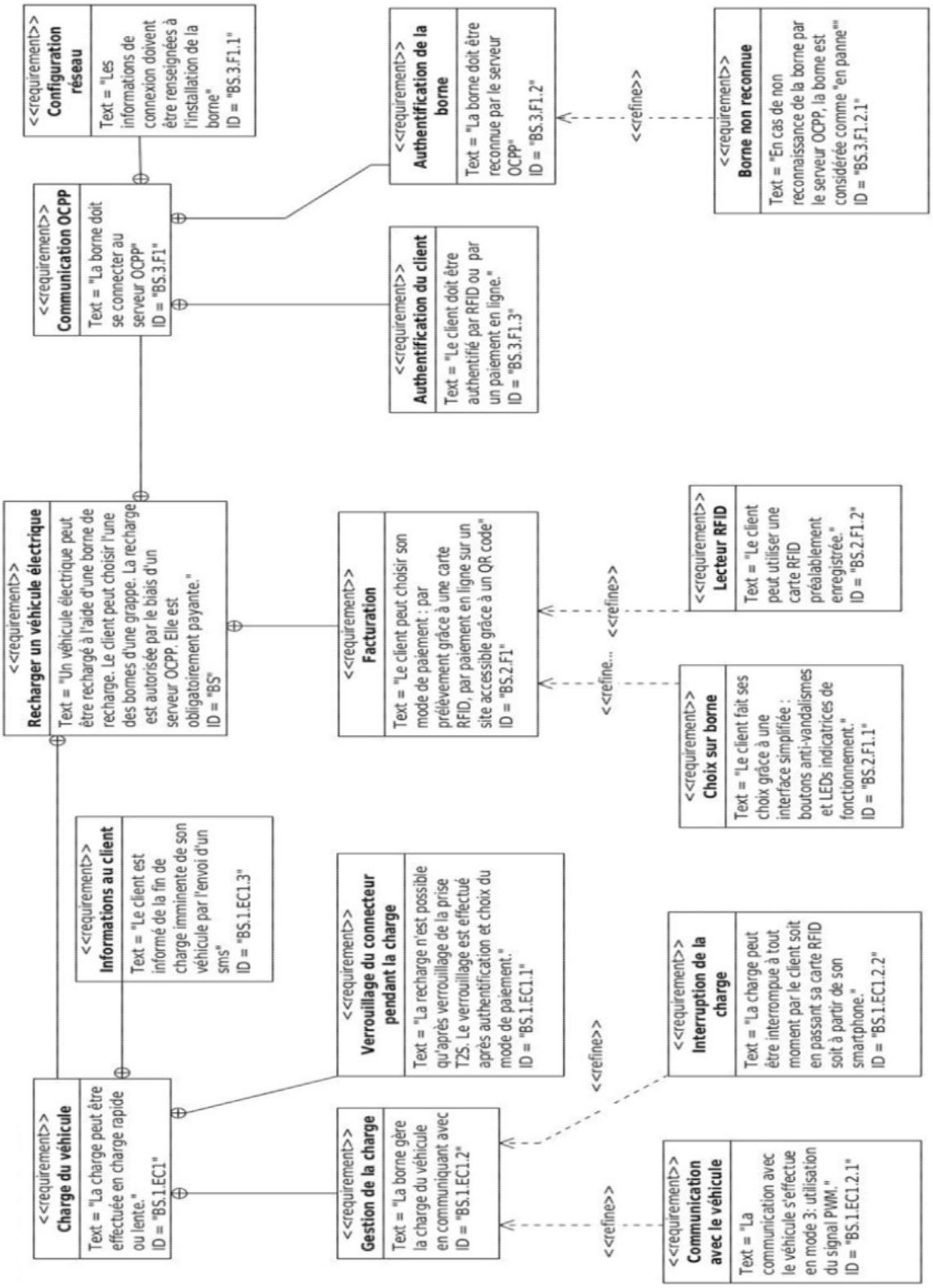
Documentation

Table des matières

DOCUMENTATION PP1 : Diagramme des exigences



DOCUMENTATION PP2 : Lecteurs RFID

INVEO

RFID IND Modbus-Mif

General features

The reader is equipped with an RS485 port supporting Modbus RTU protocol and a USB port used for configuration and testing of the module.
The device has two relay outputs and two inputs.

Technical data:

Supply voltage:12-24VDC
Power supply: 40mA (12V)

Transponders:
- Supported transponder standard: Mifare
- Carrier frequency: 13,56 MHz
- Reading distance to 10cm (depending on the type of transponder used)

Communication:
- 1 RS485 port – modbus RTU
- 1 USB port to configuration

Inputs/Outputs
- 2 relay outputs 1A@30VDC
- 2 inputs

Enclosure:
- IP Rating: IP65

PEPPERL+FUCHS

IPH-FP-V1

Technical data

General specifications		
Operating frequency		125 kHz
Transfer rate		2 kBit/s
Sensing range		
Read distance		0 ... 100 mm
Write distance		0 ... 80 mm
Width		max. 80 mm
Operating distance		maximum: 100 mm
UL File Number		E87056
Functional safety related parameters		
$MTTF_d$		710 a
Mission Time (T_M)		10 a
Diagnostic Coverage (DC)		0 %
Indicators/operating means		
LED green/yellow		green: power on green flashing: read/write attempt performed yellow: data carrier detected
Electrical specifications		
Power consumption	P_0	≤ 1.2 W
Supply		from the IDENTControl
Ambient conditions		
Ambient temperature		-25 ... 70 °C (-13 ... 158 °F)
Storage temperature		-40 ... 85 °C (-40 ... 185 °F)
Mechanical specifications		
Degree of protection		IP67
Connection		M12 x 1 connector
Material		
Housing		PBT
Base		diecast aluminum
Encapsulation compound		Fermodur
Installation		
Distance between two heads		Multiplex on: ≥ 100 mm Multiplex off: ≥ 550 mm
Mass		approx. 380 g
Compliance with standards and directives		
Directive conformity		
R&TTE Directive 1995/5/EC		EN 301489-1 V1.8.1 (2008-04), EN 301489-3 V1.4.1 (2002-08), EN 300330-2 V1.3.1 (2006-04), EN 60950-1:2006

Telemecanique
OsiSense XG series
Smart Antenna

Characteristics		XGCS4901201 - format 40	XGCS8901201 - format 80	XGCS4908201	XGCS49LB201
Temperature	Operation	-25...+70°C (-13...158°F)		-40...+70°C (-40...158°F)	
	Storage	-40...+85 °C (-40...+185°F)		-40...+85 °C (-40...+185°F)	
Degree of protection		IP65 in accordance with IEC60529			IP69K on front face/IP65 on back part
Vibration resistance EN 60068.2.27 EN 60068.2.6		2 mm (0.078 in) from 5 to 29.5 Hz / 7 g (7 gn) from 29.5 to 150 Hz 30 g (30 gn) / 11 ms			
Resistance to mechanical shocks		IK04 according to EN 50102		IK04 according to EN 50102	
Standards / Certifications		CE, cULus, EN 300330-1/2, EN 301489-01/03, FCC Part 15 IC			
Immunity to disturbances		Resistance to electrostatic discharges, radiated electromagnetic fields, fast transients, electrical surges, conducted and induced interference and power frequency magnetic field according to IEC 61000/EN 55022.			
Unit dimensions		40x40x15 mm (1.57x1.57x0.59 in)	80x80x26 mm (3.15x3.15x1.02 in)	40x40x15 mm (1.57x1.57x0.59 in) Mounting ø22 mm (0.87 in)	
RFID frequency		13.56 MHz			
Type of associated tags		Standardized ISO 15693 and ISO 14443 tags Automatic detection of the tag type			
Nominal sensing distance (according to the associated tag)		18...70 mm (0.70...2.75 in)	20...100 mm (0.78...3.94 in)	10...70 mm (0.39...2.75 in)	
Nominal power supply		24 Vdc PELV			
Power supply voltage limits		19.2...29 V ripple included			
Power consumption		< 60 mA			
Serial links	Type	RS485			
	Protocol	Modbus RTU			
	Speed	9600...115 200 Bauds: Automatic detection			
Display		1 dual color LED for network communication 1 dual color LED for RFID communication (Tag present, Smart Antenna/tag dialog)			
Lights		-			2 Multicolor lights (7 colors)
Connection		5-way male M12 connector for connection to the communication network and power supply			
Tightening torque for the mounting		< 1 Nm (8.85 lbf-in)	< 3 Nm (26.55 lbf-in)	< 2.2 Nm (19.5 lbf-in)	

