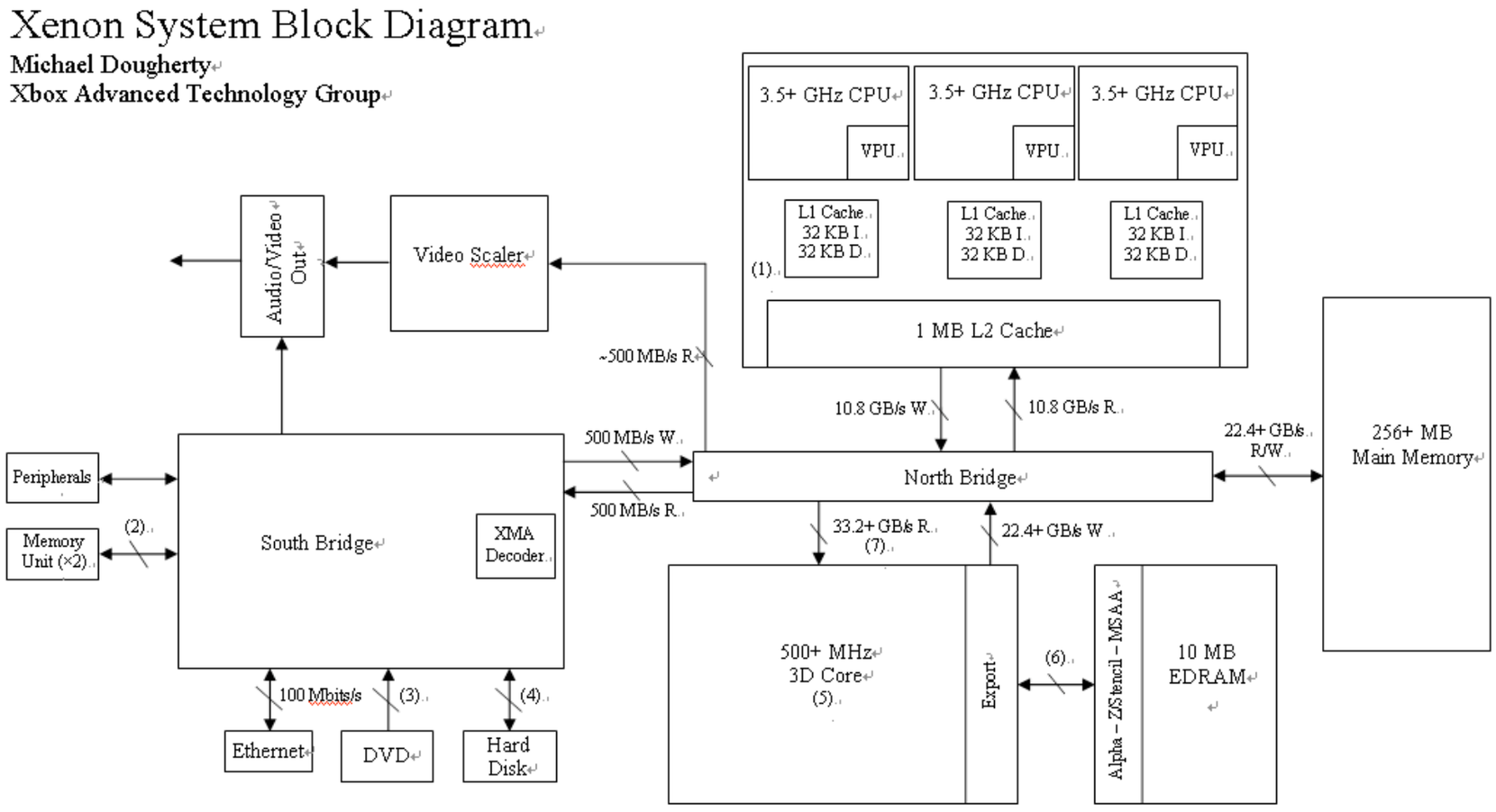
Architectures de contrôle

Architectures de contrôle

- Architecture Matérielle
- Architecture Logicielle

Architectures de contrôle

- Architecture Matérielle :



(1) CPU – Main Memory Latencies
(in CPU cycles)
- ~525 main memory to CPU
- 31 L2 to CPU
- 4 L1 I to CPU
- 5 L1 D to CPU

* L1-to-CPU latency is usually hidden by the pipelines.

(2) Memory Units
- 64 MB minimum capacity
- 8 MB/s R
- 2.5 MB/s W

* cannot read and write simultaneously

(3) DVD
- 9.4 GB capacity
- ~7 – 16.6 MB/s R (track dependent)
- ~115 ms avg. – 180-ms typical max access time (seek distance and rotational latency dependent)

(4) Hard Disk ("built in" not decided)
- 20 – 80 GB capacity
- ~15 – 30 MB/s R/W (sector dependent)
- ~13 ms avg. – 30 ms typical max access time (seek distance and rotational latency dependent)

(5) GPU Performance Overview
- 48 ALU ops per cycle
- 16 bilinear texture fetches per cycle
- 16 PS input interpolates per cycle
- VS and PS load balanced
- Max throughputs per cycle:
  - 1 vertex
  - 1 triangle
  - 2 2×2 pixel quads + Z/Stencil

