



MASTER 2 – ISIDIS


Projet de synthèse

V.A.L.

Vision Document

Table des matières

1. Objet du document	4
2. Objet du projet	4
3. Objectifs et enjeux	4
4. Positionnement	5
4.1 Positionnement du problème	5
4.2 Positionnement commercial	5
5. DETERMINER LES UTILISATEURS ET DECIDEURS	5
5.1 UTILISATEURS:	5
5.2 DECIDEURS:	6
6. LIENS ENTRE APPLICATIONS	6
7. FONCTIONNALITES DU PROJET	7
8. EXIGENCES ET CONTRAINTES SUPPLEMENTAIRES	7
8.1 EXIGENCES <<FURPS>>	7
8.1.1 Functionality:	7
8.1.2 Usability:	7
8.1.3 Reliability:	7
8.1.4 Performance:	8
8.1.5 Supportability:	8
8.2 Autres contraintes	8
8.2.1 Documentation	8
8.2.2 Organisation	8
8.2.3 Temps	8

	ESIAG 2012_2013 Vision Document Version 0.0.2
--	--

Documents de référence :

- Document sourceforge sur le vision document : [Lien](#)
- Sujet du projet de synthèse
- Cours ESIAG

1. Objet du document

Ce document a pour objectif d'identifier les décideurs et les utilisateurs affectés par le problème impliquant ce projet et la définition du périmètre du projet à réaliser afin d'en cadrer les besoins métier. Il a également pour vocation de présenter une vue globale (haut niveau) de la solution à réaliser. Enfin, ce document contient un aperçu des exigences et contraintes recensées ce qui constitue une base contractuelle.

2. Objet du projet (fonction du projet)

Le projet de synthèse sera mené par une équipe de travail de dix étudiants Miagistes et aura comme but principal la réalisation d'une solution performante, respectant les caractéristiques des systèmes distribués, et fiable.

Le projet a pour objectif la réalisation d'une application destinée à une entreprise de transports en commun exploitant un réseau ferré et des équipements automatiques assurant le transport de passagers sur un secteur géographique à définir, ainsi que l'étendue de son infrastructure (quais, couloirs, tunnels, rames, appareils de détection .. etc)

La solution logicielle doit être sûre et ce en l'utilisant via un environnement de simulation dynamique, tout en étant ouverte et évolutive pour être implantée dans toute entreprise du secteur.

3. Objectif et enjeux

Le but du projet est de fournir un SI permettant l'automatisation du réseau client, et permettre son pilotage en temps réel.

Le pilotage automatique implique la présence d'un centre de contrôle faisant la liaison entre les interfaces de communication et le réseau terrain afin de superviser toute l'activité du réseau, et de ce fait pouvoir interagir avec ce dernier en temps réel.

Les moyens de communication doivent donc être sûrs, et donc très hautement disponibles (contrôle de charge)

4. Positionnement

4.1 Positionnement du problème

Problème	Le SI actuel ne permet pas la gestion automatique en temps réel du réseau
Affecté	L'exploitation du réseau, les conditions de

	supervision
Conséquence	sur le trafic, et les performances
Solution	superviser la totalité du réseau, réactivité en cas d'incident

4.2 Positionnement commercial

Pour	sociétés de transport rencontrant des problèmes d'exploitation de leur réseau de transport
Qui (besoin)	supervise l'ensemble de ses activités, en capturant/priorisant/transmettant les informations terrain en cas d'incident afin de réguler le trafic.
RTDG (fatal-team)	interconnexion entre plusieurs composants dédiés à des tâches précises, adaptés aux contraintes clientes pour résoudre ses problèmes
Assurant	la haute disponibilité de l'interconnexion entre le centre de contrôle et les équipements terrain, la remontée d'information en temps réel selon un ordre de priorité défini.
Valeur ajoutée	par rapport à l'existant, notre solution permet une supervision globale du système, d'avoir une remontée des alertes terrain ET l'envoi d'ordre en cas d'incident grave
Le produit	répond aux problèmes récurrents liés à l'exploitation des réseaux de transport ferrés, tout en s'adaptant au réseau d'un client particulier

5. DETERMINER LES UTILISATEURS ET DECIDEURS

5.1 Utilisateurs

Voici les différents acteurs jouant un rôle dans l'élaboration de notre solution logicielle :