PEPPERL+FUCHS
IPH-FP-V1

Technical data

General specifications		
Operating frequency		13.56 MHz
Transfer rate		26 kBit/s
Sensing range		
Read distance		0 ... 130 mm
Write distance		0 ... 130 mm
Width		max. 100 mm
UL File Number		E87056
Functional safety related parameters		
$MTTF_d$		680 a
Mission Time (T_M)		10 a
Diagnostic Coverage (DC)		0 %
Indicators/operating means		
LED red/green		Green: power on Flashing green: IO-Link communication Flashing red/green: IO-Link communication interrupted
LED blue/yellow		Blue: Write/read attempt performed Yellow: Read/write tag detected
Electrical specifications		
Rated operating voltage	U_B	20 ... 30 V DC , ripple 10 %ss
Power consumption	P_0	≤ 2 W
Interface		
Interface type		IO-Link
Protocol		IO-Link V1.1
Cycle time		min. 4 ms
Mode		COM 3 (230.4 kBaud)
Process data width		32 Byte
SIO mode support		no
Directive conformity		
Electromagnetic compatibility Directive 2014/30/EU		EN 61000-6-2:2005 EN 61000-6-4:2007
Radio and telecommunication terminal equipment Directive 2014/53/EU		EN 301489-1 V1.9.2:2011 EN 301489-3 V1.6.1:2013 EN 300330 V2.1.1:2017 EN 62368-1:2014+AC:2015 EN 50364:2010

BALLUFF

BIS M-620-068-A01-00-ST29

HF (13.56 MHz)

Display/Operation

(BB) Ready	Green LED
RF	LED yellow

Electrical connection

Connection (COM 1)	X1 (RS232/supply voltage): M12x1-Male, 8-pole
Connection slots	RCA-Female X2 (IN/OUT): M12x1-Female, 8-pole

Electrical data

Control input	1 (optocoupler isolated) PNP/NPN
Control output	2 (optocoupler isolated)
Current consumption max. at 24 V DC	500 mA
Input current max. at 24 V	28 mA
Operating voltage Ub	19.2...28.8 VDC
Operating voltage, output Vs	6...30 V DC
Output current max.	500 mA (500 mA ext. supply) 100 mA (int. supply)
Residual ripple max.	10 %
Voltage control	6...30 VDC

Environmental conditions

Ambient temperature	-20...50 °C
Continuous shock load	yes
EN 60068-2-27, Shock	yes
EN 60068-2-32 Free fall	yes
EN 60068-2-6, Vibration	yes
IP rating	IP65 with connector
Storage temperature	-20...70 °C

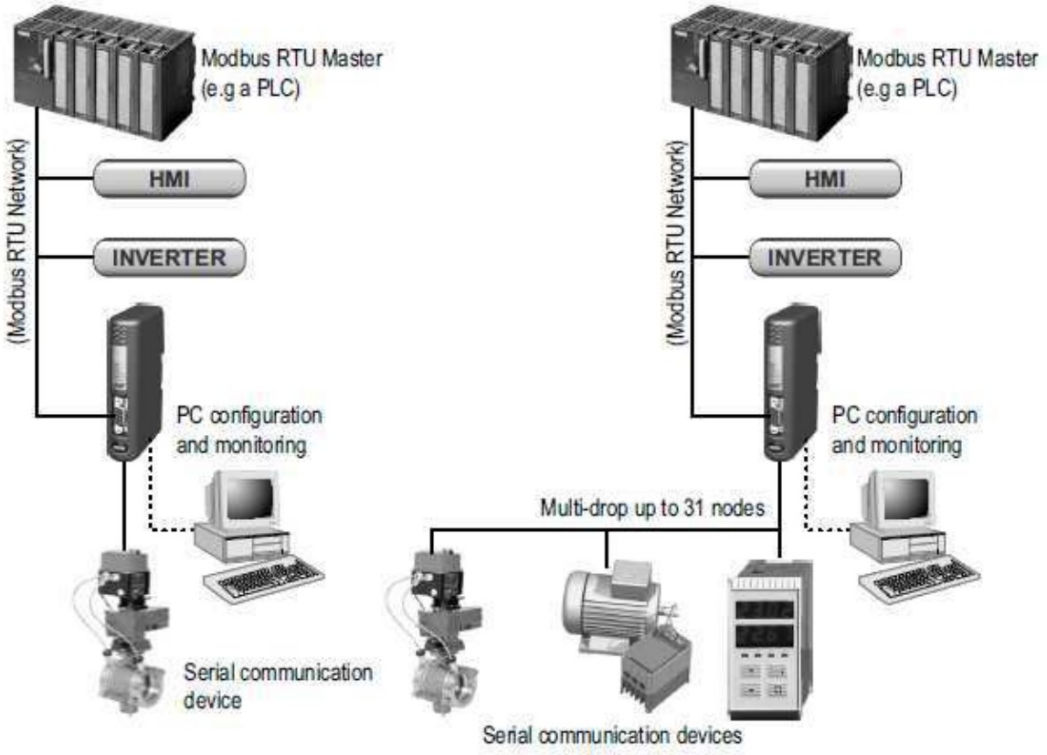
Output/Interface

Interface	RS232

DOCUMENTATION PP3 : Communicateur AnyBus

Le **AnyBus Communicator** est une **passerelle de communication** permettant de connecter des appareils non compatibles avec des protocoles industriels courants (comme Modbus) à un réseau de terrain industriel moderne.

Cette passerelle sert à traduire les données d'un protocole propriétaire vers un protocole industriel standardisé, sans nécessiter de modification sur l'équipement d'origine. Elle est fréquemment utilisée pour intégrer d'anciens équipements dans de nouvelles installations avec des systèmes modernes de supervision.



Les fonctionnalités principales incluent :

- **Compatibilité avec plusieurs protocoles** : prend en charge divers protocoles industriels, tels que Modbus RTU, Modbus TCP/IP, et d'autres standards de communication.
- **Adaptabilité** : elle peut être configurée pour dialoguer avec des équipements via des interfaces série ou des interfaces réseau.
- **Passerelle configurable** : offre des options de configuration permettant de mapper les registres de données ou les adresses mémoires des anciens équipements vers un format interprétable par un Automate Programmable Industriel ou un système de supervision.
- **Fiabilité** : conçue pour un fonctionnement continu dans des environnements industriels difficiles, garantissant une communication robuste et sécurisée.

Sous-réseau

La passerelle peut adresser jusqu'à 31 nœuds et prend en charge les standards physiques suivants :

- RS432
- RS422
- RS485

Interface Modbus RTU

La connectivité Modbus RTU est assurée par la technologie brevetée Anybus, une solution de communication industrielle éprouvée, utilisée dans le monde entier par les principaux fabricants de produits d'automatisation industrielle.

- Interface de bus isolée galvaniquement
- Accès aux bobines et registres
- Fonctionnement en RS232 ou RS485
- Interrupteurs de configuration intégrés
- Fonctionnement à des vitesses de 1200 à 57600 bps

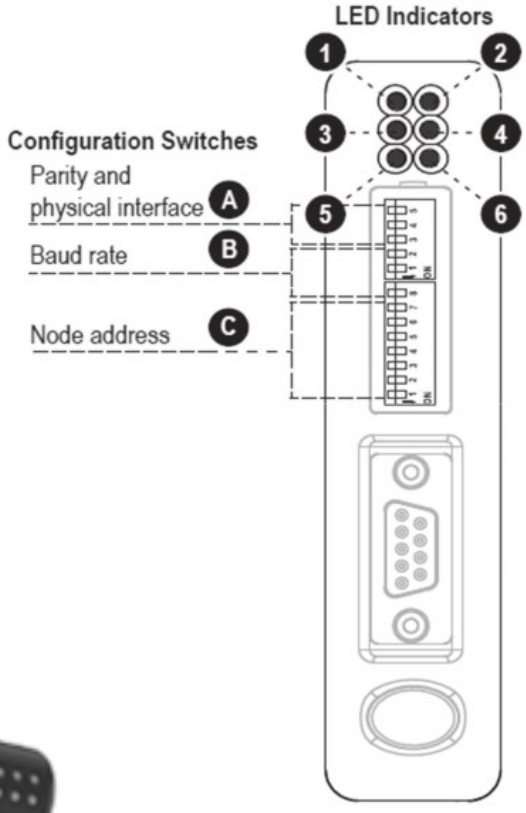
Passerelle Modbus

Les interrupteurs de configuration :

Ils déterminent les paramètres de communication de base pour l'interface Modbus.

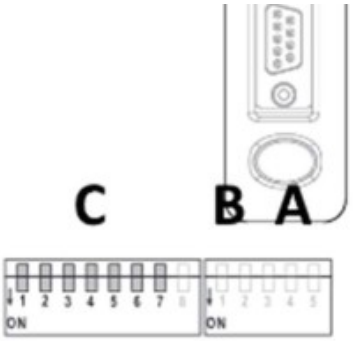
Normalement, ces interrupteurs sont recouverts par un capot en plastique. Lors de la suppression du capot, évitez de toucher les cartes de circuit et les composants. Si des outils sont utilisés pour ouvrir le capot, faites preuve de prudence.

