



This document has been prepared by the members of the ARTEMIS Strategic Research Agenda Working Group ("ARTEMIS SRA WG").

Neither the ARTEMIS SRA WG nor any person participating in this working group is responsible for the use which may be made of the information contained in the present document.

www.artemis-office.org

All rights reserved

© 2006 by





Strategic Research Agenda

Seamless Connectivity and Middleware



Strategic Research Agenda
Report of the Expert Group on
Seamless Connectivity and Middleware

Preface

ARTEMIS (*Advanced Research & Technology for Embedded Intelligence and Systems*) is a 'European Technology Platform'. This is a public-private partnership led by European industry with the goal to establish and implement a coherent and integrated European research and development strategy for Embedded Systems.

Embedded technologies are becoming dominant in many industries, such as communications, aerospace, defence, manufacturing and process control, medical equipment, automotive, and consumer electronics. This trend is likely to continue, given the ever-increasing possibilities for new applications offered by ever-advancing communications, embedded computing devices, and persistent storage.

Industries using and developing embedded systems differ significantly in business and technical requirements and constraints. Development cycles of complex industrial equipment, such as aeroplanes, industrial machines and medical imaging equipment, but also cars, are much longer than the development-cycles of other high-volume, cost-dominated devices for private customers, such as DVD players, mobile phones, ADSL modems and home gateways. Safety requirements are different for an aeroplane, for a car and for a mobile phone. Security, privacy and data-integrity all pose differing requirements in different environments.

Industry of a specific domain is increasingly confronted with the integration of requirements from other industries. In cars, for instance, there are not only the traditional safety requirements, but also the requirements from the consumer and mobile industry, with the increasing integration of entertainment and mobile communication into the

total system. We see little cross-fertilization and reuse over the different industrial domains, as segmentation of markets with their differing requirements has resulted in a fragmented supply industry and research field.

One of the main ambitions of Artemis is to overcome this fragmentation, cutting barriers between application sectors so as to 'de-verticalize' the industry, sharing across sectors tools and technology that are today quite separate and establishing a new embedded system industry that supplies tools and technology that are applicable to a wide range of application sectors.

Embedded systems usually do not operate in isolation, but are often used in combination with other systems to realize an overarching function. Such larger systems are referred to as 'systems of systems'. Examples are a digital television that is integrated in the digital home, a medical diagnostic device that is embedded in the hospital workflow environment, and a car that interchanges information with other cars in its vicinity in order to improve safety. Also the large infrastructural systems, such as the air traffic control system, the electric power grid and of course the telecommunications infrastructure may be viewed as complex systems-of-systems. These systems are characterized by large-scale networked integration of heterogeneous and often intelligent components.

In the extreme, we see the formation of 'sensor networks', even aggregations of 'smart dust', where completely new challenges have to be addressed, related to such different research areas as low power communication, energy scavenging, micro

devices, sensor and data fusion, and controlled emergence of system properties.

Given that the environment of open systems of systems cannot be controlled and specified, and thus cannot be completely modelled, it follows that the reliability and performance of a system may be compromised by unforeseen environmental behaviour. Consequently, we will also require new and so far unexplored approaches to safeguard the safety, security, reliability and robustness of the embedded systems of the future. The use and integration of off-the-shelf components is also a challenge, as such components were not designed from the perspective of the decomposition of the system at hand.

The changeover from design by decomposition to design by composition raises some of the most challenging research and development questions in the embedded systems domain today.

This change, and the ambition for cross-sectoral commonality, inspires much of the specific research proposed in the Artemis Strategic Research Agenda.

The ARTEMIS Strategic Research Agenda (SRA), published in June 2005, outlines the objectives and the research topics that need to be investigated in the field of embedded systems. This present document is one of a trio of documents that amplify that original SRA with more specific research priorities. These three parts of the 'full SRA' are concerned with:

- Reference Designs and Architectures,
- Seamless Connectivity & Middleware, and
- System Design Methods & Tools.

The Reference Designs and Architectures SRA establishes common requirements and constraints that should be taken into account for future embedded systems, and will establish generic reference designs and architectures for embedded systems that can be tailored optimally to their specific application context.

The Seamless Connectivity & Middleware SRA addresses the needs for communication at the physical level - networks; at the logical level - data; and at the semantic level - information and knowledge. Middleware must enable the safe, secure and reliable organization - even self-organization - of embedded systems under a wide range of constraints.

The Systems Design Methods and Tools SRA sets out the priorities for research into the ways that these systems will be designed in future so as to accommodate - and optimise the balance in achievement of - a number of conflicting goals: system adequacy to requirements, customer satisfaction, design productivity, absolute cost, and time to market.

Each part of the SRA was produced by a group of experts that devised their own method of working. While the three Expert Groups liaised so as to achieve coverage and avoid inconsistencies, each of the three documents has its own structure and style, with this preface being the only common element.

All three of these parts of the SRA are living documents that will be continuously refined and updated as research results arrive and as the technological and societal environment changes during the coming years.

Table of contents

1.	Introduction	6
1.1	THE AIM OF SEAMLESS CONNECTIVITY AND MIDDLEWARE	6
1.2	DOMAIN CLUSTERING	6
1.3	METHOD OF WORK	7
1.4	HOW THIS SRA SHOULD BE USED	7
2.	Industrial landscape	7
2.1	INDUSTRIAL	7
2.1.1	<i>Automotive</i>	7
2.1.2	<i>Avionics</i>	9
2.1.3	<i>Manufacturing</i>	10
2.1.4	<i>Medical</i>	12
2.1.5	<i>Security & Defence</i>	12
2.2	PRIVATE	14
2.2.1	<i>Technology trends in home architecture</i>	14
2.2.2	<i>Multimedia / Consumer electronics view</i>	15
2.2.3	<i>Personal privacy domain</i>	15
2.2.4	<i>Energy management and energy saving</i>	15
2.2.5	<i>Health</i>	16
2.2.6	<i>Connectivity at home</i>	16
2.3	NOMADIC	16
2.4	PUBLIC	17
2.4.1	<i>Energy - Power grid</i>	17
2.4.2	<i>RFID & Sensor networks</i>	18
3.	Multi-domain analysis - Domain clustering	19
3.1	THE REASON FOR CLUSTERING	19
3.2	HOW MULTI-DOMAIN ANALYSIS WAS CONDUCTED	19
3.3	MAIN CHARACTERISTICS OF A SYSTEM: QUANTITATIVE AND QUALITATIVE DIMENSIONS	20
3.4	CLUSTER DESCRIPTION	21
3.4.1	<i>Critical cluster</i>	21
3.4.2	<i>Device & Plant cluster</i>	22
3.4.3	<i>Private cluster</i>	23
3.4.4	<i>Nomadic cluster</i>	24
3.4.5	<i>Ad Hoc Connectivity cluster</i>	25
3.4.6	<i>Systems of Systems cluster</i>	26

4.	Research priorities	28
4.1	THE MIDDLEWARE PROGRAMMING MODEL	28
4.2	SYSTEM ORGANIZATION AND DEPLOYMENT	29
4.2.1	<i>Dynamic reconfiguration capabilities</i>	29
4.2.2	<i>Efficient user interaction</i>	29
4.2.3	<i>Device and service discovery</i>	29
4.2.4	<i>Ontologies</i>	30
4.2.5	<i>Conflict resolution</i>	30
4.3	RESOURCE MANAGEMENT	30
4.3.1	<i>Adaptive resource management</i>	30
4.3.2	<i>Application end-to-end QoS</i>	30
4.4	DATA DISTRIBUTION	30
4.5	ROBUSTNESS AND SUPPORT FOR DIAGNOSIS	31
4.5.1	<i>Error containment in a distributed and dynamic system</i>	31
4.5.2	<i>Global connectivity</i>	31
4.5.3	<i>Interoperability and connectivity in heterogeneous environments</i>	31
4.5.4	<i>Connectivity in constrained environments</i>	32
4.5.5	<i>Provably correct systems</i>	32
4.5.6	<i>Operating Systems & middleware for safety critical systems</i>	32
4.5.7	<i>Designing and integrating provably correct systems</i>	32
4.6	SECURITY	32
4.6.1	<i>Security of the execution platform</i>	33
4.6.2	<i>Authentication / authorization</i>	33
4.6.3	<i>Proof and Audit</i>	33
4.6.4	<i>Threat identification</i>	33
4.6.5	<i>Security management</i>	33
4.7	IMPORTANCE FOR EACH DOMAIN CLUSTER	33
5.	References	33
6.	Contributors	35

1. Introduction

The vision driving ARTEMIS is a major evolution of our society in which most systems, machines, and even ordinary objects will be transformed into digital, communicating, information processing, self managed resources. These transformations will be possible through advances in embedded systems technology and its large-scale deployment, not only in industries and services, but in all areas of human activity.

“Seamless Connectivity & Middleware” along with “Reference Designs and Architectures” and “Design Methods & Tools” have been identified as the three main technological chapters that should be developed by ARTEMIS to support the development of high value-added embedded systems.

Reference Designs and Architecture describes common requirements and constraints that should or could be taken into account for future embedded systems, and prepares the explicit issuing of designs and architectures in application domains. *Design Methods and Tools* asserts how these systems are going to be designed in the future, in order to achieve a number of conflicting goal: system adequacy to requirements, system optimization, design productivity, and eventually time to market.

1.1 The aim of Seamless Connectivity and Middleware

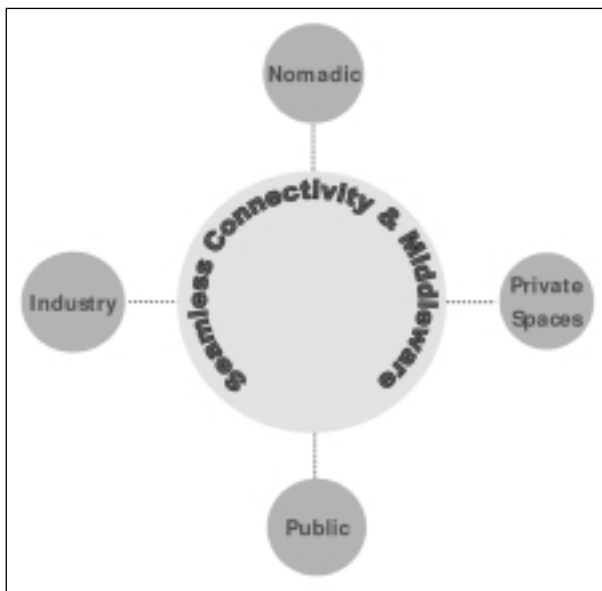


Figure 1 Seamless Connectivity & Middleware and 4 applications contexts

However, embedded systems will not exist if they are not embedded into products. This means in the future being part of wider distributed systems, or being themselves distributed systems. In a more or less dynamic way,

systems will be able to join an already existing population of communicating systems, and to become consumers, providers, or part of services to users.

So, markets will not open up if *basic technology* implementing *common standards* are not available for making *seamless connectivity* possible. Seamless connectivity means the possibility to exchange at the physical level - networks - at the logical level - data - and at the semantic level - information, and knowledge. Moreover, embedded systems must be organized or must organize themselves to achieve generic functions, and be able to do so under constraints and various events, e.g. they must be safe, reliable, must make diagnosis and maintenance easy. Once networking technology is available - and wireless networking will play a major role in the future - all this is ensured by adequate *middleware*.

So, *Seamless Connectivity and Middleware* is at the heart of the future embedded system markets, and for products including embedded systems. In fact, if ARTEMIS can provide Europe with adequate technology and standards in this area, new yet unforeseen possibilities of application will be made possible, and this will create a rich environment for creativity, innovation, and eventually growth, and high skilled jobs. It is not exaggerated to say that the impact will be even stronger than the Internet almost 15 years ago.

1.2 Domain clustering

However, if we look at the versatility and the complexity of application domains, which are grouped within ARTEMIS in the four application domains: Industrial Systems, Nomadic Environments, Private Spaces, Public Infrastructure, it is clear that this is not a simple, straightforward task. Indeed, it seems that there is not a single middleware design that can fit requirements from all those domains. Moreover, in the current situation, specific, partial technology exists, and is more or less deployed in application domains, and partially standardized in standards specific to application domains.

However, one feels that technology and standard development effort could be shared among domains; more importantly, technology *should* sometimes be shared, because it corresponds to products that will *connect*; there even exist cases where products will be developed only if some *common infrastructure* is shared among markets that are separate today. In this case, common technology and standards is absolutely mandatory.

This leads to *domain clustering*. Domain clustering aims at exhibiting a limited number of “abstract domains”, that share common characteristics of (part of) the application domains. Some clusters correspond to “could share” characteristics, others to “should share”. Domain intersections are also stressed.

1.3 Method of work

This is what the working group preparing this Strategic Research Agenda synthesized. Application domains were examined separately, and experts from the corresponding industry tried to express the kind of system that they expect to become products in the future. This analysis is only partial and incomplete, because the future is obviously blurred, and because it is not possible to be too precise in future product development. Industrial experts were asked to express when possible future systems in a simple, common “qualitative” vocabulary, that we called a multi-domain abstraction. This aims at making domain clustering easier.

If the analysis is correct, then domain clusters show important characteristics that:

- call for developing a common middleware within ARTEMIS for this cluster
- from R&D in attached Research Priorities
- towards technology standardization.

The last step consists of defining those research priorities. Research Priorities are presented, and were also discussed, according to their scientific and technological intrinsic coherence. They are put in coherence with the former analysis, by showing how they contribute, or do not contribute to clusters.

