N° d'ordre :

الجمهورية الجزائرية الديمقراطية الشعبية

République **A**lgérienne **D**émocratique et **P**opulaire

Ministère de l'**E**nseignement **S**upérieur et de la **R**echerche **S**cientifique

Université Mohamed SeddikBenyahia–Jijel

Faculté des sciences exactes et informatique

Département Informatique

Filière : Informatique

Spécialité : SIAD

Présenté Par **Kemmouche Meriem**

MEMOIRE

de fin d'études pour l'obtention du diplôme de

MASTER EN INFORMATIQUE

Intitulé

Un Framework de modélisation des aspects de sécurité dans un système d'internet des objets

Encadreur: **Belghiat Issam**

Devant le jury composé de :

Président :Yahiaoui Abdelbaki

Examinatrice : Boudebza Souad

REMERCIEMENT

Je remercie Dieu le tout puissant de m'avoir donné la santé et la volonté d'entamer et de terminer ce mémoire.

Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide et l'encadrement de ***Mr Belghiat Issam****, je le remercie pour la qualité de son encadrement exceptionnel, pour sa patience, sa rigueur et sa disponibilité durant ma préparation de ce mémoire.*

Qu'il trouve ici le témoignage de ma profonde gratitude. Je voudrais également remercier ***les membres du jury*** *pour avoir accepté d'évaluer ce travail, ainsi que* ***le personnel et les enseignants de la faculté des sciences exactes****.*

Je tiens aussi à remercier mes enseignants qui m'ont initié aux valeurs authentiques, en signe d'un profond respect.

Et enfin, à toute personne qui m'a encouragé et m'a tendu la main aux moments les plus durs.

Merci à vous tous

Dédicaces

À mes chers parents : ma source de force et d'inspiration.

À ma petite famille :

Cher mari « Fares » qui était toujours à mes côtés

Mes adorables filles : « Safa Miral » et « Arwa »

La lumière de ma vie.

À tout membre de ma grande famille du plus petit au plus vieux.

À tous mes collègues de la faculté des sciences exactes qui m'ont encouragé tout au long de ma préparation de ce travail.

Table des matières

CHAPITRE I :L'INTERNET DES OBJETS

Chapitre II: modélisation des aspects de sécurité IoT en UML

Chapitre III : IoTsec

Tables des figures

Abstract

The Internet has established itself in many areas as an essential infrastructure for individuals, companies and institutions. It is expected to enable the interaction of an increasing number of objects with each other or with ourselves; hence the term Internet of things (IoT) which is a system of interconnected computing devices that can collect and transfer data over a wireless network without human intervention.As a result, sensitive personal data of users are at risk of being lost or falsified. So, data security is considered as the big challenge, and represents an extremely wide area of expertise. due to the complexity of IoT.

In this work, we propose to develop a framework for modeling the security aspects of IoT systems. A UML Profile designed for this purpose called IoTsec is used. Several diagrams (use case diagram, class diagram, deployment diagram, sequence diagram and activity diagram) of this profile have been chosen to be incorporated in the framework, the goal is to provide a modeling of several views of IoT systems. Meta-modeling is used to generate an integrated modeling environment. Papyrus is adopted to implement our framework.

Keyword: IoT, modeling, verification, UML, MDA, security.

Résumé

L'internet s'est imposé dans de nombreux domaines comme une infrastructure essentielle pour les individus, les entreprises et les institutions. Il devrait permettre l'interaction d'un nombre croissant d'objets entre eux ou avec nous-mêmes ;on parle donc du terme Internet des objets (IdO) « Internet of thingsIoT » qui est un système d'appareils informatiques interconnectés qui peuvent collecter et transférer des données sur un réseau sans fil sans intervention humaine. Par conséquent, les données personnelles sensibles des utilisateurs risquent d'être perdues ou falsifiées. Alors, la sécurité des données est considérée comme le grand défi, être présente un domaine d'expertise extrêmement vaste. En raison de la complexité de l'IoT.

Dans ce travail, nous proposons de développer un Framework pour la modélisation des aspects de sécurité des systèmes IoT. Un Profil UML conçu pour ce fin appelé IoTsec est utilisé. Plusieurs diagrammes (diagramme de cas d'utilisation, diagramme de classe, de déploiement, de séquences et diagramme d'activité) de ce profil ont été choisis pour être incorporés dans le framework, le but est de fournir une modélisation de plusieurs vues des systèmes IoT. La métamodélisation est utilisée, elle permet de générer un environnement de modélisation intégré. Papyrus est adopté pour implémenter notre Framework.

Mot clé : IoT, modélisation, vérification, UML, MDA, sécurité.

ملخص

أثبت الإنترنت وجوده في العديد من المجالات كبنية تحتية أساسية للأفراد والشركات والمؤسسات. يسمح بتفاعل عدد متزايد من الأشياء مع بعضها البعض أو مع أنفسنا، لذلك نتحدث عن مصطلح إنترنت الأشياء IOT ، وهو نظام من أجهزة الحوسبة المترابطة التي يمكنها جمع البيانات ونقلها عبر شبكة لاسلكية دون تدخل بشري. نتيجة لذلك، تتعرض البيانات الشخصية الحساسة للمستخدمين لخطر الضياع أو العبث بها. لذلك، يعتبر أمن البيانات التحدي الكبير، حيث يمثل مجال خبرة واسع للغاية. بسبب تعقيد إنترنت الأشياء.

في هذا العمل، نقترح تطوير إطار عمل لنمذجة الجوانب الأمنية لأنظمة إنترنت الأشياء. يتم. اختيار العديد من المخططات لهذا الغرض (امخطط الحالة ، ومخطط الفصل ، ومخطط النشر ، ومخطط التسلسل ، ومخطط النشاط) ليتم دمجها في إطار العمل ، والهدف هو توفير نمذجة للعديد من وجهات النظر لأنظمة إنترنت الأشياء.تستخدم النمذجة الوصفية ، فهي تسمح بإنشاء بيئة نمذجة متكاملة تم اعتماد Papyrus لتنفيذ إطار عملنا.

الكلمات المفتاحية: إنترنت الأشياء، النمذجة ،التحقق ، UML، MDA ، الأمن.

.

Introduction générale

Introduction générale

l'internet se développe et s'améliore continuellement à cause du progrès technologique. Ainsi, le nombre d'appareils connectés augmente au fur et à mesure, passant de milliards à bientôt centaines de milliards, donc Internet se transforme progressivement en un réseau étendu, appelé IoT « Internet des objets » ou « *Internet of things* » qui représente une évolution majeure dans le domaine de la technologie d'information et qui domine et règne de plus en plus sur le marché des systèmes informatiques.

Les appareils de l'IoT sont limités en mémoire, en capacité de calcul et en énergie, et disposent de moyens de communication peu fiables, ce qui les rend vulnérables à des attaques variées. Afin de garantir la sécurité et lutter contre ces menaces dans l'IoT, des solutions de sécurité robustes doivent être considérées.

Les problèmes de sécurité de l'IoT doivent être traités dans la phase de conception lors du développement d'un système IoT, afin d'éviter toutes sortes de défauts de sécurité qui peuvent émerger et minimiser les dégâts si on en a.

après l'approche objet des années 80 et un bref passage de l'approche aspect, l'ingénierie logicielle s'oriente aujourd'hui vers l'ingénierie dirigée par les modèles (IDM), dont l'enjeu s'agit principalement d'augmenter le niveau d'abstraction des développements en permettant aux développeurs de se concentrer sur leurs préoccupations à l'aide des langages spécifiques à leurs domaines.

L'UML (Unified Modeling Language) s'est imposé comme le langage de modélisation le plus utilisé dans l'industrie et dans le milieu universitaire. Afin d'aider et d'orienter les concepteurs des systèmes IoT.

Dans ce travail, nous allons proposer une approche pour la modélisation des aspects de sécurité dans un système IoT. Nous avons opté pour l'utilisation d'une extension UML appelée IoTsec pour modéliser les aspects de sécurité d'un système IoT. Après, un passage vers une méthode formelle est envisagé afin de permettre la vérification de ces aspects.

L'objectif de ce travail est de fournir un Framework complet pour la modélisation des aspects de sécurité dans un système IoT. Il sera composé de plusieurs diagrammes IoTsec qui permettent la modélisation de différentes vues du système. En plus, une formalisation de ces diagrammes par une méthode formelle est envisagée afin de permettre leurs analyse et vérification. L'ensemble de l'approche est implémenté en utilisant la métamodélisation et la transformation de modèle. Papyrus est l'outil adopté pour réaliser nos idées. Il permet de créer des représentations visuelles pour les métas-classes des profilés (mécanisme des stéréotypes UML) en réutilisant la syntaxe concrète d'UML.

L'objectif de ce travail est de fournir un Framework complet pour la modélisation des aspects de sécurité dans un système IoT.de plus une formalisation de ce Framework par une méthode formelle est envisagée afin de permettre l'analyse et la vérification des diagrammes qui y composent.

I. Chapitre 1 : L'internet des objets

1 Introduction

Internet est devenu en quelques années le vecteur principal de diffusion de l'information, il s'est imposé dans de nombreux domaines comme une infrastructure essentielle pour les individus, les entreprises et les institutions. Toutefois, ses capacités d'extension, au-delà des seuls ordinateurs et terminaux mobiles, sont encore considérables, car il devrait permettre l'interaction d'un nombre croissant d'objets entre eux ou avec nous-mêmes. Internet se transforme progressivement en un réseau étendu, appelé « Internet des objets » « *Internet of things IoT* » qui représente une évolution majeure dans le domaine de la technologie d'information et qui domine et règne de plus en plus sur le marché des systèmes informatiques, ce réseau, relie plusieurs milliards d'êtres humains, mais aussi des dizaines de milliards d'objets (1).

La grande puissance de l'IoT repose sur le fait que ses objets communiquent, analysent, traitent et gèrent des données d'une manière autonome et sans aucune intervention humaine.

Dans ce chapitre, on va explorer les notions de base sur ce concept de l'internet des objets, passant par son architecture et ses domaines d'application, et finalisant par détailler le sujet de la sécurité l'IoT.

2 Historique de L'IOT

2.1 Évolution du Web et L'internet

Selon (2), le Web est passé par plusieurs phases distinctes :

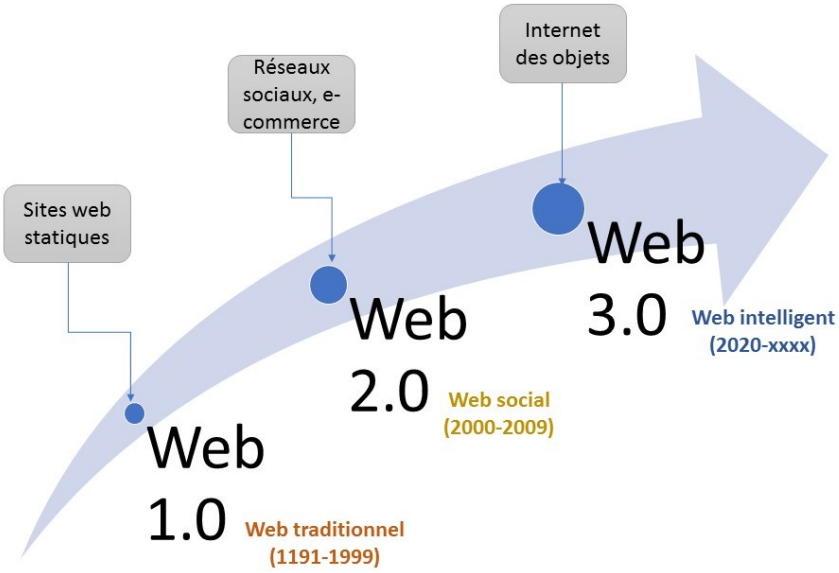


Figure 1 Évolution du Web (35)

- Tout a commencé par une phase de recherche pendant laquelle le Web était appelé ARPANET (*Advanced Research Projects Agency Network*). Le Web était alors surtout utilisé par des universitaires à des fins de recherche.
- Dans sa deuxième phase, le Web peut être qualifié de « brochure électronique ». Nous assistions alors à une véritable prise d'assaut des noms de domaine. Toutes les entreprises ressentaient le besoin de partager des informations sur Internet pour faire connaître leurs produits et leurs services.
- Lors de la troisième étape de l'évolution du Web, les données statiques sont devenues des données transactionnelles. L'achat et la vente de produits, ainsi que la prestation de services, ont vu le jour. Cette phase a été marquée par l'essor d'entreprises telles qu'ébahies et Amazon.com, mais aussi par l'explosion des entreprises « point com », malheureusement suivie de leur récession.
- Celle où nous nous trouvons actuellement correspond au Web « social » ou « expérimental ». Des sociétés telles que Facebook, Twitter et Groupon sont devenues extrêmement populaires et lucratives (à la

différence de la troisième étape du Web) en donnant aux internautes des moyens de communiquer, de rester en contact et de partager des informations (textes, photos et vidéos) avec leurs amis, leur famille et leurs collègues.

- L'IoT, la première évolution de l'Internet contrairement au Web, l'Internet se développe et s'améliore continuellement, mais il n'a pas connu de transformation fondamentale. Sa fonction reste essentiellement la même que lors de la phase ARPANET.

Par exemple, il existait au départ plusieurs protocoles de communication, notamment AppleTalk, Anneau à jeton (ou Token Ring) et IP. Actuellement, le protocole IP est devenu la norme. Dans ce contexte, l'importance de l'IoT devient considérable, puisqu'il s'agit de la première véritable évolution de L'internet.

2.2 L'origine de L'IoT

La première application, IoT est née à l'université de Cambridge en 1991.I s'agissait d'une caméra pointée sur une cafetière et connectée au réseau local de l'université. Chaque informaticien pouvait connaitre la disponibilité de café depuis son écran.

Figure 2 Equipement IoT (3)

Kevin Ashton : Le premier qui a utilisé le terme « *Internet of Things* » en 1999 pour décrire les micros puces d'identification par radiofréquence (*RFID*) (3).

En 2003, la population mondiale s'élevait à environ 6,3 milliards d'individus et 500 millions d'appareils étaient connectés à Internet. Le résultat de la division du nombre d'appareils par la population mondiale (0,08) montre qu'il y avait moins d'un appareil connecté par personne. Selon la définition de Cisco IBSG, l'IoT n'existait pas encore en 2003, car le nombre d'objets connectés était relativement faible. En outre, les appareils les plus répandus actuellement, et notamment les smartphones, faisaient tout juste leur apparition

sur le marché. En raison de l'explosion des smartphones et des tablettes, le nombre d'appareils connectés à Internet a atteint 12,5 milliards en 2010, alors que la population mondiale était de 6,8 milliards. C'est ainsi que le nombre d'appareils connectés par personne est devenu supérieur à 1 (1,84 pour être exact) pour la première fois de l'histoire (2).

2.3 Définitions et concepts de base

2.3.1 L'internet des objets

Il n'existe pas de définition standard, unifiée et partagée de l'Internet des Objets. Certaines définitions insistent sur les aspects techniques de l'IdO, tandis que d'autres se concentrent plutôt sur les usages et les fonctionnalités. Il faut réussir à exprimer ce que représente l'internet des objets tout en restant accessible aux non experts, et suffisamment concrets pour représenter son impact dans la vie quotidienne.

Figure 3 Internet des objets (4)

Selon (2), on trouve des différentes définitions :

- Conceptuellement : objets ayant des identités et des personnalités virtuelles, opérant dans des espaces intelligents et utilisant des interfaces intelligentes pour se connecter et communiquer au sein de contextes d'usages variés.
- Techniquement : une extension du système de nommage internet au sens où il est possible d'identifier de manière unifiée des éléments d'information numérique (adresses URL de sites web par exemple) et des éléments physiques (comme une palette dans un entrepôt, ou encore un mouton dans un cheptel). Mais l'identification est directe grâce à l'utilisation d'un système d'identification électronique (puces RFID, processeur et communication Bluetooth, etc.).

Croisant les deux aspects précédents, l'Iot est défini comme suit **: « *un réseau de réseaux qui permet, via des systèmes d'identification électronique normalisés et unifiés, et des dispositifs mobiles sans fil, d'identifier directement et sans ambiguïté des entités numériques et des objets physiques et ainsi de pouvoir récupérer, stocker, transférer et traiter, sans discontinuité entre les mondes physiques et virtuels, les données s'y rattachant* ».

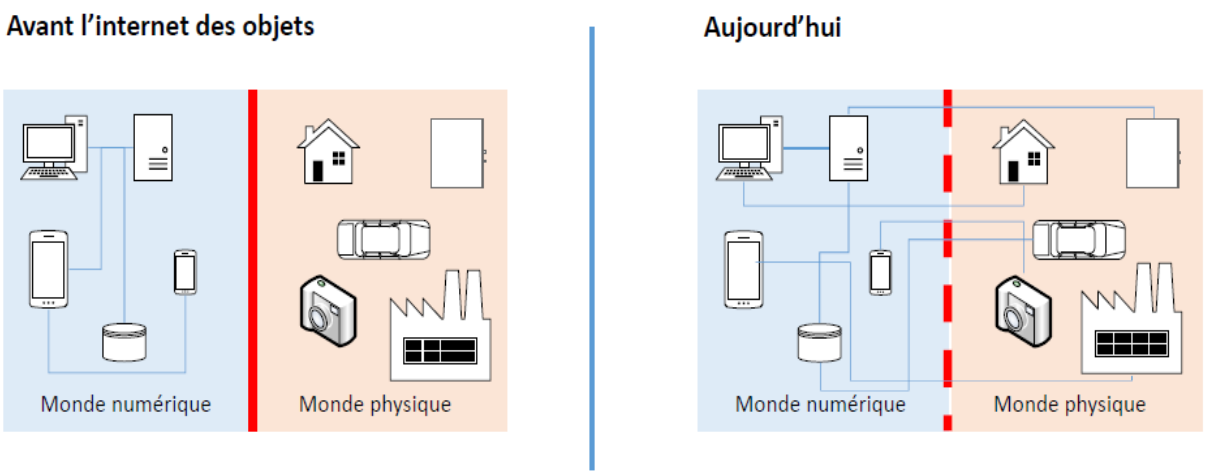


Figure 4 avant et après L'IoT (4)

2.3.2 Objets connectés

- Objets traditionnels : Ordinateurs, tablettes, smartphones… etc.

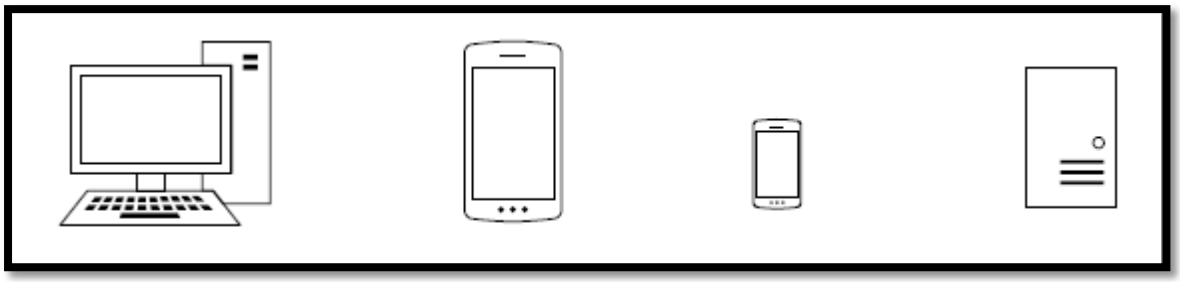


Figure 5 Objets traditionnels (4)

- Nouveaux objets connectés : appareils, électroménagers, instruments de mesures, robots, montres, véhicules… etc.

