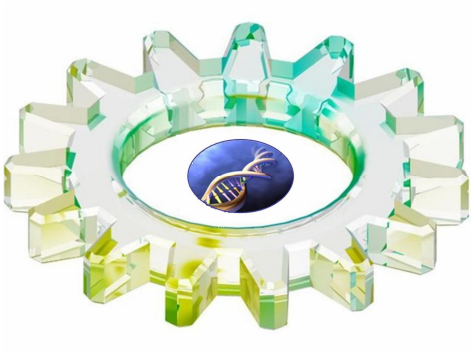

BIONETS

WP 1.1/3.1 – D1.1.4/3.1.4

SerWorks Architecture v. 2.0

Reference:	BIONETS/VTT/WP1.1/3.1
Category:	Deliverable
Editor:	Jyrki Huusko (VTT)
Author(s):	Jyrki Huusko (VTT), Janne Lahti (VTT), Helena Rivas (VTT), Daniel Schreckling (UNIPASSAU), Daniele Miorandi (CN), Francesco De Pellegrini (CN)
Verification:	Iacopo Carreras (CN), Paul Naoumenko (INRIA)
Date:	5/8/2010
Status:	Final
Availability:	Public

Executive Summary

In this deliverable we provide a revised version of the SerWorks architecture presented in the previous deliverable D1.1.3/3.1.3. The main components of the SerWorks architecture for BIONETS system were introduced already during the different phases of Subproject 1 and Subproject 3, in particular within deliverables D1.1.1, D1.1.2, D3.1.1 and D3.1.2. These deliverables presented various components and methods performing network-level as well as service-level functionalities, which formed the basis of the SerWorks architecture. The first SerWorks architecture definition was described in "SerWorks Architecture v1.0" [BIONETS_D113/313], where a first approach towards a unified architecture covering both network-level and application-level services was presented.

The previous SerWorks architecture definition illustrated some of the main architectural components and was mainly meant to introduce the overall BIONETS SerWorks concept, as it was defined by the project consortium during Y-2 and Y-3. Therefore, it did not include for example a description of the impact to existing systems and application scenarios. Furthermore, security considerations were only partially included in the first architectural description.

The main purpose of this deliverable is to provide a revised version of the SerWorks architecture, together with a set of comprehensive guidelines for system and network architects to build solutions compliant with it. We also provide some additional improvements with respect to the initial architecture definition, in particular in terms of support of security features. In addition, we illustrate the relevance of SerWorks by discussing its applicability to BIONETS application scenarios and existing systems.

The deliverable is organised as follows. In Section 1, we discuss the relevance and motivation for the Architecture and this deliverable. Section 2 illustrates the BIONETS Digital City application scenario and SerWorks impact to it. Section 2 describes the basic design principles in order to build up the system. In Section 3 we will discuss the inclusion of security features within the SerWorks architecture.

Document History

Version History

Version	*Status*	*Date*	*Author(s)*
0.1	*Created*	*1/6/09*	*Janne Lahti*
0.2	*Draft*	*5/6/09*	*Janne Lahti, Helena Rivas, Jyrki Huusko*
0.3	*Draft*	*8/6/2009*	*Janne Lahti, Helena Rivas, Jyrki Huusko*
0.4	*Draft*	*7/7/2009*	*Daniele Miorandi*
0.5	*Draft*	*8/7/2009*	*Daniel Schreckling*
0.6	*Draft*	*15/7/2009*	*Jyrki Huusko*
0.7	*Draft*	*16/7/09*	*Daniele Miorandi*
0.8	*Draft*	*17/7/09*	*Jyrki Huusko*
0.9	*Draft*	*19/7/09*	*Jyrki Huusko*
1.0	*Proposal*	*20/7/09*	*Jyrki Huusko*
1.1	*Final*	*13/8/09*	*Janne Lahti, Helena Rivas, Daniel Schreckling, Jyrki Huusko*
2.0	*Final*	*5/8/10*	*Jyrki Huusko, Daniele Miorandi*

Summary of Changes

Version	*Section(s)*	*Synopsis of Change*
0.1	*Not Applicable*	*Template and initial Table of Contents drawn*
0.2	*All*	*First contributions*
0.3	*...*	*...*
0.4	*2, 3*	*Revision of networking framework functionalities, components and APIs.*
0.5	*2*	*Security sections added*
0.6	*All*	
0.7	*Executive summary, 1, 2, 3*	*Refinements and additional material introduced*
0.8	*Executive summary, 1,2,3,4*	*Additional material and refinements*
0.9	*1, 3*	*Additional material and refinements*
1.0	*All*	*Proof-reading and material updates*
1.1	*All*	*Modifications based on the internal review*
2.0	*All*	*Revised to account for ESR comments*

Contents

1. Motivation and Relevance

Some of the current network environments are characterised by an extremely large number of networked devices possessing rich computing and communication capabilities. At the same time, there is a clear trend towards the creation of a highly ubiquitous network environment, where embedded devices get networked and integrate seamlessly within the environment. The devices in the ubiquitous environment are "shifted to the background", their presence being perceived only through the services they provide. This is consistent with the emerging trends towards pervasive computing and communication environments, where myriads of networked devices with very different features will interact with each other and with the environment, enhancing human's communication capabilities and quality of life.

At the same time, communication networks are becoming more and more information- and service-centric. This is causing some friction, as the conventional communication and service provisioning approaches start to be ineffective. The current networking solutions fail indeed to address well the heterogeneity at the device and at the service levels, the huge number of nodes raises scalability issues, node and network mobility question conventional device/network management approaches. From the communication point of view, the scalability and management issues decimates the possibilities to arrange global "always-connected" network infrastructure. In other words this means that the global network starts to resemble an archipelago of network islands.

Within the BIONETS project, we have envisaged a future computing environment characterised by the presence of large-scale networks of millions of heterogeneous devices, which outnumber the human users by several orders of magnitude. The devices utilise mobile and wireless communication means, and disconnected operations become the rule rather than the exception in the environment. On top of the networked devices, millions of localised ad-hoc services are running and accessible for the users and other devices.

In such environment we cannot rely on the availability of the backbone connection. The whole notion of traditional end-to-end connectivity will become less important, and the local connectivity between users and devices will increase. Conventional approaches for communications and service provisioning will suddenly become ineffective in such a context. The devices do not form a connected topology but they live in an ecosystem, in which highly dynamic islands of connected nodes form and disappear continuously. Services operating in such environments are bound to be autonomic, to evolve and to adapt to the surrounding environment.

In order to tackle the problems caused by the pervasive computing environments, the BIONETS project proposes an architectural solution – SerWorks - , which merges service- and network-level architectures and benefits from the biologically-inspired communication paradigms adopted by the project. In BIONETS, services become autonomic, being able to adapt to the surrounding environment by evolving their internal structure and configuration, like living organisms evolve by natural selection. Network operations will be driven by the services, providing an "ad hoc" support when and where it is needed to fulfil the users' requests. The unified service-oriented approach offered by SerWorks aims to enhance especially the flexibility of current network and protocol systems, by providing common methods for the network and service domain interactions able to ensure that the service-tailored protocols can be built and adapted on-the-fly according to service and user requirements.

The first consistent version of the SerWorks architecture was presented in [BIONETS_D113/313], including a description of the main components of the service and network levels, together with a set of common interfaces. The purpose of this deliverable is to provide a revision of the previously presented architecture, enhancing it in terms of support of security features and illustrating the impact

of the proposed solution to BIONETS application scenarios and to current research activities in the Future Internet research field.

The secondary goal of this deliverable is to provide a set of guidelines for the design and implementation of BIONETS-compliant systems. In addition we try to enhance the architectural approach from the BIONETS concept to more generic utilisation of SerWorks architecture.

The document is organised as follows. In the Section 2 we present some of the main components and underpinning principles of the SerWorks architecture. In addition, we discuss the advantages and potential impact of SerWorks with respect to the state-of-the-art in the field and to the BIONETS Digital City application scenario. In the Section 3 we illustrate the inclusion of security features within the SerWorks architecture. The final Section 4 concludes the deliverable. The main system components and interfaces are summarised at the Appendix B. More detailed information on system's components can be found in [BIONETS_D113/313].

2. Design Principles

The main purpose of this section is to illustrate the design principles of SerWorks architecture core components and to discuss the potential impact of SerWorks in terms of Future Internet research. The target is to provide a comprehensive high-level architecture definition for enabling system architects to build SerWorks-compliant solutions. This is also meant to serve the groups active in the implementation process of BIONETS project demonstrator; for such a reason we describe how the architecture maps to the BIONETS Digital City application scenario. We have also considered several other concepts on future and current internetworking to which the SerWorks architecture could have valuable impact.