Indeed, the process driving to this document was not that straightforward. We started from several points at the same time, and through a series of interactions between topics, we converged on the current version.

1.4 How this SRA should be used

It is clear that the analysis that led to the present document was necessarily limited in time and resources, and that future work will be required to confirm or refute the conclusions, to check and refine them.

However, these conclusions will not become reality unless industry from various sectors agrees to cooperate on some cluster development goals. It seems essential for ARTEMIS to strongly support discussion between industry from various domains, together with researchers, to develop in a cumulative and efficient way common middleware technology and standards. This is the very essence of ARTEMIS.

Once consensus on domain clustering is reached, the final issue concerns the strategy for ensuring success of technology on markets. Various strategies can be pursued, from open source development to selling proprietary middleware implementation. This discussion could not be undertaken by the group within its limited lifetime. It remains to be conducted in a consolidation process with the other SRA groups, and innovation group, and even probably beyond.

2. Industrial landscape

This section illustrates how experts from the application domains involved in ARTEMIS see future products. This is the first step that provides basic information and knowledge for multi-domain analysis and domain clustering in next section.

2.1 Industrial

2.1.1 Automotive

Car basic functions

In today's automotive electronic systems, the distributed Electronic Control Units (ECUs) of each federated cluster of the car are interconnected via communication networks with different protocols (e.g., Controller Area Network (CAN), Local Interconnect Network (LIN), Media Oriented System Transport (MOST)), physical layers, bandwidths (10 kbps-25 Mbps), and dependability requirements. A typical automotive system structure is depicted in Figure 2. Multiple clusters are connected via a central gateway allowing data exchange and access to the On-Board Diagnostic (OBD) system of each ECU. The comfort clusters as well as the powertrain cluster are typically implemented via the CAN protocol. The multimedia cluster is frequently based on a protocol with support for streaming audio and video (e.g., MOST), while the passive safety clusters either use CAN or vendor-specific communication protocols such as byteflight. Each of these clusters consists of components designed according to the “one function - per ECU” principle in order to simplify system integration and ensure intellectual property protection. Consequently, additional ECUs need to be added to the clusters in order to improve the functionality of the car. However, this trend of increasing the number of ECUs is coming to its limits, because systems are becoming too complex and too costly with the current practice of having each ECU dedicated to a single function.

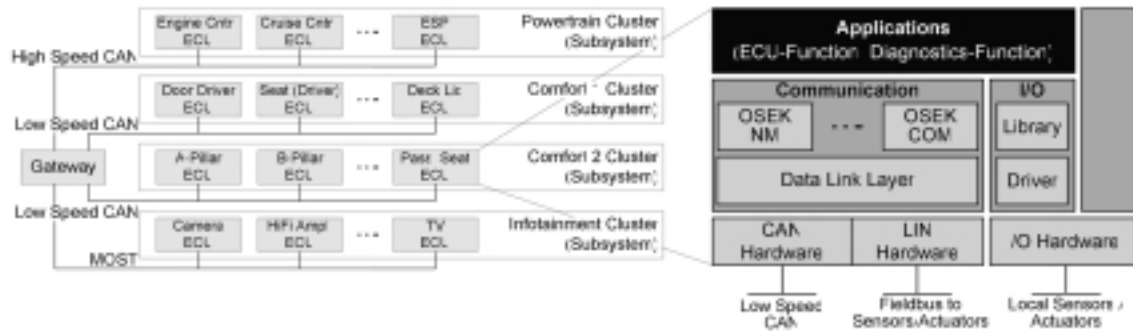


Figure 2 Structure of Automotive Electronic System

The physical system structuring can be expressed in terms of the multi domain abstraction model (see Section 3). The system encompasses the electronics of the complete car. Each cluster (e.g., body control, infotainment) represents a subsystem, which provides an identifiable function from the point of view of the product. Each of the ECUs provides an application service, which is composed into the application service of the cluster. The composition of the application services of all clusters leads to the application service of the complete car, including emerging services (e.g., transportation service, entertainment, comfort). Within each ECU, middleware services provide application-independent interfaces for the infrastructure services (e.g., communication services, network management, and operating system services in compliance with Open Systems and the Corresponding Interfaces for Automotive Electronics (OSEK/VDX) specifications¹).

Driven by the development of integrated architectures, such as the automotive industry initiative AUTOSAR (www.autosar.org) or European funded projects like EAST-EEA, EASIS and DECOS, the system structuring of automotive systems is about to change. In contrast to federated architectures, integrated architectures share networks and nodes among different application services in order to evolve beyond a “1 Function - 1 ECU” strategy. For this purpose, it is necessary to develop a distributed execution platform that allows the integration of software-modules from different sources (vendors) and with different criticality levels. This integration depends on effective partitioning - not only between ECUs but even at the level of a single hardware unit (ECU). Integrated ECUs will support multiple application services sharing the available communication and computational resources.

Changes in the multimedia cluster

Technologies that have emerged in the computing and consumer electronics markets have the potential to enhance vehicle applications by providing access to a wider range of system resources, for example high speed internet connection via a 3G telephone. Future vehicle infotainment scenarios that exploit these technologies will require the simultaneous access to a number of mobile devices that build ad-hoc networks with the built-in devices (e.g. via

Bluetooth or Ultra Wide Band) and utilize network-based services to connect the vehicle to external services as well as enabling vehicle-to-vehicle communication. Making network-based resources available to vehicle functions will result in significant added value and differentiating features such as up-to-date traffic and weather information accessed via the Internet. Due to the fact that these services will be based on wireless communication, portable devices etc., they may not always be available. A method of dynamically integrating these resources into the vehicle system and notifying appropriate applications of their availability (or otherwise) is therefore required.

The need for simple, intuitive user interfaces highly integrated with the car environment, requires a platform that is capable of hiding the inherent complexity of the underlying resource management from the user whilst nevertheless enabling advanced features and making optimal use of resources as and when they become available. For example, internet bandwidth available to in-vehicle applications should be increased as soon as the car enters a WLAN hot spot and services capable of exploiting the bandwidth automatically activated, such as the downloading of navigation “points of interest” updates. These application scenarios impose very high demands on the configurational flexibility and scalability of the systems which cannot be fulfilled by current state-of-the-art vehicle electronic architectures. In order to meet these challenges, mechanisms of dynamically self-configuring systems as well as dynamic resource discovery and management capabilities will need to be added to automotive middleware.

Intelligent systems for safety

The area of integrated and intelligent safety systems is set to grow rapidly in the coming years, and as a part of this, these systems also have to interact with various surrounding systems.

Some examples where the connectivity issues are important are:

- Warnings from road infrastructure to vehicles, e.g., slippery surface, queue, construction work, accident. Surface info can be used to calibrate ACC distance, ESP, etc.
- Warnings from vehicle to vehicle, e.g., blind spot notification.

¹ <http://www.osek.vdx.org/>

- Automatic download of navigation data, e.g., new maps. In terms of interaction with systems in the vehicle and thereby requiring middleware functionality, some of the main issues are:

- Reconfiguration in case of hardware failure
- Handling of different wireless communication technologies (guaranteeing latencies, reliability, security etc).
- Automatic upgrade of software (e.g., self-healing, self-diagnostics)

Local and global traffic management

Via communication between cars, traffic information will be propagated through the network of cars. Since the time constraints of those services are lower compared to inter-vehicle hazard warning, traffic management applications can be deployed even by lower penetrations. By complementing this application with cellular networks, a highly efficient application can be realized. For the proximity of the cars, the ad hoc network is used to propagate traffic information by sending it over multiple cars, whereby each car aggregates the data with its own data base. If no other car for data propagation is in reach, the aggregated traffic data will be sent to backend servers either via cellular networks or via road side units. By combining cellular networks, access to backend servers via road side units, and the aggregation of traffic data with vehicular ad hoc networks, some principle drawbacks of communications systems can be solved by using the other systems. E. g. the problem of scalability of cellular networks is solved by sending already aggregated traffic data over cellular networks and thus reducing the number of transmissions. The problem of ad hoc networks, especially in the first phases of market introduction, that only a few equipped cars are available, will be solved by relying on cars equipped with cellular mobile systems.

Communication between car and road signs

When deploying car-to-car communication systems, road side units will be a substantial part of the vehicular ad hoc systems. Road side units can be used as data relay and for storing traffic and hazard data. Furthermore, road side units will be installed on traffic lights to improve traffic flow at crossings and to improve traffic safety at these spots. In further deployment steps, it is possible to even equip traffic signs with road side units. These equipped traffic signs will send their message to the drivers driving by. Obviously, this application is especially important in case of temporary traffic signs, e.g. at roadwork sites. In some countries it is even in discussion to equip pedestrians with sending devices to warn drivers when pedestrians cross the roads at crossings equipped with traffic lights and road side units.

Interoperability with other domains

While it is widely accepted that on-line services will bring unparalleled benefits to drivers, the market - with roughly

35 000 active subscribers - is still almost non-existent in Europe today.

To unlock this market it is necessary to achieve the transition from closed, proprietary systems to open, IP-based systems. Open systems immediately raise the issue of security as openness implies increased opportunities for undesired access to data and functionality - these need to be protected using an end-to-end perspective.

2.1.2 Avionics

The state of the art in avionics embedded systems is the IMA concept (Integrated Modular Avionics), based on the principle of distribution of the resources and functional integration. With the objective to enable advanced management of the resources for the implementation of the aircraft application functions, it is already definitely beyond the conventional "1 Function - 1 Unit" strategy, also allowing the integration of software-modules from different sources. It has become a standard for the most advanced avionics solutions, and has entered into serial production in civil aeronautics since the early 2000's, as illustrated in figure 3.



Figure 3 State of the art IMA modules

A standard IMA architecture is depicted in Figure 4 and relies on:

- Standardized avionics modules having high computational power with off-the-shelf microprocessors, fast I/O processing capabilities to link to real time field applications through buses like A429, or CAN and operated under standardized OS and middleware applications.
- A high bandwidth communication network (AFDX) that enables deterministic time triggered communication between the modules (100Mbps) and off-line data loading.
- BITE systems (Built In Test Equipment) to monitor the physical health and the functional performance of the modules.

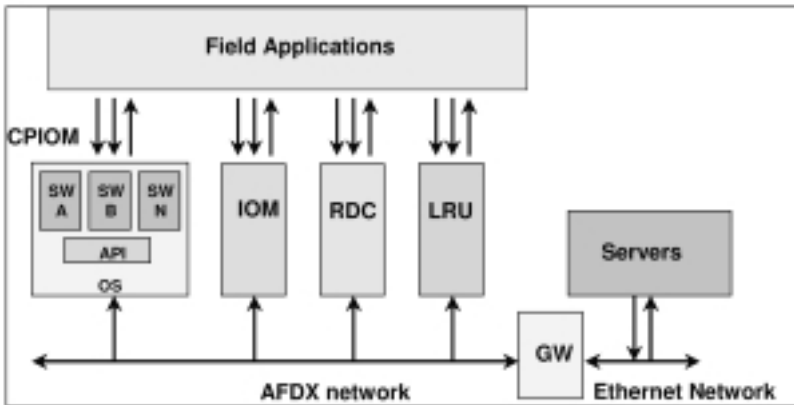


Figure 4 Principle of IMA architecture

The connectivity and middleware solutions currently used in the Core Processing I/O Modules (CPIOM) are illustrated Figure 5.

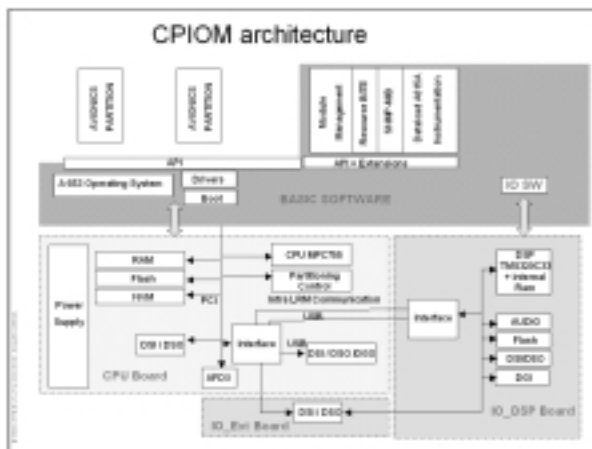


Figure 5 Core Processing I/O Module architecture

The evolution of avionics in the next decade will be driven by the following needs:

- The market demand for total quality of service in the context of an increase of the number and the complexity of the functions to be embedded, particularly health monitoring and predictive maintenance, adaptive robust performance and reconfiguration.
- The industrial requirement to manage obsolescence and enable easy upgrade of all embedded components - hardware and software components independently - all through the end-product life-cycles (20 to 30 years) .

The IMA architectures will tend towards:

- More performance of the individual components and network communication capabilities (towards 10 Gbps), also possibly based on new communication media such as Wireless and Power Line Communication

- More standardization and modularity of the physical layers in the modules, with further physical integration with the appropriate power and thermal management capabilities
- Enhanced middleware capabilities such as database storage, video streaming and graphical computation, but also middleware services to provide application-independent interfaces and to enable dynamic reconfiguration and provide more flexibility in physical resource sharing
- More functional integration, towards the global management of the distributed execution of complex functions, including dynamic reconfiguration

The interface of avionics with the Open world will tend towards adaptive network flow management taking into account security and availability issues and new COTS technologies such as Wireless IP communication.

The evolution in the connectivity and middleware technologies will be supported by parallel evolutions in the development environments towards composability and systematic design, while managing complexity and maintaining the highest levels of quality and dependability assurance.

