



GT Protocole « eRSMP ».

# Sujet des ‘acquittements’

MLD, 28/3/2023.

màj 11/6/2023

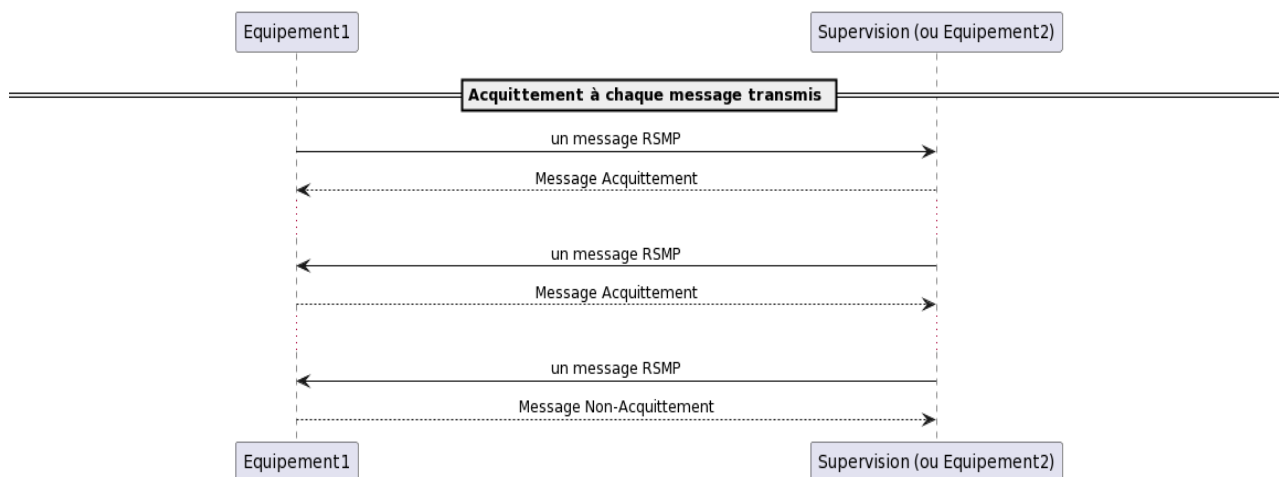
## 1 Fonctionnement actuel des acquittements sur RSMP « Core »

Le protocole RSMP prévoit que chaque message reçu d'un tiers génère un message d'acquittement indépendamment de la portée sémantique du message reçu (RSMP 3.2 §4.4.6 Message Acknowledgement).

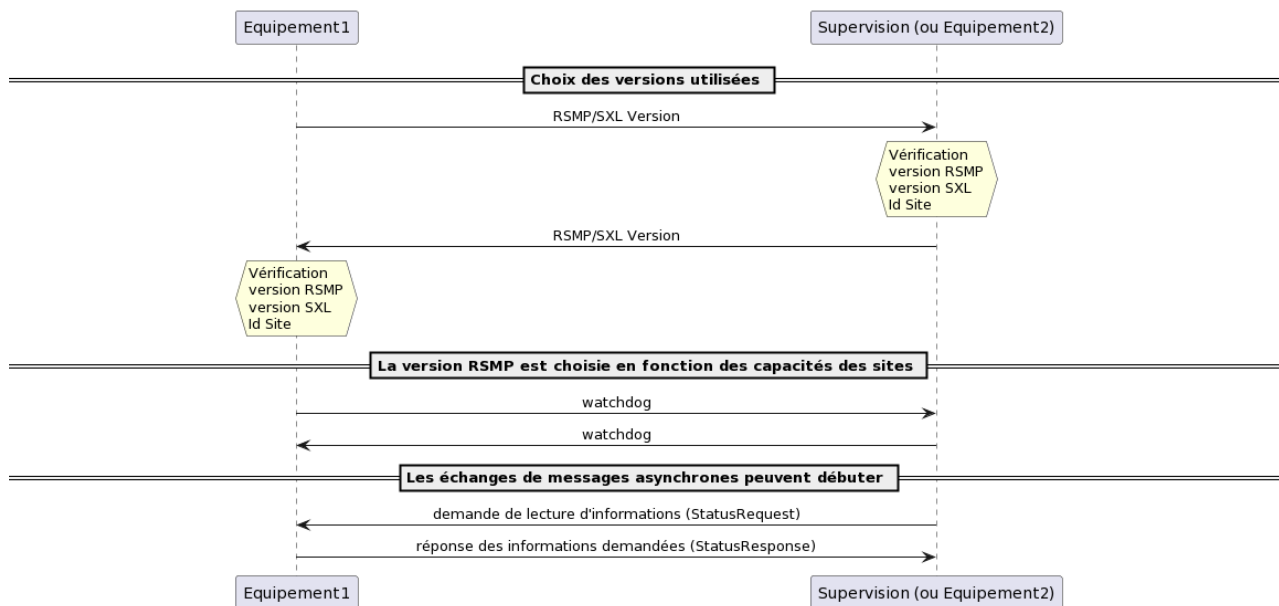
L'objectif de ce message est de vérifier la bonne distribution du message, et de confirmer la bonne compréhension (ack) ou la non compréhension (nack) du message reçu par le destinataire. Il permet de détecter immédiatement les ruptures de connexion (absence de réception de ack ou nack après un timeout, et consécutivement).