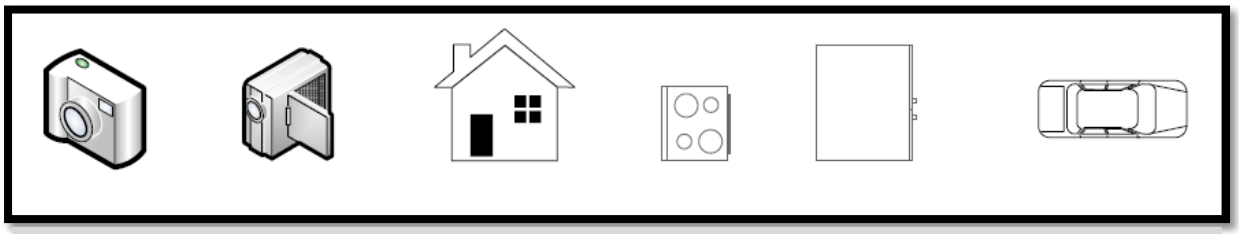


Figure 6 Nouveaux objets IoT (4)

2.3.3 Les capteurs :

Des dispositifs qui transforment une grandeur physique mesurée en une grandeur utilisable (courant ou tension électrique) à l'aide d'au moins un transducteur (5). Par exemple : un capteur de température permet de traduire l'amplitude de la température en une tension électrique. Cette dernière est numérisée puis transmise.

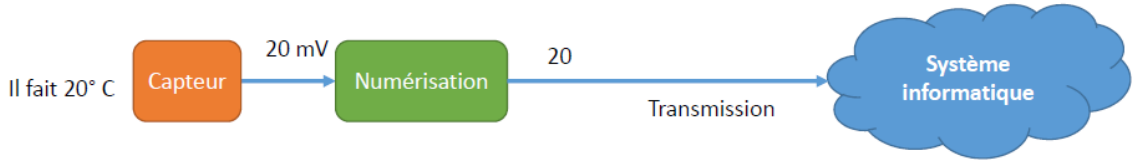


Figure 7 Capteurs (4)

2.3.4 Les actionneurs

Ils permettent d'agir dans le monde physique, c'est-à-dire, changer son état. Par exemple : un actionneur peut allumer un appareil à distance (5).

Actionneurs couramment utilisés :

• Allumage d'un éclairage

• Déclenchement d'un avertisseur sonore

• Allumage d'une machine

• Génération de mouvements (servomoteur)

• Commande de robots

• Commande de moteurs (à courant continu, pas-à-pas, etc.)

• Contrôle de débits (air, pression, liquides, etc.)

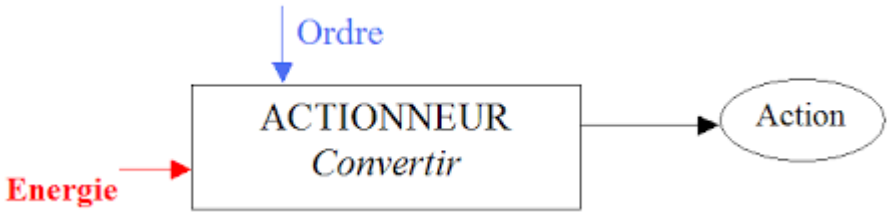


Figure Actionneur (8)

3 L'architecture de base de L'IoT

3.1 Les réseaux IoT

3.1.1 Les réseaux de communications à courte portée

RFID et NFC : La RFID (*Radio Identification*) est la technologie qui est souvent utilisée de manière passive (induction), dans des étiquettes ou des badges, afin de récolter de la donnée. Tandis que La NFC *(Near Field Communication*) est utilisée par exemple dans les paiements bancaires sans

contact. La distance de lecture est assez courte (de quelques centimètres à quelques mètres) ce qui ne permet pas de faire communiquer des objets connectés trop éloignés entre eux (choix idéal pour le télépéage par exemple).

3.1.2 Les réseaux de communications à moyenne portée

Selon (6) :

Bluetooth

Le Bluetooth est une technologie assez ancienne (1994) qui s'est développée conjointement au téléphone mobile avant d'exister de façon autonome. Ses évolutions ont permis une utilisation plus large dans le domaine de l'IoT : portée plus importante, débits plus élevés, maillage des objets.

WiFi

La technologie WIFI connecte différents objets sans fil entre eux pour permettre la transmission de données. Elle est utilisée notamment pour les équipements de la maison et pour tous les accès à Internet avec un très haut débit. La Technologie Wi-Fi utilisant un spectre sans licence dans les, bandes de 2,4 GHz et 5 GHz pour fournir une connectivité entre les appareils portables et un point d'accès Internet local.

ZigBee

ZigBee est une norme mondiale ouverte pour la technologie sans fil conçue pour utiliser des signaux radio numériques de faible puissance pour les réseaux personnels. (plus faible que le Bluetooth et le WiFi).

Z-Wave

Z — Wave est un réseau sans fil conçu par Zensys qui peut être utilisé pour contrôler l'éclairage, le chauffage et la climatisation, ainsi que pour la sécurité des appareils ménagers et de la maison et d'autres fonctions.

3.1.3 Les réseaux à longue portée bas débit

Le LPWAN « *Low Power Wide Area Network »* signifie réseau étendu à faible consommation d'énergie. Faible puissance, la technologie sans fil (LPWAN) est la réponse à la nécessité d'avoir un système qui connecte à internet des objets à faible consommation d'énergies, pour la saisie de données à faible débit, les objets connectés n'envoient souvent pas de données en continu et sont en phase quasi dormante lorsqu'une action n'est pas nécessaire et sont alimentés souvent par batterie. Ces technologies utilisent les ondes radio très efficaces dans les zones rurales. Cependant, les zones urbaines avec des immeubles de grandes hauteurs, des arbres et des maisons sont plus difficiles à parcourir par les ondes radio. En ce qui concerne les aspects réseau, les réseaux LPWAN sont généralement répartis en topologie en étoile.

3.1.4 Les réseaux cellulaires à longue portée haut débit

Les objets connectés par carte SIM ou la nouvelle génération e-Sim permet de communiquer avec les objets connectés sur de longues distances, via différents canaux (data, voix, SMS) et avec un volume de données important. De même, ces objets peuvent communiquer à un service « roaming » pour se connecter à l'international. La 1G ou la première génération de technologie cellulaire sans fil (télécommunications mobiles) ne prenaient en charge que les appels vocaux. La 2G a introduit le cryptage des appels et du texte, ainsi que des services de données tels que les SMS, les MMS et les MMS. La 2G/Edge permet un débit maximum théorique de 384 Kbit/s. Des améliorations successives ont permis l'avènement de la 3G.

La 3G/LTE

La 3G ou la troisième génération de technologies sans fil intègre des améliorations par rapport aux technologies sans fil précédentes, telles que la transmission à haute vitesse, l'accès multimédia avancé et l'itinérance mondiale. Les réseaux dérivés de la 3G permettent un débit maximum théorique jusqu'à 42 Mbit/s.

La 4G

4G la quatrième génération prend en charge des débits de données encore plus élevés que la 3G et plus de vitesse pour le téléchargement de fichiers très volumineux et permet de visionner instantanément des vidéos en haute définition. La 4G et ses évolutions permettent un débit maximum théorique jusqu'à 300 Mbit/s.

5G

La cinquième génération de technologie mobile, caractérisée par la vitesse de transmission des données et une réduction de la latence de bout en bout.

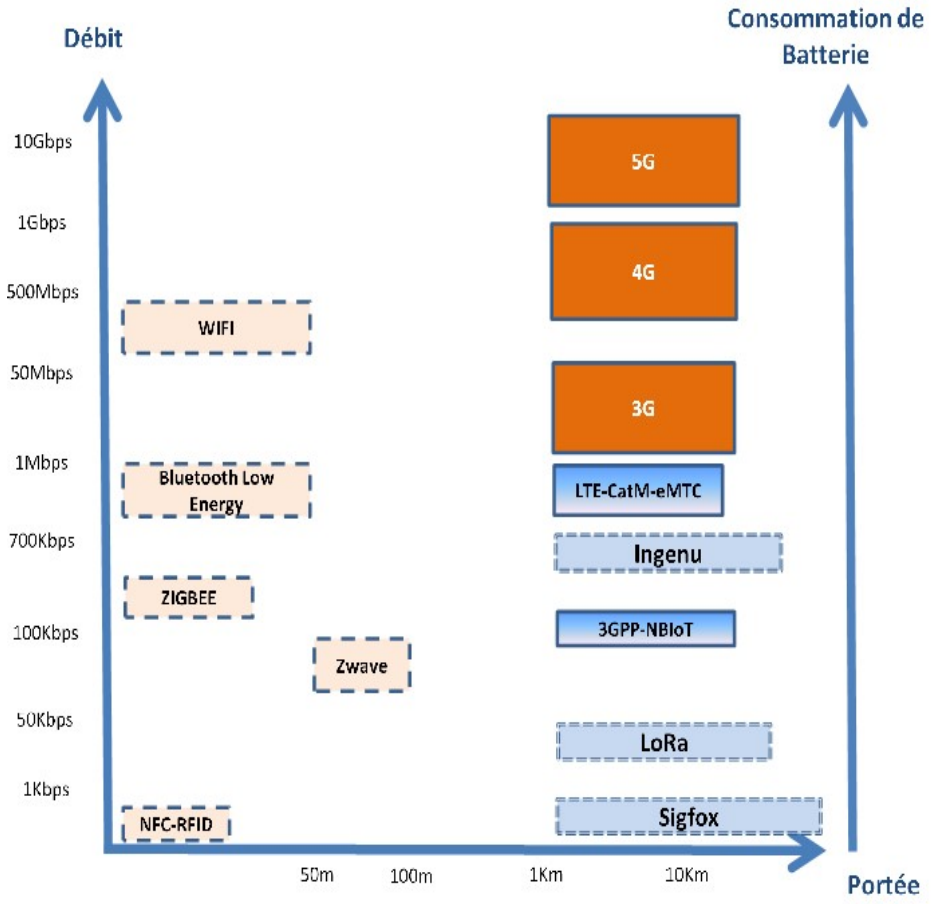


Figure 9 Types de réseaux de communication (6)

3.2 Architecture et composants de L'IoT

L'architecture d'une solution IoT varie d'un système à l'autre en se basant sur le type de la solution à mettre en place. L'architecture de base de l'IdO (représentée dans la figure ci-dessous) est composée de trois couches : une couche de perception (objets), une couche réseau (transport et traitement) et la couche application (services et applications).

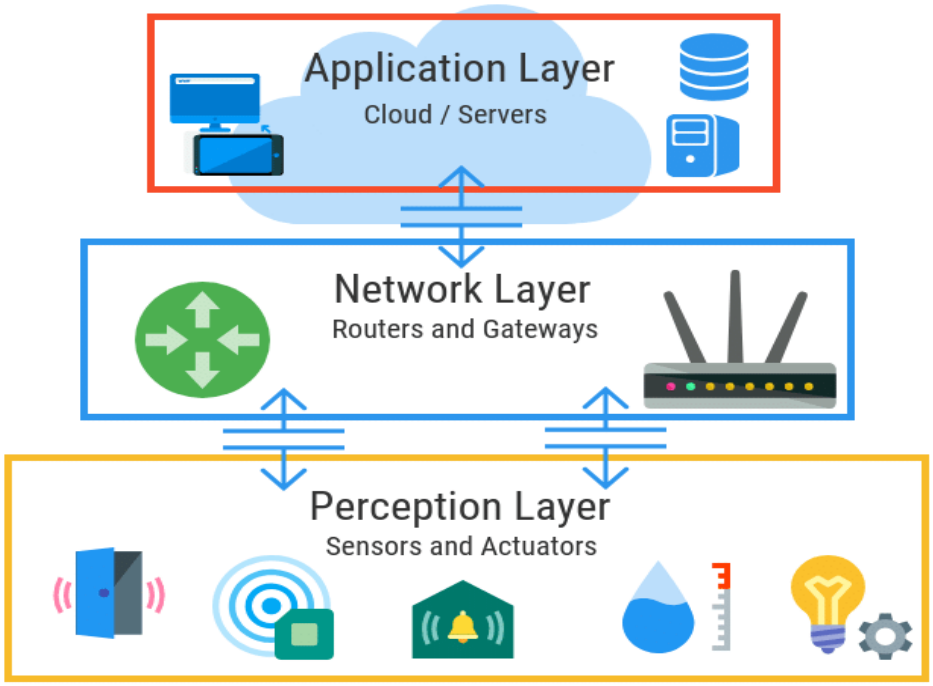


Figure 10 architecture e L'IoT (37)

3.2.1 La couche perception

est essentiellement composée des objets connectés et de la récolte d'informations. Elle comprend, des étiquettes et des lecteurs de code-barres à deux dimensions (2D), des étiquettes de radio-identification (RFID), des systèmes mondiaux de positionnement (GPS), des capteurs, des terminaux, des objets pouvant être qualifiés d'actionneurs ainsi que de capteurs et un réseau de capteurs. Cette couche de perception est chargée de convertir l'information en signaux numériques pour leurs transmissions sur le réseau.

3.2.2 La couche Réseau

est comme le système nerveux et le cerveau de l'IoT, sa fonction principale est de traiter les informations provenant de la couche perception. Elle est également chargée de la transmission des informations traitées à la couche application par le biais de diverses technologies de réseaux (WiFi, Bluetooth, Bluetooth Low Energy, Constrained Application Protocol, protocole de routage RPL, Message Queing Telemet Transport, etc.).

3.2.3 La couche application

utilise les données traitées par la couche précédente, elle constitue la partie de front de l'architecture, qui permet d'exploite tout le potentiel des données. De plus, elle a pour rôle de fournir les outils nécessaires aux développeurs pour réaliser la vision et toutes les applications possibles de l'IoT (7).

3.3 Composantes d'une solution IoT (3)

Généralement, une solution IoT est formée des composants suivants comme le montre la figure 11 : Capteur, connectivité, Données et valeur.

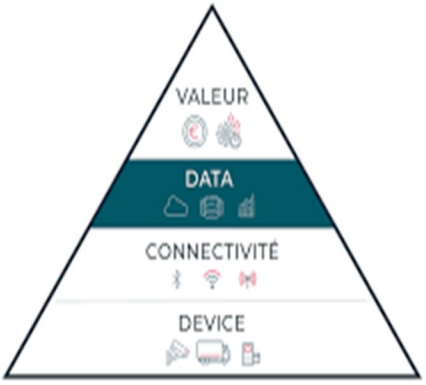


Figure 11 couches d'une plateforme IoT (38)

La première couche :

est constituée des composants physiques principaux d'un objet connecté : **les capteurs** (captent et collectent les données physiques environnantes cela peut être un taux d'humidité, une température, une présence, une pression), **les actionneurs** (déclenchent une action sans intervention humaine)**, la passerelle** (qui va envoyer les données captées sur le réseau pour qu'elles soient analysées).

La deuxième couche : est celui de la connectivité pont entre le monde physique et le monde digital, et donc à savoir comment cette donnée captée va être communiquée sur le réseau Internet. (Par Wifi, réseau cellulaire, Bluetooth, etc.)

La troisième couche : **cloud-computing** les données arrivent à l'état brut. Ce sont des suites de chiffres qui doivent être triées, analysées, stockées**.** L'endroit où sont centralisées ces données s'appelle une plateforme IoT. Cette plateforme assure également l'intégration avec d'autres systèmes ou applications.

La quatrième couche : Enfin, tout en haut de la pyramide, il s'agit de transformer ces données traitées pour leur donner du sens et de la valeur. Surtout, d'être en mesure de les présenter sous une interface compréhensible et utilisable.

4 Domaines d'application de l'IoT

4.1 Domotique

Cette catégorie regroupe les **appareils de contrôle à distance** *:* allumer et éteindre les appareils à distance pour éviter les accidents et économiser de l'énergie, **l'utilisation de l'énergie et de l'eau :** surveillance de la consommation d'énergie et d'eau pour obtenir des conseils sur la façon d'économiser les coûts et les ressources, **L'art et préservation des biens** *:* suivi de l'état de conservation à l'intérieur des musées et des entrepôts d'art, et **les systèmes de détection d'intrusion** *:* détection des ouvertures de portes, de fenêtres et des violations dans le but d'empêcher les intrusions Environnement intelligent (8).

Cette catégorie regroupe la *détection* **précoce des tremblements de terre** contrôle distribué dans des endroits spécifiques de tremblements, les **glissements de terrain et la prévention des avalanches** *:* surveillance de l'humidité du sol, des vibrations et de la densité de la terre pour détecter les tendances dangereuses dans les conditions du terrain, la **surveillance du**

niveau de neige *:* mesure de niveau de neige pour connaître en temps réel la qualité des pistes de ski et permettre la sécurité des avalanches, la **détection des incendies de forêt** *:* surveillance des gaz de combustion et des conditions d'incendie pour définir les zones d'alerte, et **la pollution de l'air** contrôle des émissions de CO2 des usines, de la pollution émise par les voitures et des gaz toxiques.

4.2 Transport et logistique

Cette catégorie regroupe la **détection d'incompatibilité** *de stockage :* émissions de conteneurs stockant des produits inflammables fermés à d'autres contenant des matières explosives, le **suivi de flotte** *:* contrôle du suivi des itinéraires pour les marchandises sensibles comme les bijoux, les médicaments ou les marchandises dangereuses, **l'emplacement des articles** *:* recherche d'éléments individuels dans de grandes surfaces comme les entrepôts ou les ports et la **qualité des conditions d'expédition** *:* surveillance, à des fins d'assurance, des vibrations, des coups, des ouvertures de conteneurs ou de leur entretien.

4.3 Agriculture intelligente

Cette catégorie regroupe le **compost** *:* contrôle de l'humidité et des niveaux de température dans le foin, la paille, etc. pour prévenir les champignons et autres contaminants microbiens, **les stations météorologiques** *:* étude des conditions météorologiques dans les champs pour prévoir la formation de glace, la pluie, la sécheresse, la neige ou les changements de vent, les **cours de golf** *:* l'irrigation sélective dans les zones sèches pour réduire les ressources en eau nécessaires, les **serres :** contrôler les conditions microclimatiques pour maximiser la production de fruits et légumes et sa qualité et l'**hydroponique** *:* contrôler l'état des plantes cultivées dans l'eau pour obtenir les cultures les plus efficaces.

