



---

# LES OBJETS COMMUNICANTS

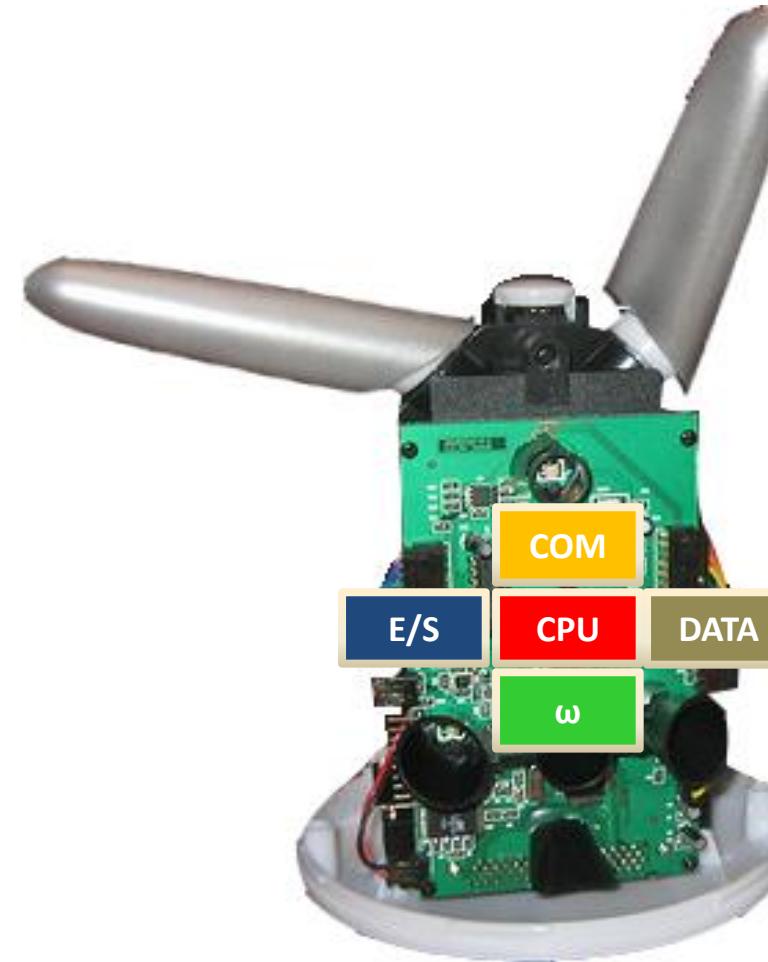
Jean-Paul Jamont, Lionel Médini, Michaël Mrissa

# COMMENT FONCTIONNENT LES OBJETS COMMUNICANTS?

---

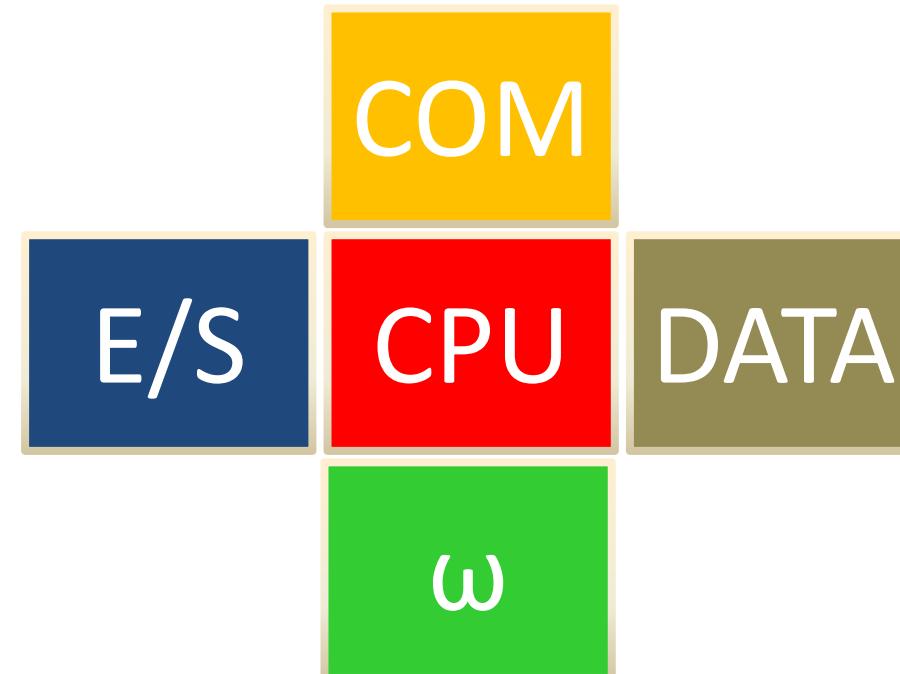


# COMMENT FONCTIONNENT LES OBJETS COMMUNICANTS?

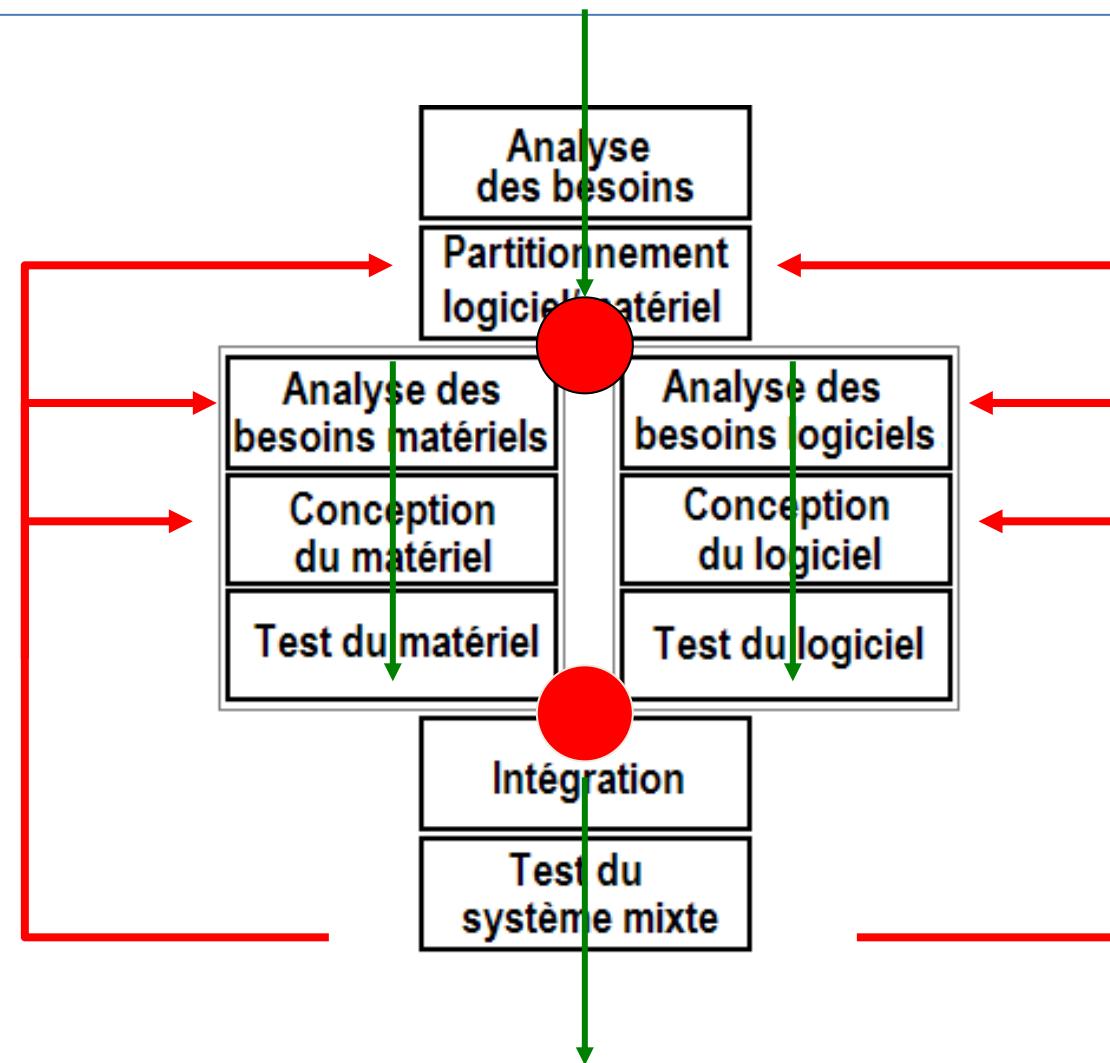


# COMMENT CRÉER SES PROPRES OBJETS COMMUNICANTS?

---

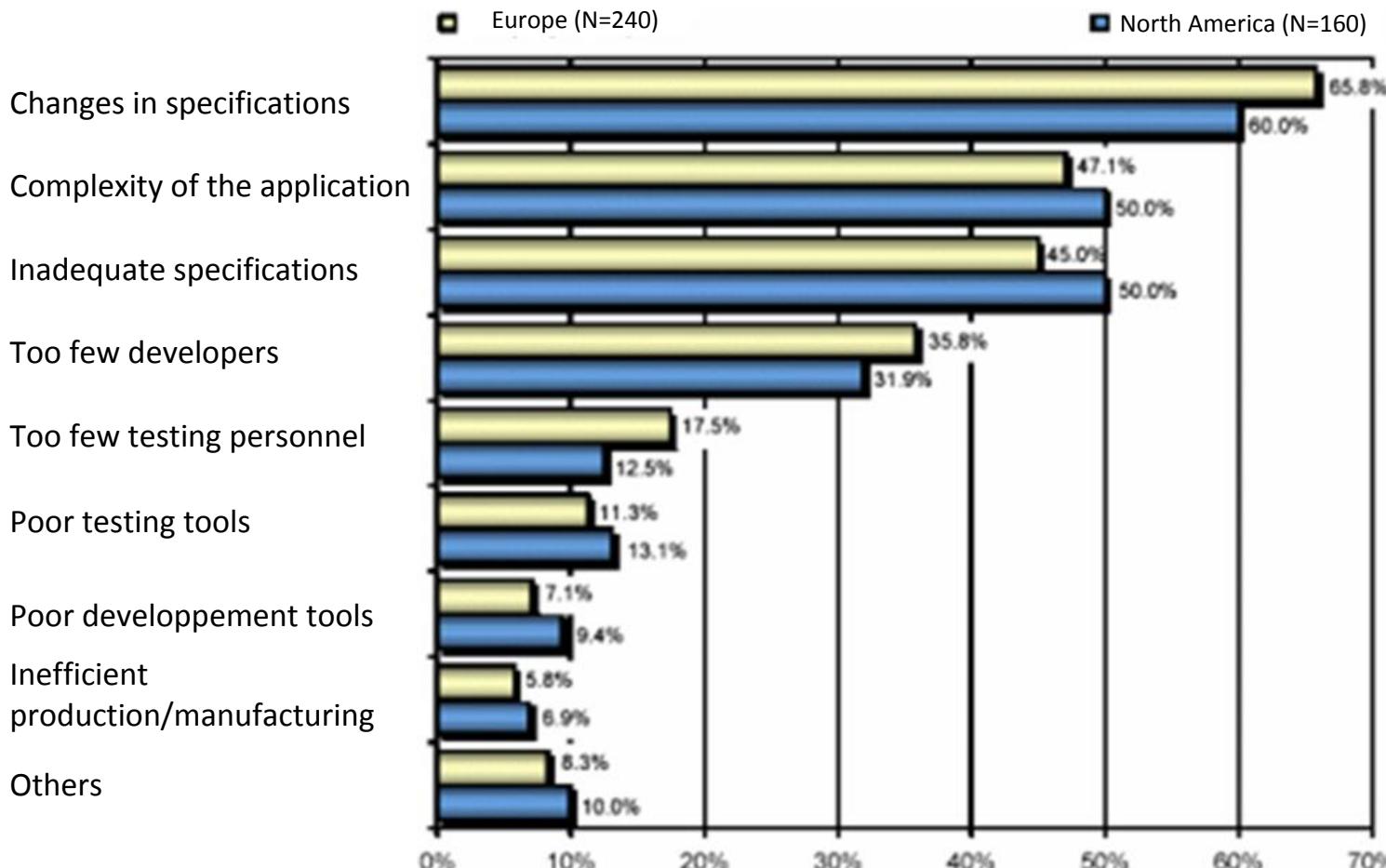


# 1. QUELLE DÉMARCHE?



# 1. QUELLE DÉMARCHE?

71.5% of traditionnal embedded system designs were not within 30% of pre-design performance expectations. [1]



# 1. QUELLE DÉMARCHE?

Recueil  
des besoins

Analyse

Conception  
générique

Partitionnement  
logiciel/matériel

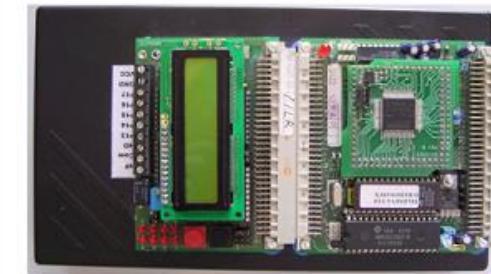
Test

# 1. QUELLE DÉMARCHE?

SOLUTION MATERIELLE  
CARTE A BASE D'EPLD/FPGA



SOLUTION LOGICIELLE  
CARTE A MICROPROSSEUR/MICROCONTROLEUR



SYNTHESE MATERIELLE

```
1 -- COMPONANT Gestion role
2 -- Version 0.95a
3 -- WARNING: Default parameter use = Synchronous version =
4 -- State translation :
5 --   S1 by vector "000"
6 --   S2 by vector "001"
7 --   S3 by vector "010"
8 --   S4 by vector "011"
9 --   S5 by vector "100"
10
11 library ieee;
12 use ieee.std_logic_1164.all;
13 use work.std_arith.all;
14
15 -- Component description
16 entity GestionRole is port(
17   reset : in std_logic;
18   NVR : in std_logic_vector(7 downto 0);
19   role : in out std_logic_vector(1 downto 0);
20   -- Call to function "election"
21   electionIn : in std_logic_bit;
22   electionOut : in std_logic_bit;
23 end GestionRole;
24
25 -- Component architecture
26 architecture arch_GestionRole of
27 begin
28   signal pastState:std_logic_vector(2 downto 0);
29   signal state:std_logic_vector(2 downto 0);
30   signal futureState:std_logic_vector(2 down to 0);
31   begin
32     exec:process(clk)
33     begin
34       if(clk'event and clk='1') then
35         if(reset='1') then
36           state<="000";
37         else
38           case state is
39             -- State S1
40             when "000" =>
41               if (NVR="00000001") then
42                 futureState = "001";
```

SYNTHESE LOGICIELLE

```
6 String futurState = "";
7
8
9 public void exec(){
10 //FIRST STATE ENTRY
11   for(:){
12     switch(presentState){
13       case(S1):
14         if(NVR == 1)
15           futurState = "S2";
16
17         if(NVR == 2)
18           futurState = "S3";
19
20         //ACTION
21         if (presentState != futureState)
22           //EXIT
23         else if (presentState != pastState)
24           //ENTRY
25         else
26           role = aucun;//STATE
27         break;
28       case(S2):
29         if(NVR==1)
30           futurState = "D";
31     }
32   }
33 }
```

## 2. QUELS SONT MES BESOINS?

---

Pour quel type d'application l'objet que je dois construire va me servir?

1. Calcul généraliste
2. Contrôle de systèmes physiques
3. Traitement du signal
4. Réseaux et communications

## 2. QUELS SONT MES BESOINS?

---

Quelles sont mes **contraintes**?

- Puissance de **calcul**
- Capacité de **communication**
- Capacité de **stockage**
- Consommation d'**énergie**
- Temps de réponse
- Tolérances aux **pannes**
- Durée de vie
- Testabilité/débogage
- Nombre d'unités produites

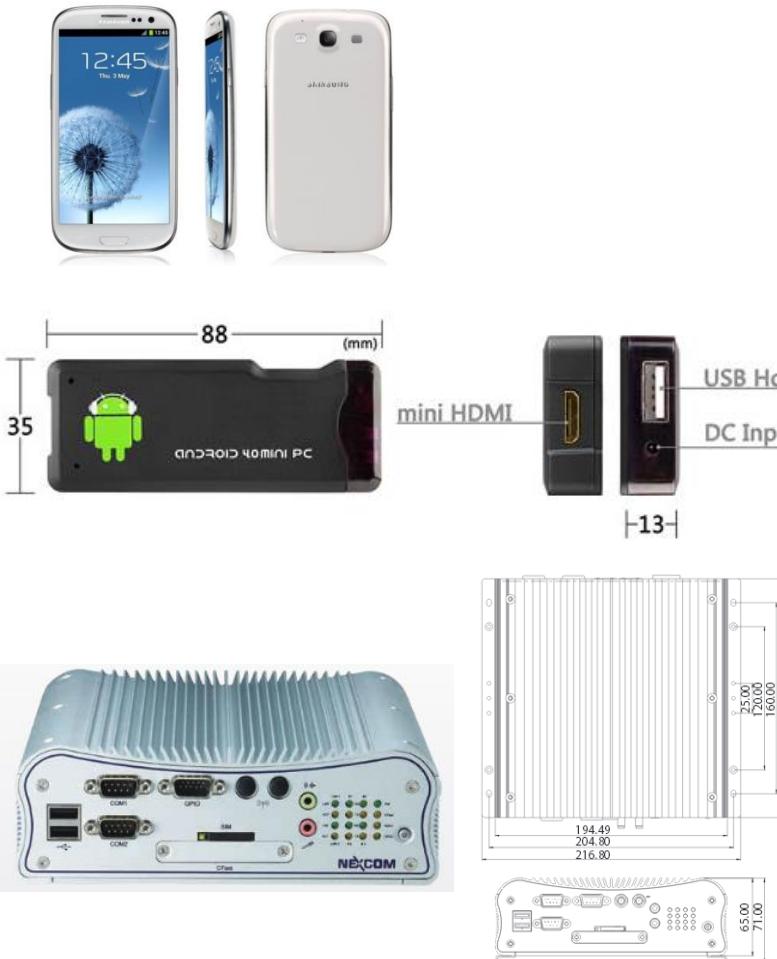
### 3. QUELS CHOIX POUR LA PARTIE MATÉRIELLE?

---

1. PDA / Mini-PC / PC Industriels
2. Une carte de développement **existante**
3. From scratch

### 3. QUELS CHOIX POUR LA PARTIE MATÉRIELLE?

- PDA / Mini-PC / PC Industriels



Modèle: Samsung Galaxy S4GT-I9505

OS: Android 4.2.2 Jelly Bean

CPU,GPU: 4 cœurs Qualcomm Snapdragon 600 (ARMv7) à 1,9 GHz, Qualcomm Adreno 320

Mémoire vive: 2 Go de RAM LPDDR3

Stockage: 32 Go de mémoire flash intégrée

Connectique: HDMI, prise casque jack 3,5 mm, port micro USB 2.0 compatible MHL 2.0, WiFi 802.11 a/b/g/n/ac (HT80), NFC, Bluetooth 4.0 (LE), LED IR3

Capteurs: de proximité, de lumière, IR, thermomètre, hygromètre, baromètre, accéléromètre et gyroscope à 3 axes, magnétomètre

### 3. QUELS CHOIX POUR LA PARTIE MATÉRIELLE?

- PDA / Mini-PC / PC Industriels



Modèle: Android Mini PC MK802

OS: Android 4.2.2 Jelly Bean ou Ubuntu ou PicUntu

CPU,GPU: Cortex-A9 à 1.6 GHz, 400 MHz Mali GPU

Mémoire vive: 2 Go de RAM LPDDR3

Stockage: 8 Go de mémoire flash intégrée

Connectique: HDMI, micro-USB 2.0, USB 2.0, microSD slot, alim. via micro-USB OTG, Wi-Fi 802.11 b/g/n, Bluetooth,

Capteurs:

### 3. QUELS CHOIX POUR LA PARTIE MATÉRIELLE?

- PDA / Mini-PC / PC Industriels



Modèle: PC industriel fanless NISE2200

OS: Linux, Windows ...

CPU,GPU: Intel® Atom™ Dual Core D2550 1.86GHz

Mémoire vive: 8Go DDR3 SODIMM

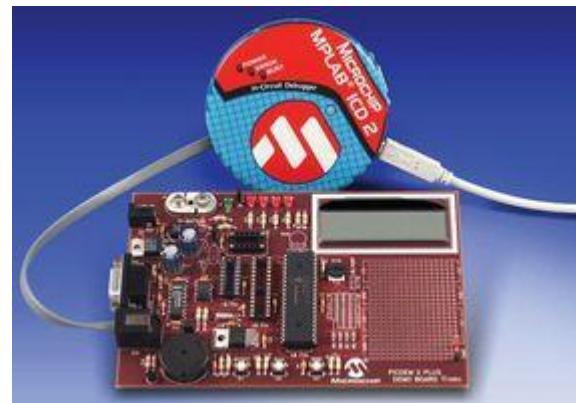
Stockage: Disque dur 2.5" SATA

Connectique: 6 ports USB2.0; 1 emplacement CFast; 1 emplacement carte SIM, 2 ports RS-232/422/485 isolés, 2 ports Ethernet Intel® 82574IT Gb, 1 port DB15 E/S , WiFi 802.11 a/b/g/n/ac or 3.5G (auto detected modules), Support 9-36V DV input, Audio Jack

Capteurs:

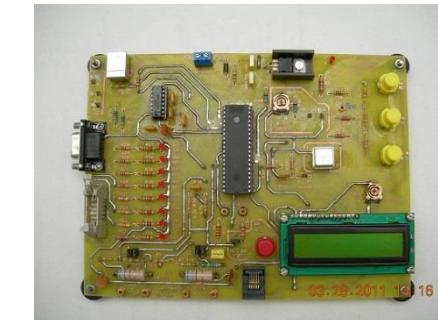
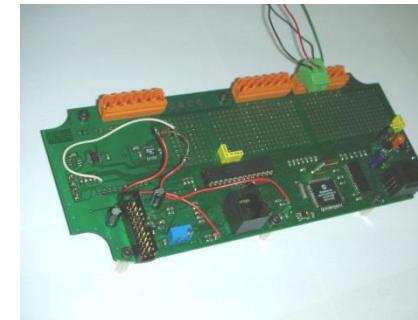
### 3. QUELS CHOIX POUR LA PARTIE MATÉRIELLE?

