ECOLE
POLYTECHNIQUE
DE BRUXELLES

ULB

UNIVERSITÉ LIBRE DE BRUXELLES

Première ligne de titre du mémoire
Deuxième ligne de titre du mémoire
Ligne du sous-titre du mémoire

Mémoire présenté en vue de l'obtention du diplôme
d'Ingénieur Civil [...] à finalité [...]

[Prénom Nom]

Directeur
Professeur [Prénom Nom]

Co-Promoteur
Professeur [Prénom Nom]

Superviseur
[Prénom Nom]

Service
[Nom du service]

Année académique
20xx - 20xx

Contents

Chapter 1

Introduction

1.1 Contexte du mémoire

1.2 Parties prenantes

1.3 Projet précédent

1.4 Structure du mémoire

Chapter 2

État de l'art

2.1 Technologies de communication à distance

Lors des dernières années, le nombre de dispositifs Internet des Objets (IoT) connaît une croissance fulgurante qui se maintiendra au cours des années qui suivent (voir figure 2.1). Ces objets sont capables d'acquérir des données sur leur environnement et/ou de prendre des actions sur celui-ci. Ils communiquent avec d'autres machines pour transmettre les informations acquises et recevoir des commandes lorsque cela est nécessaire. Les dispositifs IoT ont de diverses applications telles que les thermostats intelligents, les voitures connectées, suivi et monitorage d'actifs, et bien d'autres. Selon les conditions d'utilisation et son objectif, ces dispositifs doivent être capables d'envoyer des données sur des courtes ou des longues distances. En outre, leur consommation énergétique doit généralement rester faible, notamment lorsqu'ils sont alimentés par une batterie.

