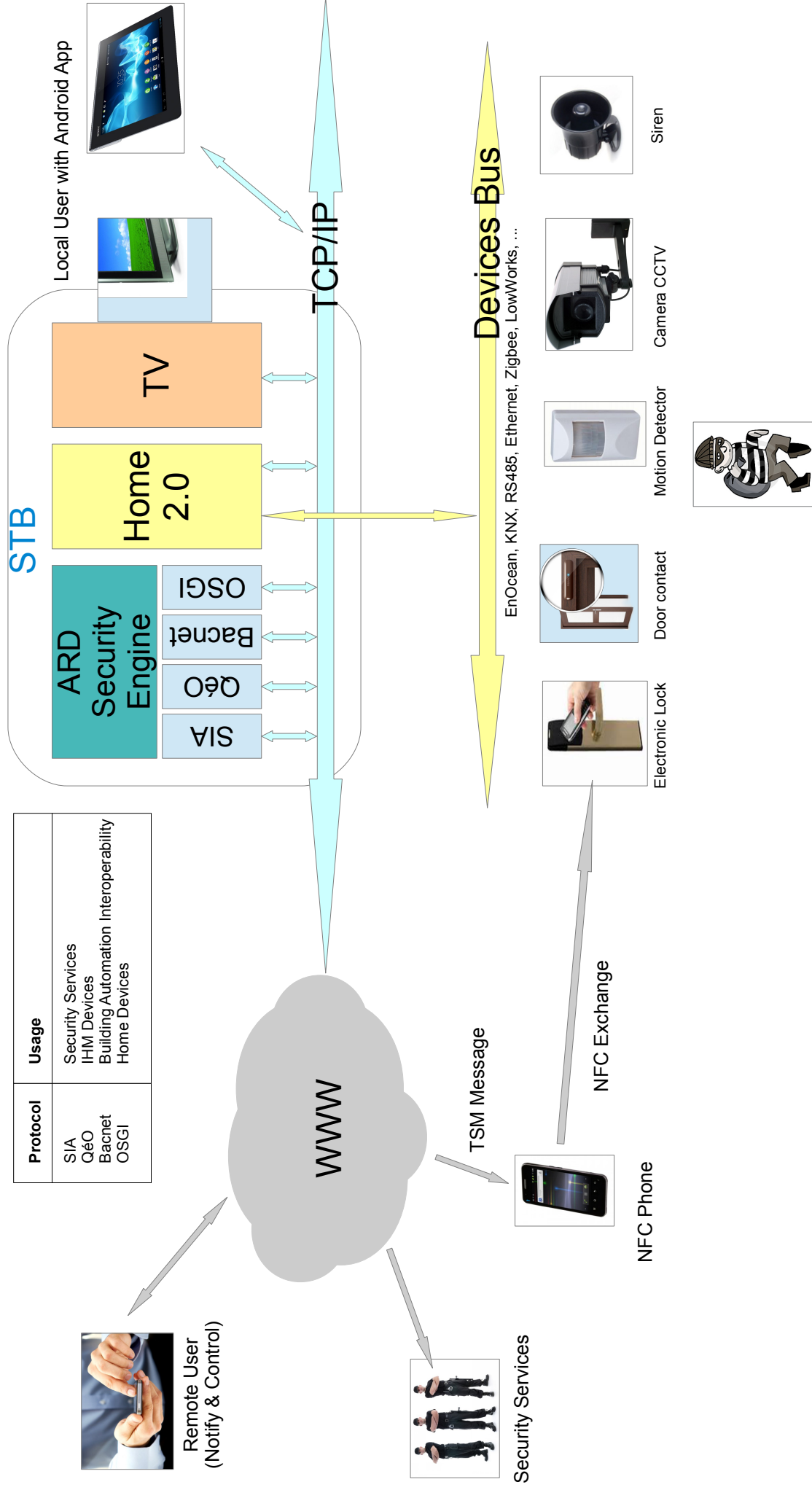


AppsGate - Home Security Architecture



Protocol	Usage
SIA	Security Services
QéO	IHM Devices
Bacnet	Building Automation Interoperability
OSGi	Home Devices