Notez que les paramètres ne peuvent pas être modifiés pendant le fonctionnement, c'est-à-dire que la passerelle doit être redémarrée pour que les modifications prennent effet.



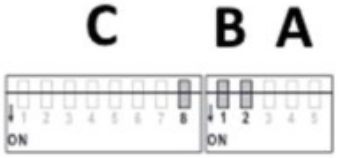
Node Address

Node Address	Sw. 1	Sw. 2	Sw. 3	Sw. 4	Sw. 5	Sw. 6	Sw. 7
(reserved)	OFF	OFF	OFF	OFF	OFF	OFF	OFF
1	OFF	OFF	OFF	OFF	OFF	OFF	ON
2	OFF	OFF	OFF	OFF	OFF	ON	OF
...	...	...	...	...	...	...	...
126	ON	ON	ON	ON	ON	ON	OFF
127	ON	ON	ON	ON	ON	ON	ON



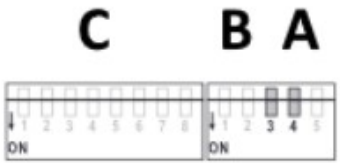
Baudrate Configuration

Baudrate	Sw. 8	Sw. 1	Sw. 2
(reserved)	OFF	OFF	OFF
1200 bps	OFF	OFF	ON
2400 bps	OFF	ON	OFF
4800 bps	OFF	ON	ON
9600 bps	ON	OFF	OFF
19200 bps (standard)	ON	OFF	ON
38400 bps	ON	ON	OFF
57600 bps	ON	ON	ON



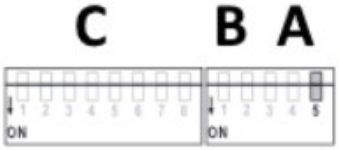
Parity & Stop Bits

Parity	Sw. 3	Sw. 4
(reserved)	OFF	OFF
No parity, 2 stop bits	OFF	ON
Even parity, 1 stop bit	ON	OFF
Odd parity, 1 stop bit	ON	ON



Physical Interface

Interface Type	Sw. 5
RS-485	OFF
RS-232	ON



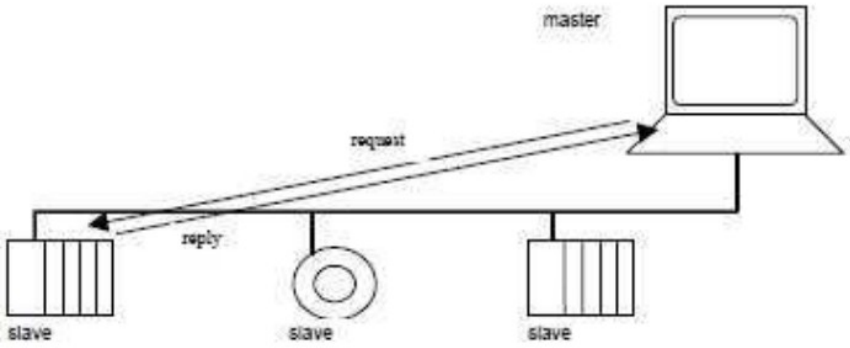
DOCUMENTATION PP4 : « Modbus sur lien série »

Couche liaison de données Modbus : Principe du protocole maître/esclave Modbus

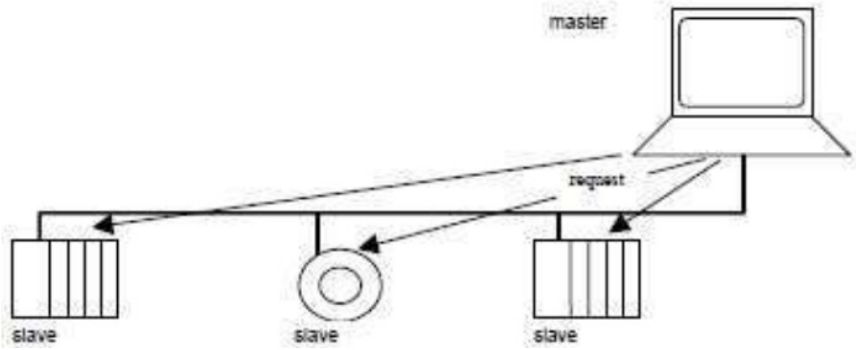
Le protocole Modbus sur ligne série est un protocole maître-esclaves. Un seul maître (à la fois) est connecté au bus, et un ou plusieurs nœuds esclaves (jusqu'à un maximum de 247) sont connectés au même bus série. Une communication MODBUS est toujours initiée par le maître. Les nœuds esclaves ne transmettront jamais de données sans avoir reçu une requête du nœud maître. Les nœuds esclaves ne communiqueront jamais entre eux. Le nœud maître n'initie qu'une seule transaction MODBUS à la fois.

Le nœud maître émet une requête MODBUS aux nœuds esclaves selon deux modes :

- En **mode unicast**, le maître s'adresse à un esclave individuellement. Après avoir reçu et traité la requête, l'esclave renvoie un message (une réponse) au maître.
  Dans ce mode, une transaction MODBUS se compose de 2 messages : une requête du maître et la réponse de l'esclave.
  Chaque esclave doit avoir une adresse unique (de 1 à 247) afin de pouvoir être adressé indépendamment des autres nœuds.



- En **mode broadcast**, le maître peut envoyer une requête à tous les esclaves.
  Aucune réponse n'est renvoyée aux requêtes broadcast initiées par le maître. Les requêtes broadcast sont nécessaires pour les commandes d'écriture. Tous les dispositifs doivent accepter les requêtes broadcast pour ces fonctions. L'adresse **0** est réservée pour identifier un échange de type broadcast.



DOCUMENTATION PP5 : Codage ATQA des circuits intégrés de cartes sans contact NXP

Table ATQA Coding of NXP Contactless Card ICs

X: depends on the COS

Bit number	Hex Value	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
ISO/IEC 14443-3		RFU				Proprietary				UID size		RFU	Bit Frame Anti-collision				
MIFARE Plus 2K (4 Byte UID or 4 Byte RID)	00 04	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
MIFARE Plus EV1 2K (4 Byte UID or 4 Byte RID)	00 04	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
MIFARE Plus 4K (4 Byte UID or 4 Byte RID)	00 02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
MIFARE Plus EV1 4K (4 Byte UID or 4 Byte RID)	00 02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
MIFARE Plus 2K (7 Byte UID)	00 44	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
MIFARE Plus EV1 2K (7 Byte UID)	00 44	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
MIFARE Plus 4K (7 Byte UID)	00 42	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
MIFARE Plus EV1 4K (7 Byte UID)	00 42	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
MIFARE DESFire	03 44	0	0	0	0	0	0	1	1	0	1	0	0	0	1	0	0
MIFARE DESFire EV1	03 44	0	0	0	0	0	0	1	1	0	1	0	0	0	1	0	0
P3SR008	00 44	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0

2. [2] The 7 byte UID MIFARE Mini has bit 7 = 1, even if the 4 byte NUID mapping is enabled.
3. [3] The 7 byte UID MIFARE Classic 1K has bit 7 = 1, even if the 4 byte NUID mapping is enabled.
4. [4] The 7 byte UID MIFARE Classic 4K has bit 7 = 1, even if the 4 byte NUID mapping is enabled.

