Présent à tous les niveaux de l'IoT, le logiciel libre est appelé à s'imposer comme modèle d'efficacité et de pérennité, comme il l'a fait dans l'Internet et le Web et dans les grandes infrastructures motorisant l'informatique moderne.

L'IoT en tant que marché est en pleine structuration. Ce guide couvre toutes les couches de l'IoT, des plus applicatives aux plus proches du matériel, et dresse un panorama des nombreux projets concrets d'IoT existant en logiciel libre. On verra une fois de plus combien le paradigme du logiciel libre et des standards ouverts est incontournable pour constituer une infrastructure pérenne et fertile pour le développement de services.

Fort des compétences historiques de l'écosystème du logiciel libre en général, et en Île-de-France en particulier, dans les domaines structurants de l'informatique moderne (outils de développement, OS libres embarqués, cloud, middleware, big data, sécurité, sûreté de fonctionnement…), le GTLL est particulièrement bien positionné pour répondre aux demandes des industriels du secteur.

Le **Groupe thématique Logiciel Libre** forme l'un des principaux viviers de l'open source en France. Il rassemble startups, PME, grands groupes, universités et centres de recherche autour d'une même vision des défis technologiques de demain et d'un engagement profond pour la compétitivité de notre économie.

Systematic
Paris Region Digital Ecosystem

Conception : Nord Compo

LES LIVRETS BLEUS DU

LOGICIEL LIBRE

Open Source pour l'IoT

PIERRE FICHEUX

Préface de
STÉFANE FERMIGIER

Systematic
Paris Region Digital Ecosystem

Couv_IoT_PRINT_03.indd 2-3 03/11/16 13:01

LOGICIEL LIBRE

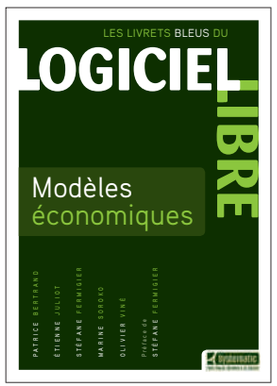
L'auteur :
Pierre Ficheux est directeur technique de Smile/Open Wide Ingénierie, et auteur du livre *Linux embarqué* (Eyrolles 2012)

Préface de **Stéfane Fermigier**, président du GTLL, fondateur d'Abilian

Vifs remerciements aux nombreux contributeurs du GTLL :
Emmanuel Baccelli (RIOT),
Florent Béranger (SMILE),
Étienne Boisseau (ENSAM),
Jérôme Molière (UCare),
Mickaël Rémond (ProcessOne),
Jean-Paul Smets (Nexedi),
Franck Spinelli (Amarisoft)

Remerciements à Muriel Shan Sei Fan pour le travail d'édition, à Didier Méresse et Mickaël De Clippeleir (Nord Compo) pour la conception et composition.

LES LIVRETS BLEUS DU GTLL : DES REPÈRES POUR COMPRENDRE



© GTLL, Systematic Paris-Region, 2017 – ISBN 978-2-9549444-9-4
Diffusion sous licence Creative Commons 4.0 CC BY-NC-ND. *Nous contacter si vous souhaitez rediffuser ce document dans un cadre commercial.*

À PROPOS DU GROUPE THÉMATIQUE LOGICIEL LIBRE DE SYSTEMATIC (GTLL)

Le Groupe thématique Logiciel Libre du pôle Systematic Paris-Region forme l'un des principaux viviers de l'Open Source en France.

Avec pour mission de **développer l'écosystème du libre en Île-de-France**, le GTLL regroupe plus d'une centaine d'acteurs de l'innovation ouverte (PME, ETI, grands groupes et académiques). Il vise à favoriser la coopération, l'innovation et l'emploi, autour de projets de R&D collaborative et grâce à des actions de soutien au développement des entreprises innovantes (promotion, marketing, stratégie, aide à la recherche de financements...), dans le cadre des principes et des valeurs de l'open source. Il est à ce jour le plus important cluster au monde à focaliser ses activités de R&D collaborative sur les logiciels libres et les défis spécifiques à l'open source, comme l'after PC (l'ère informatique du Cloud, des mobiles et des objets connectés), la qualité logicielle, et le déluge des données. *Après 9 ans d'existence, 57 projets de R&D collaborative consacrés au logiciel libre représentant un effort de R&D de près de 199,9 M€ ont déjà été financés grâce à l'aide du GTLL.*

L'action du Pôle est soutenue par :
ÎledeFrance
Partenaires stratégiques : Deloitte. Systemx GROUPE FG DESIGN

Couv_IoT_PRINT_03.indd 4-5 03/11/16 13:01

Open Source pour l'IoT

INT_IoT_PrintOK_03.indd 1 03/11/16 12:03

Sommaire

INT_IoT_PrintOK_03.indd 2 03/11/16 12:03

INT_IoT_PrintOK_03.indd 3 03/11/16 12:03

Préface

L'Internet des Objets (ou IoT – *Internet of Things* en anglais) est à la fois une évolution logique de l'Internet tout court, et une révolution annoncée de tous les pans de l'économie. On en sent déjà les prémisses dans plusieurs domaines, comme les transports avec les plateformes de covoiturage qui ont été rendues possibles par l'ubiquité des téléphones portables intelligents.

Ces enjeux économiques et stratégiques donnent une importance particulière aux nombreux défis technologiques qui se posent dans ce domaine. Le Groupe thématique Logiciel Libre (GTLL) de Systematic s'est emparé du sujet dès 2013, en inscrivant dans sa feuille de route ce que nous avons nommé l'« After-PC », dont l'IoT est un des aspects les plus disruptifs. Nous y avions identifié plusieurs thématiques technologiques clés en lien direct avec l'IoT.

- Tout d'abord les systèmes d'exploitation, middlewares et outils de développement pour objets communicants, dans le prolongement des logiciels dédiés à l'embarqué, mais aussi des outils issus du monde de l'Internet et du Web.
- Les plates-formes de développement de type *open hardware* telles qu'Arduino, Beagleboard ou équivalents, ainsi

INT_IoT_PrintOK_03.indd 4 03/11/16 12:03

que les outils de prototypage rapide d'objets physiques (CNC, imprimantes 3D…) qui permettent aux entrepreneurs, avec l'aide d'outils de développement logiciel open source dont le coût d'utilisation est également minime, de réaliser des démonstrateurs de leurs idées avec des moyens très modestes.

- Les vastes systèmes distribués, qui mettent en œuvre des terminaux (plus ou moins intelligents et connectés de manière plus ou moins intermittente), des routeurs (eux aussi plus ou moins intelligents) et des plates-formes de stockage et de traitement ; ces plates-formes IoT doivent répondre à des défis de volumes de données et de résilience inouïs dans le cadre des systèmes d'information d'entreprise traditionnels. Plus précisément, toutes les technologies susceptibles de gérer les flux de données engendrés ou capturés par un nombre sans cesse croissant de terminaux – de quelques milliers à plusieurs milliards : on parle bien de big data et de cloud computing.
- Les nouveaux protocoles qui apparaissent, ou évoluent comme HTTP pour s'adapter aux besoins spécifiques de l'IoT, et leur implémentation sous forme de briques open source. Plus généralement, l'interopérabilité des différentes

INT_IoT_PrintOK_03.indd 5 03/11/16 12:03

solutions – matérielles et logicielles – mises sur le marché par des acteurs aux visées parfois monopolistes nous semble un point d'attention stratégique majeur, sur lequel le monde du logiciel libre a déjà su se battre pour faire triompher la notion de standard ouvert.

- L'enjeu de la sécurité nous semble enfin majeur, et encore très mal pris en compte à l'heure actuelle. Que l'on parle d'objets communicants liés à la vie intime (montres, brosse à dents connectée, horloges intelligentes, balances connectées, etc.) ou d'IoT industriel – avec le détournement par des mafias d'objets communicants non sécurisés, il est difficile de faire l'impasse sur l'investissement nécessaire à la sécurisation des protocoles et des logiciels.

L'IoT en tant que marché est en pleine structuration. On y verra une fois de plus combien le paradigme du logiciel libre et des standards ouverts est incontournable dès lors qu'on s'approche de ce qui est appelé à devenir infrastructurel. Le logiciel libre s'est imposé dans le monde de la recherche informatique, de l'Internet devenu standard de fait, du Web comme infrastructure de communication. Il s'imposera aussi dans le domaine de l'IoT.

