



BIONETS

WP 1.1 – REQUIREMENTS AND ARCHITECTURAL PRINCIPLES

D1.1.2 Architecture, Scenarios and Requirements Refinements

Reference:	BIONETS/CN/wp1.1/1.0
Category:	Deliverable
Editor:	Daniele Miorandi (CN)
Authors:	Markku Tahkokorpi (NOKIA), Juhani Latvakoski (VTT), Eleonora Borgia (CN-CNR), Antonis Panagakis (NKUA), Vilmos Simon (BUTE), Francesco De Pellegrini (CN), Tomi Hautakoski (VTT), Daniel Schreckling (HITECH), Daniele Miorandi (CN)
Verification:	Ioannis Stavrakakis (NKUA), Iacopo Carreras (CN), Giannis Koukoutsidis (NKUA)
Date:	August 10, 2007
Status:	Final
Availability:	Public

SUMMARY

This deliverable presents an enhanced version of the network requirements and high-level architecture detailed in D1.1.1. In particular, it provides a more formal description of the various concepts involved, presenting a first architecture for the three kinds of nodes encompassed in BIONETS systems. Also, architectural considerations on the interworking with infrastructured IP networks are presented. Particular attention has been devoted to the inclusion of security features as a basic building block of the system's architecture.

Contents

1	Introduction	7
2	Terminology	7
2.1	Abbreviations	7
2.2	Definitions	7
3	System Features and Scenarios Analysis	9
3.1	Introduction	9
3.2	Required Features	9
4	High-Level Network Architecture	11
4.1	Nodes in BIONETS Networks	12
4.2	Relations in BIONETS Networks	13
4.3	Disappearing Network Support for BIONETS Services	15
5	Communications & Networking Framework	16
5.1	Messages	16
5.2	Naming and Addressing	16
5.2.1	General concepts	16
5.2.2	Identification of T-nodes	17
5.2.3	Identification of U-Nodes	17
5.2.4	Examples	17
5.2.5	Tagging	18
5.3	Data/Information Dissemination	19
5.3.1	Data Collection	20
5.3.2	Relaying and Filtering	20
5.4	Information Filtering and Data Management	22
5.5	Interworking with Legacy Networks	23
6	Integration of Security Framework	25
6.1	Security Requirements	25
6.1.1	T-Nodes	25
6.1.2	U-nodes	27
6.1.3	Access Points	28
6.2	High Level Security Architecture	30
6.2.1	T-nodes	30
6.2.2	U-nodes	31
6.2.3	Access Points	32
6.3	Conclusion	33

7 Outlook and Next Steps	33
References	34
A Interworking with Legacy DTNs	37

DOCUMENT HISTORY

Version History

Version	Status	Date	Author(s)
0.1	Draft	9 May 2007	Daniele Miorandi, CN
0.2	Draft	5 June 2007	Daniele Miorandi, CN
0.3	Draft	11 June 2007	Daniele Miorandi, CN
0.4	Draft	19 June 2007	Daniele Miorandi, CN
0.5	Draft	27 July 2007	Daniele Miorandi, CN
1	Final	7 August 2007	Daniele Miorandi, CN

Summary of Changes

Version	Section(s)	Synopsis of Change
0.1	ToC and skeleton	Inclusion
0.2	All	Inclusion
0.3	All	Revision
0.4	3,6	Inclusion
0.5	All	Incorporation of reviewers' comments
1	All	Revision

1 Introduction

One of the aims of the BIONETS project is to devise new scalable models for autonomic communication systems, based on localized opportunistic information exchanges among heterogeneous devices. This document presents a first architectural perspective on the BIONETS “disappearing network” concept [1, 2], defining the system’s building blocks, their mutual relationships and their role. BIONETS targets pervasive computing and communication environments [3], characterized by the presence of myriads of embedded devices, used to gather environmental information leveraged by personal devices to run context-aware user-centric services. From the networking point of view, such scenario presents two problematic features: scale (in terms of number of devices) and heterogeneity (in terms of different features supported by different nodes). BIONETS tackles such issues by providing a novel network architecture, based on localized exchanges of information upon opportunistic contacts, and by dividing system devices into two broad categories, with clearly different roles in the system and technical features supported. This deliverable represents an attempt to refine the concepts introduced in [4], while at the same time providing a first formalization of the BIONETS disappearing network architecture.