In the following Section 2.1 we will discuss the impact of the BIONETS SerWorks technology to BIONETS application scenarios and existing systems and computing environments. We present the BIONETS "Digital City" scenario, illustrating by means of different use cases how some of the bio-inspired solutions developed within the BIONETS project can enhance the current systems in such environment. We will also compare the BIONETS SerWorks to some of the current technological trends and identify areas to which the BIONETS system is applicable and may offer competitive advantages. The Section 2.2 presents the main design principles of the SerWorks architecture and the Section 2.3 the security considerations for the BIONETS concept and SerWorks architecture. It is worth remarking that the latter aspect was under development during the preparation of [BIONETS_D113/313] and was left for this second revision of the architecture.

2.1.1 BIONETS Digital City Scenario

Some of the key technologies developed inside the BIONETS project will be demonstrated in a Digital City application scenario. The Digital City aims to model the peculiar needs and features of future sensor-rich urban environments, in which data and services are accessible through a ubiquitous digital infrastructure. The Digital City enables different threads of research; some of them are related to technology (sensors and real-time data or access modality), while others are more related to new services that can be offered in such a setting. In the application scenario, user-tailored services are provided through a "Personal Scout", a personal device which is able to interface with the environment as well as with other users through some form of proximity communications, and which is able to provide context-aware services to end-users. [BIONETS_D332]

The BIONETS autonomic system will be able to self-adapt to the flexible and unpredictable flows of data and user's requests in the digital city. The system will be also able to reason and take decisions based on the gathered information. The evolution of services is seen in this first phase as a novel way of performing service compositions and aggregation depending on data, usage and environment.

In the following we present one specific use case from the Digital City application scenario in order to demonstrate the technologies and solutions developed in the BIONETS project.

Digital City: Interactive maps

The Digital City represents a relevant environment for BIONETS, as it enables the demonstration of evolutionary aspects of BIONETS system. The digital city is characterised by a plurality of users moving around the city, following (at least partially) unpredictable behaviour and making use of different sources and types of dynamic information. An autonomic and self-adaptive system can manage and take decisions starting from this huge amount of different and unforeseeable data, which could be seen as a "data-cloud" above the Digital City.

In an urban environment there is a lot of different digital devices around us, either embedded in the "city landscape" (e.g., public displays, sensors of various types, tags etc.) or that we bring around with us in our daily activities (e.g. cell phones, laptops, digital cameras, music player etc.). The use of all these digital devices during human (and machine) daily activities is generating huge data-clouds

overlooking our cities. In such a context, the dynamics of a city can be captured in real-time by collecting and correlating data and information (anonymous localisation, traffic, pollution, cultural sites, events, etc) provided by these heterogeneous sources. Furthermore, by observing these data-clouds, it can be realised that cities behave like ecosystems with self-adaptive and self-organising properties. People moving and acting in a city base their decisions on information which is in most cases not synchronised with the time and place they find themselves in when taking those decisions. Experiencing a shift between decisions and information it is very common in everyone's experience: examples include arriving to the airport just to find out that the flight has been delayed or being surprised by an unexpected traffic jam. The Digital City is concerned with the real-time mapping of the city dynamics and data to augment personalised services. This information becomes a necessary instrument for city inhabitants, allowing inhabitants to base their actions and decisions on this better and more synchronised information. [BIONETS_D332].

The interactive maps scenario is one relatively simple example that involves a distributed collection and elaboration of data and information (from both real and virtual worlds). In this scenario, users in a city may want to run services on their mobile devices that require digital maps (e.g. of the area where the users are located). In particular maps can be downloaded either from distributed access points or from neighbours' devices (avoiding to the need to access a centralised server). Standards maps are downloadable from access points. The richer maps, which can contain other content and information such as personalised and user created (audio files, images, commercial text), can be shared by users in peer-to-peer mode in each area. Real-time information (e.g. traffic jam) can be further added and correlated to maps.

Let us consider a user, who would like to find the nearest restaurant and the best route to it. The map of the area (also containing commercial information about restaurants) is downloaded from the device of a neighbouring user. Best route and traffic information is also correlated to its location. Let's assume that the restaurant has also a virtual representation in a Virtual World (e.g. Second Life). The user may want to pre-access such a virtual representation of the restaurant to get some preliminary commercial information on menu and costs.

This scenario requires technologies and (self-*) solutions capable of gathering and manipulating huge amounts data and information in order to provide distributed applications with more abstracted and correlated data. The scenario demonstrates the advantages that the BIONETS solutions will bring to the urban city environment. Observed from different perspectives, the potential benefits for different habitants of the digital city can be e.g.:

- Citizens: better knowledge of events, opportunities and environmental conditions concerning their local surroundings;
- Local authorities: better understanding of the urban system and its dynamic evolution, e.g. distribution of pedestrians, vehicles, tourists etc. in the city at different times of the day and their correlation with the events happening at specific locations (based on communication network analysis);
- Companies: ability to better promote and distribute their services/products to the local population (location and time-based eBusiness);
- Mobility: better use of public and private transports based on real time information about schedules, paths, delays, traffic condition.

The use cases such as described in the BIONETS Digital City are already emerging in a variety of settings. Targeted advertisement, experience sharing and data communications are increasing, especially in the urban environment. At the same time, the device density (including both user devices with global network connectivity and wireless and wired sensor network devices) is increasing rapidly. This leads to a situation where it is very challenging to guarantee the access and connectivity, as the network infrastructure may suffer from heavy load. In addition, the complexity related to system management tasks is growing, causing additional operational expenditures (OPEX) and harming the margins of telecommunication system and network operators. The traditional networking approaches are able to cope with urban area use case requirements in some level, but when for

example the autonomy of the management is required new approaches are needed. In a city-like environment, where figures for the node/user density and mobility are high, the SerWorks architecture and the BIONETS solutions represent competitive solutions to more traditional ones.

2.1.2 Impact to Current Systems

The SerWorks architecture was developed as a basis for developing networked systems supporting bio-inspired networking and service approaches, autonomic management of components and devices, , and evolutionary services running in an environment characterised by a multitude of wireless sensors and of user devices utilising and processing the environmental data.

Future Internet fields and applications, including, e.g., Internet of Things, Internet of Services and 3D Internet, are characterised by the user's ability to compose and create new addressable objects (namely services and content) easily. As stated for example in [eMobility08] the emergence of these new application areas will have a major impact on the architecture of the Internet, as the global network is moving towards a situation where the infrastructure management, content and services are becoming self-contained. Object creation and lookup, security and new methods for managing resources and infrastructure will become major research and engineering challenges in such future Internet scenario. The SerWorks architecture and the overall BIONETS concept could have a major impact on the previously mentioned future Internet scenarios as SerWorks is, by default, designed to handle complex service and network management functionalities and object discovery/creation mechanism with inbuilt security functionalities. Although the BIONETS concept and SerWorks architecture concentrates mainly on opportunistic communication mechanisms at the network level, it is not limited to it and the architecture can be applied also to evolutions of IP networks or to solutions making use of new routing schemes such as the Generic Path (GP) concept [Biermann09].

