

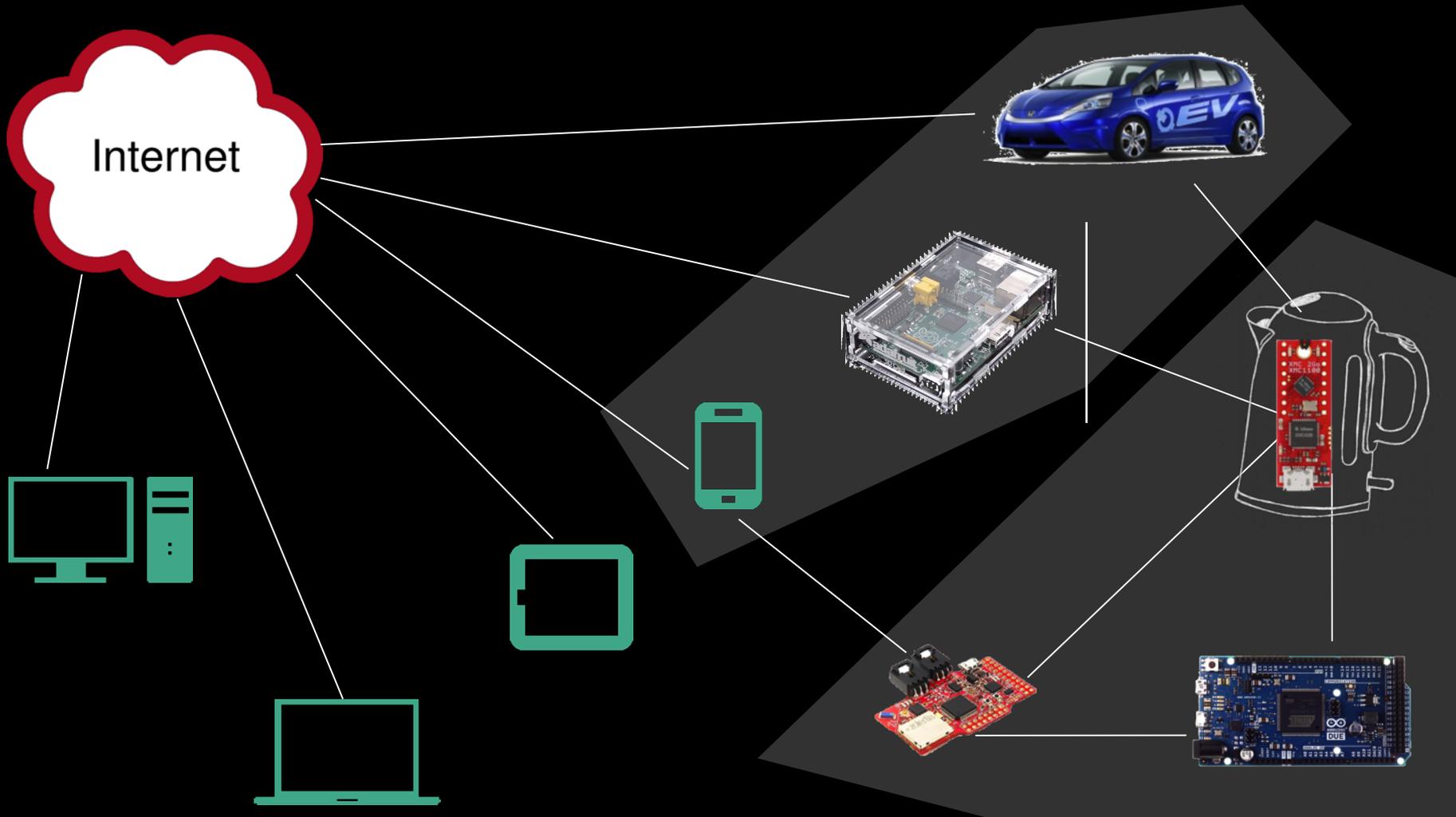


OS libre pour l'IoT

Alexandre Abadie

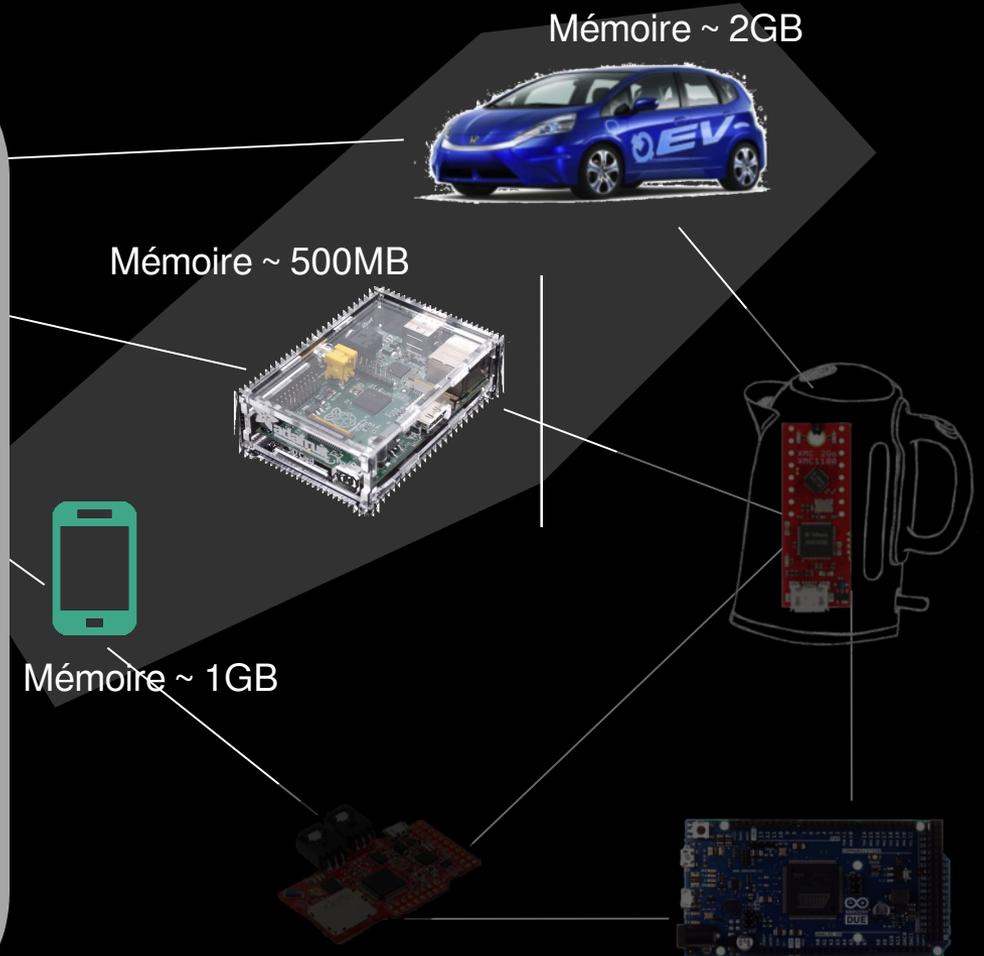
Inria

The Internet of Things (IoT)



Les objets de haut niveau

- Nano-ordinateurs
→ Raspberry Pi, véhicules connectés, smartphones...
 - Ressources équivalentes aux routeurs Internet
→ mémoire, CPU, débit réseau, etc
- **Supportent les protocoles TCP/IP classiques**
- **Supportent les OS « classiques » comme Linux**