The remainder of the deliverable is organized as follows. Sec. 2 introduces the terminology used throughout the document. Sec. 3 defines some system-level requirements, based on a set of application scenarios developed within the project. Sec. 4 introduces the BIONETS high-level network architecture, detailing the different types of nodes, their relations and the support offered to evolving services. The communications and networking framework is presented in Sec. 5. Sec. 6 analyzes the BIONETS architecture from the security viewpoint. Sec. 7 concludes the paper outlining the most promising directions to be pursued in order to refine the presented network architecture.

2 Terminology

2.1 Abbreviations

- AP: access point;
- U-Node: user-node;
- T-Node: tiny-node;
- DTN: delay-tolerant network.

2.2 Definitions

- **User-Nodes (U-Nodes):** complex powerful electronic devices with computing/communication/storage capabilities, carried around by users (hence inherently mobile) and hosting services. U-nodes interact with the environment through T-nodes, from which they gather information augmenting context-aware services. Environmental data or service-specific code (in order to enable service evolution) can be exchanged among U-nodes by proximity communications.

PDAs, laptops, smartphones, GPS car navigation systems, car-to-car platforms represent examples of U-Nodes.

- **Tiny-Nodes (T-Nodes):** simple, inexpensive embedded devices with sensing/identifying capabilities. They act as an interface with the environment and are needed to provide context-awareness to BIONETS services. They do not communicate among themselves but are just read by U-nodes in proximity. They present minimal requirements in terms of processing, storage, and communications.
- **Access Point (AP):** device presenting an interface for communicating with U-Nodes and an interface supporting the TCP/IP protocol suite. In BIONETS, they are used to provide internetworking capabilities between BIONETS islands and infrastructured IP networks.
- **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 localized 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 loses 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 (in this deliverable: node). A name in BIONETS is constituted by a set of pairs $\langle attribute, value \rangle$. 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 two-hop neighborhood. Addresses in BIONETS are *unique* within a two-hop neighborhood. 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).

- **Proxy server:** application program which services the requests of its clients by making requests to other servers.
- **Security principal:** any node in the BIONETS network architecture that can be authenticated.

3 System Features and Scenarios Analysis

3.1 Introduction

The motivation for the BIONETS paradigm comes from novel usage scenarios enabled by pervasive computing/communication environments. Among the use cases defined within the project consortium [4], four have been selected as representative ones [5]:

1. BIONETS aided guidance system — where a BIONETS type of system is used for context-enhanced navigation,
2. Wellness — which is an application dealing with person-related sensor data, where smart computing devices (U-Nodes) and sensor node (T-nodes) form a personal area network (PAN) around an individual and may potentially cooperate with co-located PANs,
3. Virtual Guide in a pervasive computing environment focusing on “traditional” pervasive computing as a starting point, where a devices around an individual form a PAN interacting with each other as well as with the rest of the environment and
4. BIONETS for handicapped people and home nursing — which aims to integrate healthcare related applications utilizing the BIONETS architecture.

These four scenarios do not have much overlap (except for some common points between the wellness and the virtual guide ones) and thus are well representative for the whole set of proposed scenarios. Based on these scenarios the service requirements have been analysed in [5].

The initial overall analysis including networking, architecture and business issues from the thirteen original scenarios was also carried out during the first six months of the project and has been documented in [4].

3.2 Required Features

The proposed use cases led to the identification of a significant number of distinguished features, which represented the basis for defining the BIONETS system architecture. One particularity which emerged was the wide diversity in requirements stemming from the various application scenarios. This was particularly evident with respect to the location and mobility of T- and U-Nodes. Indeed, depending on the scenario, the T-nodes may be stationary located indoor or outdoor, they may move together with U-nodes or they can move with some other mobile object depending on the application. Due to practical reasons there was a need to limit the amount of options when progressing with the project.

The following BIONETS specific features could be prominently observed in the proposed scenarios and need to be supported by the envisioned "Disappearing Network" architecture.

- **Devices Heterogeneity.**

The BIONETS system is characterized by a large heterogeneity in the kind of devices taking part in the system, presenting very different computational and communication capabilities. The management of the heterogeneity shall be supported by the BIONETS architecture, e.g., by relying on local interactions between nodes.

- **Scalability of the system.**

While the scalability issue is not a prominent one within the scenario descriptions, we may expect the amount of deployed T-nodes to be extremely large in the future, as well as the number of different services running in the also numerous U-nodes. The use of local interactions only for communications and of the information filtering mechanisms are envisioned to tackle the scalability problem.