4.4 Villes intelligentes

Cette catégorie regroupe le contrôle des **niveaux de champs électromagnétiques** *:* mesure de l'énergie rayonnée par les stations cellulaires

et les routeurs WiFi, la **santé structurelle** *:* surveillance des vibrations et des conditions matérielles dans les bâtiments, les ponts et les monuments historiques, la *gestion des déchets :* détection des niveaux d'ordures dans les conteneurs pour optimiser les voies de collecte, **la détection de smartphone** *:* détecter les smartphones et en général tout appareil fonctionnant avec des interfaces WiFi ou Bluetooth, les **routes intelligentes** *:* autoroutes intelligentes avec messages d'avertissement et de détournements en fonction des conditions climatiques et des événements inattendus tels que les accidents (8).

5 Caractéristiques de l'IoT

5.1 Connectivité

est une exigence importante de l'infrastructure IoT. Les objets de l'IoT doivent être connectés à cette infrastructure. N'importe qui, n'importe où, n'importe quand, la connectivité doit être garantie à tout moment sans connexion rien n'a de sens.

5.2 Intelligence et identité

L'extraction de connaissances à partir des données générées est très Importante. Par exemple, un capteur génère des données, mais ces données ne seront utiles que si elles sont interprétées correctement. Chaque appareil IoT a une identité unique. Cette identification est utile pour suivre l'équipement et parfois pour interroger son état.

5.3 Évolutivité

Le nombre d'éléments connectés à la zone IoT augmente de jour en jour. Par conséquent, une configuration IoT devrait être capable de gérer l'expansion massive. Les données générées en tant que résultat sont énormes et doivent être traitées de manière appropriée.

Dynamique et autoadaptatif (complexité)

Les appareils IoT doivent s'adapter dynamiquement aux contextes et aux scénarios changeants. Supposons une caméra destinée à la surveillance. Il doit être adaptable pour travailler dans différentes conditions et différentes situations d'éclairage (matin, après-midi, nuit).

5.4 Architecture

architecture IoT ne peut pas être de nature homogène. Il devrait être hybride, prenant en charge les produits de différents fabricants pour fonctionner dans le réseau IoT. L'IoT n'appartient à aucune branche d'ingénierie. L'IoT est une réalité lorsque plusieurs domaines se rencontrent.

5.5 Sécurité

Il existe un risque que les données personnelles sensibles des utilisateurs soient compromises lorsque tous leurs appareils sont connectés à Internet. Cela peut entraîner une perte pour l'utilisateur. Par conséquent, la sécurité des données est le défi majeur. De plus, le nombre d'équipements impliqué est énorme. Les réseaux IoT peuvent également être à risque. Par conséquent, la sécurité de l'équipement est également critique.

6 La sécurité de l'IoT

6.1 Qu'est-ce que la sécurité de l'IoT ?

Les données échangées sur Internet entre deux objets, ou entre un objet et un utilisateur, sont exposées à diverses attaques telles que l'écoute, la falsification et le déni de service, d'où l'importance de sécuriser le réseau. Ces attaques, qui ont des conséquences directes sur la confidentialité et l'intégrité des données, peuvent être contrées par l'établissement de canaux sécurisés entre les différentes entités de l'IoT. (9)(La sécurité de l'IoT est un domaine d'expertise extrêmement vaste. Non seulement parce qu'il y a énormément d'applications différentes, mais aussi en raison de la complexité de l'IoT.

L'application IoT moyenne est en effet constituée de plusieurs couches qui doivent être sécurisées.

- **Protection de la technologie** concerne la sécurité des données, des communications et des infrastructures réseaux et leurs fonctionnalités.
- **Protection des personnes** concerne la protection de la vie privée des usagers (« *privacy* ») pour éviter des litiges causés éventuellement par l'IoT.
- **Protection des systèmes interconnectés** et hébergeant les objets de l'IoT, concernera la protection des objets eux-mêmes livrés à ces systèmes et les processus qu'ils contrôleront.

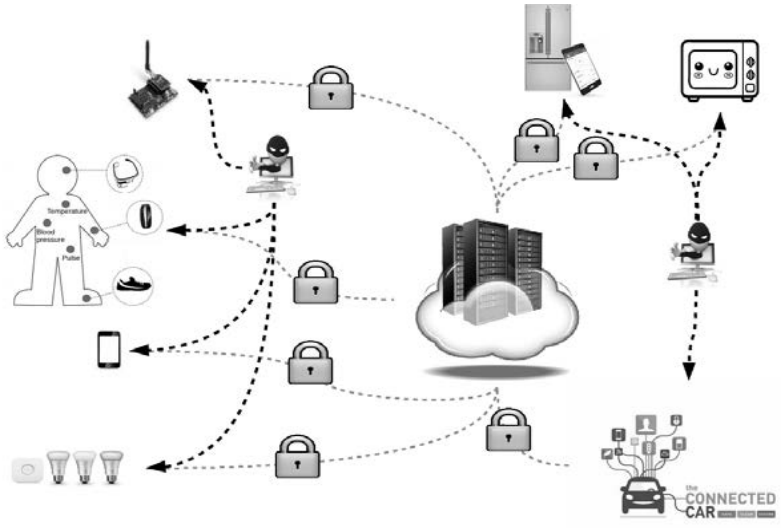


Figure 12 Mécanisme de sécurité IoT (39)

6.2 Quelle est l'importance de la sécurité ?

La sécurité est une interaction de 3 composantes : disponibilité, intégrité et confidentialité (10). Si l'un de ces aspects n'est pas garanti, on ne peut pas parler de sécurité adéquate.

- **Disponibilité** : veiller à ce que les parties autorisées aient accès aux données et aux systèmes si nécessaire. Par exemple, en prenant des mesures contre les attaques DDoS.
- **Intégrité :** empêcher toute modification non autorisée des données, par exemple pendant le transport. Ceci permet de garantir leur fiabilité et leur précision.
- **Confidentialité** : Pour les objets connectés, la sécurité consiste principalement à protéger la vie privée de l'utilisateur. Les risques varient selon les appareils, mais la plus grande mine d'or pour les pirates informatiques est probablement le smartphone. Appels, coordonnées des clients, photos, vidéos, informations de localisation, recherche et historique du navigateur.

6.3 Les problèmes de sécurité des IoT

Environnement non contrôlé : Beaucoup des choses fera partie d'un environnement hautement incontrôlé, des choses se rendre dans un environnement indigne de confiance, éventuellement sans surveillance les sous-propriétés de l'environnement non contrôlé sont : mobilité, accessibilité physique, et le manque de confiance.

- **Mobilité** : Une connectivité réseau stable et une présence constante ne peuvent être attendues dans un tel environnement.
- **Accessibilité physique** : Dans l'IoT, les capteurs peuvent être accessibles au public, par exemple, les caméras de contrôle du trafic et les capteurs environnementaux.
- **Confiance** : Les relations de confiance a priori sont peu probables pour la grande quantité d'appareils interagissant entre eux et avec les utilisateurs.

- **Hétérogénéité** : On s'attend à ce que l'IdO soit un écosystème hétérogène, car il devra intégrer une multitude des choses de divers fabricants. Par conséquent, la compatibilité des versions et l'interopérabilité doivent être prises en compte.

- **Évolutivité** : La grande quantité d'interconnexions des choses dans l'IoT exigent des protocoles hautement évolutifs. Cela a une influence sur les mécanismes de sécurité. Par exemple, les approches centralisées, telles que les infrastructures à clé publique (PKI) hiérarchiques, ainsi que certaines approches distribuées, telles que les schémas.
- **Ressources limitées** : Choses dans l'IoT aura contraintes à prendre en compte pour les mécanismes de sécurité. Cela inclut les limitations d'énergie, par exemple, les appareils alimentés par batterie, ainsi qu'une faible puissance de calcul, par exemple, les microcapteurs. Ainsi, les algorithmes cryptographiques de calcul lourds ne peuvent pas être appliqués à tous des choses (11).

6.4 Méthodes de sécurité pour Protéger IoT

6.4.1 Exigences de sécurité du réseau :

peut être divisé en confidentialité, authenticité, intégrité, disponibilité. Celles-ci s'appliquent aux architectures IoT, par exemple les ressources limitées, doivent être prises en compte. L'IoT nécessite des architectures pour faire face à l'hétérogénéité des objets. Interconnexion des objets peut nécessiter confidentialité, par exemple, pour empêcher l'écoute clandestine d'informations sensibles via la transmission Internet.

6.4.2 La gestion des identités :

pose un défi spécifique dans l'IoT en raison du nombre d'appareils, mais aussi en raison de la relation complexe entre les appareils, les services, les propriétaires et les utilisateurs, il faut donc porter une attention particulière à l'authentification, l'autorisation y compris la non-répudiation.

La responsabilité de cette gestion d'identité garantit que chaque action est clairement liée à une entité authentifiée. La responsabilité est un défi particulier dans l'IoT en raison de l'ampleur de la réutilisation des appareils, des services et des données à de nombreuses fins. Ainsi, la responsabilité doit traiter d'énormes quantités d'entités, de délégations d'accès, d'actions qui

couvrent des domaines organisationnels, ainsi que de la dérivation continue de données.

6.4.3 Confidentialité

La vie privée est considérée comme l'une des plus défis dominants dans l'IoT en raison de l'implication des citoyens et de la collecte de données de plus en plus omniprésente, par exemple dans les scénarios de maison intelligente. Il existe une pléthore de définitions de la vie privée en fonction de la vision d'une solution informatique.

6.4.4 Confiance :

La confiance est une autre exigence cruciale dans l'IoT en raison du fait qu'il est hautement distribué ainsi que fiable sur des données qualitatives. La confiance peut être décomposée en appareil de confiance, fiducie d'entité, et confiance des données. La confiance des appareils dans l'IoT est un défi, car la confiance a priori dans les appareils ne peut pas toujours être établie, par exemple en raison d'une dynamique élevée et de relations inter domaines. Par conséquent, des approches telles que l'informatique de confiance pour (les appareils standardisés) ainsi que la confiance informatique sont nécessaires pour établir la confiance des appareils. De plus, chaque entité peut évaluer la confiance dans un appareil différemment, par conséquent, les architectures IoT doivent gérer des vues non singulières de la confiance.

6.4.5 Résilience :

La fusion d'échelle de l'IoT en termes de périphériques crée une grande surface pour les attaques et les défaillances. Pour cette raison, la résilience et la robustesse contre les attaques et les défaillances s'appliquent, en tant qu'exigences importantes, à l'IoT. Les architectures doivent fournir les moyens de sélectionner avec compétence des choses, les chemins de transmission et les services selon leur robustesse (panne/attaque évitement). De plus, pour assurer la résilience, des mécanismes de basculement et de récupération doivent être fournis pour maintenir les opérations en cas de panne ou

d'attaques, et pour revenir à des opérations normales (panne/attaque atténuation.

7 Conclusion

dans ce chapitre, nous avons fait un tour sur l'IoT. Nous avons exposé quelques concepts liés à cette technologie, en montrant ses dimensions historiques et son potentiel promoteur pour le future.

Dans le prochain chapitre, nous allons aborder la modélisation des aspects de sécurité en sa basant sur le standards UML.

II. Chapitre 2 : modélisation des aspects de sécurité IoT en UML

1 Introduction

L'arrivée de l'Internet des objets (*IoT*) a soulevé de nombreux défis pour les architectes et les concepteurs de systèmes complexes. Afin de les aider et les orienter, plusieurs solutions de modélisation et de conception ont été proposées telles que les profils UML (Unified Modeling Language), ce dernier qui s'est imposé comme le langage de modélisation le plus utilisé dans l'industrie et dans le milieu universitaire (12).

Dans ce deuxième chapitre, on va aborder la modélisation en utilisant le langage UML, passant par introduire la notion de Profil UML et finissant par une description d'une extension qui prend en compte les aspects de sécurité des systèmes IoT.

2 UML

2.1 Définition

Le Langage de modélisation unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de diagramme conçu comme une méthode normalisée de visualisation dans les domaines du ***développement logiciel*** et en ***conception orientée objet*** (13), (14)**.**

2.2 Caractéristiques UML

2.2.1 Le modèle :

Un modèle est une abstraction de la réalité (L'abstraction est un des piliers de l'approche objet). Il s'agit d'un processus qui consiste à identifier les caractéristiques intéressantes d'une entité, en vue d'une utilisation précise. L'abstraction désigne aussi le résultat de ce processus, c'est-à-dire l'ensemble

des caractéristiques essentielles d'une entité, retenues par un observateur. (15).

2.2.2 Vues d'architecture d'UML

UML fournit quatre vues d'architecture différente

- **La vue logique :**

Cette vue de haut niveau se concentre sur l'abstraction et l'encapsulation, elle modélise les éléments et mécanismes principaux du système.

- **La vue de processus :**

Cette vue est très importante dans les environnements multitâches ; elle montre : la décomposition du système en terme de processus (tâches). Et les interactions entre ces processus.

- **La vue de déploiement :**

Cette vue très importante dans les environnements distribués décrit les ressources matérielles et la répartition du logiciel dans ces ressources.

- **La vue des besoins des utilisateurs :**

Cette vue définit les besoins des clients du système et centre la définition de l'architecture du système sur la satisfaction (la réalisation) de ces besoins.

- **La vue de composants :**

cette vue identifie les modules qui réalisent (physiquement) les classes de la vue logique, elle montre aussi l'organisation des modules en « sous-systèmes », les interfaces des sous-systèmes et leurs dépendances (avec d'autres sous-systèmes ou modules) (15).

2.2.3 Composants de base d'UML : (16)

UML est composé de trois blocs de construction : objets (*things),* relations (*relationships)* et diagrammes (diagrams), comme le montre

la figure suivante :

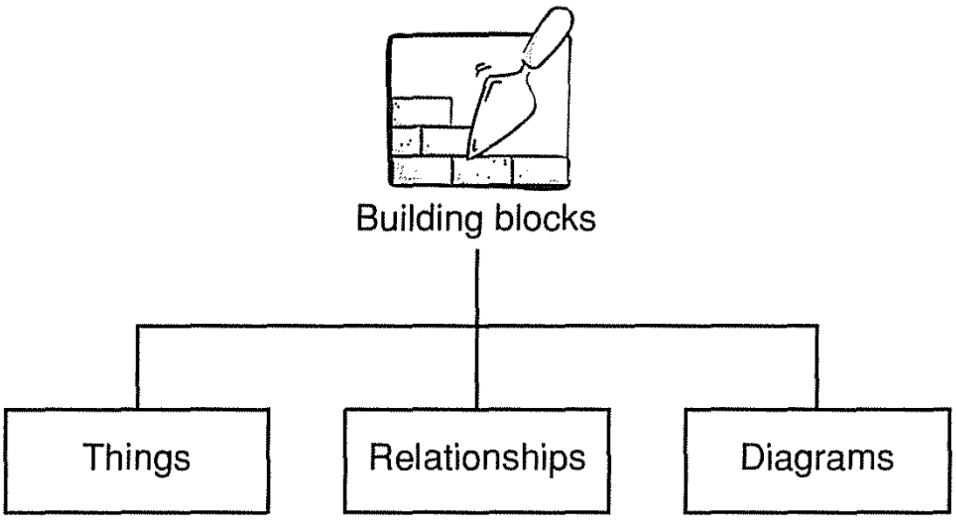


Figure 13 Composants de l'UML (16)

2.2.3.1 Objets :

Les éléments UML peuvent être divisés en choses structurelles — les noms d'un modèle UML, tels que classe, interface, collaboration, cas d'utilisation, classe active, composant, nœud ;les éléments comportementaux les verbes d'un modèle UML, tels que les interactions, les activités, machines à états ;les éléments de regroupement le package, qui est utilisé pour regrouper des éléments de modélisation sémantiquement liés en unités cohérentes.

2.2.3.2 Les relations :

Les relations vous permettent de montrer sur un modèle comment deux ou plusieurs choses sont reliées entre elles. Dans les modèles UML : elles vous permettent de capturer des connexions significatives (sémantiques) entre les choses. La compréhension de la sémantique exacte des différents types de relation est une partie très importante de la modélisation UML. Par exemple, les relations UML qui s'appliquent aux structurations et au regroupement des éléments dans un modèle sont résumées dans le tableau ci-dessous.

Class Diagram Relationship Type	Notation
Association	
Inheritance	
Realization/ Implementation	
Dependency	
Aggregation	
Composition	

Figure 14 les relations dans UML (16)

2.2.3.3 Diagrammes

Dans tous les outils de modélisation UML, lorsque vous créez une nouvelle chose ou une nouvelle relation, elle est ajoutée au modèle. Le modèle est le référentiel de toutes les choses et de toutes les relations que vous avez créées pour aider à décrire le comportement requis du modèle. Les diagrammes ne sont que des fenêtres ou des vues sur le modèle. Le diagramme n'est pas modèle lui-même.

Il existe treize types différents de diagrammes UML, qui sont énumérés à la Figure 15.

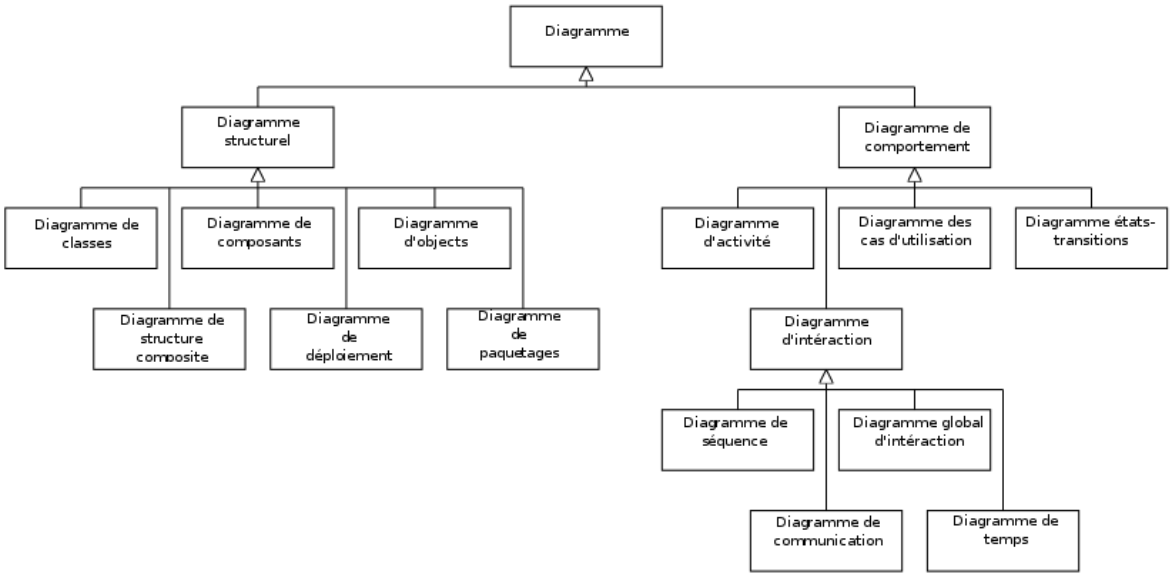


Figure 15 Digramme de classe :(Aspect structurel) (14)

- **Digramme de classe :(Aspect structurel)** un diagramme de classes décrit les types d'objets du système et les différents types de relations statiques qui existent entre eux. Les diagrammes de classes montrent également les propriétés et les opérations d'une classe ainsi que les contraintes qui s'appliquent à la manière dont les objets sont connectés. L'UML utilise le terme caractéristique comme un terme général qui couvre les propriétés et les opérations d'une classe.