The SerWorks architecture builds around a variety of principles. In particular:

- Network-level functionalities are performed in a *data-centric* fashion. While this contrasts with the conventional address-centric approaches to networking (e.g., IP), data-centric networking (also referred to as information-centric networking) is currently envisioned as one of the most promising research directions for overcoming the limitations of the current Internet and transforming it into a content-centric infrastructure [FIA-FCN-2009a,Ohlman09]. While BIONETS targets "fringe" networks, arising at the edges of the Internet, some of the solutions devised within the project could be in principle adapted for infrastructure-based systems. Further, data-centric networking is an already established research area in Wireless Sensor Networks (WSNs). The SerWorks architecture represents a step forward, providing a holistic approach to both networks and services that could enable WSNs to become reconfigurable, service-oriented platforms.
- Network services are treated in much the same way as application-level services, and can be composed at runtime according to the needs of the running applications. While this approach has been proposed in the past as a research topic (see [BIONETS_D113/313] and discussion therein), it has not found application in real-world technologies so far. A SOA-like approach to network services could help in deploying highly reconfigurable network-level solutions, representing a potentially useful feature for the experimental testing of novel networking approaches over reconfigurable (virtualised and federated) infrastructures [FIRE09].
- The essence of autonomic computing systems, such as the BIONETS service framework, is self-management, which intends to free users and operators from the burden of system operation and maintenance. The self-management procedures in an autonomic system can exist at many levels. At first, automated functions can merely collect and aggregate the available information to support decisions by users. Or, procedures can serve as advisors, suggesting possible courses of action for users to consider [Kephart03]. In some of the current systems, the autonomic systems can even make some lower-level decisions on behalf of the user. The purpose of the BIONETS system is to take the autonomic procedures even further,

where the users will need only to take relatively less frequent and predominantly higher-level decisions, which the system will carry out automatically via more numerous, lower level decisions and actions. Traditional service-oriented architectures like Web and grid services can be used as foundations of future autonomic service environments, but they do not answer to all the problems. For example, when acting as a service provider, the Service Cells will not honour requests for service unquestioningly as a typical Web service would do. The autonomic service will provide a service only if it is consistent with their "higher" goal or individual benefit. Also, as consumers, autonomic elements will autonomously and proactively issue requests to the other elements in order to carry out their objectives. Another difference with the traditional service-oriented concepts is that the autonomic elements will have complex life cycles, continually carrying on multiple threads of activity, and continually sensing and responding to the environment in which they are situated.

2.2 Basic Design Principles

In this section, we will present the basic design principles of BIONETS systems based on the SerWorks architecture. The purpose of this section is to present the main characteristics, functionalities and interactions of BIONETS SerWorks components, important to developers/users of BIONETS systems. The components and the interfaces required for component communication are described in more detailed in the following Appendix B.

2.2.1 The SerWorks Architecture

The core of the SerWorks architecture consists of three frameworks; the Service Framework in the upper layer of the architecture, the Interaction Framework in the middle layer, and the Networking Framework in the lowest layer. The Service Framework includes the application/service-level logic and the functions supporting their distributed and autonomic execution and management. The Interaction Framework provides means for establishing and maintaining communication between Service and Network Framework management embodiments, which are residing on different nodes, by offering multiple concurrent interaction models. The interaction models are realised as a shared data spaces to support the efficient communication between the distributed services. The Networking Framework provides the basic communication capabilities in the disappearing network context and supporting means for establishing secure communications over either opportunistic network or connection-oriented networks such as IP based networks.

At the node architecture level, the SerWorks architecture defines two classes for nodes, U-nodes and T-nodes. The T-node class includes, e.g., small wireless sensor nodes, whereas the U-node class is an abstraction of more powerful user devices, with one or more network interfaces and able to use, process and relay the information collected from T-nodes. These classes are described in more detail in [Carreras07] and [BIONETS_D113/313].

The following Fig. 1 illustrates the common SerWorks architecture with the main interfaces required for the system. The red arrows illustrate the management interfaces and the grey arrows data communication interfaces.

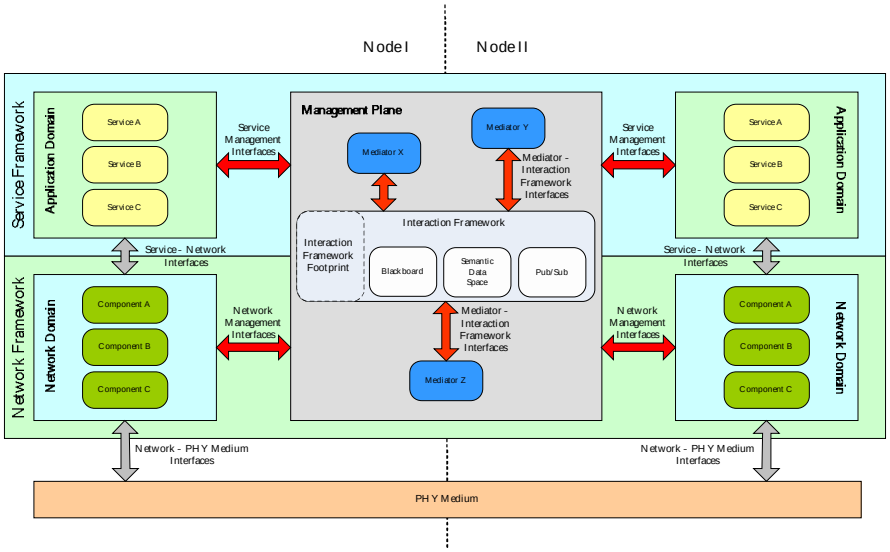

Fig. 1. The Common SerWork Architecture

The main idea of SerWorks is to apply a service-oriented approach to both application- and network-level functionalities. This implies that similar interaction paradigms and control/management mechanisms can be used for both application domain components (i.e. user services) and network domain components (i.e. network services). The management of these system components is handled by the Mediator entities through the Interaction Framework. The Interaction Framework Footprint depicted in Fig. 1 is a simplified Interaction Framework implementation for small devices and sensors with limited processing power.

The management interfaces provide the basic primitives to manage and control the system through the Interaction Framework. The "Service – Network" and "Network – PHY Medium" interfaces are generic communication interfaces for e.g. user data transmission and they can be implemented e.g. as traditional Internet socket interfaces depending on the underlying network.

Service Framework, Interaction Framework and Network Framework are treated independently. This makes it possible to implement and run the SerWorks architecture on current devices and to implement the BIONETS bio-inspired concepts for those. For example, the underlying network can be IP-based and only the services are implemented to support BIONETS paradigms. In this case, only the Service Framework is implemented with Mediators. Similarly, for network domain management, only the Network Framework with corresponding management plane Mediators and Interaction Framework can be implemented. The BIONETS SerWorks architecture can be utilised in a variety of applications, starting from the future information dissemination networks to plain autonomic network management of IP networks.

Addressing and forwarding/routing schemes are dependent on the different network and service functionalities. The generic SerWorks architecture provides the possibility to use also different types of routing schemes. For example inside BIONETS, the routing and forwarding is based on opportunistic forwarding schemes, which are carried out by the U-nodes. The addressing of the services within BIONETS is based on the semantic service description. In this way, it is possible to provide more comprehensive information also from the service content. In addition, the specific service IDs are used for security and trust guarantees. The Section 3 of this deliverable addresses the security concept for the BIONETS SerWorks architecture. Although the BIONETS specifies certain guidelines for implementing the SerWorks architecture, the generic SerWork architecture can be implemented to support also the state-of-the-art addressing and routing schemes as long as the Interaction Framework and Mediators support the functionalities and the corresponding management interfaces are implemented for the Network and Service frameworks.

2.2.2 Service Framework

The BIONETS service framework, illustrated in Fig. 2, plays a two-fold function. On the one hand, it provides a runtime environment for the execution of the service logic running on U-nodes and T-nodes. On the other one, it includes the capabilities to react in an autonomic way to changes in the environment. These capabilities include different evolution and adaptation strategies, service life-cycle handling and mobility support [BIONETS_D321].

The main building blocks of the service framework are the Service Cells (SC)/Service Individuals (SI) and the Service Mediators [BIONETS_D311, BIONETS_D312]. In order to achieve node-level autonomy in a flexible way, the Service Mediators were introduced as architectural elements complementary to the services. Service Mediators behave like agents for the services, sharing the same set of physical resources. Services and Service Mediators interact through the Interaction Framework, no matter where they are hosted.