* predictable streaming and multi-threading make the GPU a latency-tolerant device

(6) EDRAM R/W Bandwidth
(in 2×2 pixel quads per GPU cycle)

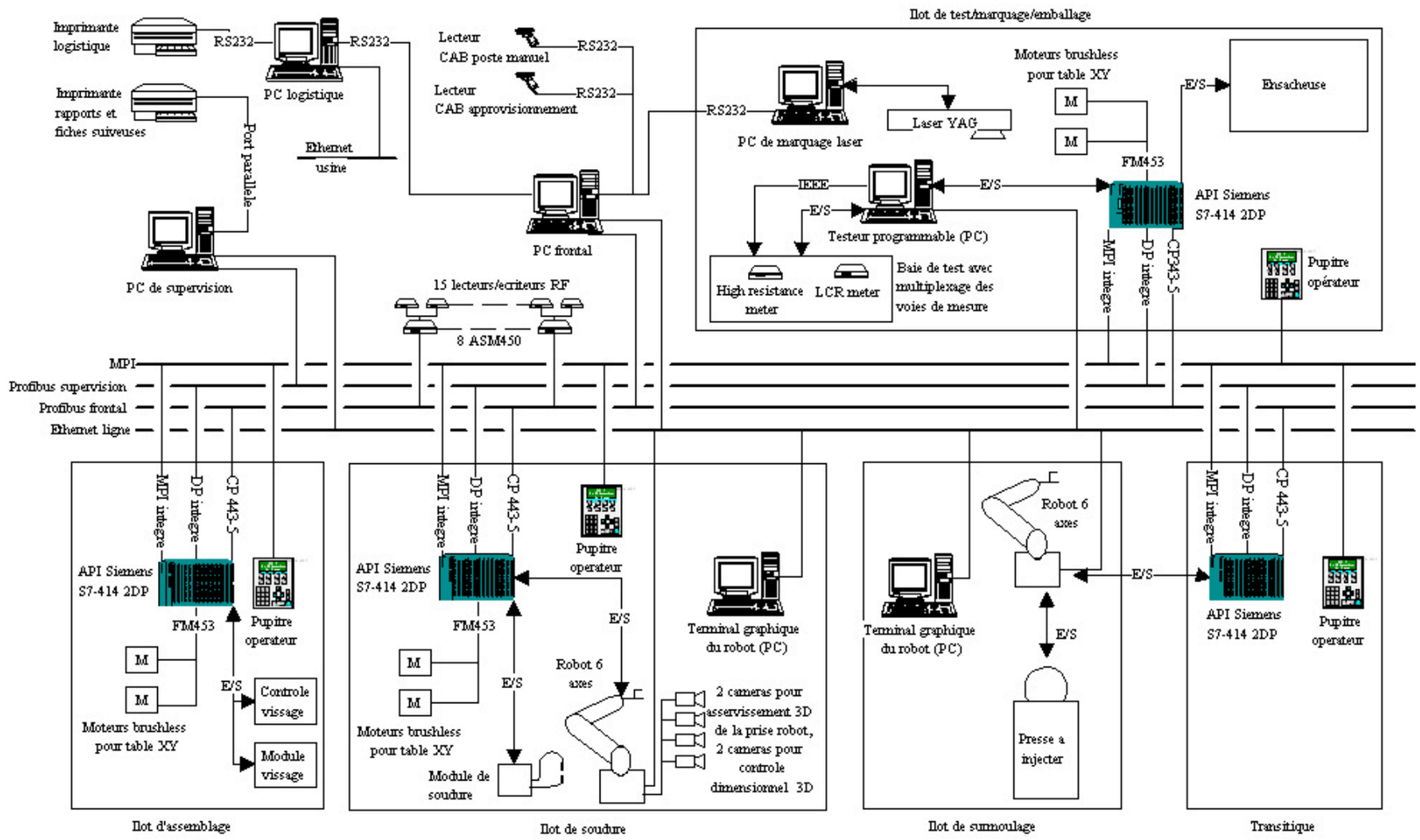
Write:	Read (resolve):
• 2 at 32 bpp + Z/Stencil	• 2 at 32 bpp
• 1 at 64 bpp +Z/Stencil	• 1 at 64 bpp
• 4 Z/Stencil only	• 2 Z/Stencil only

* Z/Stencil testing, alpha blending, and MSAA down-sampling are done in the EDRAM module

(7) 10.8 GB/s from L2 + 22.4+ GB/s from main memory peak bandwidth

Architectures de contrôle

- Architecture Matérielle :



Architectures de contrôle

- Architecture Matérielle : les éléments matériels de notre système :
  - des unites de calcul (processeurs, microcontrolleurs)
  - des connecteurs
  - des capteurs (température, IR, telemetres, pression, gyroscopes, etc...)
  - des effecteurs (moteurs)

- Forte hétérogénéité :

- OS différents ?
- connections différentes ?
  - quel bus ?
  - quel protocole de communication -> reseaux de terrains
- Quelles fréquences de fonctionnement ?
  - quelles méthodes de synchronisation ? -> temps réel
- Certains composants sont livrés avec un support, d'autre non