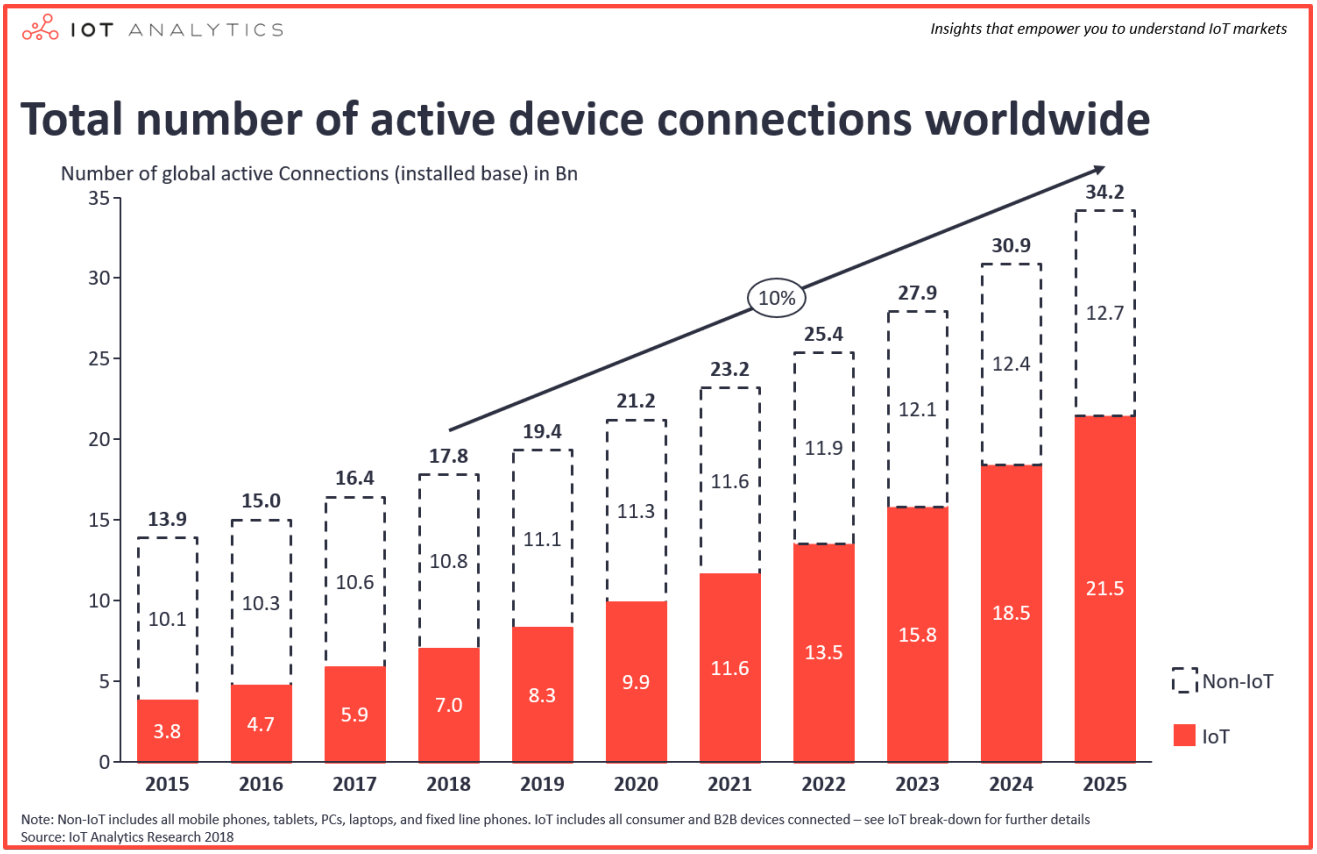

Figure 2.1: Nombre de dispositifs connectés à Internet [1]

Un réseau de communication unique et employable indépendamment du contexte du projet n'existe pas [2]. Toutefois, une multitude de réseaux avec des caractéristiques distinctes sont disponibles, tels que LoRaWAN et GSM. La figure 1 présente l'architecture simplifiée des réseaux utilisés par les dispositifs IoT pour échanger des informations avec des serveurs ou des utilisateurs qui sont connectés à Internet. Nous ne nous intéressons ici qu'aux technologies responsables de la transmission de données dans la phase 1 de la figure.

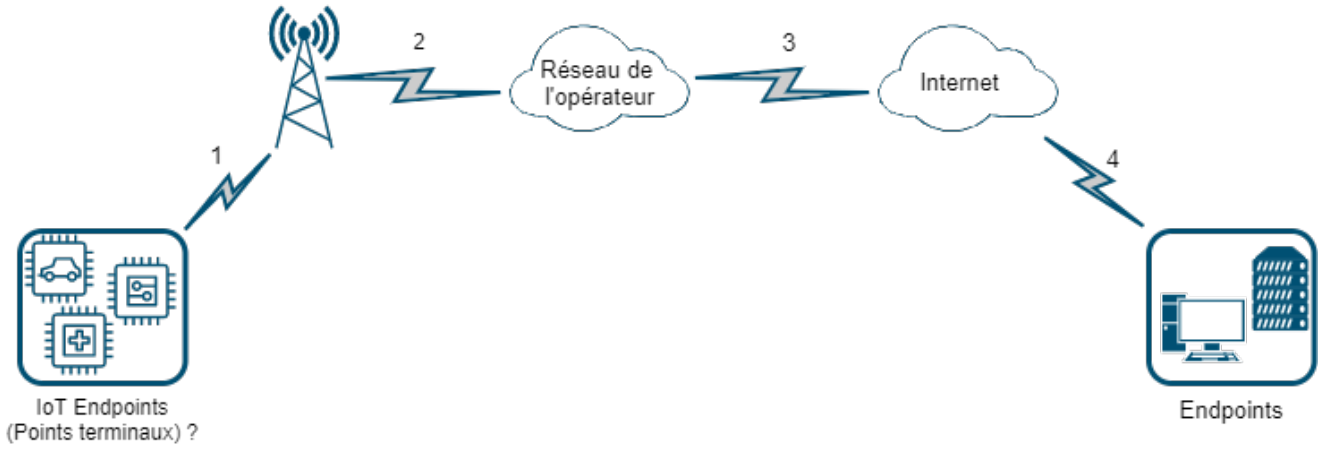

Figure 2.2: Architecture simplifiée d'un réseau avec dispositifs IoT (Basé sur [3, 4])

Actuellement, les principaux réseaux sans fil disponibles pour les dispositifs IoT sont :

- LoRaWAN
- Réseaux mobiles (2G/3G/4G/5G)
- Sigfox
- Satellite
- Wi-Fi
- Zigbee
- Bluetooth
- Réseaux mobiles IoT (NB-IoT et LTE-M)

