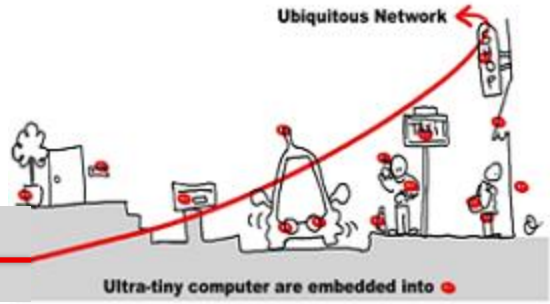
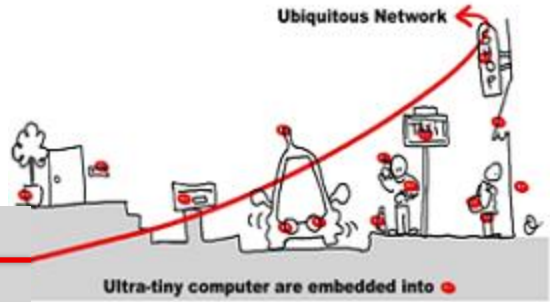
From Web Services Middleware to Web Services for Device Middleware



Associate Professor Stéphane Lavirotte
http://stephane.lavirotte.com/

University of Nice – Sophia Antipolis / Polytech'Nice - Sophia
Member of the RAINBOW research team (I3S)

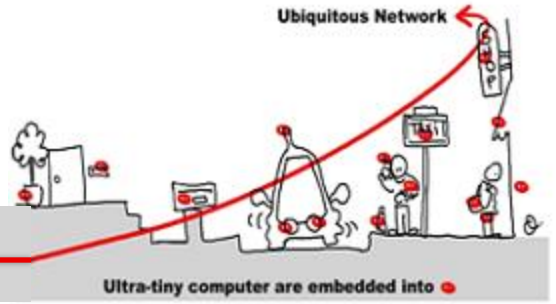
Email : stephane.lavirotte@unice.fr



INTRODUCTION

A World of Services

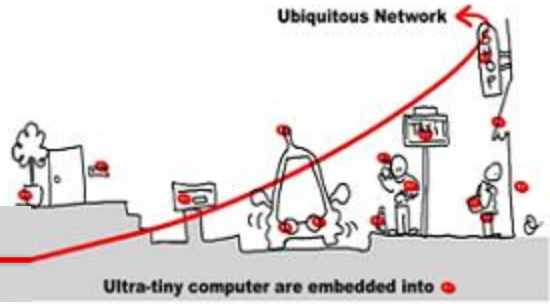
Software Engineering Evolution

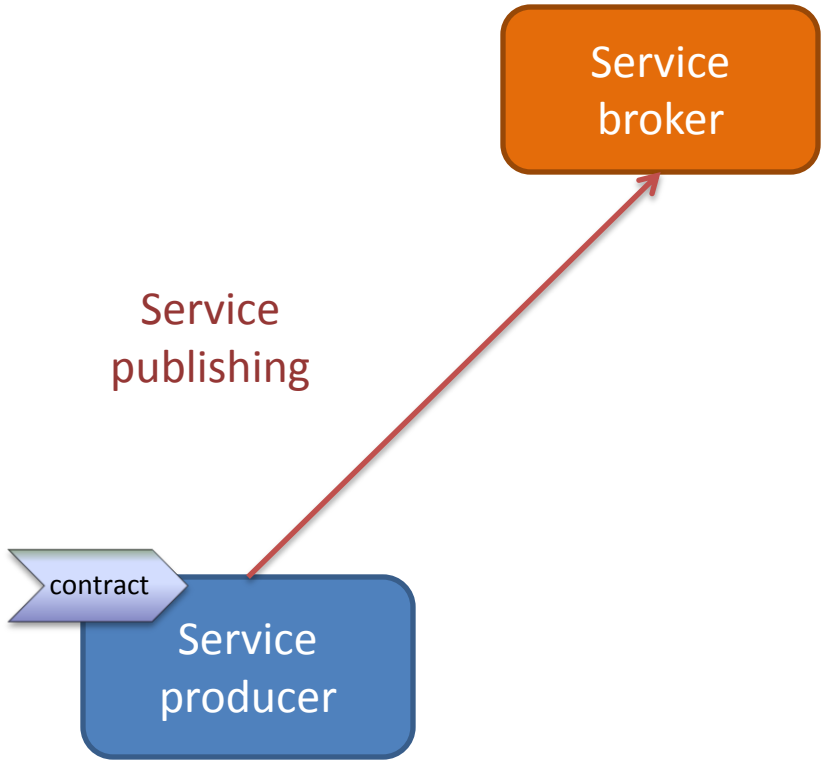


- In the field of Imperative programming
  - describes computation in terms of statements that change a program state
- Structured or Procedural programming
  - removing or reducing reliance on the GOTO statement (Dijkstra 1968)
- Object-oriented programming
  - data structures consisting of data fields and methods together with their interactions
- Component programming
  - separation of concerns in respect of the wide-ranging functionality available throughout a given software system
- Service-oriented programming
  - provide a loosely-integrated suite of services that can be used within multiple business domains