INT_IoT_PrintOK_03.indd 6 03/11/16 12:03

Fort des compétences historiques de l'écosystème du logiciel libre en général, et en Île-de-France en particulier, dans les domaines précités (outils de développement, OS libres embarqués, cloud, middleware, big data, sécurité, sûreté de fonctionnement…), les acteurs du GTLL sont particulièrement bien positionnés pour répondre aux demandes des industriels du secteurs, et notamment dans les différents marchés que touchent l'ensemble des membres de Systematic : Systèmes d'Information pour les entreprises et les administrations, Santé, Usine du Futur, Gestion intelligente de l'Energie, Télécoms, Transports et mobilité, Ville numérique…

Le livret bleu de Pierre Ficheux, auquel ont contribué de nombreux membres du GTLL, est donc l'occasion de faire la synthèse des principaux défis, mais aussi des solutions actuelles et des tendances à venir de ce marché, et d'aider tous les décideurs technologiques des industries en aval de l'écosystème des créateurs d'outils numériques ouverts à faire des choix avisés dans ce domaine.

Stéfane Fermigier

président du Groupe thématique Logiciel Libre

INT_IoT_PrintOK_03.indd 7 03/11/16 12:03

Avant-propos

Tout comme l'Internet a bouleversé notre quotidien au siècle dernier, la révolution de l'intégration d'objets physiques dans le réseau des réseaux est en marche. Et il n'est plus de nos jours un média qui n'évoque l'Internet des objets (IoT). On peut raisonnablement se demander s'il ne s'agit pas, plus que d'une révolution, tout simplement de **la réapparition sur le devant de la scène d'un ensemble de technologies autrefois ignorées du grand public.** En 1995, le chercheur français Christian Huitéma évoque dans son ouvrage *Et Dieu créa l'Internet* [1] la possibilité de connecter à Internet des objets physiques.

> *«Il y a déjà des microprocesseurs, en fait de tout petits ordinateurs dans bien d'autres endroits […]. D'ici quelques années, le développement et les progrès de l'électronique aidant, ces microprocesseurs deviendront sans doute de vrais ordinateurs élaborés et il sera tout à fait raisonnable de les connecter à Internet»*

Pour le plus grand nombre, cette opinion très avant-gardiste était passée totalement inaperçue car en 1995, connecter entre eux ne serait-ce que de vrais ordinateurs n'était pas sans difficulté. Cependant, quelques années plus tard (en 2002), un

INT_IoT_PrintOK_03.indd 8 03/11/16 12:03

certain Andy Rubin de la société Danger Inc. participe à la création d'un téléphone mobile nommé « Hiptop » capable de se connecter à Internet et d'utiliser un jeune moteur de recherche nommé Google. Les technologies mobiles d'accès à Internet étant encore limités à l'époque, le produit n'a hélas aucun avenir commercial, mais il est remarqué par les deux fondateurs de Google.

Remercié, Andy crée la société Android Inc. en 2004 et tout le monde connaît la suite ; à ce jour la grande majorité des terriens utilisent le principal (et seul ?) objet connecté grand public animé par le fameux système Android, lui-même basé sur le noyau Linux et en grande partie « open source » si l'on considère le projet AOSP – Android Open Source Project – fournissant les sources du système (mais pas les pilotes propriétaires !). Au niveau industriel on voit clairement les solutions propriétaires M2M (*Machine-to-machine*), qui sont une branche dédiée de l'IoT, migrer vers des solutions plus ouvertes avec l'adoption de standards de type Ethernet comme AFDX (Avionic Full DupleX) EtherNet/IP, EtherCAT ou openPOWERLINK, dont les sources sont disponibles [2].

Dire que le succès du smartphone est lié à l'utilisation du logiciel libre pourrait paraître un raccourci militant facile mais nous (GTLL) justifions notre opinion. Tout comme nous considérons que le succès de l'Internet est largement lié à l'utilisation de standards (les fameux RFC ou « Request For Comments » qui doivent, pour devenir des normes Internet « *présenter plusieurs implantations interopérables* » selon l'IETF),

INT_IoT_PrintOK_03.indd 9 03/11/16 12:03

il nous paraît évident que **l'enfermement de l'IoT dans une logique propriétaire serait un frein** qui pourrait remettre en cause les prédictions mirobolantes des analystes (50 milliards d'objets connectés en 2020). Lors d'une conférence récente autour de l'IoT dans l'industrie, le représentant d'un acteur majeur de l'industrie déclara « qu'un standard ne disposant pas d'implémentation libre ne pouvait être un vrai standard » [3]. Comme il ne viendrait pas à l'idée aujourd'hui de ne pas utiliser – sur l'Internet actuel – TCP/UDP/IP et ce quelle que soit la plate-forme, il n'est pas envisageable de s'affranchir pour l'IoT de standards comme CoAP, MQTT, 6LoWPAN, etc.

Ce livret ne saurait être un catalogue exhaustif des solutions disponibles pour l'IoT (il existe déjà plusieurs publications de ce type). Nous décrirons plutôt comment le développement d'un réel marché de l'IoT ne peut que s'accompagner de l'utilisation de solutions libres tant au niveau du traitement des données, des protocoles de communications, des outils de développement, des systèmes embarqués voire du matériel. Pour cela nous nous baserons sur de nombreux exemples de composants d'ores et déjà disponibles dans chacune des catégories. Bien entendu il n'est pas question de faire preuve d'intégrisme du libre et il est des domaines ou les solutions libres ont encore à ce jour inexistantes ou pas assez éprouvées.

Dans un premier chapitre nous verrons pourquoi l'adoption de solutions libres est une nécessité pour l'IoT. Le deuxième chapitre détaillera les technologies utilisées pour l'IoT et comment des solutions libres peuvent être choisies tant au niveau de

INT_IoT_PrintOK_03.indd 10 03/11/16 12:03

l'objet connecté (matériel, logiciel, système d'exploitation), que des protocoles réseau, des solutions de traitement de données et services à l'utilisateur final. Enfin le troisième et dernier chapitre fera un point sur les modèles d'affaires envisageables, toujours dans l'esprit du logiciel et matériel libre.

INT_IoT_PrintOK_03.indd 11 03/11/16 12:03

Chapitre 1

IoT et logiciel libre : standards versus silos

Si l'on regarde le panorama du marché naissant de l'Internet des objets à ce jour (mi 2016), on s'aperçoit que les prévisions des analystes ont provoqué un engouement sans précédent. L'IoT est présenté comme un Eldorado sans faille qui devrait apporter un nouvel espoir à notre économie qui selon certains est en passe d'être ruinée par l'*uberisation* de la société. Le cas fut similaire lors de l'avènement de l'Internet qui passa du monde désintéressé de la recherche à l'économie numérique. Les maîtres du monde ne sont plus aujourd'hui les maîtres de la fonte et de l'acier mais ceux qui savent obtenir et surtout exploiter les données que l'on peut considérer comme la nouvelle monnaie planétaire (les fameux « GAFA » pour Google, Apple, Facebook, Amazon). Ce marché colossal lié au commerce électronique touche toutes les couches de la société et tous les services de la vie courante (transport, communication, énergie, santé, enseignement, etc.). Ce même marché est basé sur des standards ouverts (les *Requests For Comments* ou RFC) et utilise en grande majorité des logiciels libres (machines sous GNU/Linux, logiciel serveur web Apache, etc.). Le système Android est installé sur plus de 85 % des « smartphones » qui sont aujourd'hui le terminal de prédilection pour l'accès à

INT_IoT_PrintOK_03.indd 12 03/11/16 12:03

Internet. Microsoft lui-même pour lequel le logiciel libre n'était qu'un virus communautaire il y a quelques années s'est résolu à rejoindre – en partie – les rangs des défenseurs du logiciel libre entre autres pour les solutions de cloud.

Paradoxalement, **le démarrage du marché de l'IoT dans sa version grand public (donc de masse) s'accompagne du développement d'une multitude de solutions matérielles et logicielles captives,** donc très éloignées de ce modèle libre qui a fait le succès – et pour certains la fortune – des acteurs de la version précédente de l'Internet. La situation est d'autant plus problématique que l'IoT utilise d'autres éléments du monde réel comme des micro-calculateurs embarqués et souvent des réseaux – et protocoles – de communication dédiés (principalement pour des raisons d'optimisation de bande passante et de consommation énergétique). Le GTLL a organisé en 2015 et 2016 une journée de conférences dédiées à l'IoT lors de l'Open Source Innovation Spring (OSIS [4]). De plus, la fondation Eclipse, l'IEEE et Agile IoT ont réalisé des sondages en 2015 et 2016 afin de mieux connaître les méthodes et outils utilisés par les développeurs IoT (*IoT Developer Survey* 2015 et 2016) [5][6].