Architectures de contrôle

- Exemple : Réaliser un vêtement qui mesure les activités quotidiennes

Architectures de contrôle

- Exemple : Réaliser un vêtement qui mesure les activités quotidiennes

exemple au mediaLab (MIT)

Architectures de contrôle

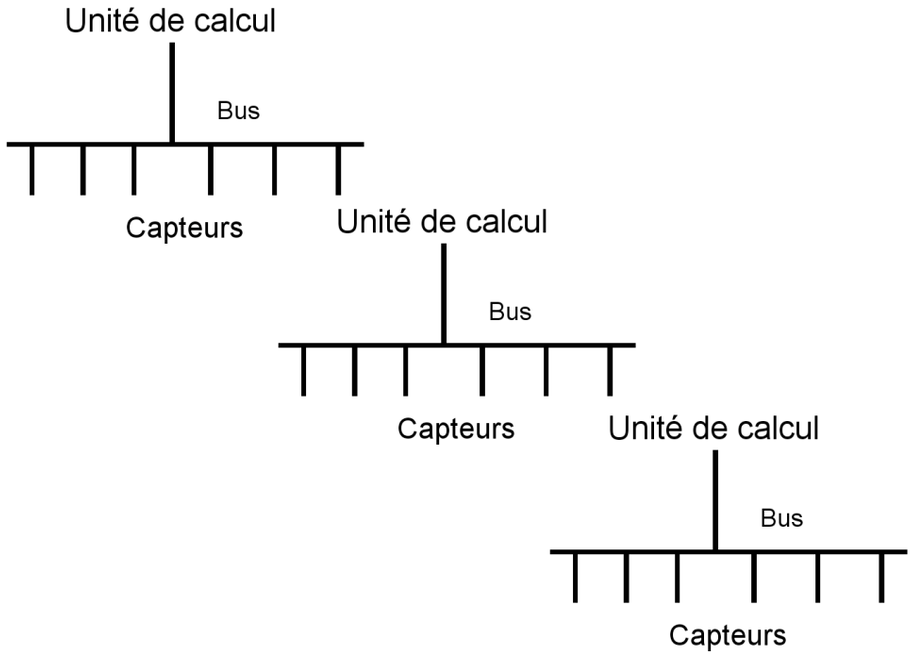
- Exemple : Réaliser un vêtement qui mesure les activités quotidiennes