- Une carte de développement **existante**

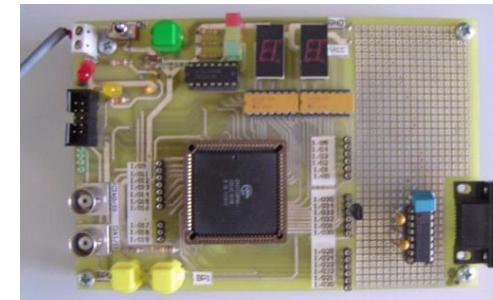


### 3. QUELS CHOIX POUR LA PARTIE MATÉRIELLE?

- From scratch
  - Architecture centrée **microcontrôleur**

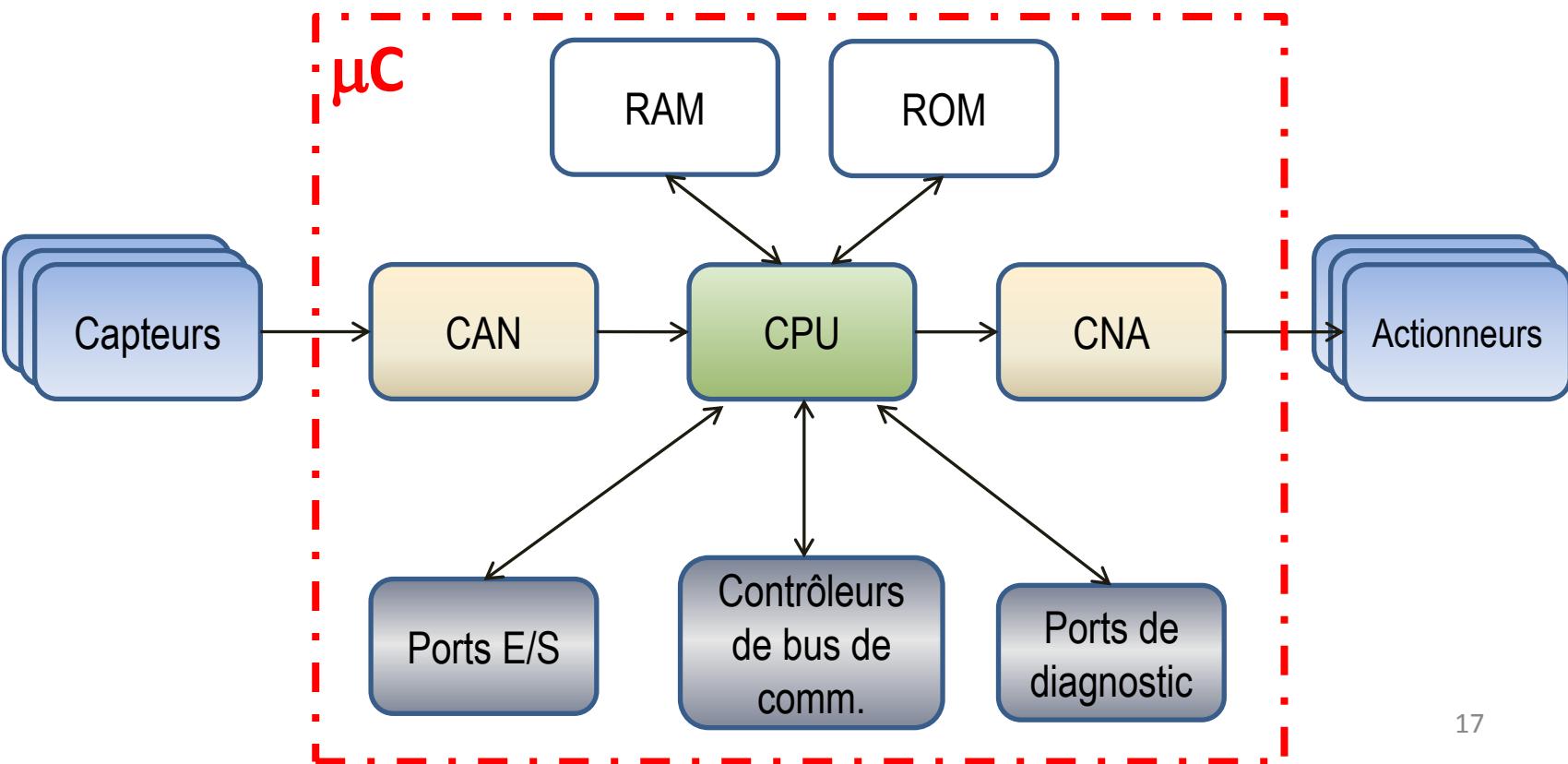


- Architecture centrée **FPGA/EPLD/ASIC?**



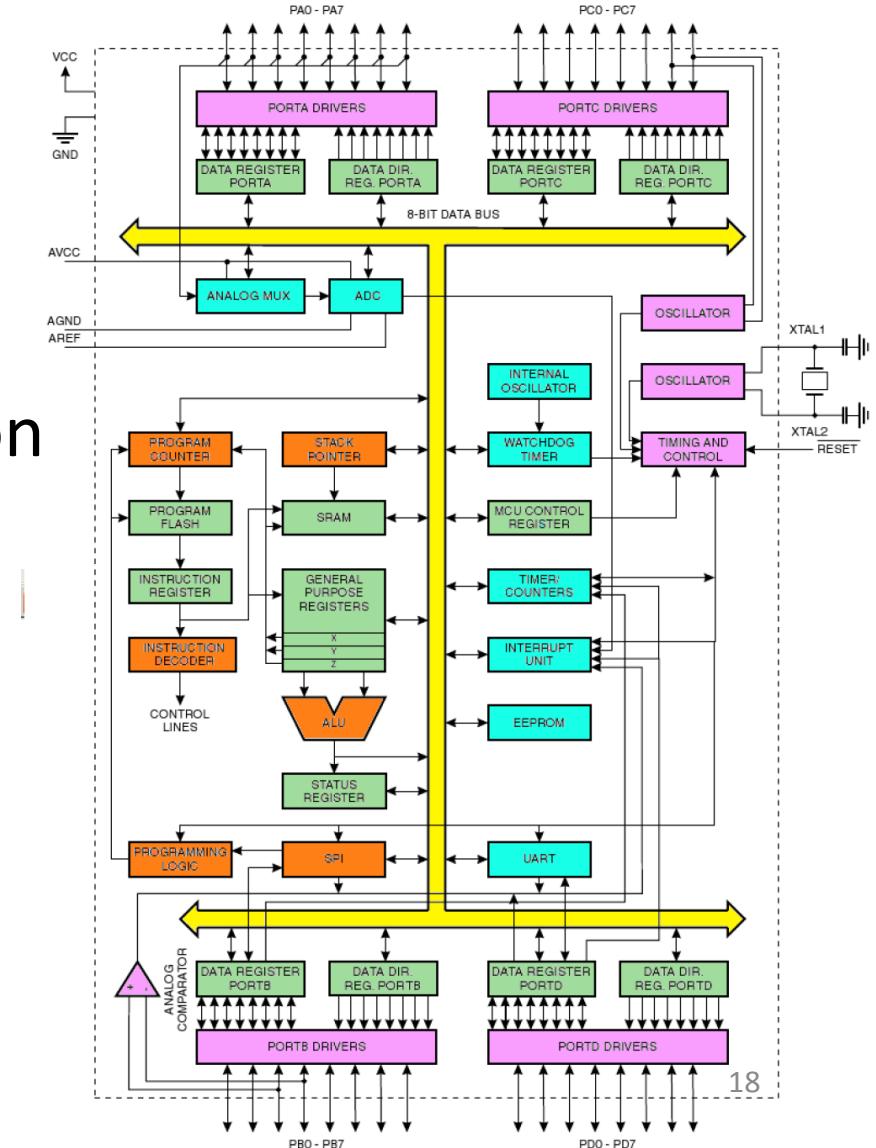
### 3. QUELS CHOIX POUR LA PARTIE MATÉRIELLE?

- Architecture d'objet communicant **basée sur un microprocesseur** ou un **microcontrôleur**



# 3. QUELS CHOIX POUR LA PARTIE MATÉRIELLE?

- Processeur
- Mémoire
- Périphériques
- Bus de communication
- Entrées/Sorties



# 3. QUELS CHOIX POUR LA PARTIE MATÉRIELLE?

- Utiliser les « Product Selector Guide »

Corporate Focus Product  
Selector Guide

# 3. QUELS CHOIX POUR LA PARTIE MATÉRIELLE?

Main - Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils ?

Main

Page précédente Page suivante www.microchip.com/maps/main.aspx Actualiser Arrêter

Les plus visités À la une Conférences acceptées http://ade52-upmf.gr... Zimbra: Réception (18)

microchip selector Téléchargements Accueil Firebug

MAPS Microchip Advanced Part Selector MICROCHIP

### About Microchip Advanced Part Selector

**Microchip Advanced Part Selector**  
MAPS

**Copyright © Microchip Technology Incorporated**

Microchip Technology Inc.  
2355 West Chandler Blvd.  
Chandler, AZ 85224-1699  
Ph:480-792-7200 Fax:480-899-9210  
[www.microchip.com](http://www.microchip.com)

The information contained in this media regarding device specification or pricing and the like is intended through suggestion only and may be superseded by updates.

The content of the database is representative of various manufacturers' product specifications and contains the latest product specification information available to Microchip Technology Inc. at the time of MAPS last update. The product specifications in MAPS are subject to change without notice.

It is your responsibility to ensure that your product selection meets with your system specifications.

All trademarks mentioned herein are property of their respective companies.

Where would you like to start?

Analog Interface Memory Microcontroller Wireless

OK

Rechercher : jamont Suivant Précédent Tout surigner Respecter la casse

# 3. QUELS CHOIX POUR LA PARTIE MATÉRIELLE?

Microcontroller - Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils ?

Microcontroller

Page précédente Page suivante www.microchip.com/maps/microcontroller.aspx Actualiser Arrêter Téléchargements Accueil Firebug

Les plus visités À la une Conférences acceptées http://ade52-upmf.gr... Zimbra: Réception (18)

Parameter Search Match ALL (AND) Match ANY Expand

Family -All- 8-bit Product Status  
Prefix -All- 16-bit  
32-bit

P.Memory(Kbytes) From Thru Program Memory

P.Memory(KWord) -All- -All-

RAM(Bytes) -All- -All- Sys. Mgmt. Features

EEPROM(Bytes) -All- -All-

I/O Pins -All- -All-

Max CPU Speed -All- -All-

Int. OSC -All- -All-

Comparator -All- -All- Analog Peripherals

A/D Ch. -All- -All-

A/D Bits -All- -All-

D/A Ch. -All- -All-

UART Ch. -All- -All- Digital Comm. Features

SPI -All- -All-

i<sup>2</sup>C™ -All- -All-

CAN Ch. -All- -All- Connectivity

USB Ch. -All- -All-

Input Capture -All- -All- Capture/Compare PWM

PWM Ch. -All- -All-

MtrCtrl PWM Ch. -All- -All- Appl. Peripherals

SMPS PWM Ch. -All- -All-

Search Results 819 MCPH Parts found

PIC32MX795F512L PIC32MX575F512L PIC32MX675F512L PIC32MX775F512L DSPIC33EP512GM310 DSPIC33EP512GM710

Reset Search Sort Results by Memory Size

To add MCPH part to side-by-side Add dsPIC33EP512MU814 Go to side-by-side

dsPIC33EP512MU814 In Production

Specifications Dev Tools Technical Docs Budgetary Pricing

P.Memory (Kbytes) 512 Flash  
P.Memory (KWord) 170  
Self-Write Flash Yes  
RAM (Bytes) 52K  
EEPROM (Bytes) 0  
DMA Ram 4096  
Auxiliary/Boot Flash 24  
I/O Pins 122  
Max CPU Speed 140 MHz (70 MIPS)  
Internal OSC 7.37 MHz, 32.768 kHz  
Code Guard™ Security Basic  
BOR, None, POR, WDT, WUR, System Mgmt Features 15-DMA, nanoWatt [Low Sleep, Fast Wake, Pwr Modes]  
Analog Peripherals 3-Comparators, Bandgap - No, OpAmp, 2A/D, 32x12-bit @ 500ksps; 10V/A, 0x4-bit @ksps  
Digital Comm. Peripherals 4-UART, 4-SPI, 2-I2C, AC97, I2S, PPS  
Connectivity 1-Full Speed (Device, Host, OTG) USB 2.0 OTG, 2-CAN, None, LIN, IrDA  
Capture/Compare PWM 16-Output Comp. & Std. PWM, 16-bit PWM, 16-Input Capture  
Digital Timers 9x16-bit, 4x32-bit  
Application Peripherals 14-MtrCtrl. PWM, 14-PS\_PWM, 2-QEI, PMP, EBI-No  
Debug/Development JTAG-Boundary Scan, ICSP, ICD debug  
Features - Yes  
Package (Pins) LQFP, TQFP (144)  
Operating Voltage (3V-3.6V)  
Temperature Ranges (-40 to +125)  
Min PCB Area (mm<sup>2</sup>)  
Show Other Features

Analog  
Interface  
Memory  
Microcontroller  
Wireless

Not sure of complete part number?  
Global Part Search

Have internet access?  
Selection Tools Home

Have a suggestion?  
Send Email

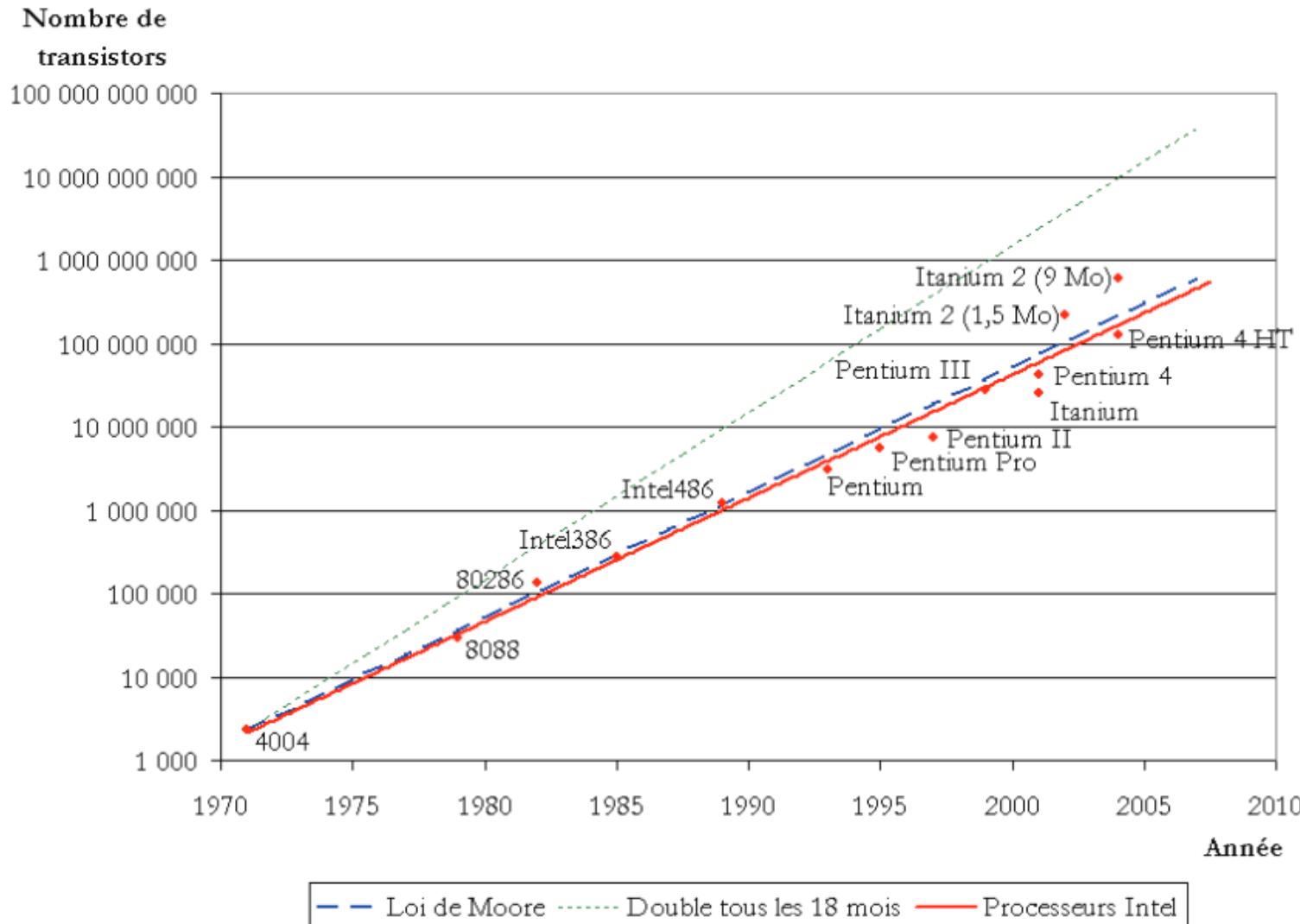
More Links  
Select a part in the search results, press one of the buttons below to view.  
dsPIC33EP512MU814

   
**MICROCHIP**

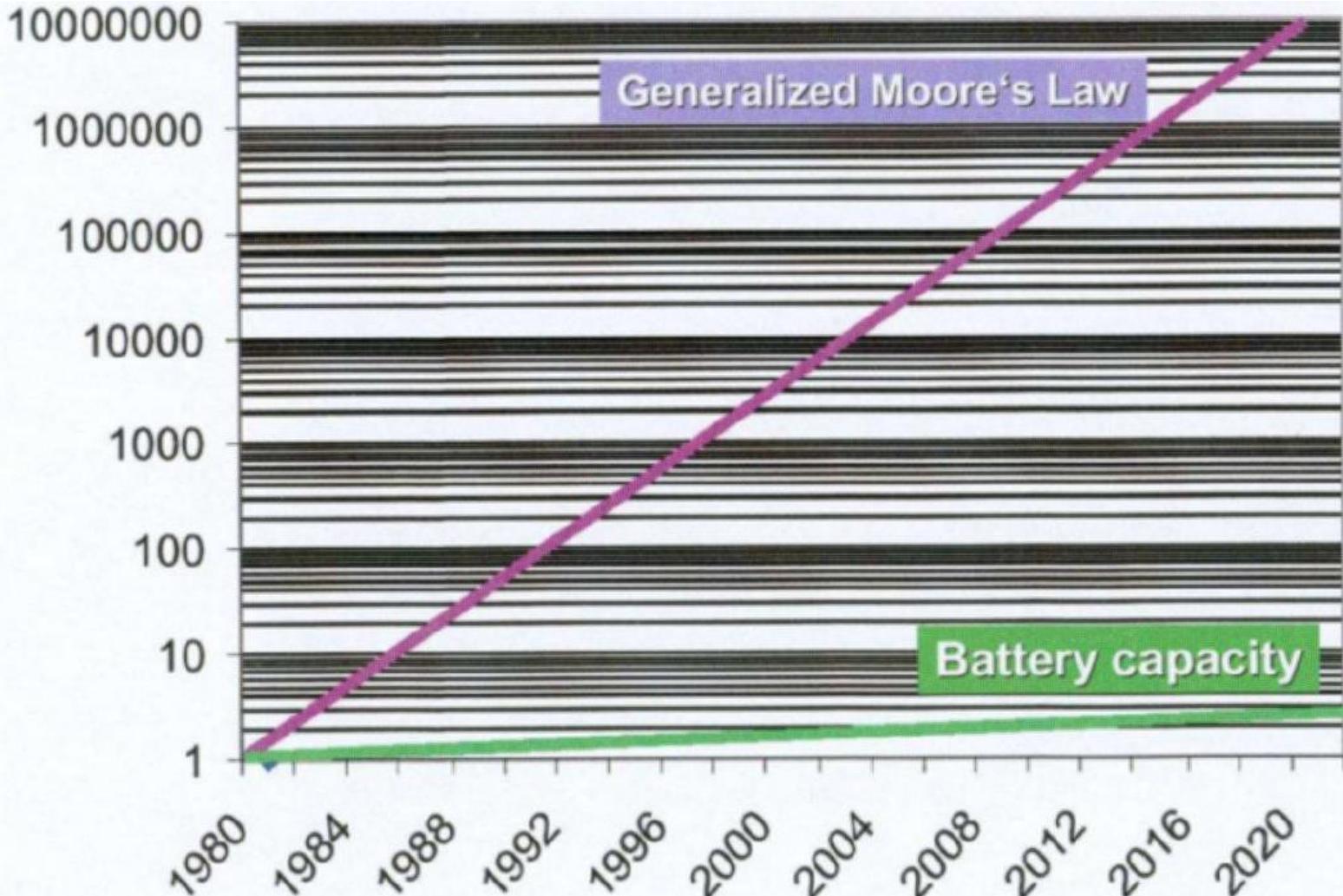
**microchip DIRECT**

Rechercher : jamont Suivant Précédent Tout surligner Respecter la casse

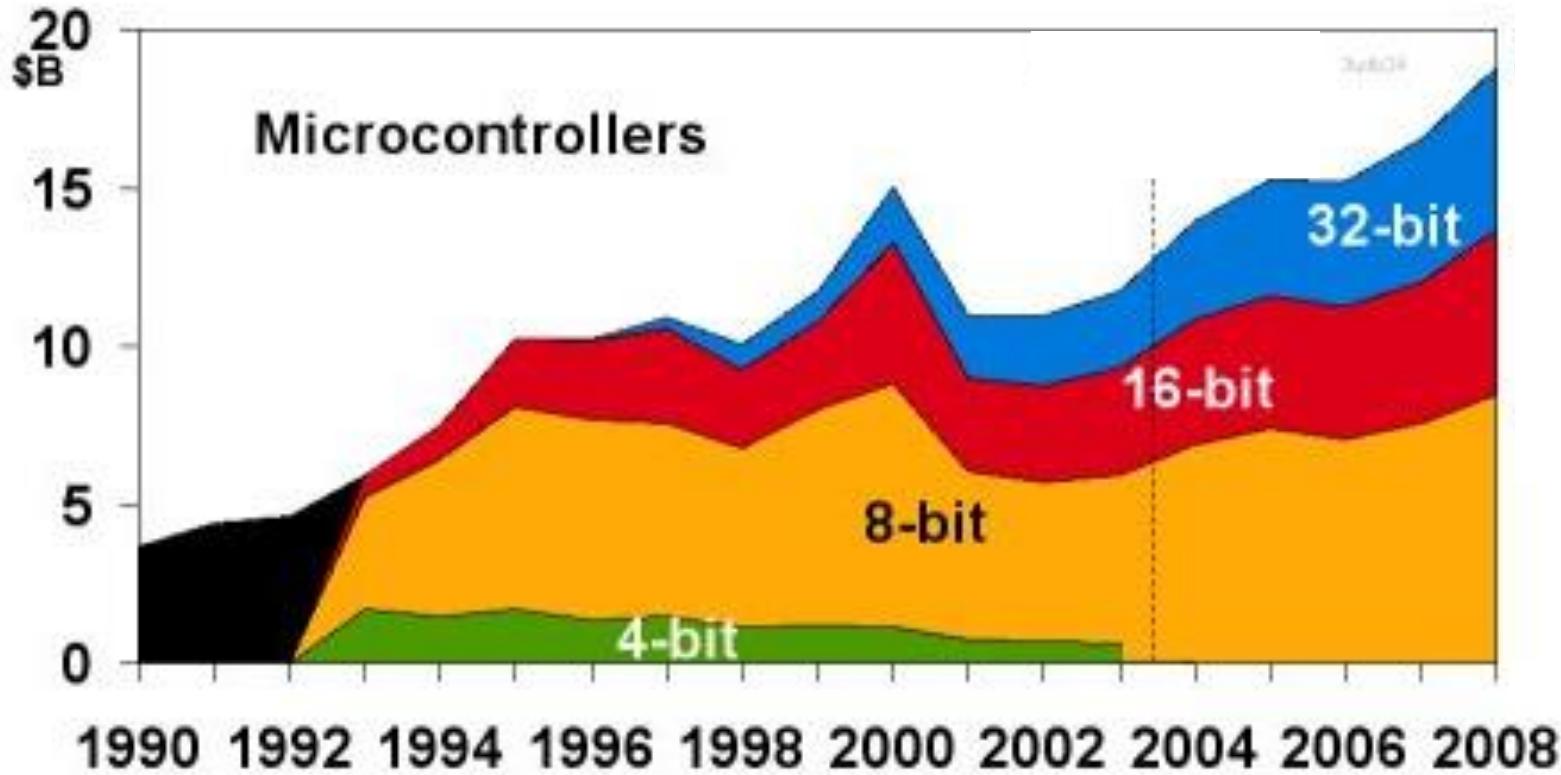
### 3. QUELS CHOIX POUR LA PARTIE MATÉRIELLE?