Ces technologies utilisent toutes des ondes électromagnétiques pour transmettre les données, mais les fréquences sur lesquelles elles opèrent sont parfois très différentes. Certaines technologies utilisent une implémentation propriétaire, d'autres se basent sur des standards open source. [5] De façon similaire, certaines technologies exploitent la bande ISM[1], alors que d'autres exploitent des fréquences non réglementées. Ces dernières permettent de déployer son propre réseau privé, à condition d'acheter et configurer tout le matériel requis ce qui peut demander un investissement initial assez important. Ces différences accordent des propriétés distinctes aux réseaux en ce qui concerne le coût d'utilisation et déploiement, la bande passante, et la portée du signal. À noter

[1]Bande industrielle, scientifique et médicale

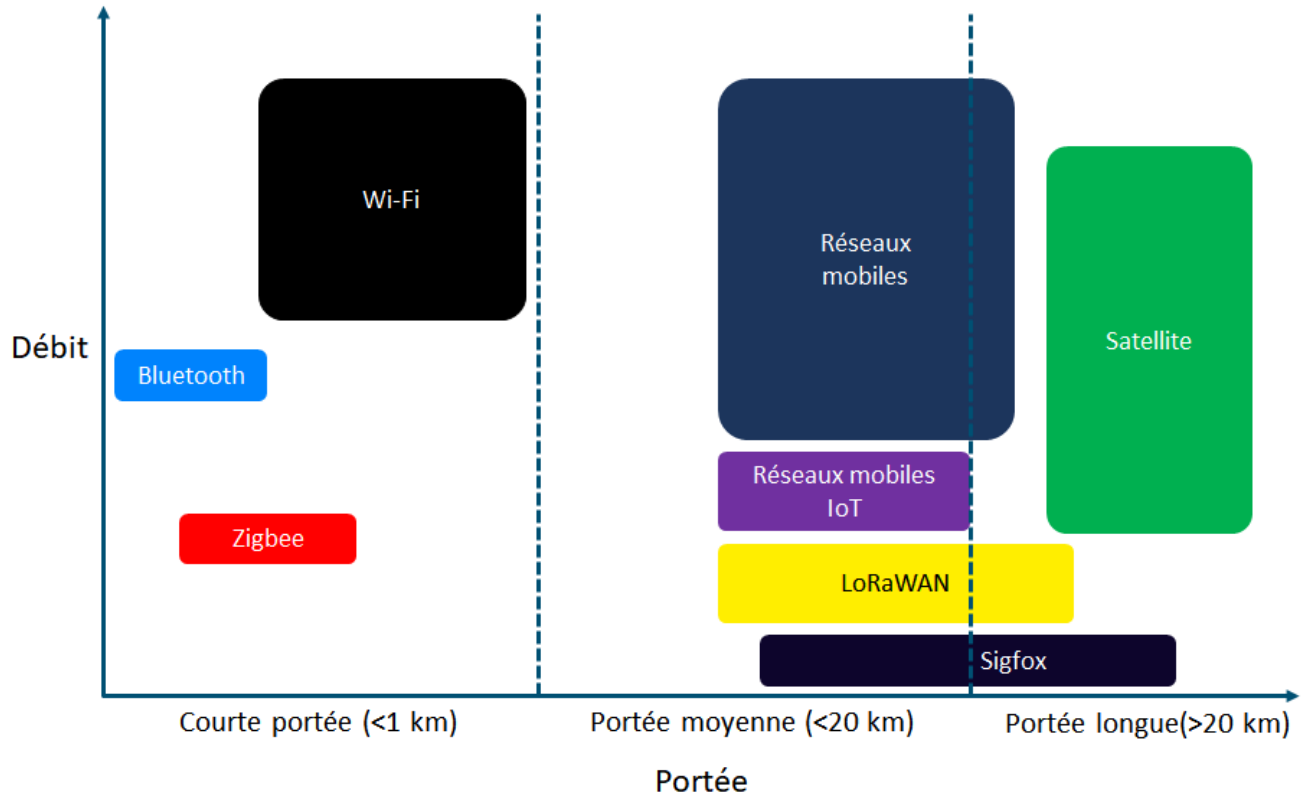

Figure 2.3: Classement des réseaux en fonction de leur portée et débit (Basé sur)