Architectures de contrôle

- Exemple : Réaliser un vêtement qui mesure les activités quotidiennes

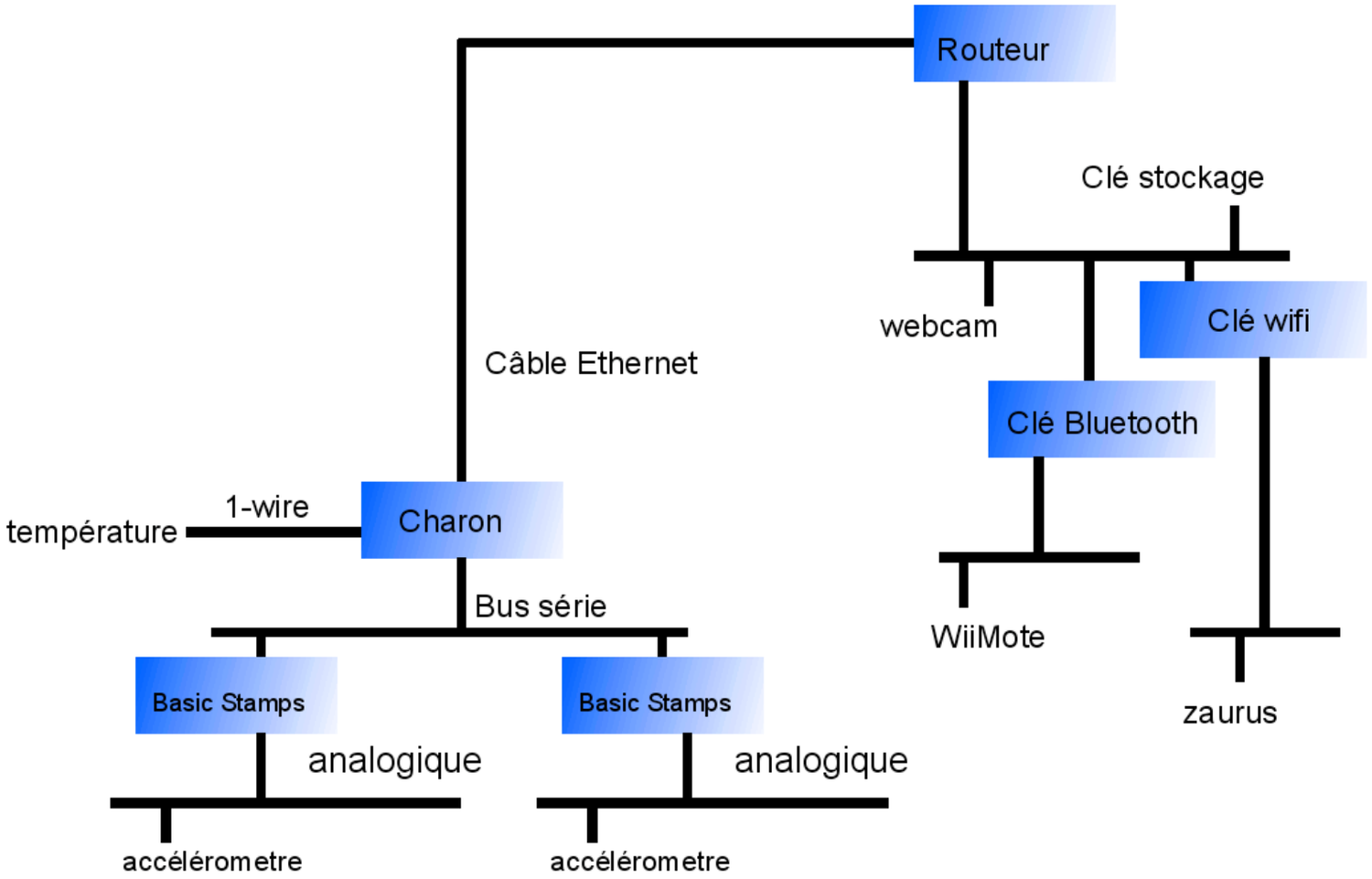
Architectures de contrôle

- Exemple : Architceture idéale :



Architectures de contrôle

- Exemple : Bus qui prennent en compte la réalité des composants : hétérogénéité



Architectures de contrôle

• Architecture logicielle : implementer, sur chaque calculateur (puce/processeur) le nécessaire pour communiquer, se synchroniser :

- Son
  - Micro WebCam
  - 1 canal – 8000hz
- Vidéo
  - WebCam
  - 160x120 RGB
  - 20 fps
- WiFI
  - Clé USB dwl-122
  - 11mbit/s
- Accéléromètres Analogiques
  - Memsic Mx2125
  - 2 axes – 30Hz
- Accéléromètres Bluetooth
  - Wiimote et Nunchunk ©
  - 3 axes – 100Hz
- Température
  - Dallas 1-Wire DS18S22

Architectures de contrôle

- Attention au temps réel !

Un système temps réel est défini comme un système dont le comportement dépend :

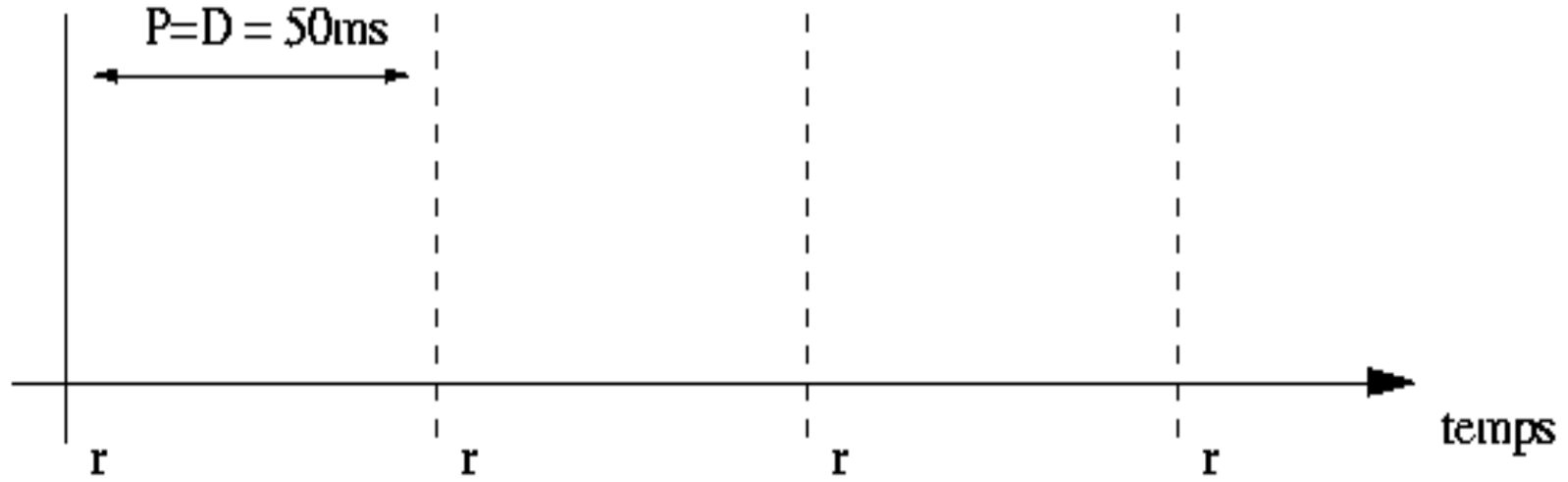
- De l'exactitude des traitements effectuées

- Du temps où les résultats sont produits

Un retard = le fait de rater une échéance = erreur du système.