La syntaxe des messages d'acquittement ne reprend que l'identifiant (GUID) du message reçu avec dans le cas d'un acquittement négatif une raison de ce retour négatif. Dans le message émis, l'identifiant est présent dans l'attribut « **mId** », et dans le message d'acquittement en retour, l'identifiant de référence initial se retrouve dans l'attribut « **oMid** ».

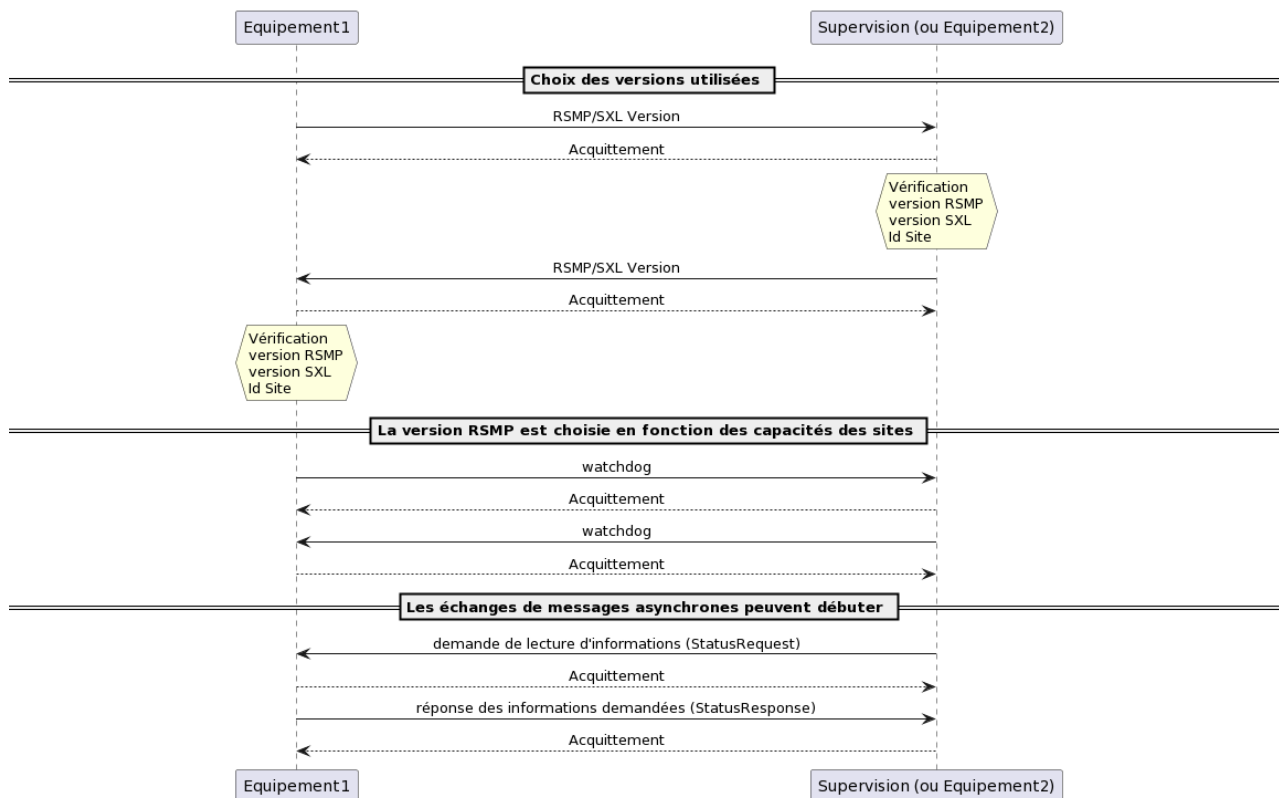
*[NdA] Aucune précision n'est donnée dans les spécifications sur la notion de compréhension : Est-ce une analyse de structure du message (format cohérent, pas de perte d'information) ou bien est-ce aussi une validation sémantique du message porté. Visiblement avec les outils simulateurs, on peut remarquer qu'une lecture de status avec une valeur non renseignée (null) génère un nack...*