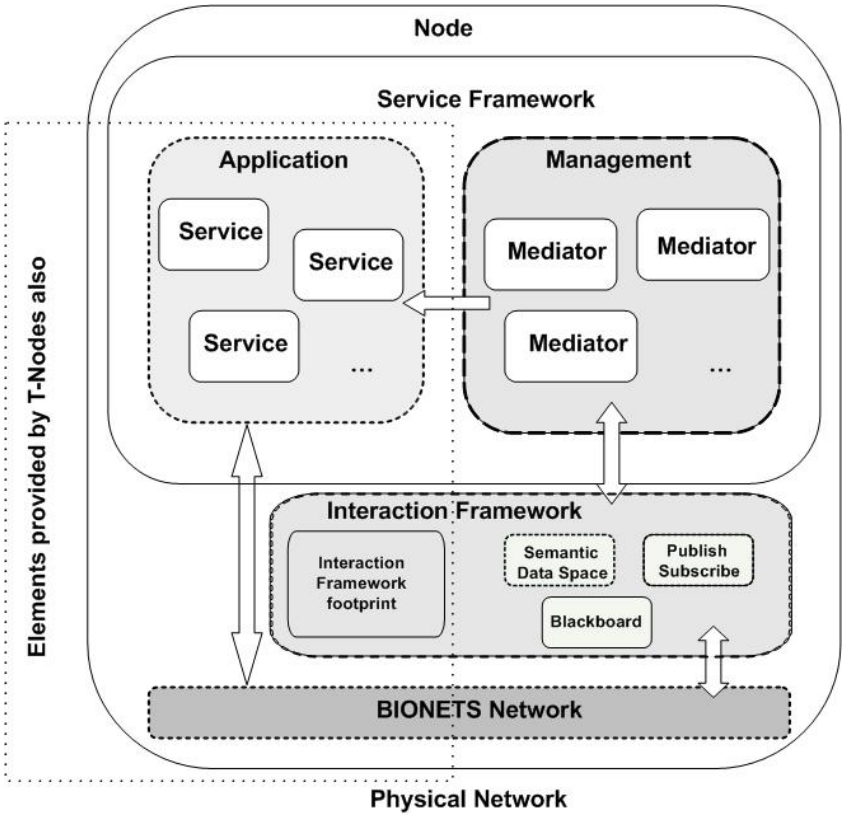

Fig. 2. SerWorks Service Framework

2.2.2.1 Service Definition

In BIONETS the definition of service follows the OASIS service definition:

The service is a mechanism, which enables access to one or more system capabilities. The access is provided using a prescribed interface and is exercised consistently with constraints and policies as specified by the service description [OASIS06].

Unlike OASIS Service-oriented architecture (SOA), BIONETS does not distinguish between application- and network-level components, so in effect any network and application domain mechanism that fulfils the above mentioned definition is considered as a service within the SerWorks architecture. In other words, the Service Cells, Service Individuals and Network Components which obey the service definition by default are treated similarly in the system. In addition, in the SerWorks architecture the services need to follow strictly the various phases encompassed by the BIONETS

service life cycle [BIONETS_D321] and to provide an interface for the specified Service Mediator components.

2.2.2.2 BIONETS Service Life-Cycle

Within BIONETS, we focus on the evolution of services over time, which causes modification to the internal structure of service composition (e.g. through the application of genetic operators), potentially modifying also to service's non-functional properties. The current service life-cycles do not support this kind of modifications at run-time, but require a "re-deployment" of the modified services. The BIONETS architecture, however provides an environment that allows services to evolve by freeing them from static descriptions and relying on behavioural observation instead, allowing the emergence of completely new services through long-running evolutionary processes.

The BIONETS service life-cycle is designed to allow services to emerge in an autonomic fashion in order to gain best support for user tasks while reducing the user's efforts in the service creation process. This goal is regarded against the background of pervasive service environments, composed of services originating from the users' computing devices and relevant in the users' context and their direct surrounding.

The BIONETS service life-cycle addresses at the beginning two major objectives, the on-demand creation of service by the user and the continuous adjustment of services to cope with changing requirements in the computing environment. While the first objective addresses basics to create a working service environment, the second objective addresses the progressive improvement and adaptation of services in the presence of highly dynamic computing infrastructures. According to the objectives, we defined three phases for the life-cycle, Preparation and Creation, Evolution and Integration. The Preparation and Creation phase comprises activities around the acquisition of user needs and contextual information as well as around the creation of functionally appropriate Services addressing the user's request. The Evolution phase covers all activities concerning improvement, adaptation and proactive creation of Services. Finally, the activities of Integration phase address the actual handling of Services in BIONETS environments.

2.2.3 Interaction Framework

Services and service mediators interact through the Interaction Framework, which decouples the Service Framework operations from the underlying communication protocols, naming/addressing schemata and network characteristics, e.g., the disappearance of nodes from a connected island. The Interaction Framework contains a variety of interaction models. The interaction models have a middleware character and they simplify, e.g., the coordination of service mediators. The purpose of the Interaction Framework is to guide the design of special communication and coordination services that allow application services and Service Mediators smooth and open working. These communication and coordination services, also called Interaction Models, are built on a few, atomic operations for resource manipulation. In the original description of the Interaction Framework, these operations were Create, Read, Update and Delete (abbreviated as CRUD). CRUD operations are supposed to be supported by the networking framework. In practical tests, the Hypertext Transfer Protocol (HTTP) pointed out as appropriate example for a network-side CRUD realisation.

In practice, the Interaction Framework on each node reflects a finite number of named resources that can be accessed by CRUD principles, but only the Interaction Framework on U-Nodes allows the operations on resources of other nodes. The behaviour of these resources, i.e. their reaction to CRUD operations, is defined through the respective Interaction Models. Thus, the Interaction Model introduces a meaning for the CRUD operations.

2.2.4 Networking Framework

The network interfaces provide communication primitives, which are implemented in order to cope with the disappearing network context. The main objective of the Networking Framework is to

provide appropriate means for effectively supporting networked services in a disappearing network environment [BIONETS_D113/313, BIONETS_D123].

The BIONETS networking framework includes eight functional modules (components), as detailed in the SerWorks Architecture v. 1.0 [BIONETS_D113/313], each one corresponding to a set of functions addressing a specific task. These modules are:

- **Opportunistic communications**: handles are communications with the network interface card(s) used, including (point-to-point) transmission/reception of packets and node discovery. It also performs fragmentation/defragmentation of service data units in order to fit the requirements of the underlying wireless technologies in terms of maximum payload length.
- **Naming system**: handles the way data is described (including definition of a set of attributes that can be used to describe content of data and for performing queries to retrieve them). It also handles mapping between node's low-level addresses and its name.
- **Information filtering**: handles the filtering of information included in service data units. It includes methods for instantiating new filters (described as a set of <attribute,value> pairs) and for removing existing ones.
- **Data dissemination**: handles the opportunistic dissemination of messages among U-Nodes; it includes primitives for spreading data virally as well as for searching for data over a BIONETS disappearing network.
- **Interoperability with IP networks**: handles protocol conversion operations to leverage possibly existing legacy IP networks (e.g., WiFi access points etc.). It enables the use of throwboxes as described in [BIONETS_D123]. It works as a proxy between BIONETS sub-networks and IP-based ones.
- **Information gathering**: handles all procedures related to the retrieval of data from T-Nodes in proximity, including querying of T-Nodes (when operated in the 'proactive' mode) and reception from T-Nodes (when operated in the 'reactive' mode).
- **Secure communications**: Ensures secure, i.e. authentic, confidential, and integrity protected message exchange between peers using cryptographic primitives able to adapt to the type and characteristics of the environment and communication parties. This module also enhances communication with privacy and trust and reputation mechanisms.
- **P2P cloud**: provides the necessary abstractions for transparently accessing data present on a BIONETS island (connected subset of nodes). It provides the interface for the Interaction Framework and Service Framework to access networking functionalities. Its operations are based on unstructured peer-to-peer systems.

Each of the components is represented by one (or multiple) service cells. Different service cells implementing one of the specific components can be present. Late binding functionalities are provided through appropriate mediators.

3. Security in SerWorks

The three components, the Service Framework, the Interaction Framework, and the Networking Framework, of the SerWorks architecture are also reflected in its security architecture. While the Service Security Framework addresses potential vulnerabilities of service-level logic through automatic generation, service distribution, or evolutionary modifications, the interaction security mechanisms ensure that concurrent interactions comply with the security requirements for the data processed and with the security requirements specified for the interacting users and services. Finally, the networking framework supports the secure communication in the disappearing network. In particular, the highly heterogeneous and transient interactions are addressed on this layer.

In general, security in SerWorks is designed to be transparent to the developer and user. While the user will have to interact with the system to define security requirements for his devices, himself, and his data, the security system adapts its security measures according to the given context. The principles how to achieve this transparency are outlined in the remainder of this section.

The following Fig. 3 illustrates the main security components at the network and service domain and management plane for the SerWorks architecture. The following subsections will illustrate the main functionalities of these security components. The more detailed description of the security concept is available at final deliverable of WP4 at the end of the project.