- **Data exchange between nodes using proximity wireless technologies.**

This is one of the core features of BIONETS, i.e. no end-to-end networking between the nodes but instead direct localized information exchange between the nodes. Although many scenarios utilize Internet as a source of relevant information, the T-Node – U-node and U-node – U-node communication is local in all these scenarios. This leads to the so called "disappearing network paradigm" where information is disseminated without requiring necessarily end-to-end network connectivity. This is also a key feature for improving the scalability of the system when it is compared with traditional architectures.

- ***Store-and-forward-type of data dissemination utilizing mobility of the U-nodes.***

Several scenarios utilize the fact that U-nodes are mobile and can carry around data they have read from T-nodes or received from other U-nodes to be utilized elsewhere or spread further. Contacts among U-Nodes are opportunistically exploited to achieve system-wide diffusion of messages. This is another aspect related to the previous item and is similar to DTN (Delay Tolerant Networking) [6] technologies (but without the end-to-end addressing aspect). Without this feature many of the scenarios would not work properly as mobility of the data and opportunistic communication via U-nodes is one of the assumptions behind the applications.

- **Location-specific information distribution.**

In some scenarios, information disseminated by T-Nodes has only relevance to nodes near the source in terms of time and space domain. The sensor information could then be filtered to avoid congestion and resource overuse. On messages which are produced by the U-nodes a filtering solution which works dynamically and takes into consideration the requirements of message dissemination range set by the services is needed. As an example, T-nodes measuring human density inside could be installed into a museum located somewhere in a big city. Now, a service running on U-nodes could control the network range of their advertisements of the museum to the tourists according to the number of visitors and their position inside at the

moment. In this way, more tourists can be invited into the museum, being routed in such a way to avoid “congestion” in front of some artworks and providing users with a better experience while maximizing the revenue of the museum.

- **Self-organization of the nodes.**

The complexity and dynamics in some of the scenarios seems to require a capability to make the nodes able to react to a wide range of different situations as well as to minimize human intervention, hence raising usability of the nodes in general and of the services running on them. These requirements can be thought to be solved by giving the nodes the ability to automatically organize themselves in a way which ensures that a node will react sanely to all kinds of inputs coming from both the users of nodes and from the network. For the node to be able to cope with the rapidly changing environment and make decisions on how to proceed, it first needs to have a way to realize and identify the current situation. This skill can be described as *situation-awareness*. It holds the ability to recognize changes in the resources a node can have, e.g. amount of battery energy or usable network interfaces, as well as the internal operating state of the node. It also includes the capability of discovering and negotiating with the neighboring nodes to reach a distributed understanding on how the nodes should behave in order to fulfill the needs of the services running on top of them. Also to make better decisions about the situations, a node should have possibility to share its own situation and enquire other nodes’ situations when necessary.

If a node has been able to recognize the current situation, it needs to be able to react in a rational way, ensuring that the node will be in a living state at all times and does not impede operations of the network. This function can be called as *adaptation to contextual situation*. The adaptation process can be a complex one as there are several factors which influence how the node should react. The hardware resources of a node can set a hard limit on how a node can and could behave to cope with a change in its situation. The user of the node can also instruct the node to adapt in a way which is more preferable to him/her. This could mean the user could use different “profiles”, defining rules on how the node should react to different situations, e.g. to save battery energy. In addition, the services running on the node can have an impact on how the node works to fulfil the goals of the service. All these abilities mentioned earlier need to work autonomously in a single node as no central authority or similar is present in the BIONETS network. Still, when possible, the nodes should engage mutual communication in their pursue to maintain the network in a working, living state.

4 High-Level Network Architecture

In this section, we present the basic architectural concepts underpinning the BIONETS disappearing network paradigm. We first introduce the three main entities in the BIONETS networks and their mutual relations, along with a possible node architecture. Then, we describe their logical role in the system, highlighting the support offered to BIONETS services.

4.1 Nodes in BIONETS Networks

BIONETS systems are built around three types of electronic devices with computing/communication capabilities: T-Nodes, U-Nodes and APs.

- **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 utilized by the U-Nodes in order to enhance the services by providing context-aware features. 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. T-Nodes are divided in four classes, depending on the supported features [7].