DOCUMENTATION PP6 : Principales requêtes SQL

Voici un résumé des commandes SQL courantes, regroupées par catégories, qui sont souvent utilisées pour gérer et manipuler les bases de données :

1. ***Commandes de gestion des bases de données (DDL - Data Definition Language)***

- **CREATE DATABASE** : Crée une nouvelle base de données.

```
CREATE DATABASE nom_de_la_base;
```

- **DROP DATABASE :** Supprime une base de données existante.

```
DROP DATABASE nom_de_la_base;
```

- **CREATE TABLE** : Crée une nouvelle table dans la base de données.

```
CREATE TABLE nomTable (
    id INT NOT NULL AUTO_INCREMENT,
    champ1 DOUBLE,
    champ2 VARCHAR(255),
    PRIMARY KEY(id)
);
```

- **DROP TABLE** : Supprime une table existante.

```
DROP TABLE nomTable;
```

- **ALTER TABLE** : Modifie la structure d'une table (ajouter, supprimer ou modifier des colonnes).

```
ALTER TABLE nomTable ADD champ3 TIMESTAMP;
```

3. ***Commandes de manipulation des données (DML - Data Manipulation Language)***

- **INSERT INTO** : Ajoute de nouvelles lignes dans une table.

```
INSERT INTO nomTable (champ1, champ2) VALUES (valeur1, valeur2);
```

- **UPDATE** : Modifie des lignes existantes dans une table.

```
UPDATE nomTable SET champ1 = nouvelleValeur WHERE id = 1;
```

- **DELETE** : Supprime des lignes d'une table.

```
DELETE FROM nomTable WHERE condition;
```

- **SELECT** : Récupère des données d'une ou plusieurs tables.

```
SELECT * FROM nomTable;  -- Pour sélectionner toutes les colonnes
SELECT champ1, champ2 FROM nomTable WHERE condition;
```

3. *Commandes de contrôle des données (DCL - Data Control Language)*

- **GRANT** : Accorde des permissions sur des objets de la base de données à un utilisateur.

```
GRANT SELECT, INSERT ON nomTable TO utilisateur;
```

- **REVOKE** : Retire des permissions précédemment accordées.

```
REVOKE INSERT ON nomTable FROM utilisateur;
```

4. *Commandes avancées*

- **JOIN :** Combine des lignes de deux ou plusieurs tables basées sur une condition commune.

```
SELECT * FROM table1 INNER JOIN table2 ON table1.id = table2.foreign_id;
```

- **GROUP BY** : Regroupe les résultats selon une ou plusieurs colonnes, souvent utilisé avec des fonctions d'agrégation comme COUNT, SUM, AVG.

```
SELECT champ1, COUNT(*) FROM nomTable GROUP BY champ1;
```

- **ORDER BY** : Trie les résultats d'une requête selon une ou plusieurs colonnes.

```
SELECT * FROM nomTable ORDER BY champ1 ASC;  -- ASC pour croissant, DESC pour décroissant
```

- **WHERE** : Filtre les résultats en fonction d'une condition.

```
SELECT * FROM nomTable WHERE champ1 = 'valeur';
```

5. *Fonctions d'agrégation*

- **COUNT()** : Compte le nombre de lignes.
- **SUM()** : Calcule la somme d'une colonne.
- **AVG()** : Calcule la moyenne d'une colonne.
- **MAX()** : Trouve la valeur maximale d'une colonne.
- **MIN()** : Trouve la valeur minimale d'une colonne.

6. *Autres commandes utiles*

- **DESCRIBE** : Affiche la structure d'une table, y compris les types de colonnes et les contraintes.

```
DESCRIBE nomTable;
```

- **ALTER USER** : Modifie un utilisateur existant dans la base de données.

```
ALTER USER 'utilisateur'@'localhost' IDENTIFIED BY 'nouveau_mot_de_passe';
```

DOCUMENTATION PP7 : SOAP

SOAP (abréviation de **Simple Object Access Protocol**) est une spécification de protocole de messagerie pour l'échange d'informations structurées dans la mise en œuvre de services web dans les réseaux informatiques. Son objectif est de fournir extensibilité, neutralité et indépendance. Il utilise l'XML Information Set pour son format de message et s'appuie sur des protocoles de couche application, le plus souvent Hypertext Transfer Protocol (HTTP) ou Simple Mail Transfer Protocol (SMTP), pour la négociation et la transmission des messages.

SOAP permet aux processus s'exécutant sur des systèmes d'exploitation disparates (comme Windows et Linux) de communiquer en utilisant Extensible Markup Language (XML). Étant donné que des protocoles Web comme HTTP sont installés et fonctionnent sur tous les systèmes d'exploitation, SOAP permet aux clients d'invoquer des services web et de recevoir des réponses indépendamment du langage et des plateformes.

Caractéristiques

SOAP fournit la couche de protocole de messagerie d'une pile de protocoles de services web. C'est un protocole basé sur XML constitué de trois parties :

- une enveloppe, qui définit la structure du message et comment le traiter
- un ensemble de règles d'encodage pour exprimer des instances de types de données définis par l'application
- une convention pour représenter les appels de procédure et les réponses.

SOAP possède trois caractéristiques majeures :

1. **extensibilité** (la sécurité et WS-Addressing font partie des extensions en cours de développement)
2. **neutralité** (SOAP peut fonctionner sur n'importe quel protocole tel que HTTP, SMTP, TCP, UDP ou JMS)
3. **indépendance** (SOAP permet tout modèle de programmation)

À titre d'exemple de ce que les procédures SOAP peuvent faire, une application peut envoyer une requête SOAP à un serveur qui a activé les services web comme une base de données de prix immobiliers avec les paramètres d'une recherche. Le serveur renvoie alors une réponse SOAP (un document formaté en XML contenant les données résultantes), par exemple, des prix, des emplacements, des caractéristiques. Étant donné que les données générées sont dans un format standardisé et interprétable par machine, l'application demandeuse peut ensuite les intégrer directement.

L'architecture SOAP se compose de plusieurs couches de spécifications pour :

- le format des messages
- les Message Exchange Patterns (MEP)
- les liaisons de protocoles de transport sous-jacents
- les modèles de traitement des messages
- l'extensibilité des protocoles

SOAP a évolué en tant que successeur de XML-RPC, bien qu'il emprunte sa neutralité de transport et d'interaction à Web Service Addressing et l'enveloppe/l'en-tête/le corps d'ailleurs (probablement de WDDX).

Terminologie SOAP

La spécification SOAP peut être largement définie comme étant constituée des trois composants conceptuels suivants :

- concepts de protocole
- concepts d'encapsulation
- concepts de réseau

Concepts d'encapsulation des données

- **Message SOAP** : Représente les informations échangées entre deux nœuds SOAP.
- **Enveloppe SOAP** : Comme son nom l'indique, c'est l'élément englobant d'un message XML qui l'identifie comme un message SOAP.
- **Bloc d'en-tête SOAP** : Un en-tête SOAP peut contenir plusieurs de ces blocs, chacun étant une unité de traitement distincte dans l'en-tête. En général, les informations de rôle SOAP sont utilisées pour cibler des nœuds sur le chemin. Un bloc d'en-tête est considéré comme étant ciblé sur un nœud SOAP si le rôle SOAP pour le bloc d'en-tête est le nom d'un rôle dans lequel le nœud SOAP fonctionne. (Exemple : Un bloc d'en-tête SOAP avec l'attribut de rôle ***ultimateReceiver*** est uniquement destiné au nœud de destination ayant ce rôle. Un en-tête avec un rôle next est destiné à chaque intermédiaire ainsi qu'au nœud de destination.)
- **En-tête SOAP** : Une collection d'un ou plusieurs blocs d'en-tête ciblés sur chaque récepteur SOAP.
- **Corps SOAP** : Contient le corps du message destiné au récepteur SOAP. L'interprétation et le traitement du corps SOAP sont définis par les blocs d'en-tête.
- **Erreur SOAP** : En cas d'échec de traitement d'un message SOAP par un nœud SOAP, les informations sur l'erreur sont ajoutées à l'élément d'erreur SOAP. Cet élément est contenu dans le corps SOAP en tant qu'élément enfant.

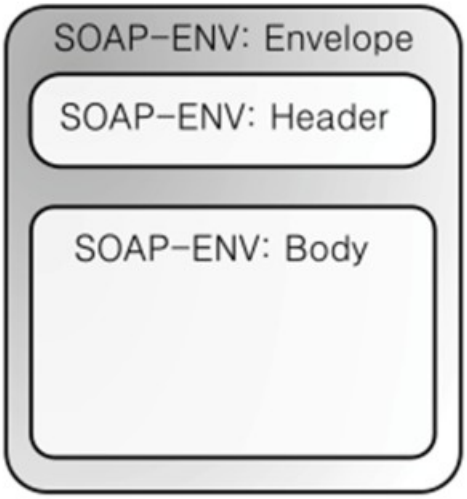
Concepts d'envoi et de réception de messages

- **Expéditeur SOAP** : Le nœud qui transmet un message SOAP.
- **Récepteur SOAP** : Le nœud qui reçoit un message SOAP (cela peut être un intermédiaire ou le nœud de destination).
- **Chemin de message SOAP** : Le chemin composé de tous les nœuds que le message SOAP a traversés pour atteindre le nœud de destination.
- **Expéditeur SOAP initial** : Le nœud qui a généré le message SOAP à transmettre. C'est la racine du chemin de message SOAP.
- **Intermédiaire SOAP** : Tous les nœuds entre l'origine du message SOAP et la destination prévue. Il traite les blocs d'en-tête SOAP qui lui sont destinés et transmet le message SOAP vers le récepteur SOAP final.
- **Récepteur SOAP ultime** : Le récepteur final du message SOAP. Ce nœud est responsable du traitement du corps du message et de tout bloc d'en-tête qui lui est destiné.