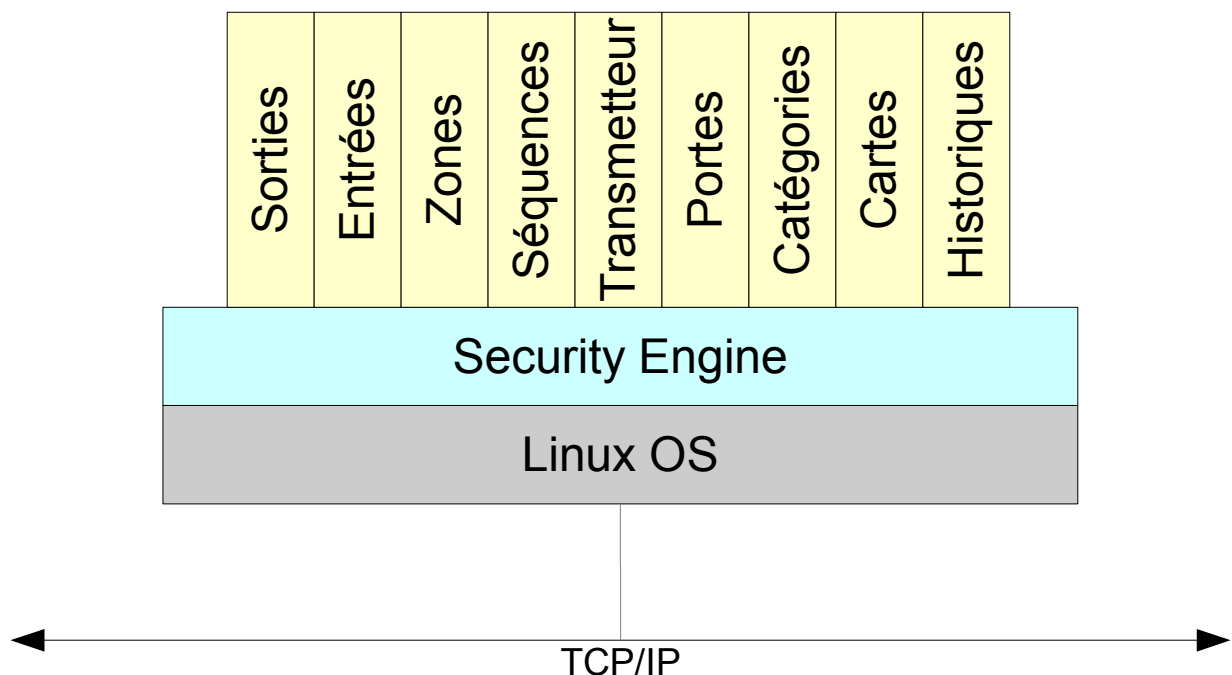
L'application de sécurité se compose d'un moteur et d'une base de données d'objets.

Sécurité :

- Détection
- Déclenchement d'actions
- Signalisation

Contrôle des accès :

- Identification
- Analyse de droits
- Gestion des dispositifs de verrouillage

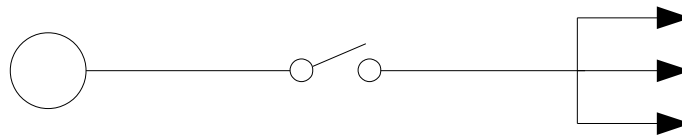


L'application utilise un protocole basé sur une construction JSON pour communiquer.

Un module d'historique permet de dater et de mémoriser les événements fonctionnels de l'application.

Entrée

Une entrée traite une information binaire issue d'un capteur ou d'un état interne. Une entrée peut être inhibée, c'est à dire rendue inactive selon des états fonctionnels de la centrale. Une entrée qui passe en « Alarme » déclenche des actions.



Valeur:

- Capteur
- Interne
- Relation Bool.

Inhibition :

- Zone
- Éjection
- Masquage

Actions :

- Pilotage
- Interne
- Séquence
- Transmission

Définition d'une entrée

* Question :

```
{
  "req_id" : n,           // Request id : de 0x0001 à 0xFFFF
  "request" : "set_input", // Définition d'une entrée.
  "input_idx" : n,        // Index de l'entrée.
  "name" : string,        // Nom de l'entrée. (20 caractères)
  "apparition_tt" : n,     // Code de télétransmission pour apparition (WORD)
  "disparition_tt" : n    // Code de télétransmission pour disparition (WORD)
}
```

Réponse :

```
{
  "req_id" : n,           // Request id : de 0x0001 à 0xFFFF
  "request" : "set_input", // Définition d'une entrée.
  "input_idx" : n,        // Index de l'entrée.
  "name" : string,        // Nom de l'entrée. (20 caractères)
  "apparition_tt" : n,     // Code de télétransmission pour apparition (WORD)
  "disparition_tt" : n,    // Code de télétransmission pour disparition (WORD)
  "result" : true         // OK.
}
```

Lecture de l'état d'une entrée

* Question :

```
{
  "req_id" : n,           // Request id : de 0x0001 à 0xFFFF
  "request" : "get_input", // Lecture de l'etat d'une entrée.
  "input_idx" : n         // Index de l'entrée.
}
```

Réponse :

```
{
  "req_id" : n,           // Request id : de 0x0001 à 0xFFFF
  "request" : "get_input", // Lecture de l'etat d'une entrée.
  "input_idx" : n,        // Index de l'entrée.
  "name" : string,        // Nom de l'entrée. (20 caractères)
  "apparition_tt" : n,     // Code de télétransmission pour apparition (WORD)
  "disparition_tt" : n,    // Code de télétransmission pour disparition (WORD)
  "status" : true,        // Etat de l'entrée true ou false.
  "ejected" : true,       // True si l'entrée est éjectée.
  "result" : true         // OK.
}
```

Fixer la Valeur d'une entrée

* Question :