The logical architecture of a T-Node is described in Fig. 1. Dashed boxes indicate optional components, which may be present or not depending upon the class the T-Node belongs to (see also D4.2 for an in-depth discussion of the security features required and supported). Three interfaces are present. The first one is used to exchange messages with U-Nodes. The second one (optional), is used for updating the T-Nodes services and operations (e.g., updating T-Nodes security credentials). The third one is used for connecting with the sensing devices hosted by the node. A data processing unit is present, which performs basic operations on sensed data, like, e.g., sampling and averaging. The (optional) data security unit is responsible for performing all tasks related to ensuring the security of the data to be transmitted (e.g., encryption, authentication, identification). The (optional) communications security unit is responsible for ensuring that messages exchanged between U- and T-nodes are of integrity, are authentic, and are confidential. The basic communication unit is responsible for performing basic tasks such as: matching queries (in the case a pull mechanism is supported), adding $\langle attribute, value \rangle$ descriptions etc.

- **U-Nodes** are complex, powerful electronic devices with computing and communication capabilities. No stringent limitations on requirements are assumed for U-Nodes. 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).

The basic architecture of a U-Node is represented in Fig. 2. Compared to the T-Nodes architectures, three additional modules are foreseen. The service execution and service composition units are meant to, respectively, perform the tasks required to execute a composite service (also referred to as “service individual” [8]) and compose new ones starting from atomic services (or “service cells” [8]). The security enforcement unit is concerned with ensuring (i) integrity of data (ii) enforcing security policies (iii) ensuring security of BIONETS services (iv) access control and related secure communications problems. Two logical commu-

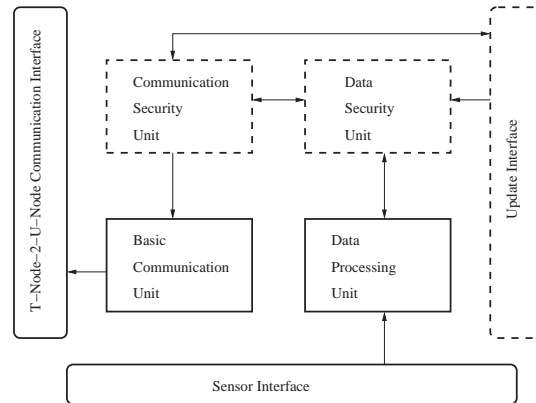


Figure 1: T-Nodes architecture. The dashed blocks indicate optional components, whose presence depends on the particular class the T-Node belongs to.

nication interfaces (one to T-Nodes and one to U-Nodes) are present. They might correspond to the same physical network interface (in the case a single radio technology is used to communicate with both T- and U-Nodes), or to two different ones.

- **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. The architecture of an AP is schematically represented in Fig. 3. A BIONETS AP presents three mandatory interfaces and an optional one. A mandatory one is used for updates of security components. The other two mandatory interfaces are communication ones. The first one is used for communicating with BIONETS U-Nodes. The other one provides connectivity to the infrastructure and supports the TCP/IP protocol suite. An optional interface may enable communications between the AP and T-Nodes. The core of the AP is represented by the proxy server unit, which handles the mapping and ensures consistency of the communications between the IP infrastructure and the BIONETS island the AP is connected to.

4.2 Relations in BIONETS Networks

BIONETS networks will be based on a two-tier network architecture [4]. The lower tier will be constituted by T-Nodes, which will be used for gathering information from the local environment. T-Nodes do not communicate with each other, but only with the U-Nodes nearby. U-Node-to-T-Node communications, as well as U-Node-to-U-Node ones, are performed by means of proximity wireless communication technologies. U-Nodes may possess a single physical wireless interface for communicating with both T- and U-Nodes, or two different ones. APs will communicate with U-Nodes nearby. The overall network architecture is depicted in Fig. 4. In such figure, the lowest plane represents the spatial position of entities in the system, whereas the other two planes reflect the hierarchical (two-tier) architecture.

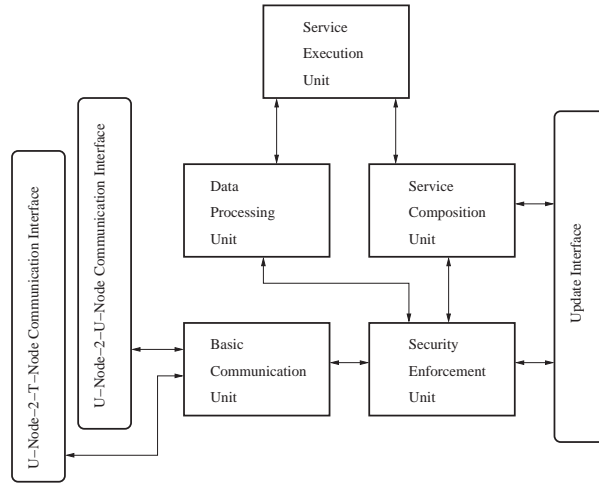


Figure 2: U-Nodes architecture.

