# SOW

1. Langage de description de driver : décrit le comportement d’un driver indépendamment de la plateforme hôte.
2. Compilateur du langage de description en code exploitable pour une plateforme de destination.
3. Une librairie de Templates extensible
4. Un outil de déploiement pour chaque plateforme supportée.
5. Une documentation couvrant tous les aspects sus cités
6. Le projet doit être maintenable et maintenu
7. Le projet doit être connu et reconnu
8. Le projet sera soumis à des licences open sources

# SSS

## Le langage de description

1. On doit concevoir un langage concis et exhaustif
2. Le langage devra avoir une sémantique forte et une syntaxe robuste
3. Capable de décrire l’intégralité des interactions entre les kernels et le matériel physique
4. Doit pouvoir embarquer des fragments de code C

## Le compilateur

1. Doit pouvoir fonctionner sur plusieurs plateformes hôte (Windows, MacOSX, BSD, Linux)
2. Doit générer du code C
3. Aura une librairie contenant des templates de générations par plateforme cible
4. Doit vérifier la syntaxe du langage
5. Doit vérifier la sémantique du langage
6. Doit pouvoir être exécuté en ligne de commande
7. Doit générer le driver grâce à un fichier de configuration
8. Utilise codeworker

## La Black Library

1. Contient des templates de code organisé par système d’exploitation
2. Doit être extensible
3. Doit vérifier la cohérence de ses informations

## Un outil de déploiement

1. Installer la solution complète (compilateur, documentation, codeworker)
2. Désinstaller la solution
3. Mettre à jour la solution

## Une documentation

1. Pour les développeurs de driver utilisant Rathaxes
2. Pour les contributeurs Rathaxes
3. Pour les contributeurs de la Black Library

## Le projet doit être maintenable et maintenu

1. La Black Library devra être extensible
2. Un outil de versionning devra être utilisé
3. Un outil de bug report devra être utilisé
4. Une communauté devra se former autour du projet
5. Un processus de validation des modifications devra être mis en place

## Le projet devra être connu et reconnu

1. Le projet devra être présenté dans des conventions
2. Avoir un site web descriptif
3. Devra être présenté à des entreprises

## Le projet sera soumis à des licences open sources

1. Une licence ISC pour la Black Library
2. Une licence LGPL pour tout le reste du projet Rathaxes

# SRS

## Le langage de description

### On doit concevoir un langage concis et exhaustif

1. Mots clés naturels pour un driver
2. Pas de double sens pour un mot
3. Syntaxe lisible
4. Pas deux mots clé pour une même sémantique

### Le langage devra avoir une sémantique forte et une syntaxe robuste

1. Le langage doit comporter des structures conditionnelles
2. Le langage doit comporter des mots clé par sémantique
3. Le langage doit être fortement typé
4. Le langage devra pouvoir être étendu

### Capable de décrire l’intégralité des interactions entre les kernels et le matériel physique

1. Description des RKM
2. Description des IO : méthode d’accès aux registres et description des registres
3. Description des différents call- backs : IRQ et DMA
4. Description des BUS : demande de datas
5. Description des différents algorithmes : automates
6. Descriptions des interfaces entre userland et kernelland

### Doit pouvoir embarquer des fragments de code C

1. Le code C sera entouré de balises
2. La balise devra renseigner à quelle sémantique est rattachée ce code

## Le compilateur

### Doit pouvoir fonctionner sur plusieurs plateformes hôte (Windows, MacOSX, BSD, Linux)

1. Doit comporter des scripts ou binaire d’exécutions par système d’exploitation

### Doit générer du code C

1. Le code générer doit respecter la philosophie du système d’exploitation sur lequel il sera générer
2. Le code C générer doit être compilable
3. Le code C ne doit pas générer de warning
4. Le code C générer doit avoir des noms de variables un minimum explicite.
5. Le code C générer doit être lisible, pour une probable relecture
6. Le code C doit comporter des commentaires expliquant la sémantique
7. Le code C générer doit inclure les bons headers selon le système d’exploitation cible