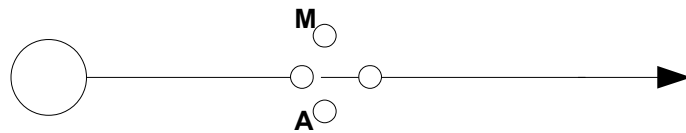
```
{
  "req_id" : n,           // Request id : de 0x0001 à 0xFFFF
  "request" : "force_input", // Forcer l'etat d'une entrée.
  "input_idx" : n,        // Index de l'entrée.
  "status" : true,        // Nouvel état de l'entrée true, false ou
                          // null(ne pas changer l'état).
  "ejected" : true        // Etat d'ejection de l'entrée : true, false ou
                          // null (ne pas changer l'éjection).
}
```

Réponse :

```
{
  "req_id" : n,           // Request id : de 0x0001 à 0xFFFF
  "request" : "force_input", // Forcer l'etat d'une entrée.
  "input_idx" : n,        // Index de l'entrée.
  "status" : true,        // Nouvel état de l'entrée : true ou false.
  "ejected" : true,       // Etat d'ejection de l'entrée : true ou false.
  "result" : true         // OK.
}
```

Sortie

Une sortie est le résultat d'une action ou la matérialisation d'un état interne. Une dérogation de forçage peut être appliquée de manière temporaire ou permanente sur une sortie.



Commande:

Action
Interne
Horaire

Forçage:

Manuel
Séquence

État:

Pilotage
Interne

Définition d'une sortie

* Question :

```
{
  "req_id" : n,                      // Request id : de 0x0001 à 0xFFFF
  "request" : "set_output",          // Définition d'une sortie.
  "output_idx" : n,                  // Index de la sortie.
  "name" : string,                   // Nom de la sortie. (20 caractères)
}
```

Réponse :

```
{
  "req_id" : n,                      // Request id : de 0x0001 à 0xFFFF
  "request" : "set_output",          // Définition d'une sortie.
  "output_idx" : n,                  // Index de la sortie.
  "name" : string,                   // Nom de la sortie. (20 caractères)
  "result" : true                    // OK.
}
```

Lecture de l'état d'une sortie

* Question :

```
{
  "req_id" : n,                      // Request id : de 0x0001 à 0xFFFF
  "request" : "get_output",          // Lecture de l'etat d'une sortie.
  "output_idx" : n,                  // Index de la sortie.
}
```

Réponse :

```
{
  "req_id" : n,                      // Request id : de 0x0001 à 0xFFFF
  "request" : "get_output",          // Lecture de l'etat d'une sortie.
  "output_idx" : n,                  // Index de la sortie.
  "name" : string,                   // Nom de la sortie. (20 caractères)
  "status" : true,                   // Etat de la sortie true ou false.
  "forced" : true,                   // Etat de forçage true ou false.
  "result" : true                    // OK.
}
```

Forcer une sortie

* Question :

```
{
  "req_id" : n,
  "request" : "force_output",
  "output_idx" : n,
  "force" : true
}
```

// Request id : de 0x0001 à 0xFFFF
// Forcer l'etat d'une sortie.
// Index de la sortie.
// Etat demandé de la sortie :
// true=forcée ON,
// false=forcée OFF ou
// null=pas forcée.

Réponse :

```
{
  "req_id" : n,
  "request" : "force_output",
  "output_idx" : n,
  "force" : true,

  "status" : true,
  "forced" : true,
  "result" : true
}
```

// Request id : de 0x0001 à 0xFFFF
// Forcer l'etat d'une sortie.
// Index de la sortie.
// Etat demandé de la sortie :
// true=force ON, false=force OFF ou
// null=not forced.
// Etat de la sortie true ou false.
// Etat de forçage true ou false.
// OK.

Séquence

Une séquence est la description d'un automatisme exprimé sous la forme d'un grafcet. Une séquence permet d'établir des liens fonctionnels entre les objets.



Lancement:

| Action
| Manuel

Exécution:

Définition d'une séquence

```
* Question :
{
  "req_id" : n,                // Request id : de 0x0001 à 0xFFFF
  "request" : "set_sequence",  // Définition du nom de la séquence.
  "seq_idx" : n,              // Index de la sequence.
  "name" : string             // Nom de la sequence. (20 caractères)
}
Réponse :
{
  "req_id" : n,                // Request id : de 0x0001 à 0xFFFF
  "request" : "set_sequence",  // Définition du nom de la séquence.
  "seq_idx" : n,              // Index de la sequence.
  "name" : string             // Nom de la sequence. (20 caractères)
  "result" : true             // OK.
}
```