Spécification

La spécification SOAP définit le cadre de messagerie, qui comprend :

- Le modèle de traitement SOAP, qui définit les règles de traitement d'un message SOAP
- Le modèle d'extensibilité SOAP, qui définit les concepts des fonctionnalités et des modules SOAP
- Le cadre de liaison des protocoles sous-jacents, qui décrit les règles pour définir une liaison à un protocole sous-jacent pouvant être utilisé pour échanger des messages SOAP entre des nœuds SOAP
- La construction du message SOAP, qui définit la structure d'un message SOAP



1 SOAP Structure

Éléments constitutifs de SOAP

Un message SOAP est un document XML ordinaire contenant les éléments suivants :

- **Enveloppe** : Identifie le document XML en tant que message SOAP. (Obligatoire)
- **En-tête** : Contient des informations d'en-tête. (Facultatif)
- **Corps** : Contient les informations d'appel et de réponse. (Obligatoire)
- **Erreur** : Fournit des informations sur les erreurs survenues lors du traitement du message. (Facultatif)

Méthodes de transport

À la fois, SMTP et HTTP sont des protocoles de couche application valides utilisés pour le transport de SOAP, mais HTTP a gagné une plus large acceptation, car il fonctionne bien avec l'infrastructure Internet actuelle, notamment avec les pare-feu réseau. SOAP peut également être utilisé via **HTTPS** (qui est le même protocole que HTTP au niveau application, mais utilise un protocole de transport chiffré en dessous) avec une authentification simple ou mutuelle. C'est la méthode prônée par le WS-I pour assurer la sécurité des services web, comme indiqué dans le WS-I Basic Profile.

Ceci est un avantage majeur par rapport à d'autres protocoles distribués comme **GIOP/IIOP** ou **DCOM**, qui sont normalement filtrés par les pare-feu. SOAP sur **AMQP** est une autre possibilité prise en charge par certaines implémentations. SOAP a également un avantage sur DCOM en ce qu'il n'est pas affecté par les droits de sécurité configurés sur les machines qui nécessitent une connaissance à la fois des nœuds émetteurs et récepteurs. Cela permet à SOAP d'être faiblement couplé d'une manière qui n'est pas possible avec DCOM. Il existe également la norme **SOAP-over-UDP** de **OASIS**.

Exemple message (encapsulated in HTTP)

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 299
SOAPAction: "http://www.w3.org/2003/05/soap-envelope"

<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:m="http://www.example.org">
        <soap:Header>
        </soap:Header>
        <soap:Body>
                <m:GetStockPrice>
                        <m:StockName>GOOG</m:StockName>
                </m:GetStockPrice>
        </soap:Body>
</soap:Envelope>
```

DOCUMENTATION PP8 : Captures échanges

Dialogue 1 : Borne → Supervision

No.	Time	Source	Destination	Protocol	Length	Info
8	2.122297	192.168.0.241	192.168.0.102	TCP	66	8080 → 44112 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
9	2.122717	192.168.0.102	192.168.0.241	TCP	60	44112 → 8080 [ACK] Seq=1 Ack=1 Win=29200 Len=0
10	2.128202	192.168.0.102	192.168.0.241	HTTP/XML	1401	POST /steve/services/CentralSystemService HTTP/1.1
11	2.159196	192.168.0.241	192.168.0.102	TCP	800	8080 → 44112 [PSH, ACK] Seq=1 Ack=1348 Win=1051136 Len=746 [TCP segment of a reassembled
12	2.159415	192.168.0.241	192.168.0.102	HTTP/XML	54	HTTP/1.1 200 OK
13	2.160003	192.168.0.102	192.168.0.241	TCP	60	44112 → 8080 [ACK] Seq=1348 Ack=747 Win=32120 Len=0
14	2.163539	192.168.0.102	192.168.0.241	TCP	60	44112 → 8080 [FIN, ACK] Seq=1348 Ack=748 Win=32120 Len=0

```
Frame 10: 1401 bytes on wire (11208 bits), 1401 bytes captured (11208 bits) on interface 0
Ethernet II, Src: Telemech_42:86:06 (00:80:f4:42:86:06), Dst: LcfcHefe_be:9c:76 (c8:5b:76:be:9c:76)
Internet Protocol Version 4, Src: 192.168.0.102, Dst: 192.168.0.241
Transmission Control Protocol, Src Port: 44112, Dst Port: 8080, Seq: 1, Ack: 1, Len: 1347
Hypertext Transfer Protocol
eXtensible Markup Language
<?xml
    version="1.0"
    encoding="UTF-8"
    ?>
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope"
    xmlns:SOAP-ENC="http://www.w3.org/2003/05/soap-encoding"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:cp="urn://Ocpp/Cp/2012/06/"
    xmlns:chan="http://schemas.microsoft.com/ws/2005/02/duplex"
    xmlns:wsa5="http://www.w3.org/2005/08/addressing"
    xmlns:cs="urn://Ocpp/Cs/2012/06/">
  <SOAP-ENV:Header>
    <cs:chargeBoxIdentity>
      EV1234
      </cs:chargeBoxIdentity>
    <wsa5:MessageID>
      urn:uuid:d7870b21-ab67-47ba-81a8-eb745829e117
      </wsa5:MessageID>
    <wsa5:From>
      <wsa5:Address>
        http://192.168.0.102:8080/
        </wsa5:Address>
      </wsa5:From>
    <wsa5:ReplyTo>
      <wsa5:Address>
        http://www.w3.org/2005/08/addressing/anonymous
        </wsa5:Address>
      </wsa5:ReplyTo>
    <wsa5:To
      SOAP-ENV:mustUnderstand="true">
      http://192.168.0.241:8080/steve/services/CentralSystemService
      </wsa5:To>
    <wsa5:Action
      SOAP-ENV:mustUnderstand="true">
      /Authorize
      </wsa5:Action>
    </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <cs:authorizeRequest>
      <cs:idTag>
        0780BA2305625D
        </cs:idTag>
      </cs:authorizeRequest>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

Dialogue 1 : Supervision -> Borne

```
> Frame 12: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
> Ethernet II, Src: LcfcHefe_be:9c:76 (c8:5b:76:be:9c:76), Dst: Telemech_42:86:06 (00:80:f4:42:86:06)
> Internet Protocol Version 4, Src: 192.168.0.241, Dst: 192.168.0.102
> Transmission Control Protocol, Src Port: 8080, Dst Port: 44112, Seq: 747, Ack: 1348, Len: 0
> [2 Reassembled TCP Segments (746 bytes): #11(746), #12(0)]
> Hypertext Transfer Protocol
v eXtensible Markup Language
  v <soap:Envelope
      xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
    v <soap:Header>
      v <Action
          xmlns="http://www.w3.org/2005/08/addressing">
          /AuthorizeResponse
          </Action>
      v <MessageID
          xmlns="http://www.w3.org/2005/08/addressing">
          urn:uuid:a5f79918-bf4e-4018-aa3f-dad2f8d8ba19
          </MessageID>
      v <To
          xmlns="http://www.w3.org/2005/08/addressing">
          http://www.w3.org/2005/08/addressing/anonymous
          </To>
      v <RelatesTo
          xmlns="http://www.w3.org/2005/08/addressing">
          urn:uuid:d7870b21-ab67-47ba-81a8-eb745829e117
          </RelatesTo>
        </soap:Header>
    v <soap:Body>
      v <authorizeResponse
          xmlns="urn://Ocpp/Cs/2012/06/">
        v <idTagInfo>
          v <status>
              Invalid
              </status>
            </idTagInfo>
          </authorizeResponse>
        </soap:Body>
      </soap:Envelope>
```