2.1.3 Manufacturing

A manufacturing system of today is schematically designed in layered approach consisting of three tiers: device, control and enterprise. This is valid for both process industry and discrete manufacturing, but there are differences in the requirements and in typical topology. The device tier is the bottom layer, which performs the real-time control and monitoring of the plant or factory production equipment. It is populated by field devices like instruments e.g. temperature and pressure transmitters for sensing, motors, valves and pumps for actuation and controllers for control. The layer is characterized by high requirements on real-time performance in a millisecond to second time scale, robustness and availability. The control is typically performed by programmable logic controllers (PLC) or distributed control system (DCS) type of controllers. The field devices and the controllers are interconnected using a fairly wide selection of digital field buses (Profibus, Foundation Fieldbus and DeviceNet being among the popular choices) or still commonly today using digital signals and dedicated analog signal wires operating on 4-20mA. The analog signals can have superimposed digital information (HART).

The next layer, control, constitutes functionality like database management and human machine interface of the controllers. Although this level also has real-time characteristics it is more relaxed than on the device level. At the top of the hierarchy is the enterprise level which provides execution control and monitoring of the whole production process; what, when and how to produce. This level is characterized by business oriented production scheduling and planning with system requirements similar

to financial and administrative systems operating in a minutes to days/weeks time scale.

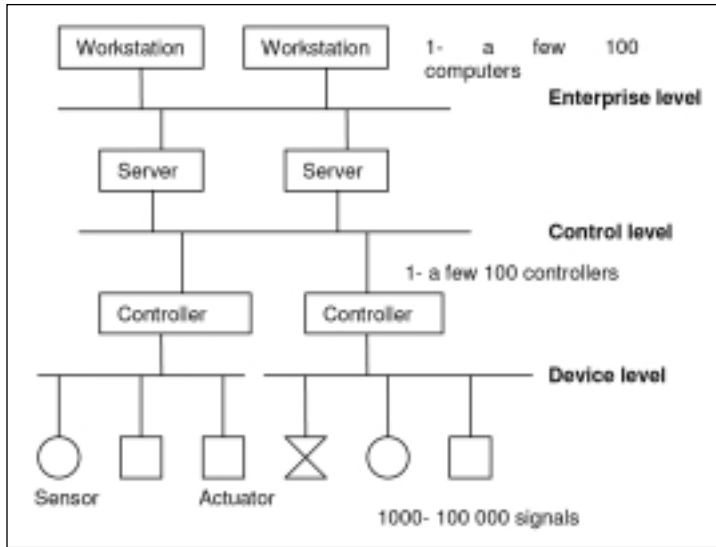


Figure 6 Typical process plant structure

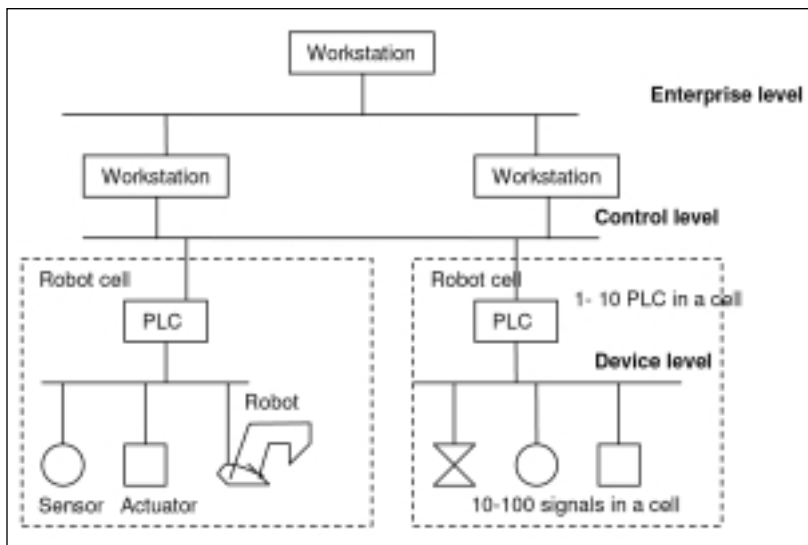


Figure 7 Typical discrete manufacturing plant

A special part of device level communication in factory automation is motion control. This is e.g. necessary in axis control of robots. The requirements here are much stricter on the real-time requirements as millisecond or sub-millisecond synchronization of actuators are needed.

Trends that challenge the traditional design are:

- Field devices are offering more and more advanced functions and are becoming more and more complex. The devices produce data and information sought not only within a tier but also across tiers.
- The fierce competition on a global market requires an increased ability to quickly respond to the market, which leads to higher demands on responsiveness, flexibility and quality control. The design of the manufacturing system must accordingly reflect these requirements to

provide the necessary flexibility and increased need for information from the shop-floor to the board room.

- Production control and monitoring system on all levels are becoming more and more sophisticated as a part of the ever-ongoing process of increasing productivity and cost effectiveness. The system requires constantly more and more information from the production process.
- Ethernet and wireless based communication is used on enterprise level and partly on control level today, and are starting to penetrate also the device level communication.

As the field devices are the basic building block of any manufacturing system it is necessary to develop technology for these devices to meet future trends and requirements. These will include demands on efficient

development and connectivity such as development tools, methodology, middleware and communication interfaces on all protocol levels. Some areas that need to be addressed include:

- Efficient development of intelligent field devices, design tools and methodologies as well as platforms for hardware and software components.
- Wireless communication, for increased flexibility and increased information need. This comprises addressing fulfillment of requirements of real-time response, safety application, usage in explosive area, low power, batteries for years of operation and low cost.
- Middleware for distributed control functions and easy data exchange within and across tiers. As the device level will continue to be a mix of a variety of different field buses and upcoming Ethernet and wireless solutions, the middleware solution must integrate such heterogeneous solutions.
- Integration with top tier application through remote services based on internet technologies like HTML, XML and web services.
- Security must be an integrated part of a plant network with solutions that can be scaled according to the hazards involved and the cost of installing and managing the infrastructure of hardware and software security protection. There is also a big need for empirical research to determine real-life risk and efficient counter-measures.
- Safety layers must be developed for usage on top of all relevant middleware solutions for usage when the control system is protecting human lives and the environment.
- Common communication infrastructure which can carry control as well as administrative traffic
- Usage of standard Ethernet communication in hard real-time motion control.

As manufacturing plants are designed to be agile, the horizontal upstream integration of the plant itself with the product design process and the supply chain becomes vital. The downstream process of logistics is intimately involved in getting the products to market and must therefore be closely connected to the manufacturing process. Research into the best solutions also for these areas are necessary to achieve the goal of the agile manufacturing plant capable of producing batches of one at the cost level of mass production.

2.1.4 Medical

The medical world is relying more and more on networks. Wide area networks permit the exchange of medical information between, on one side, the patient at home and on the move, and on the other side, the hospital or the call centre. Local area networks exchange information within the hospital for example for monitoring and telemetry. Personal area networks interconnect devices, infrastructure or functions in for example an MRI² modality. Finally the body area network is used to interconnect sensors and follow the patient during

transport in the hospital. The medical world is not the driver of connectivity standards but is very much confronted by the interoperability problem coming from the many connectivity solutions coexisting in the hospital. At this moment efforts are under way to promote IEEE 1073 standard, medical connectivity, inside standards like Bluetooth, Zigbee, and possibly UWB.

An all-pervading problem is the aspect of security and privacy in its many forms. A better understanding and modelling of the privacy aspects should lead to a better understanding of the security mechanisms needed in the heterogeneous network of the hospitals.

An important issue is to make existing standards hospital, and medical proof. For example IEEE 802.11 cannot be relied upon. Even switching on a microwave may perturb the communication significantly. Bluetooth, Zigbee and IEEE 802.11 mutually influence each other.

Many of the hospital connectivity uses can be extrapolated into the home environment. Again the additional requirements concern the reliability of the communication and the privacy aspects.

We see a clear direction into wireless communication, as exemplified in the movement of patients, and in wireless nodes performing body measurements. Actually, wireless sensor technology may allow patients to be more mobile and have longer periods out of hospital than is possible today.

2.1.5 Security & Defence

The important trend in security and defence systems is improving the situational awareness by moving towards a network centric approach; see Figure 8 for a typical example in a land-based situation. The challenges in this approach are:

- Manage the available information in a controlled manner, i.e. the right information in the required place, at the right time, with the right quality.
- Being able to operate between different providers of equipment (openness) and versions of equipment (new technology insertion).
- Modify the set-up depending on the mission or situation (scaleable, flexible....).

² Magnetic Resonance Imaging

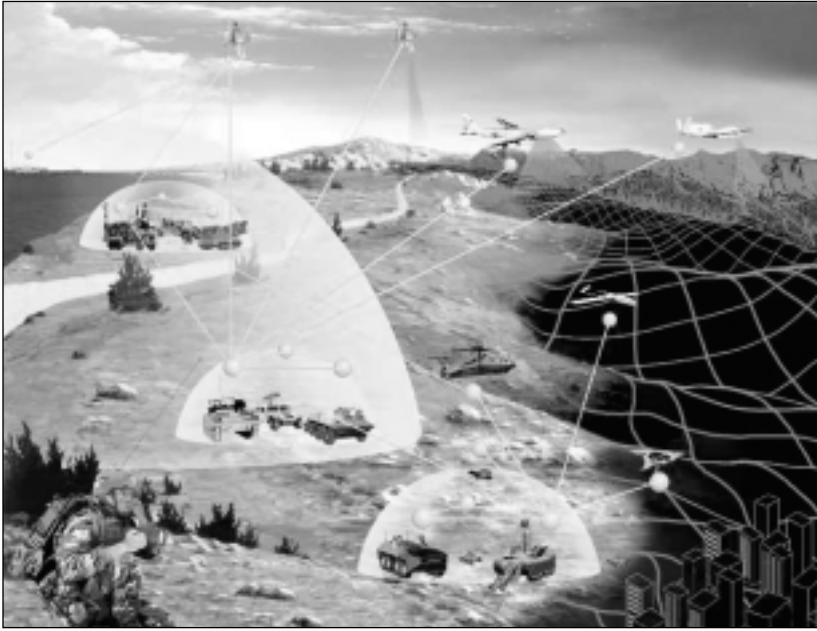


Figure 8 Future network centric battle space management

Typical defence system set-up

A typical set-up in a defence (local) command and control system is depicted in the figure below. The global approach is to connect sensors (radar, optical, support RF sensors, etc), display (operator consoles), platform, communication and actuators (weapons) in a local area network. Information is gathered if needed from different sensors (data fusion) and is used in order to present a situational surface and air picture to the operator. The same information can also be used for actuators. The same challenges as mentioned above are also applicable in this local system.



Figure 9 Typical set-up of defence system

In this kind of system, middleware software is typically used to support different frameworks (CORBA, Java/Openwings), provide generic access to data, and support this from any language. By doing so, the middleware shields the application SW from the platform, that is the operating system on top of the HW and network..

Radar sensor system

Zooming in one step further we arrive at the sensors in the system. A typical radar sensor is composed of major components, which interact via interfaces for data, control and timing/synchronisation. The main objective of the system is to measure responses on own or secondary transmitted signals at RF. Interaction

with the system is mainly through a data and control interface, which is connected to the local area network of the global command and control system, discussed before.

The radar sensor itself is composed of sub-systems like the antenna sub-system; processing sub-system, drive control system, cooling sub-system etc. Each sub-system on itself is also composed of sub-systems containing for example computer cabinets or racks, which perform a specific sub-function.

Continuing this decomposition we arrive at the PCB board level or system on chip level. An example of the latter is for example the digital processing part close to the receiver in the antenna. Such a system can be implemented on an FPGA which contains DSP modules,

microController(s), memory etc. A typical radar sensor contains between tens and hundreds of processors. The functions considered are digital beam forming on multiple receiver channels, calibration, interference suppression etc.

Inside these radar sensors the use of middleware has also been introduced in order to facilitate decoupling between the application software and the platform. The big advantage of this approach is that new technology can be rapidly inserted and for the SW developer the data and control interface functionality (in terms of data interfacing and data distribution) has a standard interface (API) on all the platforms.

In the radar sensor set-up there is a clear need for a middleware set-up which is tailored for the domain. Close to the antenna receivers the data rates are enormous (many channels, aggregate data rates of Giga bytes per second). This gives clearly different requirements to middleware and middleware services compared to the back-end processing of the sensor. In the back-end processing the data rates are lower but the variety and complexity of the data types much more complex.

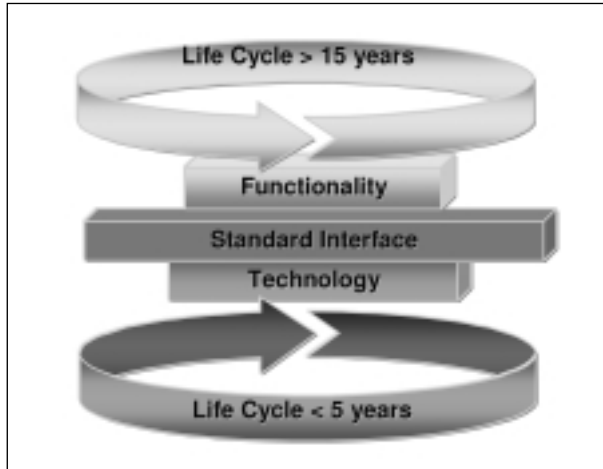


Figure 10 Middleware as the solution to decouple application software from the hardware platform

2.2 Private

2.2.1 Technology trends in home architecture

In this section we take the “home domain” as an example of private spaces. Home domain has specific requirements due to the fact that in the past it has been a closed environment. However, due to the technology disruption showed in Figure 11, the home architecture is moving into the new concept of End-to-End (E2E) architecture. The classic concept of E2E architecture involving terminal, network and service segment components is changing. It consists more of (Device-to-Device) D2D architecture issues. Therefore, the Home domain differentiates two aspects of the E2E architecture: whether the device is located in the home boundaries (i.e. the focus is on D2D

apart from some intermediate transcoding or interworking component) or whether the device is accessing the home resources from outside. Thus, the E2E architecture should include the connectivity at Home but also E2E connectivity when accessing Home devices or services remotely. Thus, access networks and intermediate servers or gateways (i.e. Residential or Home Gateway) are considered in order to define the E2E architecture.