Lecture de l'état d'une séquence

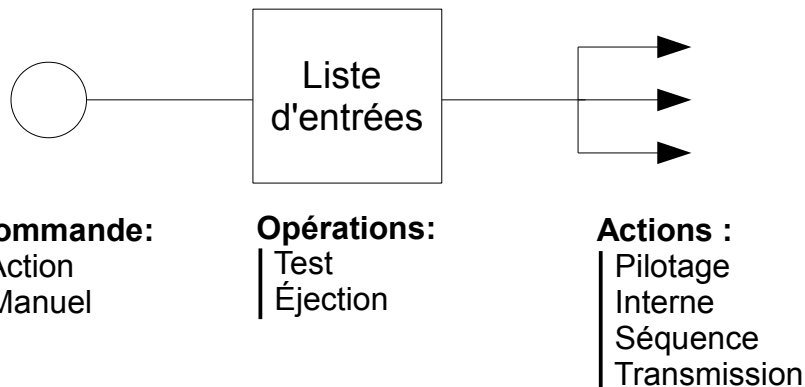
```
* Question :
{
  "req_id" : n,                // Request id : de 0x0001 à 0xFFFF
  "request" : "get_sequence",  // Lecture du nom de la séquence.
  "seq_idx" : n               // Index de la sequence.
}
Réponse :
{
  "req_id" : n,                // Request id : de 0x0001 à 0xFFFF
  "request" : "set_sequence",  // Définition du nom de la séquence.
  "seq_idx" : n,              // Index de la sequence.
  "name" : string             // Nom de la sequence. (20 caractères)
  "result" : true             // OK.
}
```

Lancer une séquence

```
* Question :
{
  "req_id" : n,                                // Request id : de 0x0001 à 0xFFFF
  "request" : "start_sequence",                // Lancemenent d'une séquence.
  "seq_idx" : n                                // Index de la sequence.
}
Réponse :
{
  "req_id" : n,                                // Request id : de 0x0001 à 0xFFFF
  "request" : "start_sequence",                // Lancemenent d'une séquence.
  "seq_idx" : n,                                // Index de la sequence.
  "result" : true                              // OK.
}
```


Zone

Une zone permet de regrouper des points (des capteurs) de détection afin d'appliquer des consignes globales de gestion.



Définition d'une zone

* Question :

```
{
  "req_id" : n,           // Request id : de 0x0001 à 0xFFFF
  "request" : "set_zone", // Définition d'une zone.
  "zone_idx" : n,         // Index de la zone.
  "name" : string,        // Nom de la zone. (20 caractères)
  "seq_on_idx" : n,       // Index de la séquence lors de la mise
                          // en surveillance de la zone.
  "seq_on_tt" : n,        // Code de télétransmission lors de la mise
                          // en surveillance de la zone. (WORD)
  "seq_off_idx" : n,      // Index de la séquence lors de la mise
                          // hors surveillance de la zone.
  "seq_off_tt" : n,       // Code de télétransmission lors de la mise
                          // hors surveillance de la zone. (WORD)
  "seq_eject_idx" : n,    // Index de la séquence lors de la mise
                          // en surveillance de la zone si des points
                          // sont éjectés.
  "seq_eject_tt" : n,     // Code de télétransmission lors de la mise
                          // en surveillance de la zone si des points
                          // sont éjectés. (WORD)
  "points" :              // Définition des points.
  [
    n, ...                // Index des entrées.
  ]
}
```

Réponse :

```
{
  "req_id" : n,           // Request id : de 0x0001 à 0xFFFF
  "request" : "set_zone", // Définition d'une zone.
  "zone_idx" : n,         // Index de la zone.
  "name" : string,        // Nom de la zone. (20 caractères)
  "seq_on_idx" : n,       // Index de la séquence lors de la MES de la zone.
  "seq_on_tt" : n,        // Code de télétransmission lors de la MES
                          // de la zone. (WORD)
  "seq_off_idx" : n,      // Index de la séquence lors de la MHS de la zone.
  "seq_off_tt" : n,       // Code de télétransmission lors de la MHS
                          // de la zone. (WORD)
  "seq_eject_idx" : n,    // Index de la séquence lors de la MES de la zone
                          // si des points sont éjectés.
  "seq_eject_tt" : n,     // Code de télétransmission lors de la MES de la zone
                          // si des points sont éjectés. (WORD)
  "points" :              // Définition des points.
  [
    n, ...                // Index des entrées.
  ],
  "result" : true         // OK.
}
```

Effacer une zone

```
* Question :
{
  "req_id" : n,                                // Request id : de 0x0001 à 0xFFFF
  "request" : "reset_zone",                    // Effacer une zone.
  "zone_idx" : n                               // Index de la zone.
}
Réponse :
{
  "req_id" : n,                                // Request id : de 0x0001 à 0xFFFF
  "request" : "reset_zone",                    // Effacer une zone.
  "zone_idx" : n,                             // Index de la zone.
  "result" : true                             // OK.
}
```

Lecture de l'état d'une zone

```
* Question :
{
  "req_id" : n,                                // Request id : de 0x0001 à 0xFFFF
  "request" : "get_zone",                      // Lecture de l'etat d'une zone.
  "input_idx" : n                              // Index de la zone.
}
Réponse :
{
  "req_id" : n,                                // Request id : de 0x0001 à 0xFFFF
  "request" : "get_zone",                      // Lecture de l'etat d'une zone.
  "input_idx" : n,                             // Index de la zone.
  "name" : string,                             // Nom de la zone. (20 caractères)
  "seq_on_idx" : n,                            // Index de la séquence lors de la MES de la zone.
  "seq_on_tt" : n,                             // Code de télétransmission lors de la MES
  // de la zone. (WORD)
  "seq_off_idx" : n,                           // Index de la séquence lors de la MHS de la zone.
  "seq_off_tt" : n,                           // Code de télétransmission lors de la MHS
  // de la zone. (WORD)
  "seq_eject_idx" : n,                         // Index de la séquence lors de la MES de la zone
  // si des points sont éjectés.
  "seq_eject_tt" : n,                         // Code de télétransmission lors de la MES de la zone
  // si des points sont éjectés. (WORD)
  "active" : true | false,                    // Zone en surveillance ou non.
  "alarm" : true | false,                     // En surveillance : au moins un point en alame.
  "eject" : true | false                      // Au moins un point éjecté.
  "points" :                                  // Définition des points.
  [
    n, ...                                    // Index des entrées.
  ],
  "result" : true                             // OK.
}
```