également, certains réseaux imposent une taille maximale sur chaque message, qu'il soit envoyé ou reçu. Le tableau comparatif de l'annexe offre une vue d'ensemble sur les différentes technologies et leurs propriétés. Il faut cependant remarquer que la portée et le débit sont juste donnés à titre comparatif et ils sont à prendre avec des pincettes. En effet, ces valeurs varient beaucoup en fonction de certains paramètres tels que la géographie du milieu (urbain ou rural) et des éventuelles interférences. Néanmoins, le débit et la portée sont très importants pour classifier ces réseaux et, avec l'aide de la figure 2.3, ces technologies peuvent être classées de la façon suivante [6] :

- IoT haut débit : Dans cette catégorie se trouvent principalement les technologies comme les réseaux mobiles, le Wi-Fi et le satellite. Les débits proposés sont généralement supérieurs au Mb/s. En contrepartie, l'utilisation de ces technologies engendre une consommation énergétique plus élevée.
- IoT bas débit : Ici se trouvent les réseaux mobiles IoT, LoRaWAN, Sigfox, et Zigbee. En fonction du réseau, le débit peut varier entre quelques bits par seconde jusqu'à un Mb/s. Ces technologies offrent une consommation énergétique faible.
- IoT critique : C'est-à-dire lorsqu'il s'avère nécessaire de transmettre des données sur un intervalle de temps donné [6]. C'est une propriété des réseaux 5G.

Un point également important est que ces technologies ont de différents niveaux de maturité. Au moins un réseau mobile est disponible dans chaque pays, mais les nouveaux réseaux comme Sigfox, LoRaWAN, et 5G sont en revanche toujours en cours de déploiement. De ce fait, ces dernières technologies ne sont disponibles que dans certaines régions possédant une infrastructure moderne. La couverture du réseau est un critère très important à considérer avant effectuer tout choix afin d'éviter de mauvaises surprises. Finalement, toujours dues à ce différent degré de maturité,

certaines technologies plus anciennes pourraient voir la fin de ses jours dans les années à venir. Par exemple, certains opérateurs vont décommissionner leur réseau 2G à partir de 2020 [7]. Les opérateurs restants devraient suivre la même tendance dans les années suivantes. De plus, le réseau 3G devrait suivre la même tendance comme affichée sur la figure 2.4.