Lors de la conférence de Philippe Krief de la fondation Eclipse, intitulée *Why IoT needs Open Source Communities* ce dernier indique que pour l'instant, **IoT rime avec *Internet of Silos,*** ce qui est hélas une image fidèle de la réalité. Il en est de même pour la prolifération des pseudo-standards qui sont souvent des

INT_IoT_PrintOK_03.indd 13 03/11/16 12:03

produits uniquement issus de solutions propriétaires (le réseau/protocole SIGFOX étant le cas le plus célèbre dans l'IoT) [7].

Les principales cibles des solutions propriétaires sont les sociétés naissantes ou adoptant ce marché à venir comme une diversification. Dans ce cas il n'est pas toujours facile de résister aux sirènes de la solution unique (du matériel au traitement des données en passant par l'outil de développement). Les technologies utilisées nécessitent fatalement des compétences et donc un investissement intellectuel et financier.

Or, à l'instar des créateurs de sites web, les concepteurs (*designers*) d'objets connectés ne sont pas forcément des techniciens émérites. En outre le spectre des technologies impliquées dans l'IoT est très large du fait de la présence de matériel – qui doit obéir à des contraintes de fiabilité car souvent destiné à un fonctionnement autonome dans un environnement hostile ou non technique.

Le cas plus industriel des logiciels embarqués est similaire. Le grand public dispose, à une large majorité, d'équipements d'accès à Internet (*set-top* box, Internet Access Devices). Sous la pression des opérateurs de services, ces équipements migrèrent il y a quelques années d'un environnement propriétaire (cœur ST/SH4 et système d'exploitation OS20/21 également de ST) vers l'environnement Linux sur de multiples plates-formes (ARM, x86) provenant de divers fondeurs.

L'adaptation ne fut pas simple car l'éditeur propriétaire (qui parfois fournit le matériel) livre une kyrielle d'outils parfaite-

INT_IoT_PrintOK_03.indd 14 03/11/16 12:03

ment adaptés et documentés. Du côté du libre, les choix sont plus complexes (puisqu'il y a le choix !) et la documentation pas toujours au rendez-vous même si les grands projets libres actuels ont fait de gros progrès. Nous pouvons citer en exemple l'outil de construction Yocto utilisé pour de nombreux projets Linux dont certains liés à l'IoT comme GENIVI ou AGL (automobile). **Le projet Yocto – soutenu par la fondation Linux et de grands industriels – bénéficie d'un écosystème structuré et de grande qualité,** y compris sur des sessions de formation. Finalement, **la migration – plus ou moins forcée – des box vers l'environnement Linux fut largement bénéfique, en particulier pour la standardisation de composants industriels** (middleware) apportant le niveau d'abstraction nécessaire au développement des applications multimédia.

Une partie importante de l'Internet des Objets consiste néanmoins en machines qui ne pourront pas utiliser Linux (ou ses dérivés) faute de ressources suffisantes en CPU, mémoire, et en énergie. Pour ces machines, la guerre du logiciel libre commence, mais n'est pas encore gagnée. En effet, ce type de matériel est produit par des constructeurs qui ont une culture software traditionnellement orientée silo et *bare metal*, très éloignée de la culture Internet. Notamment, ces constructeurs sont restés imperméables au mouvement open source, jusqu'à très récemment.

Cependant, au fur et à mesure que ces machines sont censées embarquer du logiciel de plus en plus sophistiqué (par exemple : une pile de protocoles IPv6 et 6LoWPAN), et que

INT_IoT_PrintOK_03.indd 15 03/11/16 12:03

des entreprises tierces se spécialisent dans des applications utilisant indistinctement divers modèles de machines de ce type, l'open source devient l'alternative de choix, à l'instar du phénomène Linux ailleurs sur Internet. De ce fait, des plates-formes différentes de Linux du point de vue technique, mais analogues à Linux du point de vue de l'écosystème (logiciel communautaire libre), voient leur popularité grandir, comme c'est le cas de RIOT (voir p24).

Ce type d'approche basée sur le logiciel libre dans l'IoT a pour but d'une part d'accélérer l'innovation en diminuant le coût d'entrée pour le développement de logiciel embarqué, et d'autre part de permettre plus de transparence et plus de sécurité, sur le long terme, pour le logiciel à l'œuvre de bout-en-bout, de l'Internet jusqu'aux objets connectés inclus et ce, même si ces derniers ne peuvent pas utiliser Linux.

INT_IoT_PrintOK_03.indd 16 03/11/16 12:03

Chapitre 2

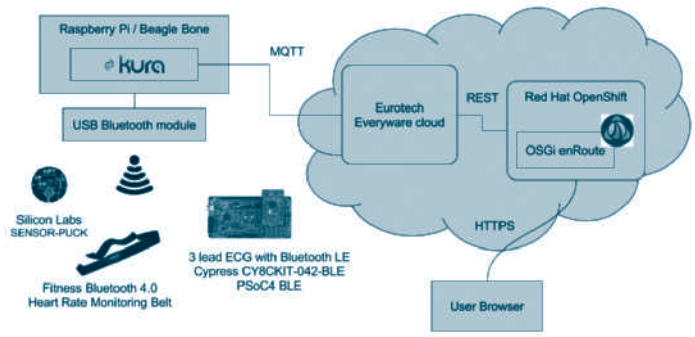
Technologies : innovation ou intégration

Dans ce chapitre, nous décrivons les solutions libres disponibles pour les différentes couches utilisées dans une infrastructure IoT. La diversité des composants implique plusieurs niveaux de compétences et de technologies.

1. Système embarqué (i. e. l'objet lui-même équipé de son système d'exploitation et/ou de son application)
2. Accès au réseau et protocoles (et passerelles diverses)
3. Récupération et traitement des données/Cloud
4. Service à l'utilisateur final

Notons qu'une fois la phase de prototypage réalisée, le déploiement de l'ensemble nécessite le plus souvent l'utilisation d'un framework spécialisé. La figure suivante représente un cas d'utilisation du framework Eclipse/Kura utilisant des capteurs et une carte Raspberry Pi comme passerelle et un accès via le protocole MQTT à un cloud *Eurotech Everywhere Cloud*.

INT_IoT_PrintOK_03.indd 17 03/11/16 12:03



Cas d'utilisation Eclipse/Kura

Couche 1 – Système embarqué

On a l'habitude d'utiliser la terminologie « système embarqué » pour l'ensemble du système constitué du matériel et du logiciel. Dans le cas du logiciel, le terme système désigne parfois un système d'exploitation (OS) mais l'objet n'en dispose pas forcément. Nous verrons cependant que l'écosystème IoT fournit des OS très bien adaptés à tous types de plates-formes.

La partie système embarqué est une couche à la fois maîtrisée et complexe. Maîtrisée car le logiciel libre (voire le matériel) est fortement présent dans ce domaine, au point de devancer voire évincer des solutions propriétaires. L'exemple le plus flagrant – et tentant, reconnaissons-le – est Microsoft Windows Embedded. Selon l'article cité en [8], Windows CE 6 représentait en 2006 99 % du marché des périphériques mobiles, mais à partir de 2011 l'industrie a totalement boudé les versions

INT_IoT_PrintOK_03.indd 18 03/11/16 12:03

suivantes basées sur Windows 7 et Windows 8, sans parler de l'OS mort-né Windows Phone. Microsoft tente de revenir sur le devant de la scène avec Windows 10 et sa variante Windows IoT [9] mais il y a de grandes chances que ces versions subissent le même sort au profit de solution libres – ou partiellement libres – comme Android, basé sur un noyau Linux, et ses dérivés, ou encore Tizen, un système d'exploitation très proche de Linux et très utilisé pour les produits d'électronique grand public chez SAMSUNG. Ce système fut présenté parmi les solutions IoT dans le cadre des journées de l'OSIS 2016 organisées par le GTLL [4].

Dans le cas de l'IoT il faut bien entendu tenir compte de l'étendue des possibilités du matériel, allant du simple capteur mono-tâche (8 bits, comme l'Atmega328 équipant l'Arduino Uno) à un module ARM ou x86 (32 ou 64 bits) capable de faire fonctionner Android ou Linux.