La figure ci-dessous illustre un diagramme de classe, dont les cases du diagramme représentent des classes, qui divisées en trois compartiments : le nom de la classe (en gras), ses attributs et ses opérations (14).

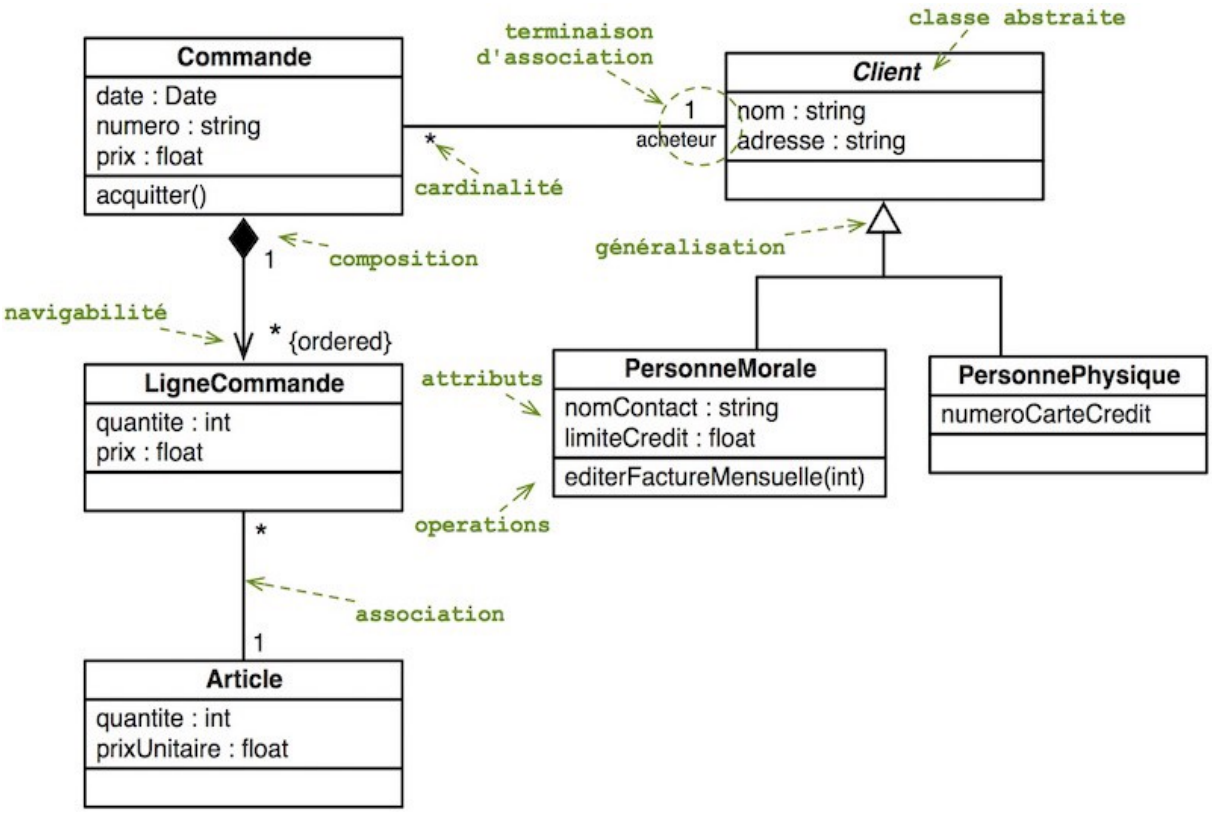


Figure 16 Diagramme de classe (17)

- **Diagramme de cas d'utilisation (aspect comportemental) :**

Les cas d'utilisation sont une technique pour capturer les exigences fonctionnelles d'un système.ils fonctionnent en décrivant les interactions typiques entre les utilisateurs d'un système et le système lui-même, en fournissant un récit de la façon dont un système est utilisé.

la figure qui suit illustre un exemple de diagramme de cas d'utilisation qui montre les acteurs, les cas d'utilisation et les relations entre eux. :

— Quels acteurs réalisent quels cas d'utilisation ?

— Quels cas d'utilisation incluent d'autres cas d'utilisation ? (14)

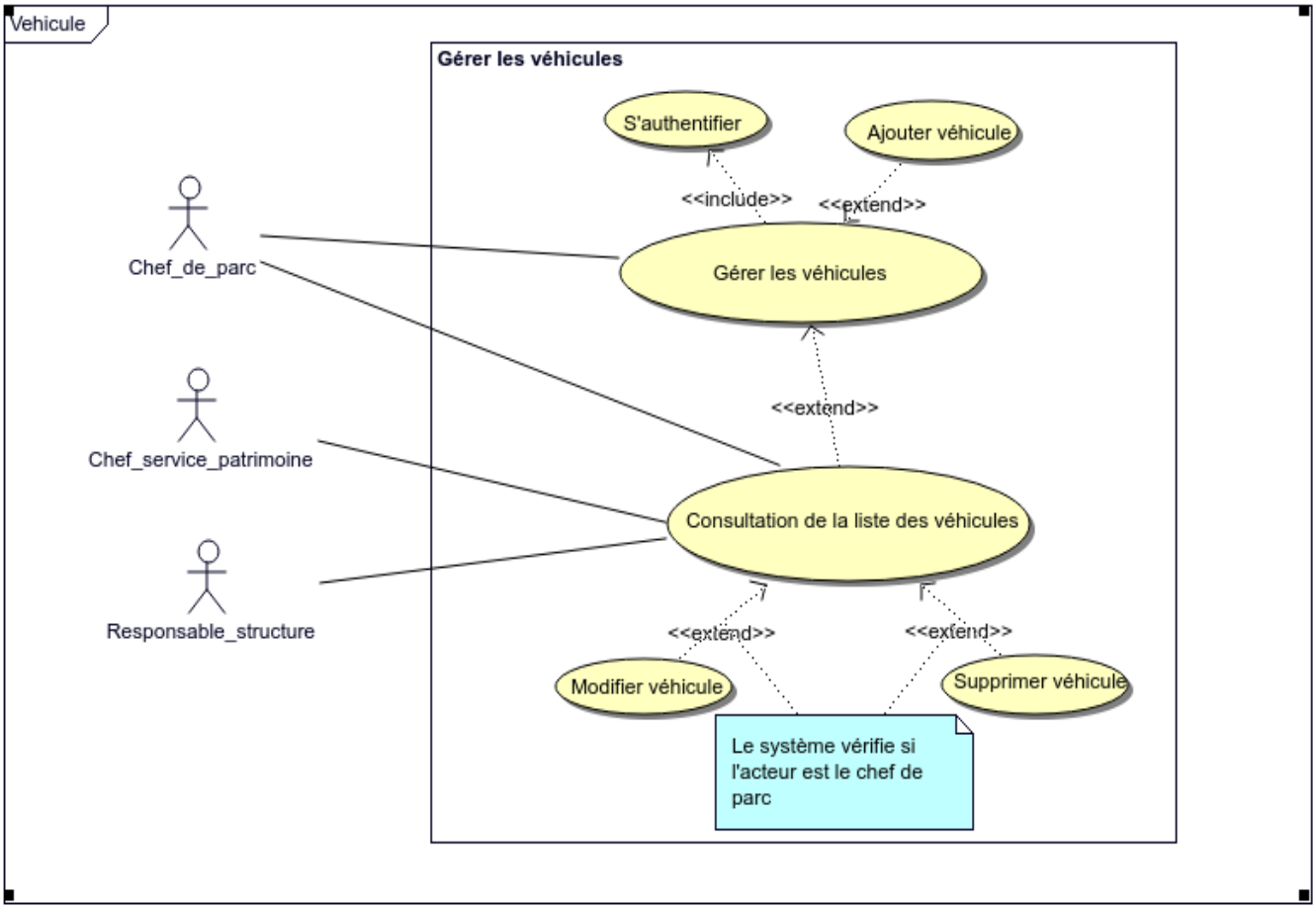


Figure 17 Diagramme de cas d'utilisation (18)

2.3 Avantages et inconvénients d'UML

2.3.1 Points forts

- **UML est un langage formel et normalisé**

- gain de précision
- gage de stabilité
- encourage l'utilisation d'outils

- **UML est un support de communication performant**

- Il cadre l'analyse.
- Il facilite la compréhension de représentations abstraites complexes.
- Son caractère polyvalent et sa souplesse en font un langage universel.

2.3.2 Points faibles :

- La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.
- Le processus (non couvert par UML) est une autre clé de la réussite d'un projet. Or, l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue.
- Sémantique floue ou mal définie pour certains types de diagrammes (15).

2.4 **Profils UML**

2.4.1 C'est quoi un Profil UML ?

Par rapport au formalisme UML standard, les développeurs souhaitent souvent rajouter des caractéristiques supplémentaires pour tenir compte de la spécificité de leur domaine d'application. Afin de satisfaire ce besoin, UML est doté d'un mécanisme d'extensibilité permet de particulariser le méta modèle UML pour qu'il prenne en considération les besoins de modélisation spécifiques. Le résultat est un profil UML (19).

- Un profil UML personnalisé est utilisé dans les situations suivantes :

- pour étendre et personnaliser le méta modèle UML (sera expliqué et détailler dans le troisième chapitre) pour un domaine particulier.
- Pour fournir une syntaxe pour les constructions qui n'ont pas de notation UML.
- Pour fournir une notation différente pour des symboles existants.
- Pour ajouter de la sémantique qui n'existe pas dans le méta modèle UML, telle qu'un minuteur ou une horloge
- pour ajouter des informations utilisées par une transformation pour générer du code à partir d'un modèle.

- Un profil UML est une collection de trois composants : stéréotypes, tagged values et contraintes, dont leurs définitions sont comme suit : (16)

- **Stéréotypes** : permettent de définir un nouvel élément de modélisation UML basé sur un élément existant.et ajoutent de nouveaux éléments au métamodèle UML.
- **Contraintes :** étendent la sémantique d'un élément en nous permettant d'ajouter de nouvelles règles.
- **Tagged values :** permettent d'étendre la spécification d'un élément en nous permettant d'y ajouter de nouvelles informations.

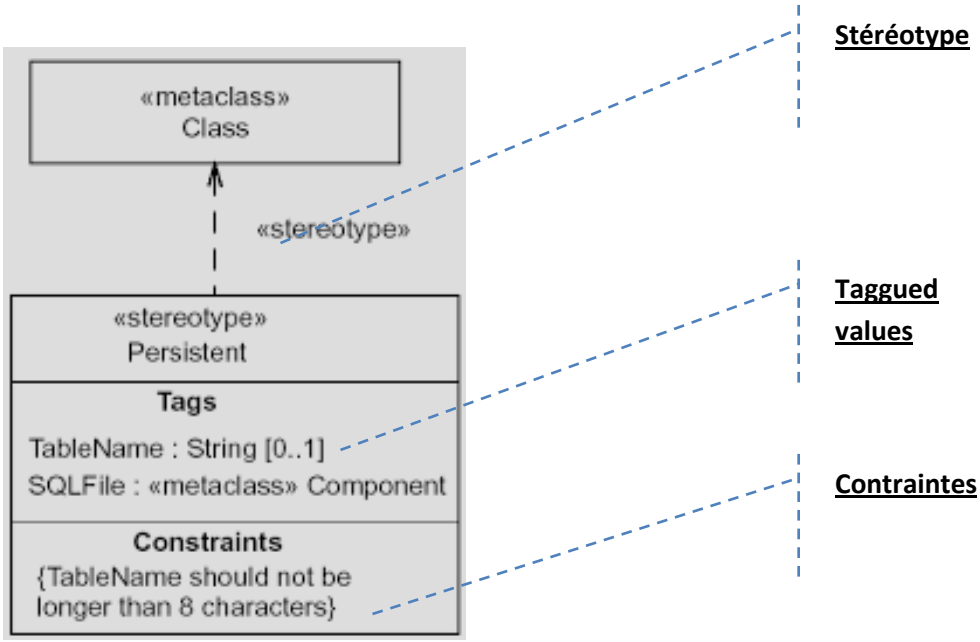


Figure Composants de Profil Uml(18)

2.4.2 Mécanisme d'élaboration de Profil UML

Le mécanisme de profil a été défini spécifiquement pour fournir un mécanisme léger d'extension pour le standard UML. En UML, les stéréotypes sont des métaclasses spécifiques, les tagged values sont des méta-attributs.

Pour récapituler, un profil UML permet de :

- Ajouter des concepts (Terminologies) relatifs à une plateforme ou un domaine particulier.

- Fournir/Changer la représentation de ces concepts, avec les images associées aux stéréotypes, pour utiliser la représentation « traditionnelle » spécifique du domaine ;
- Fixer les points de variation sémantique : faire un choix parmi les possibilités sémantiques laissées ouvertes par UML. Par exemple : les signaux (signal) peuvent être en broadcased ou mulitcasted, les State-Machines ont une queue FIFO ou LIFO, etc. ;
- Ajouter des contraintes sur les associations entre ces concepts, c'est-à-dire restreindre les associations existantes au niveau des métaclasses UML, limiter le nombre d'attributs du concept à N, forcer des règles logiques d'associations, etc. ;
- Ajouter des contraintes sur l'utilisation ou non de certains concepts selon le contexte ;
- Ajouter de l'information qui peut être utilisée lors de la transformation entre un modèle vers un autre ou vers du code (20).

3 IoTSEC

Les systèmes IoT contrôlent toute une variété de systèmes, par exemple un réseau robotique dans l'industrie automobile, des réseaux de voitures autonomes, les soins de santé humaine et de nombreux systèmes dans d'autres domaines. Par conséquent, l'existence de systèmes IoT qui n'impliquent pas de mécanismes de sécurité devrait être impensable ; néanmoins il n'y a pas de nouvelles normes qui prennent en compte la sécurité de ces systèmes d'où la nécessité d'introduire une nouvelle extension UML qui prend en considération les problèmes de sécurité.

3.1 Vue d'ensemble

La base de l'IoT, et qui résume la collecte, le traitement, la distribution et l'analyse d'informations personnelles ou industrielles. Représente de nouvelles vulnérabilités et menaces que les attaquants vont certainement exploiter. L'IoT présente les lacunes courantes en matière de sécurité Internet, tous les

développeurs IoT doivent tenir compte des exigences de sécurité au sein de leurs systèmes, notamment la confidentialité des données, la disponibilité des services, l'intégrité, l'antimalware, le contrôle d'accès, etc.

Le Open Web Application Security Project's(OWASP) (21), a donné la liste des dix principales vulnérabilités de l'Internet des objets et qui résument la plupart des préoccupations et des vecteurs d'attaque entourant les dispositifs IoT : interface web non sécurisée, l'authentification/autorisation, services réseau non sécurisés, absence de cryptage du transport, problèmes de confidentialité, l'interface mobile non sécurisée, configurabilité insuffisante de la sécurité, Logiciel/micrologiciel non sécurisé et sécurité physique insuffisante.

Les problèmes de sécurité de l'IoT doivent être traités comme une approche d'ingénierie système basée sur un modèle afin de simplifier la modélisation lors de conception d'un système Io (21).

3.2 **Modélisation de la sécurité des IoT dans IoTsec**

3.2.1 SysML

Le langage de modélisation prposé par OMG SysML est un profil UML qui représente un sous-ensemble d'UML avec des extensions, il prend en charge la spécification, l'analyse, la conception, la vérification et la validation de systèmes comprenant du matériel, des logiciels, des données, du personnel, des procédures et des installations. SysML est un langage de modélisation visuel qui fournit une sémantique (sens) et une notation (représentation du sens) ; ce n'est pas une méthodologie ou un outil. SysML a également été étendu pour décrire les systèmes IoT, il introduit un processus de conception et d'analyse soutenu par un cadre pour aider les ingénieurs à modéliser les applications IoT et à vérifier leurs propriétés, néanmoins il ne modélise pas les problèmes de sécurité qui, comme mentionnés précédemment, sont fondamentaux pour les systèmes IoT(22).

3.2.2 UMLsec

Il existe une extension de l'UML pour le déploiement de systèmes sécurisés appelée UMLsec, qui a vingt et un stéréotypes orientés vers les problèmes de sécurité, ils encapsulent les connaissances sur l'ingénierie de sécurité prudente et les mettent

ainsi à la disposition des développeurs qui peuvent ne pas être spécialisés dans la sécurité. Cette approche vise à valider les modèles à l'aide d'un langage formel (représentation mathématique), mais ils ne modélisent pas les systèmes IoT, où de plus en plus de nouvelles vulnérabilités peuvent être trouvées (23).

3.2.3 SysMLsec

Le même manque se produit dans SysMLsec , c'est un nouvel environnement SysML qui introduit des diagrammes pour les questions de sécurité et une méthodologie associée, il propose le stéréotype <<exigence de sécurité>> qui est utilisé dans le diagramme d'exigences de SysML. C'est une extension très utile, car elle peut modéliser les questions de sécurité de manière appropriée (24).

3.2.4 IoTSec : Extension uml pour la modélisation des exigences de sécurité des systèmes IoT

L'extension/le profil UML est proposé pour guider les développeurs out au long du cycle de vie de la conception des systèmes IoT, ceci en ce qui concerne les exigences de sécurité à chaque étape. Une extension UML est proposée appelée IoTsec qui est un sous-ensemble d'UML et de SysML.

IoTsec applique les mécanismes de stéréotypes UML, les diagrammes UML/SySML et les stéréotypes UMLsec. Il vise à modéliser les problèmes de sécurité dans les systèmes IoT. La figure suivante montre la place de l'IoTsec, il étend principalement UML, mais également SysML et inclut certains stéréotypes proposés dans UMLsec (21).