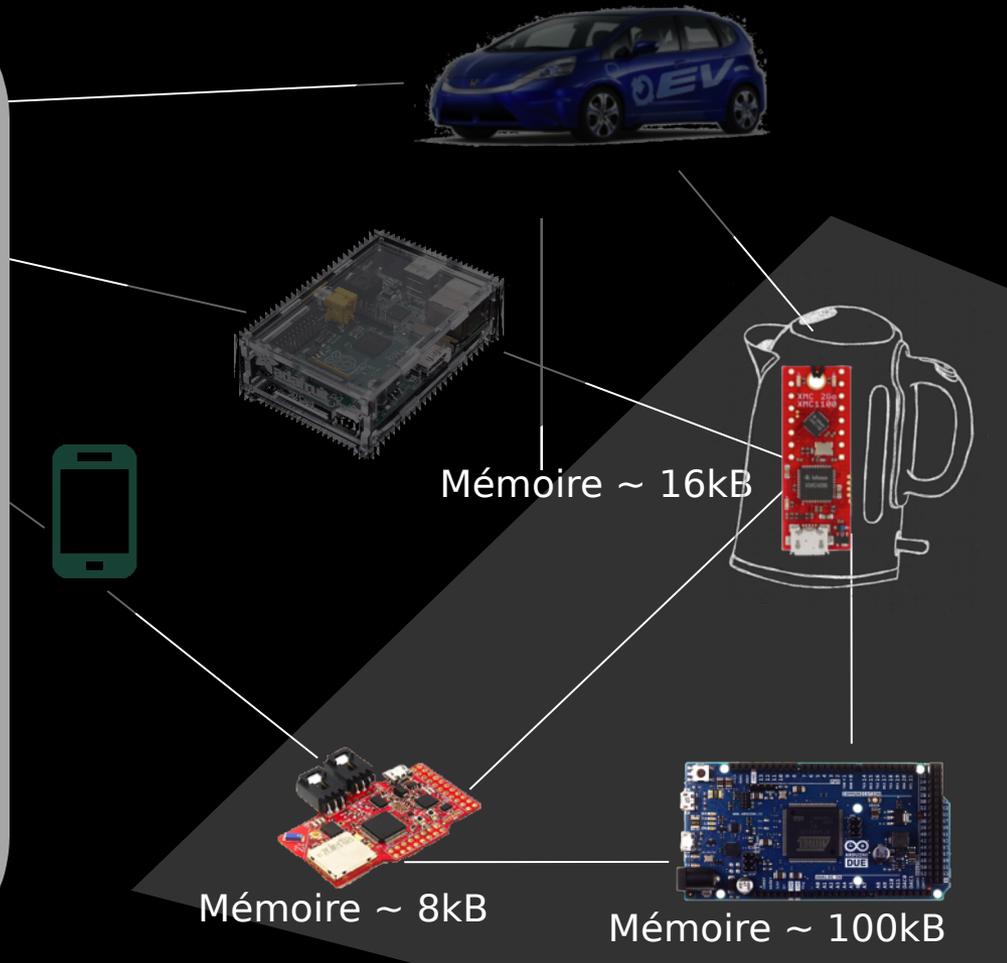


Les objets contraints

- Objets intelligents **plus petits et bon marché**
- Microcontrôleurs et radios **basse consommation**
→ utilisables sur batterie

Energy: Milliwatt instead of Watt
CPU: Megahertz instead of Gigahertz
Memory: Kilobytes instead of Gigabytes

- **Ne supportent pas TCP/IP tel quel**
- **Ne supportent un OS comme Linux : pas de MMU**

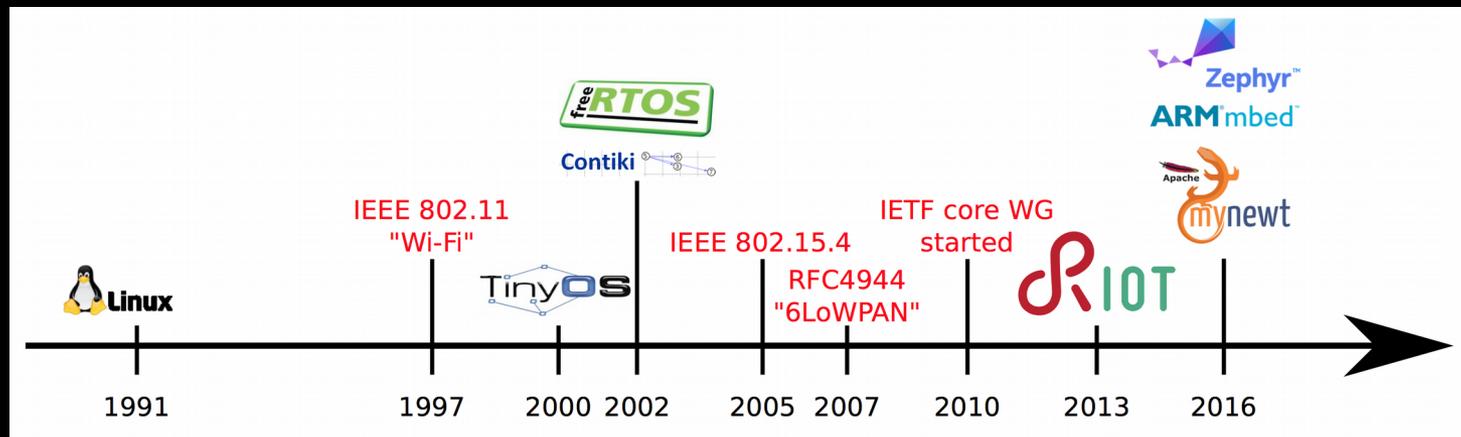


Pourquoi un OS pour objets contraints ?