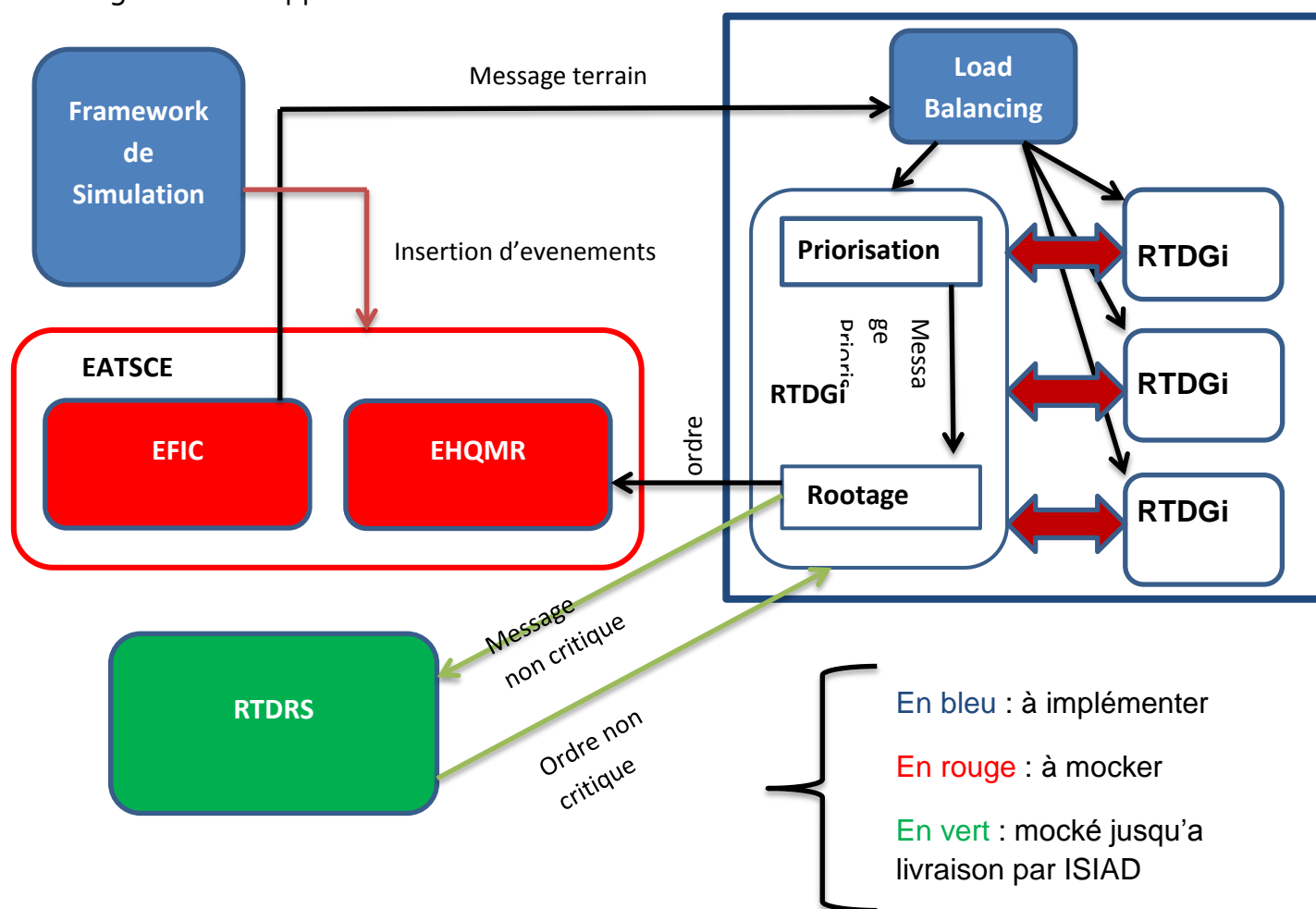
- Administrateur RTDG (Superviseur) : il se charge de démarrer et d'arrêter le composant RTDG
- Responsable RTDRS : utilise le RTDG pour recevoir l'interprétation des messages terrain, et pour envoyer les ordres en retour.

5.2 Décideurs

- Equipe pédagogique : (Alexandre Brenner et Gilles Giraud)
 - Mode d'implication: Comité de pilotage.
 - Critère de succès : Robustesse, performance, transparence, simplicité, clarté, et respect des normes.
- Product Owner : son rôle est de représenter auprès de l'équipe de développement le client/utilisateur pour que le produit final soit conforme aux attentes de ce dernier auprès de l'équipe de développement, revue de sprint et backlog, et accepter ou refuser des un travail réalisé de par sa connaissance du domaine métier, et ce au moyen de réunions ponctuelles en fonction de sa disponibilité
- Responsable RTDRS : doit nous indiquer la manière dont on communique avec le système, et se mettre d'accord sur les informations qui doivent être échangés entre le RTDRS et notre RTDG.