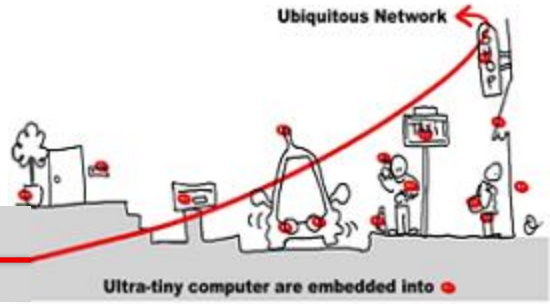
Encapsulation

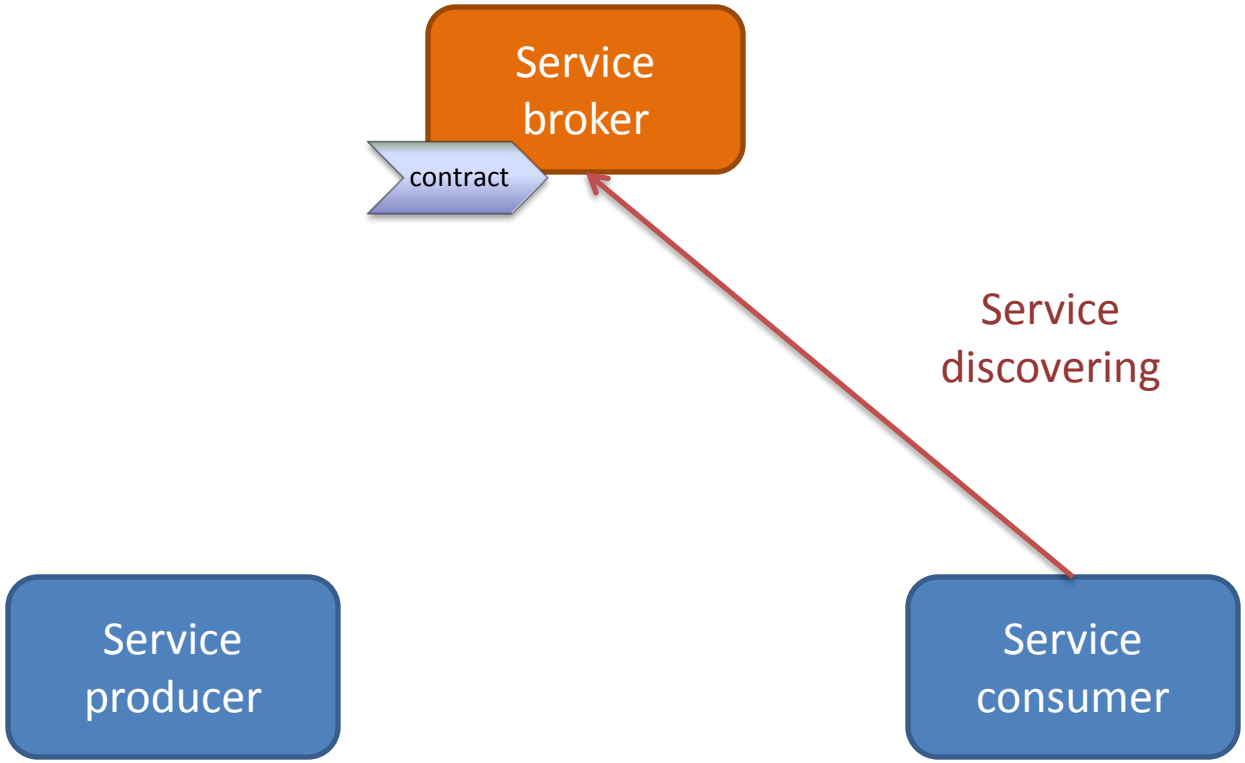
Advertising: Service Publishing



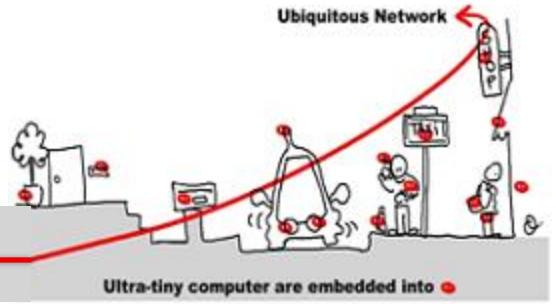


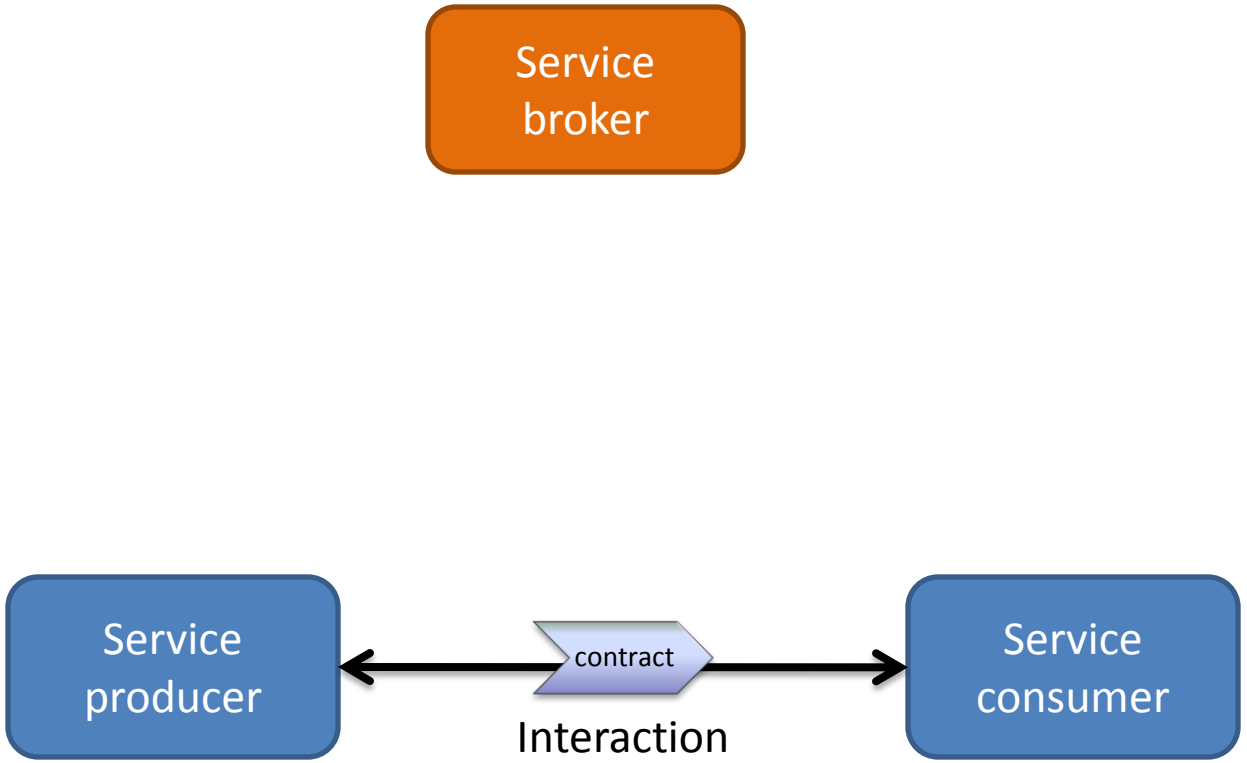
Service Discovery



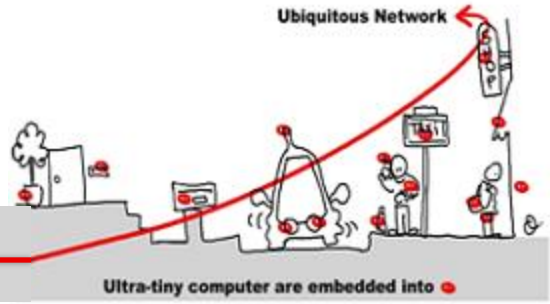


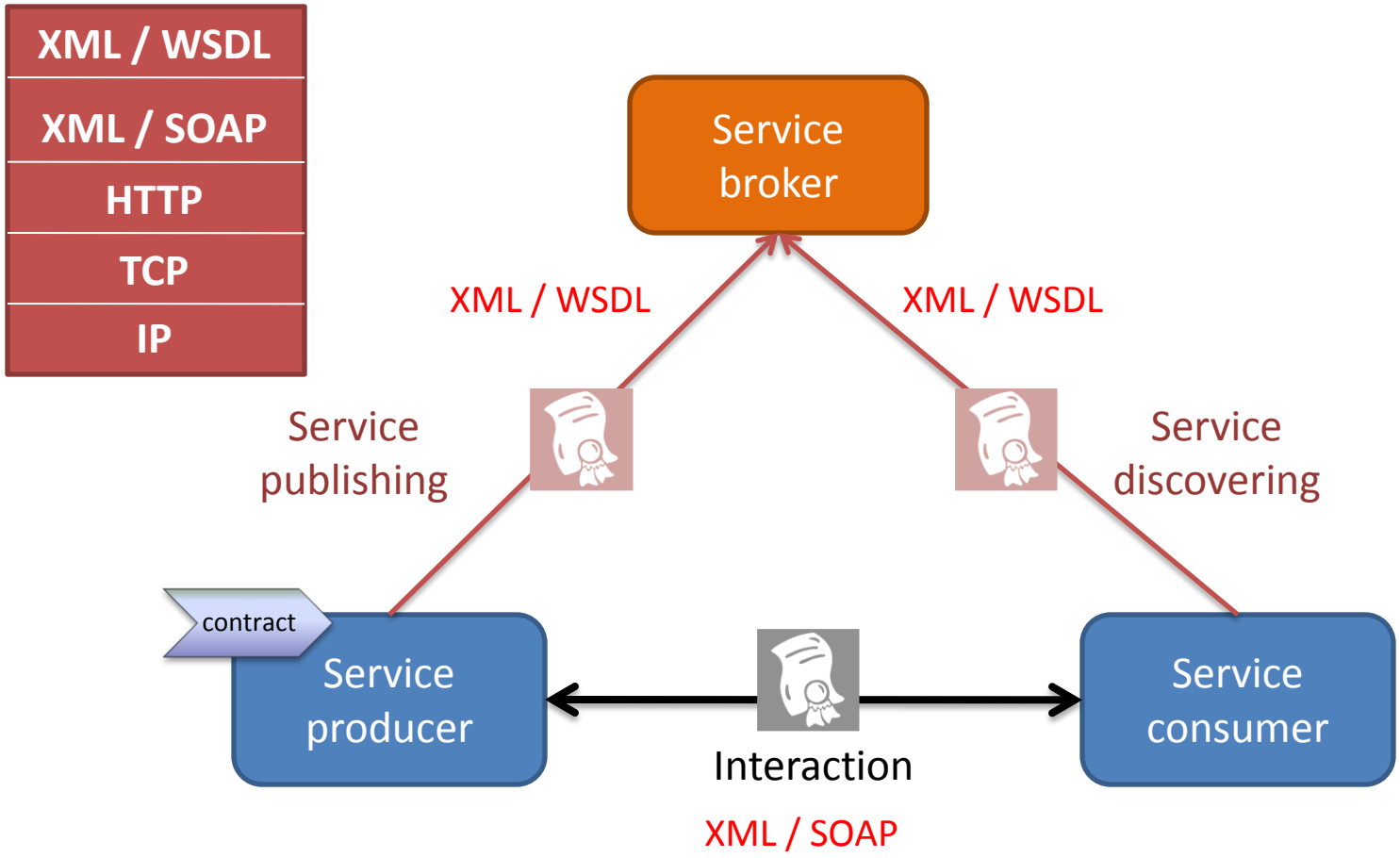
Loose Coupling



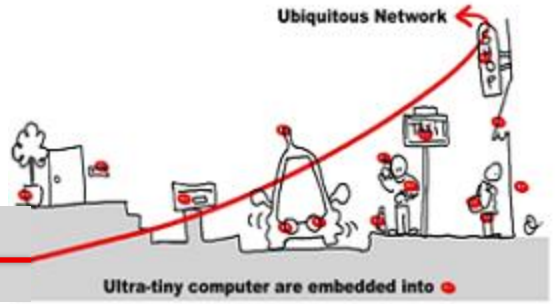


Interoperability (Web standards)



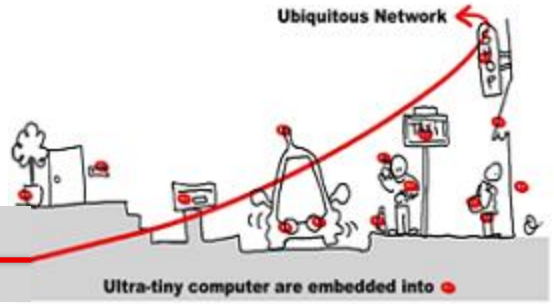


From (Web) Services…



- To sum up advantages of services approach:
  - Reusability (by sharing)
  - Modularity (replacing services by another one)
  - Advertising (by publishing)