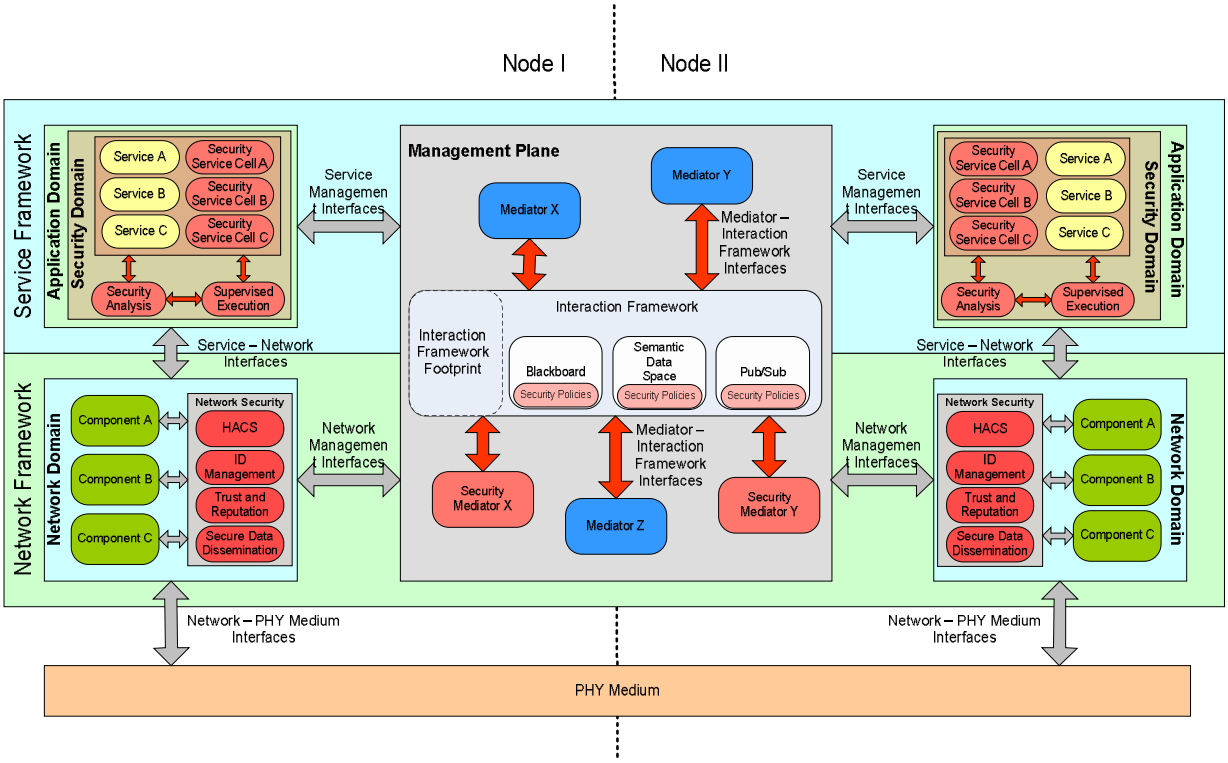

Fig. 3. SerWorks Architecture Security Components

3.1.1 Identity Management

Before we discuss the single security components mentioned above, this section clarifies the definition of the security relevant identifiers for the single entities in BIONETS, i.e. U-nodes, T-nodes, and services. As we can not present all the details in this section we refer the interested reader to deliverable D4.5 [BIONETS_D45].

3.1.1.1 U-Nodes

According to an adopted public key cryptosystem, each U-node in a BIONETS island stores a number of public and private key pairs. Candidates for these cryptosystems are for example elliptic curves as they appear most suitable for applications in mobile devices. This allows the use of common protocols for mutual authentication, e.g. ECSDA, as well as for secure key exchange, e.g. ECDHA.

An identifier that describes a verifiable physical property of a person or a device associated with the U-node is called a *physical identifier*. It uniquely determines a person, device, or some other entity. Physical identifiers include personal or company names, a person's age, phone numbers, email or home address, web URIs or information stored in tamper-resistant modules. These properties are assumed to be persistent over a period of time of interest. Node authentication based on physical identifiers consists of verifying the corresponding physical properties together with the knowledge of the secret key corresponding to the public key associated with the node.

Without physical properties, linked to a specific public key, the remaining U-Node identifiers degrade to *pseudonyms*. The only property of a pseudonym of a given node is that it is repeatedly used over time. Pseudonyms may be unique or not. In general, confidence in devices using physical identifiers is higher compared to pseudonyms. However, depending on the past behaviour of a node with a given pseudonym in the past, appropriate confidence can be established. Authentication of a node which uses pseudonyms exclusively is realised by proving the possession of the private key.

3.1.1.2 T-Nodes

Identities of T-nodes are based on physically verifiable and tamper-resistant characteristics, e.g. an identifier stored in a trusted module. The tamper resistance makes it possible to verify this ID by U-nodes in a remote way, without any direct physical contact with the node. In this way, U-nodes can issue certificates to T-nodes.

T-nodes which are not able to authenticate themselves, e.g. class 0 T-nodes, are administered by a hierarchical structure of T- and U-nodes. A dedicated node assigns T-nodes locally unique identifiers. Together with a secret symmetric key installed on the node during the setup phase, the dedicated node is able to communicate securely with the T-nodes assigned to it.

For authentication purposes the T-nodes or the dedicated nodes can generate a dynamic symmetric key using the pre-installed secret key and a publicly known key derivation function.

3.1.1.3 Services

The naming *IDs* of services in BIONETS enhanced by the following structure (*IDs*, *h*(*Ps*), *vs*, *ns*). *IDs* is a semantic description described in [BIONETS_D45] of *s*, the *vs* is a service version number, and *ns* is the hierarchically structured description of a service part or element, i.e. a service individual or service cell. *Ps* is the public key of the key public, private key pair (*Ps*, *Ss*) generated during service creation. *h*(*Ps*) is the hash over this key and as such basically identifies the key. The part of the name (*IDs*, *h*(*Ps*)) should identify the service in a globally unique way and should not change during the life cycle of the service, despite the service changes during its life cycle. *vs* is used to mark changes in a service, i.e. through modification due to adaptation, evolution, or user interaction.

The trust in the service derives from the trust in the public key from the name of the service. For this reason, it is also possible to use the same public key in the name of different services, in which case the trusts associated with these services can be aggregated. For authentication purposes the BIONETS node where a service is generated or modified generates digital signatures for the service. To increase the reliability of the system [BIONETS_D45] additionally proposes to regenerate a public-, private key at the device which receives the service.

3.1.2 Service Security Framework

The service security framework supports the two main functionalities of the BIONETS service framework: the service generation and adaptation and the service deployment.

In the first case the service security framework ensures that automatically and autonomously generated services do not implement vulnerable functionalities which would not comply with the security policies of the data the service operates on or with the security needs of the user who wants to deploy the appropriate service. Thus, the service security framework employs analytical mechanisms which test the generated or modified services for security vulnerabilities or breaches before its execution. At the end of this analysis, the service becomes subject to additional modifications to satisfy the security needs of the user, the platform, or the processed data. For this purpose the service security framework makes use of atomic security services, e.g. cryptographic primitives, authentication functions, integrity protection mechanisms, etc., implemented in service security cells (see also [BIONETS_D42, BIONETS_D44]).

The mechanisms described above are not able to cover all possible security vulnerabilities nor are able to reliably identify a real vulnerability. For this reason the service deployment is additionally protected by service security mediators. They monitor intra-service (interaction between service cells and individuals of one service) as well as inter-service (interaction between different services) interactions and prevent unauthorised information flow, service cell/individual execution or migration. For this purpose they also use the security information provided by the service interaction framework discussed in the next section.

As service compositions do not necessarily need to conflict with specific security requirements they may still expose malicious behaviour or at least may be used to support actions which harm the general performance of the system or of some devices belonging to particular users. Thus, the service security framework also provides security mechanisms which are based on behaviour, in particular, on trust and reputation [BIONETS_D41, BIONETS_D44, BIONETS_D45].

3.1.3 Interaction Security Framework

Security-wise, BIONETS considers data as the most important resource. Thus, although the interaction framework offers the very simple CRUD principle the transparent U-Node functionalities ensure that the named resources stored in the Interaction Framework are only processed according to security policies associated with these resources. This is achieved by providing each resource with a short security tag which contains a policy. These tags are delivered to the service security framework to perform a static analysis on the service before execution and to monitor the appropriate information flow during runtime. Tags are also by the security interaction framework itself. Access by CRUD primitives is monitored by the framework and in case of action conflicting with the data, user, or node security policy, the appropriate primitive is denied.