For Seamless Connectivity and Middleware the cross-domain interoperability, especially from the context point of view, has been identified as one of the most important goals. Therefore, it is important to understand not only the contexts themselves, but also what happens when the user moves from one context to another: these “interfaces” between the contexts often create problems for users and therefore offer opportunities for technology solutions. For example:

- Commuting from work to home
- Leaving work mode for leisure
- Work mode continues while physical place changes, i.e. taking work along to the train
- Doing home related things at work

In the home we see a convergence of the mobile and home domain. The mobile terminal may well be one of the driving forces behind the (promised) emergence of home networks. One scenario is that the mobile terminal enters the home network with its WiFi interfaces and most probably uses the UPnP and DLNA standards to communicate with other devices in the home (Figure 12).

The largest load on the home network will be audio video streaming and downloads of large files. The audio video streaming needs to guarantee quality in accordance with the receiving device, which is a real challenge given the convergence of home and mobile leading to an ever-diverging set of screen characteristics.

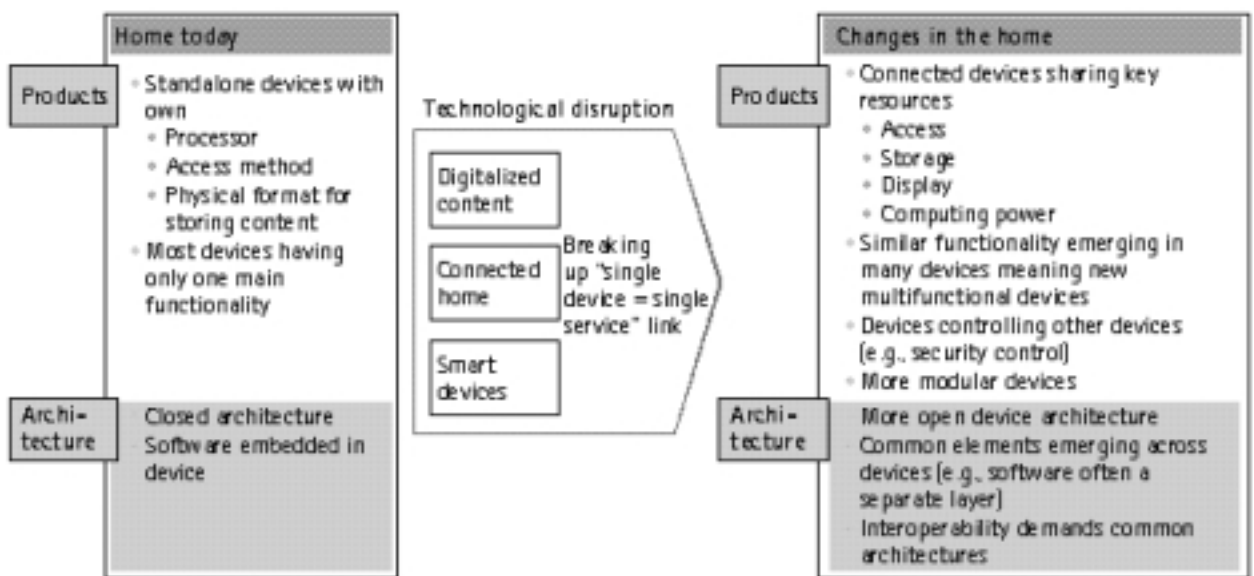


Figure 11 Technology trends leading the formation of Home architecture



Figure 12 Consumers want to share content between their mobile devices, computers and home electronics

2.2.2 Multimedia / Consumer electronics view

A multimedia system comprises a set of interconnected devices such as telephones, PDA, DVD, PC, Media-centre, and TV-sets. These devices recognize each other, provide security and privacy domains, determine their membership, and compose their logical connections as function of the exchanged information. Important aspects are the seamless establishment of the connections and quality control by the users with a minimum number of user actions. All necessary technology should be hidden from the user who will manipulate idealistic entities that are close to his personal perception and understanding.

The home environment provides connected access networks, content providers and telecom capability connected to the home via a telecom service provider. The home environment can connect to a large set of global service- and network- providers.

The system configuration is rather dynamic, with people and equipment entering and leaving the home.

Wireless transmission today leads to large quality fluctuations in transmission quality and they happen within milliseconds. The encoding of the video leads to large fluctuations in CPU time per frame at the frame level (tens of milliseconds). The CPU has a capacity which limits the frequency and size of the video that can be rendered without disturbing artefacts.

2.2.3 Personal privacy domain

Today, people install wireless networks because it is offered with their ADSL connection. The currently popular

IEEE 802.11g standard provides methods to encrypt the transmitted data, but few people understand the importance of this. However, understanding will come automatically when their neighbours start using their infrastructure. Once the channel is protected against eavesdropping, the protection of data from other people legally connected to the home network may prove more difficult to guarantee. Guests need to be identified, groups within the family, personal privacy domains and other outside players. The mechanisms for group identification and privacy enforcement are available today.

However, more research is needed to determine whether these are the right mechanisms and whether users understand what they are manipulating. A logical consequence is that ownership of device, function, storage space, content and other concepts should be expressed in standards like UPnP and DLNA or other future standards.

2.2.4 Energy management and energy saving

Modern buildings and homes are being fitted with increasingly sophisticated systems, based on modern communication technologies and network controllable things like air conditioning units, lighting dimmers, motorized shutter and blinds or white goods. The trend is an increased demand for comfort, security, openness to the world of services available in the internet and energy efficiency.

Focussing on energy efficiency, it is clear that substantial energy savings may be achieved through coupled control of things from different domains. For example, consistent control of HVAC units, lighting and shutters can in some cases avoid waste of energy. This means that a higher functional integration between traditionally independent sub-domains is strongly desirable to achieve global energy efficiency optimization.

For the building/home control system this means that the control software architecture should probably be better seen as a set of collaborative, network-neutral, possibly distributed and migrating functions rather than a set of interconnected, domain centric and fixed functionality devices.

This strongly pushes for a middleware technology that is consistent across all the levels of the technical architecture, sufficiently scalable to accommodate the resources limitations of the various targeted nodes (centralized management system, local controllers, sensors, actuators ...) and, of course, widely adopted and standardized.

2.2.5 Health

Wireless sensor networking opens the road to more health care at home (see 2.1.4). Issues are easy-to-use connectivity, privacy guarantees, long battery duration and minimal interference in the communication. As in hospitals, reliable wireless communication is a must. Automatic (re)configuration, remote diagnostics, and fault tolerance are important aspects which need research before wireless sensor networks can be deployed at a large scale. An example of a project that investigates this approach is the SAPHE project in the UK.

2.2.6 Connectivity at home

A plethora of standards has been proposed for the home. Examples of communication standards are: HomeRF, IEEE 1394, Lon networks, HomePNA, PowerLine IEEE 802.3, IEEE 802.11x (x = a,b, and g) and middleware standards like: Jini, VESA, HAVi UPnP, DLNA, Osgi, HGi, and many others. Many of the more popular middleware standards rely on the presence of the Internet Protocol (IP) stack. From a CE-vendors perspective, DLNA, which builds on UPnP is the more likely standard in the home. At the same time many vendors and groups try to sell their own solution and lock other suppliers out. From an Internet Protocol perspective, a single router or gateway in the home should be sufficient, but even vendors of wireless access points provide full-fledged routing capability and firewalls thus adding to the complexity of the installed base in the home. It is impossible to predict what kind of configurations will be present in the home 1-2 years from now. The architecture of to-day's mobile phones reflects this fact. We see many connections and connectivity standards added to the mobile terminal to provide connectivity with devices inside the home. It is clear that the home network needs to be managed. The interoperability between devices such that functions and devices are recognized, commands can be sent, results received and streams can be set up, is for a large part covered by DLNA and UPnP efforts. The management of the resources inside the home is still open. Thanks to WMM, priorities can be attributed to streams, but no bandwidth guarantees can be given to individual video and audio streams. Consequently, with two videos over a wireless link, one of the videos will stall and crash. This does not encourage users to use home networks for their entertainment. Also the service providers are reluctant because they fear that they will be overwhelmed by many complaints about quality of reception, which are outside their control. A large effort in research is needed to give the right instruments to the users to manage the streams in the home. These instruments must be standardised.

2.3 Nomadic

Nomadic systems refer to systems that enable computing and communication capabilities as well as required services and contexts to the nomad who moves from one location to another location. These capabilities should be provided

to the nomad in a seamless and transparent form. In other words, in a nomadic environment, the user's computing, communication, and service functionality are seamlessly adjusted.

The combination of computing with communications is changing the way we think about information processing. Nomadicity makes all the problems in computing and communications harder, but the potential payoffs for improved functionality are huge. Despite the technical progress of telecommunication technology, particularly, wireless communication, the dream of being able to talk to people and access information, services and contents anywhere at anytime has not yet come true. We are still seeking capabilities that must be put in place to support nomadicity. These capabilities must enable nomadic systems to be automatically and seamlessly adjustable to their place, movement, communication protocol, and communication bandwidth, and enable them easy access to information and services.

In the nomadic domain, wireless communications is a key technology enabler. Some of the key system parameters include capacity, bandwidth, latency, flexibility, reliability, error rate, delay, storage, processing power, interference, interoperability, and user interface. In addition, a desirable terminal for a nomad should be a low cost, light, and easy-to-use, with high-functionality.

Mapping the future visions for nomadic systems to four application contexts of ARTEMIS means enabling the seamless connectivity beyond the borders of the Nomadic applications context, extending it to other three application domains as depicted in Figure 13.

Some of the main system issues in developing the nomadic systems include security, wireless networking, power consumption, number of devices (capacity), and flexibility.

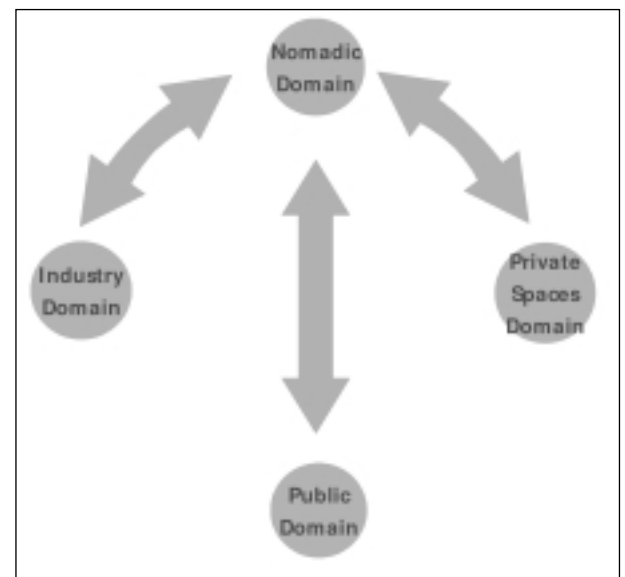


Figure 13 Nomadic future challenges for ARTEMIS to enable seamless connectivity for cross-application domains

2.4 Public

2.4.1 Energy - Power grid

The world's power grids have over the years grown from company owned electricity islands into some of the world's largest interconnected systems comparable only to the WWW.

For years mechanical relays protected equipment; huge generation facilities provided power; centralized network control centres operated the transmission grids; and through a fine grained distribution network local industries and households received energy. Low-bandwidth communication between control centre and network nodes is still rather a matter of fact than an exception. Few or no communication links exist on the level of the distribution grid. The IT systems are heterogeneous and rarely allow for seamless interoperability. Many legacy systems are still in place; a lot of proprietary protocols not meant for interoperability with the outside world are in use; applications and interfaces were designed as islands.

But changes are coming quickly: government bodies are opening electricity markets, splitting generation from transmission and distribution, and fostering energy trading across companies and state boundaries. Grids utilization is rising and the complexity of energy management in interconnected systems puts high stress on the operation of the grid. Distributed generation in smaller units, from photovoltaic equipment on customers' premises to larger wind parks in remote regions, are changing the structure of the network and flow of power. Smart sensors, intelligent electronic devices and the presence of digital equipment on all levels of the grid protect devices and provide data frequently and in vast amounts through Ethernet based communication. Internet technologies open doors between previously locked solutions and companies' and public IT systems are growing in size and complexity, while communication boundaries are dissolving under a growing concern for security and data privacy.

As a result there is an increasing need to exchange data between participants in the power grid: transmission system operators must ensure stability of whole utility networks under market conditions; single utilities want to optimize the interaction between network operations, planning, maintenance and customer services; network operations centres need real time supervision of the grid and appropriate control actions ensuring system security and optimizing the flow of power from a multitude of distributed and dispersed sources; and finally utilities and their customers need to interact on quality and price of supply.