  - Discovery (by subscribing and describing)
  - Loose coupling (between entities of application)
  - Interoperability (due to Web Services)
  - Scales (from “world wide” to devices)
- But…

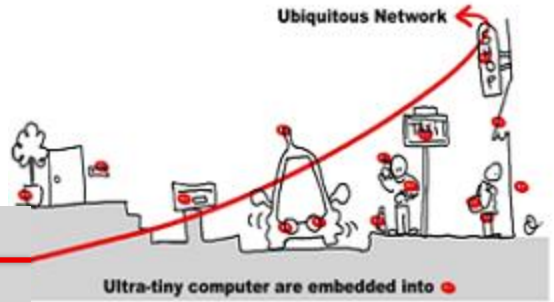
… to Web Services for Device



- Due to Ubiquitous Computing Context
  - For physical and multiple devices
  - Spontaneous Communication
    - Communications between systems not made until runtime
  - “Should compute at the right time”

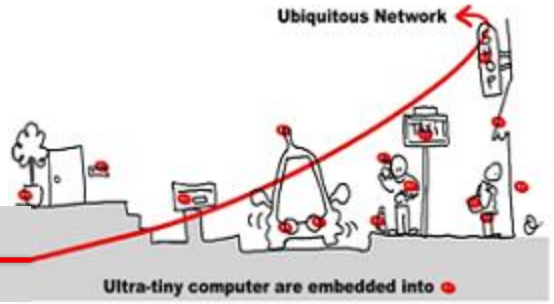
⇨New challenges

  - Dynamicity
    - Distributed dynamic research and discovery
  - Reactivity and Response Time
    - Evented interaction mechanism