Dialogue 2 : Borne -> Supervision

No.	Time	Source	Destination	Protocol	Length	Info
51	7.522515	192.168.0.241	192.168.0.102	TCP	66	8080 → 44118 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=14
52	7.522884	192.168.0.102	192.168.0.241	TCP	60	44118 → 8080 [ACK] Seq=1 Ack=1 Win=29200 Len=0
53	7.525993	192.168.0.102	192.168.0.241	HTTP/XML	1401	POST /steve/services/CentralSystemService HTTP/1.1
54	7.561072	192.168.0.241	192.168.0.102	TCP	850	8080 → 44118 [PSH, ACK] Seq=1 Ack=1348 Win=1051136 Len=796
55	7.561372	192.168.0.241	192.168.0.102	HTTP/XML	54	HTTP/1.1 200 OK

```
> Frame 53: 1401 bytes on wire (11208 bits), 1401 bytes captured (11208 bits) on interface 0
> Ethernet II, Src: Telemech_42:86:06 (00:80:f4:42:86:06), Dst: LcfcHefe_be:9c:76 (c8:5b:76:be:9c:76)
> Internet Protocol Version 4, Src: 192.168.0.102, Dst: 192.168.0.241
> Transmission Control Protocol, Src Port: 44118, Dst Port: 8080, Seq: 1, Ack: 1, Len: 1347
> Hypertext Transfer Protocol
v eXtensible Markup Language
  v <?xml
      version="1.0"
      encoding="UTF-8"
      ?>
  v <SOAP-ENV:Envelope
      xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope"
      xmlns:SOAP-ENC="http://www.w3.org/2003/05/soap-encoding"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:cp="urn://Ocpp/Cp/2012/06/"
      xmlns:chan="http://schemas.microsoft.com/ws/2005/02/duplex"
      xmlns:wsa5="http://www.w3.org/2005/08/addressing"
      xmlns:cs="urn://Ocpp/Cs/2012/06/">
    v <SOAP-ENV:Header>
      v <cs:chargeBoxIdentity>
          EV1234
          </cs:chargeBoxIdentity>
      v <wsa5:MessageID>
          urn:uuid:79c19bb1-af01-48c1-9981-43dc34a055fa
          </wsa5:MessageID>
      v <wsa5:From>
        v <wsa5:Address>
            http://192.168.0.102:8080/
            </wsa5:Address>
          </wsa5:From>
      v <wsa5:ReplyTo>
        v <wsa5:Address>
            http://www.w3.org/2005/08/addressing/anonymous
            </wsa5:Address>
          </wsa5:ReplyTo>
      v <wsa5:To
          SOAP-ENV:mustUnderstand="true">
          http://192.168.0.241:8080/steve/services/CentralSystemService
          </wsa5:To>
      v <wsa5:Action
          SOAP-ENV:mustUnderstand="true">
          /Authorize
          </wsa5:Action>
        </SOAP-ENV:Header>
    v <SOAP-ENV:Body>
      v <cs:authorizeRequest>
        v <cs:idTag>
            0780BA23056776
            </cs:idTag>
          </cs:authorizeRequest>
        </SOAP-ENV:Body>
      </SOAP-ENV:Envelope>
```

Dialogue 2 : Supervision -> Borne

```
> Frame 55: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
> Ethernet II, Src: LcfcHefe_be:9c:76 (c8:5b:76:be:9c:76), Dst: Telemech_42:86:06 (00:80:f4:42:86:06)
> Internet Protocol Version 4, Src: 192.168.0.241, Dst: 192.168.0.102
> Transmission Control Protocol, Src Port: 8080, Dst Port: 44118, Seq: 797, Ack: 1348, Len: 0
> [2 Reassembled TCP Segments (796 bytes): #54(796), #55(0)]
> Hypertext Transfer Protocol
v eXtensible Markup Language
  v <soap:Envelope
      xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
    v <soap:Header>
      v <Action
          xmlns="http://www.w3.org/2005/08/addressing">
          /AuthorizeResponse
          </Action>
      v <MessageID
          xmlns="http://www.w3.org/2005/08/addressing">
          urn:uuid:4d4500bf-8234-4e73-b3eb-070db29fed7a
          </MessageID>
      v <To
          xmlns="http://www.w3.org/2005/08/addressing">
          http://www.w3.org/2005/08/addressing/anonymous
          </To>
      v <RelatesTo
          xmlns="http://www.w3.org/2005/08/addressing">
          urn:uuid:79c19bb1-af01-48c1-9981-43dc34a055fa
          </RelatesTo>
        </soap:Header>
    v <soap:Body>
      v <authorizeResponse
          xmlns="urn://Ocpp/Cs/2012/06/">
        v <idTagInfo>
          v <status>
              Accepted
              </status>
          v <expiryDate>
              2019-08-13T16:53:05.178Z
              </expiryDate>
            </idTagInfo>
          </authorizeResponse>
        </soap:Body>
</soap:Envelope>
```

DOCUMENTATION PP9 : Extrait du fichier WSDL

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
        xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
        xmlns:s="http://www.w3.org/2001/XMLSchema"
        xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
        xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
        xmlns:tns="urn://Ocpp/Cs/2012/06/"
        targetNamespace="urn://Ocpp/Cs/2012/06/"
        xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
        xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
        xmlns:wsa="http://www.w3.org/2005/08/addressing"
        xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<wsdl:types>
<s:schema targetNamespace="urn://Ocpp/Cs/2012/06/" elementFormDefault="qualified">

<!-- Begin of types shared with ChargePointService -->
<s:simpleType name="IdToken">
    <s:annotation>
        <s:documentation>Type of string defining identification token, e.g. RFID or credit card number. To be treated as case insensitive.</s:documentation>
    </s:annotation>
    <s:restriction base="s:string">
        <s:maxLength value="20"/>
    </s:restriction>
</s:simpleType>
```

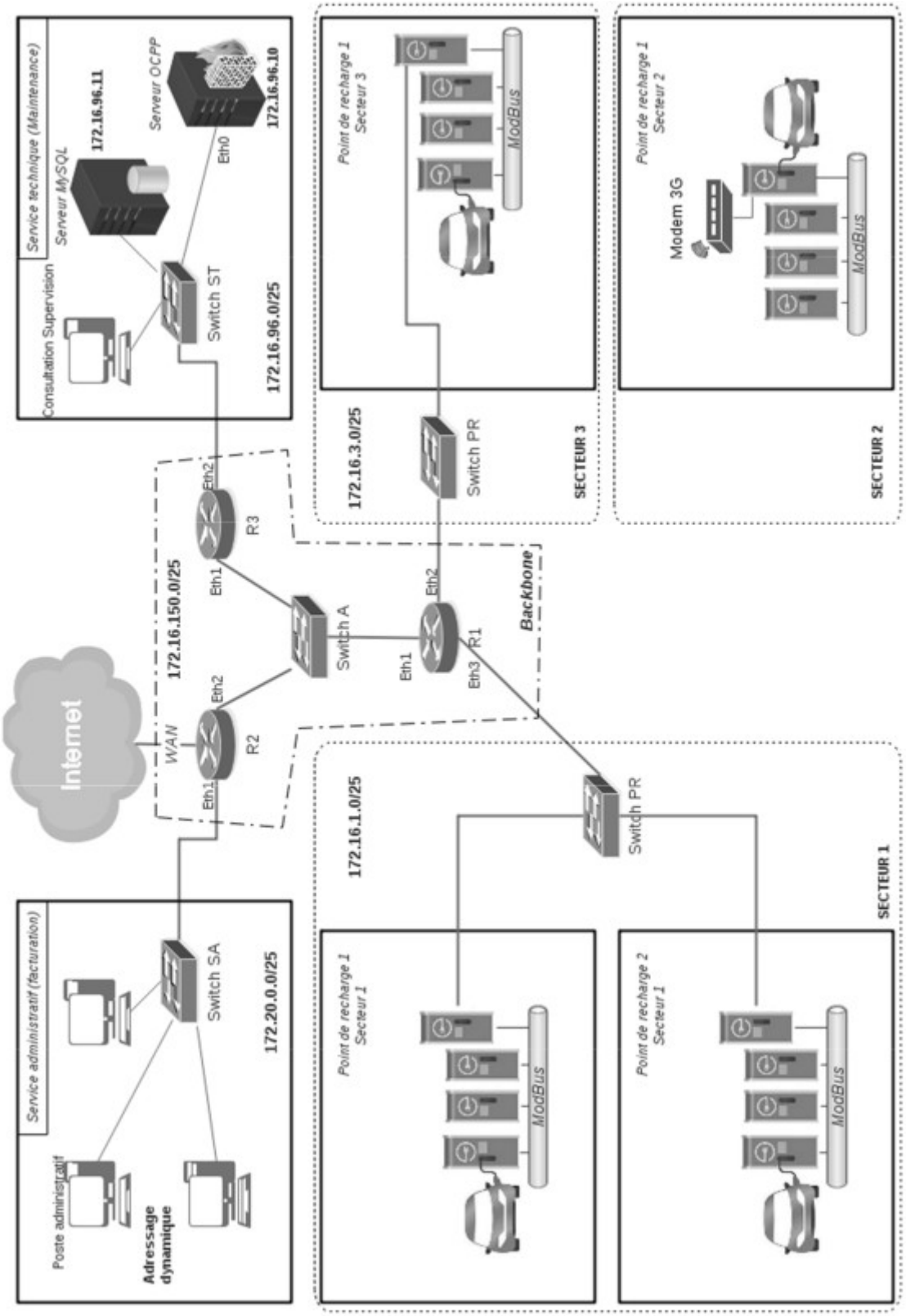
```
<s:simpleType name="AuthorizationStatus">
        <s:annotation>
                <s:documentation>Defines the authorization-status-value</s:documentation>
        </s:annotation>
        <s:restriction base="s:string">
                <s:enumeration value="Accepted"/>
                <s:enumeration value="Blocked"/>
                <s:enumeration value="Expired"/>
                <s:enumeration value="Invalid"/>
                <s:enumeration value="ConcurrentTx"/>
        </s:restriction>
</s:simpleType>
<s:complexType name="IdTagInfo">
        <s:sequence>
                <s:element name="status" type="tns:AuthorizationStatus" minOccurs="1" maxOccurs="1"/>
                <s:element name="expiryDate" type="s:dateTime" minOccurs="0" maxOccurs="1"/>
                <s:element name="parentIdTag" type="tns:IdToken" minOccurs="0" maxOccurs="1"/>
        </s:sequence>
</s:complexType>
<!-- End of types shared with ChargePointService -->
<s:simpleType name="ChargeBoxSerialNumber">
        <s:annotation>
                <s:documentation>String type of max 25 chars that is to be treated as case insensitive.</s:documentation>
        </s:annotation>
        <s:restriction base="s:string">
                <s:maxLength value="25"/>
        </s:restriction>
</s:simpleType>
```