3.1.4 Network Security Framework

As mentioned above, the main objective of the networking framework is to provide appropriate means for effectively supporting networked services in a disappearing network environment. Consequently, the network security framework has to provide mechanisms which address the corresponding network characteristics. Thus, the security module of the framework provides and uses different service cells which enable trust and reputation on the network and node layer [BIONETS_D41, BIONETS_D44], provide anonymity and fairness for data dissemination [BIONETS_D42], and offer adaptive security primitives to secure network communication in a feasible way – according to the present context [BIONETICS_2006, BIONETS_D44].

To achieve higher adaptivity and to account for required security updates the network security framework introduces one more level in the service hierarchy, service cell atoms. These very simple arithmetic and logic operations allow the assembly of security primitives at service cell level. Again

mediators are used for providing late binding and dynamic composition. Additionally, they guarantee the security requirements of the corresponding security primitives.

4. Conclusions

In this deliverable, we have introduced the revised SerWorks architecture based on the first revision of SerWorks architecture definition in [BIONETS_D113/313]. The main improvements compared to the previous version of the architecture relate to the inclusion and alignment of the security components. In addition, two new network mediator components, which were missing from the previous architecture revision are presented and explained in Appendix B, which briefly describes also the main system components. In this deliverable we have also discuss the applicability of the SerWorks architecture in the Digital City application scenario and elaborated on the impact of the SerWorks architecture to current Future Internet research paradigms on ubiquitous data/information centric networking. The target of this deliverable was to provide generic guidelines for system architects to build-up a SerWorks-compliant system and to go beyond the BIONETS concept and introduce the SerWorks architecture as a generic and reusable solution.

References

[Biermann09] T. Biermann (Ed.), "Description of Generic Path Mechanism", 4WARD project D5.2, May 2009, [online]: http://www.4ward-project.eu/index.php?s=file_download&id=38

[BIONETICS_2006] D. Bliefernicht, D. Schreckling: "Highly Adaptive Cryptographic Suites for Autonomic WSNs", In International Workshop on Technologies for Situated and Autonomic Communications (SAC 2007), Budapest, Hungary, in conjunction with BIONETICS'07, Dec. 10-13, 2007.

[BIONETS_D111] D. Miorandi (Ed.), "Application scenario analysis, network architecture requirements and high-level specification", BIONETS D1.1.1, Jul. 2006.

[BIONETS_D112] D. Miorandi (Ed.), "Architecture, Scenarios, and Requirements refinements", BIONETS D1.1.2, Aug. 2007.

[BIONETS_D113/313] D. Miorandi, J. Huusko, F. D. Pellegrini, H. Pfeffer, D. Linner, C. Moiso, and D. Schreckling, "D1.1.3/3.1.3 serworks architecture v1.0," BIONETS Deliverable (D1.1.3/3.1.3), 2008.

[BIONETS_D122] D. Raz and F. De Pellegrini (Eds.), "Disappearing Network Autonomic Operation and Evolution", BIONETS D1.2.2, Jul. 2007.

[BIONETS_D123] G. Koukoutsidis and F. De Pellegrini (Eds.), "Service Matched Networking", BIONETS D1.2.3, Aug. 2008.

[BIONETS_D133] V. Simon, S. Szabó (Ed.), L. Bacsárdi, M. Bérces, E. Varga, G. Koukoutsidis, F. De Pellegrini, J. Latvakoski, T. Hautakoski, "BIONETS Performance Analysis", BIONETS D1.3.3, Dec. 2008.

[BIONETS_D312] J. Huusko (Ed.) "Refinement of Service Architecture and Requirements", BIONETS D3.1.2. Jul. 2007.

[BIONETS_D321] H. Pfeffer, D. Linner, S. Steglich and I. Radusch (Eds.), "Specification of Service Life-Cycle", BIONETS D3.2.1, Feb. 2008.

[BIONETS_D323] F. Baude and L. Henrio (Eds.), "Graph-based Service Individual specification: Creation and Representation", BIONETS D3.2.3, Jan. 2008.

[BIONETS_D324] J. Lahti (Ed.), "Advanced Service Life-Cycle and Integration", BIONETS D3.2.4, Jul. 08.

[BIONETS_D332] S. Elaluf-Calderwood, "Economics for BIONETS Business Models", BIONETS D3.3.2, Jan. 2009.

[BIONETS_D41] A. Garg, R. Cascella (Eds.), "Trust and Reputation Management System Definition", BIONETS D4.1, June 2007

[BIONETS_D42] D. Schreckling (Ed.), "Towards Security in BIONETS", BIONETS D4.2, Aug. 2007

[BIONETS_D44] D. Schreckling, "Adaptive Security in BIONETS", BIONETS D4.4, Feb. 2009

[BIONETS_D45] M. Brunato (Ed.), "Identities, Authentication, Trust and Reputation in BIONETS", BIONETS D4.5, Jul. 2009

[BIONETS_D54] I. Carreras (Ed), "Pervasive ubiquitous peer-to-peer context-aware application", BIONETS D5.4, Aug. 08.

[Carreras07] I. Carreras, I. Chlamtac, F. De Pellegrini, and D. Miorandi. "Bionets: Bio-inspired networking for pervasive communication environments." *IEEE Trans. Veh. Tech.*, 56:218–229, Jan. 2007.

[DePellegrini08] F. De Pellegrini, I. Carreras, D. Miorandi and C. Moiso, "R-P2P: a Data Centric DTN Middleware with Interconnected Throwboxes", in Proc. of Autonomics 2008.

[eMobility08] J.M. Cabero, T. Frantti, R. Giaffreda, J. Huusko, L. Munoz, R. Sofia, R. Tafazolli, M. Ylianttila, T. Zseby and D. Zeghlache (Ed.), "White Paper on Future Internet in a Post-IP era", eMobility CA D3.2, Sept. 2008, [online]: http://www.emobility-ca.eu/deliverables/D3.2%20White_Paper_on_Future_Internet_in_a_Post_IP_era_FINAL.pdf

[FARHA06]Farha, R. and A. Leon-Garcia. Blueprint for an Autonomic Service Architecture. In Autonomic and Autonomous Systems, 2006. ICAS '06. 2006 International Conference on. 2006.

[FIA-FCN-2009a] FIA Future Content Networks (FCN), "Why do we need a content-centric Internet? Proposals towards content-centric Internet architectures", May 2009, [online]: *http://www.future-internet.eu/fileadmin/documents/prague_documents/FIAFCN_Internet_Architecture_20090507.pdf*
[FIRE09] S. Avessta et al., "FIRE White Paper", May 2009, [online]
http://www.ict-fireworks.eu/fileadmin/documents/FIRE_White_Paper_2009_v1.02.pdf
[FOSTER02]Foster I. et al., "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," Feb. 2002; http://www.globus.org/research/papers/ogsa.pdf.
[Kephart03] Kephart, J.O. and D.M. Chess, The vision of autonomic computing. Computer, 2003. 36(1): p. 41-50.
[Kreger02]Kreger, H., "Web Services Conceptual Architecture," v.1.0. 2001; http://www-4.ibm.com/software/solutions/webservices/pdf/WSCA.pdf.
[OASIS06] OASIS Reference Model for Service Oriented Architecture 1.0, Official OASIS Standard, Oct. 12, 2006 [online]: *http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf*
[Ohlman09] B. Ohlman (Ed), "First NetInf architecture description", 4WARD project D6.1, Jun. 2009, [online]: *http://www.4ward-project.eu/index.php?s=file_download&id=39*