### 3. QUELS CHOIX POUR LA PARTIE MATÉRIELLE?

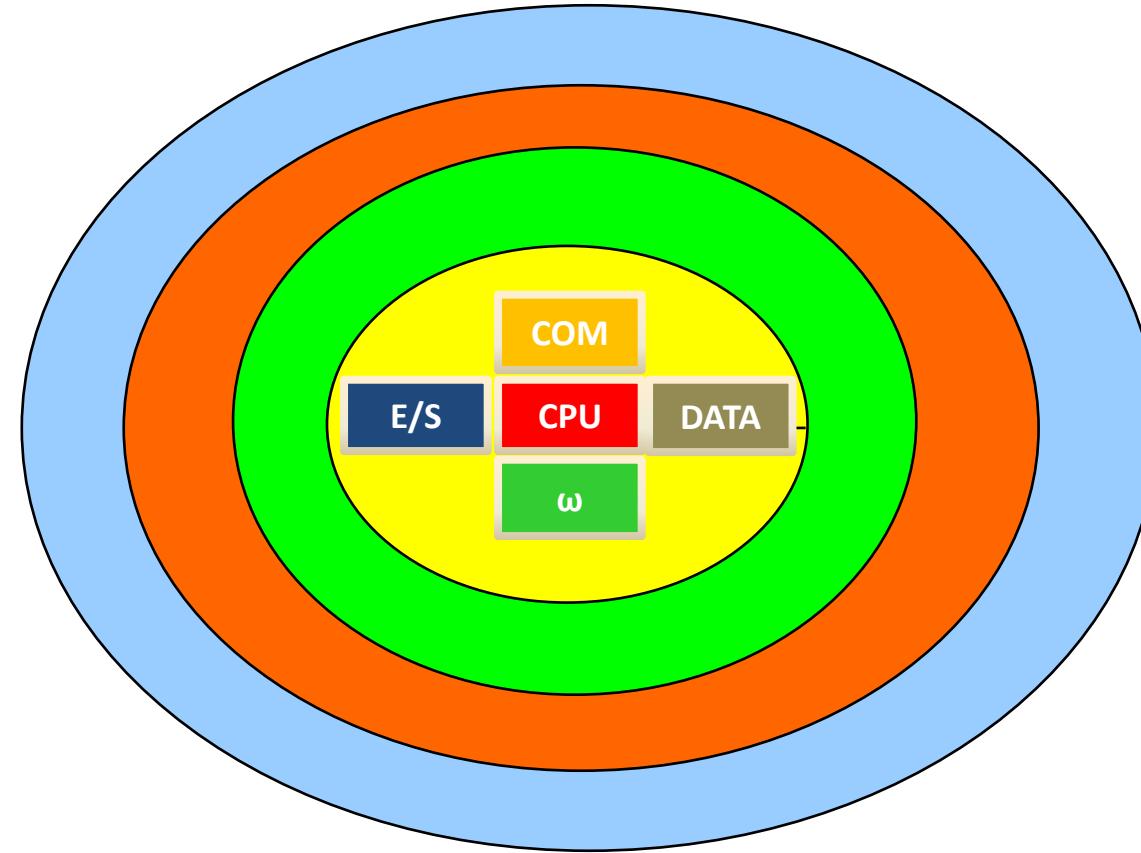


### 3. QUELS CHOIX POUR LA PARTIE MATÉRIELLE?



Source: Gartner Dataquest

# 3. VAIS-JE UTILISER UN OS?



# 3. VAIS-JE UTILISER UN OS?

Slicing the operating systems pie

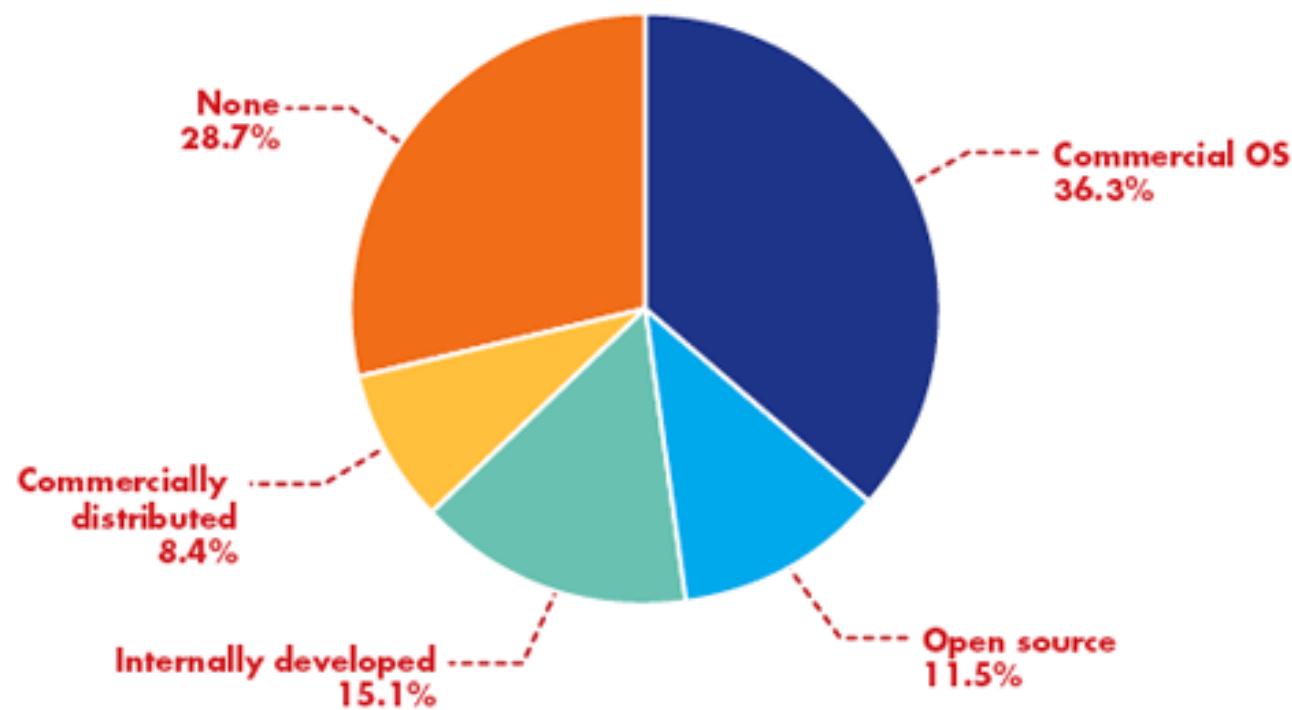


Figure 1

# 3. VAIS-JE UTILISER UN OS?

Quel type d'OS?

- **GPOS** (Normal General Purpose Operating System)  
=> **Interruptible**
- **RTOS** (Real Time Operating System)  
=> **Prédicabilité**

Critères	Temps partagé	Temps réel
<b>But</b>	Maximiser la capacité de traitement (débit) & utilisation des ressources	Etre prévisible (garantir les temps de réponse)
<b>Temps de réponse</b>	Bon en moyenne	Bon dans le pire des cas / moyenne non importante
<b>Comportement à la charge</b>	Confortable à l'utilisateur	Stabilité et respect des contraintes de temps

D'après J.Boukhobza , Systèmes d'exploitation embarqués

# 3. VAIS-JE UTILISER UN OS?

## Operating systems evaluation criteria

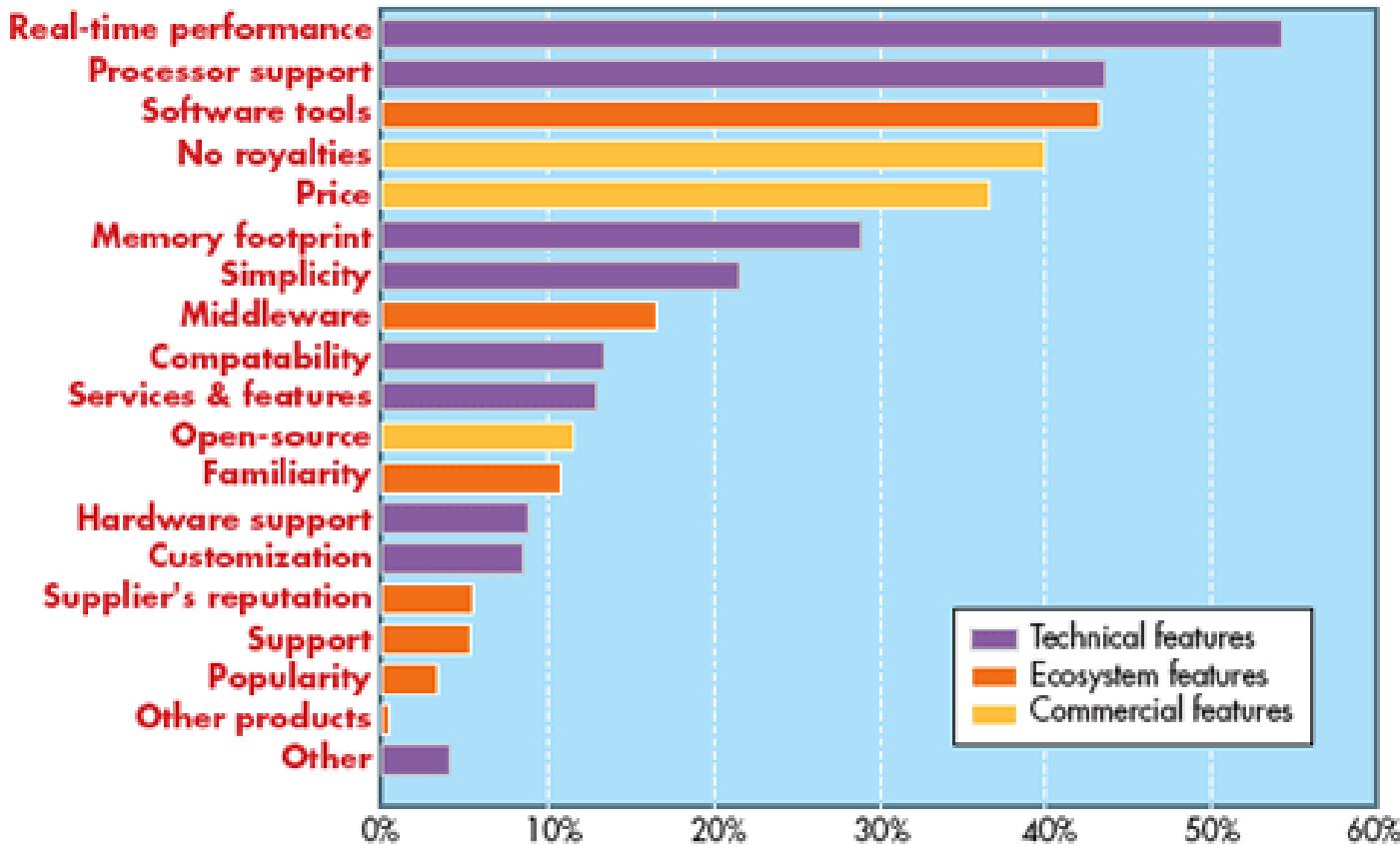


Figure 5

# 3. VAIS-JE UTILISER UN OS?

RTOS popularity now

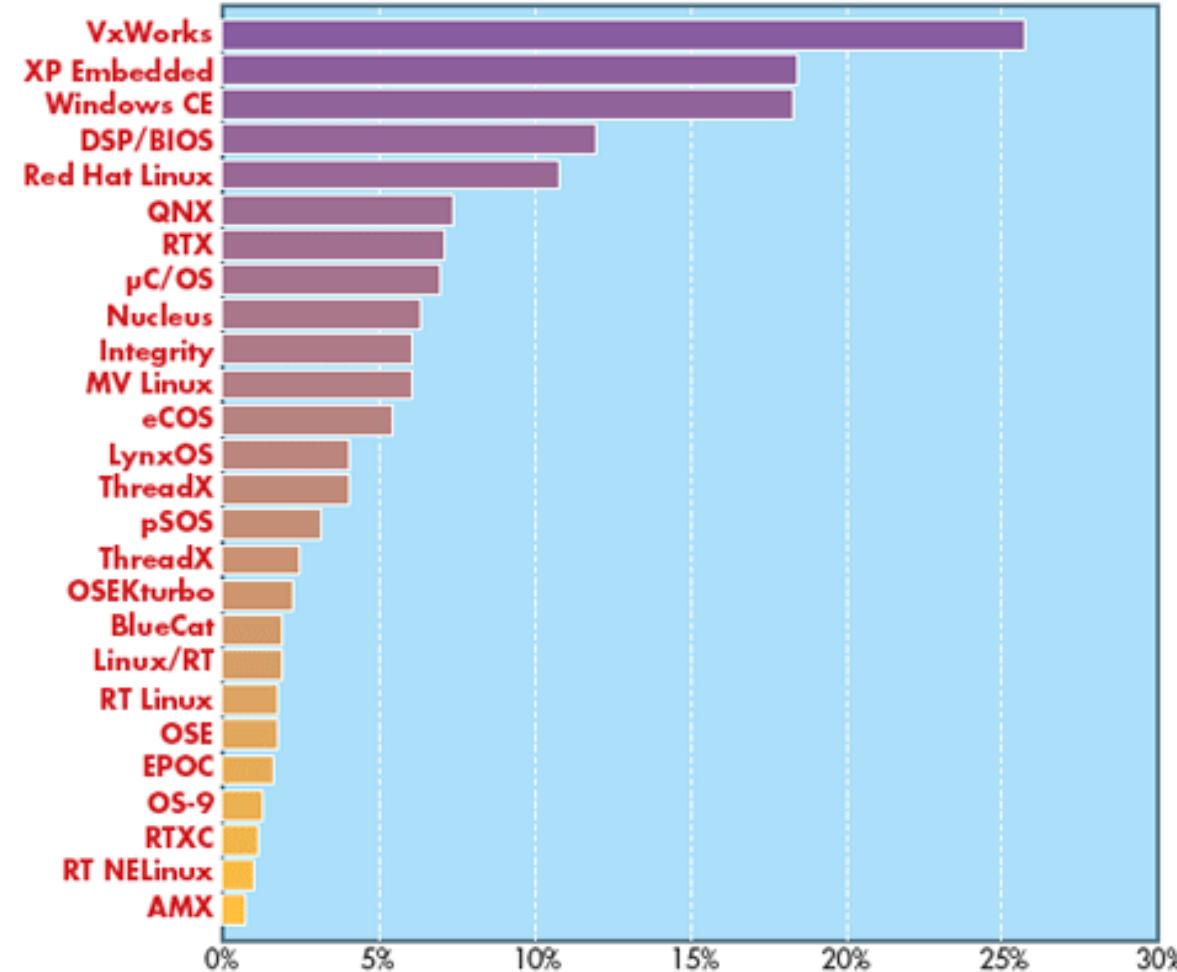


Figure 6

# 3. VAIS-JE UTILISER UN OS?

Pour créer, initialiser et activer une tâche avec VxWorks

```
int taskSpawn(  
{Task Name},  
{Task Priority} 0-255, related to scheduling},  
{Task Options - VX_FP_TASK, execute with floating point  
coprocessor  
VX_PRIVATE_ENV, execute task with private environment  
VX_UNBREAKABLE, disable breakpoints for task  
VX_NO_STACK_FILL, do not fill task stack with 0xEE}  
{Stack Size}  
{Task address of entry point of program in memory-initial PC  
value}  
{Up to 10 arguments for task program entry routine})
```

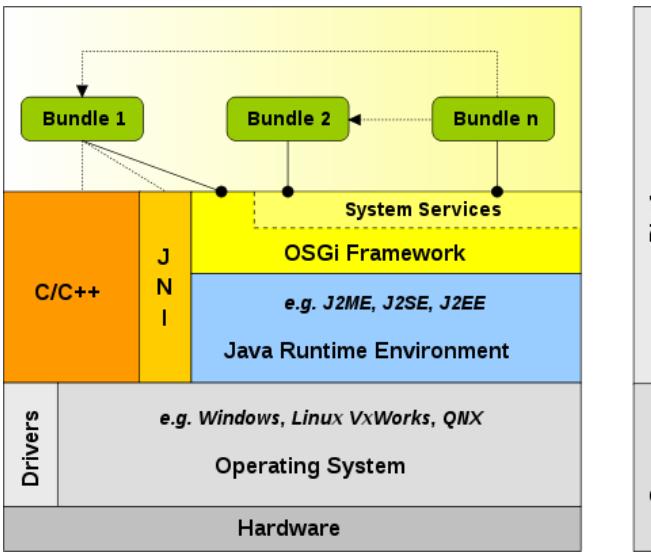
=> Après appel du taskSpawn, une image de la tâche est créée (Process Control Block , pile, programme)

# 3. VAIS-JE UTILISER UN OS?

```
// Tâche du parent qui active le l'horloge logicielle
void parentTask(void)
{
...
if sampleSoftware Clock NOT running {
newSWClkId = taskSpawn ("sampleSoftwareClock", 255,
VX_NO_STACK_FILL, 3000, (FUNCPTR) minuteClock, 0, 0, 0, 0, 0, 0,
0, 0, 0);
...
}
// Tâche executée par le programme fils
void minuteClock (void) {
integer seconds;
while (softwareClock is RUNNING) {
seconds = 0;
while (seconds < 60)
seconds = seconds+1;
}
.....
}
```

# 3. VAIS-JE UTILISER UN OS?

- Un **framework** au-dessus de l'OS?



Bundle = applications et/ou composants déployés

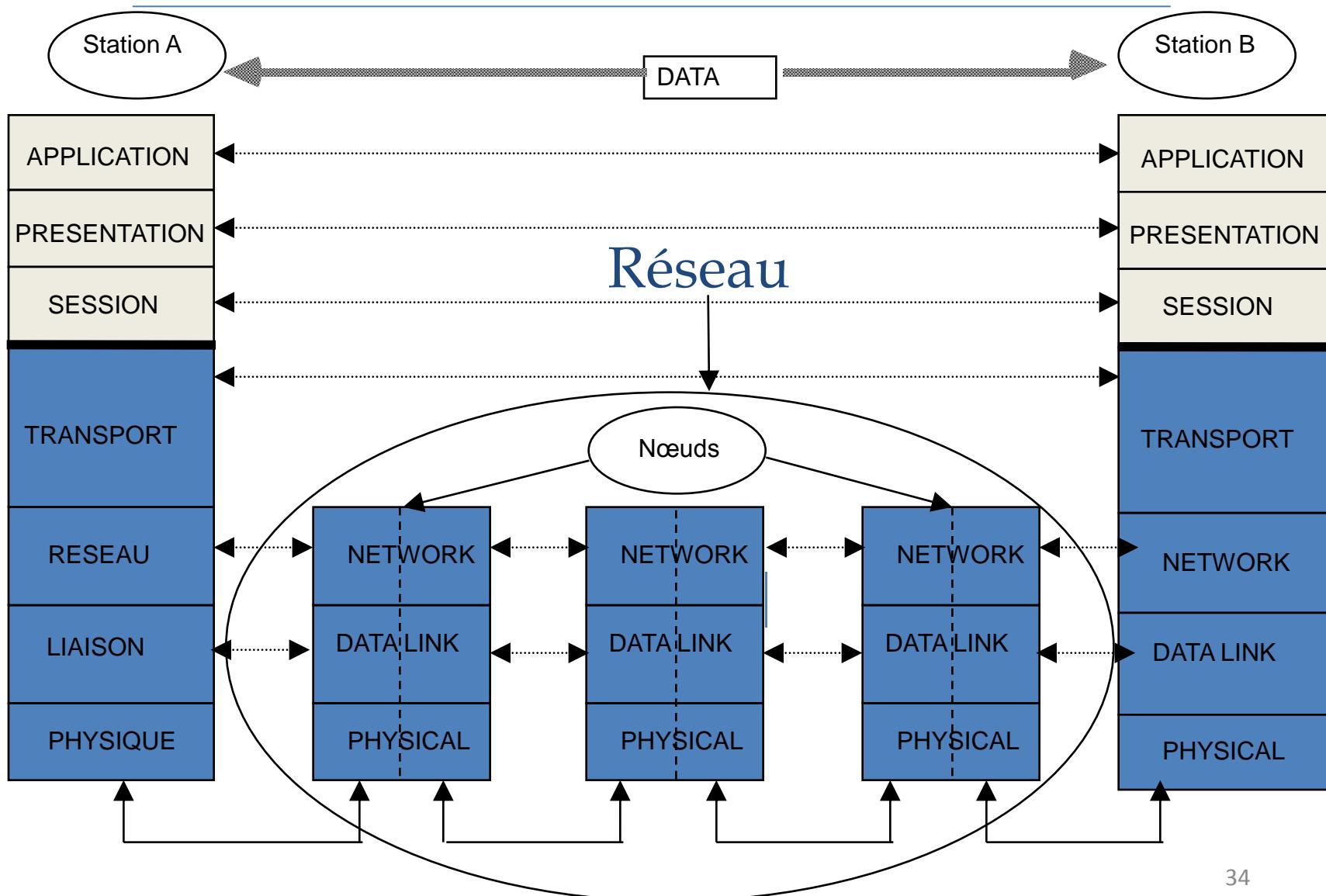
Services fournis:

- journalisation ,
- gestion des configurations ,
- le service HTTP (exec. servlets),
- l'analyse syntaxique XML, l'accès aux dispositifs (Device Access),
- l'administration de paquetage (Package Admin),
- l'administration des permissions (Permission Admin),
- le niveau de démarrage (Start Level),
- la gestion des utilisateurs (User Admin),
- le connecteur d'ES (IO Connector; IO = Input Output = Entrées Sorties),
- la gestion des connexions (Wire Admin),
- Jini, l'exportateur UPnP (UPnP Exporter),
- le pistage applicatif (Application Tracking),
- les paquets signés (Signed Bundles),
- les services déclaratifs (Declarative Services),
- la gestion de l'énergie (Power Management),
- la gestion des dispositifs (Device Management),
- les politiques de sécurité (Security Policies),
- diagnostic/contrôle et organisation en couches du cadre (Diagnostic/Monitoring and Framework Layering).