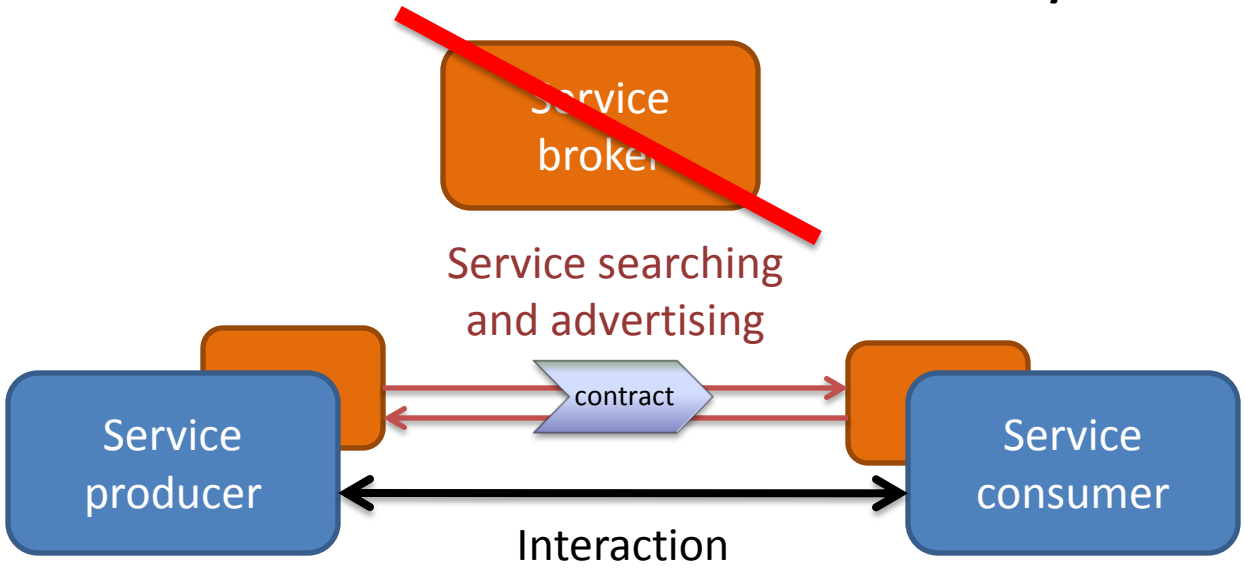


CHALLENGES FOR NEW SERVICES

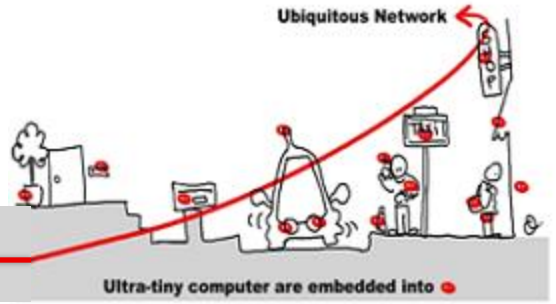
High Dynamicity



- Distributed dynamic Research and Discovery
  - Appearance and Disappearance management
  - Allow contextual research and discovery

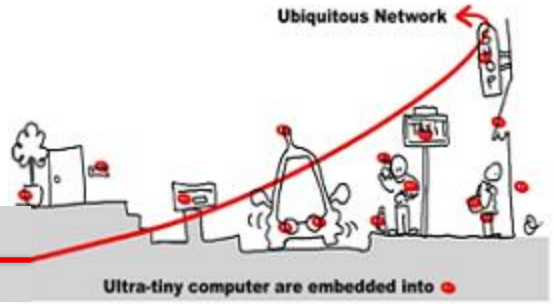


Service Discovery Protocols

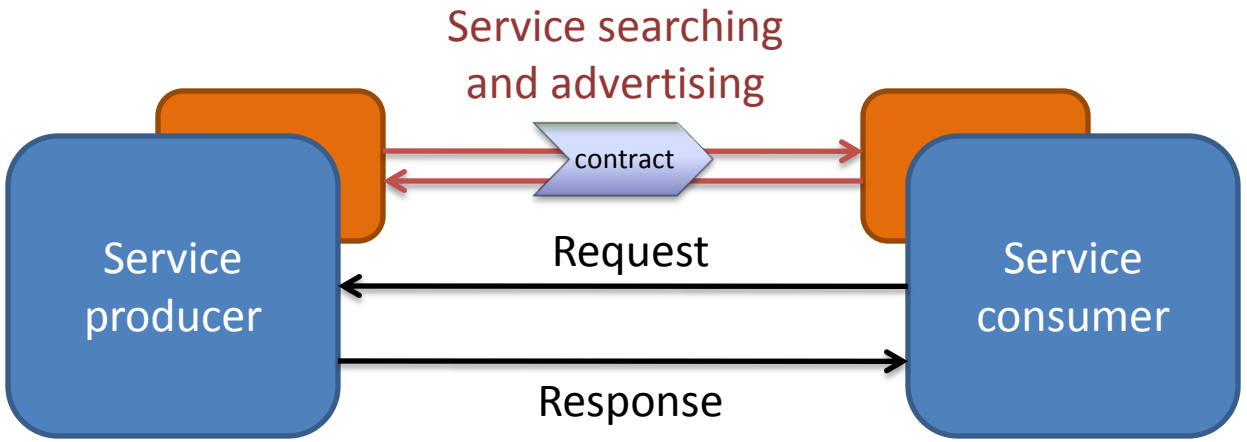


- Multicast DNS/DNS-SD: Apple's protocol
  - Multicast DNS: uses API similar to unicast DNS
- SLP: IETF proposed standard
  - Supported by HP, Novell, Sun Microsystems, Oracle
- SSDP: Microsoft's protocol
  - Uses HTTP notifications (see bellow), used since XP
- WS-Discovery: Defined by OASIS
  - Latest defined protocol, used in DPWS (see bellow)

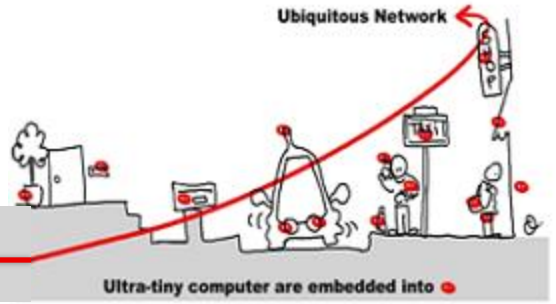
Traditional Interactions: Invocations



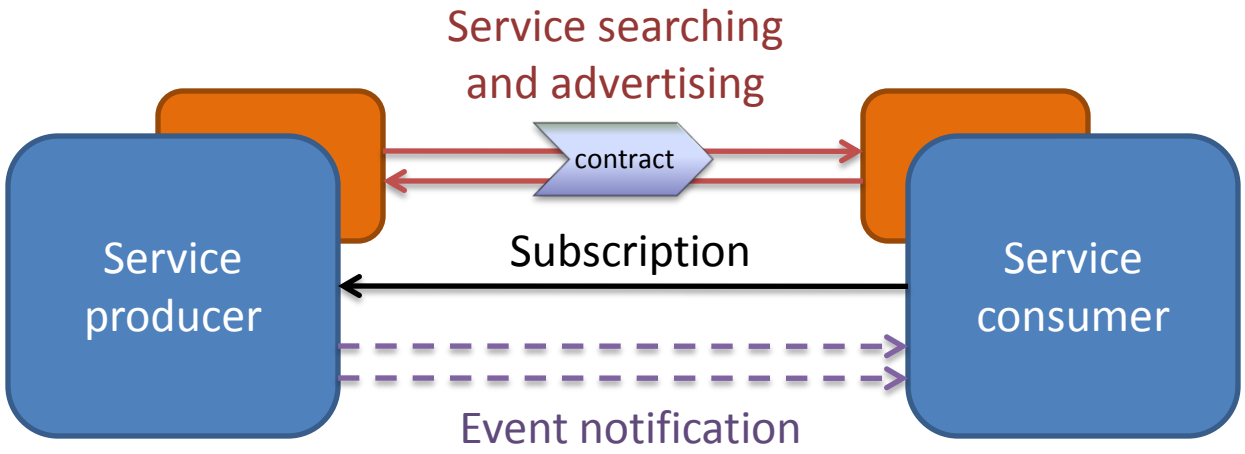
- "Classical" way to interact between services
  - Request-Response mechanism