De même, le marketing autour du simple mot «IoT» produit des effets pervers puisque la plupart des cartes sont aujourd'hui présentées comme étant une base matérielle pour des solutions IoT, ce qui n'est pas toujours le cas. Un exemple est la RioTboard commercialisée par Farnell. C'est une excellente carte peu coûteuse basée sur i.MX6 et supposée «révolutionner l'IoT» [10]… mais on peut se demander comment, car la démonstration sur le thème IoT est étonnamment basée sur une distribution Debian complète utilisant MySQL pour traiter les mesures issues d'un capteur de température…

INT_IoT_PrintOK_03.indd 19 03/11/16 12:03

Choix d'un OS embarqué

L'étude citée en [6] propose des statistiques concernant le choix d'un OS pour un objet connecté. L'OS Linux arrive largement en tête avec 73 % des suffrages suivi par les architectures *bare metal* (sans OS) avec 23 % et FreeRTOS avec presque 13 %. Les autres choix dont certains ont une approche beaucoup plus ciblée sur l'IoT se situent entre 5 % et 7 % (Contiki, RIOT, TinyOS, Mbed, etc.). Les résultats étaient à peu près les mêmes un an plus tôt (étude de 2015 [5]), à la notable différence près que certains projets libres – et assez récents – tels que RIOT ont doublé leur part de marché.

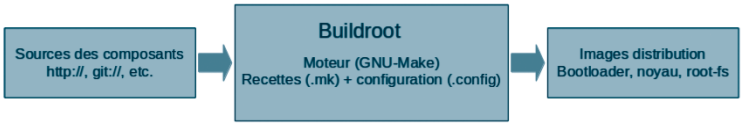
La prédominance de Linux (ou des systèmes dérivés comme **Tizen**) peut s'expliquer par le marché actuel qui est encore très orienté vers des applications professionnelles et des architectures puissantes (ARM 32 bits principalement). Les capteurs IoT utilisant du logiciel « enfoui » sont encore peu fréquents. Même dans le marché grand public, la majorité reste encore à ce jour du côté des smartphones et autres set-top box.

L'utilisation de Linux ne doit pas s'entendre comme « choisir une distribution Linux » car les contraintes liées à l'IoT sont très différentes (proches de celles des systèmes embarqués). On ne choisira pas une distribution, on devra la *construire* en utilisant des outils *build systems* comme Yocto (OpenEmbedded) ou Buildroot, largement utilisés dans le domaine. Ces outils permettent d'intégrer et de maintenir un ensemble de sources soit provenant de communautés externes (comme le noyau Linux), soit étant des composants internes à l'entreprise

INT_IoT_PrintOK_03.indd 20 03/11/16 12:03

(par exemple les applications utilisées par l'objet). Le résultat d'un tel outil est une distribution (ou un firmware Linux dans le cas de Buildroot) à installer sur la cible avec une empreinte mémoire bien moindre que celle d'une distribution classique (quelques dizaines de Mo contre quelques Go minimum pour une distribution).

La figure suivante décrit l'architecture simplifiée de Buildroot.



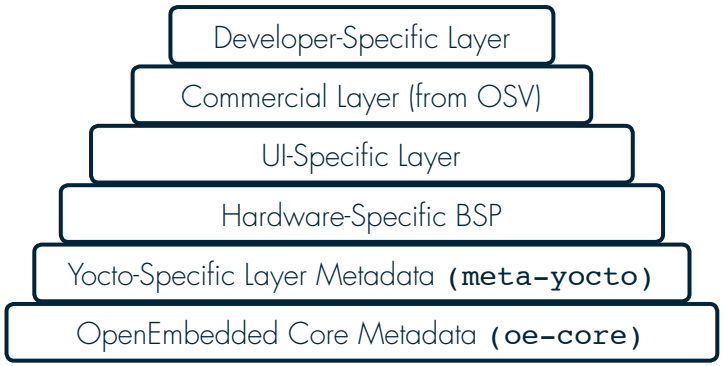
Architecture de Buildroot

À titre d'exemple la célèbre caméra GoPro utilise **Buildroot** et Linux pour son logiciel interne [11]. C'est également le cas des drones de la société Parrot qui utilisent également un système de build interne (mais open source) dérivé de celui d'Android.

Bien qu'utilisant le même principe (sources, moteur et binaires), **Yocto/OE** utilise une approche à la fois plus modulaire et plus complexe mais permettant de construire une véritable distribution intégrant un système de gestion de paquets (RPM, IPK, DEB). La puissance de Yocto/OE vient aussi de l'outil BitBake utilisé comme « séquenceur » de tâches alors que Buildroot utilise GNU-Make. BitBake est beaucoup plus avancé, utilisant des classes et un principe d'héritage très puissant permettant d'étendre l'outil à l'infini en empilant des couches (layers) au niveau matériel, système, « middleware » ou applicatif. L'em-

INT_IoT_PrintOK_03.indd 21 03/11/16 12:03

preinte mémoire sur la cible peut cependant être comparable à celle de Buildroot si l'on n'embarque par le système de gestion de paquets sur la cible.

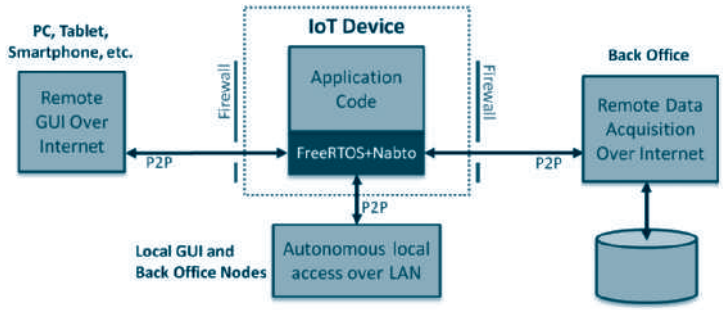


Les couches (layers) de Yocto/OE

L'utilisation de Linux n'est pas toujours ni judicieuse ni possible lorsque nous avons affaire à des micro-contrôleurs (MCU parfois 16 voire 8 bits) pour des objets plus simples. De fait, il est très probable que la foule d'objets prévue par les analystes concerne plus ce type de cible qu'un système Linux (et a fortiori Android). Une bonne partie de ces objets utilisent une architecture *bare metal* c'est-à-dire sans système d'exploitation (ce qui d'après l'étude citée représente 23 % des systèmes). La nécessité d'utiliser un OS dépend largement du côté fonctionnel, donc des capacités de communication nécessaires ainsi que des contraintes de consommation. Alors que Linux n'est pas initialement prévu pour être un OS natif de l'IoT (et qui plus est sur des MCU), plusieurs OS libres sont utilisables dans

INT_IoT_PrintOK_03.indd 22 03/11/16 12:03

ce domaine. **FreeRTOS** est un exemple typique du RTOS qui a su évoluer du marché de l'embarqué vers l'IoT en intégrant l'OS dans une solution complète (Nabto) [12].



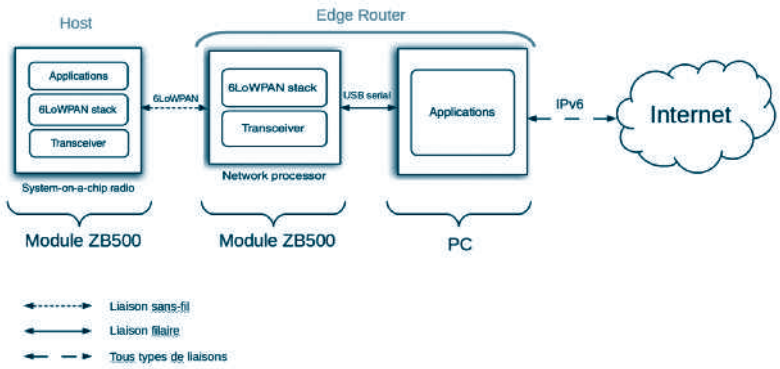
Solution Nabto à base de FreeRTOS

Outre les OS adaptés, plusieurs systèmes d'exploitation libres sont dédiés à l'IoT, les plus connus étant Contiki et RIOT.

Contiki fut créé pour les capteurs (le mot « IoT » n'existait pas encore) au début des années 2000. Le système est écrit en C mais la programmation est parfois assez éloignée du standard POSIX.

La figure suivante décrit une configuration basée sur deux modules ZB500 fonctionnant sous Contiki. Le premier module est un capteur de température communiquant grâce au protocole 6LoWPAN (dérivé d'IPv6). Il communique avec un deuxième module connecté par USB à un PC Linux utilisé comme passerelle (ou « routeur de bordure »).

