Ambient Intelligence (AmI)

Frédéric Le Mouël

University of Lyon
INRIA / INSA Lyon

Who am I ? – FLM

- Frédéric Le Mouël (frederic.le-mouel@insa-lyon.fr)
  - Associate Professor – INSA Lyon
  - Researcher – INRIA Rhône Alpes
  - http://perso.citi.insa-lyon.fr/flemouel/
- Research at CITI Laboratory / INRIA ARES Team
  - Middleware, Component- and Service-Oriented Programming and Architectures, Pervasive Systems, Ambient Intelligence, Adaptation
- Teaching at Telecommunication Department
  - Modeling and Software Engineering, Middleware, Object-Oriented Programming, Java, Dynamic Web, Pervasive Systems

Plan

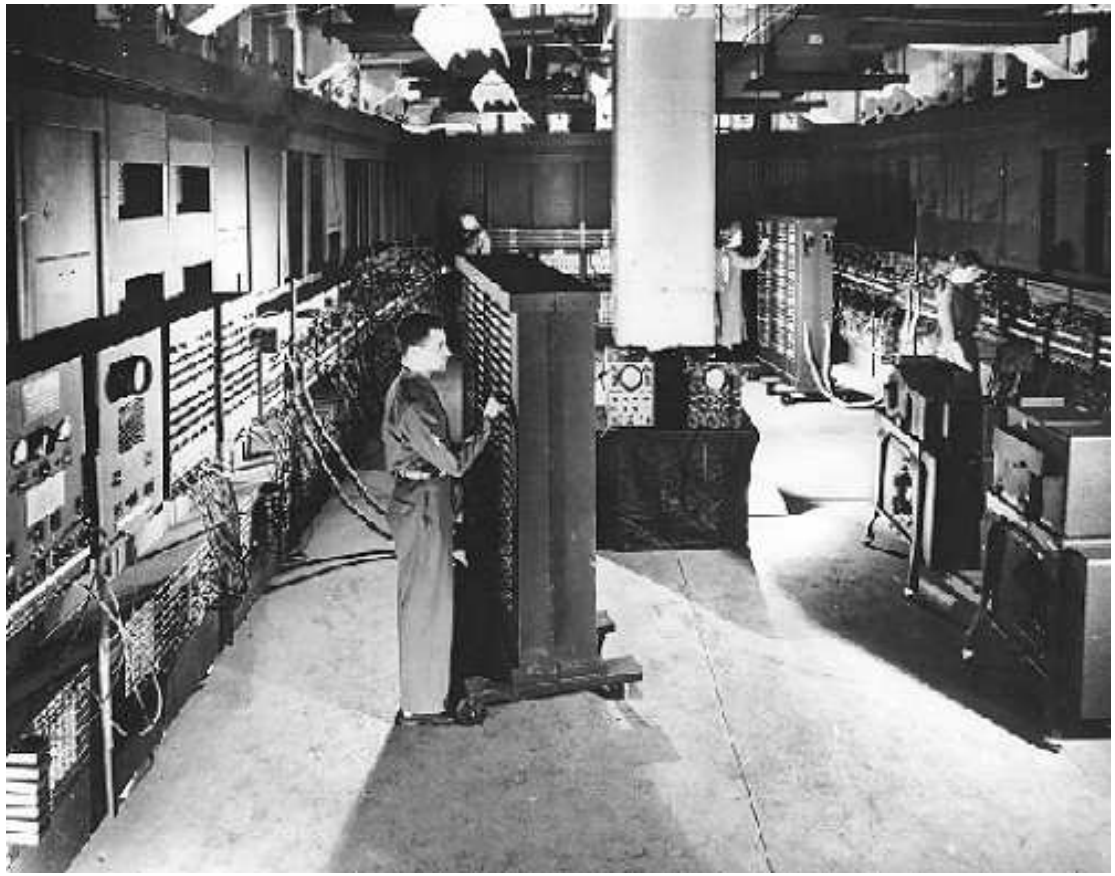
- New IT Vision
- From Pervasive/Ubiquitous Systems to Ambient Intelligence
- Definitions and Properties
- Functional Architecture

New IT Vision

New vision for Information Technologies

- Working Environment
  - Before: a virtual environment where you log in, execute applications and then log out
  - Now: a physical environment where you are always connected to execute tasks

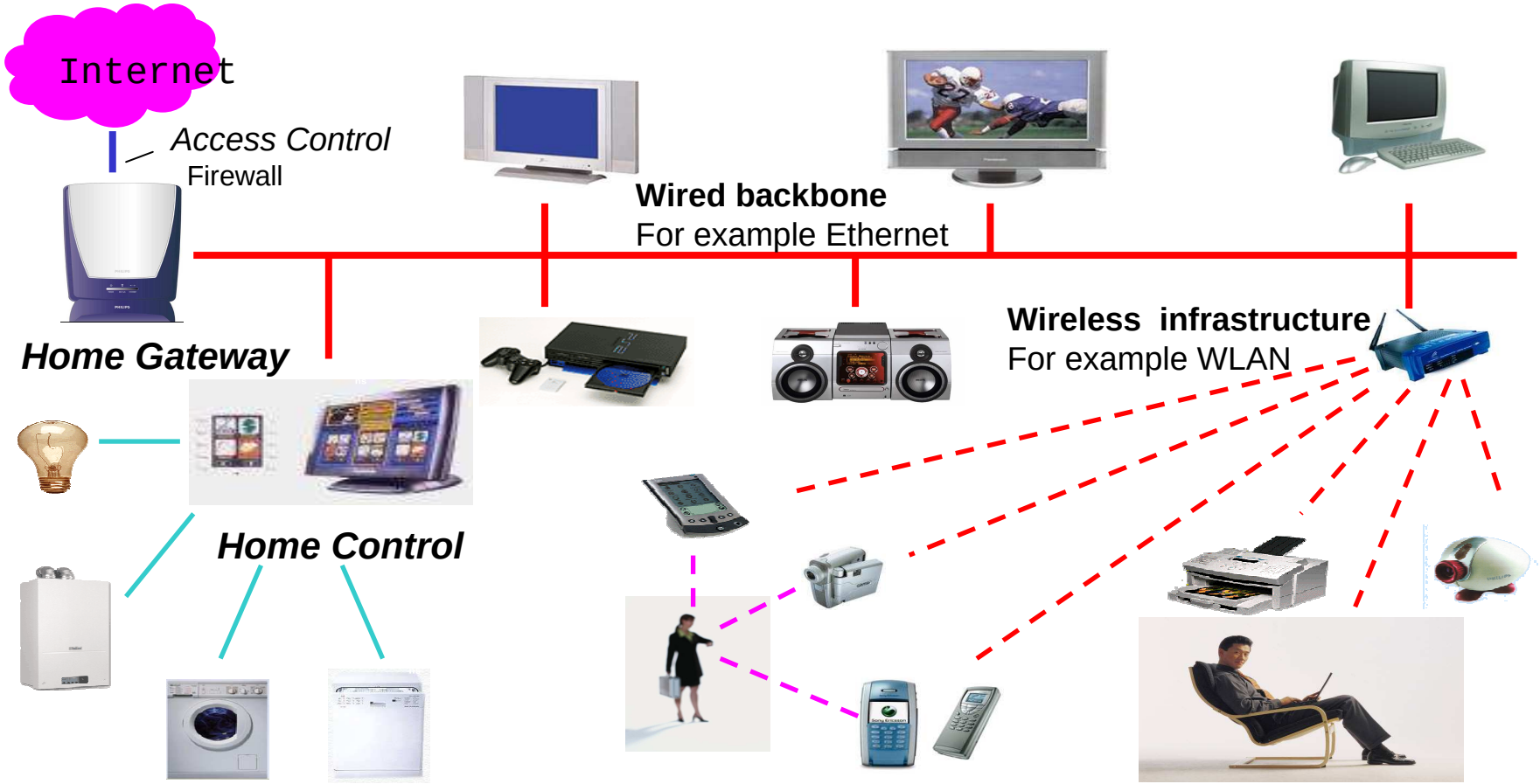
Before:
Room full with a computer



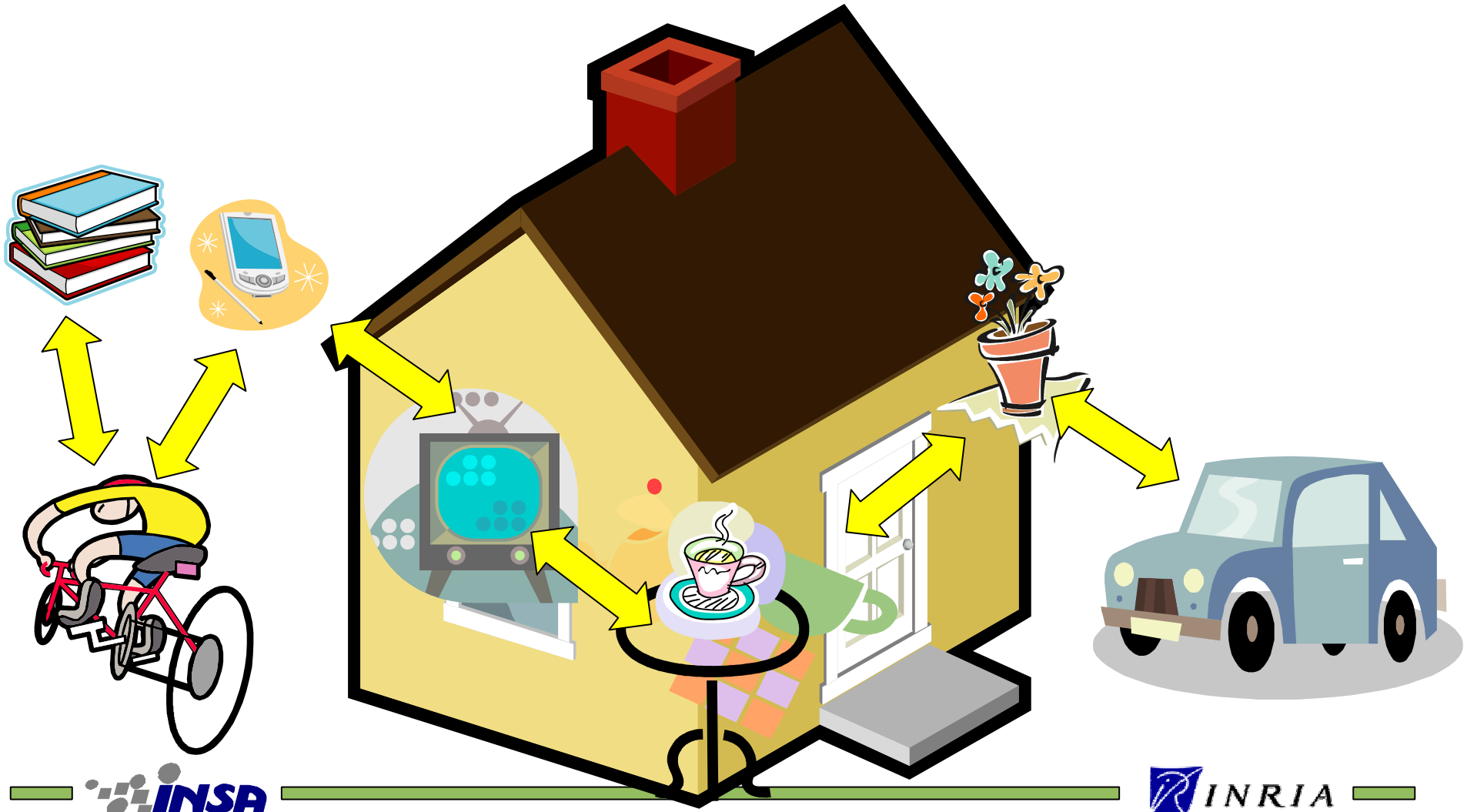
Electronic
Numerical
Integrator
and Computer
(ENIAC)
1946