The fast coalescence of communication networks and energy grids becomes a fact, motivated by the conditions to increase utilization of existing grids, to increase security of supply in existing grids and to stabilize the power system following collapses and cascading failures. The

characteristics of the distribution network will change dramatically from having millions of passive one-way nodes to a network of millions of homes and businesses that actively participate in the control of the network.

Hence, there are profound challenges driven by these factors. There is a need to develop methodologies and technologies related to the exchange and transformation of data, to the relationship and interaction between huge numbers of systems and how to embed and integrate new means of communication and advanced intelligent equipment. It must be ensured that individual sensing, smart electronic devices, communication equipment and the services offered from different vendors interoperate seamlessly. Therefore, a set of commonly adhered to standards and common methods for planning combined with a middleware infrastructure that supports linking these services is needed. It has to follow industry initiatives such as standardization efforts driven in IEC61850 or IEC WG57, but also pursue open middleware and communication standards. A set of new applications, such as wide area monitoring, control and protection of power system are requiring omnipresent infrastructure and a middleware that allows reliable delivery of services (communication, applications, network supervision) from different vendors.

The mutual dependency of energy distribution and communication on the same grid requires new means to effectively monitor and control both systems in parallel. The opposing goals of ubiquitous, seamless connectivity of the power and communication grid and of a power grid conducting mission critical operations for the whole society, emphasises the fact that IT security aspects in terms of both tamper-proof data and confidentiality of information need to be embedded deeply into resulting research activities.

And all this needs to be deployed into an existing grid structure with a primary equipment lifetime of decades.

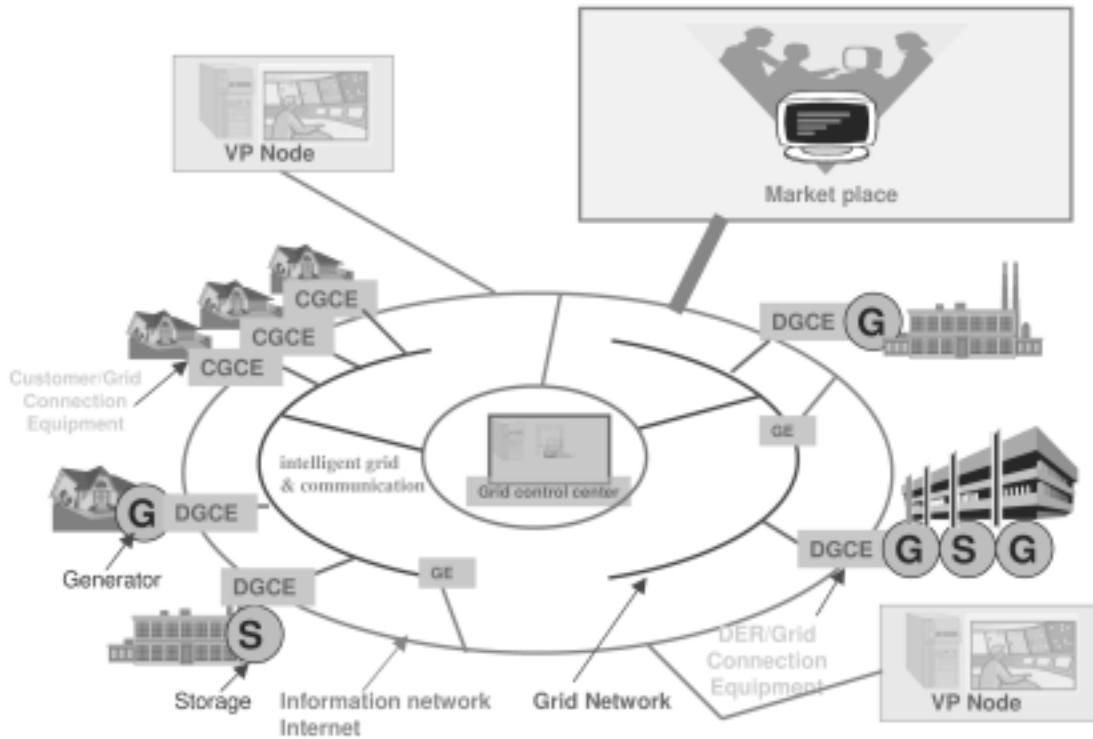


Figure 14 Future power grid structure

2.4.2 RFID & Sensor networks

Nowadays, Radio Frequency Identification Technology (RFID) is being considered for many commercial applications. It is considered to be one of today's most anticipated technologies for a broad range of applications. Basically RFID systems include (inexpensive) tags or "transponders" consisting of a very small silicon chip with an integrated antenna, readers or "interrogators", and system software (middleware). At present, depending on the type of tag, several kilobits of data can be stored on one tag. Data exchange between a tag and a reader, both reading and writing, takes place via radio waves. The tag generates electrical signals that are interpreted by the reader. The reader not only receives the signals from the tag, but may also activate the tag, in the case of using passive tags. It may also handle the communication with backend networks. The system software transforms the data into information describing goods that are carrying the tag and further handling for a specific application. It is possible to read many tags simultaneously, even if the tags are not within line of sight. Currently, RFID has been applied in many applications, for example, in asset tracking, in public transportation, in offices and in supply chain management.

RFID is one of the key enablers for the vision of ubiquitous computing. It creates an automatic way to identify and track an object or a product quickly, easily and without human error. In the near future, RFID will be increasingly used to enhance the quality of day-to-day living in different areas including games, sports, anti-counterfeiting applications, art, etc.

As already pointed out in this industrial landscape, Wireless Sensor Networks (WSN) are becoming more and more important. In the medical domain, they provide more flexibility in the hospital, but more importantly, it is expected that patient out-time will increase thanks to their deployment. In the agricultural area WSN can be deployed to sense crop and forest growth, and weather conditions. Adding a WSN to animals seems to be the answer to our increasing dependence on highly intensive animal husbandry, to monitor their health, their transportation and their meeting with other cattle herds. WSNs are assumed to be an integral part of our daily life, providing the infrastructure necessary for the realisation of ambient intelligence. Car manufacturers investigate how roadsides can be equipped by WSNs to assist drivers. Equipping the car itself with WSNs is envisaged to reduce cable costs. Reducing cable costs is also an important aspect for civil aircraft.

Two important aspects for WSNs are the communication range and the battery power. For agricultural purposes, the range for crop surveillance or climate conditions is relatively large, while the range for a body sensor network on a cow is quite short. In both cases it is necessary that the power can be maintained for periods over a year. Two techniques are applied: (1) a low duty cycle, and (2) energy scavenging. For very small sensors, the use of solar cells looks the more promising, although initiatives are starting to look at the use of gyroscopes.

A small node today typically may use IEEE 802.15.4, and consume about 1 mW of power. Its size is typically about

1 cm³. Efforts try to reduce the size to a half cm³ or less, and reduce the power consumption to 100 μ W are under way. Packaging the nodes also gets much attention. Several efforts in international consortia are looking into packaging techniques each starting from different geometrical form factors. Finally, inserting the building blocks like DSP, transceiver, and sensor into a system in a package is an important goal.

3. Multi-domain analysis domain clustering

3.1 The reason for clustering

The aim of multi-domain analysis is to exhibit meaningful *domain clusters*.

A cluster corresponds to some sort of middleware having basic well-defined characteristics, shared among application domains. It is proposed that ARTEMIS undertake their systematic development and impose them as standards.

Most of the clusters found do not correspond to application domains. Indeed, application domains are often split into different clusters; this means that systems will have to mix different kinds of middleware technology. Conversely, clusters often pick up parts of several application domains.

These clusters are formed according to two kinds of similarities between (parts of) application domains:

- Application domains can share *technology*. For example, this is the case of manufacturing, and part of the defence and security subdomains of the industrial domain. However, these clusters will not interact as markets in the foreseeable future. We call this a *weak cluster*.
- Application domains can share *infrastructure*. For example, this is the case of all actors in the home subdomain of the private domain. They will probably all share some form of gateway, and basic home connectivity services. We call this a *strong cluster*.

In the case of a weak cluster, there is no absolute need to share technology. Indeed, today, technology is not shared, or is shared without intention to do so or in a partial way, because some common standard was available and each actor built similar solutions on this. However, intentionally sharing of technology could bring many kinds of benefits: optimizing R&D investments; helping the creation of a primary industry that will support and improve the technology; imposing the technology worldwide. Last but not least, this makes it possible to share the technology

among competitors within an application domain, if not in a multi-domain way.

In the case of a strong cluster, the issue is more on *market opening* than on *technology costs*. If no action is taken, it could turn out that different infrastructures are developed by actors from different application domains. This is not very efficient, and at some point technology interoperability or merge will become an issue. But the most important drawback of this scenario is that this limits new markets to *innovative applications that can pay for their infrastructure*. There might exist many potential applications that would be perfectly viable if the infrastructure existed. Think of what the Internet made possible once it came into existence.

In both cases, ARTEMIS can probably play a key role in pushing towards the best solution for market opening and value creation. It is clear that there is no easy path to this, and that industrial actors involved must exchange deeply on strategic as well as technological issues.

This SRA, together with the results of the analysis that was conducted, opens the door towards decisions in this matter.

3.2 How multi-domain analysis was conducted

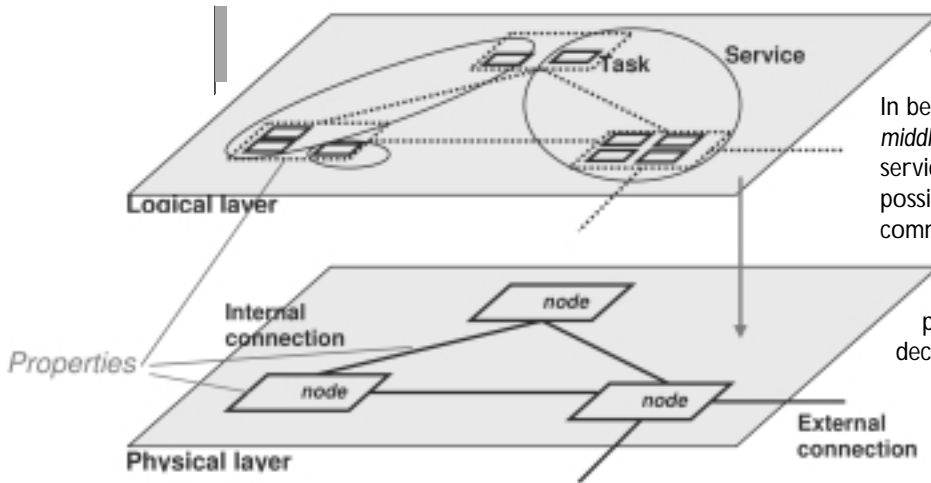
A cluster corresponds to some middleware having basic well-defined characteristics, shared among application domains, that ARTEMIS should undertake to systematically develop and impose as standard.

Indeed, the situation is not simple in embedded systems because of the huge variability in system size and requirements between the ARTEMIS application domains. What is common between a smart phone and a car?

To discover pertinent clusters, experts conducted an analysis of each domain, trying to describe their main *future* systems characteristics. This analysis is summarized in Section 2. Though very difficult, it is extremely important to consider future systems, and not to try to imagine how current systems could be better designed or decomposed.

To help express this analysis within a common framework, we proposed as a guideline a simple, qualitative *model* of a system. We called it a *multi-domain* abstraction. Experts could use this abstraction freely. Its purpose is *not* technical: it aims instead at facilitating the strategic analysis leading to domain clustering.

Essentially, a system is described in terms of a physical layer, a logical layer, and behaviour of this decomposition, in particular in terms of lifetime. It is eventually characterized by desirable non-functional properties.



embedding such computing capabilities. Nodes can also have sensing or actuating capabilities.

In between the physical and the logical layer, lies the *middleware*. The middleware consists of all the generic services common to many applications that make it possible to deploy and run services decomposed as communicating tasks.

The system is characterized by some qualitative properties of the physical and logical layers, and system decomposition characteristics. If the model is accurate for that purpose, systems having common characteristics should use similar middleware. Indeed, system characteristics should make it possible to assess required and desirable middleware features. This gives rise to domain clusters.

Figure 15 Simple system abstraction

The physical layer consists of nodes connected by communication media (networks); the logical layer, which expresses the functional view of the system, is made up of services, which decompose into tasks running on nodes and communicating through network connections. Nodes in the physical layer can be computing and communicating nodes (some chip or board element having computing and communication capability), or more complex subsystems

3.3 Main characteristics of a system: quantitative and qualitative dimensions

First of all, all elements of the system are characterized by simple, often qualitative attributes. This provides an instantaneous view of the system.

Nodes	Network connections	Tasks	Task communication	System
<i>Resources available: computing power, memory available, communication capability</i> <i>OS, MW, IW available</i> <i>Cost</i> <i>Sensing capability</i> <i>Energy available and energy consumption model</i> <i>Reliability</i> <i>Size, weight</i>	<i>Bandwidth, latency...</i> <i>Adaptation to specific flows, protocols and services</i> <i>QoS capability</i> <i>Cost</i> <i>Reliability</i>	<i>Resources needed</i> <i>Cost</i> <i>QoS, quality</i> <i>on Application level</i>	<i>Resources needed</i> <i>Cost</i> <i>QoS, quality</i> <i>on Application level</i> <i>Semantics of communication: Ontology</i>	<i>System size: number of nodes, of Connections, of tasks...</i> <i>System topology</i> <i>System organization (hierarchy)</i>

Then, we look at the dynamics of system decomposition.

