Projet RADOME

|  |  |
| --- | --- |
| **Référence :** |  |
| **Version :1.0** |  |

|  |  |  |
| --- | --- | --- |
| **rédacteur** | **Verificateur** | **Approbateur** |
| RMINECHE Aziz  Ingénieur Logiciel  Visa : | Nom Prénom  (fonction)  Visa : | Nom Prénom  (Fonction)  Visa : |

Maitrise des évolutions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date**  **De VERSION** | **Version** | **Objet de la version** | **rédacteur** |
| 18/09/14 | 1.0 | Création | Aziz RMINECHE |
|  |  |  |  |
|  |  |  |  |

Sommaire

[1. Présentation générale et objectifs 6](#_Toc368402027)

[*1.1* *Identification* 6](#_Toc368402028)

[*1.2* *Généralités sur le logiciel* 6](#_Toc368402029)

[*1.3* *Vue d'ensemble du document* 6](#_Toc368402030)

[2. Documents de références et applicables 7](#_Toc368402031)

[*2.1* *Documents projet de référence* 7](#_Toc368402032)

[*2.2* *Documents projet applicables* 7](#_Toc368402033)

[3. Décisions de conception du logiciel 8](#_Toc368402034)

[*3.1* *Entrées / Sorties* 8](#_Toc368402035)

[*3.2* *Comportement* 8](#_Toc368402036)

[*3.3* *Bases de données* 8](#_Toc368402037)

[*3.4* *Aspects sécuritaires* 8](#_Toc368402038)

[*3.5* *Contraintes physiques* 9](#_Toc368402039)

[*3.6* *Exigences* 9](#_Toc368402040)

[4. Architecture 10](#_Toc368402041)

[*4.1* *Les composants du logiciel* 10](#_Toc368402042)

[*4.1.1* *Identification* 10](#_Toc368402043)

[*4.1.2* *Aspects statiques* 10](#_Toc368402044)

[*4.1.3* *Description* 11](#_Toc368402045)

[*4.1.4* *Développement* 11](#_Toc368402046)

[*4.1.5* *Ressources* 11](#_Toc368402047)

[*4.1.6* *Diagramme d’Architecture du logiciel* 11](#_Toc368402048)

[*4.2* *Concept d'exécution / Aspect Dynamique* 12](#_Toc368402049)

[*4.3* *Conception de l'interface* 12](#_Toc368402050)

[*4.3.1* *Identification de l'interface et diagrammes* 12](#_Toc368402051)

[*4.3.2* *Identifiant unique projet de l'interface* 12](#_Toc368402052)

[*4.3.2.1* *Priorité* 13](#_Toc368402053)

[*4.3.2.2* *Type* 13](#_Toc368402054)

[*4.3.2.3* *Caractéristiques données simples* 13](#_Toc368402055)

[*4.3.2.4* *Caractéristiques données complexes* 14](#_Toc368402056)

[*4.3.2.5* *Communication* 14](#_Toc368402057)

[*4.3.2.6* *Protocole* 15](#_Toc368402058)

[*4.3.2.7* *Autres caractéristiques* 15](#_Toc368402059)

[5. Traçabilité 16](#_Toc368402060)

[*5.1* *Traçabilité ascendante* 16](#_Toc368402061)

[*5.2* *Traçabilité descendante* 16](#_Toc368402062)

Abréviations et acronymes

| Abréviations | Définition |
| --- | --- |
| X | X |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Présentation générale et objectifs

Le présent document constitue la Spécification de l’architecture du logiciel applicable au projet RADOME.

Le but de ce document est de définir les différents choix d’architecture logicielle avec les difficultés rencontrées pour chaque solution ainsi que les résultats obtenus.

# *Généralités sur le logiciel*

Aujourd’hui, on trouve des interfaces graphiques embarqués dans différents secteurs d’activité (Automobile, Aéronautique, domotique, ferroviaire, Ligne de Production, …).

Le but de ce projet est d’optimiser l’utilisation de ces interfaces en introduisant des nouvelles fonctionnalités.

Le projet RADOME aura pour objectif de réaliser un logiciel qui permet de:

* se connecter à distance sur l’interface graphique embarquée,
* pouvoir commander le système à distance,
* effectuer des diagnostics de pannes du système,
* introduction d’une commande vocale,

# Documents de références et applicables

# *Documents projet de référence*

Ce paragraphe décrit les documents de référence.

|  |  |
| --- | --- |
| Titre | Références |
| ISO26262-6:2011 | *Véhicules routiers — Sécurité fonctionnelle — Partie 6: Développement du produit au niveau du logiciel* |
| X | X |

# *Documents projet applicables*

Ce paragraphe décrit les documents du client et de MATIS Technologies applicables.