© Computer Science History

Now: Room with several visible devices



Tomorrow: Everyday life full of invisible appliances



Motivation

- Today
  - Computers
  - Internet wired connection

- Tomorrow
  - Every object will be smart (Embedded processors + memory)
  - Wireless connection (802.11*, Bluetooth, etc. + Internet New Generation, IPv6)

From Pervasive/Ubiquitous Systems to Ambient Intelligence

Pervasive Environments

- [M. Weiser, 1991]
  - « *A new way of thinking about computers in the world, one that takes into account the natural human environment and allows the computers themselves to vanish in the background* »
  - « *The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are* indistinguishable from it »

Pervasive Environments

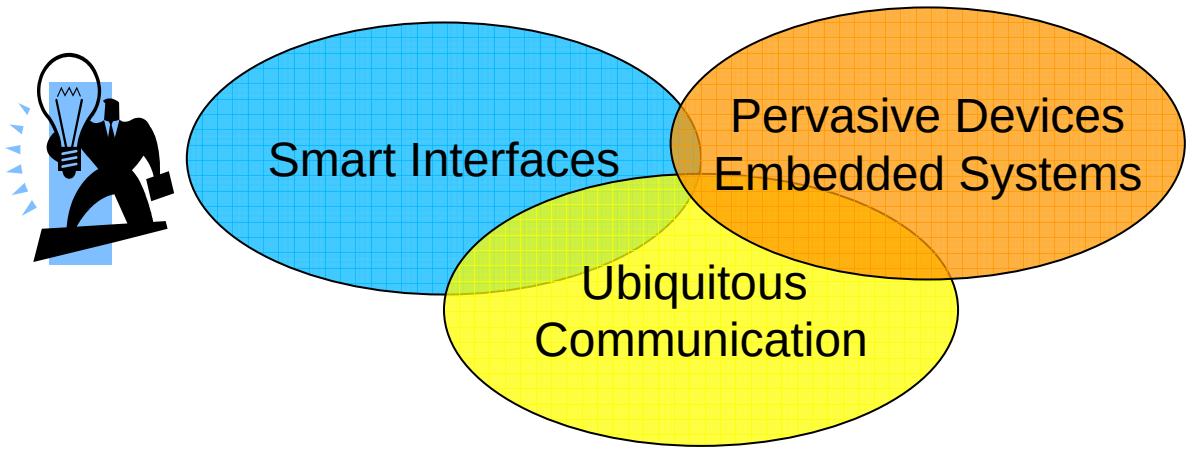
- [M. Satyanarayanan, 2001]
  - « *One saturated with computing and communication capability, yet so gracefully integrated with users that it becomes 'a technology that disappears'* »

Pervasive Environments

- [NIST, 2001]
  - « *Pervasive computing is a term for the strongly emerging trend toward: numerous, casually accessible, often invisible computing devices, frequently mobile or embedded in the environment, connected to an increasingly ubiquitous network infrastructure, composed of a wired core and wireless edges* »

Pervasive Sub-topics

- Interconnection of 3 technological domains:
  - Smart Interfaces
  - Pervasive Devices, Embedded Systems
  - Ubiquitous Communication, Connectivity

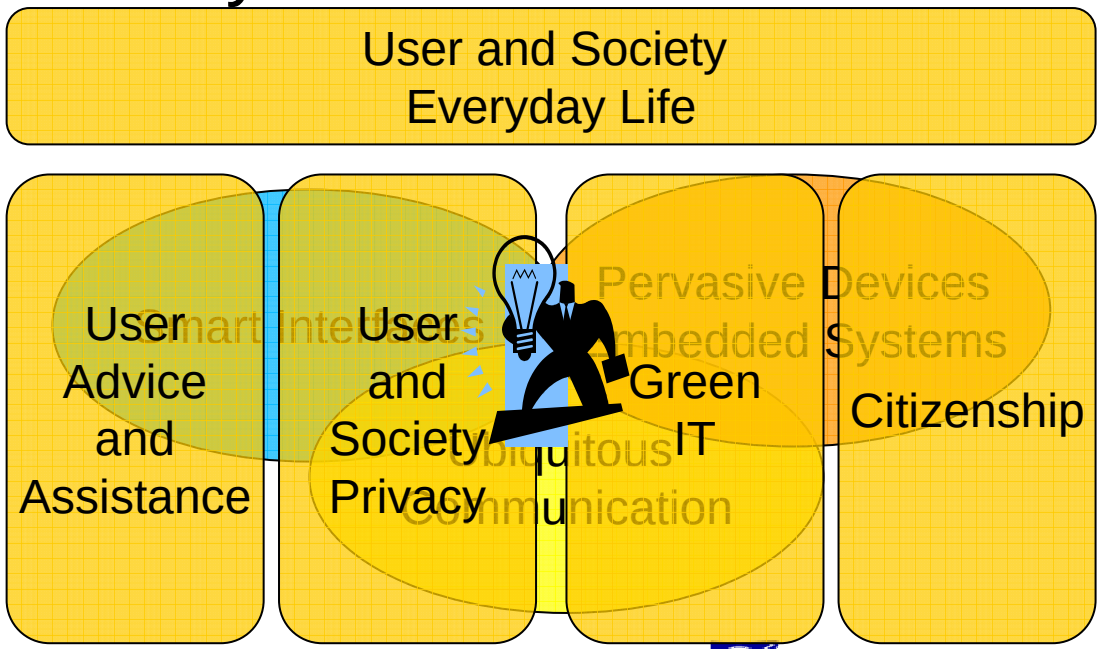


Ambient Intelligence

- [ISTAG 2001]
  - « *The concept of Ambient Intelligence (AmI) provides a vision of the Information Society where the emphasis is on greater user-friendliness, more efficient services support, user-empowerment, and support for human interactions. People are surrounded by intelligent intuitive interfaces that are embedded in all kinds of objects and an environment that is capable of recognising and responding to the presence of different individuals in a seamless, unobtrusive and often invisible way.* »

Ambient Intelligence Interests

- User and Society rules:
  - User Advice and Assistance
  - User and Society Privacy
  - Citizenship
  - Green IT



Definitions and Properties

Definitions

- Ubiquitous
  - Accessible from everywhere
- Mobile
  - Which integrates mobile devices
- Context-awareness
  - Which takes into account the execution environment
- Pervasive
  - Which associates ubiquitous, mobility and context-awareness
- Ambient
  - Which integrates a (pro)active and user-oriented assistance in user everydaylife objects and environment

AmI Properties

- Scalability
- Invisibility
- Context-awareness
- Smartness
- Pro-action

Scalability

- Management of a great amount of
  - Devices
  - Applications
  - Users
- Performance

Example: Web Server scalable ?

- Development of systems, middlewares, models, applications that are independent and can resist to a high number of devices, users, etc.

Invisibility

- Transparency for human beings
- Minimal intervention of human beings

- Adaptation to environment changes
- Self-learning

- Example: auto-configuration of gateway

Context-awareness

- Virtual representation of the physical environment
- Perception of changes of the environment

- Environment model
- Environment monitoring

- Examples :
  - User Profile, Application Meta-data, Self-descriptive Devices
  - Temperature, Location Sensors

Smartness

- Smart = showing mental alertness and calculation and resourcefulness [wordreference.com, Merriam-Webster, dictionary.com]
- “Intelligent” use of perceived changes
  - Reaction and/or anticipation model (rules, etc.)
  - Inference motor
- Example: Smart House - Power reduction by switch on/off the lights

Pro-action

- Ability to interact, “disturb” the user in order to suggest a better action

- To balance with invisibility !

- Context, environment evaluation
- Several contexts (past, current, future)
- Disturbance model to evaluate the cost/gain between Invisibility/Pro-action

- Example: Information filtering/classification -> Spam

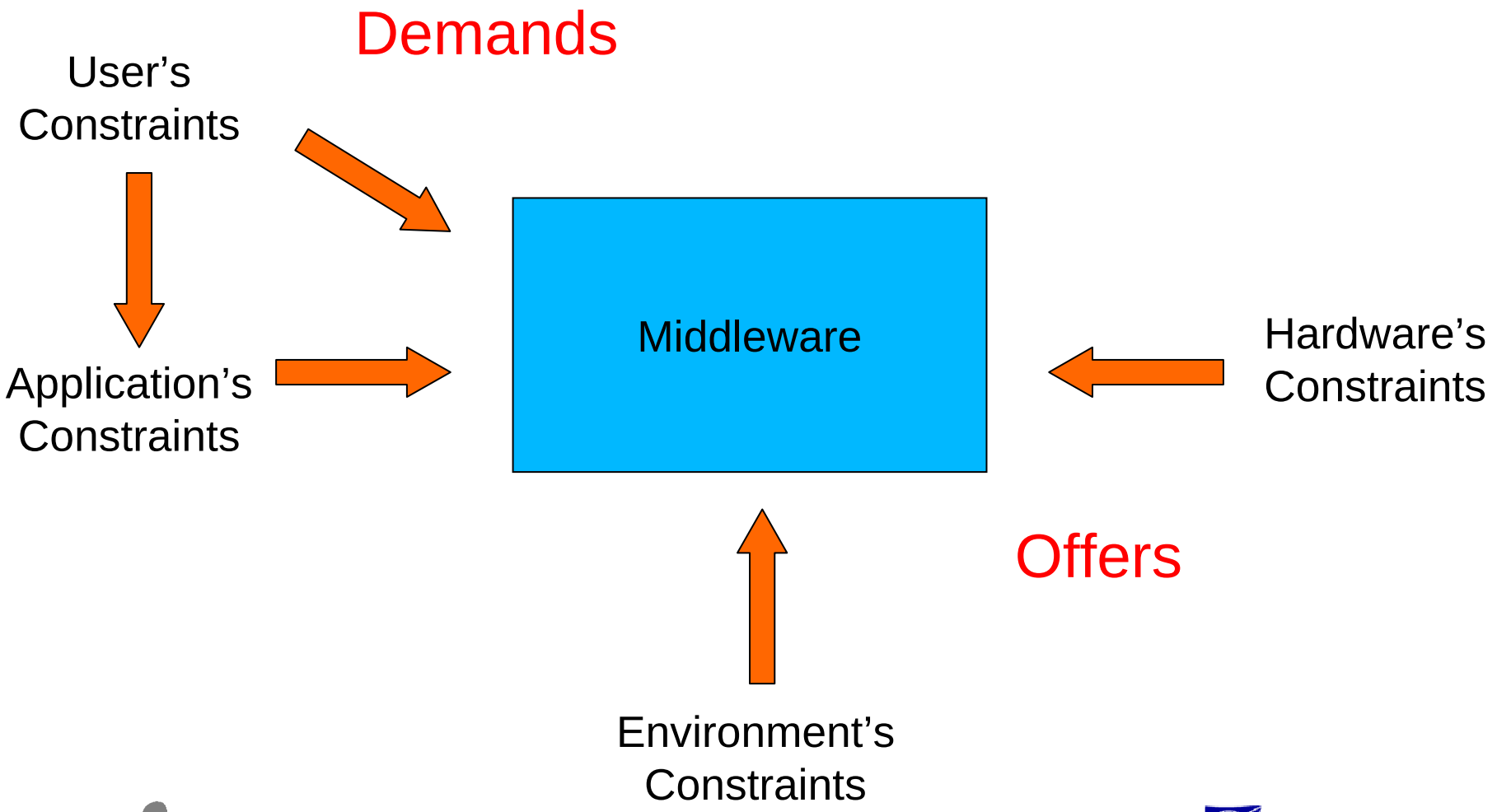
Bibliographie

- M. Weiser « The computer for the twenty-first century », Scientific American, sept 1991:94-104
- M. Satyanarayanan « Pervasive computing : Visions and challenges », IEEE Personal Communications, aug. 2001:10-17
- National Institute of Standards and Technology « Pervasive Computing Program », Pervasive Computing 2001
- D. Saha & A. Mukherjee « Pervasive computing : a paradigm for the 21st century », IEEE Computer journal, march 2003:25-31
- F. Mattern. « Ubiquitous & Pervasive Computing: A Technology-driven Motivation », Summer school on ubiquitous and pervasive computing, 2002
- F. Laforest « Cours Systèmes d'Information Pervasifs », Master Mastria, INSA de Lyon, 2007
- ISTAG « Scenario for Ambient Intelligence in 2010 », 2001