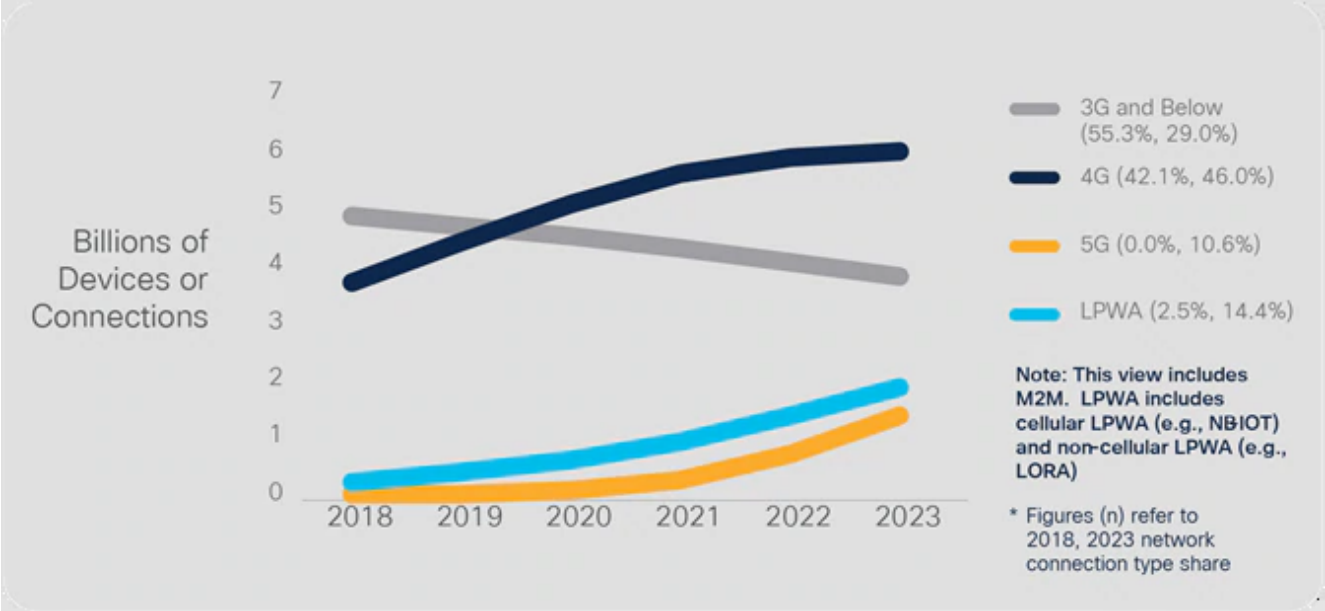

Figure 2.4: Évolution du nombre de dispositifs connectés aux différents réseaux [8]

2.2 Monitoring de l'état d'un serveur linux

La nécessité de surveiller toute l'infrastructure IT, et en particulier des serveurs tournant sous Linux, existe depuis un bon nombre d'années. Cependant, surveiller un système n'est pas trivial. Cela peut être accompli de différentes manières en fonction des évènements ou statistiques qui doivent être observées dans la machine hôte. Par exemple, l'utilisation de logs pour débugger une application ou voir l'historique de transactions d'une base de données est une technique de monitoring utilisée depuis de nombreuses années. D'autres techniques de monitoring existantes sont le profiling, le tracing et l'acquisition de métriques [9]. Dans cette section, seulement les technologies d'acquisition de métriques sont présentées puisqu'elles sont les plus utiles pour connaître l'état de la machine.

Créé à la fin des années 80, le protocole SNMP [10, 11] permet de monitorer et configurer de différents dispositifs connectés au réseau. Au long des années, des solutions de monitorage de métriques de plus haut niveau, ou tout-en-un, sont apparues sur le marché. Ces solutions utilisent différents protocoles, dont SNMP, pour surveiller les dispositifs sur le réseau et stockent toutes les informations recueillies sur une base de données. De plus, elles permettent de définir un ensemble de conditions à surveiller de plus près et si une de ces conditions est violée, une alerte est transmise vers l'équipe responsable de l'infrastructure. Deux exemples de solutions créées à la fin des années 90 et qui demeurent disponibles sont Nagios et Zabbix.

Actuellement, un grand nombre de solutions de monitoring sont disponibles sur le marché proposant différents modèles commerciaux. En effet, certaines solutions sont vendues comme un produit, mais d'autres préfèrent suivre un modèle open source. Dans le cas de ces dernières, les entreprises responsables de l'application proposent souvent leurs services en matière d'assistance

technique afin d'installer et maintenir la solution. Un autre modèle aussi appliqué est de proposer une version gratuite avec un nombre limité de fonctionnalités, et une version payante débloquant toutes les capacités de l'application. Avant d'entamer toute recherche d'une solution, il faut comprendre quelle formule s'adapte le mieux aux besoins de l'utilisateur. Comme il est affiché sur la figure 2.5, deux architectures existent pour les solutions de monitoring : *pull* et *push* [9, 12, 13, 12]. Chaque application est entièrement basée sur une de ces deux approches. Dans certains cas, une solution peut parfois supporter les deux architectures (voir exemple figure 2.6). Cependant, ces solutions détiennent toujours un modèle préféré, et l'autre possibilité sert juste à combler certaines lacunes du premier modèle puisque, en effet, chaque architecture possède ses avantages et faiblesses.