```
<s:simpleType name="ChargePointModel">
        <s:annotation>
                <s:documentation>String type of max 20 chars that is to be treated as case insensitive.</s:documentation>
        </s:annotation>
        <s:restriction base="s:string">
                <s:maxLength value="20"/>
        </s:restriction>
</s:simpleType>
<s:simpleType name="ChargePointSerialNumber">
        <s:annotation>
                <s:documentation>String type of max 25 chars that is to be treated as case insensitive.</s:documentation>
        </s:annotation>
        <s:restriction base="s:string">
                <s:maxLength value="25"/>
        </s:restriction>
</s:simpleType>
<s:simpleType name="ChargePointVendor">
        <s:annotation>
                <s:documentation>String type of max 20 chars that is to be treated as case insensitive.</s:documentation>
        </s:annotation>
        <s:restriction base="s:string">
        <s:maxLength value="20"/>
                </s:restriction>
</s:simpleType>
<s:simpleType name="FirmwareVersion">
        <s:annotation>
                <s:documentation>String type of max 50 chars that is to be treated as case insensitive.</s:documentation>
        </s:annotation>
```

```
        <s:restriction base="s:string">
            <s:maxLength value="50"/>
        </s:restriction>
</s:simpleType>
<s:simpleType name="IccidString">
        <s:annotation>
            <s:documentation>String type of max 20 chars that is to be treated as case insensitive.</s:documentation>
        </s:annotation>
        <s:restriction base="s:string">
            <s:maxLength value="20"/>
        </s:restriction>
</s:simpleType>
<s:simpleType name="ImsiString">
        <s:annotation>
            <s:documentation>String type of max 20 chars that is to be treated as case insensitive.</s:documentation>
        </s:annotation>
        <s:restriction base="s:string">
            <s:maxLength value="20"/>
        </s:restriction>
</s:simpleType>
<s:simpleType name="MeterSerialNumber">
        <s:annotation>
            <s:documentation>String type of max 25 chars that is to be treated as case insensitive.</s:documentation>
        </s:annotation>
        <s:restriction base="s:string">
            <s:maxLength value="25"/>
        </s:restriction>
</s:simpleType>
```