AmI Systems

Functional Architecture

Middleware


Demands
User's Constraints
Application's Constraints
Middleware
Hardware's Constraints
Offers
Environment's Constraints

Hardware's constraints

- Autonomy ⇦ Battery
- Limited resources (CPU, memory, screen, etc.)
  - Power ⇔/ Battery, energy dissipation
  - Capacity ⇦ Weight, size
- Low impact robustness ⇦ Weight, size
- Low security confidence ⇦ Easy access, lost

Environment's constraints

- Mobility / Nomadism
  - Transmission signal
    - Connections / disconnections
    - Variable signal strength

  ⇨ Interferences, cells scope
  - Services availability
    - Different quality of service

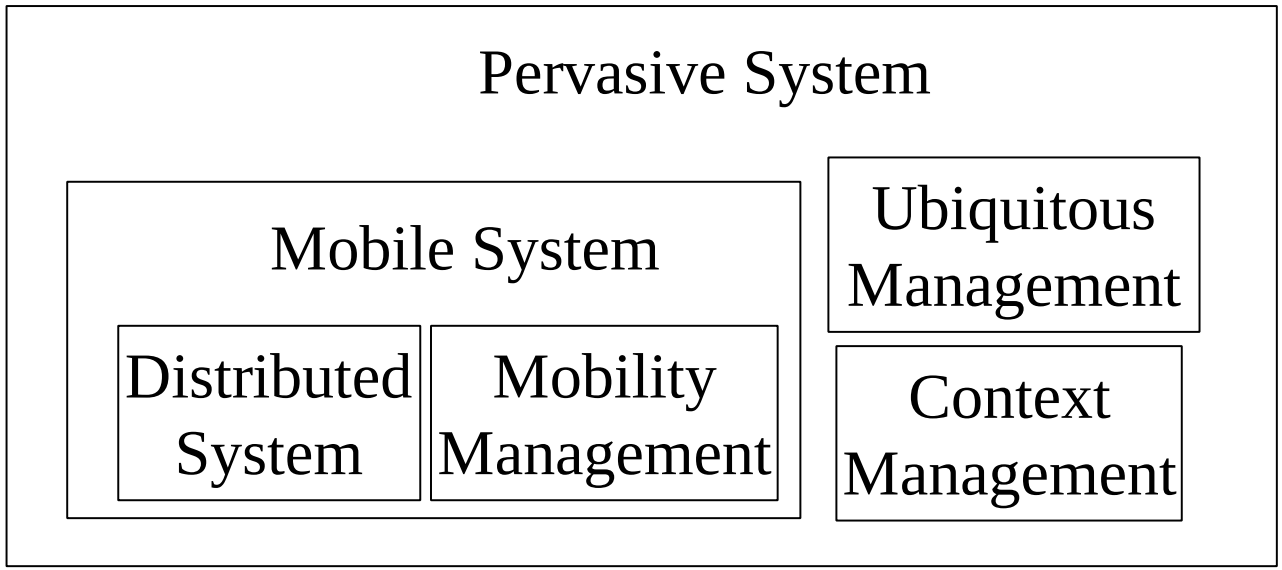
  ⇨ Devices appearance / removal

User and Application's constraints

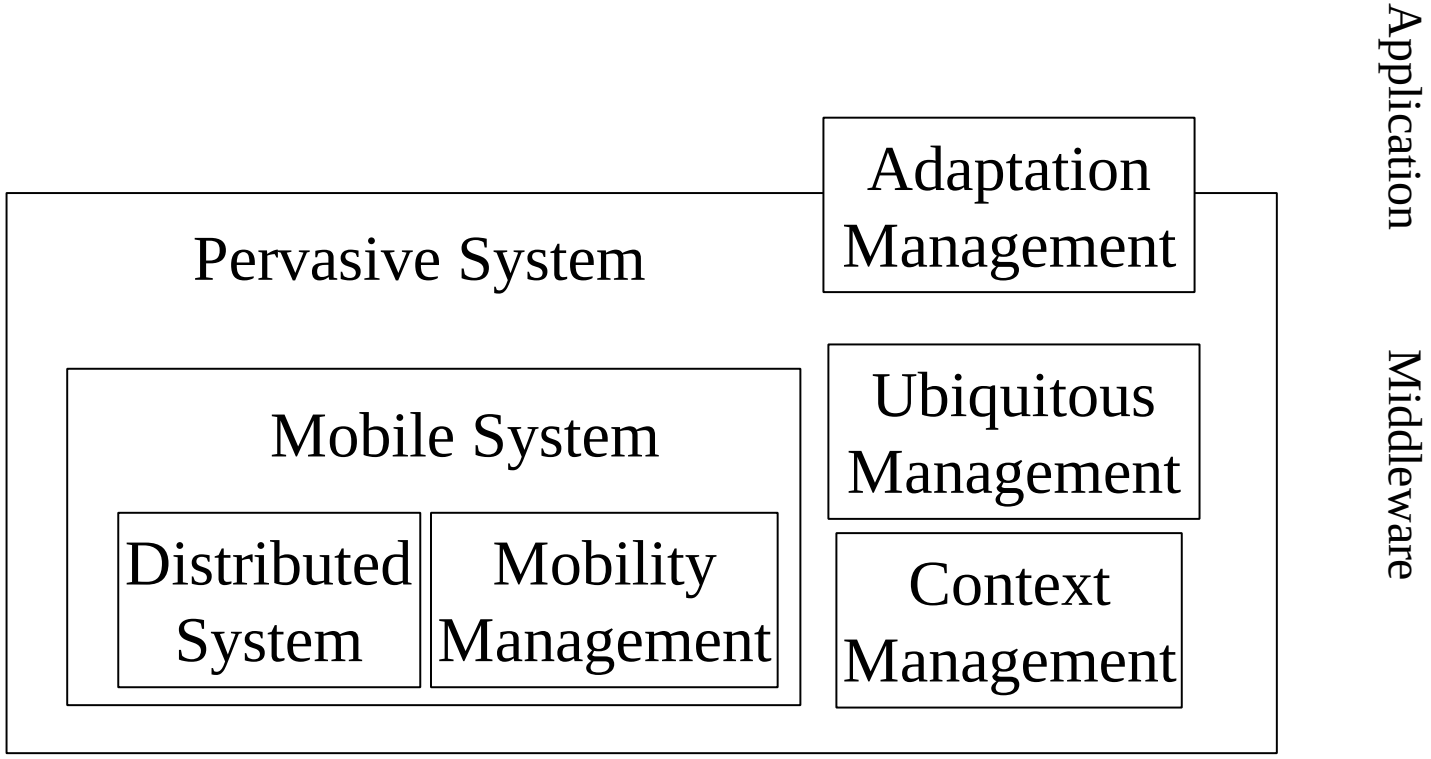
- User's constraints
  - User Profile
  - QoS required
- Application's constraints
  - Power demand
  - Storage demand

Functional view of a Pervasive System

- Adapted from [Saha & Mukherjee, 2003]

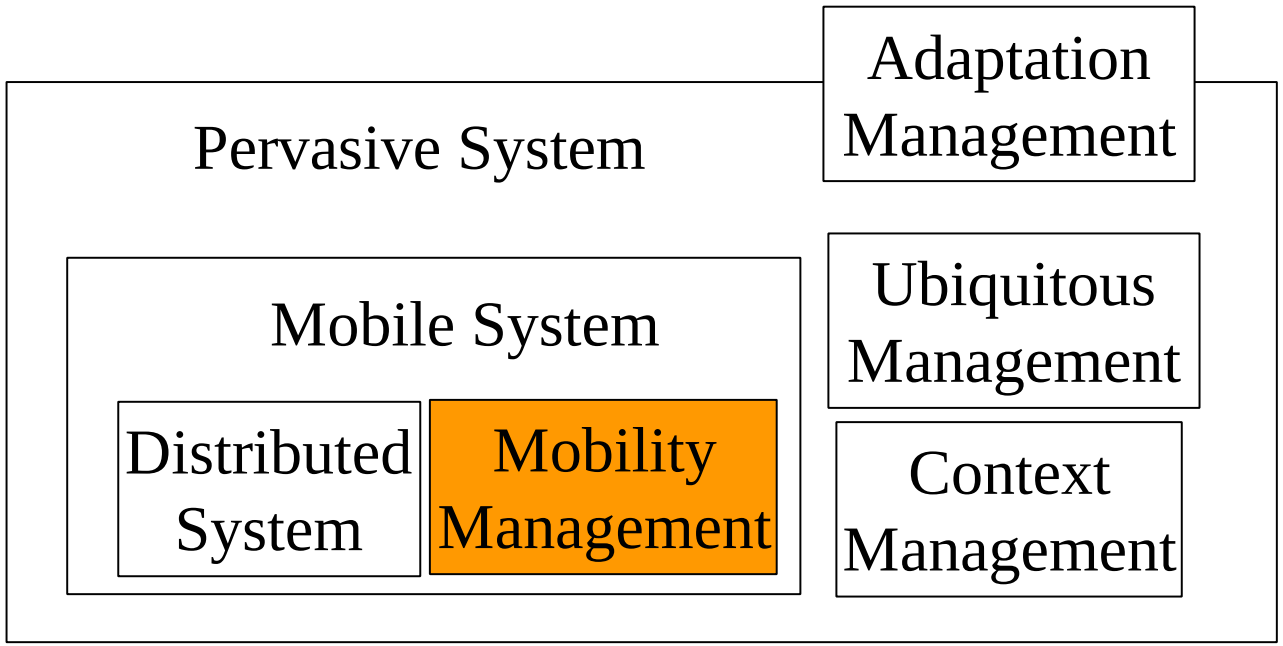


Functional view of an AmI System



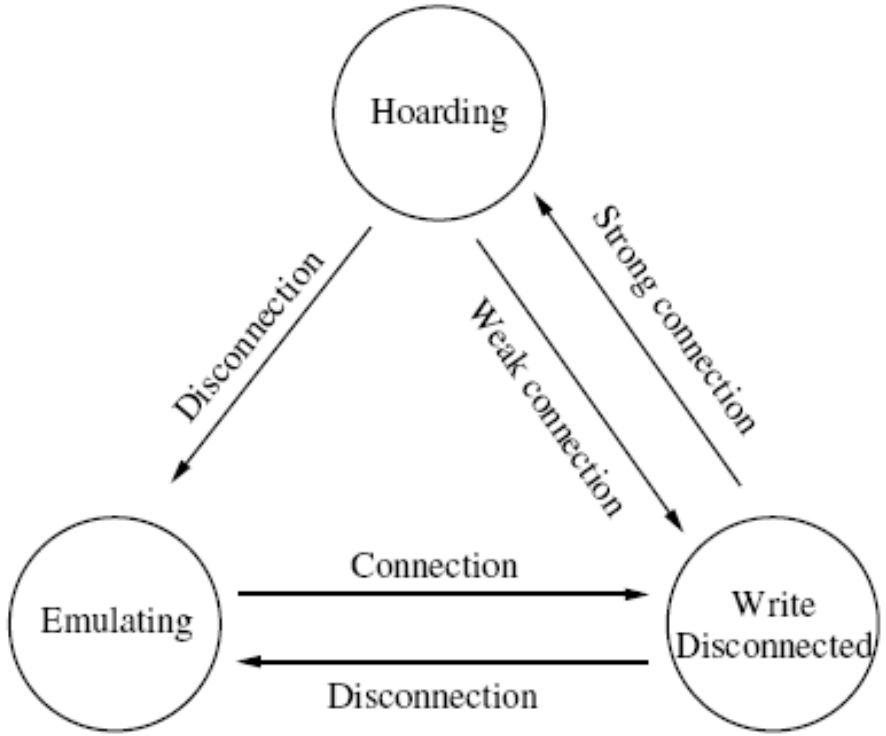
In a real system, all functionalities are interlaced !!!

