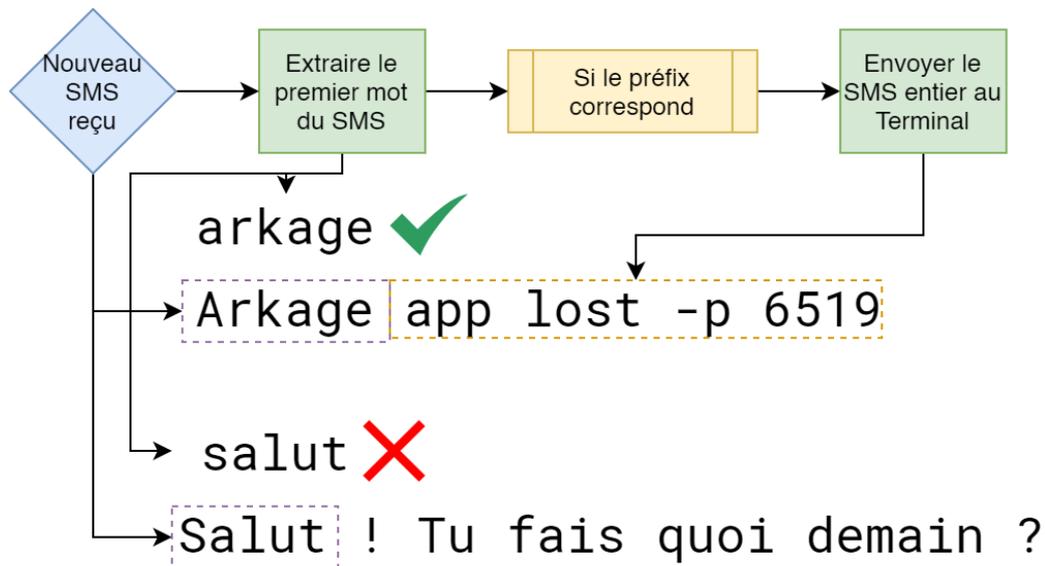
Aucune réponse ni signe de vie de la part du Kernel car le mot de passe ne sera pas le bon (4886 était attendu)

PS : La commande "destroy-smartphone" n'existe évidemment pas

SMSBridge

Le SMSBridge à un fonctionnement plutôt simple. Il doit lire tous les SMS en récupérant le premier mot, si celui-ci correspond au préfix de commande (par défaut : "arkage"), alors il analyse le SMS pour y récupérer le mot de passe. Il sera ensuite vérifié par le PasswordManager.

Si le PasswordManager confirme le mot de passe, alors la commande sera injecté dans le Terminal et le résultat de cette commande sera renvoyé par SMS à l'adresse de provenance.



PermissionManager

Le PermissionManager demandera à l'utilisateur toutes les autorisations nécessaires, puis par la suite les testera au travers d'une batterie de tests pour être sûr de leur fonctionnement.

L'utilisateur n'est pas obligé d'accepter toutes les autorisations, en contreparties certaines fonctionnalités pourrait être comprise à cause de certaines autorisations refusée. Le rôle du PermissionManager sera d'alerter l'utilisateur si criticité il y a.

Si une fonctionnalité, à savoir une commande ou une librairie, tombe en erreur à cause d'une autorisation manquante, alors une ligne est ajoutée dans les logs d'erreurs.

Si le Kernel souffre d'une autorisation non-accordé et tombe en erreur, alors une pop-up est envoyé par le PermissionManager et propose à l'utilisateur d'accéder aux paramètres.

Les alertes pop-up du PermissionManager sont désactivable dans les paramètres de l'application, mais par défaut cette option est activée.

Rapport d'utilisation

UX & UI

Par défaut le reporting n'est pas activé. Pour l'activer l'utilisateur devra aller dans les options de l'application et sélectionner la section "Rapport d'utilisation"

Dans cette section, l'utilisateur pourra :

- Activer / désactiver l'envoi des rapports d'utilisation hebdomadaire
- Consulter le dernier rapport envoyé
- Obtenir un lien pour consulter ses données en ligne

Si l'utilisateur décide de désactiver le reporting, ses données stockées sur l'appareil sont supprimées

Les données stocké sur le téléphone et consultable pendant maximum 14 jours

Premier rapport d'utilisation

Lorsque l'utilisateur "souscrit" à partager ses rapports d'utilisation, l'application va envoyer les informations suivantes aux serveurs d'Arkage :

- Nom de l'OS
- Version de l'OS
- Capacité du CPU (cores + Hz)
- RAM de l'appareil

Contre ces informations, les serveurs d'Arkage renverront un "token", une sorte d'ID généré aléatoirement.