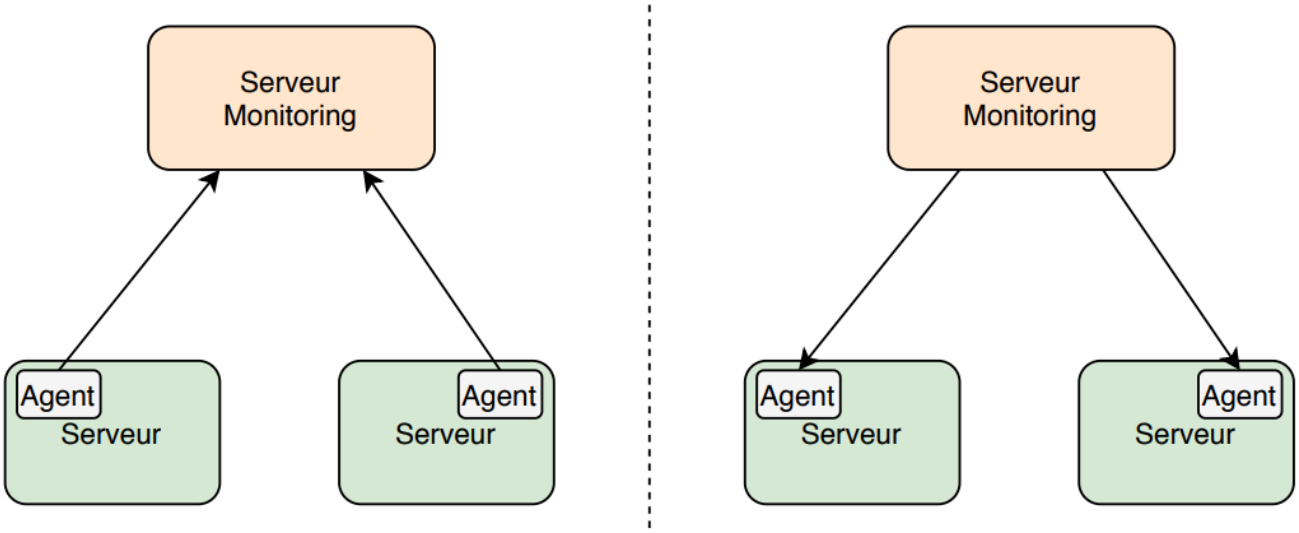

Figure 2.5: **Gauche :** Solution de monitoring avec architecture *push*. **Droite :** Solution de monitoring avec architecture *pull*.

Dans le modèle de type *push*, les agents dans la machine hôte envoient un ensemble de métriques ou des événements au serveur de monitoring. Cette technique permet notamment de définir des conditions d'alerte dans l'agent, et c'est celui-ci qui se charge de vérifier ces conditions. Lorsque le test des conditions passe, l'agent transmet un événement au serveur de monitoring. De plus, seul le système de push est capable de correctement acquérir des données à propos de jobs Batch de courte durée. [13, 14] En effet, comme affiché sur la figure 2.7, le modèle *pull* peut rater la fin du job ce qui se traduit par une perte d'informations.

Dans le modèle de type *pull*, les agents dans la machine hôte collectent et exposent les métriques, mais c'est au serveur de monitoring d'activement aller retrouver ces données. Par conséquent, les événements et alertes au niveau des agents ne sont pas pris en charge, dans la mesure où ces derniers ne savent pas quand le système de monitoring récupérera les informations. En revanche, cette méthode permet de mieux identifier si un problème s'est produit sur la machine hôte puisqu'elle ne répondra plus aux requêtes lorsque cela a lieu.[15] Ceci rend l'approche *pull* légèrement plus fiable que la technique *push*. Les deux méthodes possèdent donc leurs avantages et inconvénients qu'il faut tenir en compte lors du choix de la technologie. Cependant, dans la majorité des cas, les deux architectures offrent ces capacités très similaires [16].

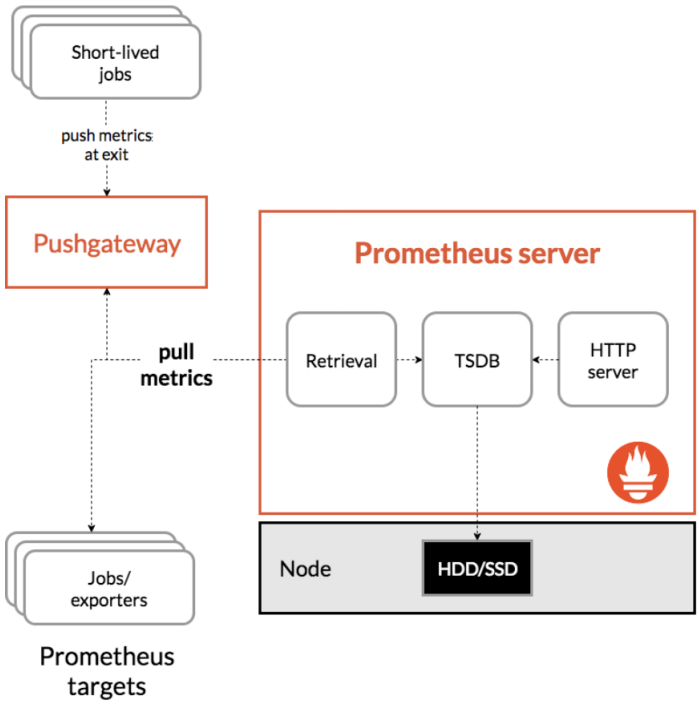

Figure 2.6: Architecture *pull* de Prometheus. Le modèle *push* est supporté grâce au Pushgateway.