Functional view of an AmI System



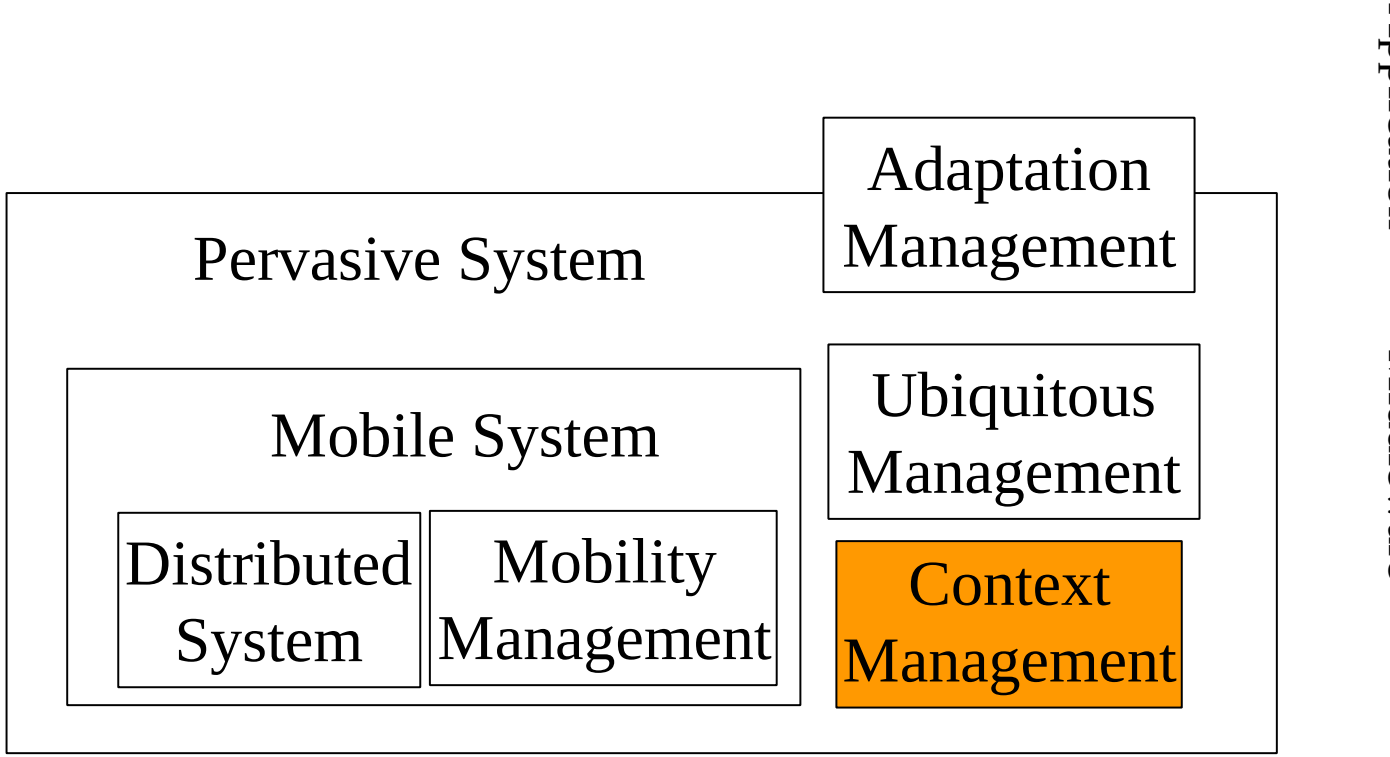
Coda

- File system
  - Hoarding mode (connected)
    - Loading
    - Prefetching
  - Emulating (disconnected)
    - Local work
    - Logs
  - Write disconnected (weak connection)
    - Loading
    - Reconciliation



⇨ **Look ahead, Prefetching**

Functional view of an AmI System



Definition

- [Salber,Dey,Abowd 99]
  - « Environmental information or context covers information that is part of an application's operating environment and that can be sensed by the application. This typically includes the location, identity, activity and state of people, groups and objects. »

Context data

- 4 axes
  - User
    - Profile, preferences, location, etc.
  - Application
    - Size, format, encoding, langue, versions, etc.
  - Hardware
    - Screen size, resolution, color depth, memory, etc.
  - Network
    - Bandwidth, signal strength, etc.

Context modeling

- 3 approaches
  - Attribute/Value
  - CC/PP Extension
  - Ontology

Attribute/Value Pairs

- Context = pairs (attribute, value)
  - User= Toto
  - Localisation = CITI

- Pairs are independent

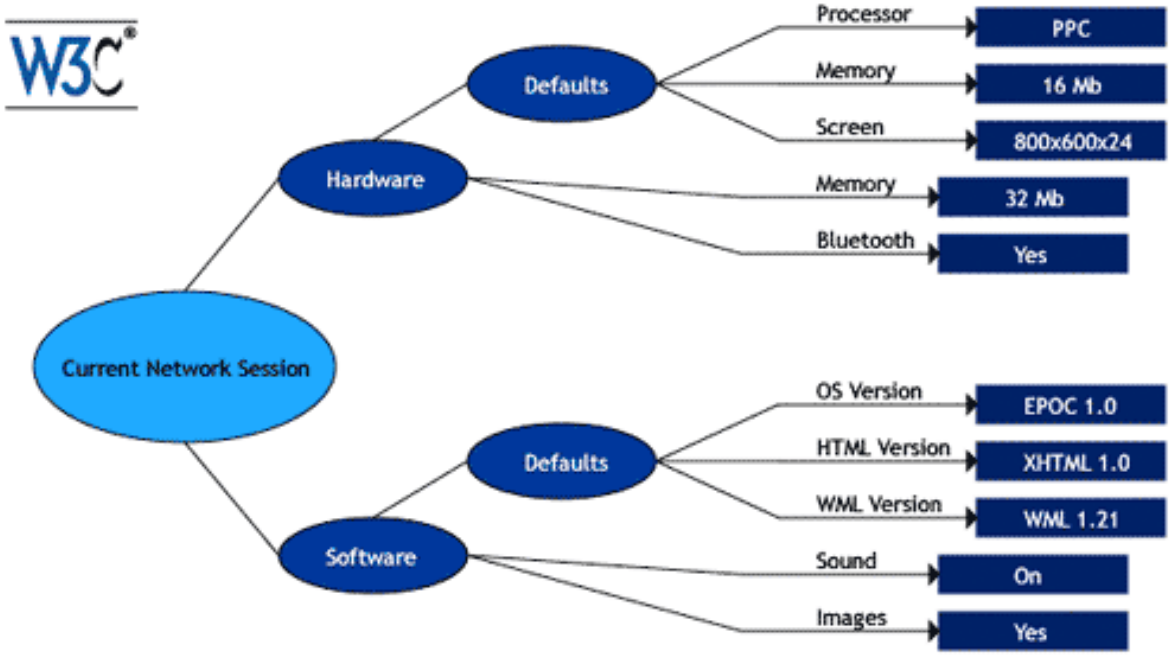
+ Easy

- Consistency

- Poor semantic expressiveness

CC/PP Extensions

- Composite Capabilities / Preferences Profile (W3C)
  - Hardware and User
  - RDF file
  - Context = Extensions proposal



+ Standard

- Extensions =>

  Complex, hard to read

CC/PP example: XML file

```
<?xml version="1.0"?>
<!-- Checked by SiRPAC 1.16, 18-Jan-2001 -->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:ccpp="http://www.w3.org/2000/07/04-ccpp#">
  <rdf:Description rdf:about="HWDefault">
    <rdf:type rdf:resource="HardwarePlatform" />
    <display>320x200</display>
    <memory>16Mb</memory>
  </rdf:Description>
</rdf:RDF>
```

Ontology

- Model of
  - Class
  - Classes relationships
  - Instances

+ Semantically expressive

+ Large-scale environments

- Complex, ontology matching

Ontology example

- Exemple CoOL [Strang & al. 03]

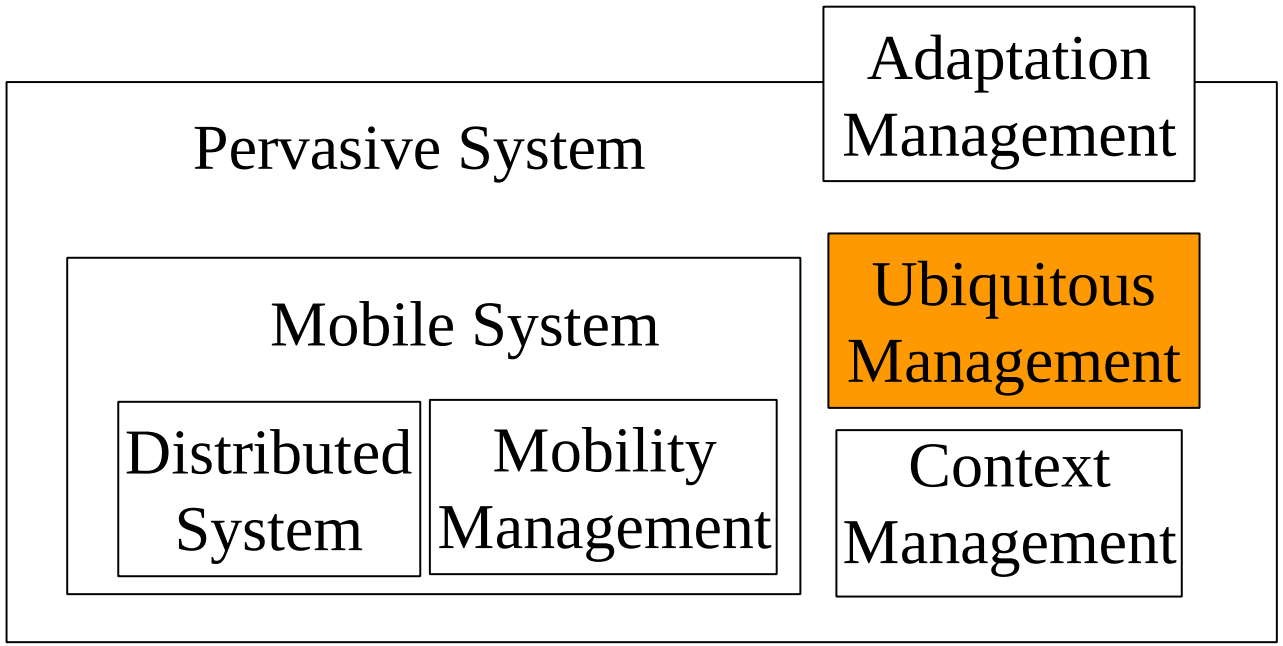
```
<instance xmlns=http://demo.heywow.com/schema/cool
xmlns:a=http://demo.heywow.com/schema/aspects
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<contextInformation>
<entity system="urn:phonenumber">+49-179-1234567</entity>
<characterizedBy>
<aspect name="GaussKruegerCoordinate">
<observedState
xsi:type="a:o2GaussKruegerType">367032533074</observedState>
<units>10m</units>
</aspect>
<certaintyOfObserver>90</certaintyOfObserver>
</characterizedBy>
</contextInformation>
</instance>
```

Synthesis

⇨Granularity of Virtual World vs Physical World

	Expressiveness and powerful	Easiness	Conflict Management
Attribute / Value	-	+	-
CC/PP	+	+	-
Ontology	+	-	+

Functional view of an AmI System



Ubiquitous Management

- Ways of accessing everything everywhere

- 2 steps
  - Discovery
  - Communication/Dissemination
    - Data results
    - Software itself

Service Discovery Protocols

- Service search
  - Where are the services ?
  - Where to store this knowledge ?

