|  |  |
| --- | --- |
| Fonctionnalités Outils Plugin | |
| FOP 1 | Interpréter les plugins |
| FOP 1.1 | Compatibilité plusieurs langages |
| FOP 1.11 | C# |
| FOP 1.12 | C |
| FOP 1.13 | Ruby |
| FOP 1.14 | Python |
| FOP 1.15 | Go |
| COP 1 | Compatibilité multi-plateforme |

|  |  |
| --- | --- |
| FOP 2 | Configurer Client/Plugins |
| FOP 2.1 | Interpréter le fichier de Config |
| FOP 2.11 | Configurer le heartbeat (la fréquence d’activation) des plugins |
| FOP 2.12 | Configurer la liste des OS disponibles pour un plugin |
| FOP 2.13 | Rendre inactif/actif un plugin |
| FOP 2.14 | Configurer les interpréteurs pour chaque OS |

|  |  |
| --- | --- |
| FOP 3 | Mettre à jour des plugins |
| FOP 3.1 | Mettre à jour des plugins côté Serveur |
| FOP 3.2 | Mettre à jour automatiquement des plugins côté Client |

|  |  |
| --- | --- |
| Fonctionnalités Gestion des données | |
| FGD 1 | Interpréter des données |
| FGD 1.1 | Traiter les données |
| FGD 1.11 | Graphe ? |
| FGD 1.12 | Traitement interne ? |

|  |  |
| --- | --- |
| FGD 2 | Stocker des données |
| FGD 2.1 | Stocker Localement |
| FGD 2.2 | Stocker à distance |
| FGD 2.21 | Stocker dans le Cloud |
| FGD 2.22 | Stocker dans un serveur distant |

Le tableau de fonctionnalité est divisé en deux grandes parties. Une première partie qui compose les fonctionnalités des outils liées aux plugins, abrégé **FOP** pour **Fonctionnalité Outils Plugin**; et une seconde partie qui compose les fonctionnalités liées à la gestion des données, abrégé **FGD** pour **Fonctionnalité Gestion des Données**.

Concernant les contraintes, elles seront abrégées **COP** pour **Contrainte Outils Plugin**, et **CGD** pour **Contrainte Gestion des Données**.

**Fonctionnalités Outils Plugin :**

Le tableau de fonctionnalité Outils Plugin est divisé en trois grandes fonctionnalités.

* Interprétation des plugins

L’outil de gestion des plugins de HAL devra être capable d’interpréter et d’exécuter les plugins écrit dans différent langages : C#, C, Ruby, Python, Go. L’exécution des plugins devra être supporté sous différentes plateformes : Windows, Linux, MacOS.

* Configuration du/des Clients et des Plugins.

L’outil doit permettre la configuration de chaque client et de chaque plugin. Chaque plugin possède des caractéristiques différentes, ces caractéristiques doivent pouvoir être configuré par l’utilisateur. Notamment la configuration du **Heartbeat,** qui est la fréquence d’activation d’un plugin. La liste des OS compatible au plugin. La possibilité de rendre le plugin inactif via la configuration. La configuration des interpréteurs de langage pour chaque OS.

La configuration doit pouvoir être fait facilement par l’utilisateur et l’outil doit être capable d’interpréter le fichier de configuration.

* Mise à jour des plugins

L’outil doit pouvoir être capable de mettre à jour les plugins côté Serveur et de manière automatique côté Client.

**Fonctionnalités Gestion des Données :**

Le tableau de fonctionnalité Gestion des données est divisé en deux grandes fonctionnalités.

* Interpréter les données

L’interprétation des données est la faculté à traiter les données récupérées afin de les traiter.

* Stocker les données

L’outil développé devra implémenter une solution permettant du stockage à long termes des données récupérer, et ceux, de manière totalement abstraite. C’est-à-dire que l’outil devra fournir une solution capable de s’adapter à la plateforme visée : localement, à distance, Cloud, Serveur,