Mise en Service d'une zone

```
* Question :
{
  "req_id" : n,                                // Request id : de 0x0001 à 0xFFFF
  "request" : "activate_zone",                 // Mise en service
  "input_idx" : n                             // Index de la zone.
}
Réponse :
{
  "req_id" : n,                                // Request id : de 0x0001 à 0xFFFF
  "request" : "activate_zone",                 // Mise en service
  "input_idx" : n,                             // Index de la zone.
  "result" : true                             // OK.
}
```

Mise hors Service d'une zone

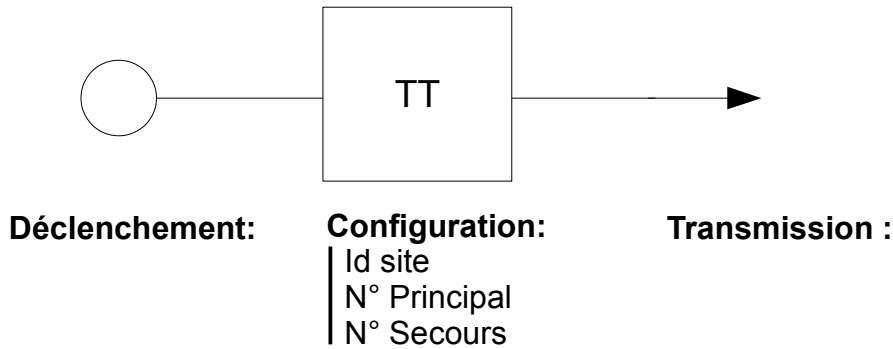
```
* Question :
{
  "req_id" : n,                                // Request id : de 0x0001 à 0xFFFF
  "request" : "deactivate_zone",               // Mise hors service
  "input_idx" : n                             // Index de la zone.
}
Réponse :
{
  "req_id" : n,                                // Request id : de 0x0001 à 0xFFFF
  "request" : "deactivate_zone",               // Mise hors service
  "result" : true                             // OK.
}
```

Test Pré-Alarme d'une zone

```
* Question :
{
  "req_id" : n,                                // Request id : de 0x0001 à 0xFFFF
  "request" : "test_zone",                     // Pré alarme : retourne la liste des
  "input_idx" : n                             // points en défaut (non éjectés).
                                              // Index de la zone.
}
Réponse :
{
  "req_id" : n,                                // Request id : de 0x0001 à 0xFFFF
  "request" : "test_zone",                     // Pré alarme : retourne la liste des
  "input_idx" : n,                             // points en défaut (non éjectés).
  "points" :                                   // Index de la zone.
  [                                             // Liste des points en défaut.
    n, ...                                     // Index des entrées.
  ],
  "result" : true                             // OK.
}
```

Télé-Transmetteur

Ce module gère la signalisation / acquittement d'un événement ou d'une alarme à destination du gestionnaire de sécurité.



Définition d'un télé-Transmetteur

* Question :

```
{
  "req_id" : n,                // Request id : de 0x0001 à 0xFFFF
  "request" : "set_conf_tt",    // Configuration de la télétransmission.
  "code_site" : "ARD GAP",      // Code du site pour le télétransmetteur
                                // (40 caractères).
  "main_number" : "0123456789", // Numéro de téléphone principal
                                // du télétransmetteur (40 caractères).
  "backup_number" : "0123456789", // Numéro de téléphone de secours
                                // du télétransmetteur (40 caractères).
}
```

Réponse :

```
{
  "req_id" : n,                // Request id : de 0x0001 à 0xFFFF
  "request" : "set_conf_tt",    // Configuration de la télétransmission.
  "code_site" : "ARD GAP",      // Code du site pour le télétransmetteur (40
caractères).
  "main_number" : "0123456789", // Numéro de téléphone principal
                                // du télétransmetteur (40 caractères).
  "backup_number" : "0123456789", // Numéro de téléphone de secours
                                // du télétransmetteur (40 caractères).
  "result" : true              // OK.
}
```

Lire la configuration d'un télé-Transmetteur

* Question :

```
{
  "req_id" : n,                // Request id : de 0x0001 à 0xFFFF
  "request" : "get_conf_tt"    // Configuration de la télétransmission.
}
```

Réponse :

```
{
  "req_id" : n,                // Request id : de 0x0001 à 0xFFFF
  "request" : "get_conf_tt",    // Configuration de la télétransmission.
  "code_site" : "ARD GAP",      // Code du site pour le télétransmetteur
                                // (40 caractères).
  "main_number" : "0123456789", // Numéro de téléphone principal
                                // du télétransmetteur (40 caractères).
  "backup_number" : "0123456789", // Numéro de téléphone de secours
                                // du télétransmetteur (40 caractères).
  "result" : true               // OK.
}
```

Carte d'identification

Dispositif permettant une identification d'un porteur. Dans la majorité des cas, il s'agit d'une carte utilisant la norme ISO 14443 pour échanger des informations d'identité avec une tête de lecture par couplage radio à 13,56MHz.