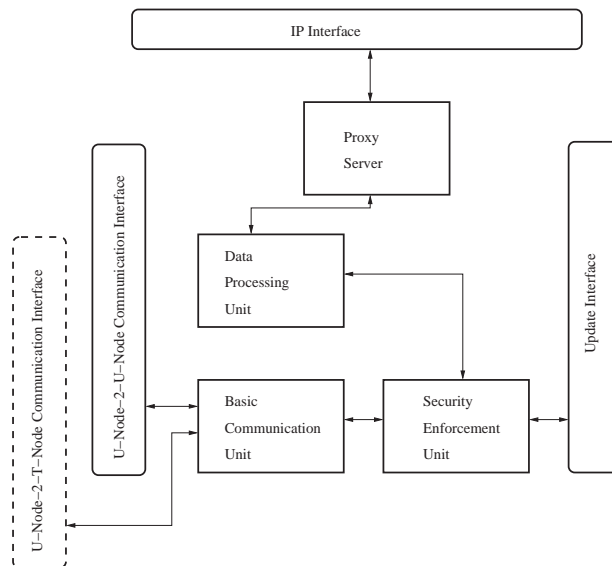


Figure 3: APs architecture.

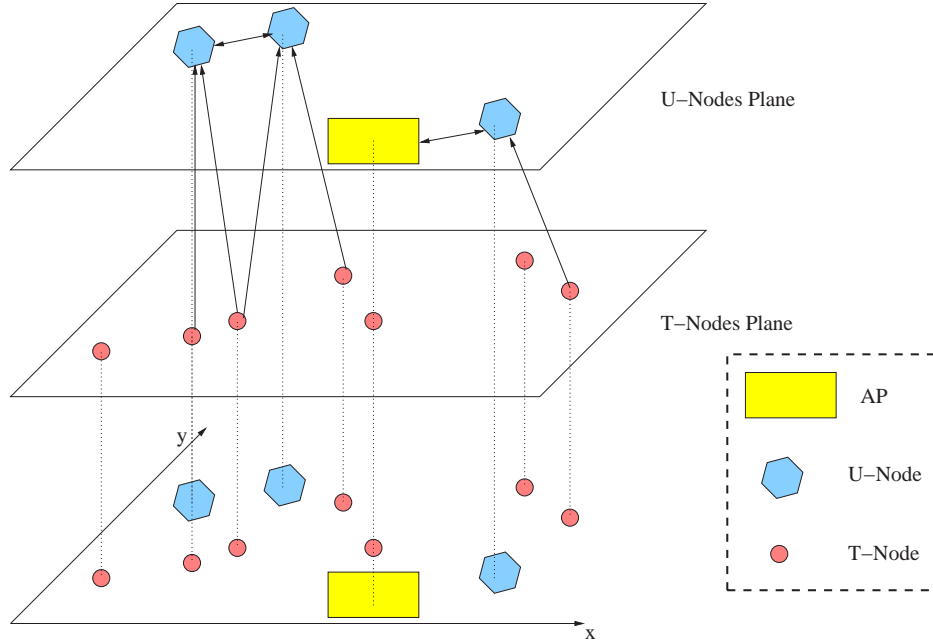


Figure 4: The BIONETS two-tier basic network architecture. The bottom diagram represents the spatial position of the devices. The other two diagrams represent the T-Nodes and U-Nodes plane.

4.3 Disappearing Network Support for BIONETS Services

The BIONETS system is designed around a service-oriented network architecture, which aims at offering a native support to the BIONETS services, as introduced in [5]. In this view, the aim is to provide distributed self-evolving services with an effective support, given the constraints inherently present in BIONETS application scenarios (heterogeneity, resources constraints, mobility and frequent disconnections etc.).

The BIONETS network architecture aims therefore at representing a lightweight, flexible platform for supporting autonomic services. Given the final aim of integrating network and services in the *SerWorks* vision [5], the network architecture has been developed in parallel with the service architecture, to ensure their alignment, in view of a merging of the two.