Service Discovery Protocols

- 2 implementation approaches
  - Service registry: centralized
  - Flooding: distributed
- Different search criteria
  - Location
  - Semantic
  - Mobility

Service Registry

- Organized set of available services
- Set provided by a dedicated host

- Examples:
  - SLP
  - Jini
  - Salutation

SLP registry: centralized data

- Service Location Protocol: Agent based
- Service Agent: the provided service
- Directory Agent
  - Registers the SAs in a LDAP registry
  - Multicast
- User Agent: the requester of the service
  - Multicast request

Jini registry: centralized soft

- Sun Jini: Java based
- Service provider
  - Identity and group broadcast
  - Renew registering
- Jini registry:
  - Stores RMI interface, proxy to service provider
  - Leasing mechanism, limited lifetime storing
- Service requester
  - Lookup request, receives proxy and location
  - Direct RMI proxy use

Salutation registry: neighbours decentralization

- Each host:
  - Stored a subset of available services
- Service provider:
  - Registers in local registry
  - And in neighbours registries
- Service requester:
  - Lookup in local registry
  - Then broadcasts to neighbours registries

UPnP: flooding discovery

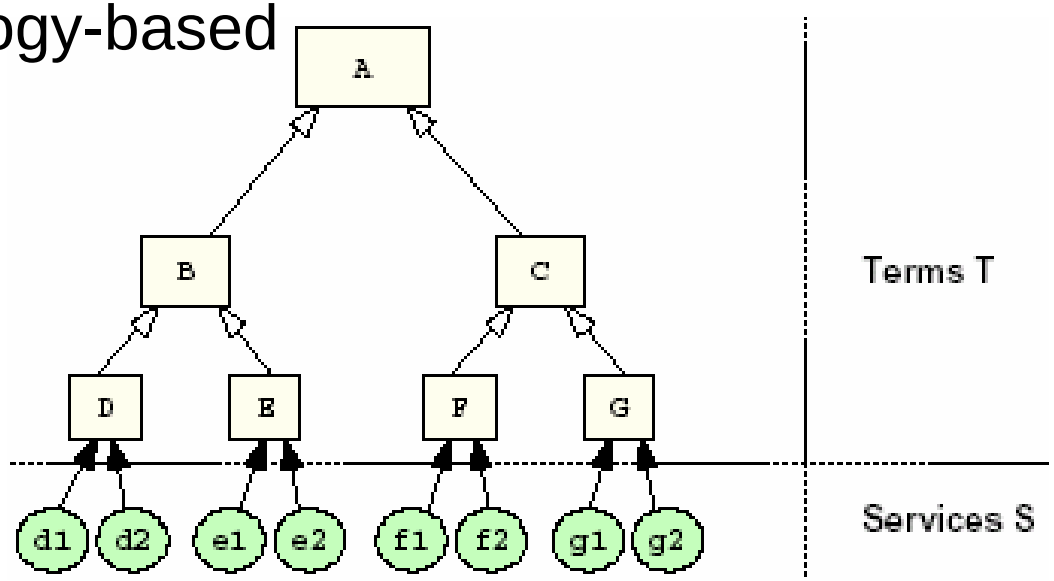
- Universal Plug and Play: Industrial consortium
- Each host
  - Available service list
  - Zero conf (DHCP, autoIP, multicast DNS)
  - Communication (point-to-point, streaming)
  - Automatic discovery
    - Multicast: XML messages, Arrival ANNOUNCE, Services available OPTIONS

Bluetooth registry: geographic service location

- Each host: SDP server
  - Stored local available services (service record: services attributes, class with unique UUID)
- Service provider:
  - Registers in local registry
- Service requester:
  - UUID lookup
  - Broadcast lookup for navigating into neighbours registries

Multi-layers clusters: geographic and semantic discovery

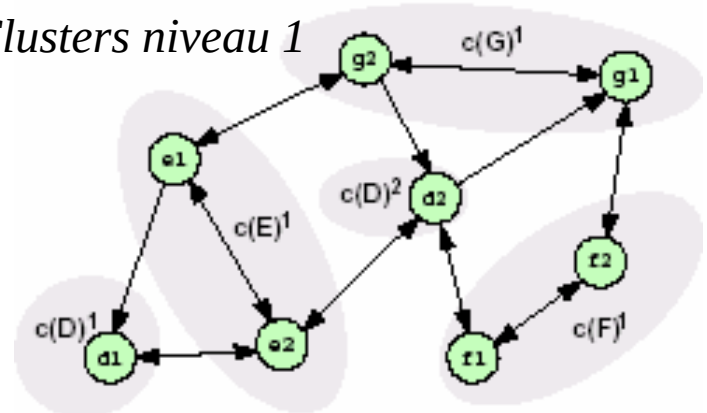
- Hosts grouping according to proximity
  - Geographic: direct routing (single hop)
  - Semantic: providing similar services
    - Ontology-based



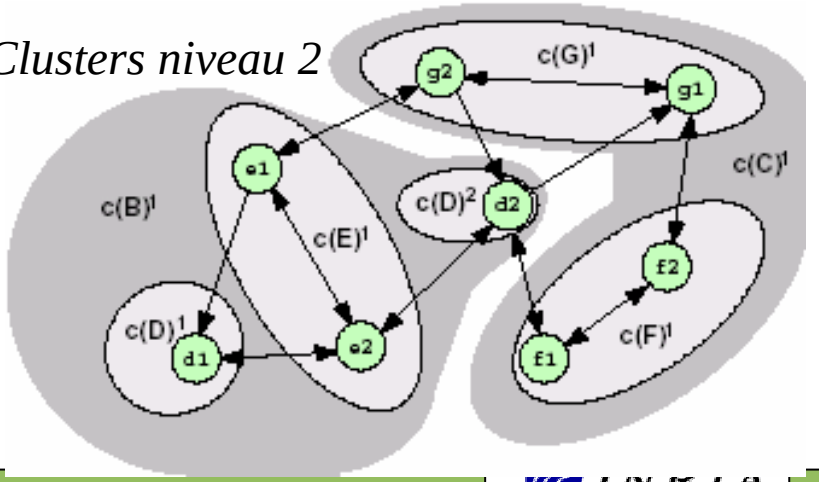
Multi-layers clusters: geographic and semantic discovery

- Service request:
  - Lookup in the closest cluster
    - Level 1: direct access or same service
    - Level 2: 2 hops or same category service
    - Level 3, etc.

Clusters niveau 1



Clusters niveau 2



Mobile Services Lookup

- During use, services can move
  - User mobility
  - Load balancing
- 3 accessing approaches
  - Location server
  - Poste restante
  - Repeaters

Mobile service discovery: location server

- Location server:
  - Stores pairs (service/location)
  - Each service warns the server of its location changes
- Service requester:
  - Asks the location server for the service location
  - Directly accesses to the service

Mobile service discovery: Poste restante

- Static proxy:
  - As the same service interface
  - Used as Poste restante for the service
  - The Service periodically reads and answers its messages
- Service requester:
  - Asynchronous communication
  - Send a message to the service
  - Proxy interception

Mobile service discovery: Repeaters

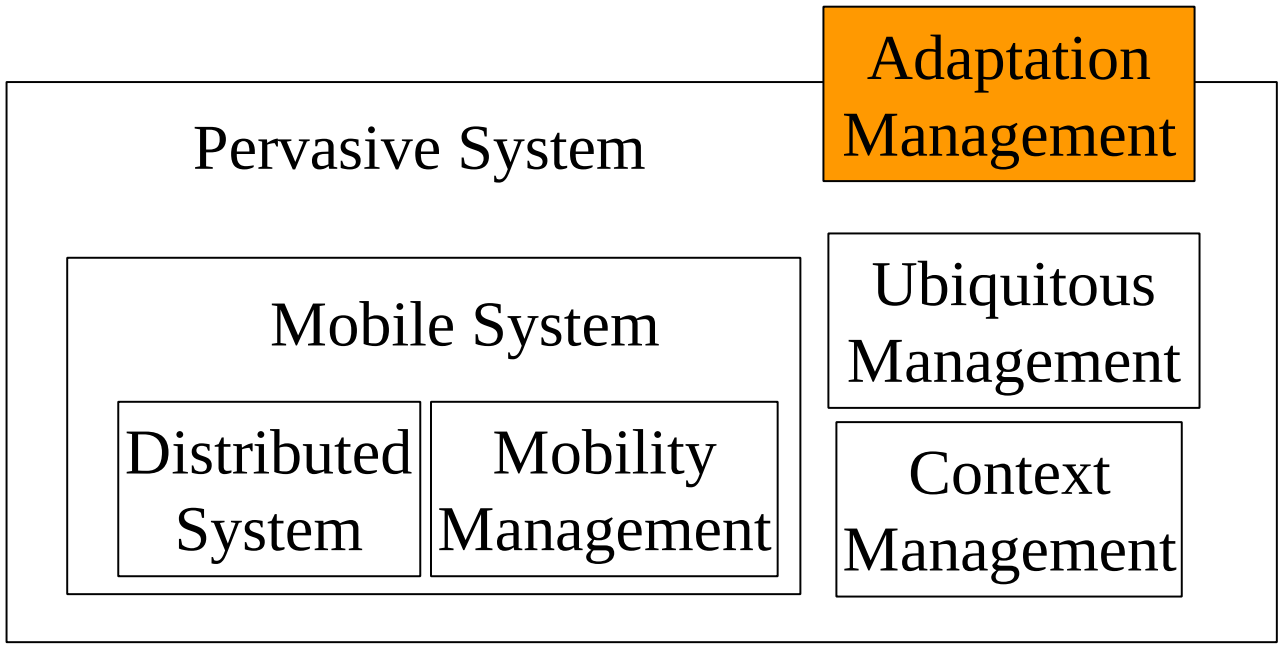
- Service provider:
  - Generates a repeater on each host it is leaving
  - A repeater knows the next location of the service
  - A repeater forwards the messages to the next service location
- Service requester:
  - Sends its request to a service at its last known location
- Several service moving: Repeaters chain

Synthesis

⇨ **Deployment scalability issue**

- **How large is my coverage ?**
- **How reactive is my system ?**
- **How robust must be my system ?**
- **How continuous must respect my services ?**

System view of an AmI System



Adaptation Management

- Way of acting according to environment changes
  - Can be a reaction to changes
  - Or can be a proactive advice or act trying to fit user's needs
- Composed of 2 parts
  - Technical mechanisms providing the way to act on the system
  - Smart rules representing the user's intentions

Technical Adaptation Mechanisms

- Dynamic adaptation

- Adaptation can focus:
  - The user interface (close to application)
  - The data
  - The services

Data adaptation

- Data adaptation: modification of data to respect the display rules of a target terminal
- Adaptation location:
  - On the client (not for light-client)
  - On the server (heavy-server)
  - On a active proxy network (load balancing, consistency to implement)

Adaptation according to data type

- Text source
  - Format conversion (html -> txt, doc -> pdf…)
  - Summary
  - Traduction
  - Compressing/uncompressing
  - Vocal synthesis
- Image source
  - Format conversion (jpeg -> png)
  - Modifications of resolution, colours number, depth…
  - Compressing/uncompressing (e.g. semantic jpeg or raw zip)

Adaptation according to data type

- Audio source
  - Format conversion
  - Textual synthesis or vocal recognition
  - Compressing/uncompressing (e.g. semantic MP3 or raw zip)
- Video source
  - Format conversion (resolution, nb images/sec)
  - Spatial Decomposition/recomposition (zoom…)
  - Compressing/uncompressing (e.g. semantic MPEG4 or brute zip)

Content adaptation operators

- Coding (Wav->MP3)
- Format (HTML ->WML)
- Modality replacement (image by descriptive text)
- Selection (size selection of images)
- Integration (multi-servers data)

Documents adaptation by WebServices

- [Berhe, Brunie 2004]
- WebServices-based adaptation architecture
  - Local Proxies
  - Content Proxies
  - Content Servers
  - Adaptation Service Proxies
  - Adaptation Services Repository
  - Profile Manager

Documents adaptation by WebServices

- 4 profile types:
  - Document
    - Physical meta-data (type, size, format…)
    - Storage meta-data (versions, repartition…)
    - Semantic meta-data (keywords…)
  - Client : user and terminal CC/PP
    - User : language, interests…
    - Terminal : hardware (screen size, memory...) and software (available, versions,…)
  - Network
    - Latency, bandwidth …
  - Service
    - WSDL : adaptation type, media type, performance, cost...