6. Liens entre applications

Les différents composants utilisés dans notre solutions communiquant sont interconnectés (via l'envoi de messages), le schéma ci dessous présente une vue globale de l'application :



7. FONCTIONNALITES DU PROJET

Nom	Description	Priorité
Configurer le démarrage des services	Démarrage du SI, cette tâche est réalisée par l'administrateur réseau, elle comprend le chargement	Haute

	total des informations de la base de données en mémoire qui serviront de référentiels, Ex : adresse MAC de chaque composant, contre mesures pour les incidents	
Prioriser messages	Associer (selon un référentiel défini) aux messages reçus par les composants terrain un ordre de priorité croissant (du plus critique au moins critique) selon la nature de la source du message (capteur)	Haute
Orchestrer les messages	Gestion des messages en provenance du centre de contrôle vers le RTDG	Haute
Effectuer les interventions automatiques	Lancer les contre mesures pour les incidents survenus en se basant sur un référentiels prévu à cet effet	Haute
Collecter les informations terrains	Réception des messages contenant les informations capturées par les équipements terrain	Haute
Expand messages	Traduire une contre mesure (message fonctionnel) en une série de messages techniques à envoyer aux composants	Haute
Transmettre messages au terrain	Transmission d'un message vers un composant identifié par son adresse MAC.	Haute

8. EXIGEANCES ET CONTRAINTES SUPPLEMENTAIRES

8.1 Exigences FURPS

8.1.1 Functionality

- la synchronisation des données doit être faite de manière à ce que la perte de données soit la plus minimale possible.

8.1.2 Usability

- Le comportement de l'application change dynamiquement selon le scenario qu'on lui fournit en entrée dans le Framework de simulation développé.
- La modification à chaud des valeurs de configuration

8.1.3 Reliability

- Notre application se doit d'être tolérante aux pannes c'est à dire qu'elle doit continuer a fonctionner normalement (ou du moins légèrement dégradée) lorsqu'un de ses composants ne fonctionne pas correctement, ceci est assuré notamment en faisant de la réplication/duplication des composants fortement sollicités
- Développement d'un environnement de simulation complet, modulaire et dynamique supportant l'insertion d'événements de simulation « à chaud » pour voir le comportement de l'application et ce afin de garantir aux clients la validité de la solution logicielle
- Il faut que tous les composants (embarqués, centre de controle, stations, .. etc) soient synchronisés sur le même temps logiciel (référence commune)

8.1.4 Performance

- Le critère performance est un atout majeur de notre application, la remontée des informations ainsi que la transmission des messages d'ordre doit être fait en temps quasi réel, c'est à dire dès leur production dans les composants embarqués, le temps doit donc être *borné* et cette borne reste à définir, notre application se basant sur la librairie gratuite pour java : *Javolution* cette borne dépendra des performances de cette librairie
- Elle doit aussi assurer une disponibilité élevée ce qui implique la mise en œuvre d'algorithmes de répartition de charge pour distribuer le travail entre les répliques du composant RTDG qu'on crée
- Le système doit pouvoir supporter une certaine volumétrie (qui peut être importante) de données, informations capturées sur le terrain, messages échangés entre les différents composants, messages d'ordres .. etc

8.1.5 Supportability

- L'application doit être adaptable à tout réseau client de *même périmètre* applicatif que notre application, la taille du reseau, le nombre de stations, ... est fourni par le groupe ISIAD et est instancié par notre RTDG
- Le déploiement doit être automatisé autant que possible, ou du moins avec le moins de paramétrage manuels possible

8.2 Autre contraintes

8.2.1 Documentation

On va définir une documentation afin d'aider l'utilisateur à bien utiliser et gérer le produit.

8.2.2 Organisation

Le projet s'inscrit dans le cadre d'un projet universitaire, l'intégration entre les membres du groupe est préliminaire par rapport au sujet, il faut prendre en considération la disponibilité et le lieu du travail de l'équipe pour la distribution des tâches qui seront réalisées dans ce cas à distance

L'impact contraintes liés à l'organisation de travail est en quelque sorte diminué grâce à la mise en place d'outils et logiciels facilitant la communication et le travail à distance (gestionnaire de version, gestionnaire de conduite de projet, etc.)

En outre une partie de l'application doit être réalisée par une équipe d'une autre formation de l'ESIAG, équipe ISIAD doit nous fournir le composant RTDRS qui nous fournit notamment le horaires des trains..etc

8.2.3 Temps

Le temps de la réalisation de chaque tâche est important, il faut alors respecter les délais estimés pour chaque tâche afin de les tester et d'optimiser le temps de la réalisation car c'est un critère fondamentale à respecter pour l'état d'avancement du projet.