Code d'identification
Dates de Validité
Catégories de population

Définition d'une Carte

* Question :

```
{
  "req_id" : n,                      // Request id : de 0x0001 à 0xFFFF
  "request" : "set_card",           // Création / Modification d'un Badge.
  "card_idx" : n,                   // Index du badge.
  "card_num" : "xxXXxxXXxxXXxxXX", // Identifiant de carte lue : 8 octets
                                      // sous forme Hex (BCD si num ARD),
                                      // cadré à droite, complété à gauche
                                      // par des 00.
  "name" : string,                  // Nom du porteur du badge. (20 caractères)
  "cat_idx" : n,                     // Index de catégorie. (profil)
  "validity_start": "YYYY/MM/DD hh:mn:ss", // Date et heure de début de validité.
  "validity_end": "YYYY/MM/DD hh:mn:ss",   // Date et heure de fin de validité.
  "validity": true | false           // Valide ou invalide le badge.
}
```

Réponse :

```
{
  "req_id" : n,                      // Request id : de 0x0001 à 0xFFFF
  "request" : "set_card",           // Création / Modification d'un Badge.
  "card_idx" : n,                   // Index du badge.
  "card_num" : "xxXXxxXXxxXXxxXX", // Identifiant de carte lue : 8 octets
                                      // sous forme Hex (BCD si num ARD),
                                      // cadré à droite, complété à gauche
                                      // par des 00.
  "name" : string,                  // Nom du porteur du badge. (20 caractères)
  "cat_idx" : n,                     // Index de catégorie. (profil)
  "validity_start": "YYYY/MM/DD hh:mn:ss", // Date et heure de début de validité.
  "validity_end": "YYYY/MM/DD hh:mn:ss",   // Date et heure de fin de validité.
  "validity": true | false,         // Valide ou invalide le badge.
  "result" : true                    // OK.
}
```

Suppression d'une carte

```
* Question :
{
    "req_id" : n,
    "request" : "reset_card",
    "card_idx" : n
}
Réponse :
{
    "req_id" : n,
    "request" : "reset_card",
    "card_idx" : n,
    "result" : true
}
```

// Request id : de 0x0001 à 0xFFFF
// Effacer (supprimer) un badge.
// Index du badge.

// Request id : de 0x0001 à 0xFFFF
// Effacer (supprimer) un badge.
// Index du badge.
// OK.

Suppression de la table des cartes

```
* Question :
{
    "req_id" : n,
    "request" : "reset_all_cards"
}
Réponse :
{
    "req_id" : n,
    "request" : "reset_all_cards",
    "result" : true
}
```

// Request id : de 0x0001 à 0xFFFF
// RAZ de toute la table des badges.

// Request id : de 0x0001 à 0xFFFF
// RAZ de toute la table des badges.
// OK.

Lecture de la définition d'une carte

```
* Question
{
    "req_id" : n,
    "request" : "get_card",
    "card_idx" : n,
}
Réponse :
{
    "req_id" : n,
    "request" : "get_card",
    "card_idx" : n,
    "card_num" : "xxXXxxXXxxXXxxXX",
    "name" : string,
    "cat_idx" : n,
    "validity_start": "YYYY/MM/DD hh:mn:ss",
    "validity_end": "YYYY/MM/DD hh:mn:ss",
    "validity": true | false,
    "result" : true
}
```

// Request id : de 0x0001 à 0xFFFF
// Lecture d'un Badge.
// Index du badge.

// Request id : de 0x0001 à 0xFFFF
// Lecture d'un Badge.
// Index du badge.
// Identifiant de carte lue : 8 octets
// sous forme Hex (BCD si num ARD),
// cadré à droite, complété à gauche
// par des 00.
// Nom du porteur du badge.
// (20 caractères)
// Index de catégorie. (profil)
// Date et heure de début de validité.
// Date et heure de fin de validité.
// Valide ou invalide le badge.
// OK.

Catégorie

C'est une table de construction des droits associant des conditions à chaque accès possible.