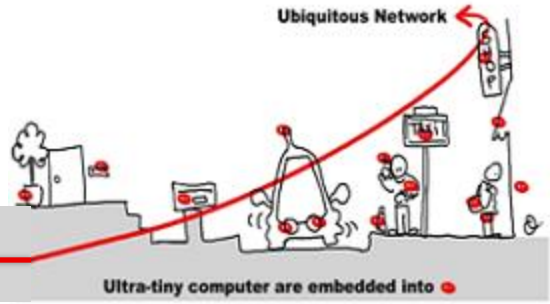
Reactivity



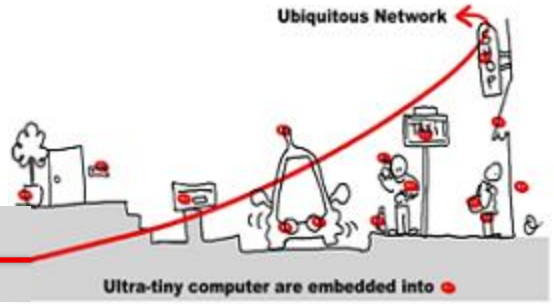
- "New way" of interacting: Eventing interaction model
  - Based on publish/subscribe design pattern
  - Asynchronous messaging (based on push mode)



References

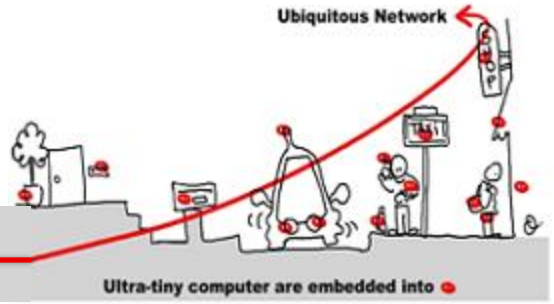


1. "**Service and device discovery: protocols and programming**", *G. G. Richard*, McGraw-Hill Professional, ISBN 0-07-137959-2, 2001.
2. "**Integration of Embedded Devices Through Web Services: Requirements, Challenges and Early Results**", G. B. Machado, F. Siqueira, R. Mittmann , C. Augusto, V. e. Vieira, in *Proceedings of the 11th IEEE Symposium on Computers and Communications*, pp 353-358, 2006.
3. "**Web Based Service for Embedded Devices**", U. Topp, P. Müller, J. Konnertz and A. Pick, LNCS, volume 2593, pp 141-153, 2009.
4. "**The device service bus: a solution for embedded device integration through web services**", G. M. Araújo, F. Siqueira, in *Proceedings of the 2009 ACM symposium on Applied Computing*, pp 185-189, 2009.
5. **"Towards an Architecture for Runtime Interoperability "**, Amel Bennaceur, Gordon S. Blair, Franck Chauvel, Gang Huang, Nikolaos Georgantas, Paul Grace, Falk Howar, Paola Inverardi, Valérie Issarny, Massimo Paolucci, Animesh Pathak, Romina Spalazzese, Bernhard Steffen, Bertrand Souville, in ISoLA, pp 206-220, 2010.



WEB SERVICES FOR DEVICES

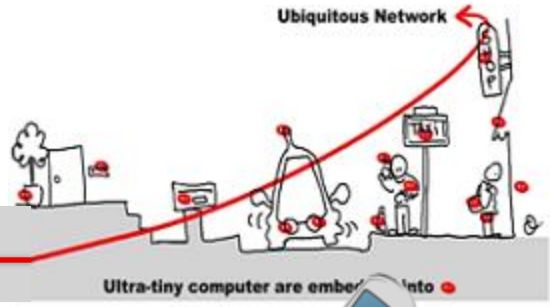
Standards and Protocols



UPNP

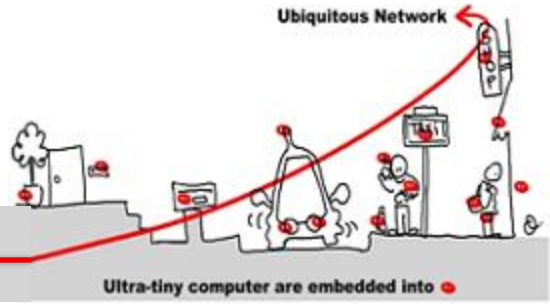
Universal Plug and Play

Universal Plug and Play



- Control Point
  - The client which discover and control UPnP servers
- Device
  - The server (receive actions)
- A physical device can be twice (CP and Device)

Example of UPnP Device Communications



Device

Addressing

DHCP

Auto-IP

Description

Discovery

Device Advertisements over SSDP channel

Discovery Search

Action Request

Control Request

Control

Action Response

Control Response

Events generated by the device

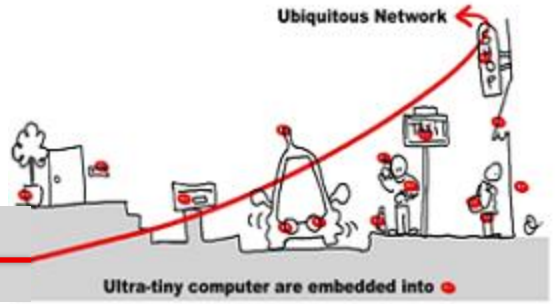
Eventing

Event subscription

Events notification

Network

UPnP Stack and Protocols



UPnP Vendor

UPnP Forum Working Commitee

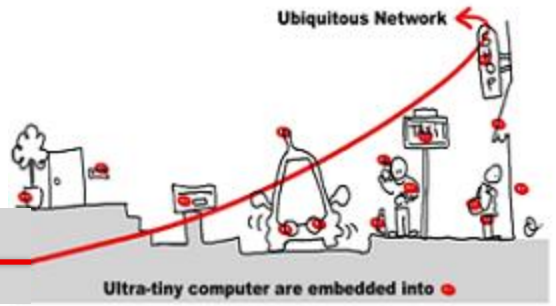
UPnP Device Architecture

Addressing **DHCP / Auto IP**	*Discovering* **SSDP**	*Description* **XML**	*Controling* **SOAP**	*Eventing* **GENA**	*Presenting* **HTML**