Le schéma générique d'un déroulement de communication suivant :



devient en ajoutant les acquittements :



## 2 Constat

On peut classer les types de messages de RSMP en deux catégories :

- les messages qui n'amènent pas de réponse, et se limitent à un acquittement en retour,
- les messages suivis d'une réponse en retour, qui se voient donc doubler par des acquittements pour chacun.

Dans la première catégorie, on retrouve notamment les messages suivants :

- « Version »
- « Watchdog » (signe de vie périodique)
- « StatusSubscribe »
- « StatusUnsubscribe »
- « StatusUpdate »

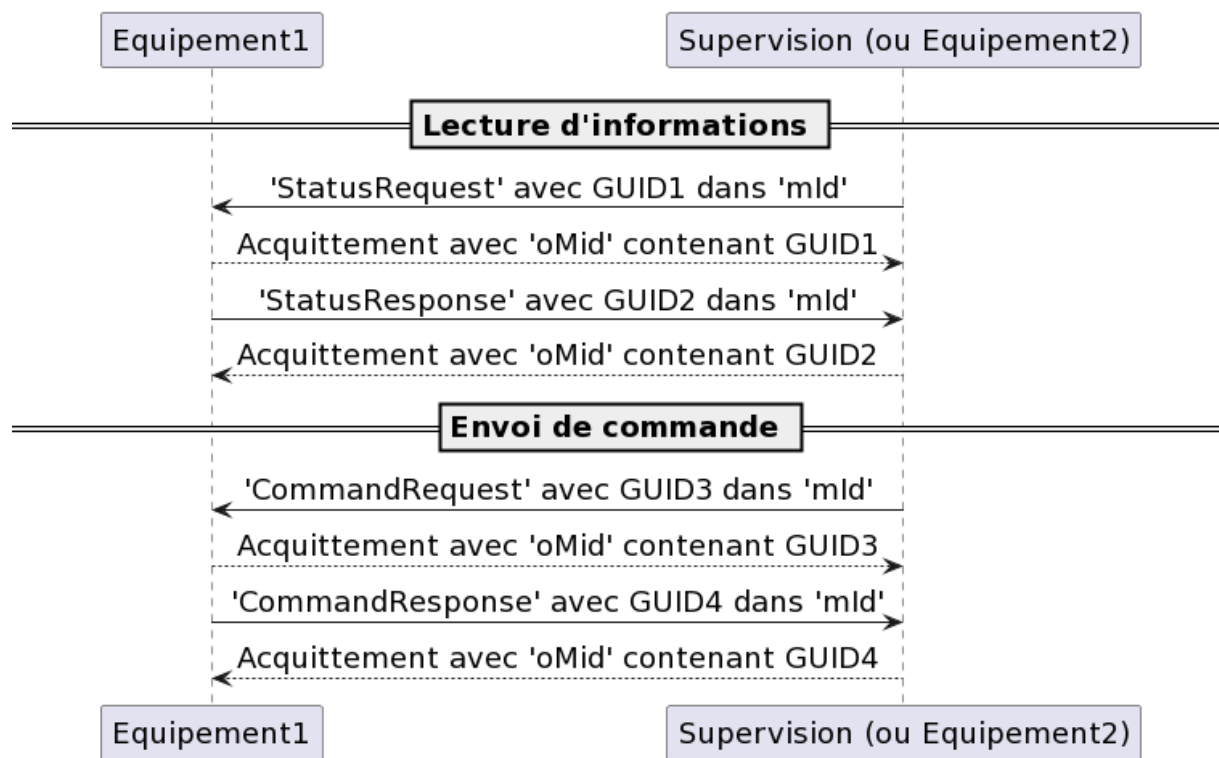
Pour cette première catégorie (messages sans réponse), il paraît judicieux de conserver les retours d'acquittements. D'autant plus que ce types de messages est principalement utilisés événementiellement ou peu fréquemment (cas du message « Version » au départ, période signe de vie à 30 secondes par-défaut).