Liste des droits

Définition d'une Catégorie

* Question :

```
{
  "req_id" : n,                // Request id : de 0x0001 à 0xFFFF
  "request" : "set_cat"        // Definition de la catégorie.
  "cat_idx" : n,               // Index de la catégorie.
  "name" : string,             // Nom de la catégorie. (20 caractères)
  "doors" :                    // Liste des portes (1 à 16) autorisées.
  [
    n,                         // Index de porte
    ...
  ]
}
```

Réponse :

```
{
  "req_id" : n,                // Request id : de 0x0001 à 0xFFFF
  "request" : "set_cat"        // Definition de la catégorie.
  "cat_idx" : n,               // Index de la catégorie.
  "name" : string,             // Nom de la catégorie. (20 caractères)
  "doors" :                    // Liste des portes (1 à 16) autorisées.
  [
    n,                         // Index de porte
    ...
  ],
  "result" : true              // OK.
}
```


Suppression d'une catégorie

```
* Question :
{
  "req_id" : n,                                // Request id : de 0x0001 à 0xFFFF
  "request" : "reset_cat",                      // Effacer une catégorie.
  "cat_idx" : n                                // Index de la catégorie.
}
Réponse :
{
  "req_id" : n,                                // Request id : de 0x0001 à 0xFFFF
  "request" : "reset_cat",                      // Effacer une catégorie.
  "cat_idx" : n,                                // Index de la catégorie.
  "result" : true                              // OK.
}
```

Lecture de la définition d'une catégorie

```
* Question :
{
  "req_id" : n,                                // Request id : de 0x0001 à 0xFFFF
  "request" : "get_cat",                       // Lecture d'une Catégorie
  "cat_idx" : n                                // Index de la catégorie.
}
Réponse :
{
  "req_id" : n,                                // Request id : de 0x0001 à 0xFFFF
  "request" : "get_cat",                       // Lecture d'une Catégorie.
  "cat_idx" : n,                                // Index de catégorie. (profil)
  "name" : string,                             // Nom de la catégorie. (20 caractères)
  "doors" :                                    // Liste des portes (1 à 16) autorisées.
  [
    n,                                          // Index de porte
    ...
  ],
  "result" : true                              // OK.
}
```

Porte

Associé à une tête de lecture, une porte physique ou virtuelle permet le contrôle d'accès à un espace géographique ou fonctionnel.

Configuration

Définition d'une Porte

* Question :

```
{
  "req_id" : n,                // Request id : de 0x0001 à 0xFFFF
  "request" : "set_door",      // Définition d'une porte.
  "door_idx" : n,              // Index de la porte.
  "name" : string,             // Nom de la porte. (20 caractères)
}
```

Réponse :

```
{
  "req_id" : n,                // Request id : de 0x0001 à 0xFFFF
  "request" : "set_door",      // Définition d'une porte.
  "door_idx" : n,              // Index de la porte.
  "name" : string,             // Nom de la porte. (20 caractères)
  "result" : true              // OK.
}
```

Suppression d'une porte

```
* Question :
{
  "req_id" : n,
  "request" : "reset_cat",
  "cat_idx" : n
}
Réponse :
{
  "req_id" : n,
  "request" : "reset_cat",
  "cat_idx" : n,
  "result" : true
}
```

// Request id : de 0x0001 à 0xFFFF
// Effacer une catégorie.
// Index de la catégorie.

// Request id : de 0x0001 à 0xFFFF
// Effacer une catégorie.
// Index de la catégorie.
// OK.

Lecture de la définition d'une porte

```
* Question :
{
  "req_id" : n,
  "request" : "get_door",
  "door_idx" : n
}
Réponse :
{
  "req_id" : n,
  "request" : "get_door",
  "door_idx" : n,
  "name" : string,
  "result" : true
}
```

// Request id : de 0x0001 à 0xFFFF
// Définition d'une porte.
// Index de la porte.

// Request id : de 0x0001 à 0xFFFF
// Définition d'une porte.
// Index de la porte.
// Nom de la porte. (20 caractères)
// OK.

Fonctions Diverses

Lecture Date et Heure application

```
* Question :
{
  "req_id" : n,
  "request" : "get_date_time"
}
Réponse :
{
  "req_id" : n,
  "request" : "get_date_time",
  "timestamp" : n,
  "date" : "YYYY/MM/DD",
  "result" : true
}
```

// Request id : de 0x0001 à 0xFFFF
// Récupération de la date et de l'heure.

// Request id : de 0x0001 à 0xFFFF
// Récupération de la date et de l'heure.
// Date time Unix.
// Date.
// OK.

Effacement de la dernière carte utilisée

```
* Question :
{
  "req_id" : n,
  "request" : "reset_card_read"
}
Réponse :
{
  "req_id" : n,
  "request" : "reset_card_read",
  "result" : true
}
```

// Request id : de 0x0001 à 0xFFFF
// Effacer le numéro de dernier badge lu.

// Request id : de 0x0001 à 0xFFFF
// Effacer le numéro de dernier badge lu.
// OK.

Lecture de la dernière carte utilisée

```
* Question :
{
  "req_id" : n,
  "request" : "door_card_read",
  "door_idx" : n
}
Réponse :
{
  "request" : "door_card_read",
  "door_idx" : n,
  "card_num" : "xxxxxxxxxxxxxxxx",
  "result" : true
}
```

// Request id : de 0x0001 à 0xFFFF
// Récupération du dernier badge lu
// à une porte.
// Index de la porte.

// Index de la porte.
// Identifiant de carte lue : 8 octets
// sous forme Hex (BCD si num ARD),
// cadré à droite, complété à gauche
// par des 00.
// OK.