HTTPU/MU | HTTP

UDP | TCP

IP



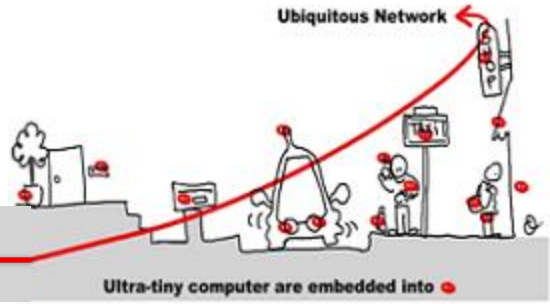
Practical Lab

FULL TUTORIAL OF THE INTEL® AUTHORING TOOLS FOR UPNP TECHNOLOGIES

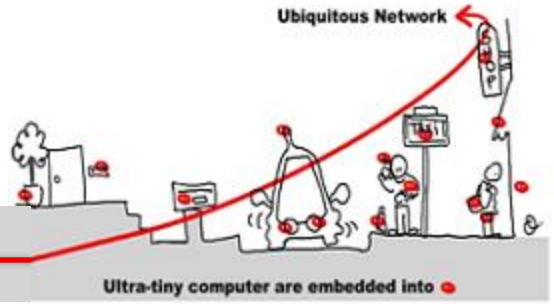
Ylian Saint-Hilaire

Senior Architect

Drawbacks of UPnP Technology



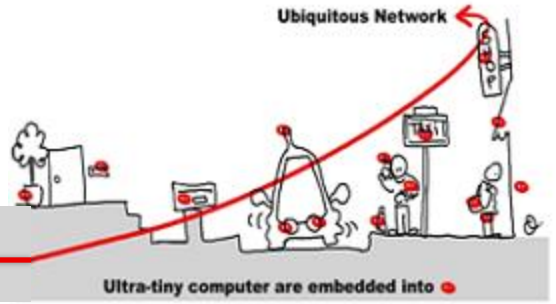
- Heavy protocol for tiny devices (web server + xml parser + soap + …): bridge solutions
- Based on non-standards protocols:
  - Discovery: SSDP over HTTPU, HTTPMU (not standardized since 2000)
  - Description: XML dialect, not standardized
  - Events: GENA (not standardized)
- No authentication protocol (security)[4]



DPWS

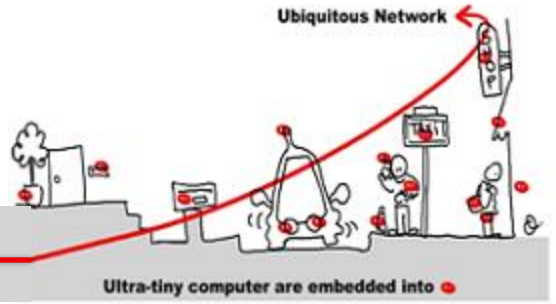
Device Profile for Web Services

Device Profile for Web Services

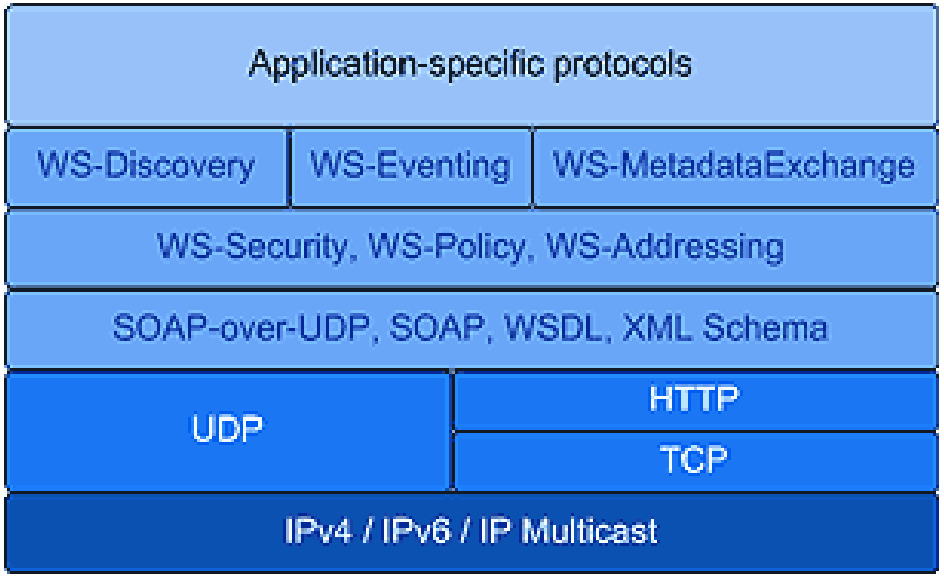


- DPWS[1]: Same goal as UPnP (UPnP v2)
  - But without backward compatibility
- Using or defining standards
  - WSDL, WS-Discovery, WS-Eventing, …
- Approved as OASIS standard on 30, june 2009
- All or some parts of DPWS already included in Vista, Micro .NET, Windows CE, …

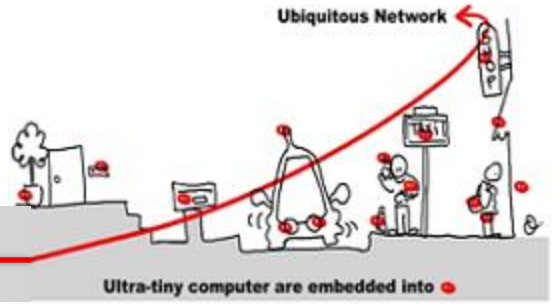
DPWS Stack and Protocols



- Only based on standards
  - SOAP 1.2,
  - XML,
  - XML Schema,
  - WSDL 1.1,
  - WS-Addressing,
  - WS-Transfer,
  - WS-Policy,
  - WS-Security,
  - WS-MetadataExchange,
  - WS-Discovery
  - WS-Eventing

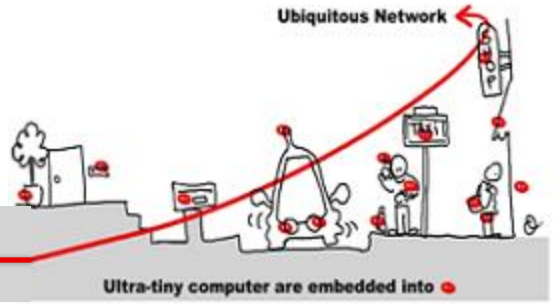


DPWS implementations emerged with the help of Research Projects



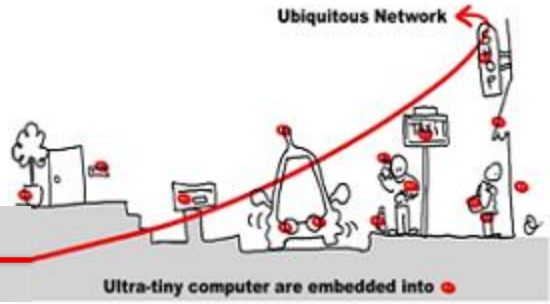
- European Research Initiative ITEA
  - SIRENA project (2003-2005)
    - SOA4D: SOA for Devices (Java and C Stack)
    - WS4D: Web Services for Devices (Java, Java ME and C Stack)
  - SODA project (Service Oriented Device and Delivery Architecture) (2006-2008)
- EU Research Project
  - SOCRADES (2006-2009) composed by heavyweights like ABB, SAP, Schneider Electric, and Siemens