- ~~Linux~~, ~~Android~~... **bare-metal** ?
- Besoin de **suivre les évolutions** de l'IoT
 - Ajout de briques logicielles de + en + complexes : nouvelles piles protocolaires, IP, sécurité
 - Applications de + en + complexes ≠ **espace restreint**
- ... il faut des **logiciels IoT portables** pour suivre ces évolutions
 - Possible avec une bonne plateforme logicielle
 - Très difficile avec des développements bare-metal

Les OS existants

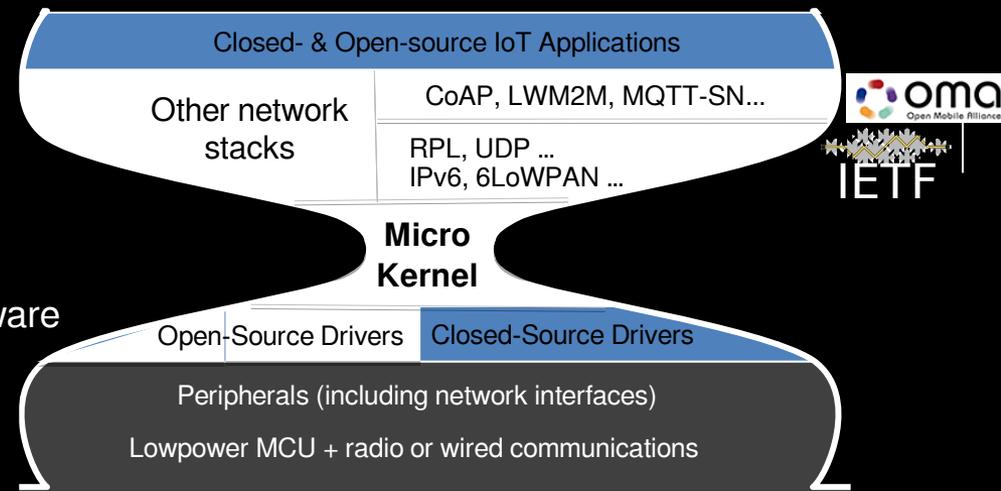
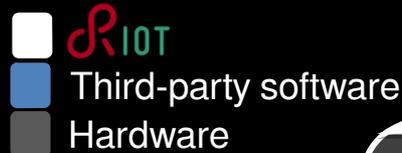
- Contiki
- RIOT
- TinyOS
- mbedOS (ARM)
- Zephyr (Intel)
- LiteOS (Huawei)
- ...
- ... and closed-source alternatives



Reference: O. Hahm et al. "Operating Systems for Low-End Devices in the Internet of Things: A survey," IEEE Internet of Things Journal, 2016.

R-IoT : un OS adapté aux objets contraints

- Une plateforme pour des applications **performantes, portables et communicantes**
- Effort concentré sur le micro-noyau temps-réel, les piles réseau et les pilotes matériels
- Architecture modulaire
→ ajout aisé de code propriétaires ou open-source
- API consistante



Performant

- Architecture **micro-noyau**
 - Empreinte mémoire très faible **~1KB** sur ARM 32-bit
 - Ordonnanceur « tickless »
 - **faible consommation**
 - Ordonnanceur déterministe $O(1)$
 - **temps-réel**
 - Gestionnaire d'interruptions rapide
 - **réactivité**
- Comportement **multi-thread**
 - Taille minimale du « Thread Control Block »
 - Utilisation minimale de la pile
 - Communication inter-processus (IPC) efficace

Interopérabilité système

- Code écrit dans un langage standard : C ANSI
- Patterns standards de programmation : multi-threading
- S'appuie sur des outils libres
Makefile, OpenOCD, GDB
- Système de paquetages externes



- Effort de portage minime
- Utilisation de bibliothèques tierces
lwIP, spiffs, u8g2, tinyDTLS

Package	Diff Size	
	Overall	Relative
libcoap	639 lines	6.3 %
libfixmath	34 lines	0.2 %
lwip	767 lines	1.3 %
micro-ecc	14 lines	0.8 %
relic	24 lines	<0.1 %

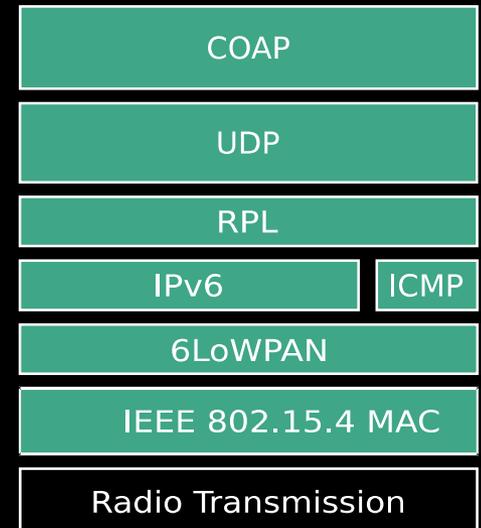
Un large support matériel

- Support pour architectures 8, 16, 32 bit parmi ARM, AVR, MIPS
- + de 80 cartes du marché supportées
- nombreux pilotes matériels : radios, capteurs, actionneurs