An example is the proposed intentional naming system (see §5.2), which could be easily extended to services in such a way to natively support resources discovery (recognized as a key and challenging task in the distributed and highly dynamic BIONETS environments). Another example is the proposed message diffusion strategy (based on epidemics-like spreading of data), which could be used to achieve, at the service level, unicast, multicast, anycast and broadcast communications. On the other hand, the BIONETS service architecture, based on loosely coupled service components (called service cells in the BIONETS terminology) matches the transient nature of BIONETS connectivity islands. In view of the long-term *SerWorks* integration, we envision that all networking functionalities (as defined in both the present document as well as in [7]) shall be integrated, as non-functional components, into the service architecture. An important role in such integration process will be played by the interaction framework [9], which interfaces, in the current architecture, the services with the underlying networking support.

5 Communications & Networking Framework

5.1 Messages

Communications in BIONETS are based on the notion 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 $\langle \textit{attribute}, \textit{value} \rangle$ 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.

5.2 Naming and Addressing

5.2.1 General concepts

BIONETS relies on *names* for identifying communicating devices. Names in BIONETS are intentional [10] and are defined as a set of pairs $\langle \textit{attribute}, \textit{value} \rangle$. Names are location-independent identifiers, i.e., they have global spatial scope and do not change as the node moves in the system. Names have limited temporal scope, i.e., they might change over time. All nodes in BIONETS have a name. Names are not unique. Names can be used for taking decisions concerning information/data forwarding. Their use, which complies with similar approaches in data-centric wireless sensor networks [11], enables the construction of a content-based architecture (as opposed to conventional IP address-based architectures). A special attribute field value, *tag*, can be used to enable keyword-based queries support.

Nodes which are subject to trust and reputation systems (i.e., all U-Nodes and some classes of T-Nodes, see D4.2) possess a *unique static* identifier called *identity*. The identity of a node has global scope in space and time. It represents a fingerprint of the node, and it is expressed as a numerical value. We assume that node identities are hardwired in the nodes by the manufacturer. Identities are not used for taking forwarding decisions, and are not exposed to the network framework. Identities are exposed to trust and reputation services only.

BIONETS encompasses also the use of identifiers with local scope in both space and time, that are termed *addresses*. An address is constituted by a numerical value, which can be associated to U-Nodes and to some classes of T-Nodes. Addresses are unique within a two-hop neighborhood. Addresses are generated locally according to a random procedure, coupled with mechanisms for resolving collisions [12, 13, 14]. The use of addresses is optional and is meant to provide bandwidth savings in one-hop communications by using short numeric identifiers instead of long, expressive, names.

5.2.2 Identification of T-nodes

T-Nodes are identified through their name. We expect that the name of a T-Node will describe, in an expressive way, the type of data it measures and the security mechanisms supported. As an example, a T-Node measuring temperature may present name $\langle \text{dataType}, \text{temperature} \rangle$. If it supports encryption, its name may be $\{\langle \text{dataType}, \text{temperature} \rangle, \langle \text{encryption}, \text{on} \rangle\}$. A T-Node may also present a $\langle \text{location}, \cdot \rangle$ field indicating its physical location (for static devices only).

In the case a push mechanism is used for T-Nodes-to-U-Nodes communications, T-Nodes shall publish their data in a message containing the T-Node name. Such name can be used by U-Nodes within communication range to filter the message and decide whether it is of interest to the running services.

In the case a pull mechanism for T-Nodes-to-U-Nodes communications is used, U-Nodes will include a description of the data they are looking for within a query message. Upon reception of such message, the Basic Communications Unit of the T-Node will perform a match with the node's name. In case of a positive match, a reply message will be generated and transmitted.

5.2.3 Identification of U-Nodes

From the networking point of view, U-Nodes are identified through their name. In case the U-Node is the personal device of a given person, its name can be related to the owner of the device. As an example, the U-Node of John Doe, from the CS department of UCSB, could have name $\{\langle \text{name}, \text{JohnDoe} \rangle, \langle \text{company}, \text{UCSB} \rangle, \langle \text{department}, \text{CS} \rangle\}$. In case U-Nodes are personal devices, names could be used as a user's profile. This is similar to what proposed in Unmanaged Internet Architecture (UIA) [15]. Some U-Nodes may expose names which include the description of services they offer.