Documents adaptation by WebServices

- Local proxy
  - Receives user requests
  - Calculates the client profile
  - Sends the request to content proxy
  - Compares the answer profile with the client profile
  - Deduces an adaptation plan and applies it
  - Integrates adapted received data
  - Collaborates with other local proxies for cache management

Documents adaptation by WebServices

- Adaptation plan
  - Determines the adaptation constraints: attributes conditions
  - Determines the adaptation operators needed
  - Selects an optimal adaptation strategy
  - Looks for adaptation services that can realize the needed adaptation operators
  - Negotiates with the adaptation services (costs, performances)
  - Uses selected adaptation services

Documents adaptation by WebServices

- Notion of adaptation path
  - Sequence of adaptation operators that verifies the constraints
  - The path can be balanced by costs, performances…
- Notion of adaptation graph
  - When several adaptations can be applied in parallel on a subset of data (eg. image et meta-data on images – DICOM format)

Bibliographic references - Content adaptation

- E. Mory et al. Adaptation de contenu multimédia aux terminaux mobiles. RTSI - ISI n° spécial systèmes d ’information pervasifs n°9, 2004, Hermès: 39-60
- G. Berhe, L. Brunie, LIRIS Adaptation de contenus multimédia pour les systèmes d ’information pervasifs RTSI - ISI n° spécial systèmes d ’information pervasifs n°9, 2004, Hermès: 39-60

Services adaptation

- Services adaptation have generally 3 parts [Cremene 04]:
  - Modifiable part: the adaptable service
  - Monitoring part: continuous evaluation of the service and its context
  - Control part: definition of reconfiguration orders, according to the service logic

Adaptation in reflexive systems

- Reflexivity
  - Ability of a system to represent itself, to monitor itself and to act on itself
  - Meta level that describe the components of a system
- Introspection
  - System property allowing to know its internal state. Allows to reason and take decision about itself
- Intercession
  - System property allowing to change its behaviour by modifying its own functionality

Adaptation in reflexive systems

- Adaptation targets
  - Entity (methods, objects, components, services…)
  - Link between entity (links between base entities and/or between base and meta entities)
  - Set of entities
- Adaptation moment
  - Compilation : code generation according to meta-entity
  - Loading : alteration of compiled code or modification of dependencies in a set of entities
  - Execution : dynamic access to the meta level, by using proxy, or by the execution platform

RAM

- [Bouraqadi et al 01] Reflexion for Adaptable Mobility
- Code mobility = non functional aspect, so at meta level
- Cluster = unity of mobile code with:
  - A set of applicative objects
  - A meta interface for policies of the cluster (migration, …)
  - A table of instantiated bindings (references to other clusters)
- Strong code mobility

DynamicTAO: reflexive middleware

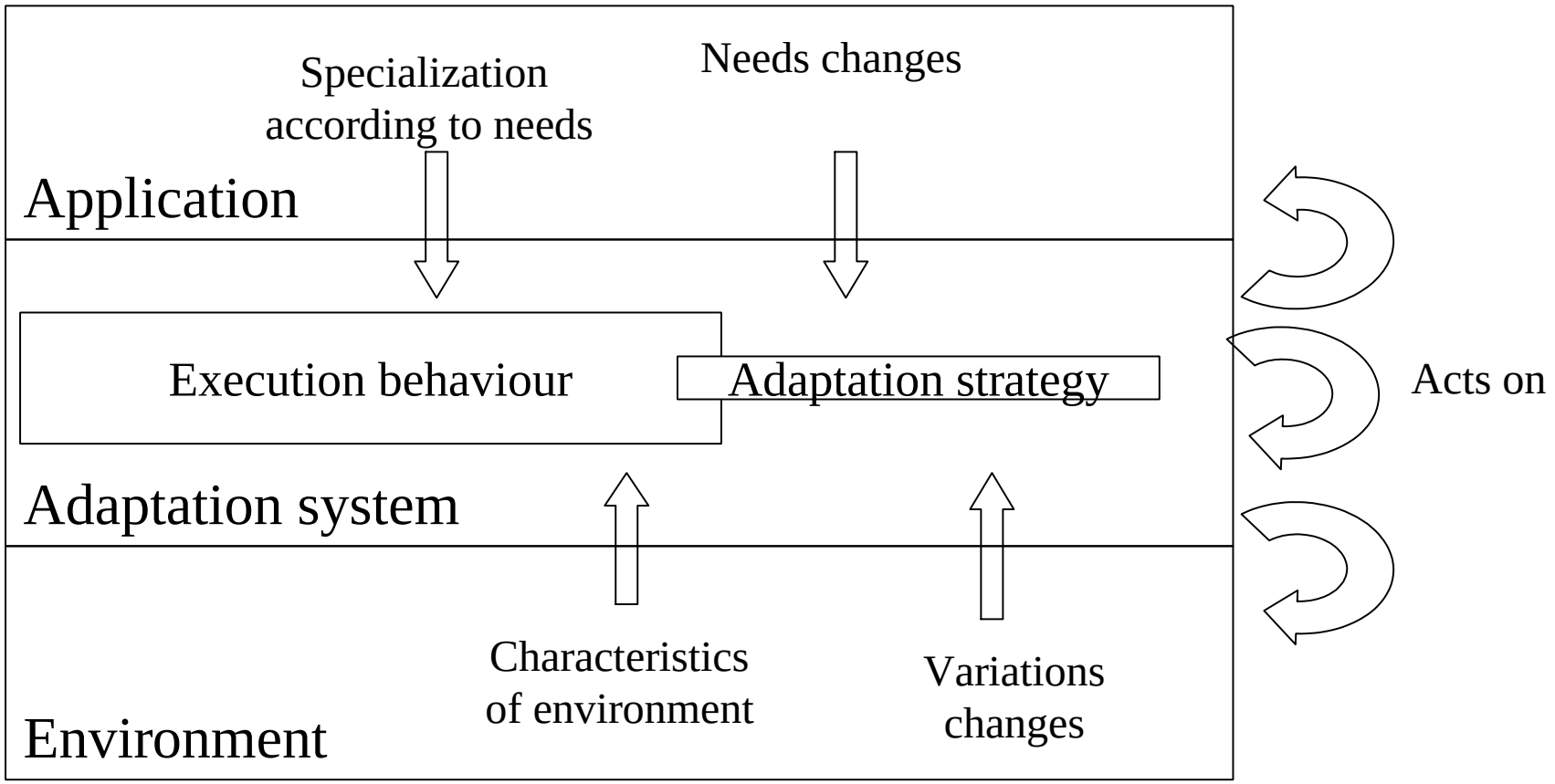
- [Kon et al. 2000] ORB reflexive based on CORBA
- Set of Component Configurators
  - A TAO Configurator maintains middleware strategies (concurrency, scheduling…)
    - Component calls interception -> strategies
    - Dynamic component load of implementations (even for strategies)
- Component dynamic reconfiguration

Dream : Dynamic REflective Asynchronous Middleware

- [Leclerc et al. 2005] « component-based framework for constructing, statically or dynamically, resource-aware, configurable message-oriented middleware (MOM) ».
- Fractal extension (Java based)
  - Dynamic assembly of components
    - Primitive components
    - Composite components
  - For each component, management interfaces:
    - BindingControler : components dependencies management
    - ContentControler: adding and removing components
    - LifeCycleControler : run, stop components

AeDEN – Software entity adaptation

- [Le Mouel 2003]

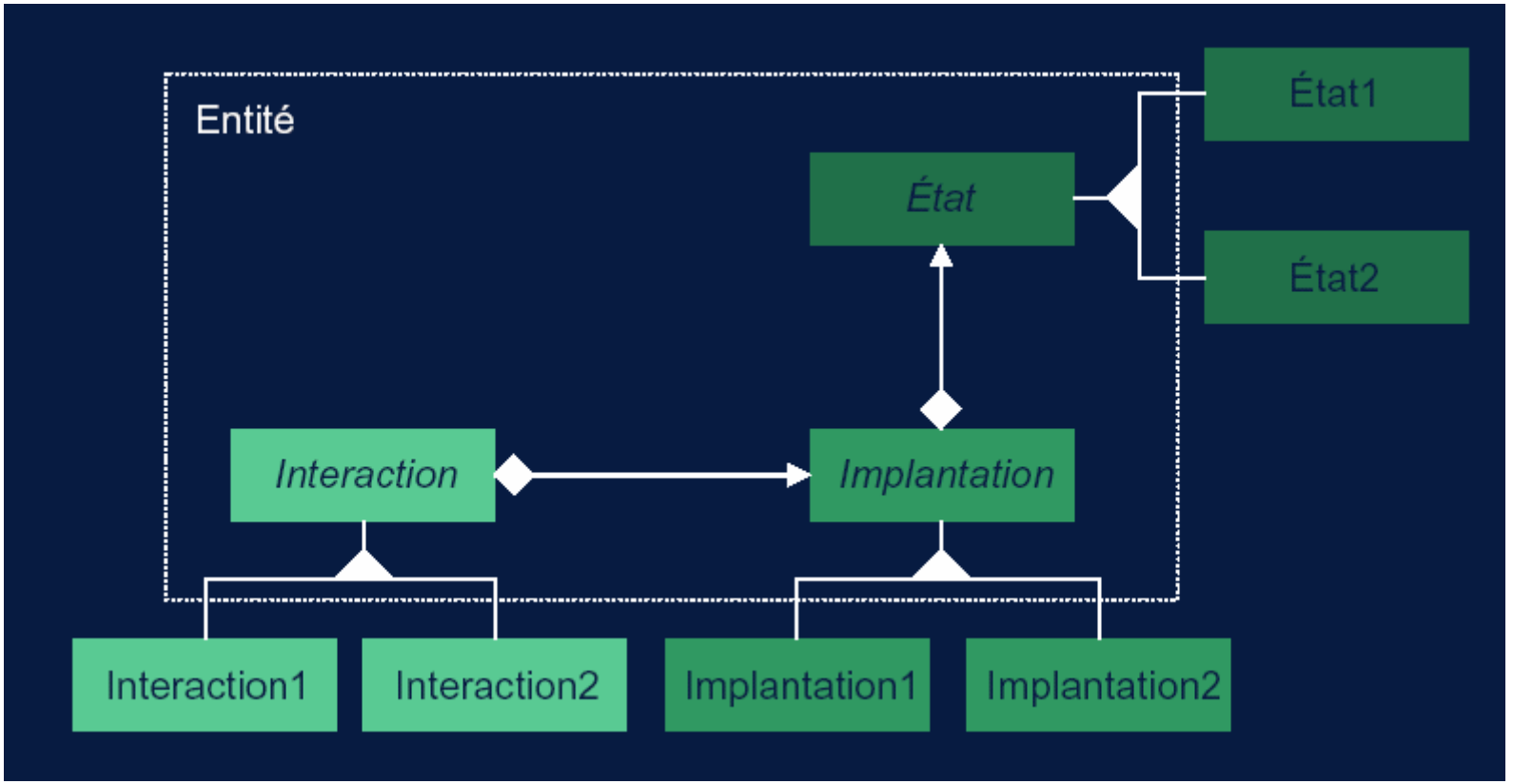


AeDEN - entity

- Entity = software conception unity
  - Abstract and specializable
  - A functionality <=> A entity
  - A service <=> A specialized entity
- 3 aspects:
  - AInteraction (communication with other entities)
  - AImplementation (business : expected treatments)
  - AState (internal state of the business part)
- Different implementation available for each aspect