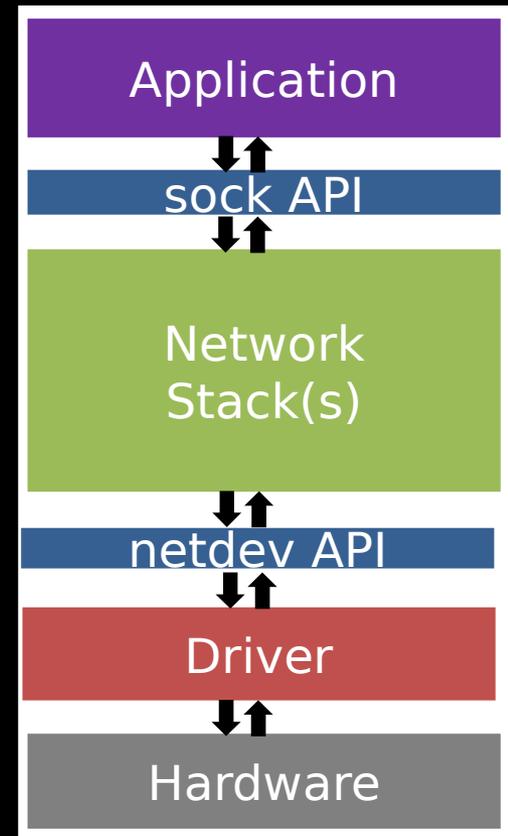
Protocoles réseau : la jungle

- Spécifications pour la **couche physique**
Radios basse conso, PLC, BACnet
IEEE 802.15.4, Z-Wave, BLE, LoRa, SigFox (et IEEE 802.11)
- Spécifications pour la **couche réseau**
Interopérables avec IP et compatibles IoT
6TiSCH, 6LoWPAN, RPL, OLSRv2, AODVv2
- Spécifications pour la couche **application**
Interopérables avec le web et compatibles IoT
CoAP, LwM2M, MQTT-SN, CBOR
Sécurisé avec DTLS, OSCOAP
- **Nouveaux paradigmes réseau**
Information-centric network (ICN)



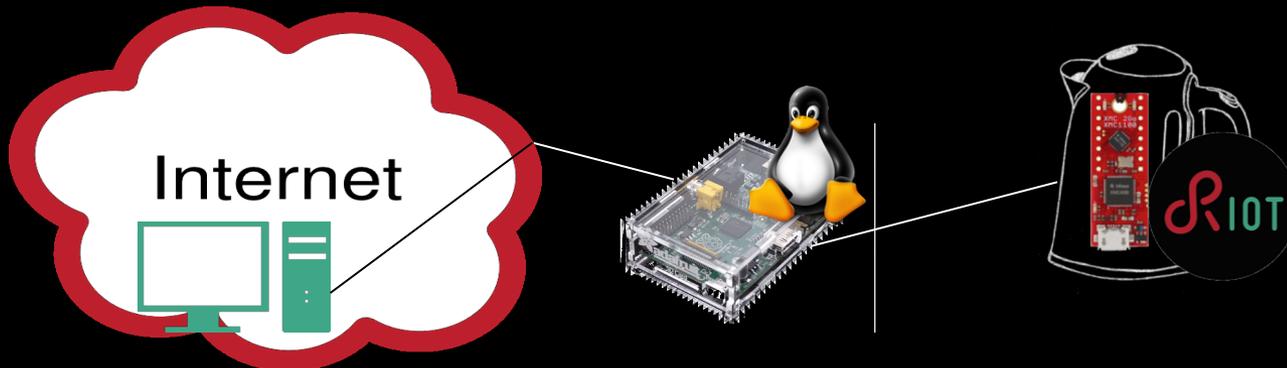
Interopérabilité réseau

- Pile par défaut : 6LoWPAN
 - GNRC : Generic Network Stack
- Paquetages tierces
 - Pile lwIP
 - Pile uIP (emb6)
 - Pile Thread (OpenThread)
 - ...
- Piles expérimentales
 - CCN-lite
 - NDN-RIOT