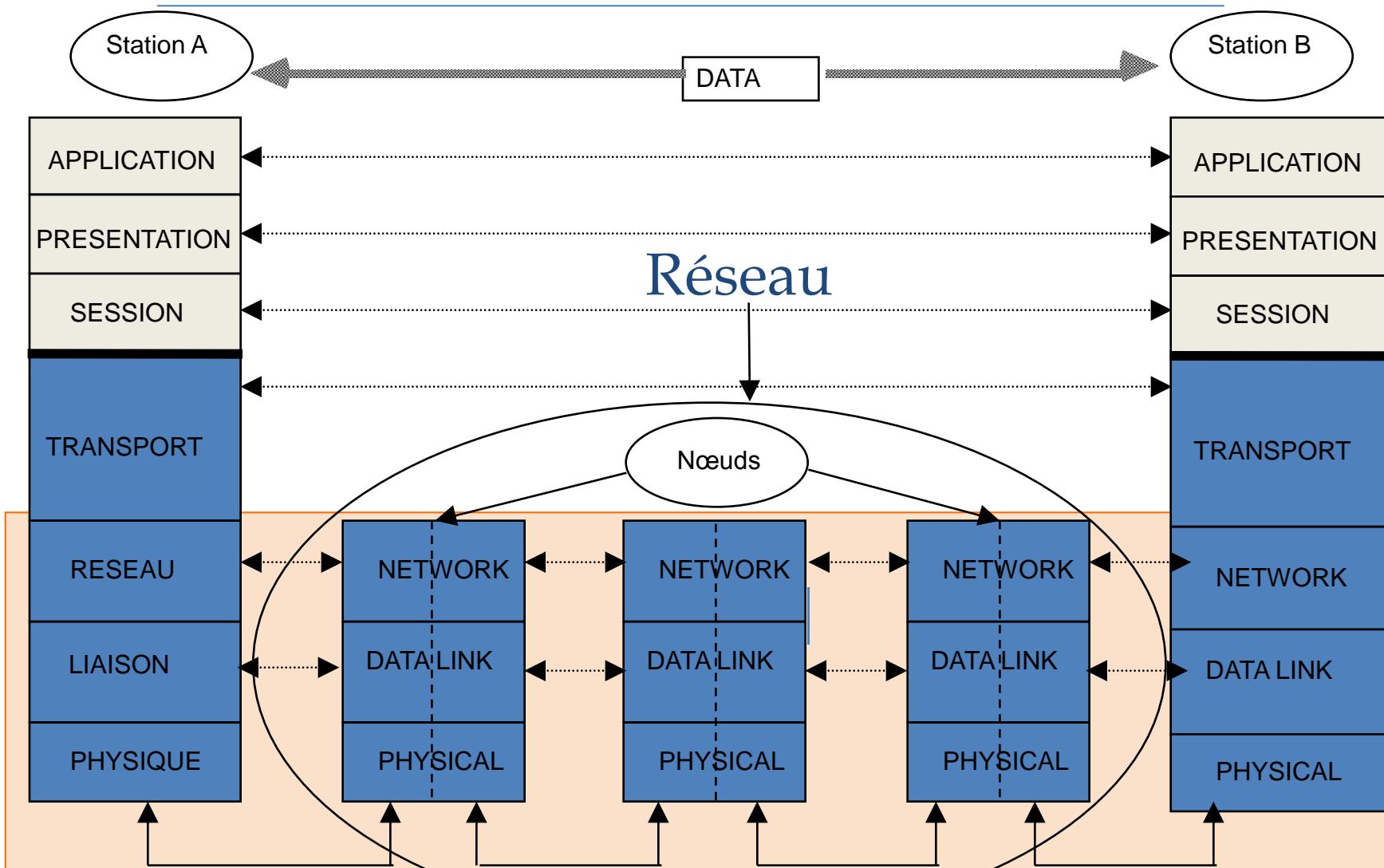
# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

- Les réseaux sans fil sont des **briques de base du Web Intelligence et de l'Internet of Things**
  - ⇒ Il faut savoir identifier les critères pertinents qui vont conditionner le choix de ces briques
  - ⇒ Il faut comprendre la caractérisation de ces briques
- Il est indispensable de comprendre les couches basses d'un **système communicant**
- Les couches basses impactent profondément la **qualité de service** d'un système communicant! En effet, il existe un **fort couplage** entre les technologies utilisées pour les couches basses et :
  - La consommation d'énergie,
  - Le déterminisme (ou non) temporel,
  - Le débit (couplé avec l'environnement du système)
  - La précision d'une radiolocalisation
  - Le taux de perte
  - Le déséquencement
  - La latence
  - La gigue
  - ...

# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?



# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?



Couches les plus impactées par le caractère sans fil du réseau

# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Bluetooth

Zigbee

Divers

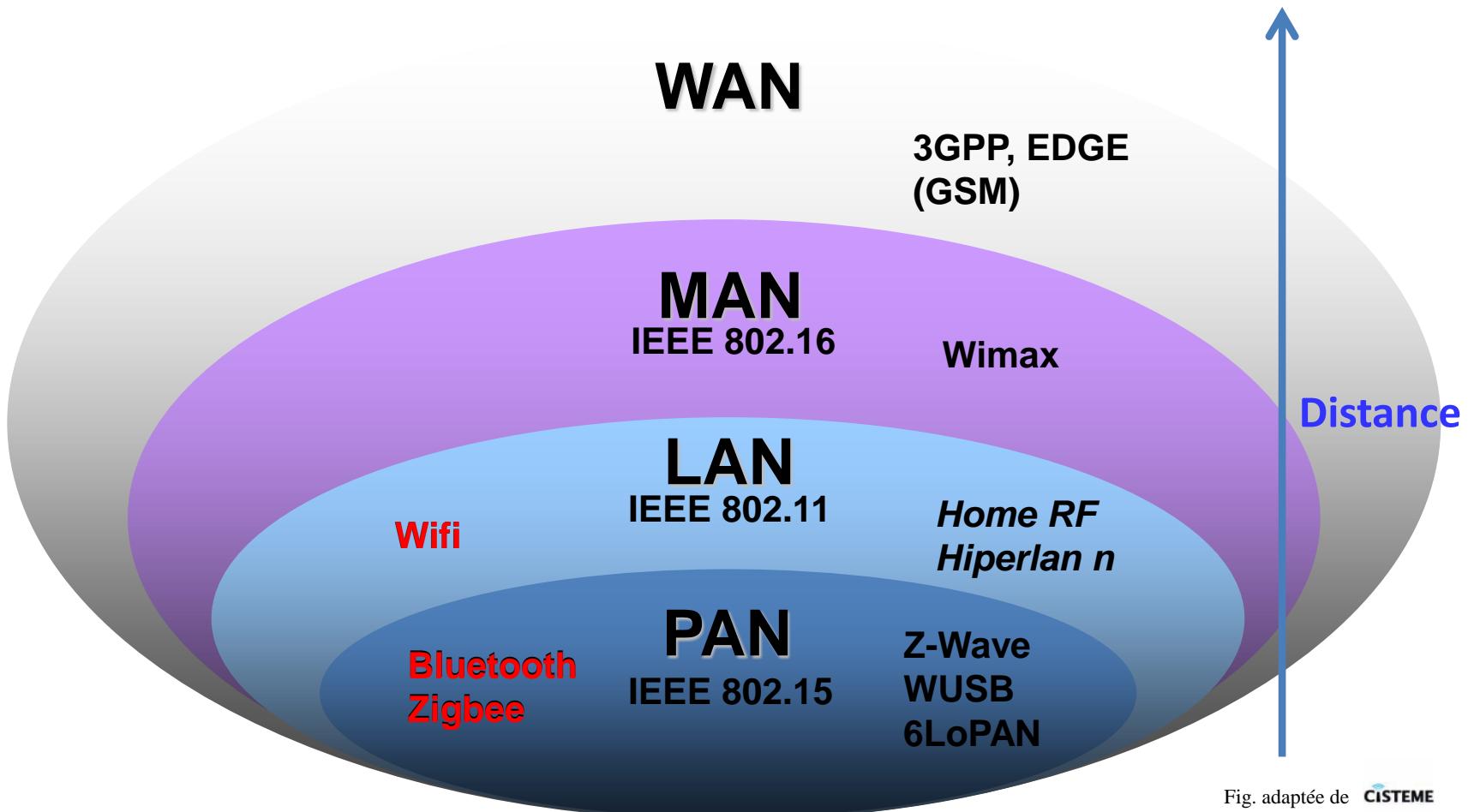


Fig. adaptée de **CISTEME**

# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Bluetooth

Zigbee

Divers

Nom	Description
802.1	Interface de haut niveau
802.2	LLC (Logical Link Control)
802.3	Ethernet
802.4	Token Bus
802.5	Token Ring
802.6	Réseau Métropolitain DQDB (Distributed Queue Dual Bus)
802.7	Réseau large bande
802.8	Fibre optique
802.9	Réseau à intégration voix et données
802.10	Sécurité des réseaux
802.11	Réseau Local sans fil
802.12	100VG AnyLAN
802.14	Réseau sur câble télévision CATV
802.15	Réseau local personnel
802.16	Réseau métropolitain sans fil BWA (Broadband Wireless Access)
802.17	RPR (Resilient Packet Ring)
802.18	Réglementations radio pour 802.11, 802.15, 802.16, 802.20
802.19	Coexistence des réseaux
802.20	Réseau large bande mobile MBWA (Mobile Broadband Wireless Access)
802.21	Gestion des handovers verticaux entre les standards sans fil du comité 802

# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

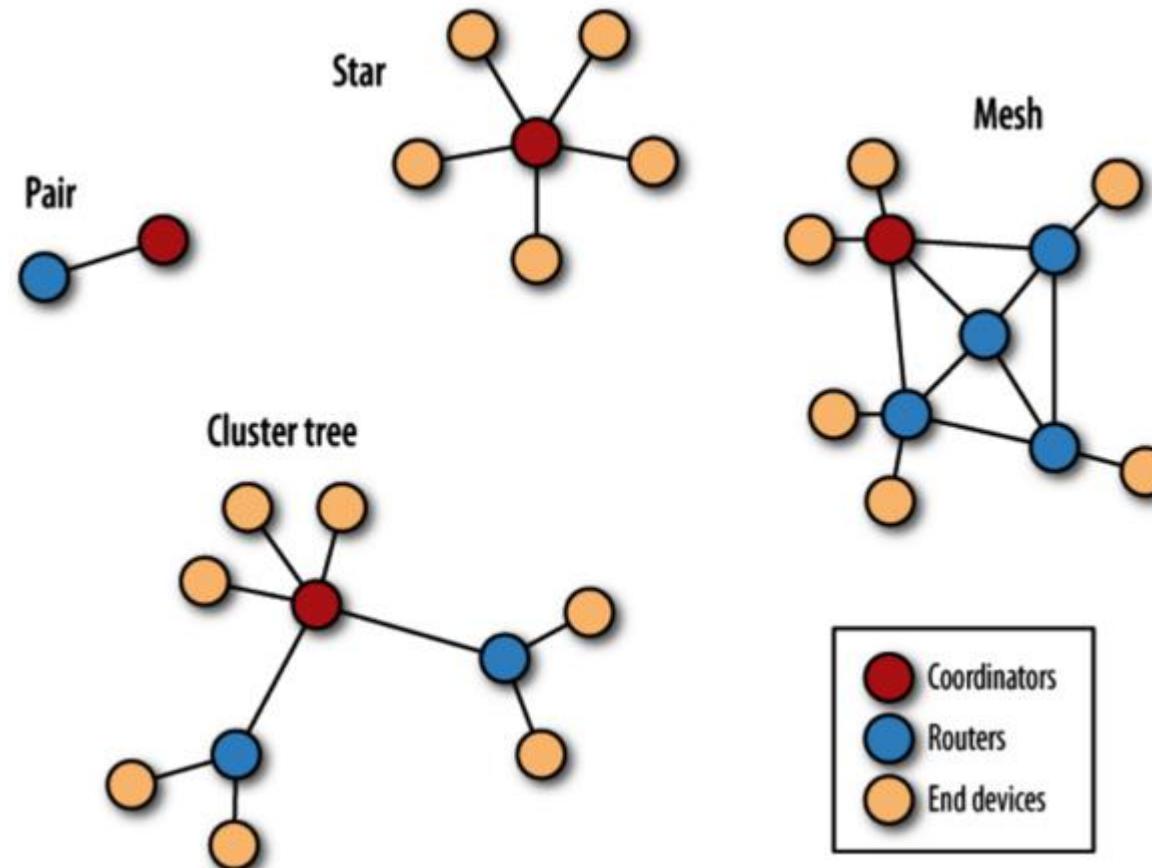
Avant-propos

Wifi

Bluetooth

Zigbee

Divers



# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Bluetooth

Zigbee

Divers

## ---Les bandes de fréquences---

- Bandes sans licence :

- ISM (2,4GHz) = Industrial, Scientific and Medical

- En France :

Intérieur	2,400- 2,4835 GHz	100mW
Extérieur	2,400 - 2,454 GHz	100mW
	2,454 - 2,4835	10mW

- U-NII (5GHz) = Unlicensed-National Information Infrastructure

- En Europe :

Intérieur	5,15 - 5,25 GHz	40mW
	5,25- 5,35 GHz	200mW
Extérieur		interdit

# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

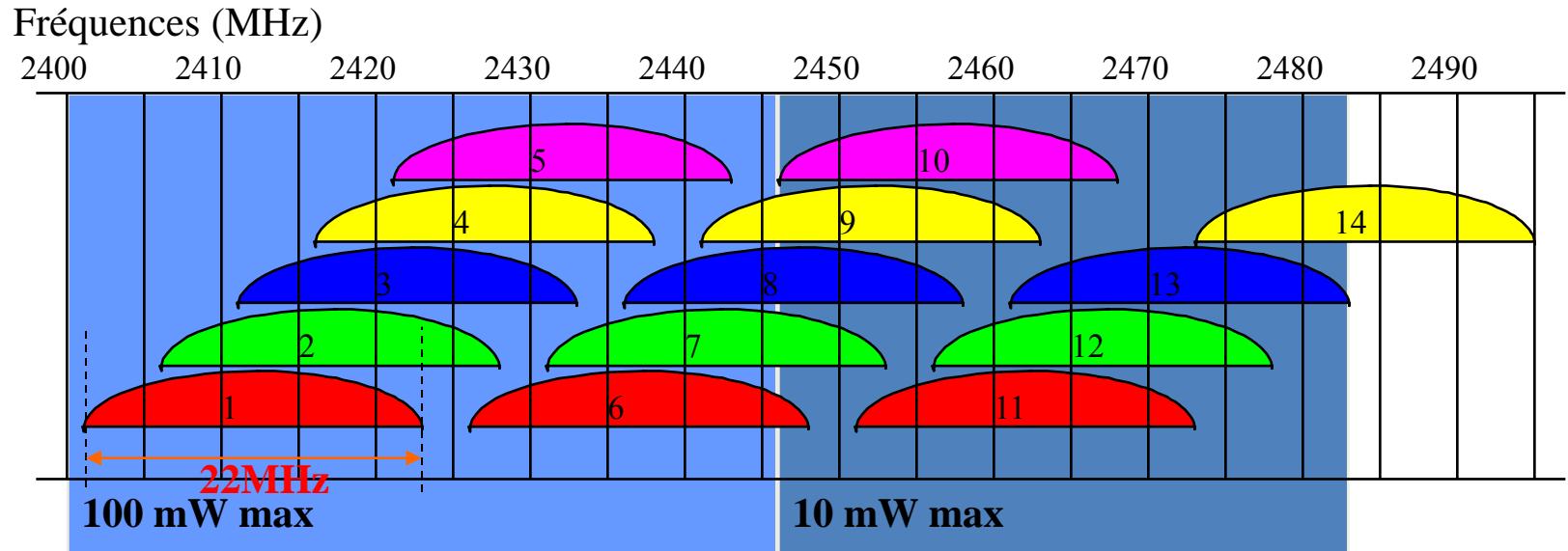
Wifi

Bluetooth

Zigbee

Divers

---Bande 2,4GHz (802.11b, 802.11g)---



- 13 canaux autorisés en France sur 14
- Dans la bande de fréquence permettant 100 mW de puissance d'émission (en extérieur) : Pas de canaux sans recouvrement
- Sur l'ensemble de la bande de fréquence : Maximum de 3 canaux sans recouvrement

# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

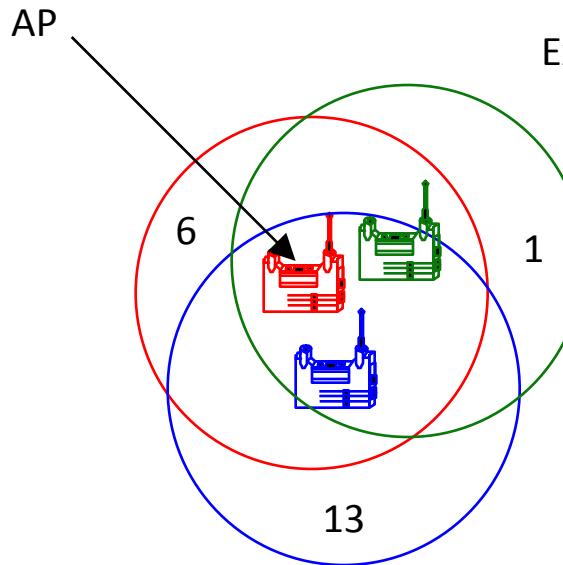
Wifi

Bluetooth

Zigbee

Divers

---Intérêt des canaux disjoints---

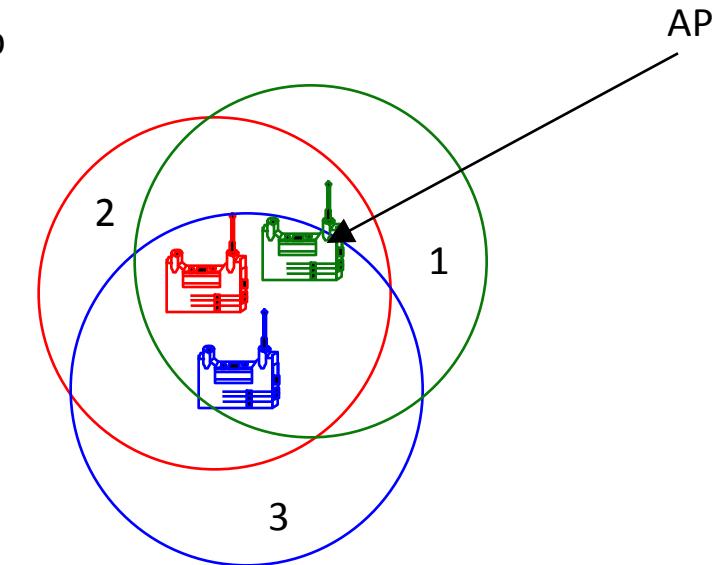


Canaux disjoints :

$$\text{Bande passante totale} = \sum \text{bandes passantes}$$

$$\text{Exemple : } 3 \times 11 = 33\text{Mb/s}$$

Exemple 802.11b



Canaux avec recouvrement :

$$\text{Bande passante totale} = \text{bande passante 1 cellule}$$

$$\text{Exemple 11Mb/s}$$

# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

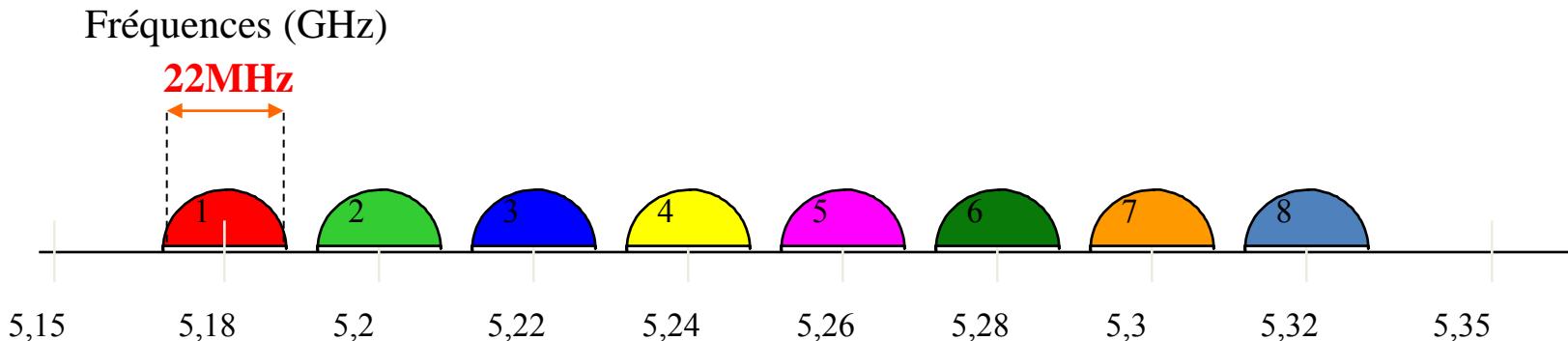
Wifi

Bluetooth

Zigbee

Divers

---Bande 5GHz (802.11a)---



- Tous les canaux sont disjoints !

# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

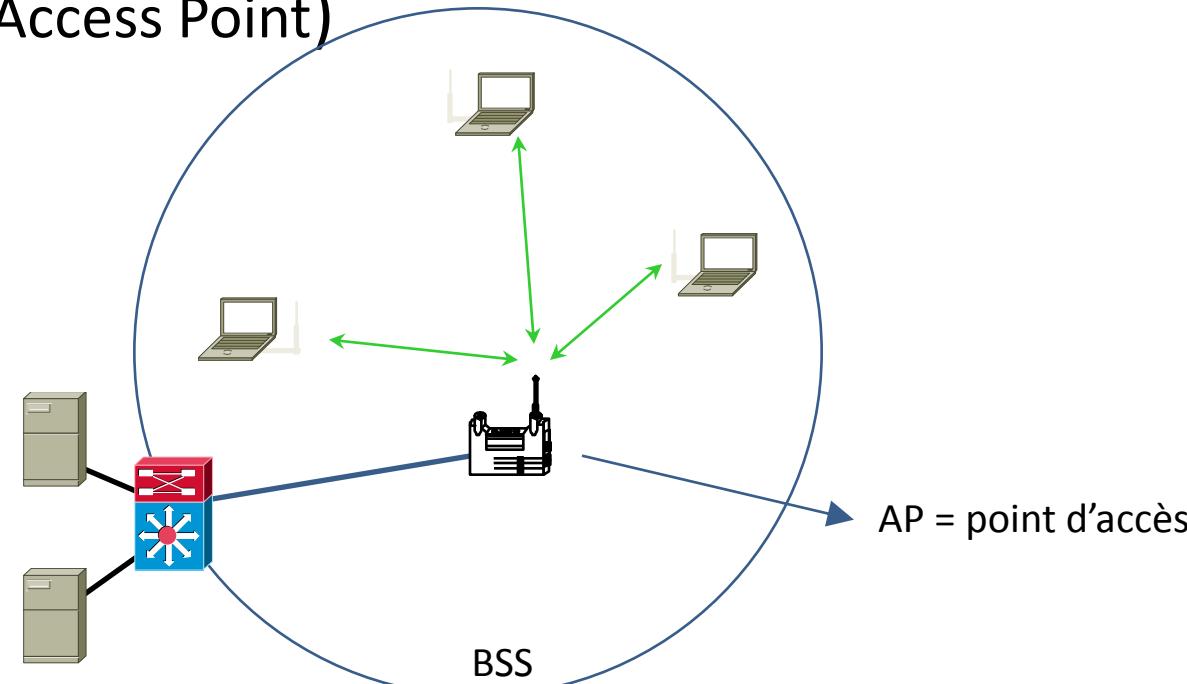
Bluetooth

Zigbee

Divers

## ---Topologie---

- Mode infrastructure :
  - BSS = Basic Set Service = 1 cellule = 1 point d'accès (AP = Access Point)



# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

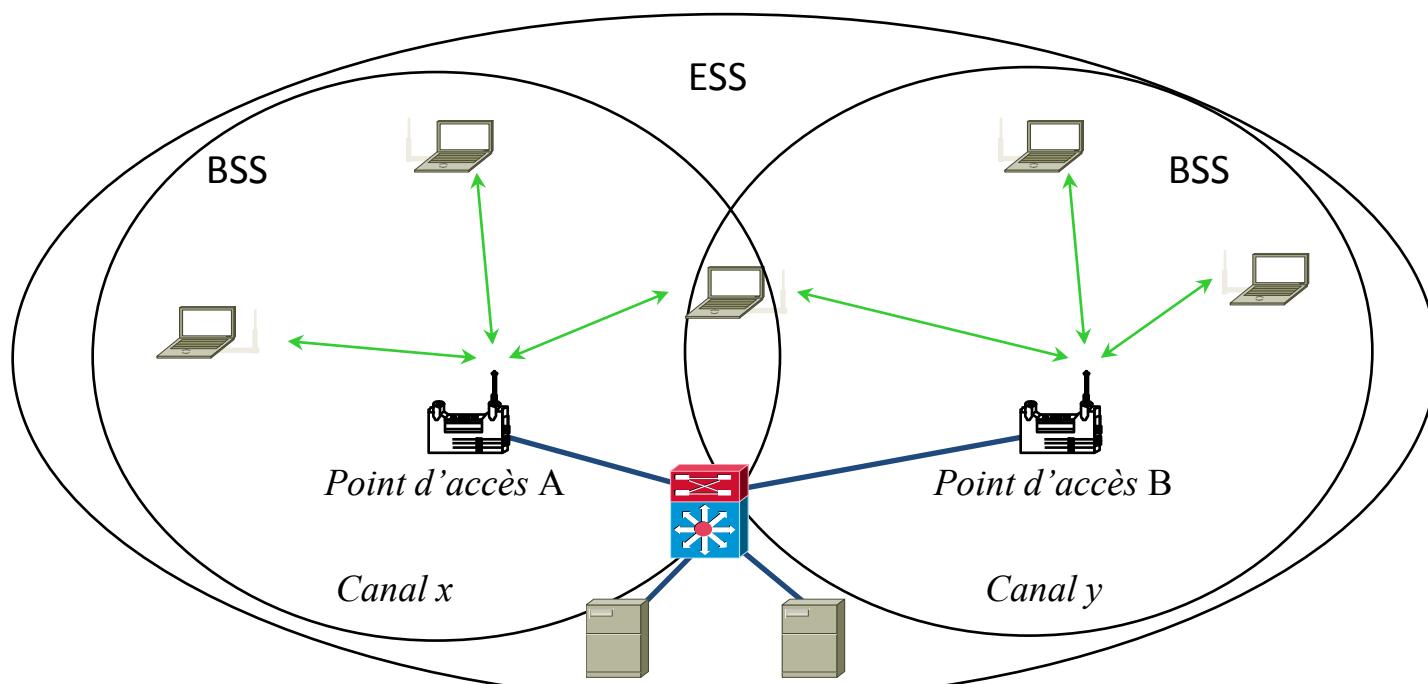
Bluetooth

Zigbee

Divers

## ---Topologie---

- Mode infrastructure :
  - ESS = Extended Service Set = plusieurs cellules = plusieurs AP



➔ Canal x et y disjoint pour de meilleures performances !

# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

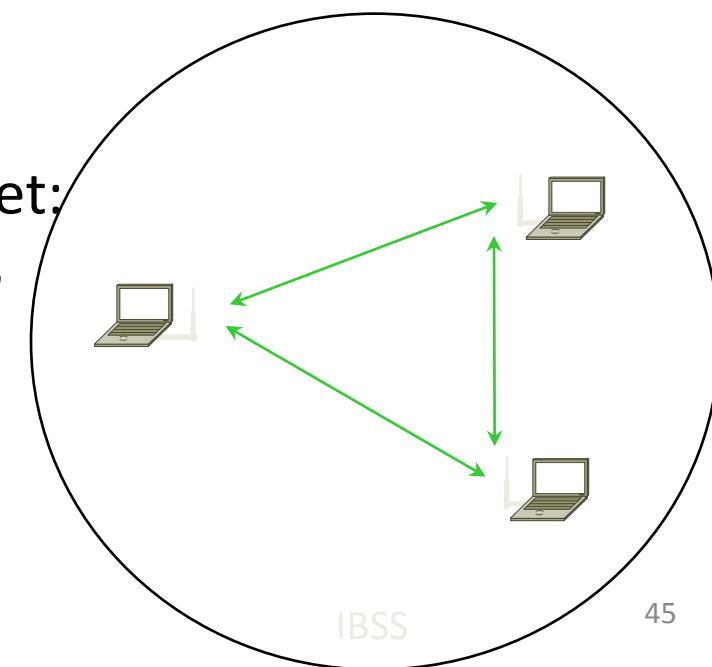
Bluetooth

Zigbee

Divers

## ---Topologie---

- Mode ad-hoc (peer to peer) :
  - IBSS = Independent Basic Set Service
  - Pas d' AP
  - Simplicité de configuration
  - Si une station connectée à Internet:
    - Partage de connexion possible ↔ BSS



# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

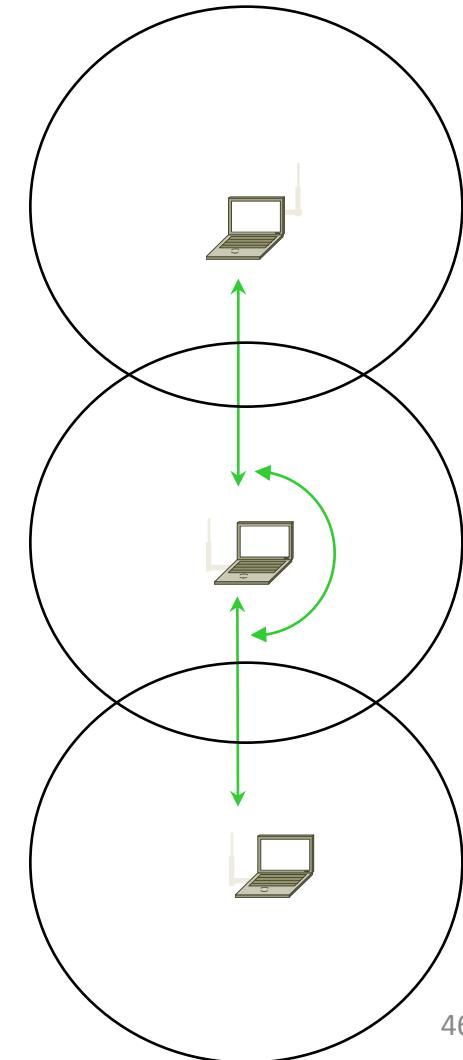
Bluetooth

Zigbee

Divers

## ---Topologie---

- Attention : mode ad-hoc et réseau ad-hoc :
  - En mode ad-hoc, les stations ne font pas transiter les données pour les autres
  - Un réseau ad-hoc le permet, mais protocoles non définis dans Wi-Fi !
    - AODV (Ad-hoc On Demand Distance Vector)
    - OLSR (Optimized Link State Routing)



# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Bluetooth

Zigbee

Divers

## ---Amendements 802.11---

Amendement	Description
802.11a	Couche physique : 54Mb/s bande 5GHz
802.11b	Couche physique : 11Mb/s bande 2,4GHz
802.11c	Incorporation des fonctionnalités de 802.11d
802.11d	Utilisation de 802.11 dans de nouveaux pays .
802.11e	Qualité de service (QoS) .
802.11f	Interopérabilité entre points d'accès (IAPP) .
802.11g	Couche physique : 54Mb/s bande 2,4GHz
802.11h	Harmonisation de 802.11a avec la réglementation européenne .
802.11i	Sécurité .
802.11j	Harmonisation de 802.11a avec la réglementation japonaise
802.11k	Radio Ressource Measurement (facilite la configuration)
802.11m	Amélioration du standard 802.11 et des amendements finalisés
802.11n	Couche physique avec débit utile 100Mb/s

# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Bluetooth

Zigbee

Divers



---Tableau récapitulatif---



Standard	802.11b	802.11a	802.11g
Approuvée	Juillet 1999	Juillet 1999	Juin 2003
Débit Max.	11 Mb/s	54 Mb/s	54 Mb/s
Modulation	CCK	OFDM	OFDM et CCK
Débits supportés	11, 5.5, 2, 1 Mb/s	54, 48, 36, 24, 18, 12, 9, 6 Mb/s	CCK : 11, 5.5, 2, 1 Mbps OFDM : 54, 48, 36, 24, 18, 12, 9, 6 Mbps
Bande	2.4 GHz (2.4 GHz to 2.497 GHz)	5 GHz (5.15 to 5.35 GHz)	2.4 GHz (2.4 GHz to 2.497 GHz)
Canaux	3 canaux sans recouvrement sur 13	8 canaux sans recouvrement	3 canaux sans recouvrement sur 13
Portée Max.	125 m	50 m	125 m
Compatibilité			802.11b
Caractéristiques	Très déployé	Pour environnement à haute densité	Remplacement de 802.11b

# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Bluetooth

Zigbee

Divers

---Les normes 802.11n et 802.11y---



Standard	802.11n	802.11y
Approuvée	2007	Sept 2008
Débit Max.	Environ 125.n Mb/s <small>n=nombre d'antennes (1 à 4) –Technologie MIMO</small>	54 Mb/s
Modulation	OFDM	
Bande	2.4 GHz ou 5GHz	3.7 GHz
Portée Max.	50 m (intérieur) 125m (extérieur)	50 m (intérieur) 5000m (extérieur)



# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

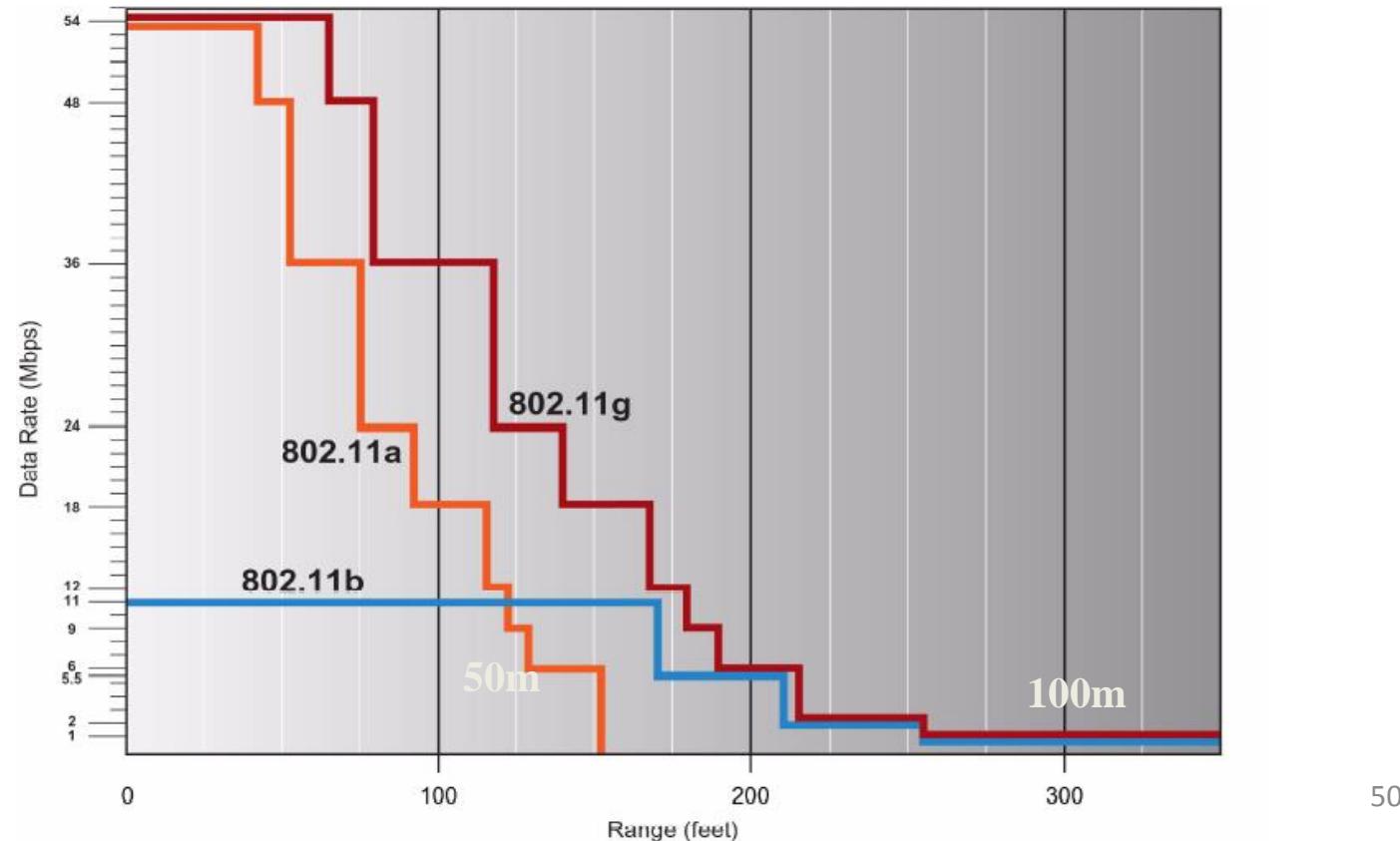
Wifi

Bluetooth

Zigbee

Divers

---Distances théoriques et débits---



# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Bluetooth

Zigbee

Divers

---Modèle de référence IEEE 802---

## Modèle OSI



## Modèle IEEE 802

**802.11 (LLC)**

**802.11 (MAC)**

**802.3**

**802.5**

**802.11  
FHSS**

**802.11  
DSSS**

**802.11  
IR**

**802.11b**

**802.11a**

**802.11g**

Ethernet Physique

Token Ring Physique

# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

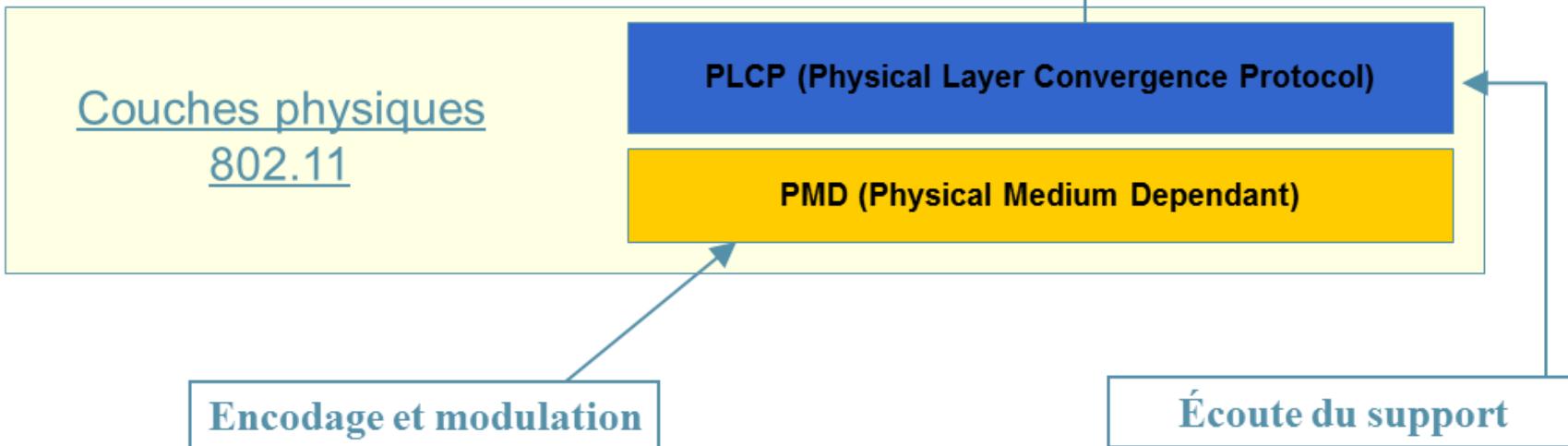
Bluetooth

Zigbee

Divers

---Modèle de référence IEEE 802---

CCA = Clear Channel Assesment



# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Bluetooth

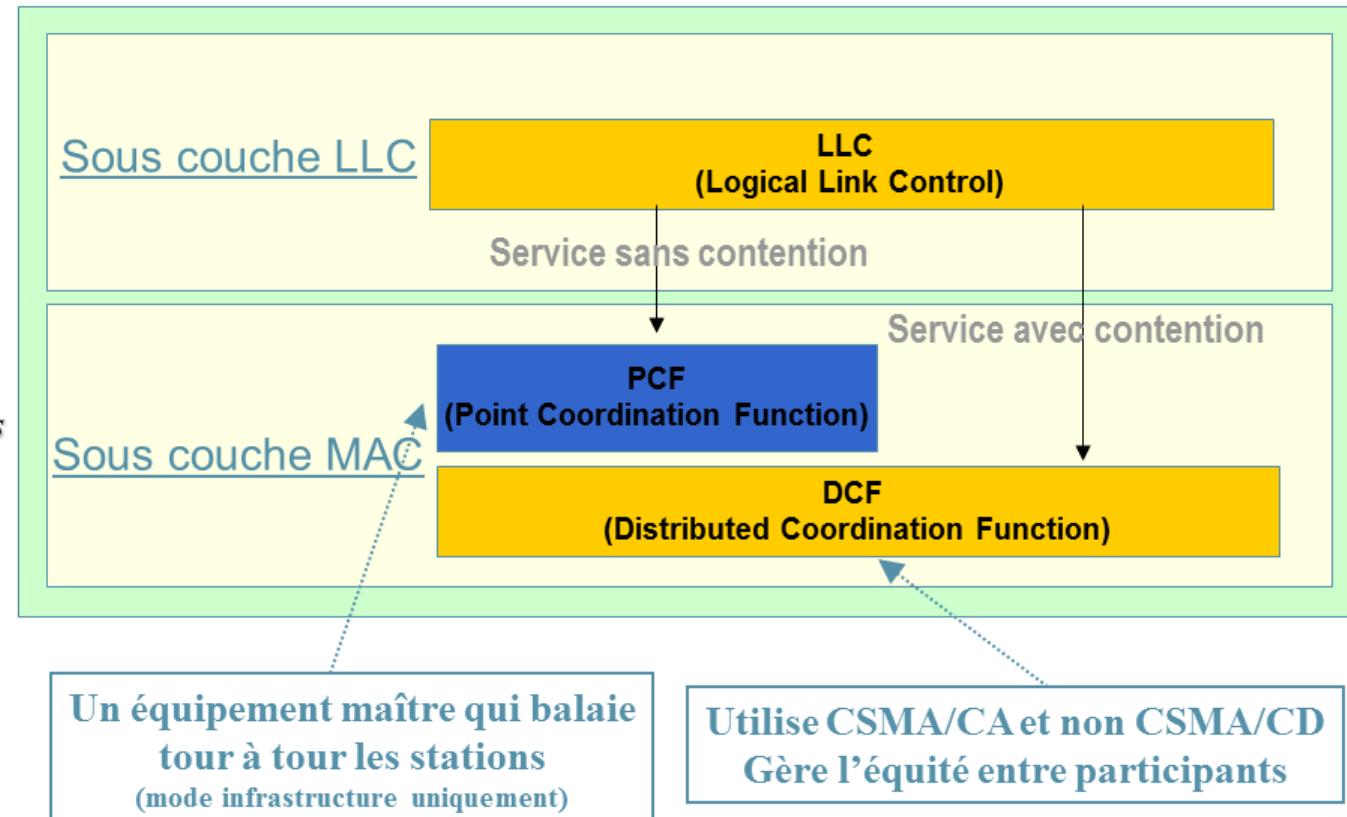
Zigbee

Divers

---Modèle de référence IEEE 802---

*Délivrance des données entre couches MAC et réseaux*

*Méthode d'accès au médium*



# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Bluetooth

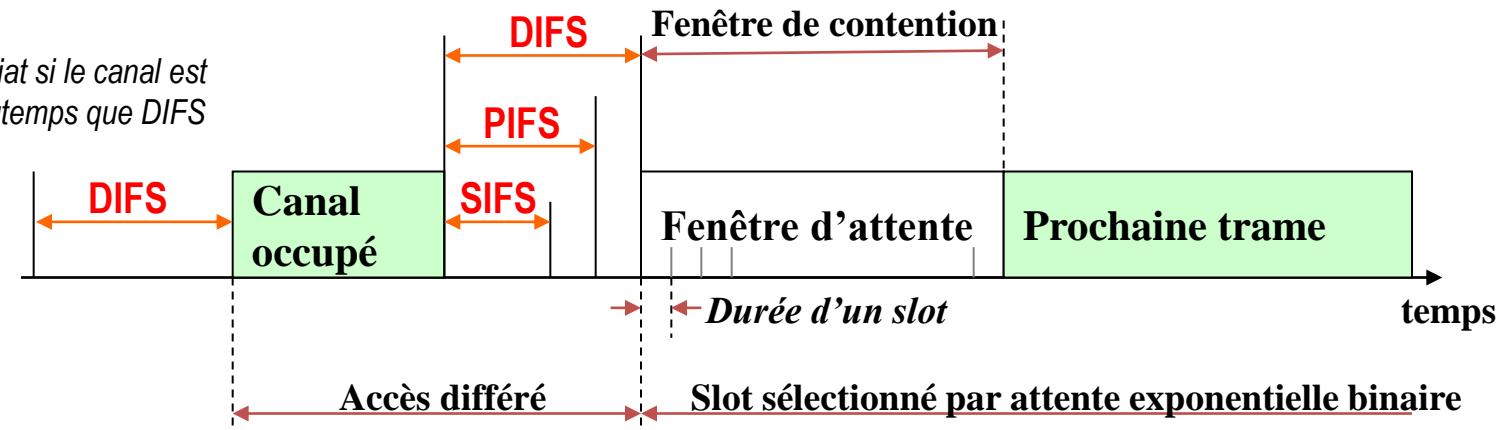
Zigbee

Divers

## ---Synchronisation couche MAC (IEEE 802.11)---

### Méthode d'accès basique

Accès immédiat si le canal est libre plus longtemps que DIFS



**SIFS (Short IFS):** Espace le plus court utilisé pour toutes actions en réponse (priorité + importante pour CTS,ACK...)