### Doit vérifier la syntaxe du langage

1. Afficher une erreur si la syntaxe du fichier d’entrée est fausse
2. Le message d’erreur doit être clair et comporter le numéro de la ligne

### Doit vérifier la sémantique du langage

1. Indiquer les non sens dans le fichier d’entrée
2. Le message devra être clair

### Doit pouvoir être exécuté en ligne de commande

1. L’outil en ligne de commande doit comporter une aide pour les différentes options de la ligne de commande
2. L’outil en ligne de commande doit être simple
3. Il doit gérer les erreurs

### Doit générer le driver grâce à un fichier de configuration

1. Est écrit en langage Rathaxes avec des mots clés
2. Qui comportera le ou les systèmes d’exploitation visés
3. Permettra de choisir le mode de génération (debug, release)

### Utilise codeworker

1. Codeworker devra être fournit avec le compilateur

## La Black Library

### Contient des templates de code organisés par système d’exploitation

1. Les templates seront regroupées par OS et par sémantique

### Doit être extensible

1. Les templates seront indexés dans un système que nous appellerons Necronomicon
2. L’ajout de nouveaux templates met à jour le Necronomicon
3. Si un nouvel OS est ajouté, toute la sémantique nécessaire gérée par Rathaxes devra être implémentée par les templates de ce nouvel OS.

### Doit vérifier son contenu

1. La validité de la Black Library sera vérifiée grâce à un programme le Black Guardian
2. Si le contenu n’est pas valide pour un OS, aucun driver ne pourra être généré pour cet OS

## Un outil de déploiement

### Installer la solution complète (compilateur, documentation, codeworker)

1. Installer Codeworker et des alias pour y accéder
2. Copier l’exécutable Rathaxes et la Black Library
3. Copier la Documentation

### Doit avoir une dépendance à codeworker

1. Si codeworker n’est pas correctement installer un message clair doit le spécifier

### Doit contenir la documentation

1. La documentation sera au format PDF

### Suivre les différentes normes d’installation des OS

1. Utiliser les différent systèmes de packaging selon l’OS (installeur, package ,…)

## Une documentation

### Pour les développeurs de driver utilisant Rathaxes

1. Elle devra préciser comment installer Rathaxes
2. Comment lancer Rathaxes
3. Définir toute la syntaxe du langage
4. Proposer des exemples
5. Définir la sémantique

### Pour les contributeurs Rathaxes

1. Expliquer comment contribuer au projet
2. Quels sont les limites de la contribution
3. Devra définir le fonctionnement de Rathaxes

### Pour les contributeurs de la Black Library

1. Devra définir les contraintes et les règles d’ajout
2. Expliquer le fonctionnement de l’arbre (Necronomicon)
3. Comment contribuer à la Black Library

## Le projet doit être maintenable et maintenu

### La Black Library devra être extensible

1. Par l’utilisateur pour son usage personnel
2. Par Rathaxes à chaque mise à jour du logiciel

### Un outil de versionning devra être utilisé

1. Pour fixer des versions et suivre l’évolution du projet
2. Pour faciliter la correction de bug

### Un outil de bug report devra être utilisé

1. L’outil devra être accessible depuis le web
2. Les utilisateurs devront s’enregistrer pour soumettre des bugs
3. Les bugs devront être assignés aux équipes concernées

### Une communauté devra se former autour du projet

1. Le projet pourra recevoir des contributions extérieures
2. L’équipe Rathaxes animera des conférences pour promouvoir le projet

### Un processus de validation des modifications devra être mis en place

1. Un outil de proposition de modification devra être mis en place
2. Chaque proposition devra être étudiée

## Le projet devra être connu et reconnu

### Le projet devra être présenté dans des conventions

1. Inscription à différentes conventions

### Avoir un site web descriptif

1. Le site web devra être référencé par rapport au champ lexical du driver

### Devra être présenté à des entreprises

1. A travers des conférences

## Le projet sera soumis à des licences open sources

### Une licence ISC pour la Black Library

1. Chaque entête de Template devra comporter un header avec la licence ISC

### Une licence LGPL pour tout le reste du projet Rathaxes

1. Chaque fichier de code Rathaxes devra comporter un header avec la licence LGPL