Timescale	Physical level	Logical level	System level
Design time	<i>Are COTS heavily used?</i> <i>Are standards common or specific to application?</i>	<i>Same question.</i> <i>Same question.</i>	<i>Is system design based on decomposition by a single design team?</i> <i>Or is it field-composed?</i>
Offline lifecycle	<i>Are nodes / networked connections redesigned?</i>	<i>Are new services added, upgraded?</i> <i>Compared lifecycle with physical layer.</i>	<i>Is change incremental or does it involve overall redesign?</i>
System operation	<i>Do nodes / network connections appear dynamically?</i>	<i>Same question</i> <i>Do tasks migrate to different nodes?</i>	<i>Change lifecycle</i> <i>System lifetime</i>

Finally, systems can be characterized by QoS or non-functional requirements, such as its level of dependability.

3.4 Cluster description

This analysis eventually gives rise to 6 clusters, that we called *Critical, Device & Plant, Private, Nomadic, Ad Hoc Connectivity, Systems of Systems*.

We describe each cluster, below, in terms of their main characteristics with respect to the multi-domain abstraction, and in terms of how they intervene in application domains.

3.4.1 Critical cluster

General comment

The *Critical* cluster corresponds to systems where safety is by far the most important issue. It imposes new, radical solutions to very fundamental problems arising from mainstream computer science, notably being able to assert the predictable, deterministic, real-time behaviour of the system. For this reason, new solutions have been developed that share little or nothing with mainstream standards and technology.

However, some fruitful exchange could be undertaken between actors from the concerned domains. The main difficulty will probably be with component and design costs, which are very different between sub domains, and with certification procedures and practice.

Application domains concerned

This cluster concerns avionics, automotive, some manufacturing (e.g. nuclear plants), and some medical devices, at least for the parts of these objects that are concerned by the *critical* issue.

Abstract description

Nodes	Network connections	Tasks	Task communication	System
Critical technology impacts e.g. redundancy, or even chip architecture	According to critical technology; very different from mainstream	Deterministic, predictable, hard real time, composable	Id. Ontology should be developed for improving design, however the strongly integrative nature of the system does not push to embed it	Complex integrative technology already developed, or remains to be developed. Standards to facilitate component exchange between actors

Timescale	Physical level	Logical level	System level
Design time		Components should be exchangeable in a safe and liable way. Standards still specific to application domains. Certification issues are difficult.	System design is based on decomposition by a single design team.
Offline lifecycle	Redesign must be limited.	Redesign limited. Upgrade limited to fault correction.	More important change would involve deep redesign.
System operation	No node nor connection changed during operation.	No task nor service changed during operation. No task migration.	Critical system lifetime is whole system lifetime, or complete redesign is necessary.

Non functional requirements

System is *critical*.

3.4.2 Device & Plant cluster

General comment

The Device & Plant cluster corresponds to real-time systems where composition from reusable, interchangeable components and COTS is an important issue. System quality is also an important issue, obviously, but not an absolute goal. This makes it possible to rely upon standards stemming from mainstream Information Technology, albeit with the necessary changes and development. Also, these systems often have an architecture where communications with the outside are concentrated at some point, and where the systems can be remotely queried, operated, diagnosed.

It turns out that this cluster can reasonably encompass large plants in the manufacturing sector as well as “devices” for example in the defence and security domain. The reason is that it is much more efficient, in particular in terms of time to market, to design systems “in box” by composing reusable components and “gluing” them together, using generic middleware technology.

The fact that some communication point concentrates communication with the outside at all semantic levels does not presume any kind of internal architecture, in particular not a pure client-server architecture, though this is also obviously possible.

Application domains concerned

This cluster concerns avionics, automotive, some manufacturing (non critical), defence & security, some public sub domain such as the power grid. Indeed, it applies whenever there is a technical system that can be locally or remotely monitored, operated, maintained, diagnosed, etc. This is the case for example in the automotive sector for maintenance operation in station, or at home or in office buildings for energy or automation system remote operation.

Abstract description

Nodes	Network connections	Tasks	Task communication	System
Wide heterogeneity of nodes, in particular in terms of cost and resources available. E.g. FPGAs should be managed in devices, sensor technology is obviously important. Energy consumption may be an issue.	Can be industry-specific, however convergent.	Must be real time	Id. Ontology essential for multi-vendor communication.	Integrative, components should be easily reusable and exchangeable. COTS heavily used.

Timescale	Physical level	Logical level	System level
Design time		Good component model required. Should be strongly standard based.	System designed by a single design team, however from COTS or reusable components. Model-based recommended.
Offline lifecycle	Change in physical platform made easier from components and model-based approach.	Redesign made easier from components and model-based approach.	Change and evolution is natural for that kind of system. Obsolescence is an issue, for the physical as well as the logical layer.
System operation	No node nor connection change during operation.	Normally, service to task distribution is fixed. However, new services can be downloaded, in particular for remote operation.	Long lifetime.

Non functional requirements

System must be real-time.

3.4.3 Private cluster

General comment

The *Private* cluster corresponds to systems at home, in transportation, or at work, that make available to users a series of comfort, entertainment and information services, and interaction devices.

This cluster relies upon an infrastructure, with networking capabilities of various kinds, notably wireless, and probably some gateway capability. Computing power, storage resources, and interaction devices could also be part of the common infrastructure, provided they are seamlessly accessible. Anyway, there is some means to access from this private space, in a seamless and secure way, services available from the outside. So, one of the most important characteristics of this cluster is that it is a strong cluster, as it is advisable that a common infrastructure be shared between actors from today's different market spheres.

Today the private space is made up of a series of islands, and there will be a strong push towards seamlessly connecting these islands. Many standards are in competition in this sector, however DLNA seems to be in advance.

If European industry can develop and adopt common middleware technology, it opens up the door to a whole ecosystem of innovative services and actors.

Application domains concerned

This cluster potentially concerns many sub domains, including health, energy, building automation, white goods, comfort and well-being, and of course multimedia and games. Private spaces are present at home, but also in cars, planes, and other transportation modes, in offices, and even in more "public" spaces such as hospitals, theatres, etc.

Abstract description

Nodes	Network connections	Tasks	Task communication	System
Nodes are highly heterogeneous. Energy consumption is a strong concern. Cost must be low, this is a mass market. Infrastructure nodes are a market concern, in particular gateways.	IP is the rule, however legacy networking might survive for a while. Wireless is of prime importance, however other networking technology, such as powerline, is in competition. Heavy multimedia streams are expected.	Must rely upon a single component model, at least for administrative issues (decomposition and composition, deployment, migration). Service Oriented Architecture is a must. They are soft real-time, though in most cases not critical.	Semantic level, and ontology is a most important issue. Ontology should be created and shared in each sub domain, and as multi-domain. One could think of developing commonsense ontology, in cooperation with the nomadic domain.	System is composed in the field. No single designer can have a global system view, indeed it is (partly) composed and configured by the user itself.

Timescale	Physical level	Logical level	System level
Design time	This is an area where convergence between information processing, multimedia and communication will take its full meaning.	Services must obey some standardized and agreed upon Service Oriented Architecture. Also, some generic services (monitoring, logging, diagnosis, maintenance, self-healing) must be made available.	Only sub systems are designed by professional designers, they must be thought from the beginning as open systems. However, system architecture (including gateway and other generic services) play a key role.
Offline lifecycle	It is in the interest of industrial actors, and of users, to have an "IT" product lifecycle similar, i.e. faster than the environment of use (e.g. home, car).	New services must be freely installable. Running on older physical infrastructure is an issue.	System is ever changing.
System operation	Nodes (devices) can come in and out the system at fast pace, and connectivity can be fixed or ad hoc. The system is supposed to self-connect and self-compose according to criteria including semantics and personal private profiles.	Service discovery and service dynamic deployment is a must. Ontology should be embedded and exchangeable in a dynamic way.	System will become as natural and eternal as universal access to electricity today.

Non functional requirements

Monitoring, diagnosis, and maintenance will become a very important technological and market issue.

3.4.4 Nomadic cluster

General comment

The *Nomadic* cluster will, in the future, be at the centre of many connectivity issues.

It will bring together several basic capabilities:

- personal and wearable computing, storage, and local area up to wide area networking capability
- universal interface to ambient agents
- it will make connectivity among various domains and clusters possible, at the level of component and middleware standards as well as ontological and semantic levels
- it can act on behalf of other domains and clusters as an infrastructure element, in particular as some kind of gateway
- it will provide very basic and important services, such as
 - location and context information
 - personal privacy domain, link to communities
 - commonsense ontology for minimal universal seamless connectivity
- it will be the gateway for Body Area and Personal Area networks (BAN and PAN).

Application domains concerned

The Nomadic cluster is an application domain per se. However some of its constituent components could be embedded components in other domains, as they provide required universal functions. Indeed, almost all domains are concerned.

Abstract description

Nodes	Network connections	Tasks	Task communication	System
<p><i>Nomadic</i> cluster relies upon a complex computing platform, that implements convergence between information technology, communication and multimedia.</p> <p>It normally consists of a single device, though it might use resources available in the environment for computing power, storage, sensing or more sophisticated services. Bears various sensing capabilities, or connects to them.</p>	<p>Speaks many networking idioms. IP and wireless is the rule.</p> <p>There are some aspects specific to multimedia. Networking can be ad hoc as well as global on xG networks.</p>		<p>Interacts with information services available locally as well as globally.</p> <p>Has the ability to dynamically assign tasks to ambient resources, or to use services available locally.</p> <p>It potentially speaks many service idioms and ontologies.</p> <p>It uses generic commonsense ontology.</p>	<p>The device might itself be internally designed using the <i>Plant & Design</i> cluster technology.</p>

Timescale	Physical level	Logical level	System level
Design time	The core device.	Generic services are supported, plus access to other domains and clusters.	The core device is designed in an integrative way. However, its role of universal connector, and the fact that some services cannot be run if access to ambient resources cannot be found, makes it a widely open system.
Offline lifecycle	The core device follows an IT-like lifecycle. However, it must be able to communicate with older technology.	New services must be freely installable.	IT-like lifecycle.
System operation	Can dynamically use ambient resources.	Service discovery and service dynamic deployment. Alien ontology should be downloadable in a dynamic way.	The device can connect to ambient devices and agents in an ad hoc and seamless way, and so form short life systems (see 3.4.5).

Non functional requirements

Security, reliability and trust are very important technological and market issues.

3.4.5 Ad Hoc Connectivity cluster

General comment

The Ad Hoc Connectivity cluster corresponds to ensembles of a limited number of systems that are composed for a limited time, in an ad hoc way, and for achieving a specific goal. It is a kind of short-term, peer-to-peer like, specific purpose System of System (see 3.4.6).

This is a weak cluster. Indeed, it is an issue to know whether it should be considered as a cluster at all. However we can see various initiatives from various domains coming down to this definition.

Application domains concerned

The Ad Hoc Connectivity cluster covers (part of) the automotive, avionics, for local traffic management and sensing and information processing, e.g. between cars or between cars and digital signs. It can be used in the defence and security domain, in public infrastructures such as the power grid, as a technology enabling local, distributed behaviour, so avoiding heavier loading and the increased risk of failure of a centralized infrastructure.

It is also extremely important for the nomadic domain, where the user can quickly and locally connect to ambient information spots, for example.

Abstract description

Nodes	Network connections	Tasks	Task communication	System
AHC nodes are systems, in a limited number, some of which can have limited capacity. Sensors play an important role in AHC.	Wireless is the rule. Connection is ad hoc. Multi-hop networking is among the possibilities. QoS is probably limited. Connection is probably limited in time.	Due to limited operation time, only tasks corresponding to services known from all the systems involved can play a role in AHC.	Ontology must be shared among the systems involved. Ontology is domain and even application specific.	

Timescale	Physical level	Logical level	System level
Design time	Systems must be prepared for the fast connection.	Tasks and roles are assigned to types of nodes at design time.	Though system is composed in a dynamic way, services are designed through a conventional divide-and-conquer strategy.
Offline lifecycle		New services must be downloadable and deployable.	IT like lifecycle and obsolescence cycle applies.
System operation	Highly dynamic, ensembles are short term.	Service discovery is simplified. Service static deployment is the rule. Systems should be able to connect through shared ontology without negotiation.	System is composed on the field. Interoperability between similar services from different origins or sellers will turn out to be an issue.

Non functional requirements

Probably the most important requirement is that AHC system will have to remain under the user's control. Security and trust is also an essential issue.

If information exchanged is used for critical operation, then different technology might be used, which has the same functionality as AHC, though being based on radically different technology used in critical systems.

3.4.6 Systems of Systems cluster

General comment

The Systems of Systems cluster corresponds to very large, highly dynamic, highly complex systems. They are composed of heterogeneous systems, themselves networked in various ways. Systems of systems represent a full area of research, which is still in its infancy. Here, we consider only the connectivity and middleware issues of SoSes.

There is no typical architecture for Systems of Systems. All currently imagined architectures can probably contribute to a System of Systems, either hierarchical or peer-to-peer like, for example.

Nevertheless, Systems of Systems information infrastructure will be built incrementally, and evolution of the underlying “physical” system will push the research issues in this problem. Consider as an example the problems posed both by deregulation and the advent of distributed generation for the power grid management.

Application domains concerned

The Systems of Systems cluster covers (part of) the defence and security domain in a dual way, public infrastructures such as the power grid, and in the future large scale health systems and very large traffic management systems.