**PIFS (PCF-IFS):** Espace moyen utilisé par le contrôleur central dans une configuration PCF lors de l'émission d'une trame de polling (priorité moyenne pour opération en time bounded)

**DIFS (DCF-IFS):** Espace le plus long utilisé comme délai minimal pour les trames asynchrones rivalisant pour obtenir le canal (PIFS+128us : priorité la plus basse)

**EIFS ( Extended-IFS):** Utilisé à la suite d'un mauvais FCS (EIFS>DIFS)

# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Bluetooth

Zigbee

Divers

-  **Bluetooth®** : présentation

- Technologie de télécommunication **multiservice**
- Ordinateur, périphérique, téléphone portable, assistant personnel, réfrigérateur, ...
- **Coût extrêmement bas** pour être embarqué n'importe où

➔ Peut être considéré comme l'équivalent de l' **USB pour le sans fil.**

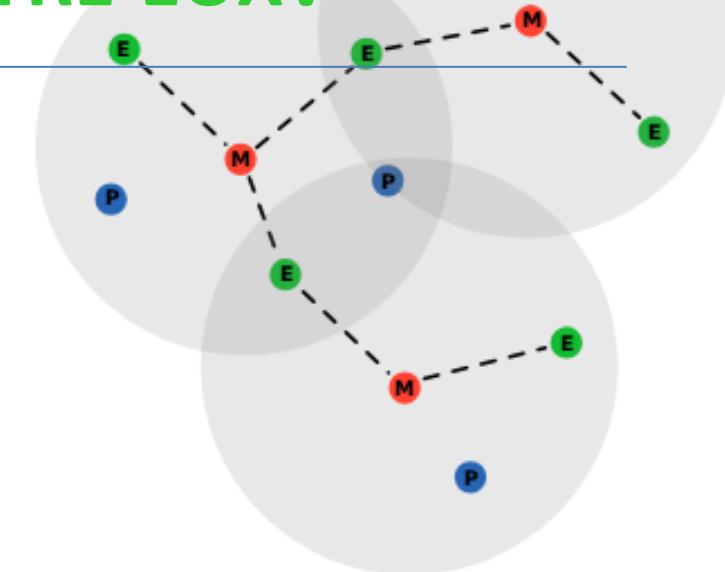
# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Blu

- **Bluetooth®**: caractéristiques
  - Architecture en étoile
  - 8 équipements actifs dans un « piconet » (200 inactifs)
  - « scatternet » = plusieurs « piconets »
  - 3 classes de puissances : 1mW, 2,5mW et 100mW
    - 10cm, 10m et 100m
  - 720kb/s maximum
  - Liens **asynchrones** (données)
  - Liens **synchrones** (voix)
  - Bande **2,4GHz**



# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

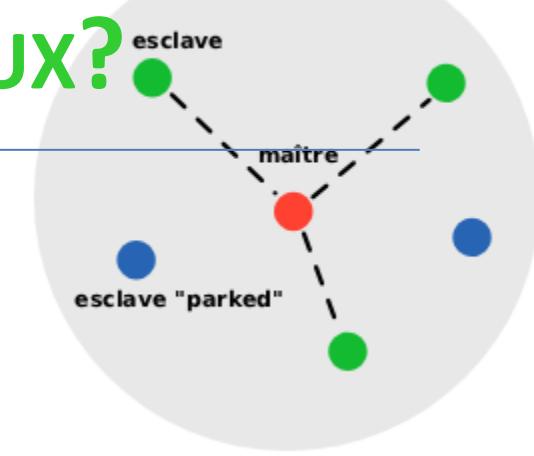
Avant-propos

Wifi

Bluetooth

-  **Bluetooth®**: caractéristic

- Architecture en **étoile étendue**
- **8 équipements actifs** dont 1 maître par « **piconet** » (200 inactifs)
- « **scatternet** » = plusieurs « **piconets** »
- **3 classes de puissances** : 1mW, 2,5mW et 100mW  
10cm, 10m et 100m
- **720kb/s maximum**
- Liens **asynchrones** (données)
- Liens **synchrone**s (voix)
- Bande **2,4GHz**



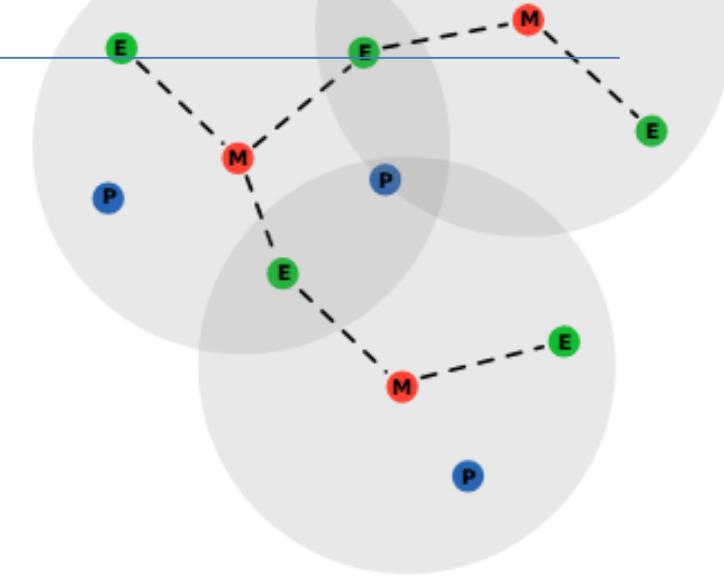
# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Blu

- **Bluetooth®**: caractéristiques
  - Architecture en étoile
  - 8 équipements actifs dans un anneau (200 inactifs)
  - « scatternet » = plusieurs « piconets »
  - 3 classes de puissances : 1mW, 2,5mW et 100mW
    - 10cm, 10m et 100m
  - 720kb/s maximum
  - Liens **asynchrones** (données)
  - Liens **synchrones** (voix)
  - Bande **2,4GHz**



# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Bluetooth

Zigbee

Divers

Technology	Classic Bluetooth technology (BR/EDR) <sup>1</sup>	Bluetooth low energy technology <sup>2</sup>
Radio Frequency	2.4 GHz	2.4 GHz
Distance / Range	10 to 100 meters <sup>3</sup>	10 to 100 meters <sup>3</sup>
Over the air Data Rate	1-3Mbps	1Mbps
Application Throughput	0.7-2.1 Mbps	0.2 Mbps
Nodes/Active Slaves	7 / 16777184 <sup>5</sup>	Unlimited <sup>6</sup>
Security	64b/128b and applications layer user defined	128b AES and application layer user defined
Robustness	Adaptive fast frequency hopping, FEC, fast ACK	Adaptive fast frequency hopping
Latency (from a non connected state)		
Total time to send data (det.battery life) <sup>8</sup>	100ms	<3ms
Government Regulation	Worldwide	Worldwide
Certification Body	Bluetooth SIG	Bluetooth SIG
Voice capable	Yes	No
Network topology	Scatternet	Star-bus
Power Consumption	1 as the reference	0.01 to 0.5(depending on use-case)
Peak current consumption (max 15 mA to run on coin cell battery)	<30 mA	<15 mA
Service discovery	Yes	Yes
Profile concept	Yes	Yes
Primary Use Cases	Mobile phones, gaming, headsets, stereo audio streaming, automotive, PCs, consumer electronics, etc.	Mobile phones, gaming, PCs, watches, sports & fitness, healthcare, automotive, consumer electronics, automation, industrial, etc.



# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

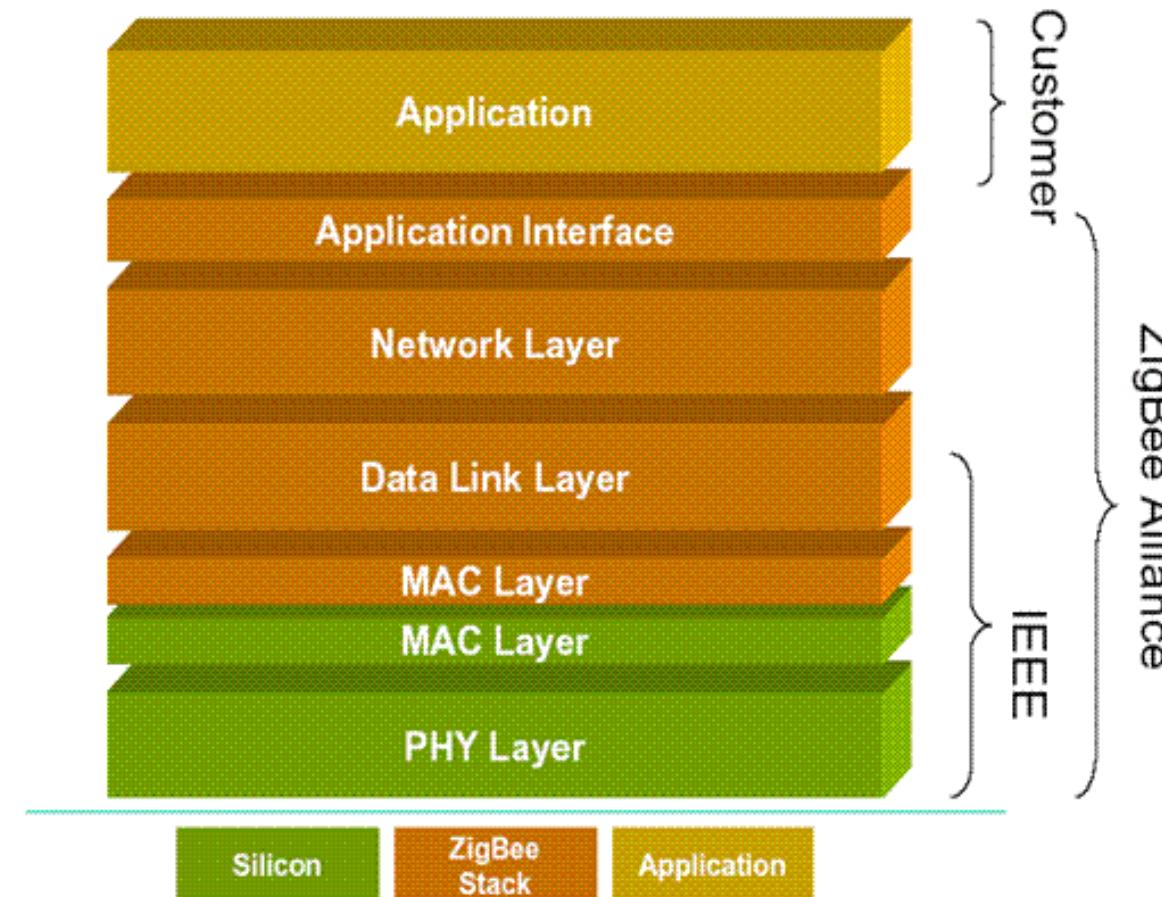
Avant-propos

Wifi

Bluetooth

Zigbee

Divers



# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Bluetooth

Zigbee

Divers

Series 1

Series 2

Typical (indoor/urban) range	30 meters	40 meters
Best (line of sight) range	100 meters	120 meters
Transmit/Receive current	45/50 mA	40/40 mA
Firmware (typical)	802.15.4 point-to-point	ZB ZigBee mesh
Digital input/output pins	8 (plus 1 input-only)	11
Analog input pins	7	4
Analog (PWM) output pins	2	None
Low power, low bandwidth, low cost, addressable, standardized, small, popular	Yes	Yes
Interoperable mesh routing, ad hoc network creation, self-healing networks	No	Yes
Point-to-point, star topologies	Yes	Yes
Mesh, cluster tree topologies	No	Yes
Single firmware for all modes	Yes	No
Requires coordinator node	No	Yes
Point-to-point configuration	Simple	More involved
Standards-based networking	Yes	Yes
Standards-based applications	No	Yes
Underlying chipset	Freescale	Ember
Firmware available	802.15.4 (IEEE standard), DigiMesh (proprietary)	ZB(ZigBee 2007), ZNet 2.5 (obsolete)
Up-to-date and actively supported	Yes	Yes

# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Bluetooth

Zigbee

Divers

---Un exemple simple de réseau ZigBee---

## Coordinator

One per PAN  
Establishes/Organizes a PAN  
Mains-powered



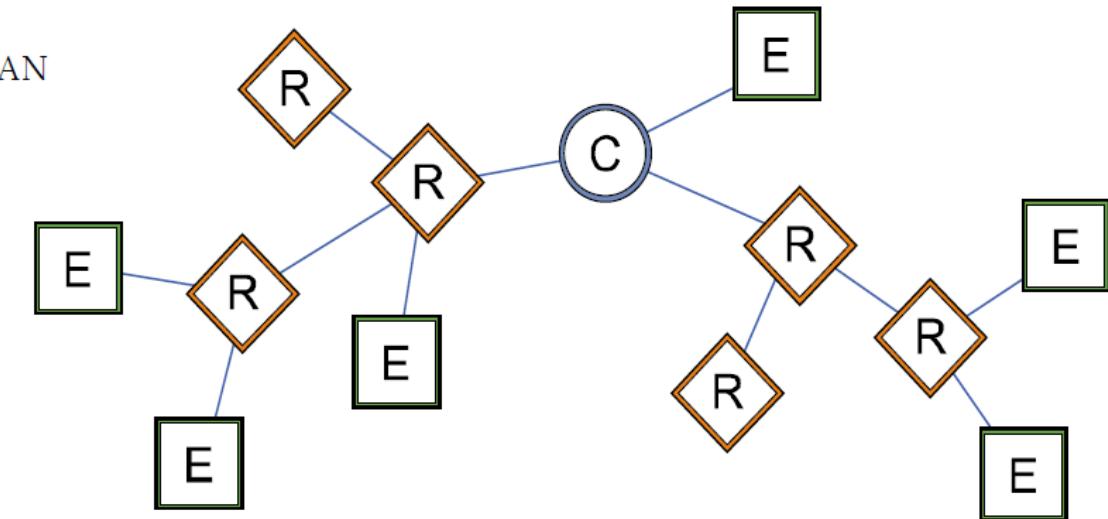
## Router

Optional  
Several can be in a PAN  
Mains-powered



## End Device

Several can be in a PAN  
Low power



# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Bluetooth

Zigbee

Divers

## ---Limites d'un réseau *MaxStream ZigBee PAN*---

### ■ Adressage:

- Un nœud possède une adresse unique sur 64 bits
- Quand il est connecté au réseau, le nœud reçoit une adresse réseau sur 16bits

### ■ Routage:

- Utilisation de AODV

### ■ Communication « broadcast »

- Utilisation d'un schéma d'acquitements passif
  - Utilise l'écoute indiscrète pour vérifier que tous les voisins ont bien répétés le message.
  - Si un voisin n'a pas retransmis on re émet le message à broadcaster



## 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

## Avant-propos

Wif

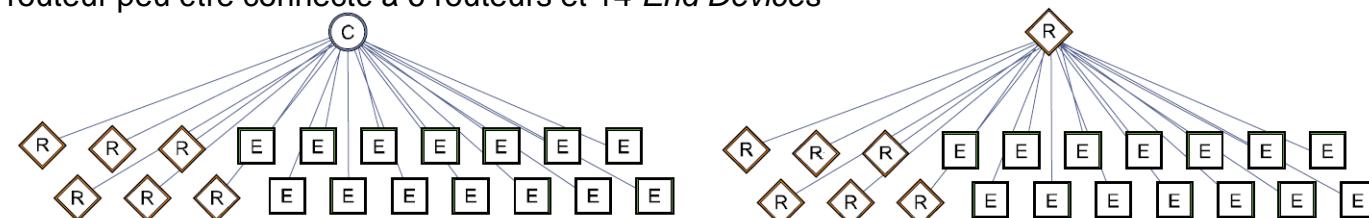
## Bluetooth

# Zigbee

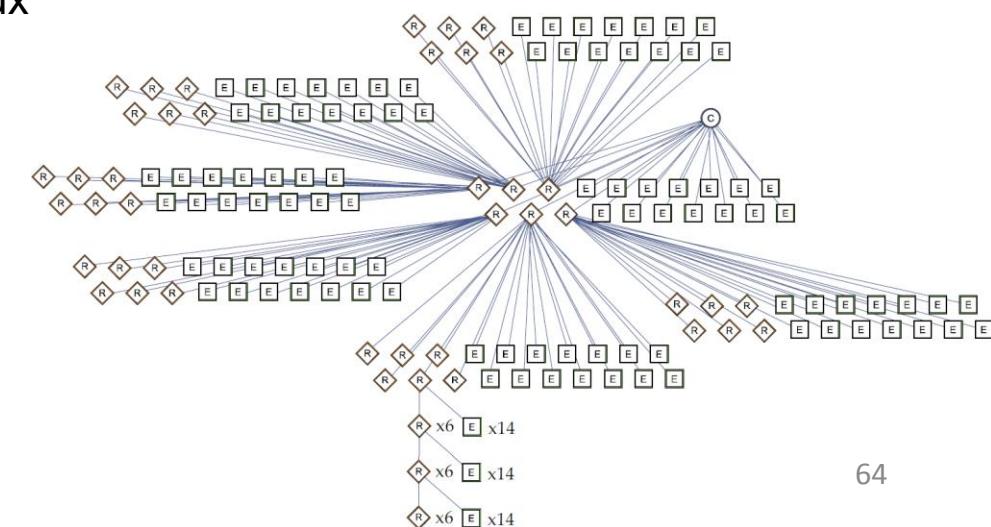
## Divers

# ---Limites d'un réseau *MaxStream ZigBee PAN*---

- Nombre maximum d'enfants par parent:
    - 1 coordinateur/PAN,
    - 1 routeur peu être connecté à 6 routeurs et 14 *End Device*



- Profondeur maximale du réseau
    - 5 niveaux maximum



# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Bluetooth

Zigbee

Divers

	Contrôleur	RAM	EEPROM	Flash	Fabriquant
<b>WiSMote</b>	MSP430x5 CC2520	16k	nc	256k	Arago Systems <sup>171</sup>
<b>MICAz</b>	Atmel ATmega128L	nc	4k	128k	Crossbow <sup>172</sup>
<b>TELOSB</b>	TI MSP430	10k	16k	48k	Crossbow <sup>173</sup>
<b>JN5139</b>	32-bits RISC processor	96k	192k	externe	Jennic <sup>174</sup>
<b>RC2xxx</b>	Single-cycle high performance 8051	8k	4k	32-256k	Radiocrafts <sup>175</sup>
<b>WPC-IP</b>	MSP430F5	16k	nc	256k	Watteco <sup>176</sup>
<b>NanoStack</b>	TI CC1110	4-8k	32-64k	32k	Sensinode <sup>177</sup>
<b>Tmote Sky</b>	TI MSP430	10k	nc	48k	Moteiv ⇒ Sentilla <sup>178</sup>
<b>eSPOT</b>	ARM 926ej-S	1M	nc	8M	Sun SPOT <sup>179</sup>
<b>PN2420</b>	Atmega128L / MSP430	4k	nc	128k	PicosNet <sup>180</sup>

# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

Avant-propos

Wifi

Bluetooth

Zigbee

Divers

---Bien choisir son **protocole de routage**--

- **Fiabilité du protocole**

Est-on sûr que le message arrive à bon port?  
Sinon, est-on prévenu?

- **Rendement du protocole**

Quelle définition du rendement?

Quid du traditionnel  $\Gamma = \frac{U}{U+G+K+Tt*D} * (1-T_e)^{(U+G+K)} ?$

- **Caractère optimal des routes?**

– Quels critères à optimiser?

# 4. COMMENT VONT COMMUNIQUER LES OBJETS ENTRE EUX?

---SYNTHESE---

	Zigbee	Bluetooth	Wi-Fi
Besoins en mémoire	4-32 Kb	250 Kb	1 Mb
Autonomie avec pile	Années	Jours	Heures
Nombre de nœuds	65 000+	7	32
Vitesse de transfert	250 Kb/s	1 Mb/s	11-54-108... Mb/s
Portée	10-100m	10-100 m	300 m

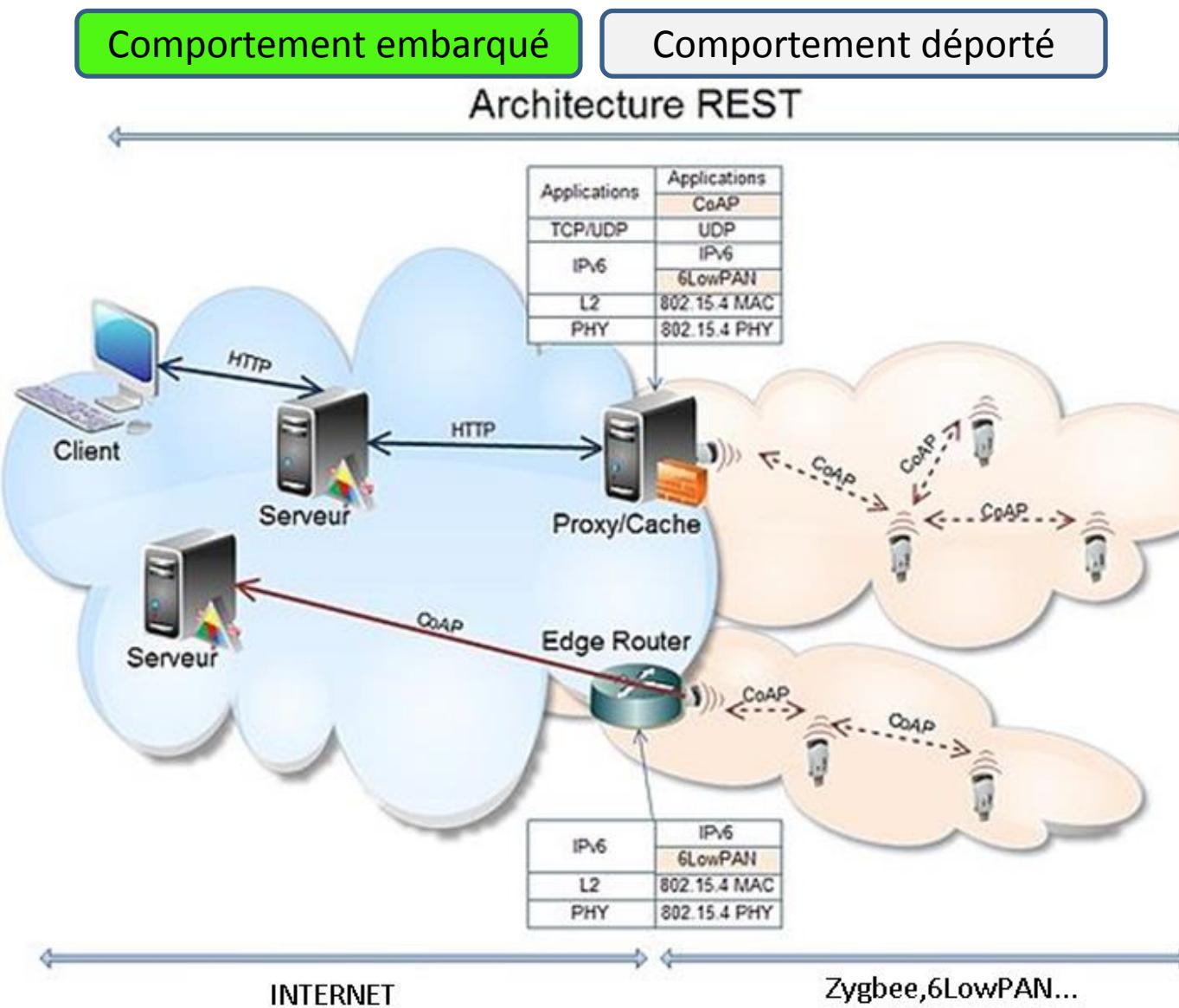
# 5. COMMENT MON OBJET ACCÈDE À INTERNET?