INT_IoT_PrintOK_03.indd 23 03/11/16 12:03



Démonstration avec capteur sous Contiki

L'utilisation du **standard POSIX** est un point important car cela permet de pérenniser le code source sur de nombreuses années et ce indépendamment de la cible utilisée. Le standard POSIX est largement utilisé dans les industries sensibles (aéronautique en particulier) pour lesquelles les durées de vie des programmes (et donc de la maintenance) sont très longues. Certains OS libres utilisables pour l'IoT fournissent nativement une compatibilité POSIX comme RIOT ou Lepton.

RIOT est un système d'exploitation pour les objets connectés qui ne peuvent pas utiliser Linux faute de ressource suffisantes en CPU, mémoire et énergie. Lancé en 2013 par Inria et deux universités allemandes dans le cadre d'un projet de recherche, RIOT rassemble aujourd'hui une grande communauté de développeurs venant du monde entier. Selon ses concepteurs, le but de RIOT est de devenir « le Linux de l'IoT ». RIOT dispose d'une API optimisée pour l'IoT, partiellement compatible avec POSIX, et se base sur une architecture logicielle avec micro-

INT_IoT_PrintOK_03.indd 24 03/11/16 12:03

noyau temps réel. L'accent est mis sur la portabilité de code proche du matériel, offrant le support d'un grand nombre de plates-formes incluant ARM Cortex M, TI MSP430, AVR, x86 notamment. RIOT étant conçu pour l'IoT, il inclut le support des principaux standards de communication du domaine, par exemple, IPv6, 6LoWPAN et CoAP. D'autre part, son architecture favorisant la portabilité de 95 % du code, combinée au support de langage pur (ANSI C) pour les applications, offre une facilité accrue dans le développement, l'exécution et le test de logiciel IoT, permettant l'utilisation d'outils connus tels gdb, valgrind, gcc entre autres (voir http://riot-os.org/).

Lepton est (partiellement) un OS contenant une couche POSIX que l'on peut adapter à des noyaux existants (libres ou pas) ce qui permet de s'affranchir des API propriétaires. Il fut au départ développé pour garantir la compatibilité du code entre plusieurs gammes d'appareils de mesures fonctionnant sous eCOS (libre) ou embOS (propriétaire) [13].

En février 2016 la fondation Linux a présenté le projet **Zephyr** qui est un OS similaire à Contiki ou RIOT au niveau des plates-formes concernées. Cet OS est dérivé du noyau de « Rocket OS » développé par Wind River.

Matériel libre, un passage tardif des bits aux atomes

Le développement du matériel libre (*open hardware*) est assez récent si on le compare avec le logiciel. En effet, pour pouvoir partager, modifier, réutiliser, etc. (i.e., rendre libre) un objet, il

INT_IoT_PrintOK_03.indd 25 03/11/16 12:03

est nécessaire que celui-ci soit, d'une certaine manière, digitalisé. Ainsi, dans l'open hardware, ce n'est pas l'objet lui-même qui est partagé (un objet physique ne peut évidemment pas être en plusieurs lieux à la fois), mais ses «sources». Ces dernières peuvent être de plusieurs types : plans de conception (fichiers *.pdf, *.dwg, *.cat), fichiers pour découpeuse laser (dessin 2D vectoriel standards (*.pdf, *.eps, *.svg, etc.) ou spécialisés (*.ai, *.dwg, *dxf), ou machine de fabrication additive (*.stl, *.obj, *.3mf), etc. Le but de ces «sources» est de fournir une description univoque de l'objet, qui n'a alors plus «qu'à être fabriqué» (on peut comparer cela à la différence entre une recette de cuisine et le gâteau réalisé) [13b].

Or la démocratisation du numérique dans la production d'objets tangibles (dans les année 1960 pour le début du CAM – *Computer Aided Manufacturing*), puis la conception de produits (fin des années 1970 pour le CAD[esign]) est plus tardive. Pour autant, cela n'explique que partiellement le retard.

La seconde explication est à trouver dans la démocratisation de la production et de la conception elles-mêmes. De la même manière que le logiciel libre est venu d'utilisateurs voulant retrouver la liberté sur des logiciels, l'open-design s'est développé lorsque les utilisateurs ont souhaité retrouver leur liberté sur les objets qu'ils utilisent. Il a fallu pour cela la démocratisation des moyens de production via l'apparition de machines de fabrication numérique à faible coût (imprimante 3D, découpe laser, etc.) et de nouvelles structures pour fabriquer et concevoir (FabLabs, makerspaces, etc.). Cependant, il serait faux

INT_IoT_PrintOK_03.indd 26 03/11/16 12:03

de voir le phénomène de l'open hardware comme totalement nouveau : dans l'Histoire, certaines initiatives de partage de connaissances, savoir-faire, et technologies ont déjà eu lieu, comme avec les soyeux lyonnais au XVIII^{e} siècle. Cependant ces mises en commun étaient limitées à un groupe et une zone géographique, et généralement motivées par des aspects économiques plus qu'idéologiques.

L'open hardware dans l'IoT

Il est clairement plus simple de (ré) implémenter un protocole ou un logiciel en libre qu'un matériel. Cependant les ambitions des utilisateurs de matériel libre ont été revues à la baisse ce qui permet de donner un nouveau souffle à cette approche. Alors qu'il y a quelques années on tentait de cloner un processeur en libre (voir l'exemple – décevant – d'aeMB pour Microblaze), la tendance actuelle est de disposer des « sources » (schéma, nomenclature) d'une carte basée sur des composants du commerce (dont le processeur) afin d'en étendre les fonctionnalités (ou tout simplement d'en réaliser un copie). Le projet **Arduino** a suivi cette logique et c'est un succès planétaire. Créé par une équipe d'enseignants et d'étudiants en Italie, l'Arduino – ainsi que son environnement de développement – est devenu un standard de fait et les industriels de l'électronique – comme ST Microelectronics – commercialisent désormais des produits concurrents « compatibles Arduino ».

Un autre exemple connu est la **BeagleBone** Black de la communauté BeagleBoard qui utilise un System-On-Chip (SoC)

INT_IoT_PrintOK_03.indd 27 03/11/16 12:03

Sitara de TI et fonctionne sous Linux. Dans la même gamme, la Raspberry Pi est encore plus célèbre mais malgré ses qualités – et contrairement au apparences – elle n'est pas vraiment libre du fait de l'utilisation du SoC Broadcom. Ces cartes utilisent des extensions matérielles (*shield*) dont la plupart sont également libres. L'utilisation de telles cartes permet de réaliser une maquette d'objet connecté pour un investissement minimal si l'on utilise des outils comme Buildroot ou Yocto cités au paragraphe précédent.

Certains projets réels sont d'ores et déjà réalisés en modifiant (ou en « dérivant » pour reprendre un terme du logiciel libre) des cartes open hardware, quitte à modifier la qualité des composants. À titre d'exemple la carte mère de l'imprimante 3D Ember est basée sur une BeagleBone Black étendue [14].

De même le projet **WeIO** a été créé par une petite entreprise parisienne afin de réaliser une plate-forme générique pour l'IoT [15]. Outre son côté *open hardware*, la carte fonctionne sous **OpenWrt**, une distribution Linux dérivée de Buildroot. Le WeIO est fourni avec un environnement de développement permettant de le programmer en HTML5 ce qui est dans la logique de l'IoT qui est avant tout une extension de l'Internet (donc du Web) aux objets physiques.

INT_IoT_PrintOK_03.indd 28 03/11/16 12:03

Couche 2 – Accès au réseau/ protocoles

Nous avons déjà évoqué les protocoles de communication lors de la partie consacrée aux systèmes d'exploitation. Comme pour les OS, il convient de **séparer les objets de type capteurs des autres objets plus complexes**. Ces derniers utilisent des protocoles proches ou dérivés de ceux utilisés dans l'informatique classique et donc relativement ouverts (IP, TCP, UDP, Wi-Fi, Bluetooth, etc.). On peut citer **6LoWPAN**, adaptation d'IPv6, ou Bluetooth Low Energy (**BTLE**). À un niveau supérieur (*application layer*) on peut citer l'éternel **HTTP** ainsi que **CoAP**.