Architectures de contrôle

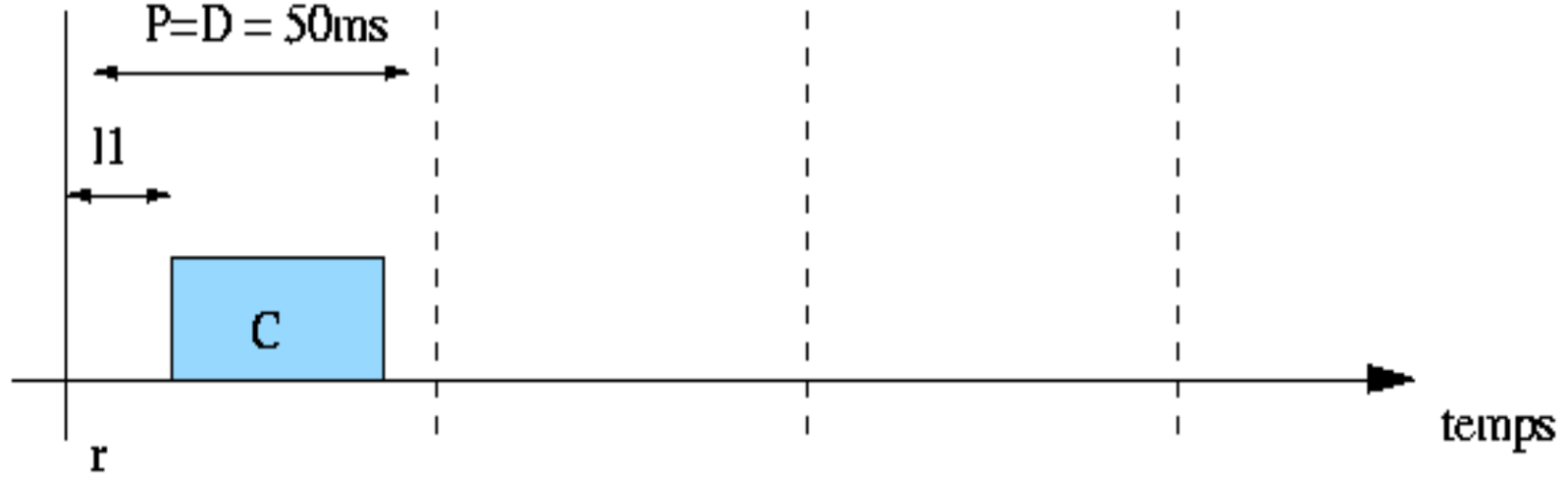
- Attention au temps réel !



P : période
D : échéance
C : Temps de traitement
r : réveil de la tache

Architectures de contrôle

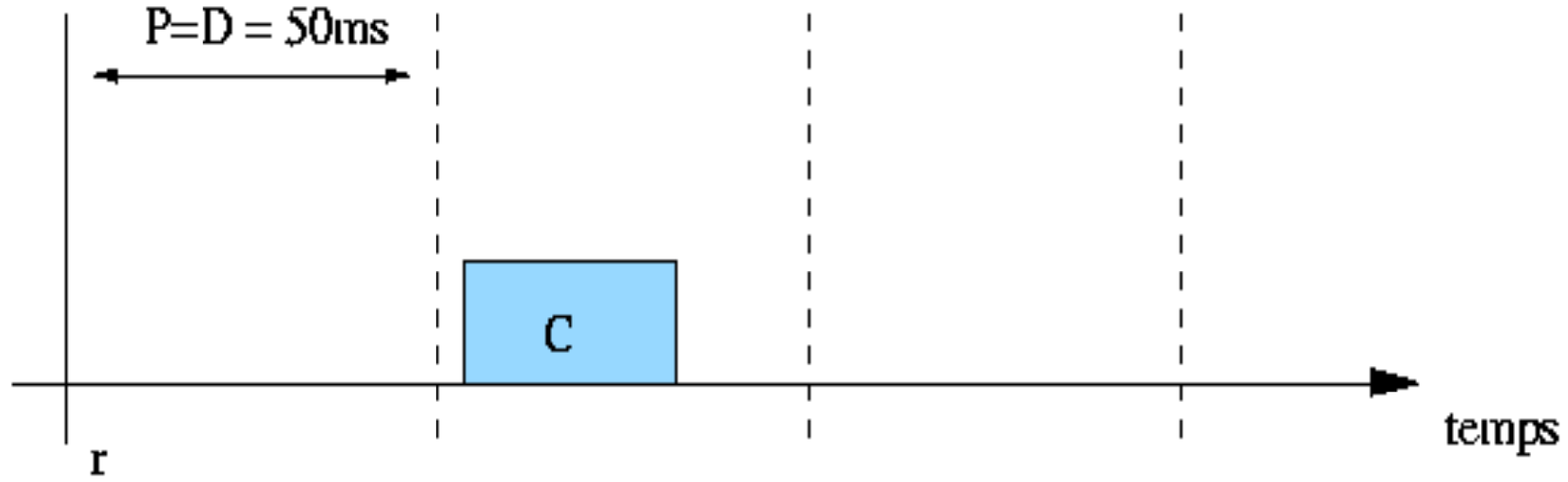
- Attention au temps réel !



P : période
D : échéance
C : Temps de traitement
r : réveil de la tache
l : latence du système

Architectures de contrôle

- Attention au temps réel !