In order to save bandwidth, U-Nodes may use, for one-hop communications with other entities already discovered, shorter identifiers with local scope in time and space. Such identifiers are called addresses. U-Nodes support a procedure for generating randomly numerical identifiers, and mechanism for resolution of collisions (i.e., capable of ensuring the uniqueness of addresses within a two-hop neighborhood).

5.2.4 Examples

We will consider two cases. The first one concerns a U-Node querying nearby T-Nodes (pull mechanism) for temperature data. The procedure is depicted in Fig. 5. In (a), the initial situation is represented. There is a U-Node with name $\langle \text{name}, \text{JohnDoe} \rangle$ and two T-Nodes, one with name $\langle \text{dataType}, \text{temperature} \rangle$ and one with name $\langle \text{dataType}, \text{pressure} \rangle$. In (b), the U-Node issues a query requesting temperature data, $\text{query}(\langle \text{dataType}, \text{temperature} \rangle)$, which reaches both T-Nodes. In (c), the T-Nodes perform a match with their name. In (d), the T-Node whose name matches the query content replies broadcasting a message containing its sensed data, and the corresponding metadata $\langle \text{dataType}, \text{temperature} \rangle$. The metadata will be used at the U-Node interface for filtering incoming packets and passing it to the appropriate service.

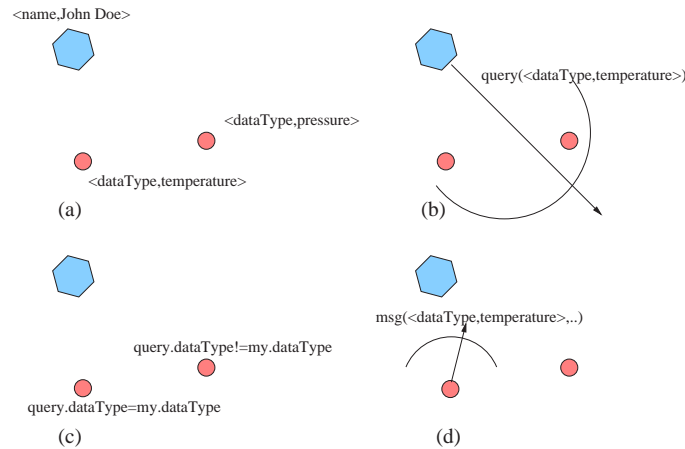


Figure 5: Example of a U-Node querying for temperature data and obtaining a query from one nearby T-Node.

In the second case, we consider two U-Nodes getting within mutual communication range, and performing an exchange of some data to be disseminated. This includes the generation of addresses. The operations are depicted in Fig. 6. In (a) and (b) two nodes, with name, respectively, $\langle name, JohnDoe \rangle$ and $\langle name, LizSmith \rangle$ get within mutual communication range and exchange hello messages. In (c) and (d), the first node generates an address $xyyyzz$ and sends a message to $\langle name, LizSmith \rangle$ with its name and address. (We leave out for the moment the procedure for address contention resolution.) In (e), the second node binds $\langle address, xyyyzz \rangle$ to $\langle name, JohnDoe \rangle$ and generates an address, let's say $qqwwee$. In (f), it then sends a message to $\langle address, xyyyzz \rangle$ with its new address, which is then bound to its name. The two nodes can then use only address fields to exchange messages.

5.2.5 Tagging

A tag is a relevant keyword or term associated with or assigned to a piece of information (like picture, article, or video clip), thus describing the item and enabling keyword-based classification of information it is applied to. Tags can be used in BIONETS systems for text-based identification of objects (not limited to nodes, but also to services and content).

Tagging provides some degrees of freedom and flexibility for users to provide description of object which do not adhere strictly to the standard $\langle attribute, value \rangle$ pairs. Tags can be exploited to perform keyword-based search, which shall be “less structured” than with $\langle attribute, value \rangle$ pairs. In the naming framework presented in Sec. 5.2, tags can be understood as description of unknown type. We may therefore introduce a simple rule for including tagging in the naming system. Tags will be formally attached to an object as a pair $\langle tag, xyyyzz \rangle$, where $xyyyzz$ is the tag associated to the object. Tags can therefore be transparently treated as part of names and/or metadata associated to messages. The support of tagging is expected to have an impact in facilitating the arising of BIONETS communities of users, enabling people to use tags in much the same way they will get used to as in Web2.0.