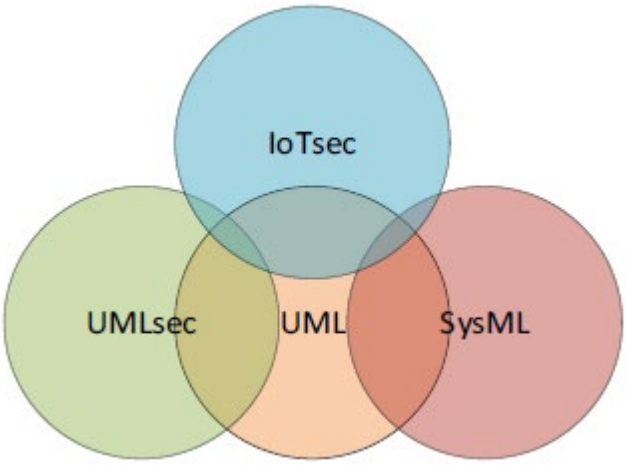


Figure 18 IoTsec parmi d'autres approches (21)

Il existe d'autres approches pour modéliser les systèmes IoT et les problèmes de sécurité, l'analyse de chacun a été faite sur les points suivants, et représentés dans le tableau suivant :

1. Extension spécifique à l'IoT.

2. Modèle de problèmes de sécurité du système.

3. Extension UML ou représentation visuelle.

4. Modélisation des exigences de sécurité.

Extension ou langue	1	2	3	4
UMLsec	X	✓	✓	✓
Iota	✓	X	✓	X
SysML	X	X	✓	X
SysMLsec	X	✓	✓	✓
UML4IOT	✓	X	✓	X
Approche IBM	✓	X	✓	✓
ThingML	✓	X	X	X
UML	✓	X	X	X
IoTsec	✓	✓	✓	✓

Figure 19 Comparaison des extensions (21)

IoTsec propose une nomenclature avec des problèmes de sécurité au sein de chaque élément. La nomenclature comprend quinze éléments :

- N : Authentification
- Z : Autorisation
- C : Chiffre
- D : Déchiffrer
- SS : Stockage sécurisé
- SC : Communication sécurisée

- B&B : courtier ou pont IoT
- T&R : Confiance et réputation
- KM : gestion des clés
- MI : gestion des identités
- Ps : Pseudonyme
- CA : autorité de certification
- RA : Autorité d'Enregistrement
- TP : Protection contre la falsification
- CC : contrôle personnalisé

Ces éléments sont utilisés à l'intérieur des diagrammes UML étendus, car ce sont des exigences de sécurité d'abstraction de haut niveau et ils encapsulent les piliers de la sécurité : Confidentialité, Intégrité et Disponibilité (CIA).

L'UML a treize diagrammes chaque diagramme étendu sera introduit dans le troisième chapitre.

4 Conclusion

Dans ce chapitre, on a focaliser sur la modélisation des systèmes IoT avec UML, ce dernier qui s'est imposé comme le langage de modélisation le plus utilisé dans l'industrie et dans le milieu universitaire, suite a une analyse générale de l'approche IoTsec avec une comparaison avec les autres approches qui modélisent les aspects de sécurité.

Dans le prochain chapitre, on va détailler notre approche proposée en se basant sur les principes de l'ingénierie dirigée par les modèles.

III.Chapitre 3 : IoTsec

1 Introduction

l'ingénierie logicielle s'oriente aujourd'hui vers l'ingénierie dirigée par les modèles (IDM), au sein de laquelle un système est vu non pas comme une suite de lignes de code, mais comme un ensemble de modèles plus abstraits et décrivant chacun une vue (c'est-à-dire une préoccupation) particulière sur le système.

Dans ce chapitre, on va proposer une approche pour modéliser les aspects de sécurité des systèmes IoT, en utilisant les concepts de base de la métamodélisation.

2 L'approche MDA

2.1 Principe général

Sur la base de la modélisation et des méthodologies de conception basées sur les modèles, l'OMG propose l'approche MDA, dérivée de l'IDM. Elle est construite sur les mêmes bases que l'IDM, à savoir : le métamodèle, le modèle et la transformation de modèle. L'idée de base du MDA est de séparer les spécifications fonctionnelles d'un système des détails de son implémentation sur une plate-forme donnée (25).

2.2 Définitions

2.2.1 Métamodèle

Un métamodèle est un modèle qui définit le langage d'expression d'un modèle (25), c.-à-d. le langage de modélisation. Autrement dit, le métamodèle représente ou modélise le langage. À titre d'exemple, le métamodèle d'UML offre des concepts permettant de décrire les différents modèles (diagramme de classes, diagramme de cas d'utilisation…) d'un système (26)

2.2.2 Types de modèles

MDA définit une architecture de spécification structurée en plusieurs types de modèles : (26)

- **CIM (Computational Independent Model)** : modélise les exigences d'un système, son but étant d'aider à la compréhension du problème, mais aussi de fixer un vocabulaire commun pour un domaine.
- **PIM (Platform Independent Model)** : connu aussi sous le nom de modèle d'analyse et de conception, est un modèle abstrait indépendant de toute plate-forme d'exécution. Le PIM a pour but de décrire le savoir-faire ou la connaissance métier d'une organisation.
- **PDM (Platform Description Model)** : modélise la plate-forme sur laquelle le système va être exécuté. Plus précisément, il définit les différentes fonctionnalités de la plate-forme et précise comment les utiliser.
- **PSM (Platform Specific Model) :** cible une plate-forme d'exécution spécifique en se basant sur les PDMs pour améliorer la portabilité et augmenter productivité. Il représente une vue technique détaillée du système.

2.2.3 L'architecture du MDA

Afin d'organiser et de structurer les modèles cités dans la section précédente, l'OMG a défini une architecture appelée : « Architecture à quatre niveaux » (27)comme le montre la figure suivante **:**

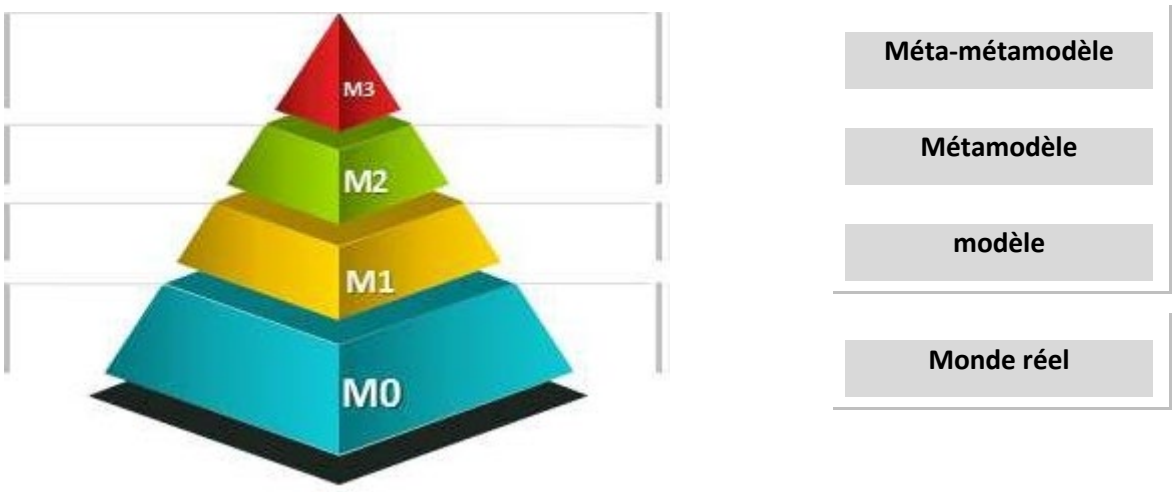


Figure 20 Architecture MDA (27)

- **Le niveau M0** : est le niveau des données réelles, est composé des informations que l'on souhaite modéliser. Ce niveau est souvent considéré comme étant le monde réel.
- **Le niveau M1** : est composé de modèles d'information. Lorsque l'on veut décrire les informations de M0.
- **Le niveau M2** : est donc composé de langages de définition des modèles d'information, appelés aussi méta-modèles. Typiquement, le méta-modèle UML qui est décrit dans le standard UML appartient au niveau M2 ; il définit la structure interne des modèles UML.
- **Le niveau M3** : est composé d'une unique entité qui est le langage unique de définition des métamodèles. Aussi appelé le méta-méta-modèle ou le MOF (*Meta-Object Facility*).

2.2.4 Les standards de l'OMG

Plusieurs standards existent dont l'approche MDA repose sur citant :

- **Le MOF :** Il fait partie des standards définis par l'OMG et il peut être vu comme un sous-ensemble d'UML, Le MOF spécifie la structure et la syntaxe de tous les métamodèles. Le modèle MOF est appelé un méta-métamodèle parce qu'il est utilisé pour définir des métamodèles tels que l'UML. (28)
- **UML :** déjà décrit dans le deuxième chapitre, il appartient au niveau M2 dans l'architecture MDA.
- **CWM() :** (*Common Warehouse Metamodel*) définit un Framework permettant de décrire des méta-données concernant des sources de données, transformations de données ainsi que des processus de gestion d'entrepôts de données**.** (28)
- **XMI() :** qui offre une représentation concrète des modèles sous forme de documents XML, en définissant un format d'échange de métadonnées. C'est sur XMI que reposent les techniques de sérialisation de modèles (29).

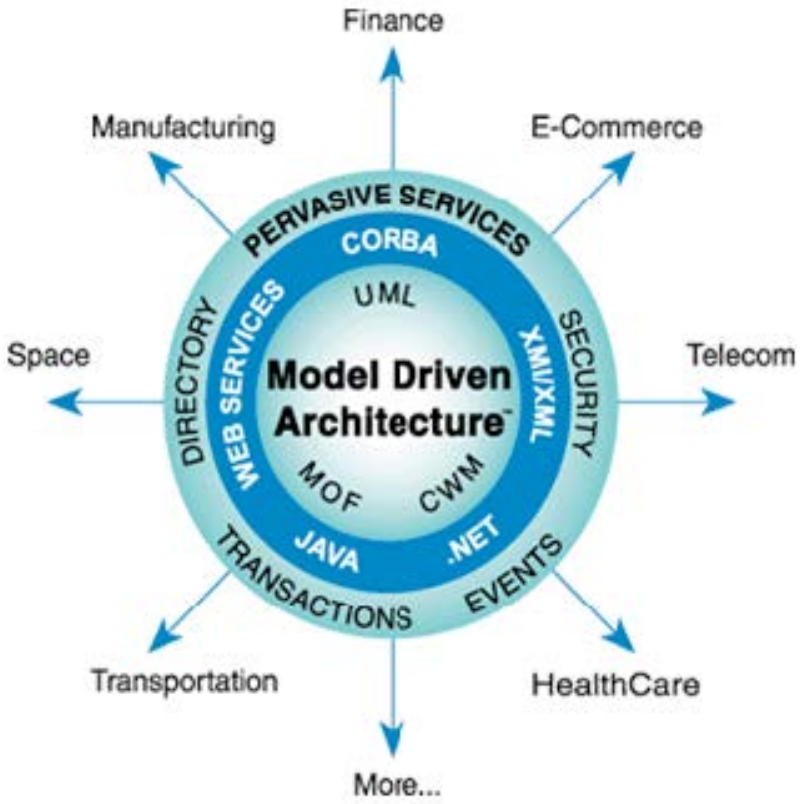


Figure 21 Architecture dirigée par les modèles (31)

3 Transformation de modèles

3.1 Aspect général

La transformation d'un modèle est le processus de conversion de ce modèle en un autre modèle (relatif au même système). Les transformations sont la base de l'approche MDA : elles permettent d'obtenir différentes vues d'un modèle, de raffiner ou d'abstraire un modèle, ou encore de réécrire un modèle dans un langage différent (31).

Les transformations possibles entre ces différents types de modèles sont représentées sur la figure ci-dessous :

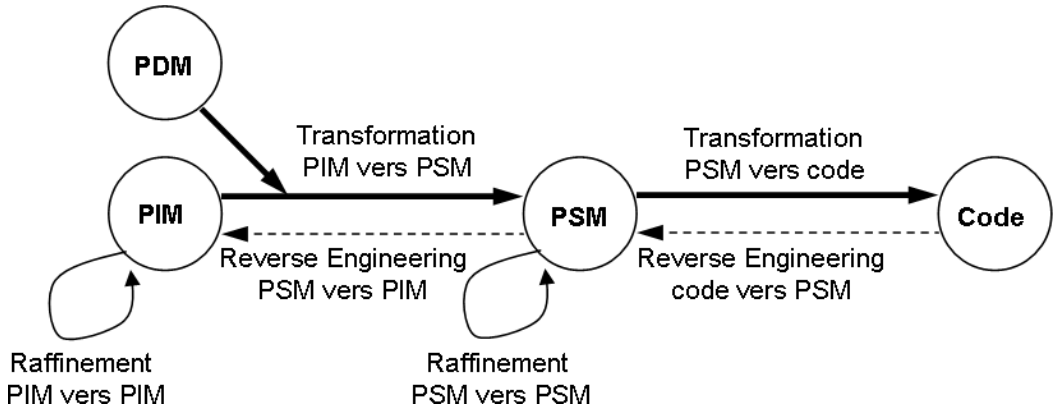


Figure 22 Transformation de modèles

- **Transformations PIM PIM et PSM PSM ,**

Les transformations de ce type visent à spécialiser le modèle. Il s'agit de transformations de modèle à modèle elles sont automatisables (ou partiellement automatisable).

- **Transformation PIM PSM :**

La transformation de PIM vers PSM permet de spécialiser le PIM en fonction de la plate-forme cible choisie. Cette transformation de modèle à modèle est réalisée en s'appuyant sur les informations fournies par le PDM. (32)

Cette transformation est une transformation de type modèle à texte, elle n'est pas toujours suffisante pour permettre la génération de code.

- **Transformations inverses PIM PSM et PSM code :**

C'est une opération de rétro ingénierie (*reverse engineering*) qui est assez complexe à réaliser et difficilement automatisable. Ces transformations sont néanmoins nécessaires pour permettre l'intégration d'applications existantes dans le processus MDA.

3.2 Caractéristiques de transformation de modèles

3.2.1 Utilisation de paramètres

Il n'y a pas de façon unique pour effectuer une transformation de modèles. Pour en choisir concernant, il peut y avoir recours à l'utilisation de paramètres dont l'utilisateur spécifie les valeurs. Ces choix peuvent aider à compléter un modèle, ils servent aussi à optimiser le modèle cible produit par l'outil.

3.2.2 Traçabilité

Pour tout élément généré dans le modèle cible, la traçabilité permet de retrouver l'élément générateur dans le modèle source.

3.2.3 La cohérence incrémentielle

L'ajout manuel d'information au modèle cible est conservé lors d'une génération ultérieure du modèle cible.

3.2.4 La bidirectionnalité

Elle consiste en la possibilité de retrouver le modèle source par une transformation inverse. Malheureusement ceci est généralement inaccessible, car en passant à un niveau d'abstraction moins élevé, on ajoute des détails moyennant des choix pris par défaut ou même des choix de valeurs de paramètres influençant la transformation (33).

3.3 Types de transformations de modèles

Selon (32), on distingue deux grandes familles de s transformations :

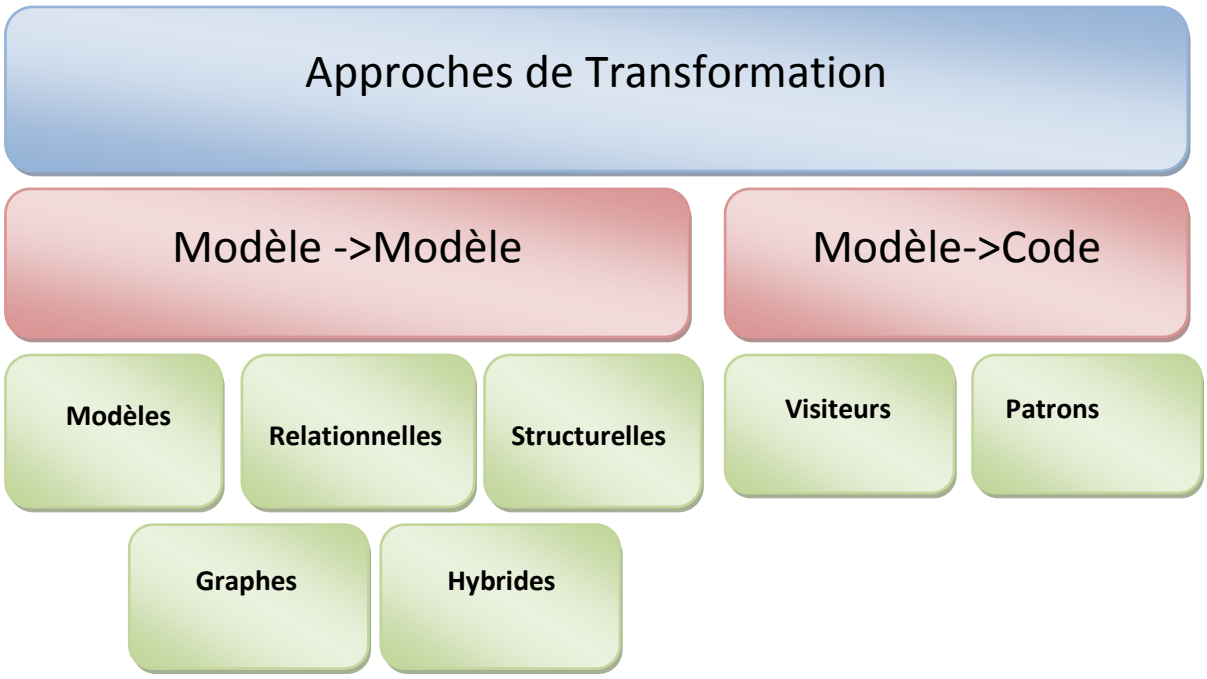


Figure 23 Types de transformations

3.3.1 Transformations de type modèle vers modèle

Les transformations de modèle à modèle traduisent entre les modèles source et cible, qui peuvent être des instances de métamodèles identiques ou différents. Les techniques de transformations de ce type peuvent être classées en cinq catégories :

- **Approches manipulant directement les modèles :** Ces approches offrent une représentation interne en se basant sur l'utilisation des APIs(*Application Programming Interface*) pour les manipuler.
- **Approches relationnelles :** L'idée de base est d'indiquer le type d'élément source et cible d'une relation et de la spécifier à l'aide de contraintes.
- **Approches guidées par la structure** : Les approches de cette catégorie comportent deux phases distinctes : la première phase concerne la création de la structure hiérarchique du modèle cible, tandis que la seconde phase définit les attributs et les références dans la cible.
- **Approches basées sur la transformation de graphes** : Cette catégorie est fondée sur les travaux théoriques des transformations de graphes. En

particulier, ces approches opèrent sur des graphes typés, attribués et étiquetés

- **Approches hybrides :** les approches hybrides combinent différentes techniques des catégories précédentes. On peut notamment retrouver des approches utilisant à la fois des règles déclarative et impérative.

3.3.2 Transformations de type modèle vers code

- **Approches basées sur les visiteurs** : consiste à fournir un mécanisme de visiteur pour traverser la représentation interne d'un modèle et écrire du code dans un flux de texte.
- **Les approches basées-patrons :** Ces approches sont actuellement les plus utilisées dans les outils MDA, basées sur l'utilisation des morceaux de méta-code dans le code cible, utilisés pour accéder aux informations du modèle source.

4 L'approche proposée

4.1 Vue générale de l'approche

en raison des différentes limites trouvées dans le domaine de l'IoT tel que : la faible capacité de calcul et de stockage des équipements IoT et insuffisance d'énergie, on est donc face a un environnement hautement incontrôlé, éventuellement sans surveillance d'où la nécessité d'aborder les aspects de sécurité lors de la phase de modélisation de ces systèmes .

Dans ce, nous allons proposer un Framework basé sur une extension UML pour la modélisation des aspects de sécurité dans un système IoT, appelée IoTsec. Le Framework comporte plusieurs diagrammes pour permettre une modélisation multivues du système.