Using DPWS



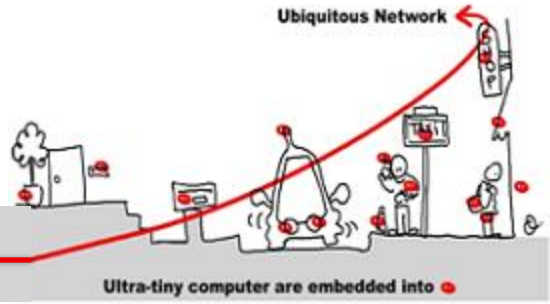
- Also Microsoft implementations
  - Micro .NET framework
  - .NET framework (.NET 4.0)
  - Included in Vista (WSDAPI)
- But…
  - For the moment, the 3 main implementations (SOA4D, WS4D, Microsoft) of DPWS do not communicate with other ones…
  - So everybody is a standard !

Comparing Implementations



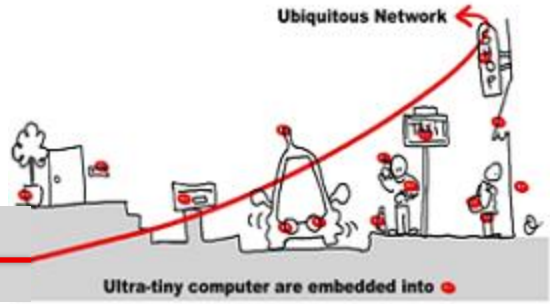
Stack \ Explorer	Java Explorer	WS-Management	Hello/Bye Event Catcher Micro .NET	Hello/Bye Event Catcher WCF
WS4D - JMEDS				
DPWS Core				
Micro .NET	Exception System.Xml.XmlException			
WCF			Hello/Bye message received	

UPnP vs DPWS

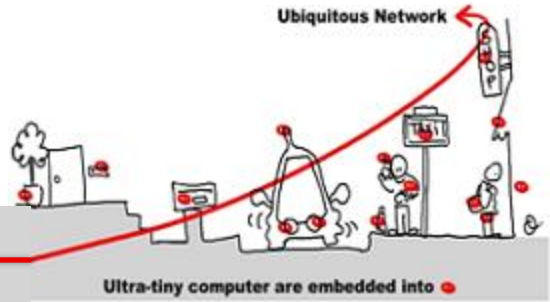


	UPnP	DPWS
Addressing	DHCP, AutoIP	DHCP, AutoIP
Discovery	SSDP	WS-Discovery
Description	UDA Schema	WSDL 1.1
Control	SOAP 0.9, 1.1	SOAP 1.2
Eventing	GENA	WS-Eventing
Presentation	HTTP, HTML	HTTP, HTML

References

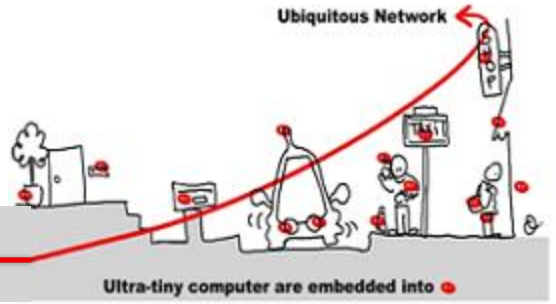


1. "**Service-Oriented Device Communications Using the Devices Profile for Web services**", F. Jammes, F., A.Mensch, H. Smit, in Proceedings of 21st International Conference on Advanced Information Networking and Applications Workshops, 2007.
2. "**Towards the UPnP-UP: Enabling User Profile to Support Customized Services in UPnP Networks**", T. *B. M. de Sales, L. M. de Sales, M. Pereira, H. Almeida, A. Perkusich, K. Gorgônio and M. A. de Sales*, in *Proceedings of the 2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies* (UBICOMM), pp 206—211, 2008.



MIDDLEWARE FOR COORDINATION OF UBIQUITOUS WEB SERVICES

Why Middleware using Services ?

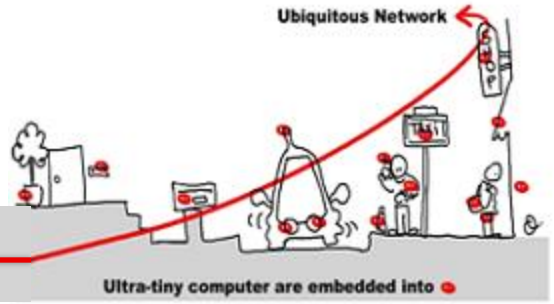


- Web Services support and simplify exchange of information
- Web Services for Device add dynamicity and reactivity

⇨So, add the possibility of context-awareness

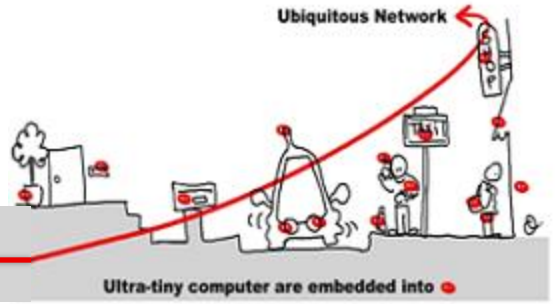
  – will improve how software adapts to dynamic changes influenced by various factors during the operation of the software

Examples of Middleware based on Web Services Coordination



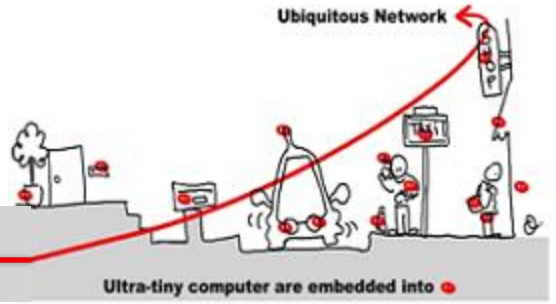
- Context-aware Web Service Middleware
  - COWSAMI
    - Based on Web Services
- Service Composition Middleware
  - WComp
    - Based on Web Services for Device

COWSAMI[7]



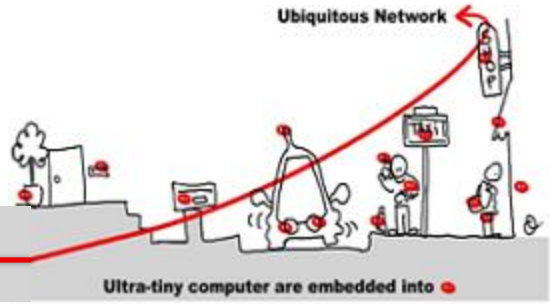
- Middleware infrastructure allowing context awareness[11] in open AmI environment
  - WSDL for describing
  - Compact SOAP as communication mechanism
  - SLP (Service Location Protocol)
- Context aggregator and interpreter
  - To integrate new context sources and users at runtime