*Documents applicables du Client :*

|  |  |
| --- | --- |
| Titre | Références |
| Cahier des charges | X |
| X | X |

*Documents applicables de MATIS TECHNOLOGIES:*

|  |  |
| --- | --- |
| Titre | Références |
| Manuel d’Assurance Qualité | X |
| Proposition Technique et Commerciale | X |
| Matrice des exigences | X |
| Plan de gestion de configuration | X |
| Plan de développement | X |
| Plan d’Assurance Qualité | X |
|  |  |

# Architecture 1 : Serveur Multi threads Client mono connexion :

*Avant de détailler les solutions choisies, je vais présenter rapidement le besoin technique du projet. Notre besoin consiste à établir une connexion à distance entre le serveur embarqué dans la voiture et des clients qui vont se connecter à distance, chaque client peut posséder soit une ou plusieurs connexions web sockets.*

*La première solution choisit pour notre logiciel est une architecture avec un serveur multitâches et un client qui tourne sur une seule connexion WebSocket.*

SERVEUR

CLIENT

***Le serveur :***

SERVEUR

THREAD 2

THREAD 3

THREAD 1

THREAD principal

*Le serveur est constitué de plusieurs threads :*

* *un thread principal qui se met en écoute permanente sur la web socket du client afin de détecter un éventuel évènement du client.*
* *3 threads secondaires qui seront chargés d’envoyer les données en permanence sur chaque appli du client (un thread par application).*

***Le client :***

**CLIENT**

1 Connexion Web Socket

APPLI 1

APPLI 2

APPLI 3

APPLI n

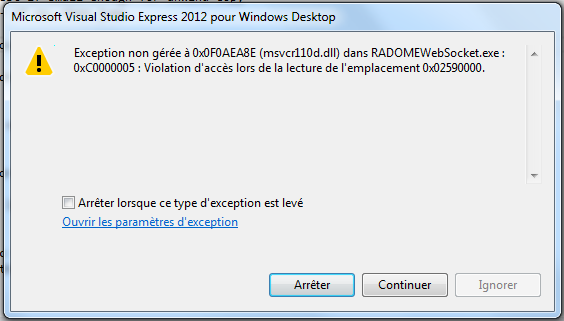
La partie « CLIENT » est architecturé de la manière suivante :

* Chaque client possède une connexion WebSocket, qui se met à l’écoute des informations reçues du serveur,
* Chaque connexion WebSocket gère plusieurs applications (CAN, AUDIO, VIDEO, MAP,….),

**Difficultés rencontrées :**

Les principales difficultés rencontrées sur cette solution est la mise en place de la technologie multithread coté serveur, en effet si plus de trois threads sont lancés simultanément pendant une durée supérieure à trois minutes, le serveur renvois une erreur d’exception et stop l’envoie des données. L’origine de cet erreur n’est pas encore définit à 100% mais des hypothèses sont émises :

* Le serveur ne gère pas plus de trois threads en même temps,
* L’accès à une partie de la mémoire partagée par plusieurs threads peut provoquer un conflit entre les différents threads,



# 4. Architecture 2 : Serveur Mutlitâches client multi connexions

La deuxième architecture envisagée et qui reste à valider est une solution basée sur un serveur plusieurs connexions WebSocket dans un même client.

Ce mode de fonctionnement reste à valider afin de définir l’architecture côté serveur.

***Serveur :***

THREAD n

SERVEUR

THREAD 3

THREAD 2

THREAD 1

THREAD principal

# 

Le serveur devra gérer 2 threads un thread principal qui ouvre les connexions Web Sockets et un thread secondaire qui aura pour tâche le traitement des données envoyées au client.

**Client**:

**CLIENT**

**Web socket**

**Web socket 1**

**Web socket 2**

**Web socket n**

**Web socket 3**

**APPLI 1**

**APPLI 2**

**APPLI 3**

**APPLI n**

Le client devra gérer plusieurs connexions web sockets :

* Une web socket principale qui permettra d’établir la connexion entre le serveur et le client, et de rediriger l’information reçue du serveur vers la socket du traitement en fonction de l’entête de la trame,
* Pour chaque application, une connexion Web Socket sera ouverte pour effectuer le traitement des données coté client,

**Résultats attendus :**

Cette solution n’a pas été testée, les résultats obtenus seront développés par la suite.