Les premiers ont des contraintes de fonctionnement fortes (faible consommation, isolés) et transmettent le plus souvent une très faible quantité de données (trame de température, pression, etc.) ou utilisent des réseaux de type WPAN (Wireless Personal Area Network) par exemple dans des applications de domotique. Plusieurs protocoles propriétaires (comme Zigbee ou Z-Wave) existent depuis de nombreuses années, entre autres pour réduire les coûts (et la consommation d'énergie) de matériel qui aurait pu utiliser un standard comme Bluetooth. Le développement de l'IoT a permis d'améliorer peu à peu les protocoles standards (BTLE, 6LoWPAN) ce qui pourrait à terme limiter l'utilisation de ces protocoles propriétaires. Autre exemple, le protocole *Thread* initié par Google via sa filiale Nest (qui produit des thermostats et détecteurs de fumée connectés) est basé sur 6LoWPAN.

INT_IoT_PrintOK_03.indd 29 03/11/16 12:03

La cas des capteurs isolés est différent car il est nécessaire de déployer un réseau dédié. Outre la définition de la technologie, cette dernière doit être déployée par des opérateurs. Les deux principales technologies à ce jour sont SIGFOX et LoRaWAN (toutes deux françaises). Le premier est entièrement propriétaire et SIGFOX met en place lui-même son réseau en France et s'appuie sur des partenaires dans les autres pays. Située à Toulouse, SIGFOX est un peu la coqueluche de la communauté « high tech » française depuis sa levée de fonds de 100 M€ en 2015. La technologie **LoRa** est quant à elle issue d'une start-up grenobloise (Cycleo désormais Semtech). Une alliance a été créée autour de cette technologie et rassemble de grands noms de l'industrie (Orange, Bouygues, SAGEMCOM, etc.). Contrairement à SIGFOX, la technologie est ouverte ce qui permet de trouver des composants libres (serveurs LoRaWAN) [15b].

Indépendamment de tout jugement technique de valeur, l'approche LoRa est bien entendu beaucoup plus conforme aux règles de fonctionnement du libre. Cependant il est tout à fait possible d'utiliser l'un ou l'autre des protocoles y compris pour une maquette, sachant que **plusieurs adaptateurs SIGFOX ou LoRa sont désormais disponibles pour des plates-formes comme la Raspberry Pi ou l'Arduino,** ce qui prouve que la réalisation d'une maquette IoT avec un tel matériel n'est pas totalement dénuée de sens !

Conscient de son retard, le comité de standardisation LTE 3GPP a considérablement accéléré sur le sujet de l'IoT pour définir une norme faisant partie intégrante de la 4G. Son nom est le

INT_IoT_PrintOK_03.indd 30 03/11/16 12:03

NB-IoT. Très peu de constructeurs télécoms supportent cette nouvelle norme sur leurs équipements aujourd'hui. Ce nouveau réseau est soutenu par plus de 20 opérateurs dans le monde, qui fournissent des communications à plus de 2,9 milliards de clients et servent géographiquement plus de 90 % du marché IoT [16]. Huawei et Vodafone souhaitent déployer ce réseau l'année prochaine (2017). Il est à noter que le seul éditeur de suite logicielle LTE français restant (après le rachat de Nokia par Alcatel) supporte déjà cette norme. En effet, après avoir annoncé le support du NB-IoT sur son eNodeB, la SAS **Amarisoft** envisage par la suite de fournir un kit de développement programmable par script, incluant un module NB-IoT reposant sur des puces génériques du marché (fin décembre 2016). Elle rejoint ainsi le cercle très restreint des constructeurs capables de fournir une solution NB-IoT de bout-en-bout.

Couche 3 – Cloud et traitement des données

La mise en œuvre d'une chaîne de valeur IoT nécessite, par définition, des services connectés ; c'est ce qui permet de créer une forte valeur ajoutée, au-delà des fonctions matérielles et embarquées. Cela se traduit par des briques d'interfaçage (API) couplées à :

- des composants de valorisation de données IoT : ingestion des remontées de données, traitement, stockage et exploitation des données ;

INT_IoT_PrintOK_03.indd 31 03/11/16 12:03

- des services digitaux à destination des IoT : enrichissement contextuel voire communautaire (aux niveaux objets et utilisateurs), actions, etc.

Par essence, la vague IoT se combine aux technologies de Big Data [17][18], de « machine learning » et de cloud.

Briques d'interfaçage

L'interfaçage des objets repose sur des standards ouverts tels que MQTT [19] (protocole de messagerie léger adapté à l'IoT) ou REST. Côté solutions, nous distinguerons les brokers d'ingestion Kafka *fluentd* (avec module MQTT), Mosquitto, ainsi que l'API Manager de WSO2.

Cependant, **le challenge de la connexion des objets en réseau repose sur bien plus que les protocoles bas niveau** et doit aller au-delà de la simple connexion d'un objet à une plate-forme serveur.

L'utilité d'un objet connecté est démultipliée par sa capacité à interagir avec d'autres objets. Pour rendre ces interactions possibles, **la question de l'identité globale unique des objets se pose.** Cette identité unique permet d'imaginer un système fédéré dans lequel les concepteurs d'objets peuvent utiliser leur propre plate-forme, tout en faisant participer leurs objets à un réseau plus vaste, grâce à cette identité.

À ce titre, **ProcessOne** développe une plate-forme d'interfaçage hybride et multi-protocoles. Cette plate-forme utilise **MQTT** pour connecter les objets en utilisant un protocole léger et

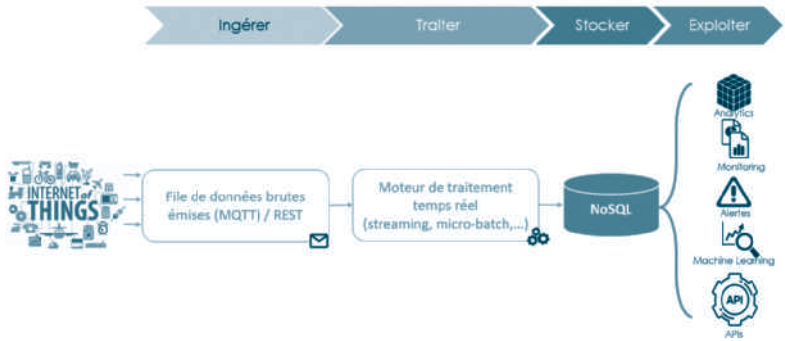
INT_IoT_PrintOK_03.indd 32 03/11/16 12:03

intègre le protocole **XMPP** (eXtensible Messaging and Presence Protocol : http://xmpp.org) pour la fédération de plates-formes serveur, les instructions de contrôles et la définition de l'identité des objets.

Cette vision doit conduire à éliminer les objets fonctionnant uniquement dans le silo limité du système de leur fournisseur et doit permettre de **les inscrire dans un réel écosystème, en s'appuyant sur des protocoles ouverts et des implémentations libres.** Il est possible de rêver de voir ses enceintes, sa télévision, son réfrigérateur, etc. participer à un workflow fluide entre objets de différentes marques, sans verrous propriétaires.

Valorisation de données IoT

La valorisation de données est au cœur de la majorité des stratégies et modèles économiques de l'IoT. Les modes d'exploitation des données peuvent être combinés afin d'apporter un maximum de pertinence métier.



Modèle d'architecture générale de valorisation de données IoT

INT_IoT_PrintOK_03.indd 33 03/11/16 12:03

Les architectures de valorisation de données reposent également majoritairement sur des briques Big Data, dont les composants majeurs du marché sont issus de l'écosystème du logiciel libre.

Intégration et stockage

Un système IoT, ne subit pas de sollicitations humaines comme sur un site web classique mais plutôt des actions automatiques à forte vélocité de plusieurs centaines de milliers voire millions d'objets. Les technologies pour supporter ces volumétries extrêmes (*Fast Data*) reposent sur les principes du *clustering* et du *sharding*. Au niveau des traitements d'intégration, Spark Streaming ou Wendelin est particulièrement adapté pour traiter des masses de données en temps réel, de manière répartie.

Pour le stockage, des bases NoSQL avec capacités de haute vélocité et de modélisation de séries temporelles (*time series*) telles que Cassandra ou NEO sont également adaptées.

À noter qu'il peut être utile de s'appuyer sur un cloud et des solutions DevOps [20], qui apportent une grande souplesse dans les évolutions de configuration de clusters, ceci afin de gérer la capacité et ainsi accompagner les montées en charge, parfois brutales. Les technologies de DevOps ont également permis de réduire considérablement la taille d'une équipe d'exploitation et de mettre ainsi le déploiement d'un système d'IoT à la portée d'une petite équipe ou d'une PME [21].