WComp[10]



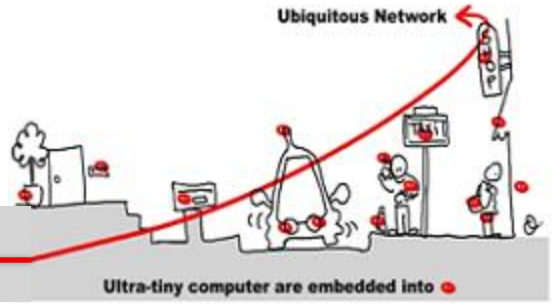
- Middleware infrastructure allowing dynamic structural application adaptation by service composition[9]
  - Based on an infrastructure of WS for Device (UPnP) dealing with dynamicity and reactivity
  - Composing WS for Device from infrastructure
    - Local composition and Distributed composition (see Tutorial 1)
  - Reactive adaptation using Aspect of Assembly

References

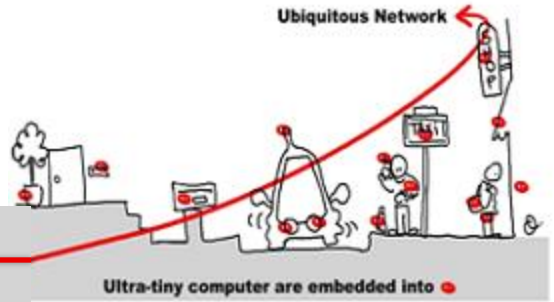


1. "**An Architecture to Support Dynamic Composition of Service Components**", D. Mennie and B. Pagurek, in *Proceedings of the 5th International Workshop on Component-Oriented Programming* (WCOP), 2000.
2. "**Semantics-based dynamic Web Service composition**", K. Fujii and T. Suda, in *International Journal of Cooperative Information Systems*, volume 15, number 3, pp 293-324, 2006.
3. "**Design and implementation of a device-independent system for adaptive composition of web services**", Q. H. Mahloud, W. Zahreddine, in *International journal of Internet protocol technology*, volume 1, number 3, pp 168-176, 2006
4. "**An ECA-based framework for decentralized coordination of ubiquitous web services**", J.-Y. Jung, J. Park, S. –K. Han, K. Lee, in *Information and Software Technology*, volume 49, issue 11-12, 2007
5. "**Building Home Monitoring Applications: From Design to Implementation into The Amigo Middleware**", W. Jouve, N. Ibrahim, L. Réveillère, F. Le Mouël, and C. Consel. in Proceedings of the 2nd International Conference on Pervasive Computing and Applications (ICPCA), 2007.
6. "**Context-Aware Service Composition in a Home Control Gateway**", *A. Bottaro, J. Bourcier, C. Escoffier, P. Lalanda*, IEEE International Conference on in Pervasive Services, pp 223-231, 2007.

References

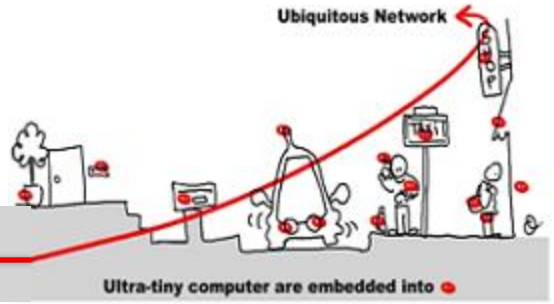


7. "**CoWSAMI: Interface-aware context gathering in ambient intelligence environments**", D. Athanasopoulos, A.V. Zarras, V. Issarny, E. Pitoura and P. Vassiliadis, in *Pervasive and Mobile Computing*, volume 4, number 3, pp 360-389, 2008.
8. "**Incontext: A pervasive and collaborative working environment for emerging team forms**", H.L. Truong, S. Dustdar, D. Baggio, S. Corlosquet, C. Dorn, G. Giuliani, R. Gombotz, Y. Hong, P. Kendal, C. Melchiorre, etal, in *International Symposium on Applications and the Internet* (SAINT), 2008.
9. "A Survey on Service Composition Middleware in Pervasive Environments", N. Ibrahim, F. Le Mouël, in *International Journal of Computer Science Issues* (IJCSI), volume 4, numéro 1, 2009.
10. "**Lightweight Service Oriented Architecture for Pervasive Computing**", J.-Y. Tigli, S. Lavirotte, G. Rey, V. Hourdin and M. Riveill, in *International Journal of Computer Science Issues* (IJCSI), volume 4, numéro 1, pages 1-9, 2009.
11. "**A Survey on Context-aware Web Service Systems**", H.-L. Truong and S. Dustdar, in *International Journal of Web Information Systems*, volume 5, number 1, pp 5-31, 2009.



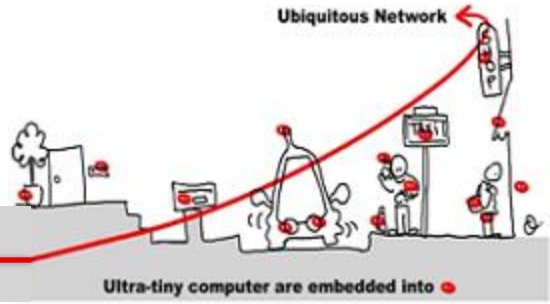
CONCLUSION AND CHALLENGES

Conclusion



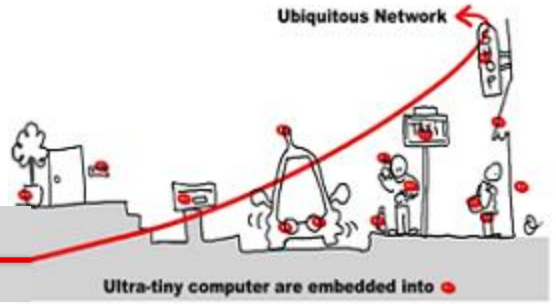
- Web Service for Device offers:
  - Dynamicity
  - Reactivity
  - Sharing information and actions
- Mature implementation: UPnP
- New standard: DPWS
- A lot of Ubiquitous Computing middleware based on Web Service approaches
  - Allows dynamic adaptive composition

Challenges



- Research & Discovery of available services
  - Cross-subnets (VPN, NAT: what about performances ?)
  - Cross-Internet (no support for Internet-scale discovery)
  - Cross-heterogeneous networks (Bluetooth, Ethernet, …)
  - Context-awareness (no support for rich queries over service descriptions)
- Description
  - Weak description language (lack details and semantics)

Challenges



- Heterogeneity
  - Heterogeneous discovery protocols
  - Heterogeneous middleware protocols (interoperability)
  - Application level heterogeneity
  - Data representation heterogeneity
- Interoperability:
  - Connecting heterogeneous systems at runtime
- Physical device
  - State full service
  - Should synchronize service's control