C'est grâce à ce token que les statistiques envoyés par ce téléphone seront regroupé, sans pour autant être capable d'identifier le téléphone

Données récoltées pour un rapport

- Activité du noyau Arkage
Ces données informent les heures de démarrage / arrêt du système Arkage
- Activité des entrées Terminal
Ces données précisent les heures où le noyau a reçu des interactions avec le Terminal ainsi que le sous-système Arkage qui a été impliqué.
- Erreurs interne
Le rapport de toutes les erreurs survenue avec les détails de l'erreur, le sous-système du noyau impliqué, la description fourni par le sous-système et l'heure de l'apparition de l'erreur

Stockage sur les serveurs Arkage

Les rapports seront envoyé hebdomadairement aux serveurs d'Arkage au travers d'une API web. Les données seront stocké sur le serveur pendant maximum 6 mois.

Le but de ces données seront de générer des résumés et des statistiques.

Structures des bases de données

Il serait préférable que la base de données qui regroupe tous les rapports d'utilisation soit dans une base dédié.

Cette base devra s'appeler "**usage_report**"

Table des utilisateurs du reporting

Pour structurer ces données, il faut une table qui regroupe tous les utilisateurs du reporting d'Arkage. Il n'est pas question d'enregistrer des noms, mais seulement des "tokens" pour les identifier.

Voici les différentes informations qui devront être disponible par lignes :

Cette table s'appellera "**user_report**"

Type	Nom	Description
primary id	id	ID unique de l'utilisateur dans le système
string	token	Token d'utilisation pour l'échange avec les API
string	os_name	Nom de l'OS (Android, Windows, Linux etc...)
string	os_version	Numéro d'API pour Android (ex 21)
int (mo)	total_ram	RAM installée sur l'appareil en Mo
int (MHz)	cpu_frequency	Fréquence des coeurs du CPU (en MHz)
int	cores_number	Nombre de coeurs disponible dans le CPU
int	time_create	Heure de création de l'utilisateur
int	time_update	Heure de la dernière mise à jour de l'utilisateur
int	time_delete	Heure de suppression de l'utilisateur

Table des rapports

Cette table à pour but d'identifier les rapport, à qui ils appartiennent mais aussi les heures à laquelle il a été enregistré

Cette table d'appellera "**report**"

Type	Nom	Description
primary id	report_id	ID de la ligne
id	user_id	Utilisateur auquel cette information est attachée
time	time_reported	Timestamp qui désigne le moment où l'info à été enregistré sur le serveur

Table des états du Kernel

Les états du Kernel seront enregistré dans une seule table avec une colonne dédiée à l'id des utilisateurs.

Cette table s'appellera "**kernel_state**"

Type	Nom	Description
primary id	state_id	ID de la ligne
id	report_id	ID du rapport auquel il est rattaché
string	action	Action du Kernel (boot stop crash...)
string	reason	Raison pour laquelle l'action est arrivée (user os error)
time	time	Heure à laquelle l'information a été enregistrée
time	time_reported	Timestamp qui désigne le moment où l'info à été enregistrer sur le serveur

Table des entrés Terminal

Cette table répertorie toutes les entrés Terminal. Le but est de savoir la fréquence des requêtes du Terminal.

Cette table s'appellera "**terminal_input**"

Type	Nom	Description
primary id	input_id	ID de la ligne
id	report_id	ID du rapport auquel il est rattaché
string	type	Type d'entrée dans le Terminal (habituellement "command" ce champs est prévu en cas d'évolution)
string	provider_module	Module qui a demandé à exécuter cette commande
time	time	Heure à laquelle l'information a été enregistrée
time	time_reported	Timestamp qui désigne le moment où l'info à été enregistrer sur le serveur

Table des erreurs

Cette table répertorie toutes les erreurs d'Arkage rencontrés sur les appareils

Cette table s'appellera "**error**"

Type	Nom	Description
primary id	error_id	ID de la ligne
id	report_id	ID du rapport auquel il est rattaché
string	stack_trace	Stack trace de l'erreur
string	provider	Nom du module / système qui a rencontré l'erreur
string	description	Optionnel - Description fourni par le code si l'erreur est connue
time	time	Heure à laquelle l'information a été enregistrée
time	time_reported	Timestamp qui désigne le moment où l'info a été enregistré sur le serveur

Table des paramètres

Cette table à pour but d'enregistrer certains paramètres des API ou du reporting (stats globales etc...)

Cette table s'appellera "settings"

Type	Nom	Description
primary id	setting_id	ID de la ligne
string	key	Utilisateur auquel cette information est attachée
string	value_1	valeur associée à la clé
string	value_2	Optionnel - seconde valeur associée à la clé
string	value_3	Optionnel - troisième valeur associée à la clé
time	time_create	Heure à laquelle l'information a été créée
time	time_max_validity	Heure d'échéance de la validité de l'information
time	time_update	Heure de la dernière mise à jour de l'information
time	time_delete	Heure de suppression de l'information

Liste des settings

Ce tableau pourrait venir à être changé

Clé	Valeur	Description
api_password	N#Jz!Q&Fpw6^	Mot de passe pour prouver que c'est bien une application officiel d'Arkage qui est en train

		d'utiliser l'API
active_users	int	Nombre d'utilisateurs actif
unsubscribed_users	int	Nombre d'utilisateurs désinscrit
total_reports	int	Nombre total de rapports enregistré

Arkage UI

Arkage Website

Technologie & Framework

Le serveur web devra comporter plusieurs technologies, à savoir :

- Serveur PHP
- Serveur MySQL
- Framework Symfony
- Serveur Node Docusaurus

Documentation "Docusaurus"

La documentation du site sera générée au travers du framework Docusaurus.