4.2 Papyrus

4.2.1 Définition

Eclipse Papyrus est une plateforme de langage spécifique au domaine (DSL) basée sur le langage de modélisation standard le plus répandu, le langage de modélisation unifié (UML). Cette application open source a deux objectifs principaux. Premièrement, elle vise à mettre en œuvre la spécification UML complète (actuellement la version 2.5), ce qui lui permet d'être utilisée comme implémentation de référence pour la norme de l'Object Management Group (OMG)). Deuxièmement, il vise à fournir un outil ouvert, robuste, hautement évolutif et hautement personnalisable pour définir les DSL et les outils correspondants. Pour ce faire, il utilise le mécanisme de profil UML ainsi que de puissantes fonctions de personnalisation de l'interface utilisateur**. (34)**

4.2.2 Discussion sur l'outil

- Papyrus permet aux utilisateurs de bénéficier à la fois des avantages des intérêts bien connus de l'utilisation de solutions basées sur des normes et de l'efficacité des solutions de modélisation spécifiques à un domaine.
- Papyrus vise également à soutenir les projets industriels à grande échelle. Il offre une alternative efficace et efficiente aux outils DSL personnalisés et propriétaires, sans perdre les avantages d'une norme internationale.
- Papyrus peut également servir de plate-forme expérimentale pour les chercheurs qui construisent des prototypes de validation de concept. Construit

sur Eclipse en tant que projet open source, Papyrus est un candidat idéal à cette fin.

Il y a actuellement quelques limitations identifiées ainsi que quelques bogues mineurs :

- Prise en charge des profils/stéréotypes : Il n'est pas possible de restreindre un sélecteur à un stéréotype appliqué. Il n'existe actuellement aucun moyen de spécifier qu'un style ne doit s'appliquer qu'aux classes sur lesquelles le stéréotype est appliqué.
- Il n'est pas possible d'appliquer un style à tous les descendants d'une métaclasse donnée. Par exemple, un comportement n'héritera pas des styles d'une classe, bien qu'un comportement soit en fait une classe.

4.3 Diagrammes UML étendus pour le profil IoTsec

L'UML a treize diagrammes, nous avons opté pour les plus basiques entre eux : Diagramme de cas d'utilisation, Diagramme de classe, diagramme de déploiement, diagramme de composants et diagramme d'états – transitions. L'objectif est de fournir un Framework basé IoTsec et qui permet la spécification de la sécurité d'un système IoT de différentes vues.

4.3.1 Diagramme de cas d'utilisation IoTsec

- **Définir le méta modèle**

- Il ya quatre catégories d'acteurs de l'IoT ont été identifiées dans presque tous les domaines : capteur, actionneur, humain et IoTdevice qui vont hérités de stéréotype « Actor ».
- La nomenclature (éléments de sécurité) peut être présentée comme cas d'utilisation si les exigences de sécurité le demandent, qu'on va illustré dans notre étude de cas.
- Spécifier les méta classe des différents stéréotypes de notre profil du diagramme de cas d'utilisation étendu, ici on a la méta-classe Actor et Use case d'UML.

- Personnalisation des différents stéréotypes du profil par le biais d'icône ajouté aux différents éléments stéréotypés.
- Et enfin enregistrer le profil obtenu pour le manipuler et le vérifier.
- La figure 25 montre le méta modèle proposé pour le diagramme de cas d'utilisation IoTsec.

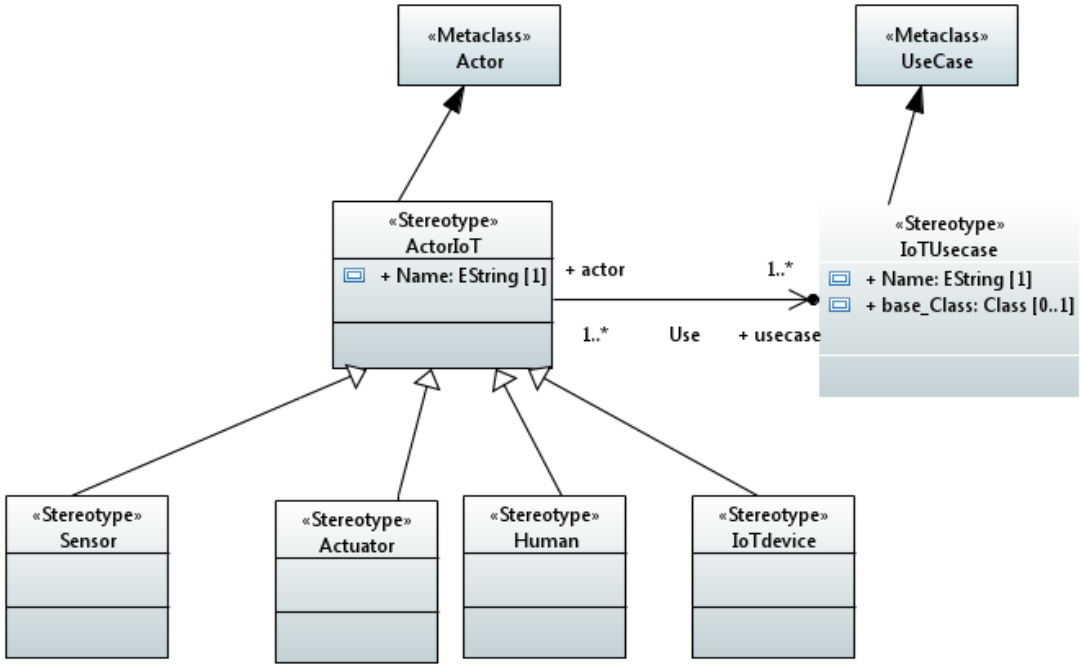


Figure 24 Métamodèle de diagramme UsecaseIoTsec

- La figure 26 illustre l'environnement généré à partir de ce métamodèle, et qui permet de modéliser n'importe quel diagramme de cas d'utilisation IoTsec.

❖ Exemple de diagramme de cas d'utilisation IoTsec modélisé

Maintenant on va appliquer le nouveau profil IoTsec qui prend en charge les nouveaux stéréotypes qu'on a définis dans le métamodèle précédent :

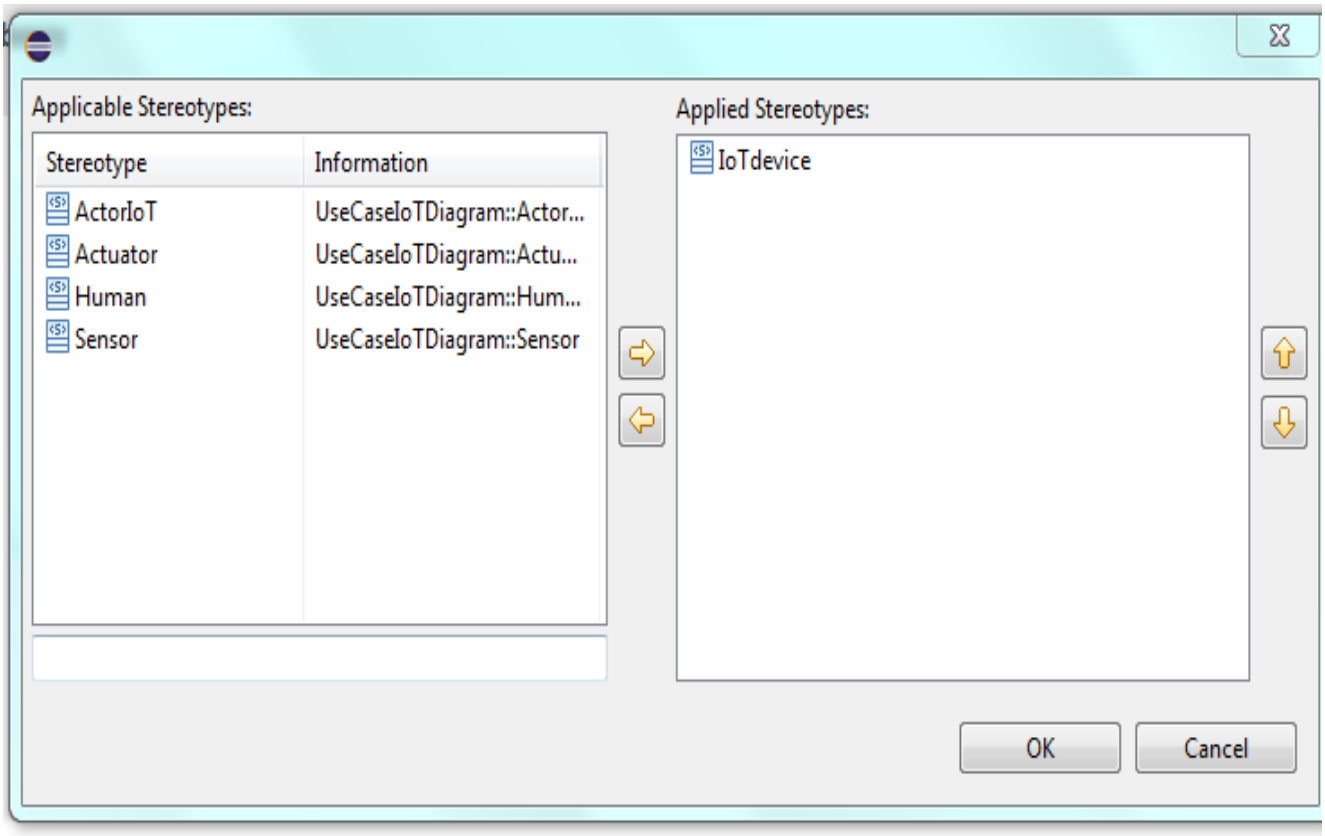


Figure 25 stéréotypes appliqués

- Il existe un IoTdevice qui doit authentifier, autoriser et chiffrer des données.
- Donc ces cas d'utilisation appliquent les éléments de nomenclature N, Z, C, ainsi que le stéréotype <<IoTdevice>> une restriction de pseudonyme (Ps).
- Néanmoins, D, N, KM et IM pourraient également être modélisés comme des cas d'utilisation. Cela permettra un processus de conception plus agile pour les exigences de sécurité, car même les développeurs qui ne sont pas impliqués dans la sécurité peuvent reconnaître ces éléments.

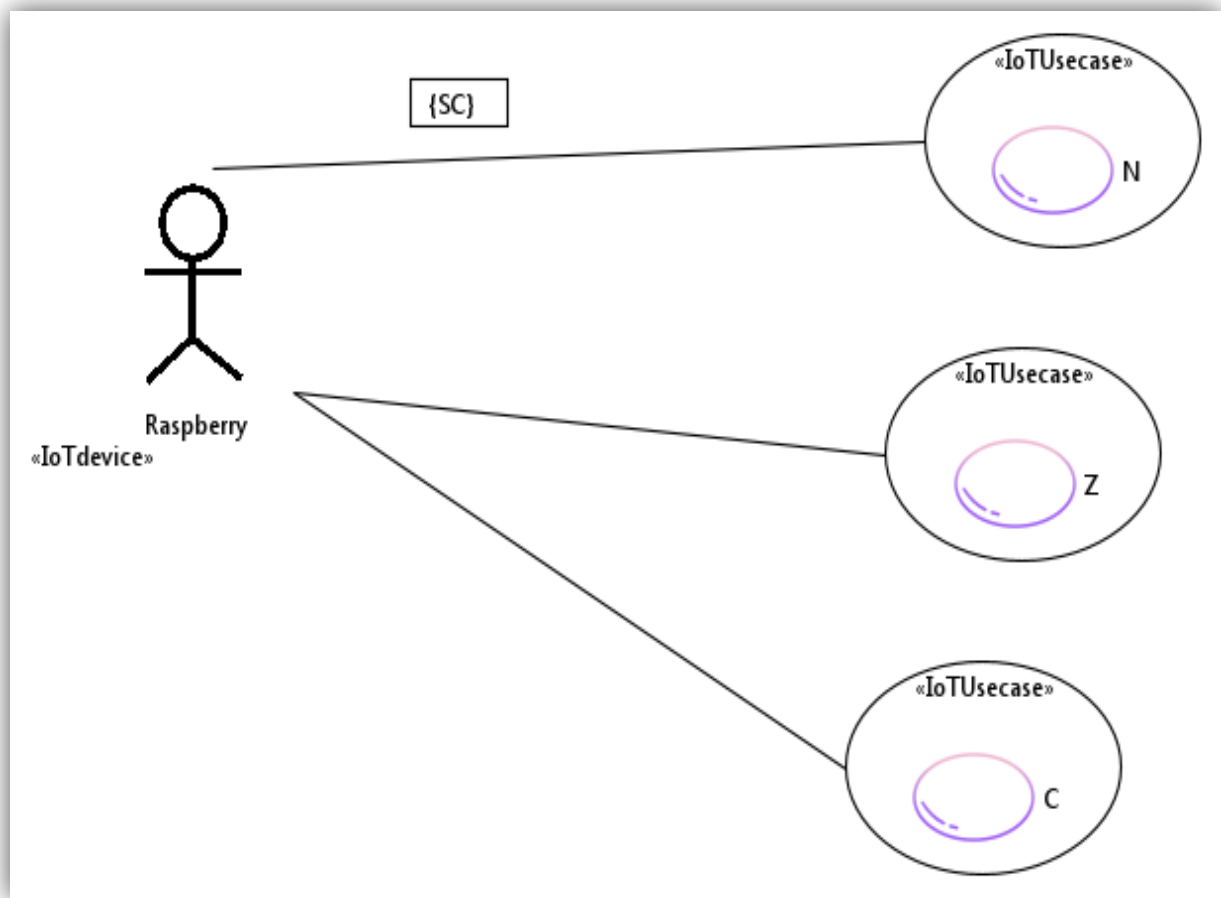


Figure 26 : Diagramme de cas d'utilisation IoTsec

4.3.2 Diagramme de classe IoTsec

❖ **Définir le méta modèle**

- Ici les différents stéréotypes des acteurs vont hérités de la méta classe « class » d'UML.
- Aussi les différents liens entre les éléments IoT héritent de la méta classe « class ».
- La figure 27 montre le méta modèle proposé pour le diagramme de classe IoTsec.

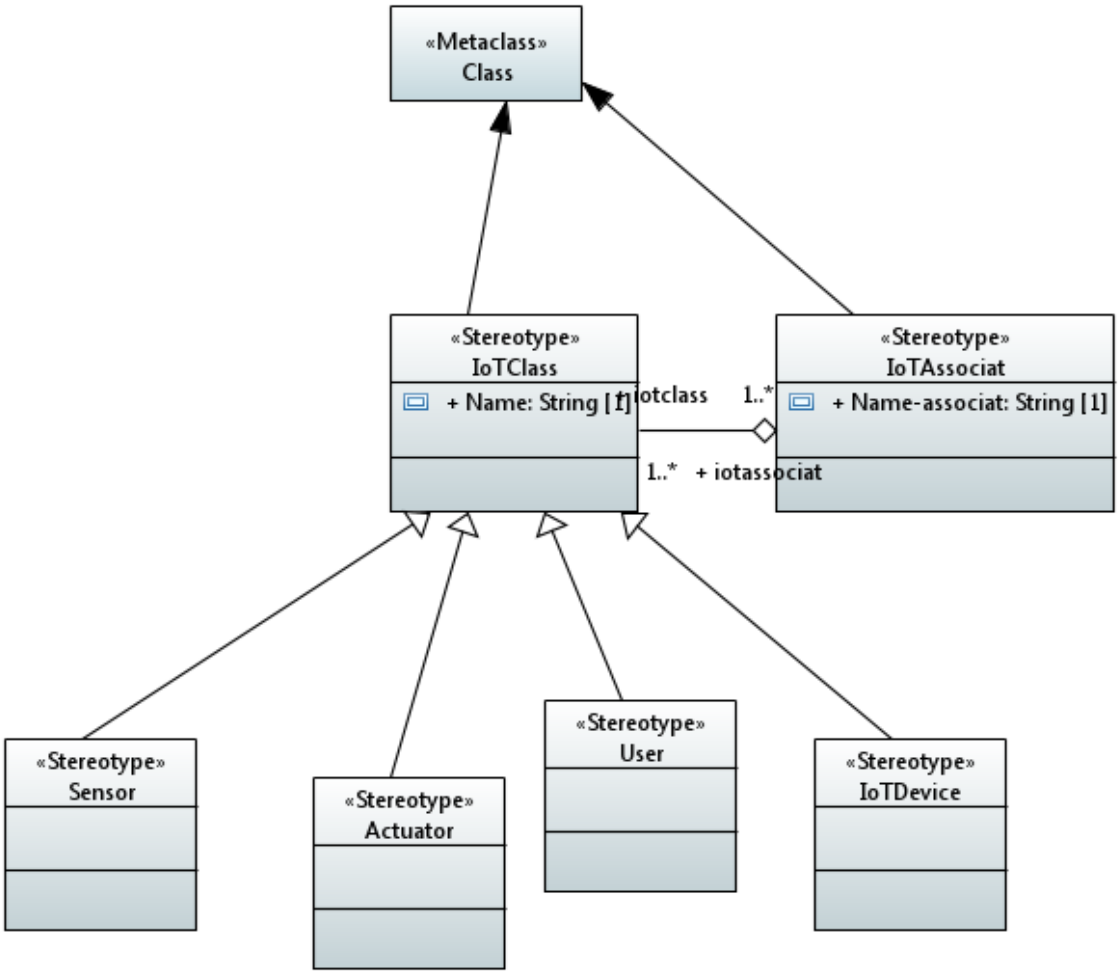


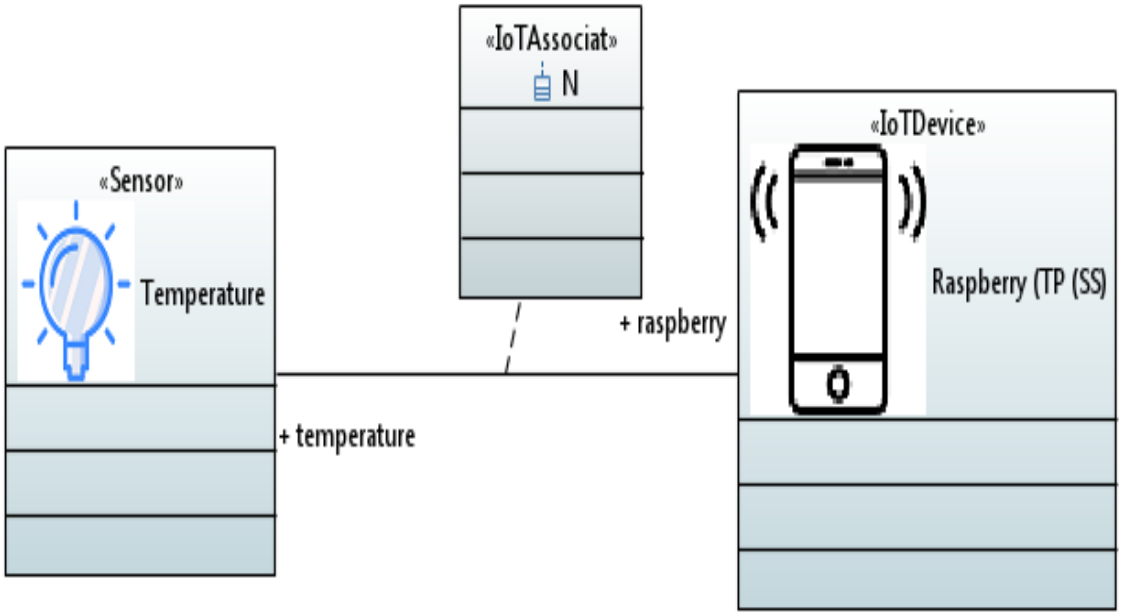
Figure 27 Métamodéle de diagramme Classe IoTsec

❖ **exemple**

- Dans le Diagramme de classe, les éléments de la nomenclature peuvent être modélisés en tant que classes.