INT_IoT_PrintOK_03.indd 34 03/11/16 12:03

Monitoring et alertes

Les solutions de monitoring et d'alertes, telles Elastic, offrent la possibilité de visualiser l'évolution en temps réel de valeurs : métriques métiers, états de santé d'un parc d'objets, etc. Les alertes peuvent être combinées à de l'apprentissage statistique (*machine learning*), afin de notifier des écarts par rapport à un modèle statistique optimal.

Machine learning

Les solutions de machine learning, exploitant les masses de données issues des objets, permettent de définir :

- des modèles d'usages et de potentiels écarts par rapport à des modèles optimaux,
- une prédiction de valeur en appliquant un ou plusieurs modèles statistiques.

Cette pratique est courante dans les domaines énergétiques, infrastructures, industriels et des transports.

Parmi les solutions de machine learning, nous distinguerons les langages R et Python (notamment la bibliothèque d'apprentissage Scikit-learn qui a fait l'objet d'une journée PyData Paris lors de l'OSIS 2016 [4]), le moteur réparti Spark MLLib ainsi que les notebooks (interfaces web permettant de développer de manière itérative et collaborative des analyses statistiques) Jupyter [22], Zeppelin et ShinyR.

INT_IoT_PrintOK_03.indd 35 03/11/16 12:03

Analytics

Des outils OLAP, tels que Saiku Analytics, offrent la possibilité à un analyste métier de ventiler, en glisser-déposer, les métriques collectées suivant différents axes d'analyse.

API

Les données issues de l'IoT (brutes ou valorisées) peuvent également être projetées via des API à destination d'autres composants du SI (étendu). **Ici encore, dans la démarche de valorisation, l'identité et la fédération des objets peuvent jouer un rôle central.**

Services digitaux à usage de l'IoT

Les services digitaux peuvent être publiés par le système central ou bien entre objets.

Dans le premier cas, ces services permettent d'enrichir de manière contextuelle (météo locale, autres utilisateurs de la solution alentour) les objets connectés d'informations générales ou centralisées, sous forme de collections d'APIs. Ils permettent également de **déporter des tâches complexes, nécessitant des ressources importantes,** comme résultats d'algorithmes (machine learning) permettant d'optimiser le fonctionnement de l'objet (calculs déportés d'un thermostat connecté). Enfin ces services permettent des mises à jour applicatives.

Dans le deuxième cas, ces services entre objets d'un même écosystème de marque ou d'un autre écosystème (interaction

INT_IoT_PrintOK_03.indd 36 03/11/16 12:03

entre ville connectée et véhicule connecté) permettent d'enrichir les informations de contexte (réseau de capteurs dans une maison pour détecter une présence) ou bien de passer des ordres (réserver une place de parking).

Dans les deux cas, **l'utilisation de protocoles standards et ouverts permet de faciliter le développement d'écosystèmes autour d'une solution.**

Pour déployer : outils et frameworks d'intégration

L'intégration de toutes les couches fonctionnelles décrites peut s'avérer complexe dans le cas du déploiement réel d'un système IoT. Un outil d'intégration global (ou « framework ») est alors fortement recommandé. Cela induit souvent quelques contraintes car le framework ne peut prendre en compte toutes les configurations possibles (OS, langages, protocoles, etc.). Du fait du marché prometteur de l'IoT, de nombreux éditeurs ou fabricants de matériels se sont engouffrés dans la brèche et il existe d'ores et déjà des dizaines de produits. L'offre libre est bien entendu plus réduite mais a le gros avantage de s'appuyer sur des communautés (Eclipse et fondation Linux).

À titre d'exemple le framework **Kura** [23][24] est maintenu par la fondation Eclipse. L'utilisation de Kura permet bien entendu de profiter de l'environnement intégré Eclipse qui est devenu un standard industriel disponible y compris pour des outils

INT_IoT_PrintOK_03.indd 37 03/11/16 12:03

comme Yocto. Selon l'étude [6] déjà citée, Java est le langage de prédilection de l'IoT (52 %, juste devant le langage C à 48 %) alors qu'il n'est pas vraiment celui des systèmes embarqués. La couple Java/embarqué est tumultueux depuis longtemps car l'empreinte mémoire utilisée par la machine virtuelle (JVM) est traditionnellement assez importante même s'il en existe une version réduite (Java ME). La situation a un peu évolué grâce à Android et sa version optimisée du bytecode utilisée dans Dalvik puis ART.

Kura intègre cependant des standards de l'IoT comme Linux en tant qu'OS (auquel il faut ajouter la JVM) ainsi que le protocole MQTT déjà évoqué. La documentation est bien fournie et propose des cas d'exemples sur des plates-formes simples d'accès comme la Raspberry Pi.

Le framework **IoTivity** [25] soutenu par la fondation Linux est une autre solution libre permettant l'intégration simplifiée d'objets dans un réseau tout en étant multi plate-forme et multi-langages (C, C++, Java). Il constitue l'implémentation de référence des spécifications du consortium OIC (*Open Interconnect Consortium*) rassemblant des leaders de l'industrie comme Intel ou SAMSUNG. IoTivity est compatible avec Tizen [26], un système d'exploitation (déjà cité) proche de Linux et disposant de plusieurs profils (mobile, TV, wearable) le rendant utilisable dans de nombreux produits grand publics. Tizen est déjà largement utilisé dans les produits SAMSUNG.

INT_IoT_PrintOK_03.indd 38 03/11/16 12:03

Le choix d'un framework propriétaire peut cependant être justifié dans le cas de l'utilisation d'une chaîne de production spécifique et totalement intégrée couvrant tous les niveaux évoqués. Un exemple est la solution SYNERGY de RENESAS, leader mondial de l'électronique embarqué automobile qui intègre un grand nombre de composants matériels mais aussi de composants logiciels « métier » comme AUTOSAR ou ITRON. De même le framework construit autour de MicroEJ (une version optimisée de JVM) permet de simplifier largement le déploiement d'un réseau d'objets fonctionnant sous Java.

Pour exploiter : outils et frameworks de gestion d'IoT

L'IoT n'est pas uniquement une affaire de technique mais aussi de modèle d'affaires (voir chapitre suivant). De grands acteurs du monde propriétaire tels que Thomas Siebel ou CISCO l'ont bien compris en lançant dès 2009 une nouvelle génération de logiciels de gestion dédiés à l'IoT. Ces logiciels permettent notamment de suivre l'*ensemble du cycle logistique et d'exploitation* des objets connectés en y intégrant parfois une dimension de monétisation et de traçabilité des métadonnées. Bien qu'ils soient encore peu connus dans le monde des développeurs et notamment des *data scientists*, ils recueillent les faveurs des dirigeants d'entreprise qui y voient un moyen plus évident d'atteindre leurs objectifs financiers liés à l'IoT.

INT_IoT_PrintOK_03.indd 39 03/11/16 12:03

Le framework Wendelin de la société Nexedi, dérivé d'une solution de cloud décentralisé datant de 2010, est l'un des rares logiciels libres de gestion d'IoT disponibles sur le marché. Il est actuellement déployé en Allemagne pour surveiller une centaine d'éoliennes connectées [27].

INT_IoT_PrintOK_03.indd 40 03/11/16 12:03

Chapitre 3

Vers un réel modèle d'affaires ?

Le modèle économique de l'IoT n'est pas encore clairement défini, du moins au niveau du marché de masse. Le marché de l'IoT industriel trouve sa rentabilité dans les économies générées par l'amélioration des processus et de la qualité. En revanche, les produits dits « connectés » (balance, montres, etc.) connaissent un succès mitigé qui rappelle la période des micro-ordinateurs avant l'IBM-PC. Comme à cette époque, l'incertitude règne quant à l'utilité réelle de ces produits et la sécurité des données qui y transitent.

À ce jour le seul objet connecté grand public dont le modèle économique est en place (et rentable) est le smartphone puisque la grande majorité de la population en possède un, de « gré ou de force » car de nombreux services sont désormais exclusivement disponibles (ou presque) via cet outil et ses fameuses applications. Les autres fonctionnalités sont intégrées à des biens de consommation déjà acquis par l'utilisateur (Internet/TV, domotique, automobile). Hormis l'Internet/TV, le modèle économique lié à ces nouveaux marchés proches de l'IoT reste à prouver. À titre d'exemple le constructeur Tesla Motors – présenté comme leader de l'automobile du futur – a

INT_IoT_PrintOK_03.indd 41 03/11/16 12:03

vendu un peu moins de 30 000 véhicules au premier semestre 2016 (quand il espérait en vendre 33 000).