Appendix A: Terminology

- **T-Nodes**. T-Nodes are simple, inexpensive devices with sensing/identifying and basic communications capabilities. T-Nodes act as an interface with the environment and are used to gather contextual information which is utilised by the U-Nodes to provide context-awareness. T-Nodes do not communicate among themselves but are just read by U-Nodes in proximity. They present minimal requirements in terms of processing/storage/communications.
- **U-Nodes**. U-Nodes are complex, powerful electronic devices with computing and communication capabilities. PDAs, laptops and smartphones represent examples of a U-Node. U-Nodes are typically carried around by users and therefore are inherently mobile. Their mobility is exploited, in BIONETS, to provide system-wide diffusion of messages. U-Nodes host services. They interact with the environment through T-Nodes, from which they gather the contextual information necessary to provide the users with services enhanced by context-aware features. U-Nodes may communicate among themselves to exchange information, such as environmental data or service-specific code (in order to enable service evolution).
- **Access Points (APs)**. Access Points are complex powerful devices that may be used for (i) accessing IP-based services by the BIONETS networks (ii) collecting environmental data (through BIONETS system) from a remote IP service (iii) providing IP shortcuts among disconnected BIONETS islands. APs are envisioned to act as *proxies* between BIONETS networks and IP networks.
- **Service cell (SC)**: an atomic logic entity, which may provide knowledge, content, or functionality to other services and users.
- **Service individual (SI)**: a SI is a composition of Service Cells. Service Individuals may also be composed of other Service Individuals in a recursive way. Service Individuals are expressed with a graph representation [BIONETS_D323], wherein nodes represent types of Services (the elements of the composition); edges (links) represent the interactions (communication) between the Services.
- **Mediators**: logic entities performing control functions related to the operations of SCs and SIs. In particular, mediators implement the autonomic functionalities encompassed by the BIONETS service lifecycle.
- **Messages**. Communications in BIONETS are based on the exchange of messages. Messages are service data unit, i.e., encapsulation of data items meaningful to a service. In general, messages will be much larger than standard IP packets. (This is because single IP packets usually do not expose meaningful data to the service layer.) Messages will consist of a payload (or content) and metadata (expressed as a set of <attribute,value> pairs) carrying the necessary information for the node to decide which operations should be undertaken. Communications in BIONETS are asynchronous and connectionless. Messages are treated as datagrams, and the whole system can be thought as a message-switching engine.
- **Opportunistic Forwarding**: mechanism for diffusing information in a highly partitioned network, based on the exploitation of "contact opportunities" between nodes in the system. Opportunistic forwarding is based on localised interactions only and exploits mobility of the nodes to ensure network-wide diffusion of messages.
- **Information Filtering**: mechanism for limiting the diffusion of data messages with low information content. This is related to the fact that, in most context-aware applications, context-related data looses its usefulness when being far (in both space and time dimensions) from the originating context. Information filtering is an essential building block of data management in BIONETS systems.
- **Identifier**: an identifier is a finite sequence of symbols of a given alphabet, used to identify an entity within a set of entities. Identifiers have a scope (in space and time) which determines the domain within which they can be used for identifying entities.

- **Name**: location-independent (i.e., with global spatial scope) identifier of a logical BIONETS entity. A name in BIONETS is constituted by a set of pairs < attribute, value >. Names in BIONETS are intentional, i.e., they can be used by services and applications to specify what they are looking for. Names in BIONETS are dynamic, i.e., they may change over time (equivalently: they have a limited scope in time).
- **Address**: location-dependent identifier (i.e., with local spatial scope) of a logical BIONETS entity (in this deliverable: node). Addresses can be used for identification purposes only within a connectivity island. Addresses have a limited scope in time. Addresses can be generated on-the-fly according to a random procedure by each U-Node and by some classes of T-Nodes. Addresses are numeric sequences and can be dynamically bound to names. Addresses may be used for performing one-hop point-to-point communications among BIONETS entities.
- **Identity**: globally unique identifier associated to each U-Node (and to some classes of T-Nodes). An identity has global scope in both time and space. Being location-independent, an identity is technically a name (even if we will use the two terms separately to avoid confusion).
- **Security principal**: any node in the BIONETS network architecture that can be authenticated.
- **Filter:** a set of <attribute,value> pairs, used for defining entities in BIONETS SerWorks. Filters can be used, e.g., to define queries for data of a certain type. The 'attribute' fields are taken according to the convention identified by the 'Naming System' component of the SerWorks networking framework.

Appendix B: System Components

In the following section, the main components of BIONETS SerWorks architecture and required interfaces/communication primitives are described briefly.

1.1 Basic Components

The mandatory components of the SerWork architecture include the Mediators, Interaction Framework and Network Framwork components. A more detailed description of the basic component can be found from [BIONETS_D113/313] and this section only summarises the main mandatory components.

The Interaction Framework needs to support create, read, update and delete operations (CRUD). In addition the push operation can be utilised e.g. for pushing/publishing node-specific information or, by means of a network-specific Event Handling and Information Service Mediators to push one network-specific information within the node island between the different mediators and repositories. The actual implementation of the Interaction Framework is application-dependent. One solution i.e. the Semantic Data Space for BIONETS concept has already introduced in [BIONETS_D113/313] and [BIONETS_D321]. The Interaction Framework can be implemented also as a publish/subscribe or blackboard type of information delivery model. The selected Interaction Framework model affects mainly the service and node repository implementation.

The main components (as depicted also in Fig. 1 and Fig. 2) of the Service Framework are the service specific Mediators in the control plane and the Service Cells (SC) and Service Individuals (SI) in the application plane. Similarly, the Network Framework consists of networking components in the network plane and component-specific mediators in the management plane. The Network Framework components were introduced already in Section 2.2. These components are further divided into mandatory and optional components for the BIONETS SerWorks architecture as described in the following Table 1, where the [M] denotes a mandatory component and [O] optional component.

Type of Node	*Networking Framework Components*
T-Node	opportunistic communications [M] naming system [M] information filtering [O] secure communications [O]
U-Node	opportunistic communications [M] naming system [M] information filtering [M] data dissemination [M] information gathering [M] secure communications [M] P2P cloud [M]
AP	opportunistic communications [M] naming system [M] interoperability with IP networks [M] information gathering [O] secure communications [M]

Table 1. Network Framework components

The main Mediator components and their logical interfaces were described in the [BIONETS__D113/313] from the service framework perspective.

The mandatory mediators in Service Framework are presented in the following Table 2:

Mediator	Description
Request Handling Mediator	The *Request Handling Mediator* maps the request specified by the user to a final set of gains and outputs (see deliverable D3.2.4 [BIONETS_D324] in line with the specification of the service model).
Creation Mediator	The *Creation Mediator* accesses the *Description Repository*, looking for appropriate Service Individuals (SI) matching the requests. In case a suitable SI is found, the *Creation Mediator* requests bindings for each service blueprint from the *Binding Mediator* and finally passes the resulting BSI (bound Service Individual) to the *Execution Mediator*. If there is no matching SI found, a mechanism for dynamic service composition is started.
Service Provisioning Mediator	The *Service Provisioning Mediator* publishes service descriptions at remote nodes (where they are stored in the *Description Repository*) and waits for an according request, given by a notification message. It then provides the service to the given requester.
Binding Mediator / (Service Discovery)	The *Service Binding Mediator* requests service blueprints from the local or remote *Service Provisioning Mediators*. In case multiple nodes host appropriate services, the mediator tries to request the service from all nodes simultaneously. The first node sending an ACK is selected for service provisioning.
Security Mediator	The *Security Mediator* verifies before executing a bound service individual that the service complies with the local security policies. In addition, if, during the execution, a BSI performs an unauthorised action it can abort the execution.
Execution Mediator	The *Execution Mediator* executes a service based on its workflow graph.
Context Mediator	The *Context Mediator* receives input from sensors and/or users about the environmental changes and sends well-defined messages to the *Evolution Mediator* in order to trigger evolutionary procedures.
Evolution Mediator	The *Evolution Mediator* is the main piece for service evolution. Evolutionary procedures can be triggered by a message from the *Context Mediator*, a request from the *Creation Mediator* to evolve an already existing Service Individual or Service Cell or from feedback from the *Evaluation Mediator*.
Evaluation Mediator	The *Evaluation Mediator* evaluates the fitness of a bound Service Cell with regard to its non-functional properties. An appropriate model for

	representing such properties in a lightweight way is introduced in deliverable D3.2.5.
Transformation Mediator	The *Transformation Mediator* has to be considered as a placeholder for multiple Mediators capable of processing an evolutionary step on service level. One example is a Mediator applying genetic operators on service compositions as introduced in [BIONETS_D324].
Service Migration Mediator	The Service Migration Mediator enables the movement of service between different nodes. An extensive description of the related migration procedures can be found in deliverable D3.2.4 [BIONETS_D324]. The migration is inhibited if it conflicts with the local security policies, e.g., if it conflicts with the users privacy policy.

Table 2. Service specific Mediator components in BIONETS SerWorks architecture

The network component specific Mediators are mainly used for controlling the component parameters. Each network component node may have a specific control mediator for implementing the adaptation logics or the component can utilise directly the network state information itself. The mandatory Network Framework side Mediators are presented in the following Table 3.

Mediator	Description
Information Service Mediator	The *Information Service Mediator* collects network information from the physical medium and underlying LLC. This information is further utilised by the adaptation logics directly at the network components or the network component specific control mediators.
Event Handler Mediator	*Event Handler Mediator's* main purpose is to act as a trigger generator and to filter of the different network events for the other mediator use. It provides also the trigger events to service framework from the network components, which can be utilised e.g. by the Service Migration mediator.