Dans la deuxième catégorie, on retrouve les messages suivants :

- « StatusRequest » → « StatusResponse » (lecture d'informations)
- « CommandRequest » → « CommandResponse » (envoi de commandes)

Pour cette catégorie de messages de type séquentiel question / réponse entre deux équipements, on peut considérer que les messages d'acquittement sont surnuméraires.

Schéma 'actuel' du déroulement pour « StatusRequest » et « CommandRequest » avec noms des attributs échangés pour le GUID dans les requêtes :



### 3 Intérêt d'apporter une optimisation

Il ne faut pas perdre de vue que le premier cas d'usage de RSMP qui va en être fait consistera à l'utiliser en tant que passerelle des protocoles actuels LCR/DIASER. Hors ces deux protocoles fonctionnent uniquement en mode question / réponse en polling (pas de notion de mécanisme d'abonnement). Donc laisser tel que le principe des acquittements actuels sachant qu'on utilisera le message « CommandRequest » pour véhiculer les trames LCR/DIASER conduirait de base à doubler toutes les trames échangées par-rapport à l'existant...

Également, à l'avenir, lorsque les PCs migreront des protocoles DIASER/LCR vers le nouveau protocole, il est possible qu'ils continuent toujours à fonctionner en polling classique (au moins dans un premier temps) sans utiliser les nouvelles possibilités d'abonnements. Donc l'optimisation sur les acquittements dans ce cas d'échange restera tout-à-fait utile.

### 4 Evolution envisagée.

On peut très bien optimiser les types de messages suivis de réponse en intégrant directement l'attribut « mId » reçu dans la réponse « fonctionnelle » (on n'aurait donc plus qu'un seul GUID pour une même transaction, au lieu de 2 distincts actuellement).

Dans la réponse, on aurait donc directement l'attribut « oMid » reprenant le GUID de la question.

Initialement, la piste envisagée était de rajouter un choix d'activation « globale » de mode des acquittements lors de l'établissement de la connexion (dans le message existant « Version » de

départ, afin que les deux côtés se mettent d'accord sur la gestion des acquittements à avoir pour les trames impactées « StatusRequest » et « CommandRequest »).

Au lieu de devoir passer une option au niveau de la trame « Version » existante, une autre piste préférable consiste à ajouter de nouveaux types de messages dédiés.