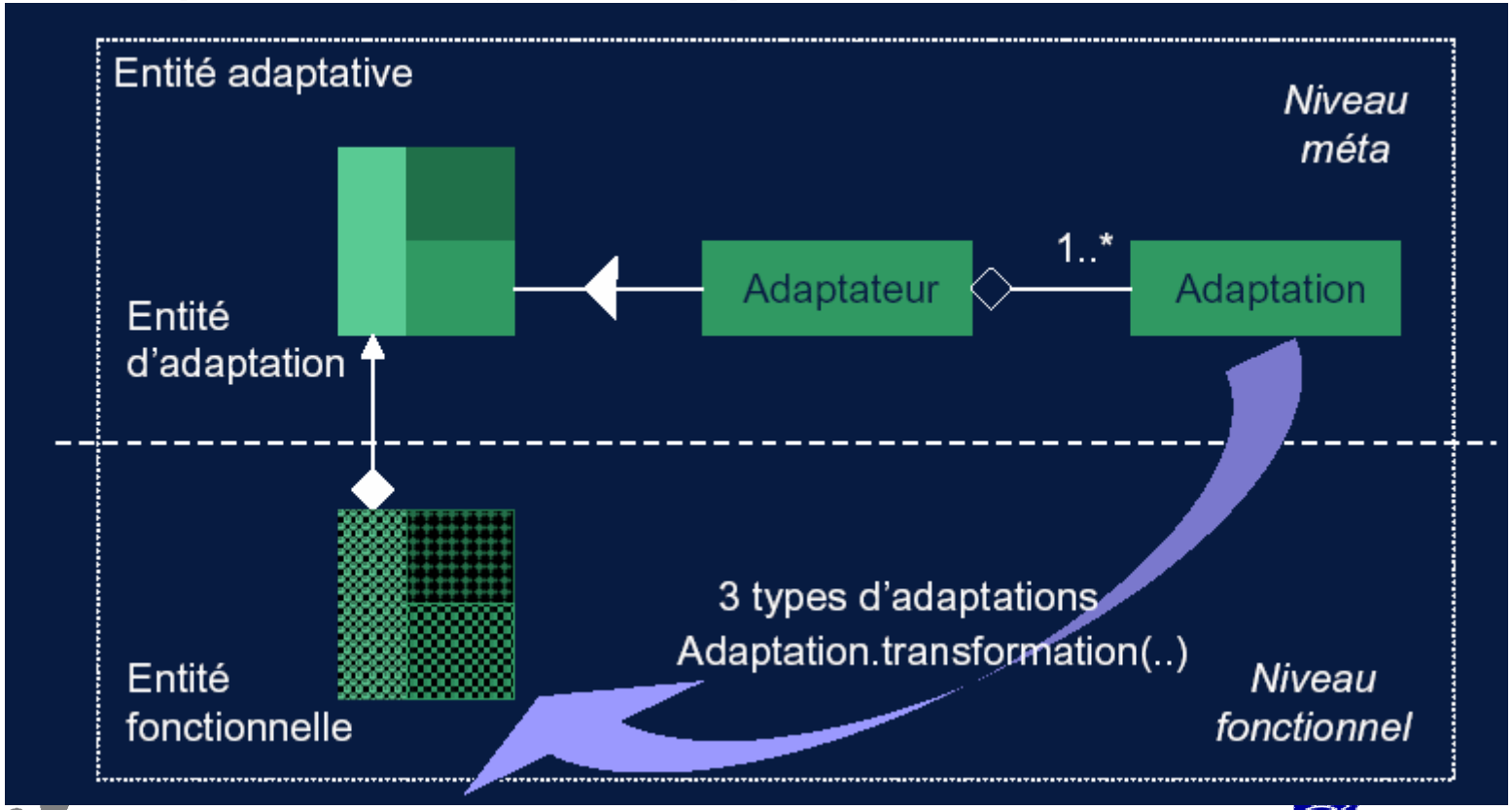
AeDEN - entity

- Abstract entity + possible specializations



AeDEN – Adaptive entity

- Entity + Adaptation entity
  - Adaptations by introspection and intercession

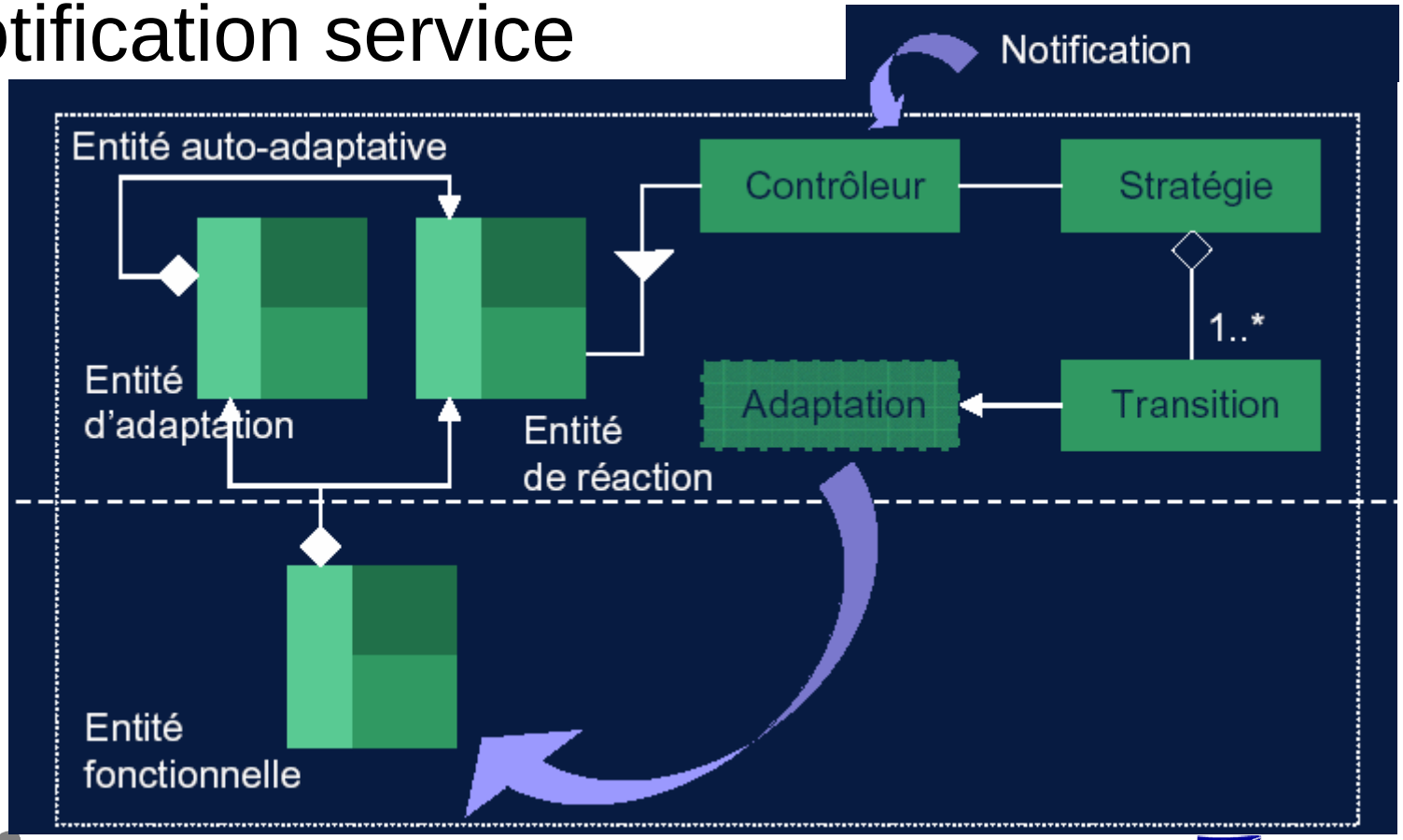


AeDEN - introspection and intercession

- Each entity has the following methods:
  - getInteraction(), setInteraction()
  - getImplementation(), setImplementation()
  - getState(), setState()
- Each adaptive entity has the following methods:
  - getFunctionalInteraction(), setFunctionalInteraction()
  - getFunctionalImplementation(), setFunctionalImplementation()
  - getFunctionalState(), setFunctionalState()

AeDEN - Adaptive and reactive entity

- Adaptation entity + reaction entity linked to a notification service



Technical Adaptation Mechanisms - Synthesis

- **Adaptation granularity vs Development complexity**
- **Exhaustiveness vs Transparency**
  - **By definition, data and interface are more difficult to transparency adapt vs internal services adaptation are transparent**

Bibliographic references - Adaptation services

- M. Cremene et al. « Adaptation dynamique de services », Decor ’2004, Grenoble, octobre 2004, pp 53-64
- CAAD : T. Chaari, F. Laforest, A. Celentano design of context-aware applications based on web services Rapport de recherche LIRIS, octobre 2004 - RR-2004-033
- RAM : N.M.N. Bouraqadi-Saadani et al. « A reflexive infrastructure for coarse grained strong mobility and its tool-based implementation. » Int. Workshop on experiences with reflexive systems. Sept. 2001
- DynamicTAO : F. Kon et al. « Monitoring, security and dynamic configuration with the dynamicTAO reflective ORB » Middleware 2000
- AeDEN : F. Le Mouël « Environnement adaptatif d’exécution distribuée d’applications dans un contexte mobile » mémoire de thèse de doctorat en informatique, Université Rennes I, 1er décembre 2003
- M. Leclercq, V. Quma, J.-B. Stefani, DREAM: A Component Framework for Constructing Resource-Aware, Configurable Middleware, IEEE distributed systems online, vol. 6, no. 9, September 2005

Smart rules for Adaptation

- Way of expressing users and applications needs and intentions [Augusto 2007] :
  - Rules are user and application-specific
  - Rules goals are to optimize

Examples of application goals

- User Advice and Assistance
  - Collective intelligence for user profiling
- User and Society Privacy
  - Reduce the user information leak
- Green IT
  - Reduce of greenhouse gas emission
- Citizenship
  - Improvement of responsiveness, efficiency and transparency of public e-government services