- Deux grands types d'objets physiques
  - Objets **communicants** et/ou **intelligents**
    - A base de micro-contrôleur(s) /FPGA équipés d'interface(s) de communication
    - Modèle de **COMPORTEMENT EMBARQUÉ** sur l'objet
  - Objets « **chipless** » tagués
    - Objets sur lesquels on a apposé une étiquette RFID (ou autre)



- Modèle de **COMPORTEMENT DÉPORTÉ** sur un serveur distant

# 5. COMMENT MON OBJET ACCÈDE À INTERNET?

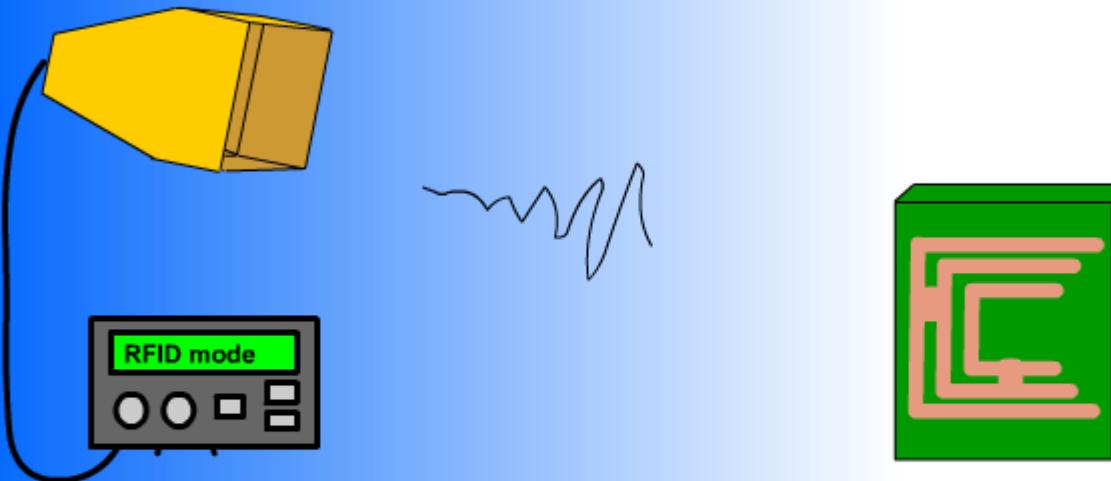


# 5. COMMENT MON OBJET ACCÈDE À INTERNET?

Comportement embarqué

Comportement déporté

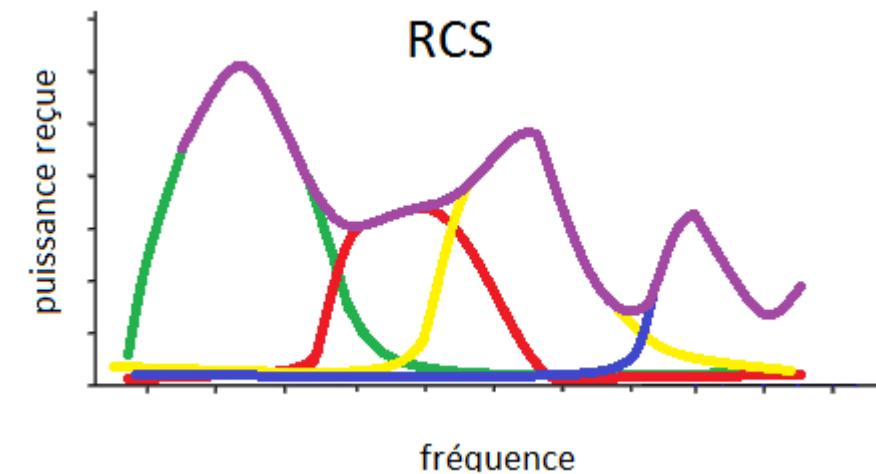
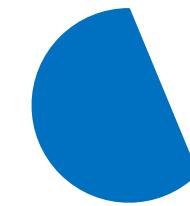
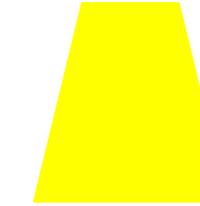
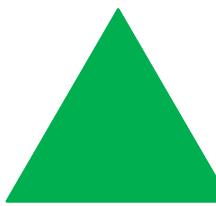
## RFID chipless tag system



# 5. COMMENT MON OBJET ACCÈDE À INTERNET?

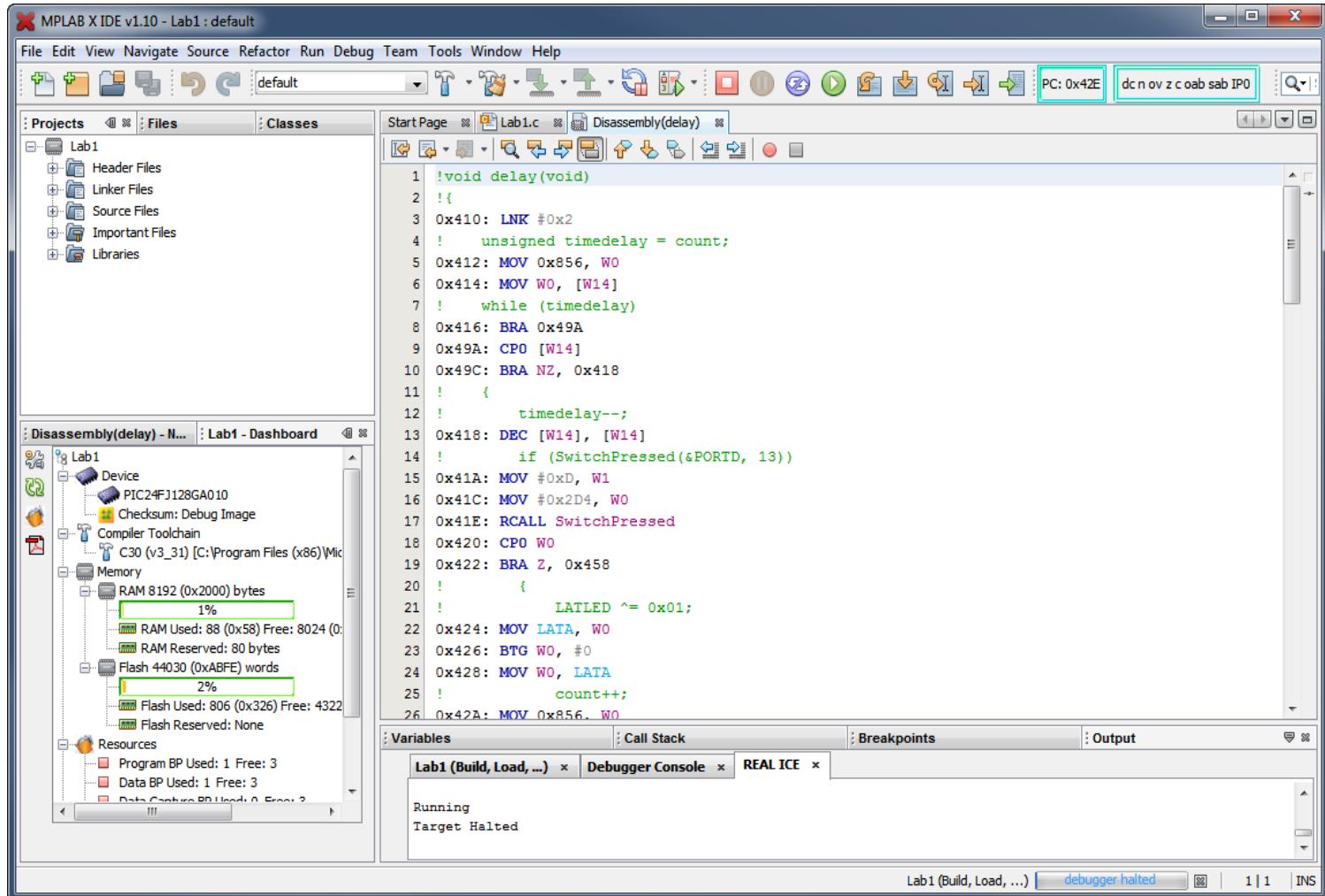
Comportement embarqué

Comportement déporté



# 6. JE DÉVELOPPE MON APPLICATION

- Choisir la chaîne de développement



# 6. JE DÉVELOPPE MON APPLICATION

- Choisir la chaîne de développement



The image shows a screenshot of the MPLAB X IDE v1.00 interface. The left pane displays a file tree for the 'emSystem' project, showing directories like 'emSystem', 'HAL', 'emTest', 'FreeRTOS', etc., and files such as 'ConsoleFont8x13x96.C', 'emBitmap.h', 'emCallBack.c', etc. The right pane is a code editor with the file 'MikroeMMB32\_Touch.c' open. The code is written in C and includes comments explaining touch module functions like 'emTouchInit' and 'emSetTouchMode'. A status bar at the bottom indicates 'PC: 0x0'.

```
150 // Touch module code.
151 //*****
152 153 /**
154 * Function: emTouchInit
155 * Args: none
156 * Returns: none
157 * Desc: Put the touch interface into a known state. Call
158 * once only at startup time. At later times instead
159 * use SetTouchMode(TOUCH_MODE_OFF)
160 */
161 void mCode16 emTouchInit(void)
162 {
163     touchStatus.mode      = emTOUCH_MODE_OFF;
164     touchStatus.touching  = touching  = FALSE;
165     touchStatus.touchMake = touchMake = FALSE;
166     touchStatus.touchBreak= touchBreak = FALSE;
167
168     // Setup event transmission.
169 #if TOUCH_REPORTING == POLLED_MODE
170     writeSeq = 0;
171     readSeq = 0;
172 #elif TOUCH_REPORTING == EVENT_MODE
173     vSemaphoreCreateBinary(touchSemaphore);
174 #endif
175
176     // Setup the default limit values.
177     tp.xl = 200;
178     tp.xu = 800;
179     tp.yl = 200;
180     tp.yu = 800;
181 }
182
183 /**
184 * Function: emSetTouchMode
185 * Args: mode - a member of the TouchMode enum that
```

176 | 26 | INS

# 6. JE DÉVELOPPE MON APPLICATION

- Choisir la chaîne de développement



The screenshot shows the MPLAB X IDE v1.20 interface for developing a PIC24\_GA010\_GFXv3\_EX16\_8MP\_QVGA1 project. The main window displays a C code file for a Graphics Object Layer (GOLMsgCallback). The code handles various events like time delays, progress bars, and static text updates. The interface includes a Projects pane, a Files pane, a code editor with syntax highlighting, a Call Stack pane, a File Registers pane showing memory dump, and a Call Graph pane showing the execution flow between functions like main, GOLDraw, and GOLFindObject.

```
792 L WORD GOLDrawCallback(void)
793 {
    static DWORD prevTick = 0;                                // keeps previous value of tick
    static BYTE direction = 1;                                // direction switch for progress bar
    static BYTE arrowPos = 0;                                // arrows pictures position for custom control demo
    static WORD wDelay = 40;                                 // progress bar delay variable
    OBJ_HEADER *pObj = NULL;                               // used to change text in Window
    SLIDER *pSlid;
    LISTBOX *plib;
    #if !defined(_dsPIC33FJ128GP804_) || defined(_PIC24HJ128GP504_)||defined(GFX_PICTAIL_LCC)
    static DWORD prevTime = 0;                                // keeps previous value of time tick
    WORD i;
    // check first of the objects for the time display is already created.
    if (GOLFindObject(ID_STATICTEXT1) != NULL) {
        // update the time display
        if((screenState & 0x0000F300) != 0x0000F300)
        {
            // process only when NOT setting time and date
            if((tick - prevTime) > 1000)
            {
                RTCCProcessEvents();
                i = 0;
                while(i < 12)
                {
                    dateTimeStr[i] = _time_str[i];
                    dateTimeStr[i + 13] = _date_str[i];
                    i++;
                }
                dateTimeStr[12] = 0x000A;          // (XCHAR) '\n';
                dateTimeStr[25] = 0x0000;          // (XCHAR) '\0';
                if(pObj = GOLFindObject(ID_STATICTEXT1))
                {

```

Airbus' Vision for RFID-enabled Business Processes

# 6. JE DÉVELOPPE MON APPLICATION

- Etudier les *datasheets* du micro-contrôleur

## 7.3 Reading the Data EEPROM Memory

To read a data memory location, the user must write the address to the EEADR:EEADRH register pair, clear the EEPGD control bit (EECON1<7>), clear the CFGS

control bit (EECON1<6>) and then set the RD control bit (EECON1<0>). The data is available for the very next instruction cycle; therefore, the EEDATA register can be read by the next instruction. EEDATA will hold this value until another read operation or until it is written to by the user (during a write operation).

### EXAMPLE 7-1: DATA EEPROM READ

```
MOVLM DATA_EE_ADDRH      ; Upper bits of Data Memory Address to read
MOVWF  EEADRH              ; Lower bits of Data Memory Address to read
MOVLM DATA_EE_ADDR          ; Point to DATA memory
MOVWF  EEADR
BCF    EECON1, EEPGD        ; Access EEPROM
BCF    EECON1, CPGS
BSF    EECON1, RD            ; EEPROM Read
MOVF   EEDATA, W             ; W = EEDATA
```

## 7.4 Writing to the Data EEPROM Memory

To write an EEPROM data location, the address must first be written to the EEADRH:EEADR register pair and the data written to the EEDATA register. Then the sequence in Example 7-2 must be followed to initiate the write cycle.

The write will not initiate if the above sequence is not exactly followed (write 5h to EECON2, write AAh to EECON2, then set WR bit) for each byte. It is strongly recommended that interrupts be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable writes. This mechanism prevents accidental writes to data EEPROM due to unexpected code

execution (i.e., runaway programs). The WREN bit should be kept clear at all times except when updating the EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, EECON1, EEADRH, EEADR and EEDATA cannot be modified. The WR bit will be inhibited from being set unless the WREN bit is set. Both WR and WREN cannot be set with the same instruction.

At the completion of the write cycle, the WR bit is cleared in hardware and the EEPROM Write Complete Interrupt Flag bit (EIF) is set. The user may either enable this interrupt or poll this bit. EIF must be cleared by software.

### EXAMPLE 7-2: DATA EEPROM WRITE

```
MOVLM DATA_EE_ADDRH      ; Upper bits of Data Memory Address to write
MOVWF  EEADRH              ; Lower bits of Data Memory Address to write
MOVLM DATA_EE_ADDR          ; Point to DATA memory
MOVWF  EEADR
MOVLM DATA_EE_DATA          ; Data Memory Value to write
MOVWF  EEDATA
BCF    EECON1, EEPGD        ; Access EEPROM
BCF    EECON1, CPGS
BSF    EECON1, WREN          ; Enable writes

BCF    INTCON, GIE          ; Disable Interrupts
MOVLM 0x55
MOVWF  EECON2
Required Sequence           ; Write 5h
MOVLM 0xA0
MOVWF  EECON2
BSF    EECON1, WR            ; Set WR bit to begin write
BSF    INTCON, GIE          ; Enable Interrupts

BSF    EECON1, WREN          ; User code execution
                                ; Disable writes on write complete (EIF set)
```

# 7. JE VALIDE MON SYSTÈME

## ---Validation---

Risques acceptables / Satisfaction des utilisateurs

Sûreté de fonctionnement	Satisfaction des utilisateurs
Fiabilité	QoS
Disponibilité	Performances
Maintenabilité	
Sécurité confidentialité	
Sécurité innocuité	
...	

# 7. JE VALIDE MON SYSTÈME

## ---Des outils spécifiques--

### MODELS INVOLVED IN THE SYSTEM SIMULATION

#### *Environnement description models*



Environment physical model



Energy consumption model



Wave propagation model

#### *Entities involved models*



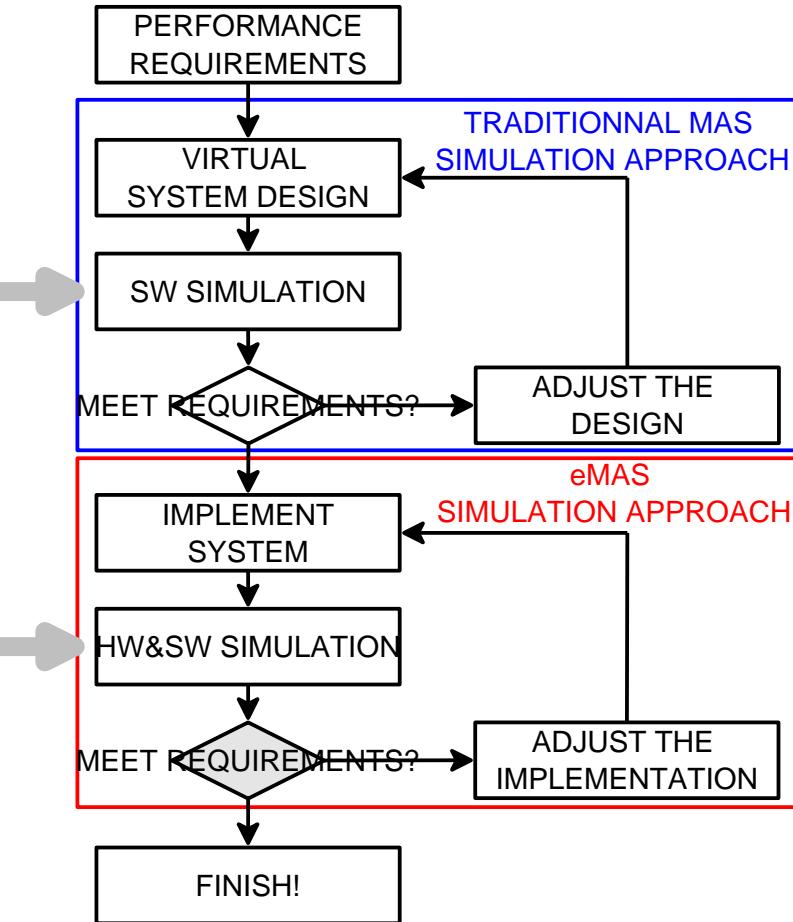
Entity behavior



Interaction model

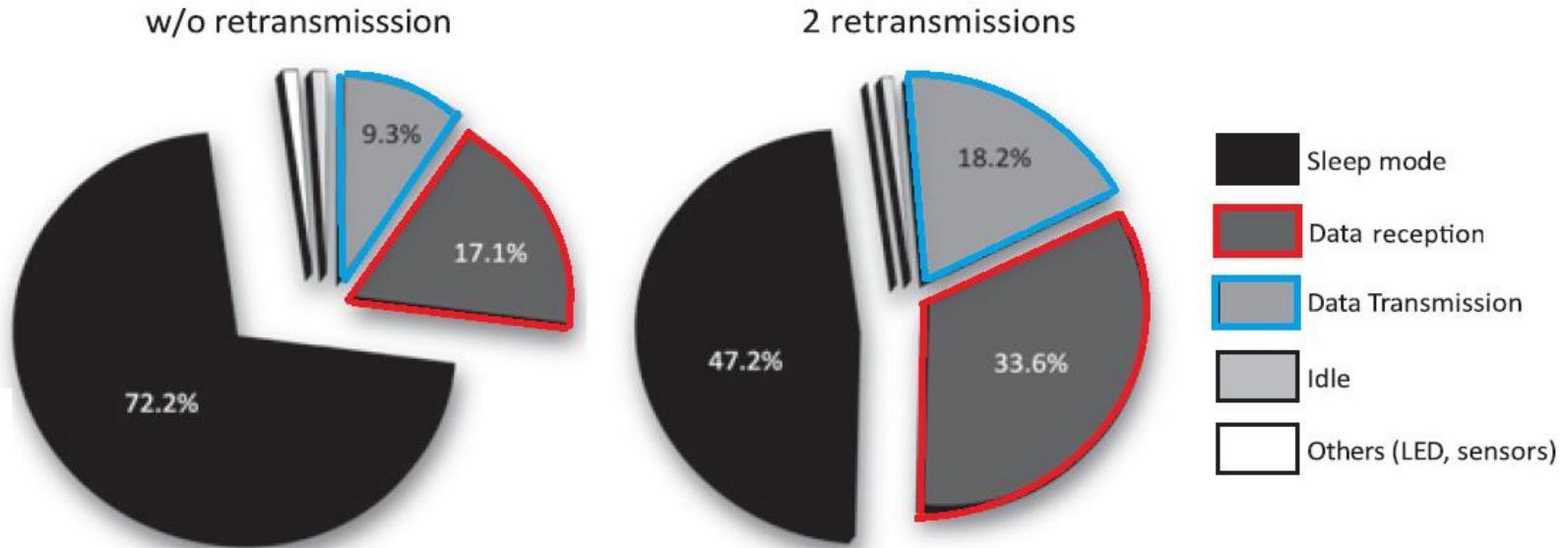


Organization model



# 7. JE VALIDE MON SYSTÈME

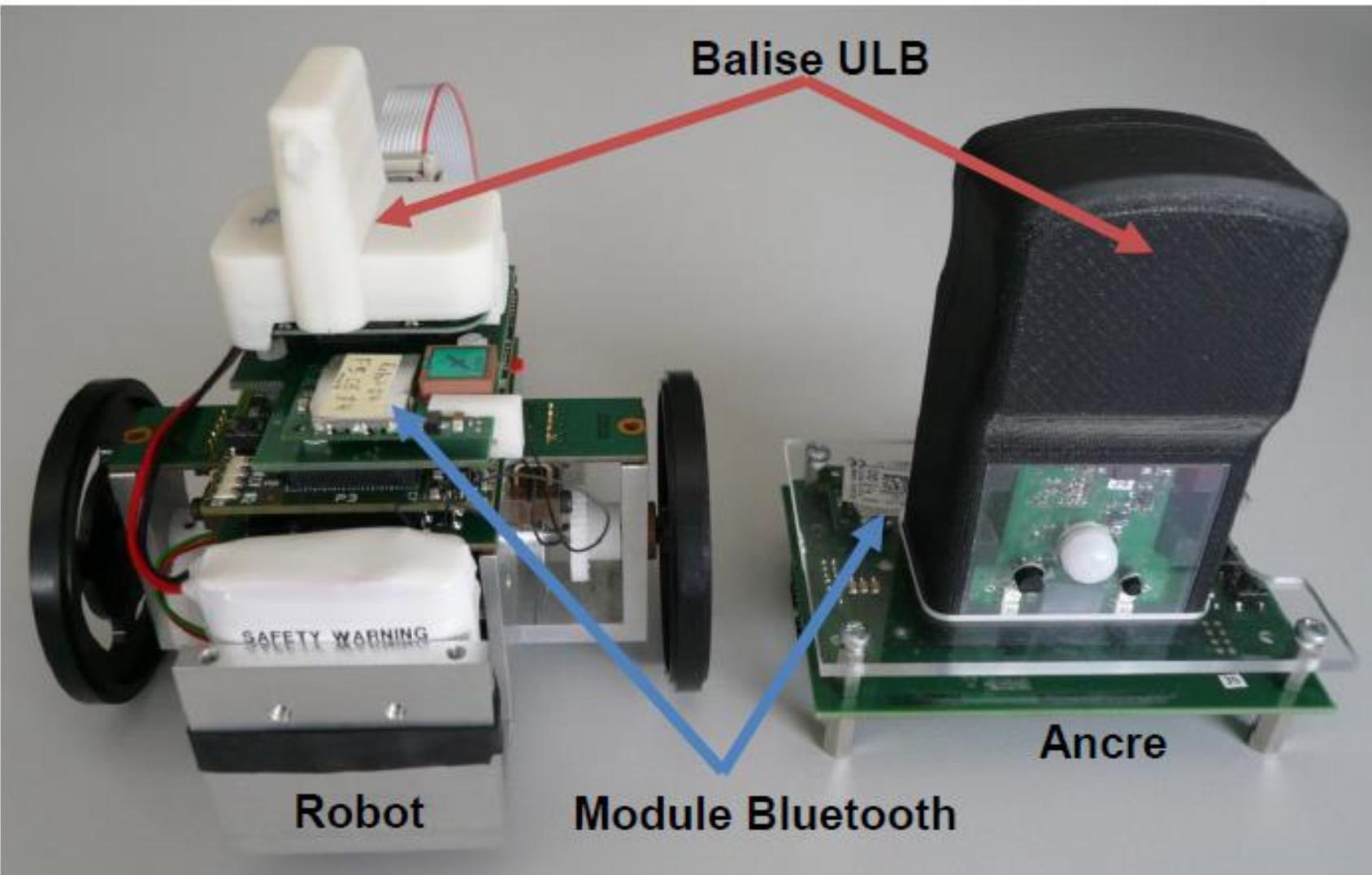
## ---Performances énergétiques---



**Fig. 6.** Energy consumption repartition for different scenari.

[Fourty12]