Événements



Abonnement aux événements

```
* Question :
{
    "req_id" : n,
    "request" : "subscribe",
    "timestamp" : n,
    "event_idx" : n
}
Réponse :
{
    "req_id" : n,
    "request" : "subscribe",
    "timestamp" : n,
    "event_idx" : n,
    "count" : n,
    "result" : true
}
```

évenements.
si première fois).
première fois).
évenements.
si première fois).
première fois).
le dernier événement reçu.

// Request id : de 0x0001 à 0xFFFF
// Abonnement pour la réception des
// Timestamp du dernier événement reçu (0
// Index du dernier événement reçu (0 si
// Request id : de 0x0001 à 0xFFFF
// Abonnement pour la réception des
// Timestamp du dernier événement reçu (0
// Index du dernier événement reçu (0 si
// Nombre d'événements disponible depuis
// OK.

Polling d'un événement

```
* Question :
{
    "req_id" : n,
    "request" : "get_events"
    "timestamp" : n,
si première fois).
    "event_idx" : n
première fois).
}
Réponse :
{
    "req_id" : n,
    "request" : "get_events",
    "timestamp" : n,
si première fois). (Copie de la question)
    "event_idx" : n,
première fois). (Copie de la question)
    "events" :
    [
        // Voir "event" plus loin.
        ...
    ]
    "result" : true
}
```

```
// Request id : de 0x0001 à 0xFFFF
// Lecture d'événements
// Timestamp du dernier événement reçu (0
// Index du dernier événement reçu(0 si
// Request id : de 0x0001 à 0xFFFF
// Lecture d'événements.
// Timestamp du dernier événement reçu (0
// Index du dernier événement reçu(0 si
// Les événements (max 20).
// OK.
```

Interprétation d'un événement

```
* Événement :
{
  "event" :                               // Un événement.
  {
    "timestamp" : n,                      // Timestamp de l'événement.
    "event_idx" : n,                      // Index de l'événement.
    "class" : "system",                  // Classe de l'événement parmi :
"system", "zone", "sequence", "output", "input", "card"

    => Le reste dépend de la classe de l'événement :

    "class" : "system",                   // Événement system
    "cause" : "reset",                   // Redémarrage d'isoview
    "version" : n                         // Version d'isoview. 300 => 3.00
    OU
    "cause" : "reinit",                   // Réinitialisation de la base de
données.
    "version" : n                         // Version d'isoview. 300 => 3.00
    OU
    "cause" : "date_time",                // Changement de date et heure
    "date" : "YYYY/MM/DD"                // Date (sans heure...)
    OU
    "cause" : "archives",                 // Débordement de capacité des archives
(On a fait le tour du tableau)
    "version" : n                         // Version d'isoview. 300 => 3.00

    "class" : "zone",                     // Événement de zone
    "zone_idx" : n,                       // Index de zone
    "active" : true                       // Indique que la zone change d'état.
    OU
    "input_idx" : n,                       // Index de l'entrée.
    "ejected" : true                       // Ejection ou réintégration d'un point
dans la zone.
    OU
    "input_idx" : n,                       // Index de l'entrée.
    "eject" : true,                       // Il y a ou non des points éjecté dans
la zone.
    "active" : true,                       // La zone est active ou non.
    "alarm" : true                         // La zone est en alarme ou non.

    "class" : "sequence",                 // Événement de lancement de séquence
    "seq_idx" : n                         // Index de la séquence lancée

    "class" : "output",                   // Événement de sortie
    "output_idx" : n,                     // Index de la sortie
    "old_status" : true,                   // Ancien état
    "new_status" : true                    // Nouvel état

    "class" : "input",                     // Événement d'entrée
    "input_idx" : n,                       // Index de l'entrée
    "old_status" : true,                   // Ancien état
    "new_status" : true                    // Nouvel état
    "old_ejected" : true,                  // Ancien état d'éjection
    "new_ejected" : true                    // Nouvel état d'éjection
```

```

        "class" : "card",                // Événement concernant les badges.
        "cause" : "raz_cards"           // Pas de status, la table des badges à
été effacée.
    OU
    // Fonction du status :
    "status" : "ok",                    // Badge accepté
    "door_idx" : n,                     // Index de la porte ayant lu le badge.
    "card_idx" : n                      // Index du badge.
    OU
    "status" : "ko",                    // Badge refusé
    "door_idx" : n,                     // Index de la porte ayant lu le badge.
    "card_idx" : n,                     // Index du badge.
    "cause" : "Bloked card"             // Cause du refus permis : "Bloked card",
    "Invalid card", "Not yet valid card", "No more valid card", "No rights", "Refused
door", "Invalid rights", "Timezone off", "Invalid door" ou "???" (Inconnue)
    OU
    "status" : "??",                    // Badge inconnu.
    "door_idx" : n,                     // Index de la porte ayant lu le badge.
    "card_c1" : n,                      // Dernier chiffre du badge.
    "card_c2" : n,                      // Avant dernier chiffre du badge.
    "card_c3" : n,                      // .. chiffre du badge.
    "card_c4" : n,                      // .. chiffre du badge.
    "card_c5" : n,                      // .. chiffre du badge.
}
}

```