Table 3. Network Component Mediators

The mediator communications are divided into local and remote communications. The local communications occur inside the single node between either different mediators (mediator-to-mediator) or between mediators and application services/network modules (mediator-to-service), which are located in corresponding node. The local and remote mediator-to-mediator –communication uses CRUDP (Create, Read, Update, Delete and Push) primitives for communication and the remote communications are performed by default through the interaction framework. The local mediator-to-mediator -communications should use the similar methods as the remote mediator-to-mediator –communication.

1.2 Interfaces

In this section we present the primitives needed to support the SerWorks architecture [BIONETS_D113/313] and its components. The Fig. 4 represents the required communication and control interfaces between the different components in BIONETS SerWorks architecture.

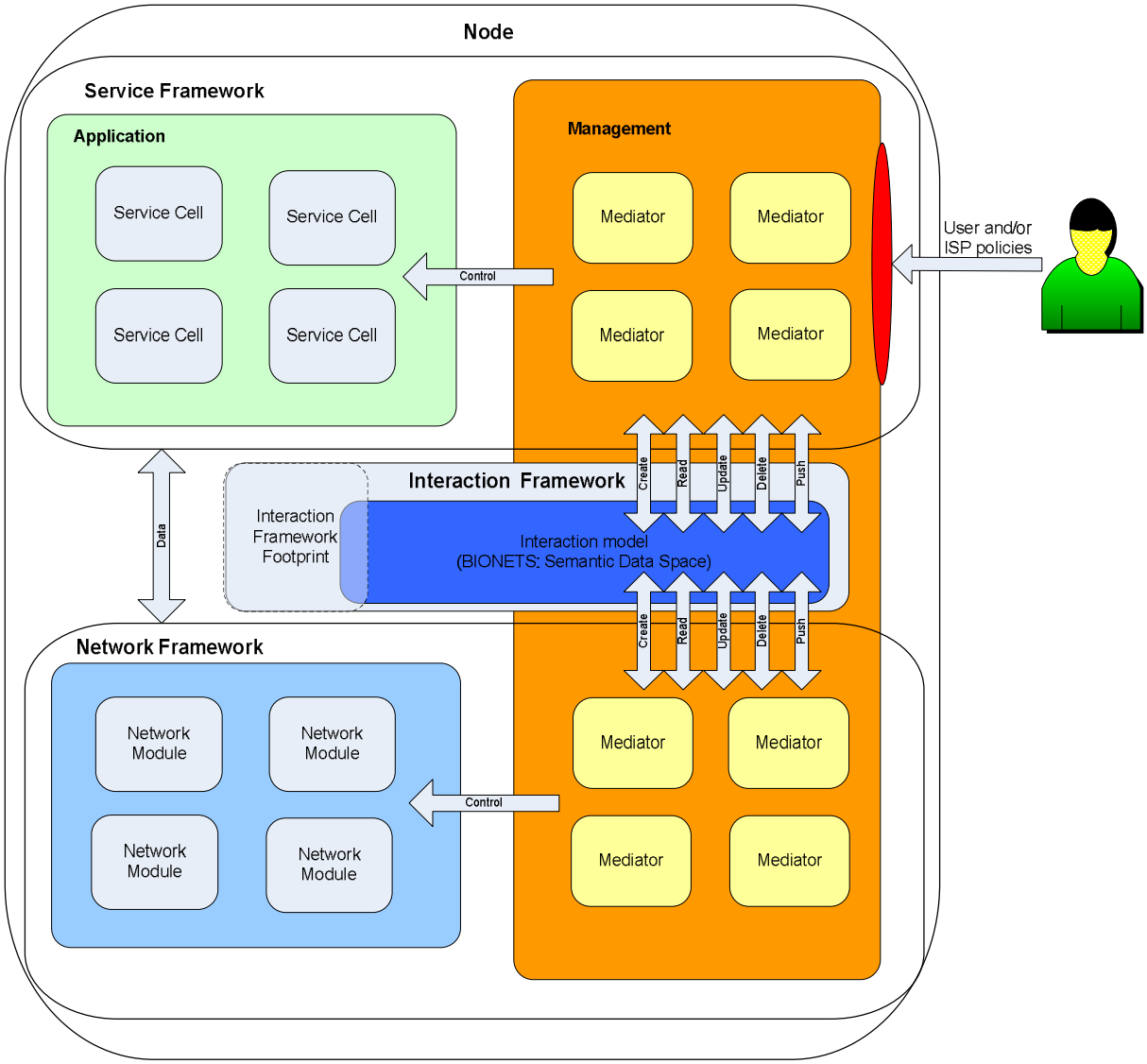

Fig. 4. Control and Management Interfaces

1.2.1 Service Framework

The Service Framework's basic primitives are targeted to support different evolution and adaptation strategies, service life-cycle handling and mobility support. These primitives can be divided in three

groups to support the different capabilities of the Service Framework. The following Table 4 summarises the required interface primitives for Service Framework.

Capability	Primitives	Description
Life-cycle	service.**deploy()** service.**start()** service.**execute()** service.**stop()** service.**deprecate()**	These primitives are used for service life-cycle operations such as executing and deprecating the services.
Evolution	service.**recombine()** service.**mutate()**	These primitives are used for service evolution process for service cross-overs and mutations and are used together with life-cycle primitives to manage the service evolutionary behaviour.
Migration	service.**replicate()** service.**migrate()** service.**resume()**	These primitives are used for migration and service mobility processes. They are also used together with the life-cycle primitives in order to manage execution and deprecation of the services.

Table 4. Service Framework Primitives

1.2.2 Interaction Framework

The Interaction Framework purpose is to provide multiple concurrent interaction models supporting the communication among the distributed services and the realisation of a shared data space. The interface offered by the interaction framework to the service framework consist of two separates parts, one for interaction with mediators and the other for service-to-service interactions. The Interaction Framework interfaces are summarised in the Table 5.

Capability	Primitives	Description
direct service-to-service (or S2S) interactions	service-request() service-registration() invocation-request() invocation-response() search-data() query-registration() query-resolution() event-subscription() event-publishing() event-notification()	These are used by the service cells and/or the service individuals to perform and data searches. It provides primitives for implementing different communication patterns among services (e.g., point-to-point, or point-to-multipoint, epidemic information dissemination, epidemic information retrieval).
mediator interactions	**c**reate()	These are used in order to share and/or exchange information

	read() **u**pdate() **d**elete() push()	relevant for service declaration, selection, etc. (e.g., announce service definitions, publish fragments of ontologies etc.). It is based on CRUD primitives extended with a primitive for propagating notifications in a "push" mode, in order to handle publish/subscribe/notify model.

Table 5. Interaction Framework Primitives

More detailed explanation of these interfaces can be found from [BIONETS_D113/313].

1.2.3 Networking Framework

The lowest layer is the Networking Framework, which provides the basic communication capabilities in the disappearing network context, supporting means for establishing secure communications over opportunistic networks. The set of principal interfaces offered by Networking Framework and its components are listed below in the Table 6. Implementation in the BIONETS Simulator is described in [BIONETS_D133].

Module	Primitives	Description
Opportunistic Communications	send(message,filter) connectTo(filter) doDiscovery() getNeighbors() getActiveQueries() alreadyHasData() registerNeighborEventListener(timePeriod)	Interfaces with the underlying wireless technologies, providing means to transmit/receive packets and discovery nodes in communication range.
Information Filtering	registerFilter(filter) unregisterFilter(filterID),	Handles instantiation of filters to decide the kind of information needed by the running services.
Data Dissemination	getData(filter) searchData(filter) pushData(message) advertise(filter)	Handles opportunistic dissemination of messages among U-Nodes in the network.
Naming System	getAddressByName(filter) getIdentitiesBy-Name(filter) getNameByAddress(filter)	Includes set of attributes to be used to build filters. Handles the mapping between names and addresses of nodes.
P2P Cloud	publish (message)	Provides interfaces for the Interaction

	deprecate(filter) search(filter) send(message,filter) receive(message) retrieve-request(filter) retrieveresult(filter)	Framework and Service Framework to access networking functionalities.
Information Gathering	read(), registerFilteredListener(filter)	Provides means to access data available on T-Nodes, based on the currently instantiated filters.
Interoperability with IP	search(filter) notify(message)	Provides means for BIONETS islands to exploit opportunistically the presence of legacy IP networks. An example implementation is the R-P2P described in [DePellegrini08]

Table 6. Network Framework Primitives