```
<s:simpleType name="MeterType">
        <s:annotation>
                <s:documentation>String type of max 25 chars that is to be treated as case insensitive.</s:documentation>
        </s:annotation>
        <s:restriction base="s:string">
                <s:maxLength value="25"/>
        </s:restriction>
</s:simpleType>
```

DOCUMENTATION PP10 : Infrastructure réseau



DOCUMENTATION PP11 : iptables

iptables - outil d'administration pour le filtrage de paquets IPv4 et le NAT

DESCRIPTION

- **iptables** est utilisé pour mettre en place, maintenir et inspecter les tables des règles de filtrage des paquets IP du noyau Linux. Différentes tables peuvent être définies. Chaque table contient plusieurs chaînes prédéfinies et peut aussi contenir des chaînes définies par l'utilisateur.
  Chaque chaîne est une liste de règles que peuvent vérifier un ensemble de paquets ; dans ce cas, on dit qu'on cherche à établir une correspondance avec la règle. Chaque règle détermine ce qui doit être fait avec un paquet qui correspond. Cette action est appelée une «cible», qui peut être un saut vers une chaîne définie par l'utilisateur dans la même table.
- **filter** : C'est la table par défaut (si l'option -t est omise). Elle contient les chaînes prédéfinies INPUT (pour les paquets entrants dans la machine), FORWARD (pour les paquets routés à travers la machine) et OUTPUT (pour les paquets générés localement).

COMMANDES

Ces options précisent une action particulière à accomplir. Une seule option peut être indiquée sur la ligne de commande, sauf indication contraire. Pour tous les noms en version longue des commandes et des options, vous avez le droit d'utiliser un nombre restreint de lettres du moment qu' iptables peut identifier chaque commande sans ambiguïté.

-A, --append *chaîne règle*

Ajoute une ou plusieurs règles à la fin de la chaîne sélectionnée. Lorsque les noms source et/ou destination désignent plus d'une adresse, une règle sera ajoutée pour chaque combinaison d'adresses possible.

-**D, --delete** *chaîne règle*

Les chaînes standards :

FORWARD : chaîne désignant les paquets désirant traverser le pare-feu
INPUT : chaîne désignant les paquets s’adressant au pare-feu lui-même
OUTPUT : chaîne désignant les paquets expédiés par le pare-feu lui-même
PREROUTING : chaîne désignant les paquets attendant d’être routés
POSTROUTING : chaîne désignant les paquets venant d’être routés

PARAMÈTRES

Les paramètres suivants composent une spécification de règle (quand ils sont utilisés dans les commandes add, delete, insert, replace et append).

-p, --protocol [!] protocole
Protocole de la règle ou du paquet à vérifier. Le protocole spécifié est l'un des suivants : tcp, udp, icmp, ftp, ssh ou all, ou bien sous forme d'une valeur numérique, représentant un de ces protocoles ou un protocole différent. Un nom de protocole issu du fichier /etc/protocols est aussi autorisé. Un «!» avant le protocole inverse le test. La valeur zéro est équivalente à *all*. Le protocole *all* correspond à tous les protocoles ; c'est aussi la valeur par défaut lorsque cette option est omise.

-s, --source [!] adresse[/masque]
Spécification de la source. L'adresse peut être un nom de réseau, un nom d'hôte (attention : spécifier un nom à résoudre avec une requête distante de type DNS est vraiment une mauvaise idée), une adresse de réseau IP (avec /masque) ou une simple adresse IP. Le masque peut être un masque de réseau ou un nombre entier spécifiant le nombre de bits égaux à 1 dans la partie gauche du masque de réseau (bits de poids fort). Par conséquent, un masque de 24 est équivalent à 255.255.255.0. Un «!» avant la spécification d'adresse inverse la sélection d'adresse. L'option –src est un synonyme de --source.

-d, --destination [!] adresse[/masque]
Spécification de la destination. Voir la description du paramètre -s (source) pour une description détaillée de la syntaxe. L'option --dst est un synonyme de –destination.

-j, --jump cible
Ceci détermine la cible de la règle ; c'est-à-dire ce qu'il faut faire si le paquet correspond à la règle. La cible peut être une chaîne définie par l'utilisateur (autre que celle dans laquelle se situe cette règle), une des cibles prédéfinies qui décide immédiatement du sort du paquet, ou une extension (voir EXTENSIONS ci-dessous). Si cette option est omise dans une règle, la correspondance d'un paquet avec la règle n'aura aucun effet sur le sort du paquet, mais les compteurs seront incrémentés.

-i, --in-interface [!] [nom]
Nom de l'interface qui reçoit les paquets (seulement pour les paquets passant par les chaînes **INPUT**, **FORWARD** et **PREROUTING**). Lorsqu'un «!» est utilisé avant le nom d'interface, la sélection est inversée. Si le nom de l'interface se termine par un «+», il désigne toutes les interfaces commençant par ce nom. Si cette option est omise, toutes les interfaces réseau sont désignées.

-o, --out-interface [!] [nom]

Nom de l'interface qui envoie les paquets (seulement pour les paquets passant par les chaînes **FORWARD**, **OUTPUT** et **POSTROUTING**). Lorsqu'un «!» est utilisé avant le nom d'interface, la sélection est inversée. Si le nom de l'interface se termine par un «+», il désigne toutes les interfaces commençant par ce nom. Si cette option est omise, toutes les interfaces réseau sont désignées.

CIBLES

Une règle de pare-feu spécifie des critères de correspondance pour un paquet et une cible. Si le paquet correspond, la règle suivante est déterminée par la valeur de la cible, qui peut être une des valeurs spéciales suivantes : *ACCEPT, DROP*,…

ACCEPT signifie que le paquet est autorisé à passer

DROP signifie que le paquet est rejeté ou détruit.

Exemples :

iptables -A FORWARD -i eth0 -o eth1 -p ftp -j ACCEPT

ajoute une règle qui autorise les paquets ftp à traverser la machine s'il rentre par l'interface réseau eth0 et sort par l'interface réseau eth1.

iptables -A INPUT -s 192.168.0.0/24 -i eth0 -j DROP

ajoute une règle qui rejette tous les paquets provenant des machines du réseau 192.168.0.0/24 entrants par l'interface réseau eth0 et destinés à cette machine.

iptables -A INPUT -i eth0 -p tcp -j ACCEPT

ajoute une règle qui autorise tous les paquets TCP entrants par l'interface eth0 et destinés à cette machine.

DOCUMENTATION PP12 : La série Cisco 890

La **série Cisco 890** est une gamme de routeurs conçus pour les petites entreprises, les bureaux distants et les environnements qui nécessitent une connectivité fiable et sécurisée. Voici un aperçu des principales caractéristiques et avantages de cette série :

Caractéristiques principales de la série Cisco 890 :

1. **Connectivité WAN et LAN** :
    - Les routeurs de la série Cisco 890 prennent en charge plusieurs options de connectivité WAN, y compris **Ethernet**, **xDSL** (ADSL2+, VDSL2), **3G/4G LTE** et **T1/E1**. Cela permet une flexibilité dans le choix de la connexion Internet.
    - Ils incluent également des ports **Gigabit Ethernet** pour les connexions LAN, permettant des vitesses élevées et un réseau local performant.
2. **Sécurité intégrée** :
    - **Pare-feu intégré** : Les routeurs de cette série intègrent des fonctionnalités de sécurité robustes comme un pare-feu avec des listes de contrôle d'accès (ACL) pour protéger le réseau contre les menaces extérieures.
    - **VPN** : Supporte des connexions VPN (IPsec, SSL) pour sécuriser les communications entre les sites distants.
    - **Cisco IOS Security** : Avec les services de sécurité Cisco IOS, ces routeurs offrent une protection contre les intrusions, le cryptage des données et la protection des applications.
3. **Haute disponibilité et fiabilité** :
    - La série Cisco 890 est conçue pour assurer une connectivité Internet continue avec des options de basculement automatique entre plusieurs connexions WAN (par exemple, passer de l'Ethernet à la 3G/4G LTE en cas de défaillance).
    - **Qualité de Service (QoS)** : Les routeurs incluent des fonctionnalités de QoS avancées pour garantir une gestion efficace du trafic réseau, priorisant les applications critiques telles que la voix et la vidéo sur IP.
4. **Gestion avancée et facilité d'administration** :
    - Support de **Cisco Configuration Professional (CCP)**, qui permet une gestion simplifiée de l'appareil via une interface graphique conviviale.

- Prend en charge **SNMP** et d'autres outils de gestion réseau pour une surveillance et un contrôle efficaces.

5. **Performance et services intégrés** :
   - Ces routeurs offrent des **services unifiés** tels que le routage, la commutation, la sécurité, la mobilité, et le sans-fil, combinés dans un seul appareil.
   - Ils sont idéaux pour les environnements nécessitant une **connectivité haut débit**, la **convergence des services voix et données**, ainsi que des capacités de **surveillance réseau**.

Feature	Specification
Cisco IOS Software: Advanced IP Features Set (Default)	
IP and IP services	• Routing Information Protocol Versions 1 and 2 (RIPv1 and RIPv2) • Generic Routing Encapsulation (GRE) and Multipoint GRE (MGRE) • Cisco Express Forwarding • Standard 802.1d Spanning Tree Protocol • Layer 2 Tunneling Protocol (L2TP) • Layer 2 Tunneling Protocol Version 3 (L2TPv3) • Network Address Translation (NAT) • Dynamic Host Configuration Protocol (DHCP) server, relay, and client • Dynamic Domain Name System (DNS) • DNS Proxy • DNS Spoofing • Access control Lists (ACLs) • IPv4 and IPv6 Multicast • Open Shortest Path First (OSPF) • Border Gateway Protocol (BGP) • Performance Routing (PfR) • Enhanced Interior Gateway Routing Protocol (EIGRP) • Virtual Route Forwarding (VRF) Lite • Next Hop Resolution Protocol (NHRP) • Bidirectional Forwarding Detection (BFD)
xDSL	• True Multimode VDSL2 and ADSL2+ over Annex A, B, J, and M including traditional G.DMT and T1.413 • World-class interoperability with industry-standard DSL access multiplexer (DSLAM) chipsets • Highest field reliability with Impulse Noise Protection over REIN/SHINE, Extended INP-Delay, G.INP, Physical Layer Retransmission, SRA, and Bitswap • VDSL2 Persistent Storage Device (PSD) profiles up to 17a/b with support for Spectral Shaping • VDSL2 Vectoring to offer blazing fiber speeds over copper • Support for 4-pair multimode G.SHDSL; that is, ATM and EFM • Remote management with TR069 and CWMP • Investment protection with GE and SFP for future fiber that could replace xDSL deployment
Switch features	• Auto Media Device In/Media Device Cross Over (MDI-MDX) • 25 802.1Q VLANs • MAC filtering • Four-port 802.3af and Cisco compliant PoE • Switched Port Analyzer (SPAN) • Storm Control • Smart ports • Secure MAC address • Internet Group Management Protocol Version 3 (IGMPv3) snooping • 802.1x

Modèles typiques :

Les modèles dans la série Cisco 890 varient selon les options de connectivité et de performance. Certains modèles populaires incluent :

- **Cisco 891** : Un routeur Ethernet avec 8 ports Gigabit Ethernet.
- **Cisco 892FSP** : Ajoute des ports SFP pour la connectivité fibre optique.
- **Cisco 897VAG** : Intègre un modem xDSL et une connectivité 4G LTE.

Product Part Number	Product Description
Integrated Services Routers	
C892FSP-K9	Cisco 892FSP Gigabit Ethernet security router with SFP
C896VA-K9	Cisco 896VA Gigabit Ethernet security router with SFP and VDSL/ADSL2+ Annex B
C897VA-K9	Cisco 897VA Gigabit Ethernet security router with SFP and VDSL/ADSL2+ Annex A
C897VAW-A-K9	Cisco 897VA Gigabit Ethernet security router with SFP and VDSL/ADSL2+ Annex A with Wireless
C897VAW-E-K9	Cisco 897VA Gigabit Ethernet security router with SFP and VDSL/ADSL2+ Annex A with Wireless
C897VA-M-K9	Cisco 897VA Gigabit Ethernet security router with SFP and VDSL/ADSL2+ Annex M
C897VAM-W-E-K9	Cisco 897VA Gigabit Ethernet security router with SFP and VDSL/ADSL2+ Annex M with Wireless
C897VAB-K9	Cisco 897VA Gigabit Ethernet security router with SFP and VDSL2/ADSL2+ Bonding over POTS
C898EA-K9	Cisco 898EA Gigabit Ethernet security router with SFP and 4 channel multimode G.SHDSL (EFM/ATM)
C891F-K9	Cisco 891F Gigabit Ethernet security router with SFP
C891-24X/K9	Cisco 891 Gigabit Ethernet security router with SFP and 24-ports Ethernet Switch
C891FW-A-K9	Cisco 891F Gigabit Ethernet security router with SFP and Dual Radio 802.11n Wifi for FCC -A domain
C891FW-E-K9	Cisco 891F Gigabit Ethernet security router with SFP and Dual Radio 802.11n Wifi for ETSI -E domain
Cisco 892FSP is supported only on Cisco IOS Software Release 15.2(4)M and later Cisco 896, 897, 898EA is supported only on Cisco IOS Software Release 15.2(4)M1 and later Cisco 891F is supported only on Cisco IOS Software Release 15.3(3)M2, 15.4(1)T and later C897VAB is supported only on Cisco IOS Software Release 15.4(3)M1 and later C891-24X is supported only on Cisco IOS Software Release 15.5(1)T and later	

Avantages :

- **Sécurité et fiabilité** pour les petites et moyennes entreprises.
- **Gestion simple** grâce aux outils Cisco et à l'interface graphique.
- **Options de connectivité flexibles**, avec plusieurs interfaces WAN/LAN.
- **Services intégrés** pour la convergence des données, voix et vidéo.

DOCUMENTATION PP13 : Schéma entités – relations incomplet de la BDD de la supervision.