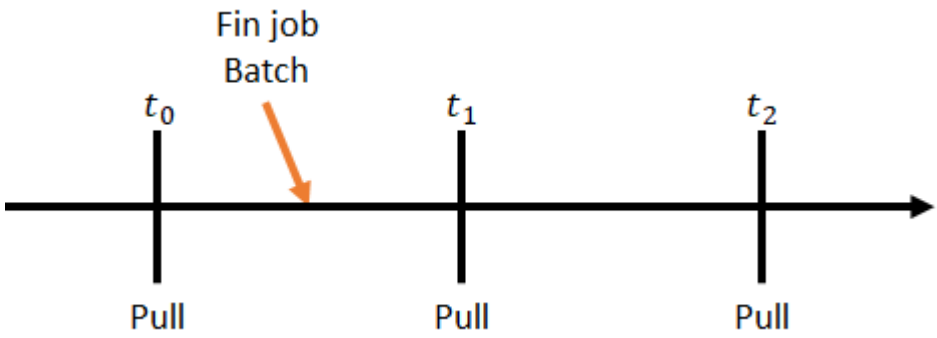

Figure 2.7: Faiblesse du modèle *pull*. Le système de monitoring a raté les informations de la fin du job

Indépendamment de l'architecture utilisée, toutes les informations recueillies ou reçues par le serveur de monitoring doivent être stockées sur une base de données. Les solutions plus anciennes comme Nagios et Zabbix, utilisent principalement des bases de données relationnelles [17, 18]. Ceci n'est plus le cas pour les solutions plus récentes telles que Prometheus et le TIG Stack[2] qui utilisent des bases de données orientées séries temporelles pour la persistance des données. Cette sorte de base de données, comme son nom indique, est mieux optimisée pour des valeurs horodatées [19]. En effet, ces bases de données possèdent les propriétés suivantes [20]:

- Elles supportent des écritures simultanées et avec de grands débits. Typiquement, dans ces bases de données se produisent énormément d'écritures, mais pas beaucoup d'accès.
- Ces bases de données doivent stocker d'énormes quantités de données temporelles. Elles utilisent des algorithmes de compression spécifiques pour stocker les données de manière efficace. [21]

[2]Le TIG stack est la combinaison de Telegraf, InfluxDB et Grafana pour créer un outil de monitoring

- Toutes les requêtes sont effectuées sur base d'un intervalle dans le temps
- Les données ont une durée de rétention. À partir d'une certaine date, la base de données élimine automatiquement les informations car elles sont jugées non utiles.

De ce fait, les bases de données orientées séries temporelles offrent un stockage plus efficace pour les métriques surveillées. Un dernier aspect très important d'une solution de monitoring est la visualisation de données. La majorité des solutions offrent soit un outil de visualisation propriétaire (Nagios ou Zabbix), soit elles exploitent des outils existants tels que Grafana (Prometheus et le TIG Stack).

2.3 Plateformes de visualisation de données

Chapter 3

Cahier des charges

3.1 Caractéristiques du prototype

3.2 Fonctionnalités à implémenter

3.3 Priorité des tâches et méthodologie

Chapter 4

Prototype

4.1 Le premier prototype (2018-2019)

4.1.1 Problèmes détectés

4.2 Nouveau Prototype

4.2.1 Solutions considérées

4.2.2 Thingstream

4.2.3 Résultat

4.2.4 Tests réalisés

Chapter 5

L'application

5.1 Client

5.1.1 Architecture

5.1.2 Base de données

5.1.3 Implémentation des fonctionnalités

Mappage du réseau local

Surveiller la présence des divers composants

Monitoring du serveur

Envoie des données

5.1.4 Sécurité

5.2 Serveur

5.2.1 Architecture

5.2.2 Base de données

5.2.3 Implémentation des fonctionnalités

5.2.4 Sécurité

Chapter 6

Tests et résultats

Chapter 7

Conclusion

Bibliography

[1] Knud Lasse Lueth. State of the iot 2018: Number of iot devices now at 7b – market accelerating. https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/, Aug 2018. [Accédé le 28-Avril-2020].