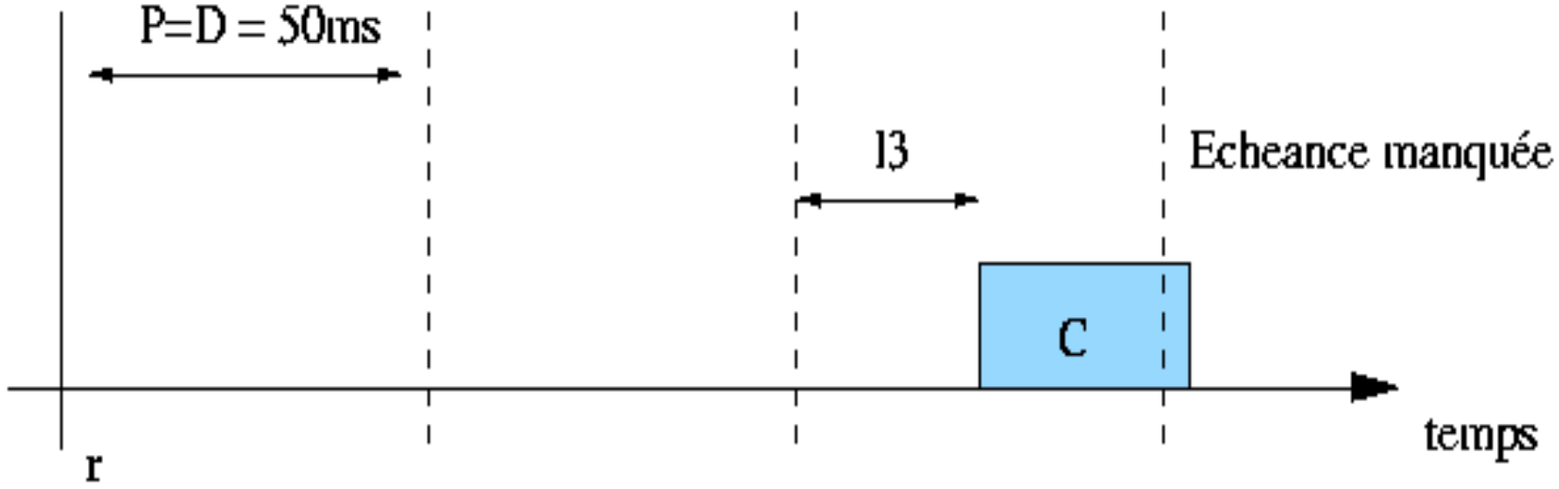


P : période
D : échéance
C : Temps de traitement
r : réveil de la tache

l : latence du système

Architectures de contrôle

- Attention au temps réel !



P : période
D : échéance
C : Temps de traitement
r : réveil de la tache
l : latence du système

Architectures de contrôle

- Attention au temps réel !

  - Temps réel dur et lâche :
    - Si le retard d' un traitement = exception (traitement spécifique en cas d'erreur) : on parle d'échéance dure, et il s'agit d'une *défaillance*
    - Si le retard ne provoque pas d'exception: échéance lâche
    - Un système dont aucune échéance ne doit être dépassée: Temps réel dur (*hard real time*)
    - Si un dépassement occasionnel ne met pas en danger le système : temps réel lâche, ou mou (*soft real time*)

Architectures de contrôle

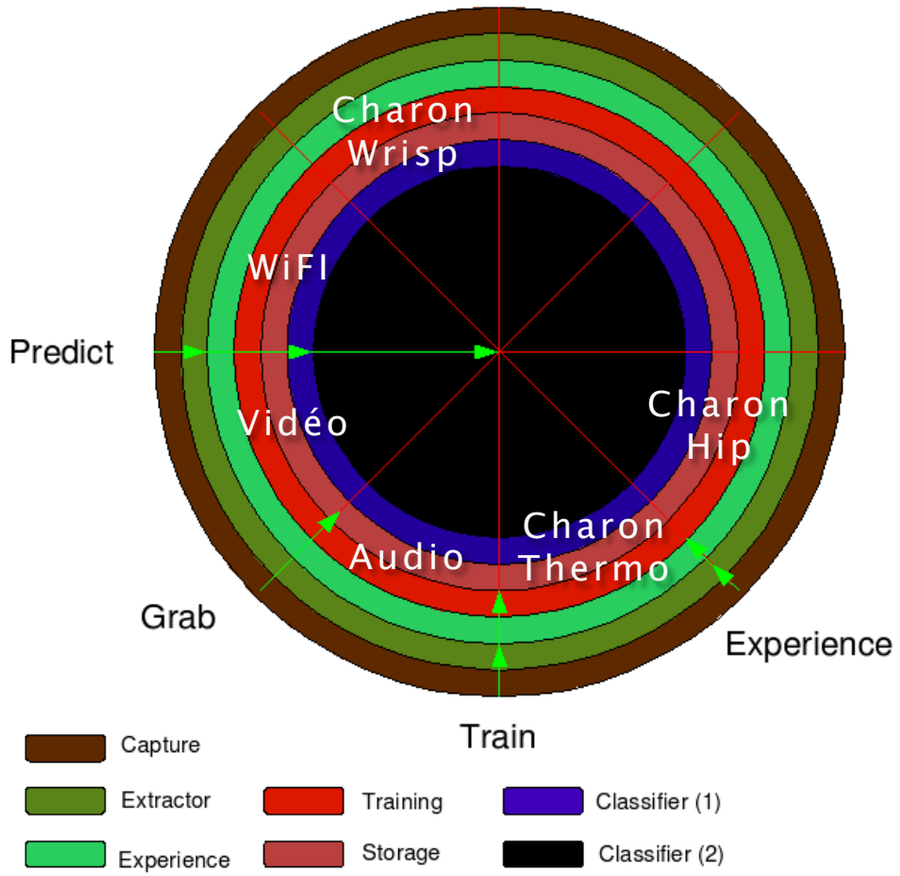
• Architecture logicielle : retrouver un mode de fonctionnement qui fait abstraction (autant que possible) des contraintes matérielles :