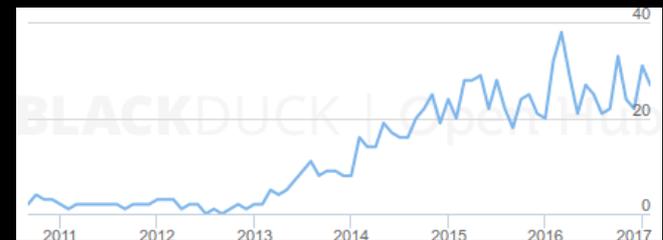
De la sécurité

- **Qualité logicielle**
 - ✓ Procédure de revue de code sur GitHub
 - ✓ Utilisation d'outils d'analyse statique du code (cppcheck)
 - ✓ Intégration continue distribuée
- Procédure interne de **remontée** et de **correction des failles**
- En combinant RIOT et Linux
 - ✓ Solution 100% open-source → **confiance**
 - ✓ Solution sécurisé → **nœuds contraints isolés d'Internet**
 - ✓ Standards ouverts de communication



La communauté

- Similaire à **Linux**
 - Libre => **LGPL**
 - Pilotée par sa base de développeurs => **indépendance**
- **Variée**
 - **Académiques** : Inria, FU Berlin, UH Hambourg, UCLA, Berkeley, MIT, TZI
 - **PME** : Zolertia, Loci Controls, OTAKeys, Mesotic, EisoX
 - **Industriels** : Cisco, Samsung, ImgTec
 - Makers
 - Développeurs indépendants
- **Active**
 - **140** contributeurs depuis les débuts en 2013
 - Dont 90 sur la dernière année
 - Release stable tous les 3 mois



La communauté

- Rencontre annuelle : RIOT Summit
- Cette année les 25 & 26 septembre @ FU Berlin

<http://summit.riot-os.org/2017/>



Quelques fonctionnalités

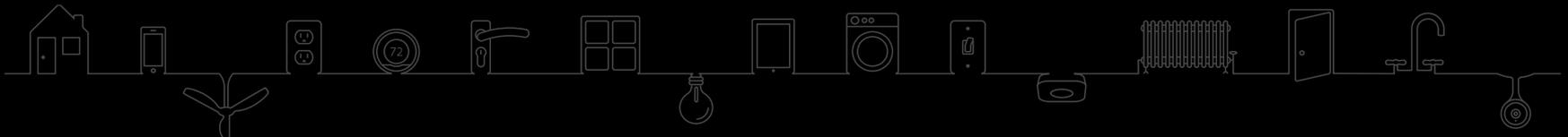
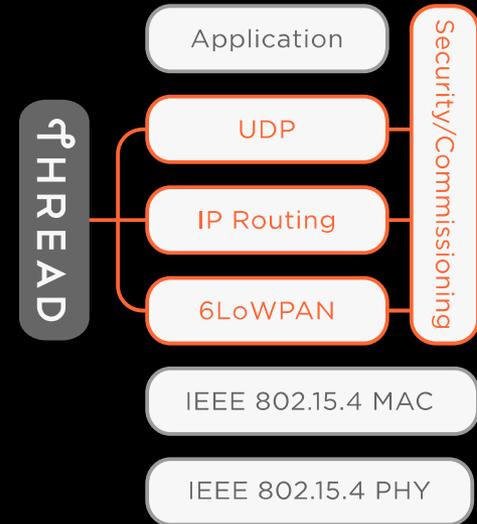
- Large support matériel :
 - ✓ Support de (presque) toute la gamme de cartes **ST Nucleo**
 - ✓ Support de (presque) tous la gamme de cartes Arduino
 - ✓ Support de l'architecture **MIPS (PIC)**
 - Drivers matériels : radios, capteurs, actionneurs
- ✓ Implémentations réseau :
 - ✓ MQTT-SN
 - ✓ CoAP : modes client, protocole OBSERVE
 - ✓ **OpenThread**
 - ✓ Bus CAN
- Drivers et systèmes de fichiers pour les supports de stockage SPIFFS, FATFS

Toutes intégrées sur les 12 derniers mois !