Abstract description

Nodes	Network connections	Tasks	Task communication	System
System of Systems nodes are themselves systems, which expose some identity and some capability (computing, networking, content and knowledge, sensing, actuating). Some systems can have generic roles w.r.t. SoS management and behaviour, such as local command and control, monitoring, action planning, diagnosis, failure repair, etc.	Many heterogeneous networking capabilities are involved. They should be able to expose their capability.	SoS does not manage low level tasks, this is devoted to systems. SoS works at the level of service composition and orchestration.	Ontologies play a key role in SoSes at all levels of information processing, from resource management up to high level services. They are the condition for establishing and performing meaningful connectivity.	SoS has a more or less reflexive architecture.

Timescale	Physical level	Logical level	System level
Design time	SoS is composed of few to many complex systems. Some systems can implement generic organizational functions, which can bring complexity back from many to few.	Operational services, plus generic services.	System can be partially designed through a divide-and-conquer analysis, at least some scenarios can be studied. SoS modelling and simulation are essential, for design, reasoning, simulation, teaching, etc.
Offline lifecycle			Systems of systems are often eternal systems, so change and maintenance operations can only be done while operating. Otherwise, all system issues are present.
System operation	Systems can come in and out an SoS at fast pace, or have their capability dynamically changing from full capacity to none, via degraded behaviour. Network connections also dynamically change, together with the capability they expose.	Service discovery and service dynamic deployment is the rule. Systems should be able to connect through ontology negotiation.	System is composed on the field. The reflexive capabilities of SoSes, including complex models of its components and of itself, can be used dynamically.

Non functional requirements

Non-functional requirements are complex, and depend on the application.

4. Research priorities

Middleware technologies, such as CORBA and .NET have been widely used in many application areas to mask out problems of system and network heterogeneity, and alleviate the inherent complexity of such distributed systems. However, applications in embedded systems, real-time systems, and multimedia, impose new challenges which most existing middleware platforms are unable to tackle. These new application areas impose more demands in terms of resource sharing, dynamicity, security and timeliness. Therefore, these areas also require additional properties from the underlying middleware.

From the analysis conducted in *Sections Industrial landscape and Multi-domain analysis - Domain clustering*, this section presents the main research priorities that are necessary to implement middleware for the various clusters. These research priorities are grouped in 8 main topics:

1. Programming model
2. Organization and Deployment
3. Resource Management
4. Data distribution
5. Robustness and Diagnosis
6. Global connectivity
7. Provable correct systems
8. Security

A matrix at the end of the section shows the importance of each of these topics in relation to the six market clusters from Section Multi-domain analysis - Domain clustering.

4.1 The Middleware Programming Model

Product vendors and service providers rely on the middleware's programming interface and deployment mechanisms for developing applications and making them available to users. As such, the expressiveness, overall usability, and standardization of the programming model exposed by the middleware to application developers constitute a critical issue that has deep consequences on the success of the approach that ARTEMIS promotes.

Middleware as Key Enabler for Declarative Paradigm for Programming

The first key requirement for middleware is to provide application developers with a modular programming model that makes explicit the inter-module dependencies, or in other words that formalises the "integration contract" of application software components. This enables development teams to work independently from each other while keeping the ability to seamlessly integrate the application,

based on well-defined inter-module interactions. Modules are hereafter referred to as components.

The other key requirement is to enable application developers to focus on the functional, business behaviour of the application, while leaving implementation of non-functional/technical aspects of the application to the middleware. In other words, the middleware shall enforce a strong separation of concerns. Indeed, targeted applications have a wide variety of non-functional requirements. For instance, some applications may require the underlying middleware to support fault-tolerance and strongly encrypted communication (e.g. a card payment terminal), while others would require no security (e.g. an audio device commenting art pieces in a museum).

Due to the variety of underlying hardware and communication mechanisms, it is not possible for an application developer to foresee every single potential physical configuration the software may be deployed on. Middleware will have to offer solutions that make the fewest number of assumptions on the underlying hardware. The proper compromise between static middleware configuration and dynamic middleware adaptation undoubtedly depends on the application and the specific domain. As a consequence, middleware might be able to expose different domain-specific flavours. This is a necessary condition for making the middleware applicable to the different application domains and clusters targeted by ARTEMIS.

Additionally to these high-level requirements, one can identify several other important features the middleware might support. First, the inherent complexity of targeted applications suggests a hierarchical component composition mechanism that would enable applications, services, subsystems and systems to be composed in an abstract manner, regardless of the technicalities involved in managing the underlying platforms. Ability to consider several components as a single, coarser-grained component is essential to reasoning about assemblies and applying properties to those assemblies. While several middleware technologies already support such a hierarchical composition mechanism, doing this without sacrificing memory footprint and real-time performance is still an open problem.

Moreover, the programming model of the middleware might support a wide variety of architectural styles, such as workflows, dataflows, semi-structured event-based interactions, interactive applications, stream-based multimedia applications, cooperative multi-user applications, and so on. Today's middleware usually focuses on a small set of architectural styles, and this limits the integration capabilities.

To sum up, the general philosophy of research orientations presented above is to make middleware become the cornerstone of a declarative application development paradigm: the application components' internal and

external structure, the non-functional features they require, as well as the middleware's own customisation and configuration, shall be stated declaratively rather than programmatically. In conjunction with code generation techniques, and thanks to future enrichment of catalogues of available middleware services, the declarative approach to application development may dramatically contribute to increase productivity of software product vendors and software-based service providers.

4.2 System organization and deployment

Pervasive computing infrastructures are by definition highly distributed and dynamic. In order to successfully realize applications for such a fluid environment, application developers need software technologies that are able to manage the adaptation, computation and communication requirements in an efficient and transparent manner. The requirements for such pervasive middleware technologies vary significantly across the different application domains and clusters.

In recent years, a substantial number of generic agent communication languages as well as different agent platforms that support interoperability of heterogeneous networked devices and applications have been proposed. On the other hand, specialized middleware concepts for the management of pervasive computing issues are being developed by a number of research groups. As the number of pervasive middleware concepts is constantly growing, it becomes increasingly important to develop a common understanding of the mandatory feature set and to identify suitable solution concepts.

4.2.1 Dynamic reconfiguration capabilities

Seamless connectivity for intelligent embedded systems is not only constrained by communication and computation resources but also by the environment with dynamic characteristics such as wireless communication environments.

In case of dynamic scenarios, where people come together in an ad-hoc manner, where each of the users brings his/her own personal devices, or where each may be moving, or where people are buying new sensors or devices for extending their existing device ensembles, dynamic reconfiguration is indispensable. This spontaneous ensemble requires a kind of assistance different from a fixed scenario.

This kind of Intelligence requires more than setting up a central control application in advance or providing a sophisticated off-line configuration tool. It requires the ability of the devices to autonomously configure themselves into a coherently acting ensemble that is fully distributed.

This approach means dismantling the hand-crafted designs and implementations of component and device ensembles

as it is done today and replacing them with new methods of self-organized ad-hoc device cooperation.

4.2.2 Efficient user interaction

With increasingly complex technology, the user is in danger of losing sight of his goals or of changing his goals, because he is not able to find the appropriate combination of strategies for device functions. The user is forced to pay more attention to complete lists of functions than to her actual goal. Instruction manuals for today's devices are a good example of the *metaphor change* that has taken place. Furthermore, which goals could be achieved, if devices were to be interconnected, is generally not mentioned at all.

Considering that the vision of Ambient Intelligence is to realize the creation of reactive and reasonable intelligent environments for the user's needs and well-being, the basic requirements are the following:

- The environment and its devices must be aware of the user's current situation, his interaction within his environment and its own current state and possible changes in this state.
- In addition, the environment must be able to interpret those occurrences into user goals and, accordingly, into possible reactions that enable a cooperative, proactive support for the user.
- In a final step, the environment must be able to translate the interpreted goals into strategies that can be fulfilled by the environment's devices and functionalities in order to adapt itself to the user's needs.

4.2.3 Device and service discovery

For dynamic systems it is very important to synthesize and describe to users new functionalities that are provided by a device entering the environment. Thus operational integration needs a form of service discovery framework common to all devices in related application domains. The environment should be able to dynamically discover devices and services as they become available.

This can be realized with explicit modelling of the semantics of device operations with precondition / effect rules that have to be defined over a suitable environment ontology. Furthermore, the environment and the client should be able to automatically start interacting with the newly discovered devices / services without any programming.

The aim for environments in which devices can join the network and dynamically discover the services that they need has already motivated the development of a number of service discovery and interaction protocols, including Jini, HAVi, SLP, UPnP and others. None of these protocols are really suited for a mobile ad-hoc networking environment. They are either very limited in scope or based on aggregating information into central directories which would introduce single points of failure.

Thus a mechanism is needed where the aggregated

information and the structure of the network nodes are also distributed.

4.2.4 Ontologies³

Semantic-based technologies and ontologies can play an important role in seamless connectivity, particularly from the cross-domain point of view. One of the main goals of ontologies is to provide “a shared and common understanding of a domain that can be communicated between people and application systems”. Lack of a shared understanding leads to poor communication not only within and between people but also within and between embedded systems.

Today, there is much activity related to the Semantic Web. The main purpose of this is to make the content of the Web more understandable to machines by introducing semantic markup. Semantic integration requires that contents from different application domains become understandable between them.

Research priorities on ontology for seamless connectivity relate to the question of building shared knowledge for domain and cross-domain seamless connectivity. This includes building ontologies for specific domains, including support for legacy systems, but also interoperability through cross-domain ontology mappings.

One can eventually think of a generic ontology that could apply to very general situations, and be shared among domains and clusters, that would make it possible to exchange *common sense* information and knowledge.

4.2.5 Conflict resolution

In device ensembles, handling of conflict resolution strategies are needed that guarantee data-flow even if there are competing network components, devices or sensors. If all components are able to “see” the messages of the other components, some conflicts will arise as to which component(s) are allowed to process the message. Those conflicts of competing components have to be solved by conflict resolution strategies, which are part of the channel message handling capabilities.

These conflict resolution strategies on each channel decide which component gets a message (or rather to decompose a message into multiple messages that can be handled better by the subscribing components). The conflict resolution strategy is eventually based on the communication channel's ontology and on the semantics of the messages that are communicated across the channel.

Since the number of components is permanently increasing, methods of parallel computing become more

³ The word “ontology” has been used in Knowledge Engineering Community for more than a decade. Many definitions have been given about what an ontology is but the most commonly quoted definition is “an explicit specification of a conceptualization”

and more important. Mechanisms have to be found to calculate channel strategies in a distributed fashion.

4.3 Resource management

4.3.1 Adaptive resource management

As embedded systems become more seamlessly connected to each other, they are expected to be more and more subject to changes in their physical and logical environment. They are expected to dynamically adapt to such changes. Adapting their execution to the changing environment will be more efficient than applying too pessimistic hard real-time dimensioning techniques, but to do this dynamically is a big challenge for real-time embedded systems.

Resource management is needed for ensuring that the resource reserves or budgets are guaranteed. It will allow high utilization of the system resources such as CPU, memory, network, and energy, in order to enhance the overall system performance. Also, it will distribute and allocate system resources according to the application requirements. For this purpose, resource usage accounting, budget enforcement, and monitoring are essential mechanisms to be provided by the real-time kernel.

4.3.2 Application end-to-end QoS

The capacity to specify end-to-end QoS requirements implies integration of system-wide policies to parameterize adaptive resource management algorithms. There are some proposals for describing QoS characteristics, contracts, quality levels, etc, although none of these can be considered as really satisfactory. One of the reasons is probably that these concepts are not completely understood and it is not clear how to handle them, to achieve the desired goals.

End-to-end QoS has to be provided by the connected devices in the system. These are likely to have different ways to express and provide QoS individually. Thus, end-to-end QoS decomposition is a key challenge, providing for interoperability of devices with minimum overhead.

4.4 Data distribution

Several application domains and clusters express a requirement for the support of flexible distributed data management. As they will become seamlessly connected to each other and to enterprise or global information systems, embedded systems will need to move large quantities of data over a variety of heterogeneous and often non-deterministic networks. It is however critical that bridging information between global, enterprise and embedded systems does not compromise the performance and real-time behaviour of those systems.

Leveraging existing standards for database and distributed data management is of course essential in the deployment

of such solutions. The OMG's DDS standard, in particular, is quickly gaining importance in the information systems area through its publish/subscribe data communication paradigm.

A research priority for ARTEMIS is thus to adapt and extend those approaches and standards to deal with the variety of constraints of the embedded systems and the heterogeneity and variability of transport networks. An example could be to propose an extension of the OMG-DDS standard to include QOS support for WAN or disconnected networks, and also information model mappings to optimize data transfers between sub-systems with heterogeneous computing capabilities.

4.5 Robustness and support for diagnosis

Robustness is the capability of a system to deliver an acceptable level of service despite the occurrence of transient or permanent hardware faults, design faults, imprecise specifications, and accidental operational faults. The research challenge is to devise middleware services that improve the robustness of the infrastructure services and support developers in ensuring robustness of application services.

In conjunction with architecture-level mechanisms (e.g., structuring of the overall system into fault containment regions that fail independently), middleware can contribute to achieving the goal of developing a robust system.

4.5.1 Error containment in a distributed and dynamic system

An error containment region is a well-defined subsystem where the consequences of an error that occurs within the subsystem will not propagate outside the subsystem without being detected. Error containment is a prerequisite for building fault-tolerant systems since without error containment a single fault has the ability to corrupt the whole system. Error containment also facilitates system integration and liability issues in systems composed of components from different vendors.

In order to achieve error containment, an error has to be detected or masked within the error containment region. Research challenges comprise the analysis of how the different types of faults in the fault hypothesis (e.g., hardware faults, software faults, interaction faults) manifest themselves on the distributed state, as well as the development of suitable error detection and error masking mechanisms.