*On peut simplement ajouter les 4 nouveaux types de messages suivants (tout en conservant les types de messages actuels) :*

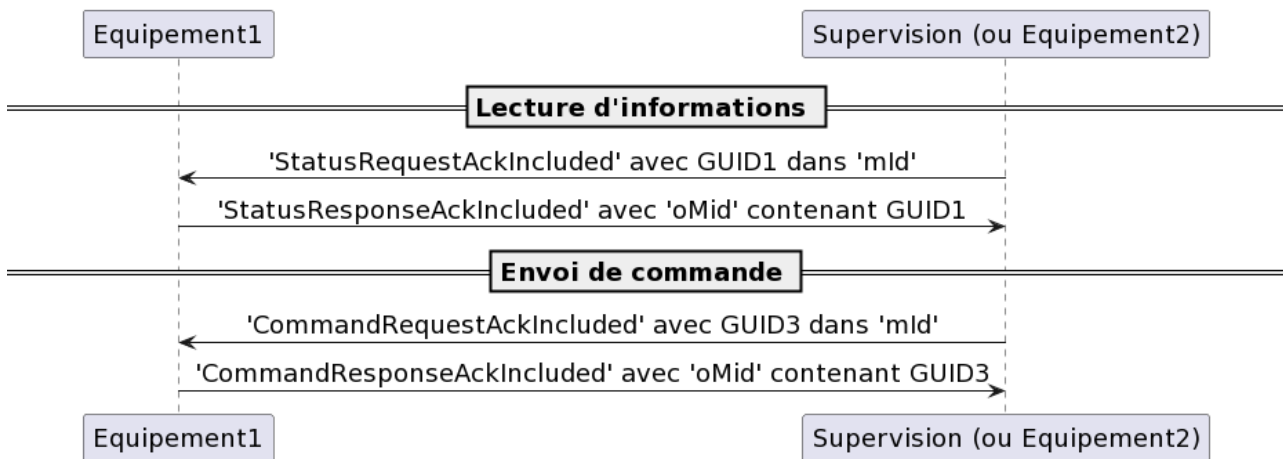
« **StatusRequestAckIncluded** » & « **StatusResponseAckIncluded** »

« **CommandRequestAckIncluded** » & « **CommandResponseAckIncluded** »

Ce principe permet à un système de supervision/équipement de pouvoir, au choix indifféremment et à tout moment, d'utiliser soit les requêtes Status/Command d'origines, ou soit les nouvelles optimisées avec acquittement intégré dans la réponse. Et on peut imaginer qu'en terme d'évolution vis-à-vis de RSMP « Core » ce principe soit nettement préférable à faire accepter, car plus à considérer comme une extension additionnelle, sans modification du comportement de l'existant qui reste conservé...

(ça évite également d'avoir une variation de nom d'attribut pour un même type de message suivant le mode utilisé...)

Schéma 'futur' de déroulement de l'évolution illustrant les 4 nouveaux types de messages avec noms des attributs échangés pour le GUID dans les requêtes :



La supervision ne doit pas renvoyer d'acquiescement final suite à la réception du message « **StatusResponseAckIncluded** » ou « **CommandResponseAckIncluded** ». On considère ici qu'il n'est pas pénalisant que l'équipement n'ait pas connaissance de la bonne réception de sa réponse, sachant qu'il n'a de toute manière pas la possibilité de le renvoyer et doit attendre une prochaine requête de la supervision.

A-noter que l'on peut toujours continuer à recevoir en retour un acquittement négatif (nack), en cas de message de lecture ou d'envoi de commande considéré comme erroné (mauvaise compréhension).

Il faudrait également et éventuellement aussi étendre le cas du message de status 'agrégé' « AggregatedStatus », avec une version incluant l'acquiescement.

Mais pour ce message, on peut peut-être supposer qu'une supervision ne fera pas de demande de lecture en polling, donc sans doute pas gênant d'avoir un acquiescement systématique préalablement à la réponse. A-discuter d'ailleurs de l'intérêt réel de ce message, notamment pour le cas des contrôleurs trop limité et inutilisable, mais peut-être utile pour d'autres types d'équipements métiers... ? (en LCR on retrouve une notion très similaire)