En outre le smartphone est l'interface privilégiée de certains objets (capteurs) connectés en local par Bluetooth ou Wi-Fi, le smartphone faisant office de passerelle pour placer les données sur le cloud. **Le smartphone tend donc à vampiriser le marché de l'IoT grand public de par sa polyvalence** (comme auparavant le PC). Il suffit souvent d'une simple application pour remplacer des objets spécialisés dont le coût de conception et de maintien à jour est pénalisant pour le constructeur. À titre d'exemple le marché des GPS a été laminé par les applications sur smartphone [28] dont Waze (Google) qui est probablement le meilleur outil du genre car il profite des données de Google et remplit de plus la fonction d'assistant afin de prévenir des « dangers de la route » comme le fait son concurrent local Coyote (qui fonctionne sur une plate-forme dédiée bien moins puissante qu'un smartphone actuel).

Cependant, outre qu'on pourrait un jour imaginer de monnayer la garantie de *privacy* que seul peut offrir un équipement dédié, certains fabricants de GPS tirent leur épingle du jeu en fournissant des licences de leur logiciel aux équipementiers ou constructeurs automobiles dans un domaine où l'hégémonie de Google fait peur. Si tel n'était pas le cas, il est probable que Google (via son OS Android) aurait déjà envahi ce marché (et il n'est pas dit que cela n'arrive pas car c'est une volonté forte de Google). Dans le domaine de la TV numérique, notons que deux fournisseurs majeurs (Free et Bouygues Telecom)

INT_IoT_PrintOK_03.indd 42 03/11/16 12:03

proposent d'ores et déjà une set-top box de type Android/TV (Freebox Mini 4K et Bbox Miami). Bref, dans un bon nombre d'applications classiques (utilisant simplement un écran et aucune interface spéciale) il est bien difficile de concurrencer le smartphone qui fournit la même fonctionnalité, souvent de meilleure qualité et sans surcoût pour l'utilisateur – sans compter la facilité de mise à jour.

Le salut du marché de l'IoT peut venir de solutions originales impossibles à mettre en place sur le matériel/logiciel existant car nécessitant des interfaces spéciales. Lors du dernier OSIS du GTLL en 2016 [4], la société Tarkett – leader mondial des revêtements de sol – a présenté un « sol connecté » permettant de détecter les chutes de personnes dans des lieux médicalisés par une technique de capteurs piézoélectriques intégrés au sol. Les données sont collectées sur le cloud et consultables sur smartphone ou tablette, qui n'est dans ce cas qu'un terminal banalisé, et n'est pas au centre de la valeur ajoutée.

Le domaine médical est probablement l'un des rares marchés liés au grand public et connaissant un démarrage significatif car l'IoT permet d'y limiter les frais liés à des manipulations simples ou automatisables (mesures effectuées par le patient ou bio-capteurs). Dans ce cas une passerelle autonome s'impose car elle permet d'automatiser les transferts de données sans passer par une application, cette dernière étant utilisable – souvent par un autre utilisateur valide – comme un terminal de consultation. En effet, il faut prendre en compte que

INT_IoT_PrintOK_03.indd 43 03/11/16 12:03

la population concernée (personnes âgées) n'est pas toujours capable d'utiliser une application mobile.

Ainsi la solution UCare destinée aux personnes âgées et aux travailleurs isolés a choisi de partir sur une solution matérielle faite maison plutôt que de s'appuyer sur une pile logicielle, de manière à être beaucoup plus légère (tournant sur un MCU Cortex ARM M0 + et FreeRTOS, une autre perle en provenance de l'open source).

Autre exemple, la société Nest – déjà évoquée – doit bien entendu son succès à son acquisition par Google (qui reste à ce jour un modèle économique éprouvé !) mais également au fait que ses produits sont liés à des capteurs matériels. Il faut cependant tenir compte du budget réduit du client final (un détecteur de fumée Nest coûte plus de 100€) qui n'est pas spécialement un technophile et dont le budget « connexion et services numériques » est le plus souvent déjà attribué au smartphone, connexion Internet et TV numérique.

INT_IoT_PrintOK_03.indd 44 03/11/16 12:03

Conclusion

Il est difficile d'entrevoir à l'heure actuelle pour l'IoT un marché de masse comparable à celui de la téléphonie/Internet car bon nombre de solutions sont déjà couvertes sur ce marché existant et l'utilisateur n'est pas forcément prêt à augmenter ses dépenses pour en profiter.

Le cas du marché professionnel est très différent car de nombreux concepts sont en place depuis longtemps. On a collé un nom (IoT) sur un concept qui existe – certes de manière partielle – depuis plusieurs années (M2M et « usine du futur »). Le modèle économique est totalement différent puisque dans ce cas il n'est pas lié à des revenus de masse mais à une optimisation des coûts. On reste donc sur de la fourniture d'outils et de prestations, points sur lesquels le logiciel (et désormais le matériel) libres ont déjà fait leurs preuves.

INT_IoT_PrintOK_03.indd 45 03/11/16 12:03

Bibliographie

[1] « Et Dieu créa l'Internet » (Christian Huitéma)

[2] Code source openPOWERLINK https://sourceforge.net/projects/openpowerlink/files

[3] Citation de Jean-Marie DAUTELLE (Airbus) à la conférences IoT / Aerospace Valley (26 avril 2016) http://www.aerospace-valley.com/agenda/journ%C3%A9e-liot-dans-les-domaines-a%C3%A9ronautique-spatial-automobile-sant%C3%A9

[4] Journée OSIS 2016 sur l'IoT http://www.open-source-innovation-spring.org/open-source-pour-linternet-des-objets/

[5] IoT Developer Survey 2015 http://fr.slideshare.net/IanSkerrett/iot-developer-survey-2015

[6] IoT Developer Survey 2016 http://iot.ieee.org/images/files/pdf/iot-developer-survey-2016-report-final.pdf

[7] « How standards proliferate » https://xkcd.com/927

[8] « Android is ousting Windows from its last mobile bastion » http://www.infoworld.com/article/3014357/android/android-is-ousting-windows-from-its-last-mobile-bastion.html

[9] Windows (10) IoT https://developer.microsoft.com/fr-fr/windows/iot

[10] RioTboard http://riotboard.org

INT_IoT_PrintOK_03.indd 46 03/11/16 12:03

[11] Composants libres pour caméra GoPro https://fr.gopro.com/help/articles/block/Open-Source-Software

[12] Solution Nabto/FreeRTOS http://www.freertos.org/FreeRTOS-Plus/Nabto/what_is_freertos_plus_nabto.shtml

[13] Lepton http://o10ee.com/lepton

[13b] Lapeyre, *M. Poppy : Open source 3D printed and modular humanoid robot for Science, Art and Education*. Inria, 2014

[14] Ember 3D printer http://learn.ember.autodesk.com/blog/ember-open-source-electronics-and-firmware

[15] WeIO http://we-io.net/hardware

[15b] Protocole LoRa 1.0 https://www.aruco.com/2015/06/lora-alliance-lorawan-r10/

[16] http://www.servicesmobiles.fr/nouveau-reseau-nb-iot-concurrent-direct-lora-et-sigfox-33889/

[17] Livre blanc Big Data Smile http://www.smile.fr/Ressources/Livres-blancs/Erp-et-decisionnel/Big-data

[18] Article Smile Iot et Big Data http://blog.smile.fr/Integrer-des-donnees-d-iot-en-temps-reel-avec-talend-real-time-big-data

[19] MQTT http://mqtt.org

[20] Livre blanc Smile DevOps http://www.smile.fr/Ressources/Livres-blancs/Systeme-et-infrastructure/Devops-et-industrialisation

INT_IoT_PrintOK_03.indd 47 03/11/16 12:03

[21] https://www.nexedi.com/success/slapos-IMT-Documents.Teralab.Success.Case

[22] Notebook Jupyter http://blog.smile.fr/Introduction-au-notebook-jupyter

[23] Framework KURA http://www.eclipse.org/kura

[24] KURA/M2M par Eurotech http://fr.slideshare.net/Eurotechchannel/kuram2miotgateway

[25] Wiki IoTivity https://wiki.iotivity.org/community

[26] IoTivity et Tizen http://fr.slideshare.net/SamsungOSG/iotivity-on-tizen-how-to

[27] https://www.nexedi.com/NXD-Presentation.IoT.Wind.Energy?portal_skin = CI_slideshow#/

[28] Plates-formes de navigation GPS http://www.yext.com/fr/resources/pocket-guide-to-the-us-gps-navigation-market/current-navigation-platforms

INT_IoT_PrintOK_03.indd 48 03/11/16 12:03