- Assurer la communication entre l'application et les différents éléments
  - lecture
  - écriture
  - prise de décision
  - sauvegarde

- architecture centralisée ?
- architecture distribuée ?

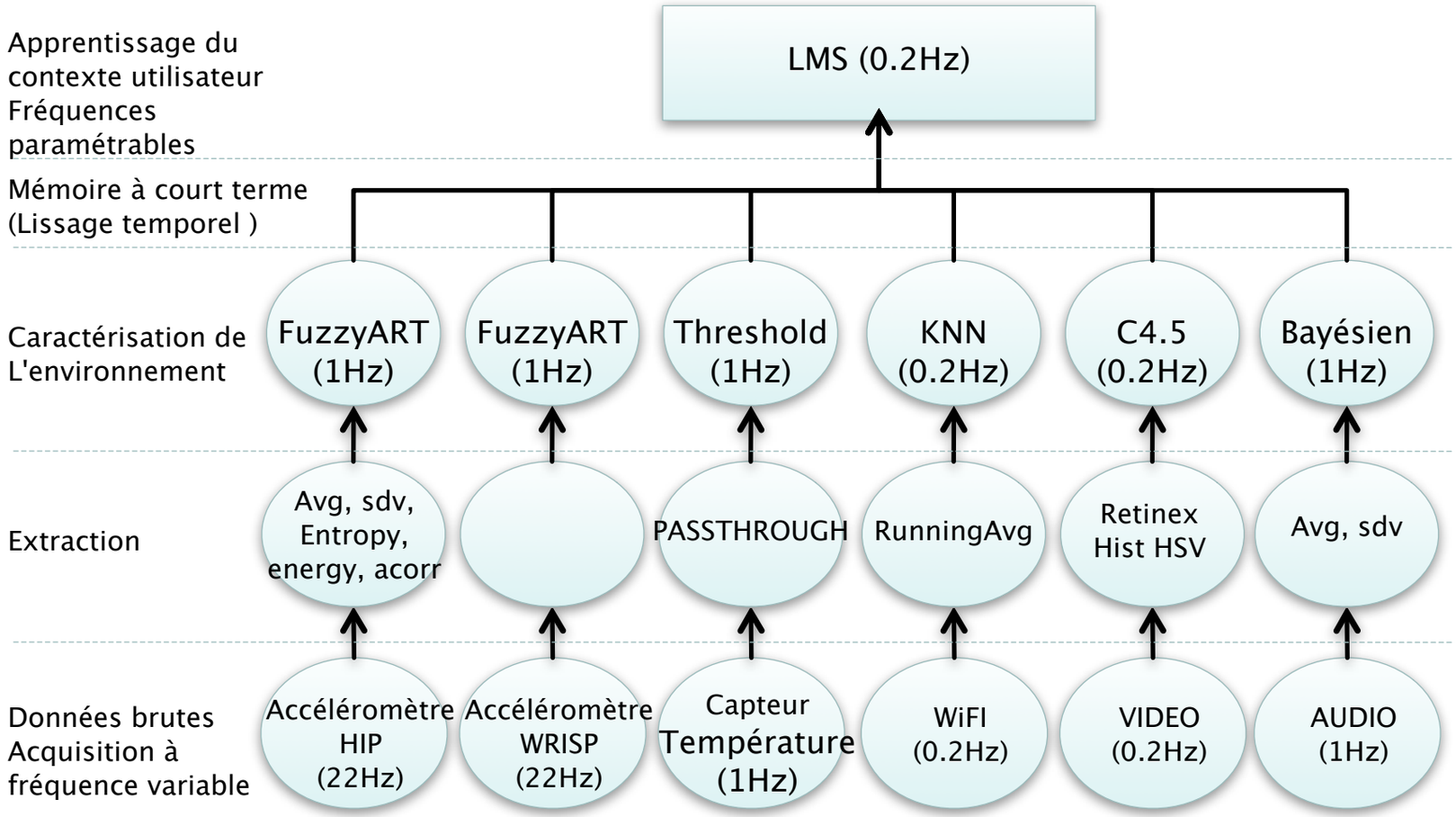
Architectures de contrôle

• Architecture logicielle : retrouver un mode de fonctionnement qui fait abstraction (autant que possible) des contraintes matérielles : les OS sont un bon exemple !