OpenThread



- Utilisable sur les radios 802.15.4
- Intégré comme pile protocolaire externe
- Adapté aux usages domotiques
- Sécurisé



Travaux en cours

- Support des technologies sans-fil « long range »



- Drivers pour modules SEMTECH
- Support LoRaWAN

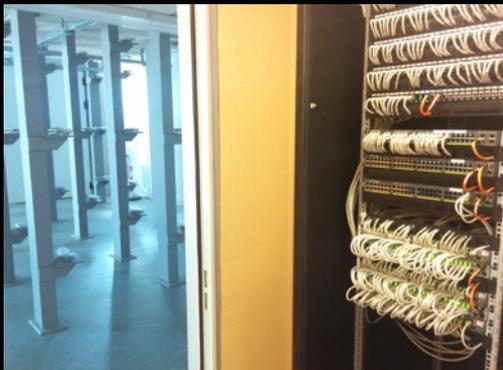


- Drivers pour modules ATA8520
 - Arduino MKRFOX1200
- PR : <https://github.com/RIOT-OS/RIOT/pull/7093>

- Drivers matériels pour le bus CAN (automobile)
- Mises à jour de firmware « Other The Air » (OTA)
PR : <https://github.com/RIOT-OS/RIOT/pull/6924>

RIOT en action

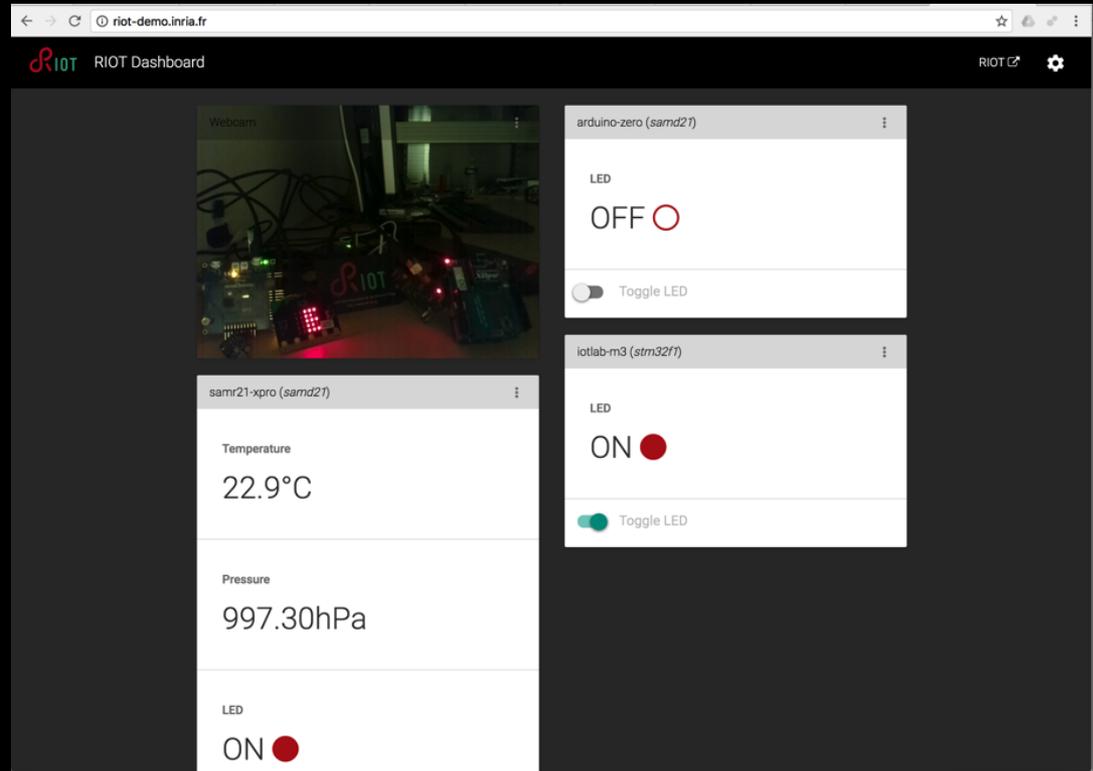
Testez RIOT sur les plate-formes d'expérimentation FIT/IoT-Lab
<https://www.iot-lab.info>



A screenshot of the FIT IoT-Lab website. The page is titled "FIT IoT-Lab" and has a navigation menu with "NEWS", "PLATFORM", "DEV CENTER", "COMMUNITY", and "GET STARTED". There is also an "ACTIVITY" button and a "Access the testbed" button. The main content area is divided into sections: "RIOT" (with sub-sections for M3 nodes, A8-M3 nodes, and CoAP server), "OPENWSN" (with sub-sections for M3 and A8-M3 nodes and Testing Board Support), and "Tools" (with sub-sections for CLI tools, Experiment CLI Client, and Node CLI Client). Each sub-section includes a brief description and a small icon representing the topic.