Expressing the adaptation rules

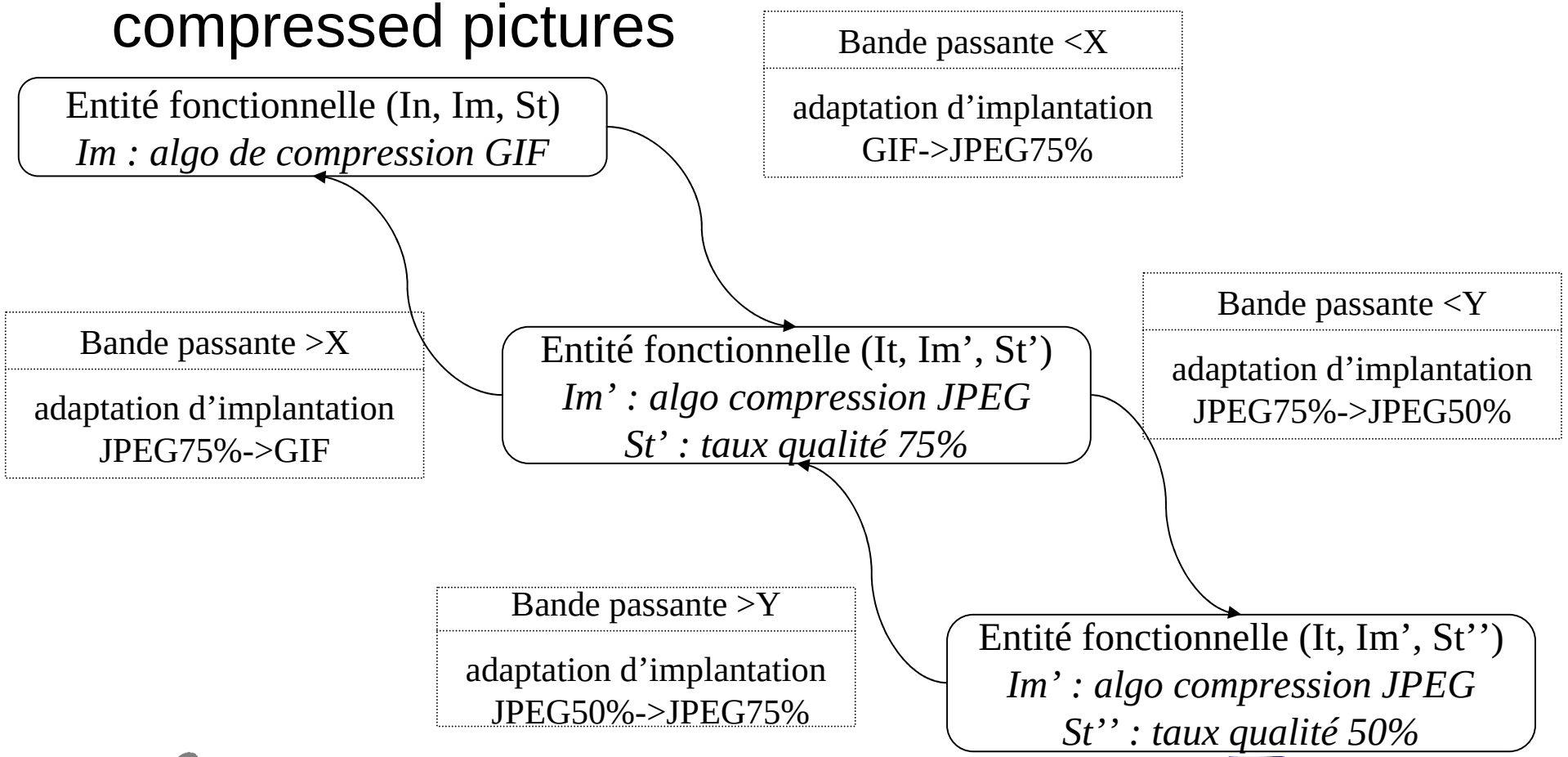
- Different models and algorithms [Aarts 2005, Bikakis 2008]
  - ECA model:
    - on Event if Condition do Action
  - Example: Human-based fuzzy rules [Acampora 2006] :
    - IF Velocity is LOW AND LeftEye is CLOSED AND RightEye is CLOSED AND Speech is LOW AND Position is BED
      THEN SLEEPING is ON

Expressing the adaptation rules

- Different models and algorithms
  - Automaton model:
    - Ex: Markov chains
    - Powerset construction: Algorithm to convert nondeterministic automaton to deterministic automaton.
  - Example : Hidden Markov Models for Activity Recognition in Ambient Intelligence Environments [Sanchez 2007]

AeDEN – Automaton Strategy

- Adaptation strategy for the transmission of compressed pictures



Expressing the adaptation rules

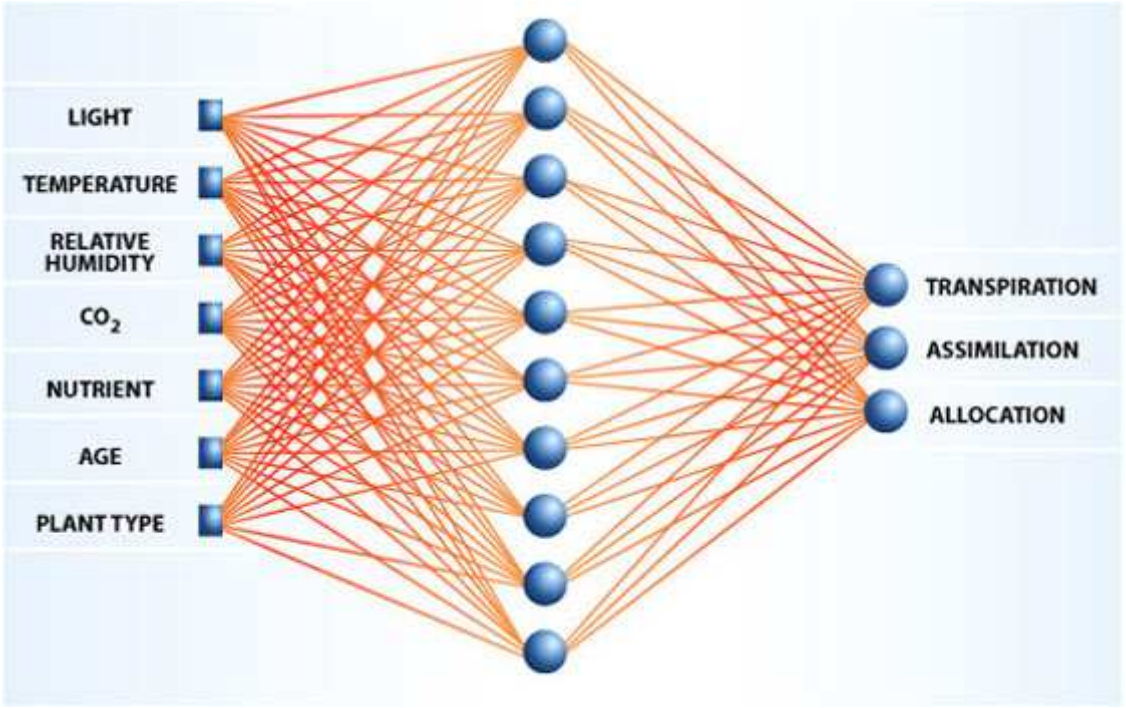
- Different models and algorithms
  - Genetic algorithms
    - Population Initialization
    - Selection
    - Reproduction (hazard and mutations)
    - Termination
  - Example: Learning User Profile with Genetic Algorithm in AmI Applications [Venturini 2008]

Expressing the adaptation rules

- Different models and algorithms
  - Neural networks
    - Weight modifications

Neural networks

- User and environment behavior
- Often coupled with a sensor network



Smart rules Synthesis

⇨**Infering new rules vs Performance**

	Infering new rules	Performance
ECA	+ / -	- / ++
Automaton	--	+
Genetic	-	--
Neural	+	-

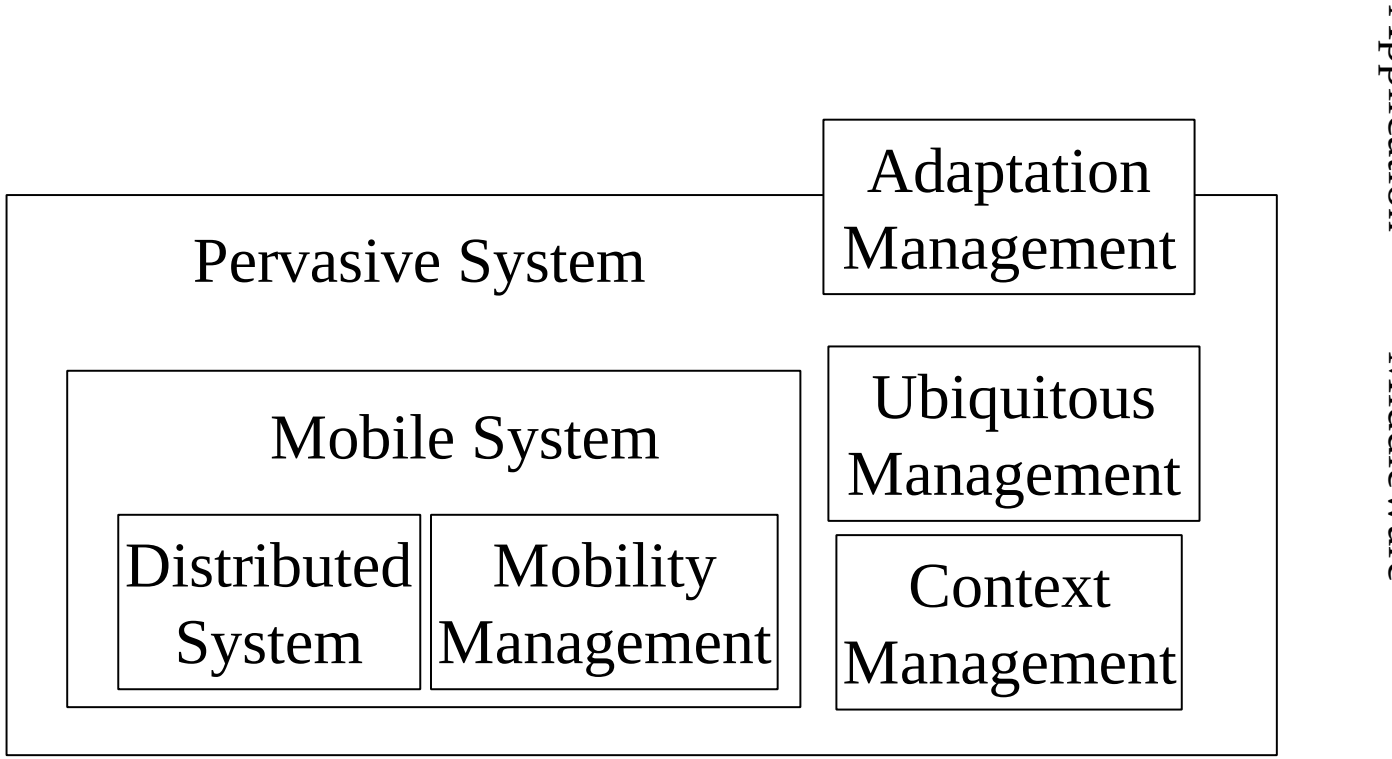
Bibliographic references - Adaptation rules

- J.C. Augusto “ Ambient intelligence: The confluence of pervasive computing and artificial intelligence”, Springer, Heidelberg, 2007
- E. Aarts, J. Korst W.F.J. Verhaegh « Algorithms in Ambient Intelligence », Ambient Intelligence, Springer, doi:10.1007/3-540-27139-2_16, 2005
- A. Bikakis, T. Patkos, G. Antoniou, D. Plexousakis « A Survey of Semantics-Based Approaches for Context Reasoning in Ambient Intelligence », AmI 2007 Workshops, CCIS 11, pp. 14-23, 2008
- G. Acampora, V. Loia, M. Nappi, S. Ricciardi « Human-Based Models for Ambient Intelligence Environments », Artificial Intelligence and Integrated Intelligent Information Systems: Emerging Technologies and Applications, Chapter 1, ISBN 1599042495, 2006
- D. Sanchez, M. Tentori, J. Favela «Hidden Markov Models for Activity Recognition in Ambient Intelligence Environments », Proceedings of the Eighth Mexican International Conference on Current Trends in Computer Science, pp. 33-40, ISBN 0-7695-2899-6, 2007
- V. Venturini, J. Carbo, J.M. Molina, “Learning User Profile with Genetic Algorithm in AmI Applications”, HAIS 2008, LNAI 5271, pp. 124-131, doi: 10.1007/978-3-540-87656-4, 2008
- AeDEN : F. Le Mouël « Environnement adaptatif d’exécution distribuée d’applications dans un contexte mobile » mémoire de thèse de doctorat en informatique, Université Rennes I, 1er décembre 2003

Adaptation conclusion

- General needs [Le Mouel 2003]:
  - Genericity : use by different kind of applications
  - Modularity : splitting and decorrelation
  - Evolution : integration of new technologies and new functionalities
  - Context-aware : integration of environment modifications
  - Dynamicity : reaction to changes without stopping the system
  - Efficiency : performance and stability

Synthesis



In a real system, all functionalities are interlaced !!!

Synthesis

- **Roadmap to build an AmI System – Questions?**
  - Which kind of functionalities of a classic distributed system do I need (load balancing, fault tolerance, etc) ? Can I reuse one, or do I have to redevelop ?
  - How do I model my context ? What do I have to include in my context ? Expressiveness/Physical World level required ?
  - What are the changes expected in my system ? Where and how do I have to adapt to these changes ?
  - What is the level of smartness required by my applications/users ?