Architectures de contrôle

- Architecture intelligente :
  - Détection d'événements
  - Classification (supervisée, non supervisée)
  - Apprentissage
  - Reconnaissance
  - Prédiction
  - Adaptation



Architectures de contrôle

- Réalisation : les grandes étapes
  - Evaluation des objectifs
  - Liste du matériel nécessaire : capteurs, actionneurs, unités de calcul
  - Documentation : BIBLIOGRAPHIE !!! (pilotes, compatibilité, programmation, etc)
  - Identification des bus / types de communication (serie, usb, eth, mixte) / mise au point des protocoles de communication si nécssaire.
  - Schémas de l' Architecture Matérielle, avec évaluation des débits, fréquences de fonctionnement, etc.
  - Schémas de l'architecture logicielle (quels programmes, et où ?)

Architectures de contrôle

- Réalisation : les grandes étapes
  - Evaluation des objectifs
  - Liste du matériel nécessaire : capteurs, actionneurs, unités de calcul
  - Documentation : BIBLIOGRAPHIE !!! (pilotes, compatibilité, programmation, etc)
  - Identification des bus / types de communication (serie, usb, eth, mixte) / mise au point des protocoles de communication si nécssaire.
  - Schémas de l' Architecture Matérielle, avec évaluation des débits, fréquences de fonctionnement, etc.
  - Schémas de l'architecture logicielle (quels programmes, et où ?)

Architectures de contrôle

Développement unitaires, et tests unitaires des "services de base", chaque composant **indépendamment**
- lecture/envois d'une donnée par un capteur
- réception par un calculateur
- écriture/exécution d'une consigne par un effecteur

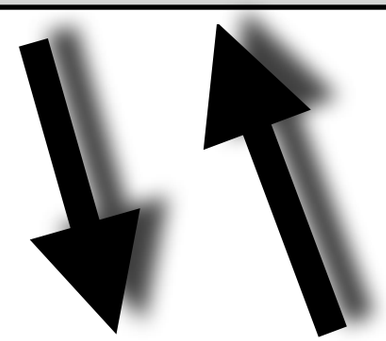
Architectures de contrôle

Développement unitaires, et tests unitaires des "services de base", chaque composant **indépendamment**
- lecture/envois d'une donnée par un capteur
- réception par un calculateur
- écriture/exécution d'une consigne par un effecteur

Integration dans l'architecture globale
- chaque composant peut envoyer/recevoir l'info
- test, vérification et ajustement des débits et fréquences max

Architectures de contrôle

Développement unitaires, et tests unitaires des "services de base", chaque composant **indépendamment**
- lecture/envois d'une donnée par un capteur
- réception par un calculateur
- écriture/exécution d'une consigne par un effecteur

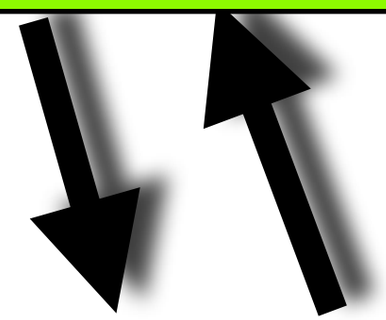


SUR MACHINE

Integration dans l'architecture globale
- chaque composant peut envoyer/recevoir l'info
- test, vérification et ajustement des débits et fréquences max

Architectures de contrôle

Développement des applications sur chaque composant **indépendamment SI POSSIBLE**
- par eemple lecture + pré-traitement



SUR MACHINE

Intégration dans l'architecture globale
-On suit les fonctionnalités de l'architecture logicielle : chaque info alimente les principaux process de l'architecture

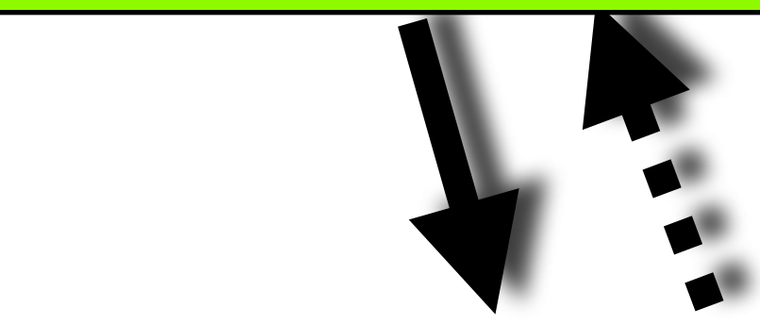
Architectures de contrôle

Développement des applications sur chaque composant **indépendamment SI POSSIBLE**
- par exemple lecture + pré-traitement

Intégration dans l'architecture globale
-On suit les fonctionnalités de l'architecture logicielle : chaque info alimente les principaux process de l'architecture

Architectures de contrôle

Développement des applications sur chaque composant **indépendamment SI POSSIBLE**
- par exemple lecture + pré-traitement



SUR MACHINE

Architceture intelligente :
le tout est plus que la somme des parties
fusion des informations (categorisation)
debug sur systeme complet nécessaire
bonne fiabilité et robustesse des composants