RIOT en action

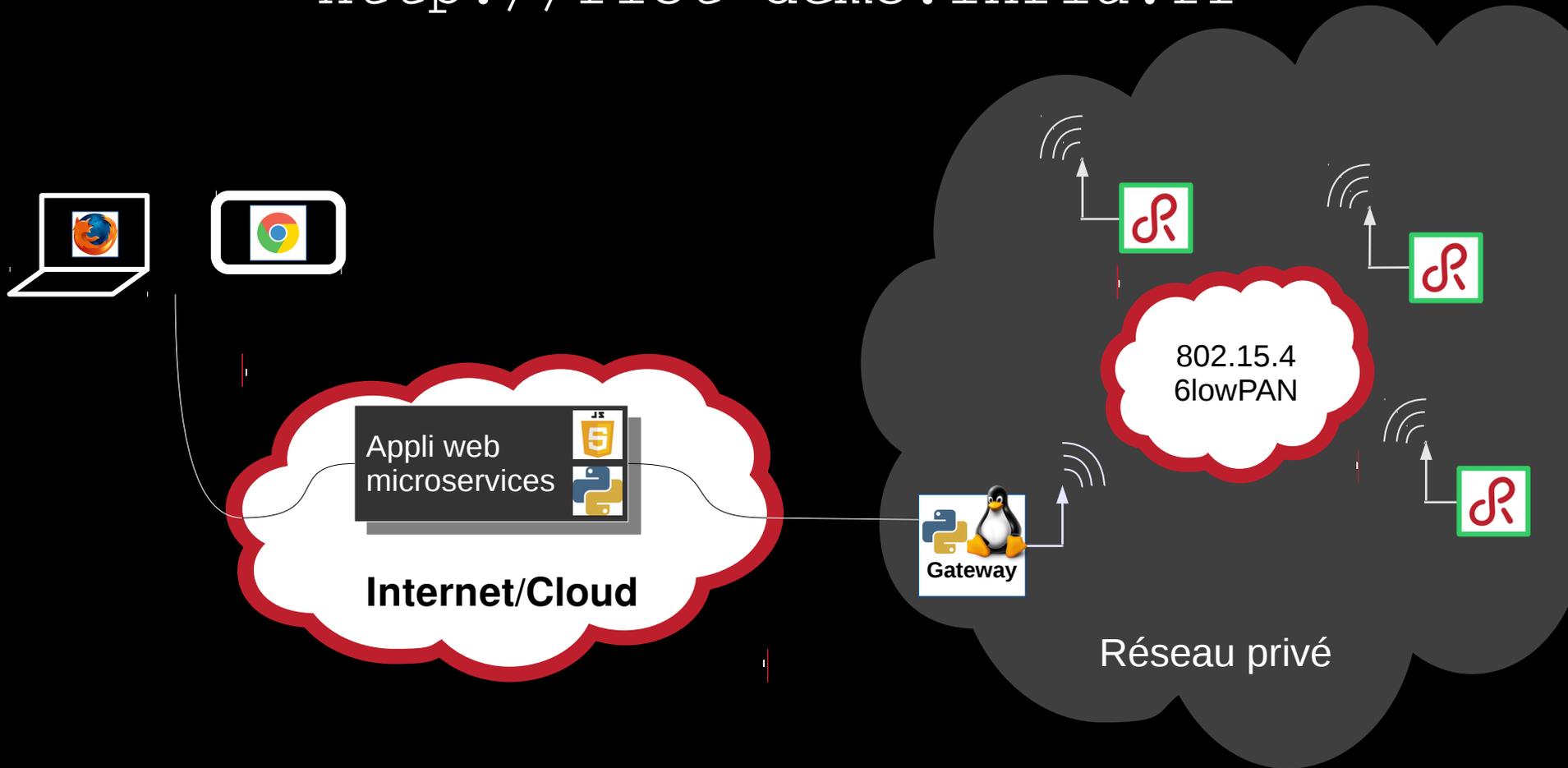
- ✓ Matériel du commerce
- ✓ Tableau de bord web
- ✓ COAP/IPv6/6LoWPAN
- ✓ Open source de bout en bout
- ✓ Utilise des standards de communication



<http://riot-demo.inria.fr>

Démo

<http://riot-demo.inria.fr>



<https://github.com/pyaiot/pyaiot>

Merci!

Suivez RIOT

https://twitter.com/RIOT_OS

Contribuez

<https://github.com/RIOT-OS/RIOT>

Support & discussions sur IRC

[#riot-os](irc.freenode.org)