Un sous-domaine sera dédié à ce site.

Docusaurus est un serveur NodeJS, il suffira de faire une redirection apache vers son port pour faire fonctionner le site de documentation

Docusaurus fonctionne grâce à du Markdown, le site sera facilement modifiable, même pour des personnes avec de faibles notions de développement.

Domaines, sous-domaines et hébergement

Initialement le site web devait être hébergé derrière le DNS "**arkage.com**", mais celui-ci semble pris par un site web inactif. Nous tenterons de récupérer ce DNS en contactant le propriétaire, mais en attendant nous utiliserons le DNS "**arkage.app**"

Le DNS "**arkage-app.com**" pourrait être amené à être acheté lui aussi si "arkage.app" fourni un mauvais référencement.

www.arkage.app

Ce sous-domaine est l'accès "grand public" destiné aux utilisateurs du site.

api.arkage.app

C'est depuis ce sous-domaines que les API d'Arkage sont accessible. Les API des rapports d'utilisations communiqueront avec ce sous-domaine

admin.arkage.app

Ce sous domaine permet d'utiliser certains outils d'administration.

Par exemple plutôt que PHPMyAdmin soit accessible via www.arkage.app/phpmyadmin, il sera uniquement accessible via admin.arkage.app/phpmyadmin.

Ceci permettra d'effectuer des attaques automatisées sur PHPMyAdmin par exemple

doc.arkage.app

Ce sous domaine est dédié pour la documentation Docusaurus. Dès qu'un utilisateur sera à la recherche d'une documentation il sera redirigé vers une URL pointant vers ce sous-domaine.

Plus de détails sur <https://docusaurus.io/>

Statistique et utilisation du site

Pour obtenir efficacement et simplement des statistiques sur l'utilisation de notre site, il sera nécessaire de faire surveiller notre site par Google Analytics

API rapport d'utilisation

Toutes les spécifications techniques des API sont disponible dans le "cahier des charges API reporting" à la suite de ce document.

Sécurité

Attribution des droits

Pour sécuriser au mieux le serveur web, une technique très simple sera de limiter en droits l'utilisateur qui exécutera les projets web. Il ne faut surtout pas faire exécuter les projets web avec l'utilisateur root, cela faciliterai des piratages.

Injection SQL

Pour éviter l'injection SQL, il sera nécessaire d'effectuer l'échappement des caractères spéciaux dans les formulaire.

Il sera interdit de faire de la concaténation de chaînes de caractère pour générer des lignes SQL, car cela nous rendrait vulnérable aux injections SQL

Tests unitaires

Les tests unitaires seront réalisé sur Android avec JUnit, une documentation de son utilisation sera fourni.

Document "Documentation JUnit.doc"

Environnement de développement & maintenance

Les 3 projets, à savoir Arkage (Kernel), Arkage UI et le site web Arkage seront stockés sur GitHub.

Arkage et Arkage UI seront pour la première version qu'un seul projet. Ce projet sera public pour permettre la collaboration.

Concernant le site web Arkage, il sera aussi sur GitHub mais ce dépôt sera privé.

Conclusion

Arkage est un gros projet. Nous avons définitivement besoin de nombreuses compétences pour mener à bien ce projet. Pas seulement des compétences de développement, mais aussi des compétences en marketing, communication et design.

Sans ces dernières Arkage risquerait d'être une technologie morte car incapable de se vendre.

Les possibilités d'Arkage sont immenses. Un très grand nombre de fonctionnalités pourront être développées grâce à ce moteur. Ce qui pourra grandement intéresser certaines entreprises

Le projet en lui-même pourrait lui aussi comporter beaucoup plus de fonctionnalités, mais nous avons préféré limiter nos idées dans ce cahier des charges pour avoir quelque chose de réalisable pour la V1.

Le futur d' Arkage

Arkage ne s'arrêtera pas fin juillet après la fin du cursus My Digital School, bien au contraire !

Une première version d'Arkage est prévue le 24 juillet dans le but de tester et recevoir des premiers retours mais, Arkage a pour objectif d'être poursuivi par un des étudiants du groupe au sein d'un Master, plus axé sur le développement et l'IA.

En effet, Arkage est dans un premier temps développé pour le grand public, une stratégie marketing et une étude UX/UI était donc nécessaire.

La V1 n'aura pas de source de revenus directs hormis le souhait de mettre en place différents partenariats avec des compagnies d'assurance par exemple. La réalisation de différents concours comme la GAE sur Angers est également une alternative de revenu.

Cette application sera ensuite développée à l'occasion d'une V2 dans le but de proposer un potentiel mode "premium" comportant des fonctionnalités plus poussées et plus variées.

Texte rédigé par Swann Lambard

Lexique & annexes