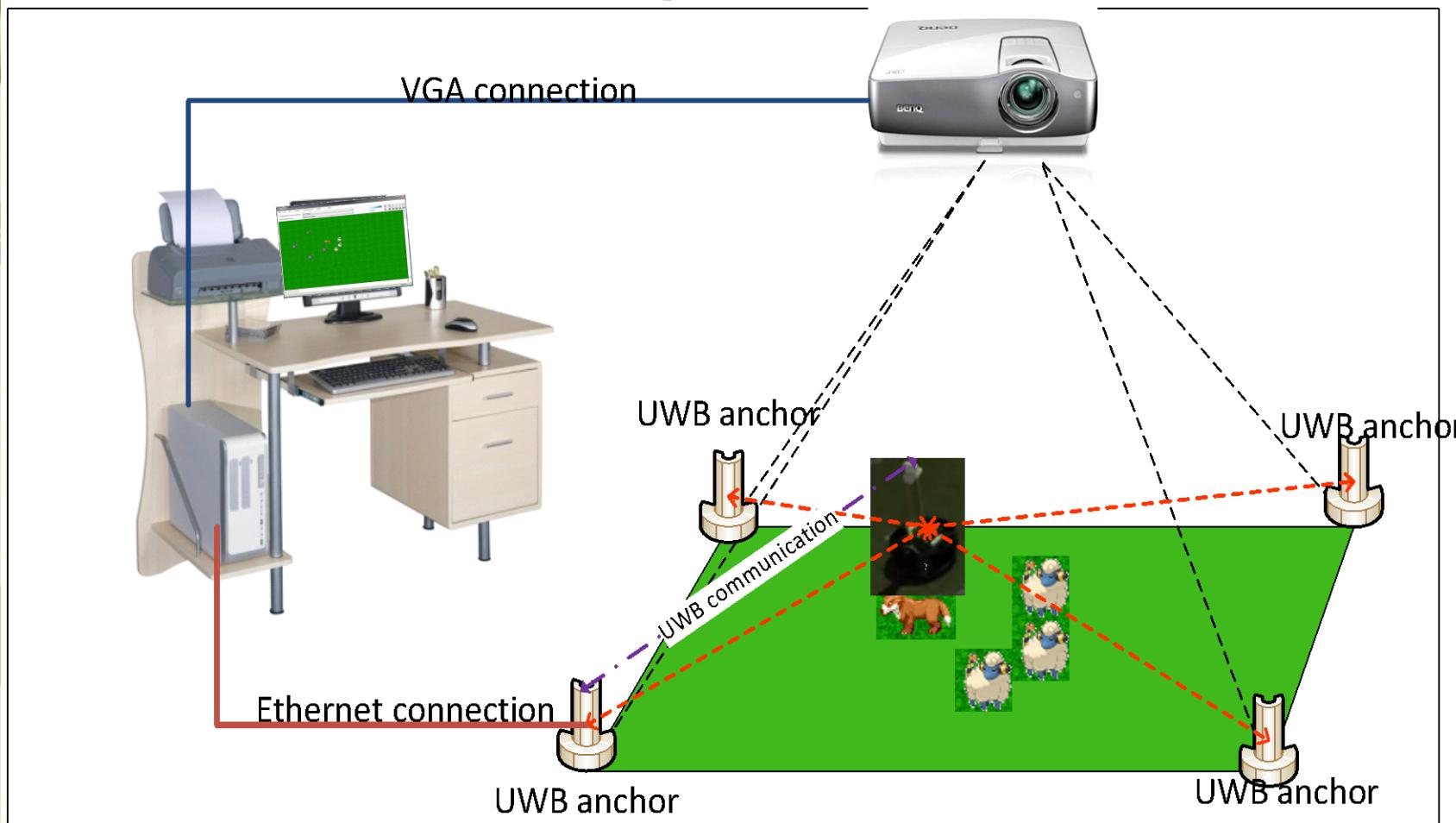
# CONCLUSION : UNE ILLUSTRATION PROJET KURASU (CEA-LETI & Cosy/LCIS/G.INP)



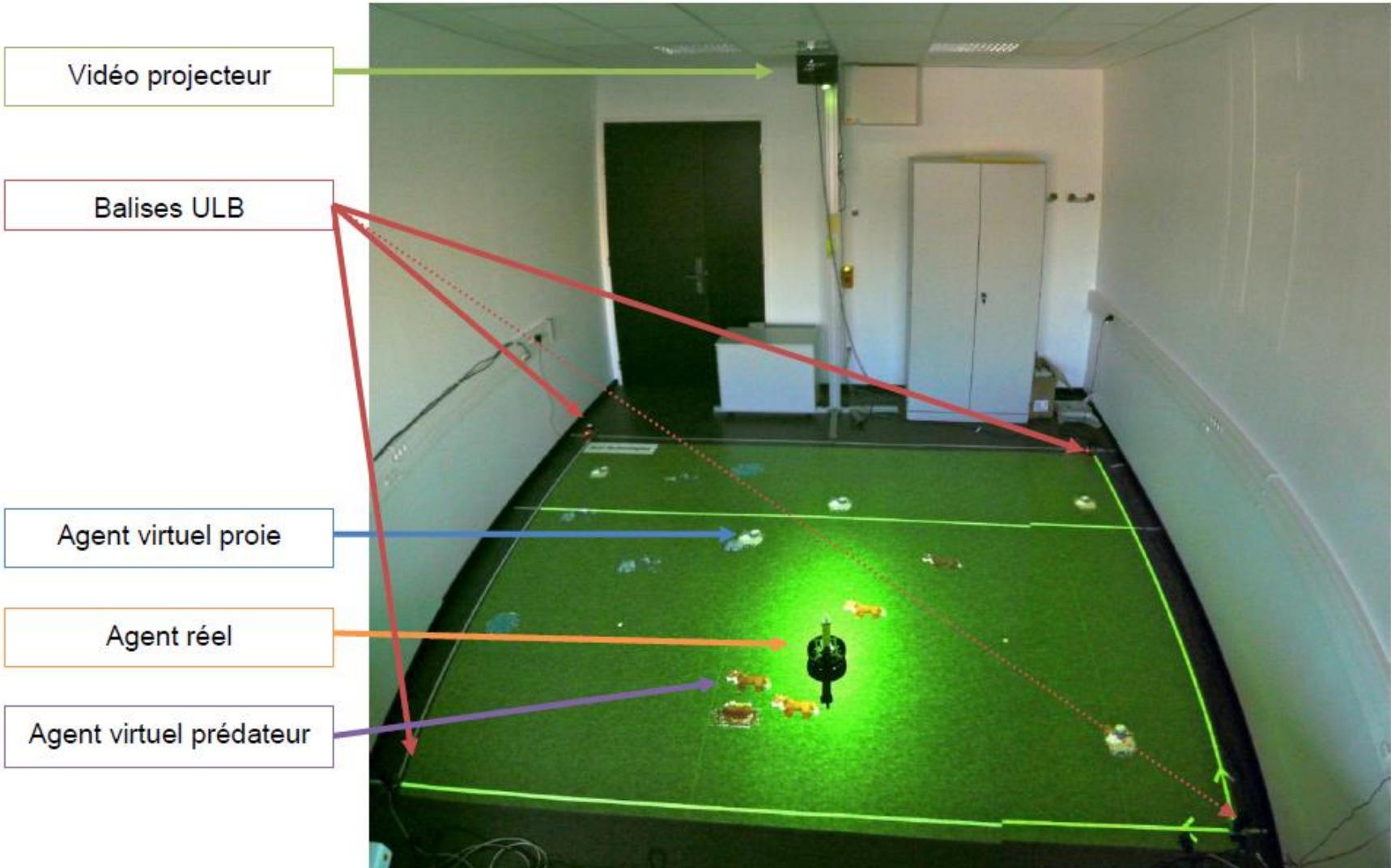
# CONCLUSION : UNE ILLUSTRATION PROJET KURASU (CEA-LETI & Cosy/LCIS/G.INP)



# CONCLUSION : UNE ILLUSTRATION PROJET KURASU (CEA-LETI & Cosy/LCIS/G.INP)



# CONCLUSION : UNE ILLUSTRATION PROJET KURASU (CEA-LETI & Cosy/LCIS/G.INP)



# CONCLUSION : UNE ILLUSTRATION PROJET KURASU (CEA-LETI & Cosy/LCIS/G.INP)



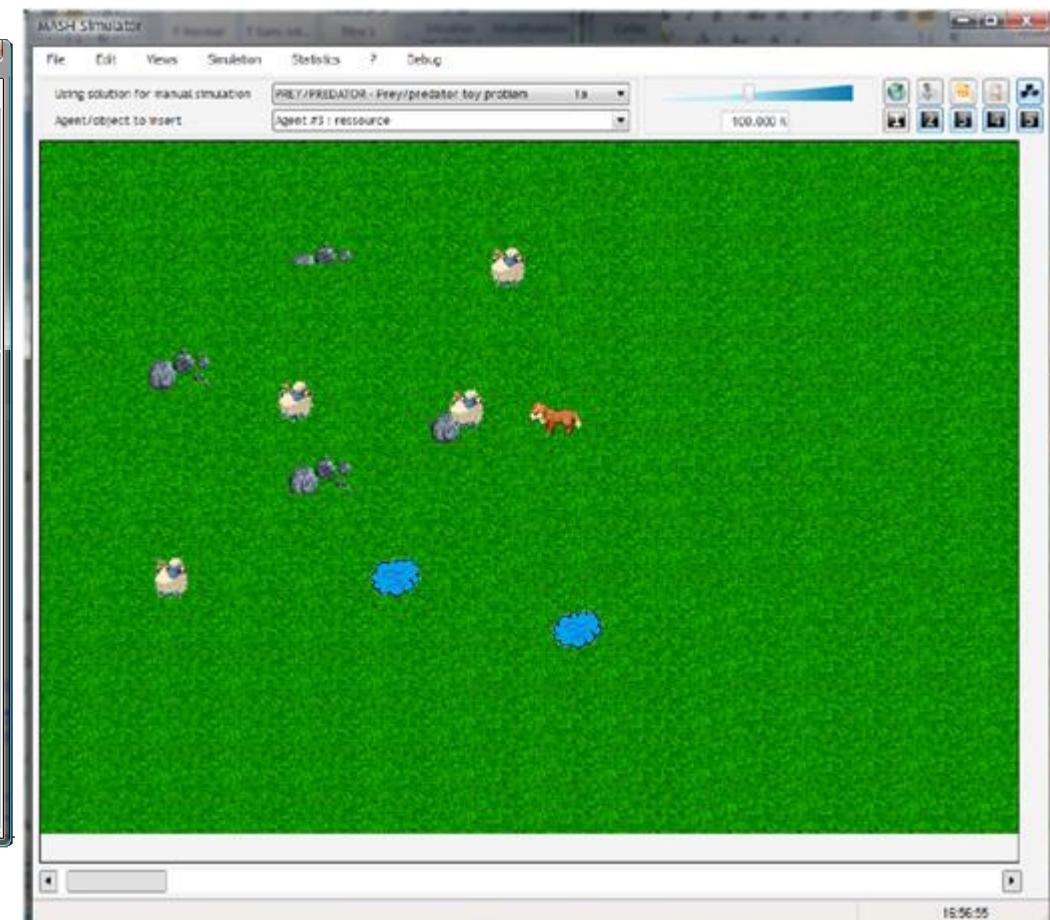
Spy agent #1 SPY\_1

State and events Statistics Command

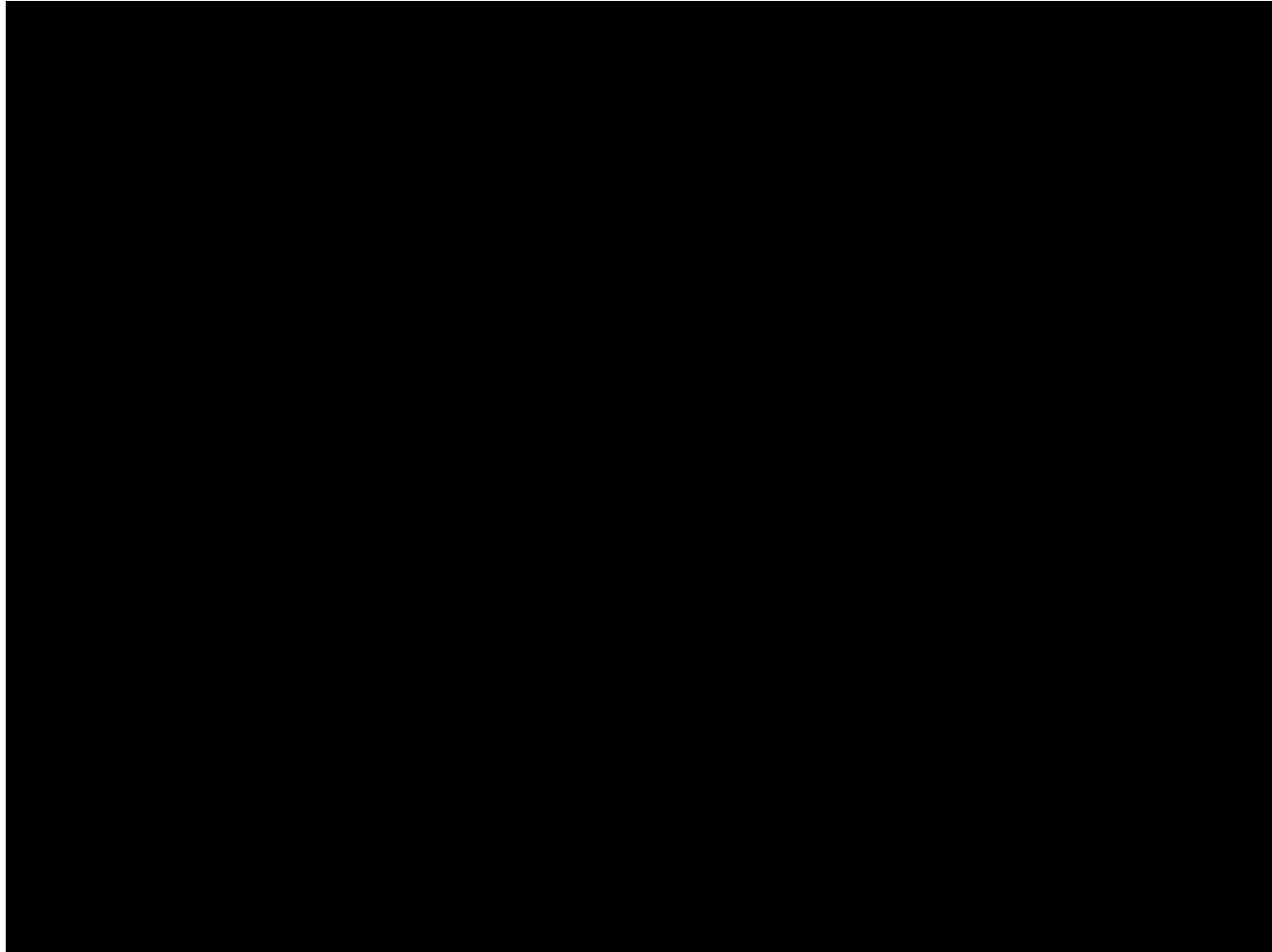
Agent #1  
role = PREY  
Energie = 57.000122  
Position (385, 242)  
vecteur\_deplacement(0.0, 0.0)  
vecteur\_d'indication[]  
decissionVectorList[]

Date Event

00:00:00.539 Object 1: Received frame Frame from 21 to all surrounding neighbors. Message is Introduction message instanciated  
00:00:00.665 Object 1: Received bytes [aa 55 00 00 00 03 ff ff ff 00 0e 00 00 00 03 ff ff ff 04 03 00 00 00 03 ]  
00:00:00.665 Object 1: Received frame Frame from 3 to all surrounding neighbors. Message is WhoAreMyNeighbors message instanciated  
00:00:00.665 Object 1: Received bytes [aa 55 00 00 00 04 ff ff ff 00 0e 00 00 00 04 ff ff ff 04 03 00 00 00 04 ]  
00:00:00.666 Object 1: Received frame Frame from 4 to all surrounding neighbors. Message is WhoAreMyNeighbors message instanciated  
00:00:00.666 Object 1: Received bytes [aa 55 00 00 00 02 ff ff ff 00 0e 00 00 00 02 ff ff ff 04 03 00 00 00 02 ]  
00:00:00.666 Object 1: Received frame Frame from 2 to all surrounding neighbors. Message is WhoAreMyNeighbors message instanciated  
00:00:00.667 Object 1: Received bytes [aa 55 00 00 00 05 ff ff ff 00 16 00 00 05 ff ff ff 04 02 00 00 00 02 00 00 00 03 00 00 ]  
00:00:00.667 Object 1: Received frame Frame from 5 to all surrounding neighbors. Message is WhoAreMyNeighbors message instanciated  
00:00:00.687 Object 1: Received bytes [aa 55 00 00 00 14 ff ff ff 00 12 00 00 14 ff ff ff 04 02 00 00 00 03 00 00 00 04 ]  
00:00:00.687 Object 1: Received frame Frame from 20 to all surrounding neighbors. Message is WhoAreMyNeighbors message instanciated  
00:00:00.689 Object 1: Received bytes [aa 55 00 00 00 15 ff ff ff 00 12 00 00 15 ff ff ff 04 02 00 00 00 03 00 00 00 04 ]  
00:00:00.689 Object 1: Received frame Frame from 21 to all surrounding neighbors. Message is WhoAreMyNeighbors message instanciated  
00:00:00.765 Object 1: Received bytes [aa 55 00 00 00 03 ff ff ff 00 0d 00 00 00 03 ff ff ff 03 00 01 38 80 ]  
00:00:00.765 Object 1: Received frame Frame from 3 to all surrounding neighbors. Message is Conflict resolution message instanciated  
00:00:00.766 Object 1: Received bytes [aa 55 00 00 00 04 ff ff ff ff 00 0d 00 00 00 04 ff ff ff 03 00 01 86 ad ]  
00:00:00.766 Object 1: Received frame Frame from 4 to all surrounding neighbors. Message is Conflict resolution message instanciated  
00:00:00.766 Object 1: Received bytes [aa 55 00 00 00 09 ff ff ff 00 09 00 00 00 02 ff ff ff 01 ]  
00:00:00.766 Object 1: Received frame Frame from 2 to all surrounding neighbors. Message is Introduction message instanciated



# CONCLUSION : UNE ILLUSTRATION PROJET KURASU (CEA-LETI & Cosy/LCIS/G.INP)



Vidéo (E. Jez, JP Jamont)

# MERCI POUR VOTRE ÉCOUTE

---



# BIBLIOGRAPHIE

---

- Culture mobile – Le Blog <http://blog.culturemobile.net>
- Jalil Boukhobza, Systèmes d'exploitation pour l'embarqué (cours)
- Julien DeAntoneti, Introduction à la programmation micro-contrôleur (cours)
- Nicolas Fourty, Adrien van den Bossche, Thierry Val: An advanced study of energy consumption in an IEEE 802.15.4 based network: Everything but the truth on 802.15.4 node lifetime. Computer Communications 35(14): 1759-1767 (2012)
- Jean-Paul Jamont, Michel Occello, Eduardo Mendes, Decentralized intelligent real world embedded systems : a tool to tune design and deployment, Advances in Intelligent and Soft- Computing, Springer, ISSN: 1867-5662, 2013.
- Jean-Paul Jamont, Les réseaux sans fils (cours)
- Emmanuel Jez, Développement d'une colonie de robots mobiles « Kurasu ».
- Stéphane Lavirotte , Objets Communicants et Terminaux Mobiles (cours)
- Etienne Perret, RFID sans puce, de l'identification au capteur .
- Olivier Rance, Construction de la signature électromagnétique d'un tag RFID sans puce à partir d'un assemblage de résonateurs.
- Jim Turley , Source: <http://www.eetimes.com/discussion/other/4025674/Operating-systems-on-the-rise>