Incomplete knowledge about computational latency and input load also leads to imprecise temporal specifications (platform non-determinism).

Since error propagation occurs via shared resources, error containment mechanisms for different types of resources

(e.g., communication resources, computational resources) need to be devised.

The research challenge is to support error containment with such imprecise temporal specifications.

Global diagnosis architecture and detection of correlated errors

Diagnostic systems operating on only the local state of a component preclude the possibility to detect and analyze correlated failures or system anomalies. Therefore, a research challenge for future diagnostic systems is to provide the means to establish a holistic view on the system by operating on the distributed state.

This includes also context information such as environmental parameters (e.g., temperature) that can give important hints in the identification of spurious anomalies. A pivotal property of any diagnosis system must be the recording of the state of the system at the time of occurrence of a behavior deviating from the expected service. Such an on-the-fly analysis is vital, since many faults are not active at the service station, which renders understanding through the service technician impossible.

4.5.2 Global connectivity

For the vision of seamless connectivity, we also need to identify, compose, configure, and maintain a multitude of interconnected embedded systems, each with different capabilities. These systems will have to locate and recognize objects and people and to analyze the context, adapt, and learn from the users around them. Today, most embedded systems and devices are not aware of their environment and therefore cannot make timely, context-aware decisions. This is an architectural shortcoming of today's embedded systems. Intelligent environments are also prerequisites to meet the challenge of seamless connectivity.

4.5.3 Interoperability and connectivity in heterogeneous environments

Connectivity has to be enabled across borders between embedded systems & subsystems, networks, services and environments with seamless handover between heterogeneous access schemes and sessions. These technologies may include the current wired and wireless technologies by adding some extra functionality on higher layers of software and hardware implementations.

There is no single wired or wireless access or radio technology to provide system connectivity in all scenarios and in all application domains. In the future, we will face heterogeneous networks that include some of the current wireless access schemes in addition to some advanced complementary technologies even for niche scenarios and application cases.

Developing efficient and robust⁴ communication protocols to enable seamless communications between systems as well as inside systems in heterogeneous environments is thus a research priority.

Ontologies (see Section 4.2.4) are also required to ensure seamless connectivity and interoperability in heterogeneous environments, notably through cross-domain ontology mapping.

4.5.4 Connectivity in constrained environments

On the other hand, there is a need to make devices smaller, lighter with much longer battery life. Mobile devices are powered by battery, but the rate at which battery performance improves is fairly slow. It is doubtful that significant improvement in battery performance can be expected in the foreseeable future. While trying to improve the battery performance, we should consider carefully the design of communication protocols so that mobile devices can perform more efficiently.

Driving down the power consumption and optimizing the communication protocols specifically for power efficiency is thus also an important priority.

4.5.5 Provably correct systems

Building distribution platform for seamlessly connected systems is a complex task. One has to cope with the restrictions enforced to achieve (real-time) embedded systems, or to meet stringent requirements. Thus, one has to be able to assert middleware properties, e.g. functional behavioural properties such as absence of deadlocks, request fairness, or correct resource dimensioning, but also temporal properties, to validate real-time properties.

4.5.6 Operating Systems & middleware for safety critical systems

The formal-based verification of distributed application behavioural properties is usually the domain of verification-domain experts, using specific verification techniques, e.g. calculi, formal methods. However, such a verification process is usually used only to verify the semantics of the application (e.g. set of correct message sequences). This provides no information on the underlying distribution framework or middleware integrated to the system; and thus reduces the scope of the properties proved for the application under study.

So, the verification process of a distributed application should also focus on the middleware as a building block, and thus middleware architecture should be made verification-ready so as to ease this process. However, there is a double combinatorial explosion when considering middleware as a whole: the number of possible

⁴ Both physical & semantic robustness are required here

execution scenarios for one middleware configuration increases with the interleaving of threads and requests; the number of possible configurations increases with middleware adaptability and versatility. Finally, the behaviour of a middleware highly depends on the configuration parameters selected by the user.

Some contributions following these middleware requirements have already been made in the domain of hard real-time systems, but long-standing issues of adequate reliability and trust using off-the-shelf and low-cost components have not reached full maturity and common usage, and must therefore continue to be improved. This is clearly true in the Automotive domain where we face strong requirements for small memory footprint, low-cost embedded processors, and general multi-tasking operation.

Moreover, these first proposals do not address the very challenging domain of seamlessly and dynamically connected systems, which raises a much wider set of requirements.

4.5.7 Designing and integrating provably correct systems

Modelling for verification purposes (behavioural aspects), and middleware engineering (architectural and functional aspects) are usually considered as two different expert domains. They are usually considered either by separate teams of the project or at separate steps in the design process. There is a strong need to reconcile system modelling and middleware engineering.

Some middleware requirements can already be listed to achieve this goal and in particular a clear separation of concerns to provide a separate analysis of middleware components using formal techniques.

4.6 Security

Much effort has been devoted, in various information systems domains, to technology for enforcing security on distributed computing architectures. As long as they are not widely connected to each other or to the enterprise information systems, embedded systems computing platforms are not as strongly impacted by those security issues. Due to the interconnection of systems and the needs for interoperability, however, the scope of security has drastically evolved during recent years.

The framework to manage global security and access policies in an heterogeneous and distributed environment still has to be developed. There are some proprietary frameworks based on existing standards like SSL or X509 that partially cover these requirements. But dedicated and complex developments are still needed to enforce the desired security. Middleware providing (1) a strong authentication to access confidential information and (2)

efficient transfer of right from site to site, is foreseen as a “Grail” for most foreseen application domains.

4.6.1 Security of the execution platform

To cover the base of security, the middleware has to cluster resources to isolate information by user, application or transaction. A possible technical approach is to use the Multi Security Level (MLS) paradigm based on military studies.

In this kind of system, one has the ability to isolate data and context by establishing mandatory protection barriers. The end-user application runs in a trusted process. When a new process is initiated, a specific management application allows physical resources dedicated to this process (memory, cpu, communication bus, etc.). To limit low level attacks and to avoid information leakage, low level drivers and services are also started and dedicated to each specific process. Exchange between processes can be allowed through filtering, monitoring and protection mechanisms.

Specific management services control all physical resources. At the end of a process, they erase all residual data before allowing them again. One main issue, in the foreseen applications, is the use of specific mobile devices. Some of them have to be protected by specific mechanisms to avoid information disclosure.

4.6.2 Authentication / authorization

Management services must provide the authentication of users and the authorization to services and data in the system. They must reduce actions from the end-user but also perform rights transfer management between distant and heterogeneous systems.

4.6.3 Proof and Audit

To allow applications to build homogeneous proof transactions, the middleware has to offer different key services :

- Proof constitution by using digital signature
- Secured time stamping system
- Notarization of the essential elements
- Service of proof verification

Middleware applications will also have to conserve logs for an audit and proof of transaction in case of dispute resolution. These elements could be restored long after the operation so the system could perform notarization for the proof information needed for legal issues. These restitutions have to be simple and consistent even if data from different systems have to be recovered.

4.6.4 Threat identification

Once an environment is set, the system has to protect itself against intelligent attacks (not only network attacks). To perform this analysis without impacting the process,

this service has to be optimized to identify and eradicate those threats.

4.6.5 Security management

Some security management services have to be provided by the middleware to ensure and deploy global security policies define through a security analysis. Those services have to be compliant with some security methodology in use (for example BS7799).

Those security management services will also be highly critical. They will need to be protected with specifics mechanisms to avoid an attack with administrator rights.

4.7 Importance for each domain cluster

The table gives the relative importance of each research priority for the six domain clusters described in Section 3.

5. References

- [ARTEMIS 2004] ARTEMIS, “Building ARTEMIS”, Report by the High-level Group on Embedded Systems, 2004.
- [ARTEMIS 2005] ARTEMIS, “Strategic Research Agenda - Short Version”, June 2005.
- [ACARE 2004] ACARE - Advisory Council for Aeronautics Research in Europe, Strategic Research Agenda, 2004.
- [AMSD 2003] AMSD - Accompanying Measure System Dependability, “A Dependability Roadmap for the Information System in Europe”, CaberNet sponsored report, IST-2000-25088, 2004.
- [ARTIST 2003] ARTIST - Advanced Real-Time Systems, “Roadmap”, May 2003.
- [Bates 1998] John Bates, “The State of the Art in Distributed and Dependable Computing”, CaberNet sponsored report, 1998.
- [DSoS 1999] DSoS - Dependable Systems of Systems, “DSos Conceptual Model”, IST-1999-11585.
- [eMobility 2005] Mobile Communications and Technology Platform, “eMobility, Staying Ahead!”, March 2005.
- [eSafety 2002] eSafety, “Final Report of the eSafety Working Group on Road Safety”, 2002.
- [Haase 2002] Paul Haase, “Intelligrid: A Smart Network of Power”, EPRI Journal, pp. 28-32.
- [ITEA 2004] ITEA, “ITEA Technology Roadmap for Software Intensive Systems”, 2nd Edition, May 2004.
- [NEM 2005] NEM - Network and Electronic Media, “NEM Strategic Research Agenda, June 2005.
- [PROGRESS 2002] PROGRESS, “Embedded Systems Roadmap”, edited by Ludwig D.J. Eggermont, Technology Foundation Utrecht, NL, March 2002.
- [Romanowsky 2004] Alexander Romanowsky, “CaberNet Vision of Research and Technology Development in Distributed and Dependable Systems”, CaberNet sponsored report, IST-2000-25088, 2004.

RESEARCH PRIORITIES		DOMAIN CLUSTERS					
		DEVICE & PLANT	CRITICAL	PRIVATE / HOME	NOMADIC	SYSTEM OF SYSTEMS	AD HOC CONNECT.
Programming	Middleware as Key Enabler for Declarative Paradigm for Programming	****	****	****	****	****	**
Organization & Deployment	Dynamic reconfiguration capabilities	***	*	***	****	****	****
	Efficient user interaction	**	*	****	****	***	**
	Device and service discovery	**	*	****	***	***	****
	Ontologies	*	*	****	****	***	****
	Conflict resolution	*	*	****	***	***	**
Resource Management	Adaptive resource management	**	*	***	***	***	**
Data distribution	Application end-to-end QoS	***	**	***	***	****	**
	Data distribution	****	*	**	****	****	**
Robustness & Diagnosis	Error containment in a distributed and dynamic system	****	****	**	**	***	**
	Global diagnosis architecture and detection of correlated errors	***	***	**	**	***	*
Global connectivity	Connectivity in heterogeneous and constrained environments	***	*	***	****	***	****
Provably correct systems	Operating Systems & middleware for safety critical systems	***	****	**	**	**	**
	Designing and integrating provably correct systems	***	****	**	**	**	**
Security	Security	***	**	****	****	****	****

6. Contributors

The following contributors helped building this analysis and report, through group meetings, and many discussions on the phone, or by email.

ARM	Ian Phillips
CEA	Jean-Luc Dormoy
CEA	Christophe Lécluse
ENST	Laurent Pautet
Nokia	Sassan Iraj
Philips	Peter van der Stok
Thales	Virginie Watine
Thales	Hans Schurer
Thales	Ton Peerdeman
Thales	Antoine Delautre
TU Wien	Roman Obermaisser
FhG-IGD	Reiner Wichert
ARTIST - Uni KL	Gerhard Fohler
Ericsson	Andras Toth
ABB	Nis Leffler
University College London	Stefanos Zachariadis
STM	Alun Foster
Thales	Laila Gide
Ericsson	Andras Toth
Daimler	Vera Lauer
Airbus	Sylvain Prudhomme
Airbus	Marie-Line Valentin
Infineon	Knut Hufeld
CRF	Marco Novaro
Schneider Electric	Hervé Jacquet
Schneider Electric	Franck Bernier

We must also thank the ARTIST 2 Network of Excellence for his help, and fruitful contribution to this work.

The strategic objective of the ARTIST2 Network of Excellence on Embedded Systems Design (<http://www.artist-embedded.org/FP6/>) is to strengthen European research in Embedded Systems Design, and promote the emergence of this new multi-disciplinary area.

The ARTIST2 Network of Excellence implements an international and interdisciplinary effort to create a European virtual centre of excellence. It seeks integration within and between 6 clusters (distributed research teams), corresponding to essential topics including: Real-Time Components, Adaptive Real-Time, Compilers and Timing Analysis, Execution Platforms, Control for Embedded Systems, and Testing and Verification for Embedded Systems.

The activities in this ARTEMIS working group are most closely represented in the Adaptive Real-time Cluster (ART) of ARTIST2. The objective of this cluster is to affect the characteristics of next generation real-time

kernels/networks/middleware produced by the major vendors of this type of software, in order to support those embedded systems that operate in dynamic environments, where the load cannot be easily predicted in advance, but where a predictable timing behaviour is still required and the quality of requirements are of primary importance. The different expertise of the partners that compose the cluster represents a unique opportunity to achieve this goal.

The cluster provides expertise on adaptive real-time systems and networks, including middleware, quality-of-service and control issues. It is complementary to the Hard Real-time cluster of ARTIST2 with forms a crystallisation point for activities with the clusters adaptive real-time and control, timing analysis for adaptive systems. Furthermore, it works to reduce the gap between hard real-time and soft real-time. Current activities include providing a common infrastructure for adaptive real-time systems, flexible scheduling technologies, adaptive resource management for CPUs and networks, QoS aware components, and real-time languages. ■