- Les stéréotypes des acteurs sont appliqués avec des contraintes SS (Stockage sécurisé) et TP (Tampering protection).

- Un appareil IoT nommé Raspberry authentifie un capteur de température en utilisant une classe d'association N. (ses attributs pourraient être n'importe quel protocole d'authentification.



- **Figure 28 Diagramme de classe de IoTsec**

- La figure 28 illustre l'environnement généré à partir de ce métamodèle, et qui permet de modéliser n'importe quel diagramme de classe IoTsec.

4.3.3 Diagramme de séquence IoTsec

❖ **Définir le métamodèle**

La figure 29 montre le méta modèle proposé pour le diagramme de séquences IoTsec.

❖ **Exemple**

- les classes mentionnées précédemment qui appliquent la nomenclature apparaissent dans le diagramme de séquence en tant qu'objets.

- Un protocole d'authentification est illustré dans ce diagramme, il implique l'objet T&R, l'objet IoTdevice et sensor.
- SC décrit ici une exigence de communication sécurisée entre les objets.

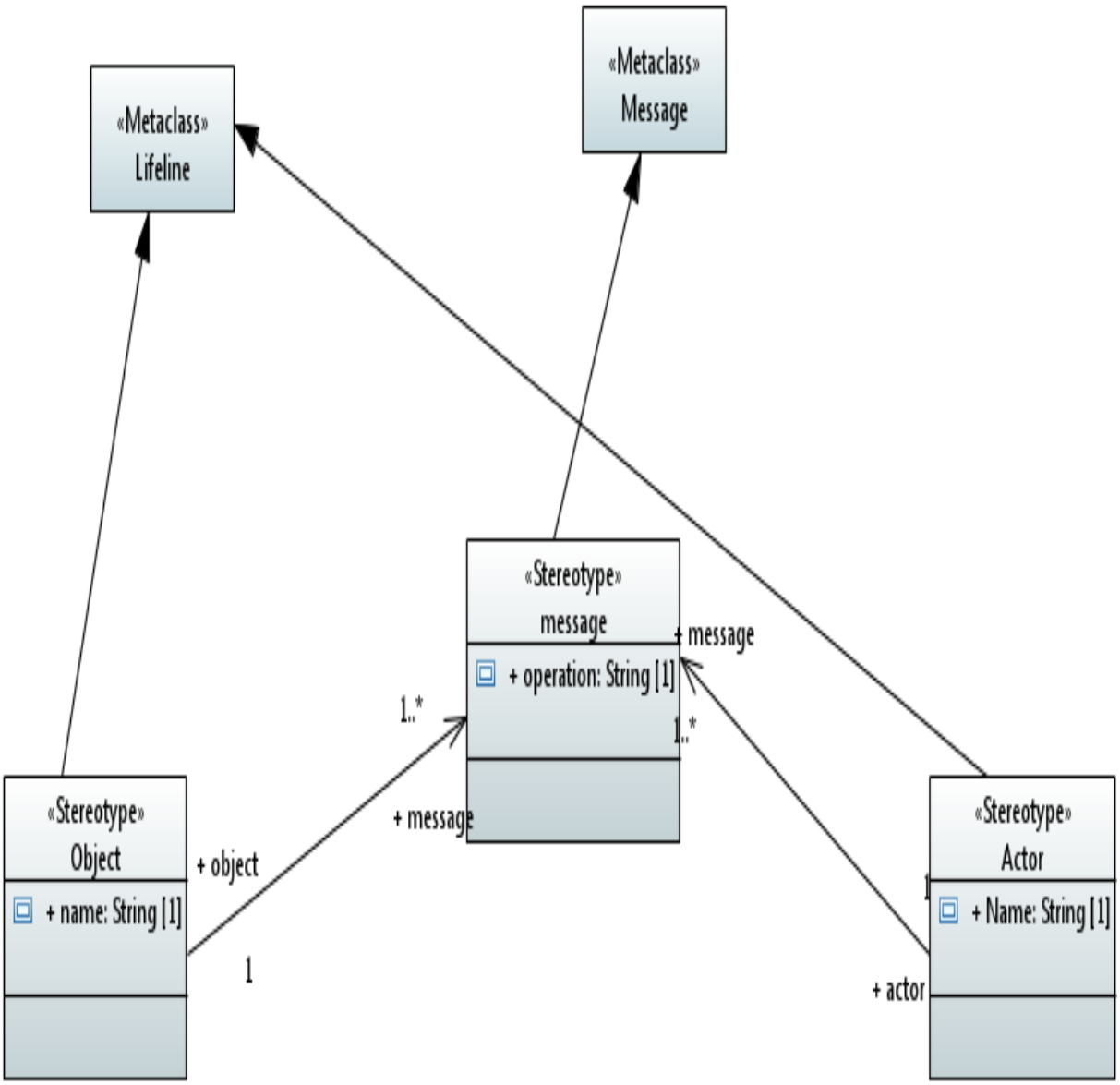


Figure 29 Métamodèle de Diagramme de séquences

- La figure 30 illustre l'environnement généré à partir de ce métamodèle, et qui permet de modéliser n'importe quel diagramme de séquences IoTsec.

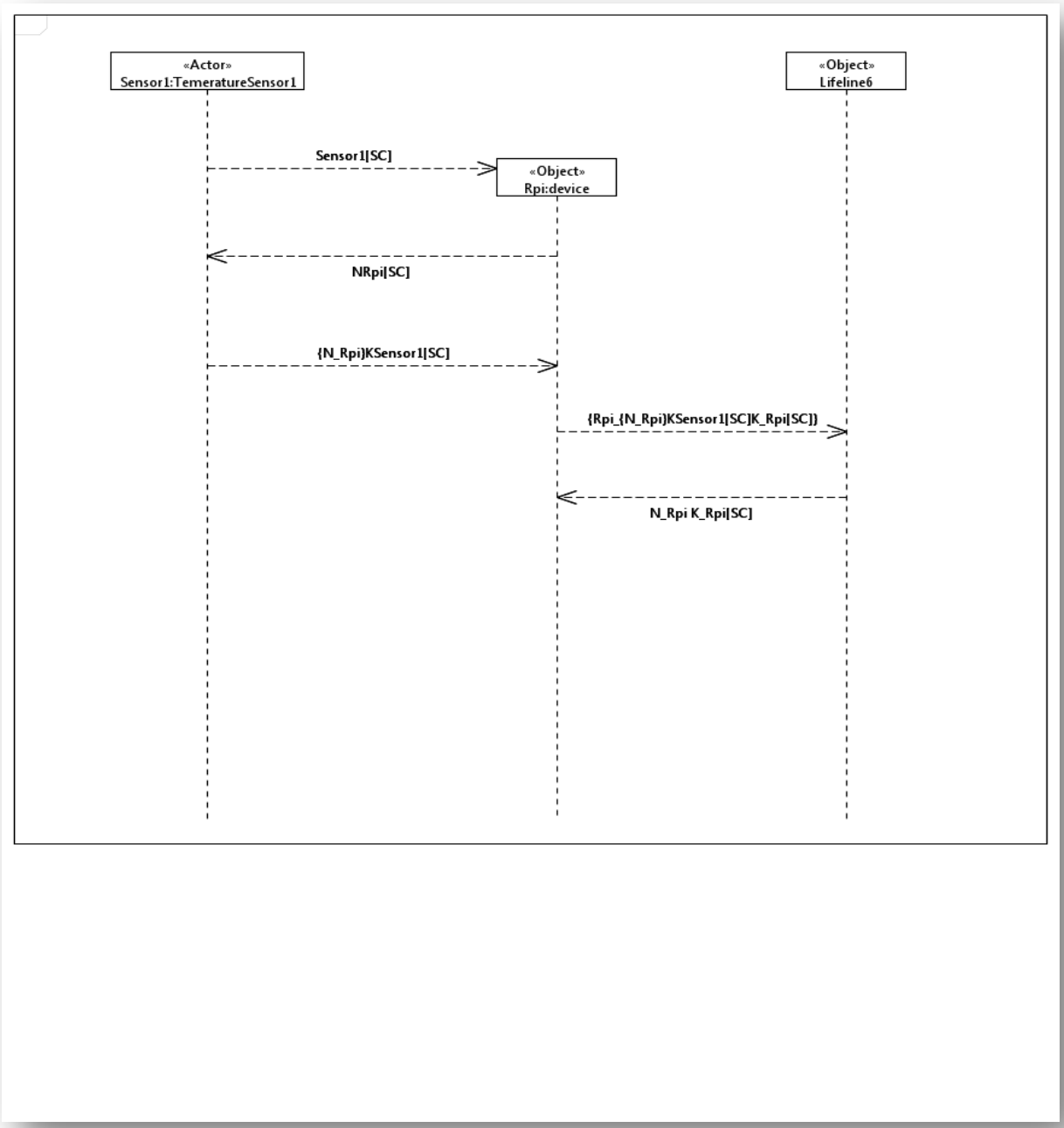


Figure 30:Diagramme de séquences IoTsec

4.3.4 Diagramme de Composants

- **Définir le métamodèle**

- Les stéréotypes composant et composant complexe sont générés a partie de la méta classe « Component » d'UML.
- À ce stade, ce qu'on a vraiment besoin c'est les éléments du nomenclature utilisés dans le diagramme de classes et qui peuvent être facilement migrés vers les diagrammes de composants**.**
- Une personnalisation des stéréotypes est possible par le mécanisme d'icône de Papyrus**.**

- La figure 31 montre le modèle proposé pour le diagramme de composants IoTsec.

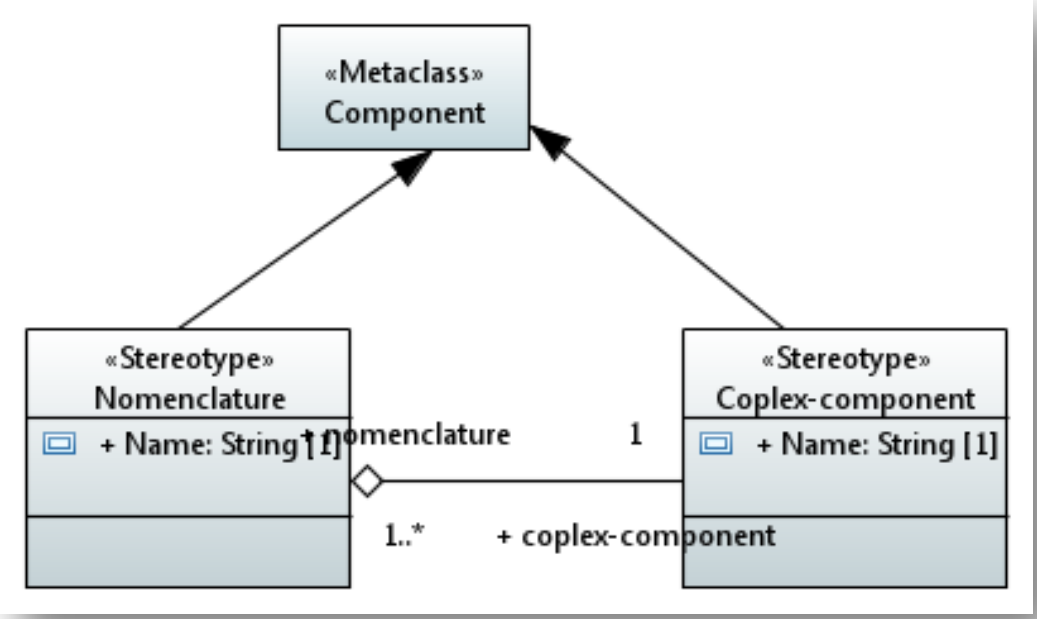


Figure 31 Métamodèle de Diagramme de Comosants

❖ **Exemple :**

- certains composants de sécurité sont représentés, il y a un composant KM qui a une communication avec le contrôle d'accès.
- Ce dernier composant comprend Z et N.
- le contrôle d'accès reçoit également des informations de T&R et N reçoit d'IM les identités nécessaires à son processus..
- La figure 32 illustre l'environnement généré à partir de ce métamodèle, et qui permet de modéliser n'importe quel diagramme de composants IoTsec.

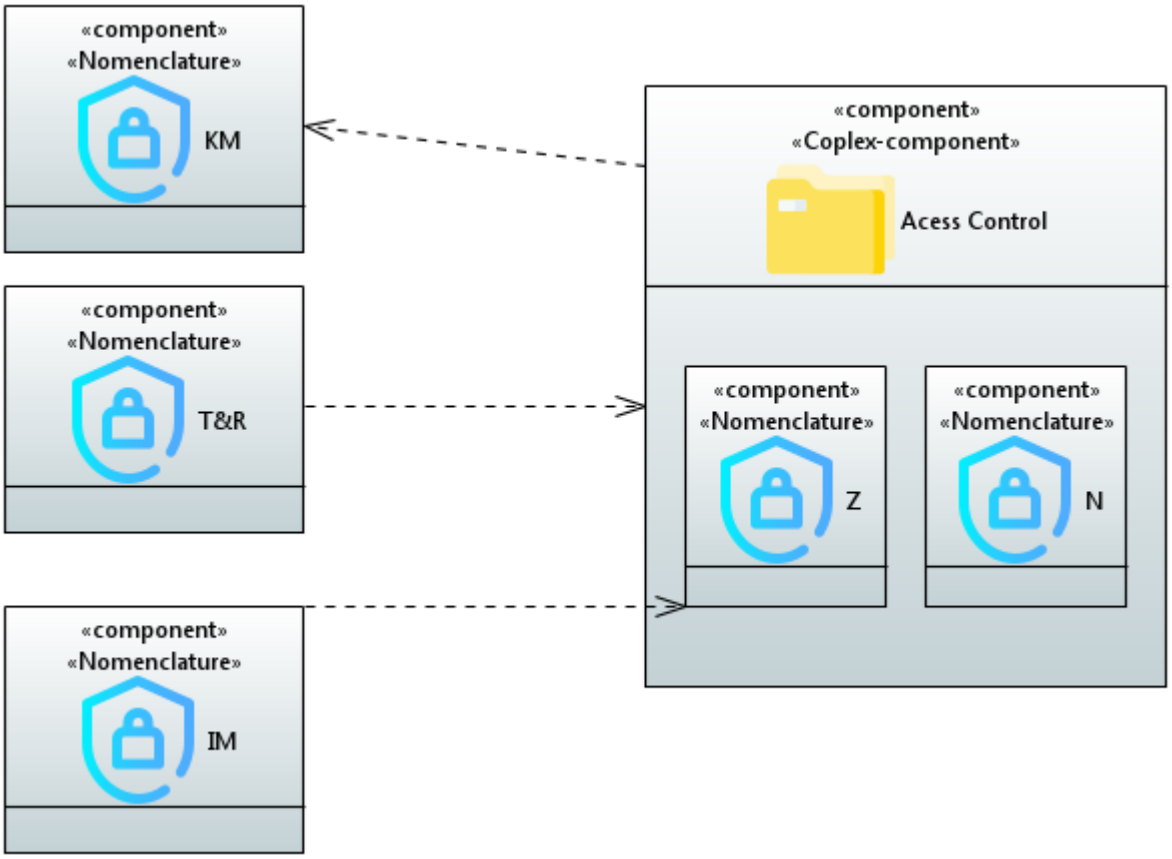


Figure 32 Diagramme de composants IotSec

4.3.5 Diagramme de déploiement IoTsec

- Définir le méta modèle

- Dans cette étape, on va stéréotyper les équipements qu'on trouve dans une architecture matérielle : Sensor, Actuator , IoTdevice et Device.
- Ces stéréotypes héritent de la métaclasse « Device » d'UML.
- La figure 29 montre le méta modèle proposé pour le diagramme de déploiement IoTsec.

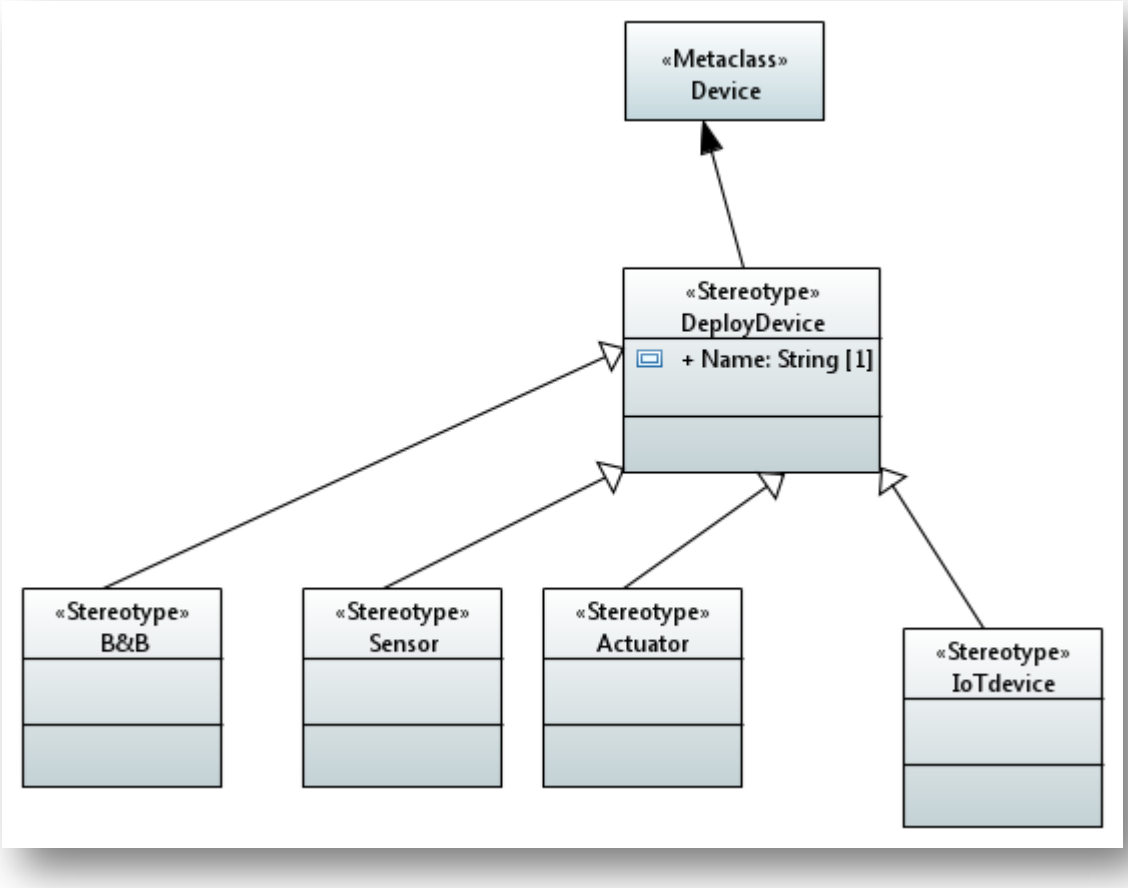


Figure 33 Métamodéle de Diagramme de déploiement IoTsec

❖ **Exemple :**

- Le diagramme de déploiement montre l'architecture matérielle du système, les différents stéréotypes IoT sont utilisés comme le montre l'exemple.
- La communication sécurisée entre les différents équipements est illustrée par les contraintes SC et SS.
- La figure 34 illustre l'environnement généré à partir de ce métamodèle, et qui permet de modéliser n'importe quel diagramme de déploiement IoTsec.