[2] Frederic Vannieuwenborg, Sofie Verbrugge, and Didier Colle. Choosing iot-connectivity? a guiding methodology based on functional characteristics and economic considerations. *Transactions on Emerging Telecommunications Technologies*, 29, 10 2017.

[3] Ramon Sanchez-Iborra and Maria-Dolores Cano. State of the art in lp-wan solutions for industrial iot services. *Sensors*, 16(5):708, 2016.

[4] Kais Mekki, Eddy Bajic, Frederic Chaxel, and Fernand Meyer. Overview of cellular lpwan technologies for iot deployment: Sigfox, lorawan, and nb-iot. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 197–202. IEEE, 2018.

[5] Brandon Foubert and Nathalie Mitton. Long-range wireless radio technologies: A survey. *Future Internet*, 12:13, 01 2020.

[6] David Feugey. Comment la 5g part à l'assaut de l'internet des objets. https://www.orange-business.com/fr/blogs/usages-dentreprise/mobilite/comment-la-5g-part-l-assaut-de-l-internet-des-objets, Dec 2016. [Accédé le 28-Avril-2020].

[7] Renouvellement du réseau mobile. https://www.swisscom.ch/fr/about/entreprise/portrait/reseau/remplacement-2g.html. [Accédé le 28-Avril-2020].

[8] Cisco. Cisco annual internet report (2018–2023). Technical report, Cisco, 03 2020.

[9] B. Brazil. *Prometheus - Up and Running: Infrastructure and Application Performance Monitoring*. O'Reilly Media, 2018.

[10] Jeffrey D. Case, Mark Fedor, Martin Lee Schoffstall, and Chuck Davin. Simple network management protocol (snmp). RFC 1098, RFC Editor, April 1989. http://www.rfc-editor.org/rfc/rfc1098.txt.

[11] Jeffrey D. Case, Mark Fedor, Martin Lee Schoffstall, and James R. Davin. Simple network management protocol (snmp). STD 15, RFC Editor, May 1990. http://www.rfc-editor.org/rfc/rfc1157.txt.

[12] Tech hub guides - monitoring at scale - monitoring architecture. https://developer.lightbend.com/guides/monitoring-at-scale/monitoring-architecture/architecture.html. [Accédé le 29-Avril-2020].

[13] Giedrius Statkevičius. Push vs. pull in monitoring systems. https://giedrius.blog/2019/05/11/push-vs-pull-in-monitoring-systems/, May 2019. [Accédé le 29-Avril-2020].

[14] Giedrius Statkevičius. Prometheus monitoring : The definitive guide in 2019. https://devconnected.com/the-definitive-guide-to-prometheus-in-2019/, May 2019. [Accédé le 29-Avril-2020].

[15] Faq: Why do you pull rather than push? https://prometheus.io/docs/introduction/faq/#why-do-you-pull-rather-than-push? [Accédé le 29-Avril-2020].

[16] schkn. Prometheus and the debate over 'push' versus 'pull' monitoring. https://thenewstack.io/exploring-prometheus-use-cases-brian-brazil/, 2019. [Accédé le 29-Avril-2020].

[17] Ethan Galsta. Ndoutils documentation version 2.x. https://assets.nagios.com/downloads/nagioscore/docs/ndoutils/NDOUtils.pdf, Mar 2017. [Accédé le 29-Avril-2020].

[18] Zabbix documentation 4.4: Database creation. https://www.zabbix.com/documentation/current/manual/appendix/install/db_scripts. [Accédé le 29-Avril-2020].

[19] Bastien L. Time series database : qu'est-ce que c'est, à quoi ça sert ? https://www.lebigdata.fr/time-series-database-definition, May 2018. [Accédé le 29-Avril-2020].

[20] Zhaofeng Zhou. Key concepts and features of time series databases. https://www.alibabacloud.com/blog/key-concepts-and-features-of-time-series-databases_594734, Apr 2019. [Accédé le 29-Avril-2020].

[21] Sheng Di, Hai Jin, Shengli Li, Jing Tie, and Ling Chen. Efficient time series data classification and compression in distributed monitoring. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 389–400. Springer, 2007.