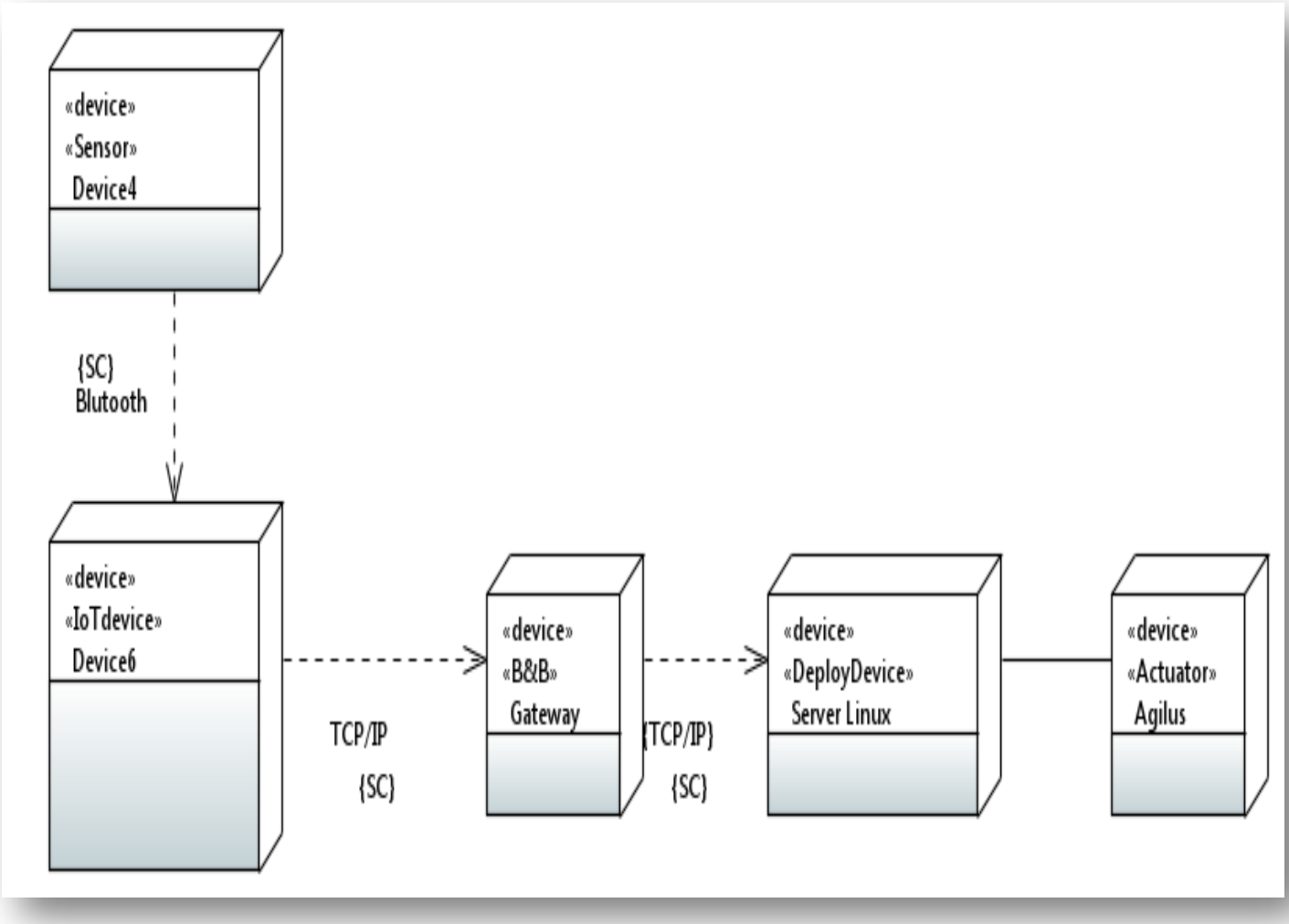


Figure 34 : Diagramme de déploiement IoTsec

4.3.6 Diagramme d'état IoTsec

❖ **Définir le méta modèle**

- Les stéréotypes Start_request et End_Request héritent de la classe State , sa métaclasse est la classe State d'UML.
- Stéréotype transition est défini par la métaclasse Transition d'UML.
- La figure 35 montre le méta modèle proposé pour le diagramme d'état IoTsec.

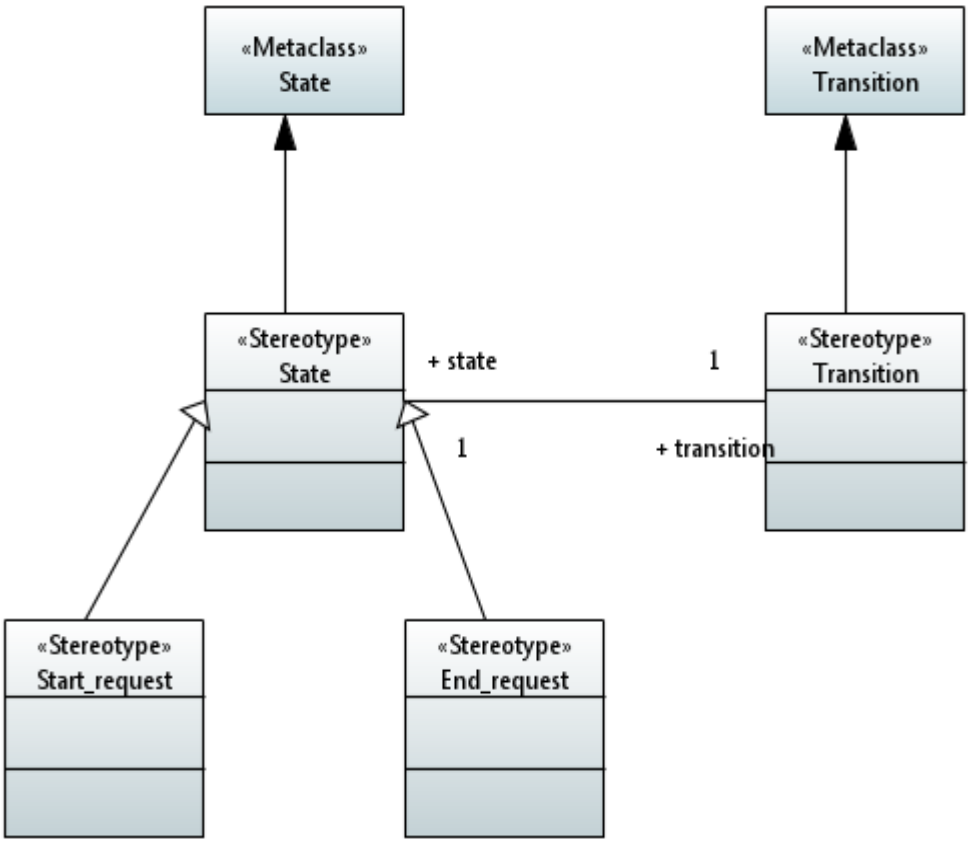


Figure 35 Métamodèle de Diagramme d'état

❖ Exemple

- Ici un exemple de mécanisme d'authentification est représenté, et montre le comportement d'un objet depuis sa création jusqu'à sa destruction.
- Les éléments de nomenclature qui apparaissent dans la figure si dessous sont les contraintes SS et SC.
- La figure 36 illustre l'environnement généré à partir de ce métamodèle, et qui permet de modéliser n'importe quel diagramme d'état IoTsec.

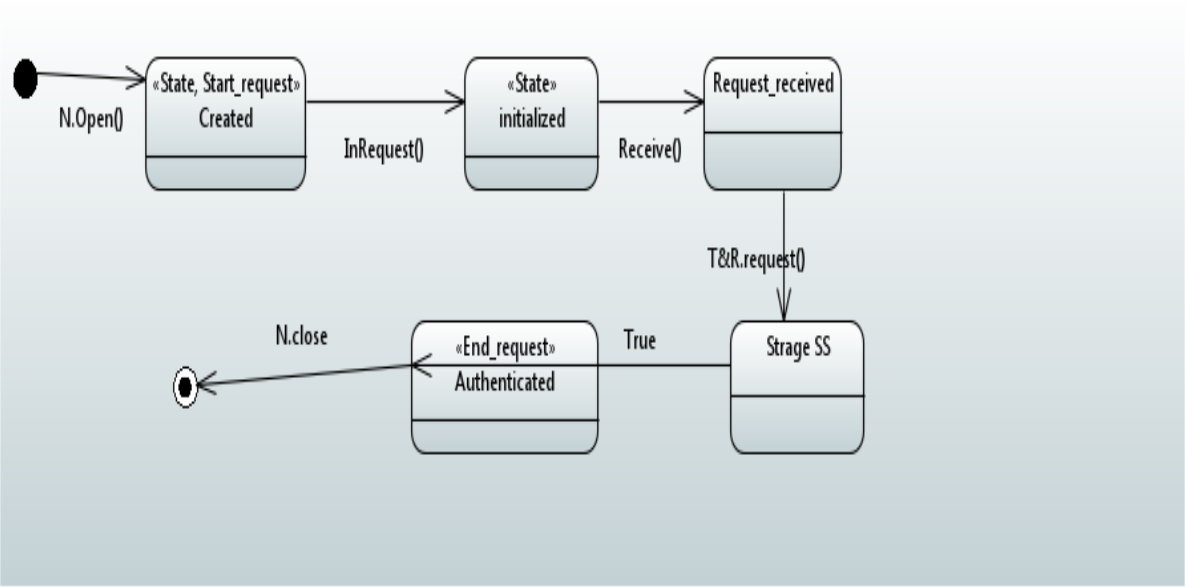


Figure 36 Diagramme d'état IoTsec

5 Conclusion

Dans ce chapitre, nous avons proposé un Framework pour modéliser les aspects de sécurité des systèmes IoT, on se basant sur une extension IoTsec qui étend UML pour modéliser les aspects de sécurité dans un systume IoT. Le framework est composé de plusieurs diagrammes pour permettre une modélisation de ces systèmes selon plusieurs vues.

IV. Conclusion et perspectives

❖ Conclusion

Dans ce mémoire, nous avons travaillé dans le domaine de IoT« Internet of Things » qui représente une idéologie globale qui supprime les frontières entre le monde physique et le monde virtuel, et qui couvre la plupart des systèmes informatiques et technologies d'information, et en raison des différentes limites trouvées dans ce domaine tel que : la faible capacité de calcul et de stockage des équipements IoT et insuffisance d'énergie, on est donc face a un environnement hautement incontrôlé, éventuellement sans surveillance d'où la nécessité d'aborder les aspects de sécurité lors de la phase de modélisation de ces systèmes .

Nous avons proposé un framework basé sur une extension UML pour la modélisation des aspects de sécurité dans un système IoT, appelée IoTsec. Le Framework comporte plusieurs diagrammes pour permettre une modélisation multivues du système.

Pour atteindre notre objectif, nous avons tout d'abord fait une étude bibliographique pour choisir un langage riche pour modéliser les aspects de sécurité dans un système IoT. Nous avons opté pour IoTsec car c'est le plus facile à comprendre et à utiliser notamment pour les non-spécialisés. Après nous avons chois plusieurs diagrammes ((Diagramme de classe, Diagramme de cas d'utilisation, Diagramme de séquences, Diagramme de composants, Diagramme de déploiement, Diagramme d'états-transitions) dans le but de fournir une modélisation complète qui prend en considération toutes les vues du système. Nous avons ensuite opté pour l'utilisation de Papyrus qui nous a permet de réaliser nos idées et par conséquent d'implémenter notre Framework. Nous avons utiliser la méta-modélisation pour spécifier les méta-modèles des diagrammes choisis et de faire générer leurs environnements par la suite. Nous avons essayé d'illustrer notre approche par des exemples concrets.

❖ **Perspectives**

Comme perspective, le travail est le point de départ d'un langage de modélisation plus robuste pour les systèmes IoT en ce qui concerne la sécurité, nous recherchons également des scénarios de validation plus réels.

dans un futur travail, nous envisageons de :

- Compléter notre étude par l'élaboration des autres diagrammes du standard UML en se focalisant de la même façon sur les aspects de sécurité des systèmes IoT.
- Doté le Framwork IoTsec par un langage formel qui permet l'analyse et la vérification par la suite.

V. Références bibliographiques

1. Nemri, M. (2015). Demain l'internet des objets. *France Stratégie, Note d'analyse*.

2. Evans, D. (2011). L'Internet des objets. *The Internet of Things. The Cisco White, Aprill*.

3. Hadji, H(2020). *Les fondamentaux de.* Corée du Sud : International Telecommunication Union "ITU".

4. https://www.pdfprof.com/PDF_Image.php?idt=50427&t=37.

5. Patrice W. (2015). *Objets connectés.* Laboratoire MIPS Université de Haute-Alsace. mulhouse : s.n.

6. Kassaa, P (2019) Thése. *L'internet des Objets (IoT) et les réseaux haut et bas débit, est-ce un outil essentiel pour une ville intelligente et moins Polluée ?* Beyrouth : OGERO Telecom.

7. Martin, A. (2019). IOTFLA: une architecture de domotique sécurisée respectueuse de la vie privée. 8. LEMOINE, Frédéric.Thèse. *Internet des Objets centré service autocontrôlé.* CNAM École doctorale Informatique, Télécommunications et Électronique. Paris : s.n., 2019.

9. EL JAOUHARI,S (2016). *La sécurité des objets connectés.*. Rennes : s.n., 2016, Vol. pp. 54-58. MISC N°88.

10. Cordier, F. *https://fr.readkong.com/page/la-s-curit-de-l-internet-des-objets-ou-iot-c-est-parti-8479812.*

11. Vasilomanolakis,E. Daubert,J.(2015) *Sur la sécurité et la confidentialité des architectures et des systèmes de l'Internet des objets.*

12. *Saidi, A., Kacem, M. H., Tounsi, I., & Kacem, A. H. (2021). A Meta-Modeling Approach to Describe Internet of Things Architectures. In TACC (pp. 25-36).*

13. https://fr.wikipedia.org/wiki/UML_(informatique). [En ligne]

14. Fowler, M. (2004). *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley Professional.

15. Piechocki, Laurent. *www.developpez.com.* [En ligne] 2009.

16. Arlow, J., & Neustadt, I. (2005). *UML 2 and the unified process: practical object-oriented analysis and design*. Pearson Education.

17. *https://medium.com/@manurnx/le-diagramme-de-classes-2447602613f2.*

18. *https://www.developpez.net/forums/d1895698/general-developpement/alm/modelisation/uml/cas-d-utilisation.*

19. Nassar, M., Coulette, B., Guiochet, J., Ebersold, S., El Asri, B., Crégut, X., & Kriouile, A. (2005). Vers un profil UML pour la conception de composants multivues. *Revue des Sciences et Technologies de l'Information-Série L'Objet: logiciel, bases de données, réseaux*, *11*(4).

20. https://www.omg.org/spec/UML/2.4.1/.

21. *Robles-Ramirez, D. A., Escamilla-Ambrosio, P. J., & Tryfonas, T. (2017, November). IoTsec: UML extension for internet of things systems security modelling. In 2017 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE) (pp. 151-156). IEEE.*

22. Friedenthal, S., Moore, A., & Steiner, R. (2008). OMG SysML TM Specification• Specification status. *Management*.

23. Jürjens, J. (2002, September). UMLsec: Extending UML for secure systems development. In *International Conference on The Unified Modeling Language* (pp. 412-425). Springer, Berlin, Heidelberg.

24. Apvrille, L., & Roudier, Y. (2013). SysML-Sec: A SysML environment for the design and development of secure embedded systems. *APCOSEC, Asia-Pacific Council on Systems Engineering*, 8-11.

25. Combemale, B. (2008). Ingénierie Dirigée par les Modèles (IDM)--État de l'art.

26. Diaw, S., Lbath, R., & Coulette, B. (2010). État de l'art sur le développement logiciel basé sur les transformations de modèles. *Tech. Sci. Informatiques*, *29*(4-5), 505-536.

27. Bézivin, J., & Blanc, X. (2002). MDA: Vers un important changement de paradigme en génie logiciel. *Développeur référence v2*, *16*, 15. 28. *Meta Object Facility specification.* V1.3, OMG-MOF. 2000.

29. Hardebolle, C. (2008). *Composition de modèles pour la modélisation multi-paradigme du comportement des systèmes* (Doctoral dissertation, Université Paris Sud-Paris XI).

30. *http://www.omg.org/mof/.».*

31. *Czarnecki, K., & Helsen, S. (2006). Feature-based survey of model transformation approaches. IBM systems journal, 45(3), 621-645.*

32. ABD-ALI, J. A. M. A. L. (2006). MÉTAMODÉLISATION ET TRANSFORMATION AUTOMATIQUE DE PSM DANS UNE APPROCHE MDA.

33.Gérard,S.*https://www.eclipse.org/community/eclipse_newsletter/2016/april/article1.php.*

34. httpsfr.linkedin.compulseinternet-des-objets-ou-web-40-4%C3%A8me-g%C3%A9n%C3%A9ration-de-sika-technologie. [En ligne]

35. Benghozi, P. J., Bureau, S., & Massit-Folea, F. (2008). L'Internet des objets. Quels enjeux pour les Européens ?

36. httpswww.researchgate.netfigureThe-Basic-Architecture-of-IoT-Perception-layer-To-fix-the-issue-of-data-collecting-in_fig2_351986473.

37. https://www.matooma.com/fr/s-informer/actualites-iot-m2m/architecture-solution-iot.

38. Nguyen, K. T. (2016). *Protocoles de sécurité efficaces pour les réseaux de capteurs IP sans-fil et l'Internet des Objets* (Doctoral dissertation, Institut National des Télécommunications).

39. Cariou, E. (2003). *Contribution à un processus de réification d'abstractions de communications* (Doctoral dissertation, Rennes 1).