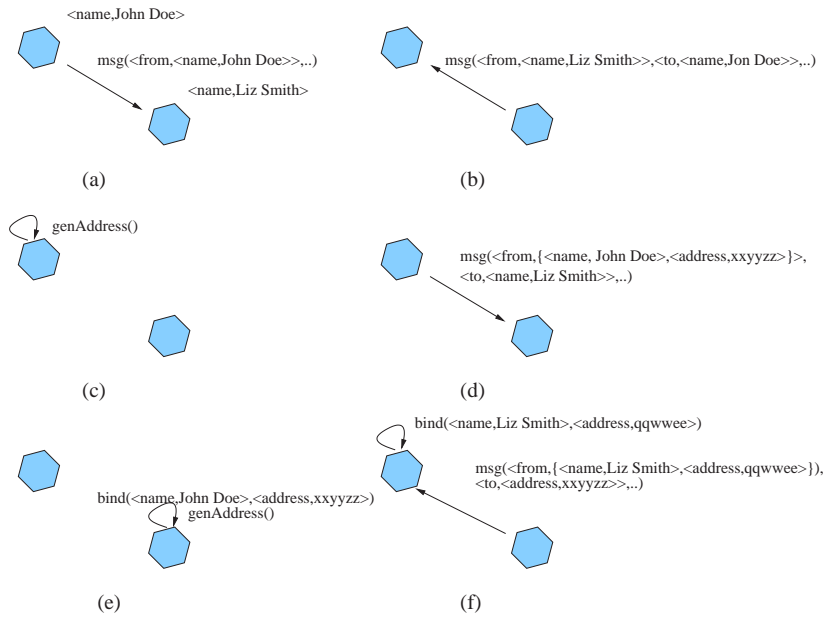


Figure 6: Example of two U-Nodes exchanging data and using addresses for identification.

5.3 Data/Information Dissemination

In this section, we will review the fundamental requirements imposed by the scenario analysis and features outlined in §3 on the data and information dissemination mechanisms, together with the corresponding design choices made within the BIONETS framework.

Within the environment of Bionets, where connectivity is not guaranteed, data is lavishly generated at the T-nodes while it becomes available at the U-nodes after the latter collect it by the T-nodes they pass by. This data may concern a number of different services, and thus U-node groups, as well as refer to a specific time- and space-bounded area. These inherent limitations make the communication paradigm expected for Bionets differ from those in traditional wireless (and connected) networks. Moreover, in the typical case, data in Bionets concern some characteristics of the environment and are regularly collected by the T-nodes. In such an environment, traditional communication services, as for example reliable and in order delivery of data, become meaningless or difficult to be accomplished.

In BIONETS, no single point of operation should be expected (not a single set of parameters should be the most preferable one to achieve the best performance in the entire network and for the network's lifetime). On the contrary, the desirable performance goals vary over time and space, as the network and its supported services are dynamically deployed. The data collection/dissemination mechanisms should be adaptable to the changes that may occur in the network so that they cater to the newly introduced dissemination or data requirements.

Moreover, along with data collection/dissemination, there might be other processes running in parallel; some U-nodes may be informed of which nodes (or the area at which they may) acquire the information they are interested in (e.g. through data advertisement). Data dissemination should be capable of responding to requests that are based on the knowledge of the potential recipients

of the information required and avoid wasting the limited network resources (especially when the number of U-nodes interested in specific pieces of information is limited). To allow for the adaptability of the data dissemination mechanism, data dissemination should be triggered by the output of the other processes to route data accordingly in order to reach the interested parties efficiently without flooding a large part of the network with unwanted data.

5.3.1 Data Collection

In contrast to the traditional networks, data in BIONETS are not generated at the part of the U-nodes that intend to communicate them with a specific other U-node. Here, data are generated at every spot a T-node resides and they may concern a specific service and/or a group of U-nodes that are subscribed/interested in this type of information; such data may also have different temporal and spatial characteristics (meaning that they may be valid within specific time or space limits that might be different for different services). The acquisition of the T-nodes' data by the U-nodes, referred to as data collection, may be achieved through two modes of communication targeting at minimizing the unnecessary waste of the limited T-nodes' resources: the push or pull mode. Under the push mode, T-nodes transmit their content every time there is an indication of a U-node's presence within their communication range. Under the pull mode, U-nodes choose the T-node they are going to interact with in order to obtain the information they are interested in. In either case, data collection follows a specific temporal and spatial T-node sampling pattern, defined by the interaction between the T- and U-nodes that may depend on the density of the T-nodes in the area, the density and mobility pattern of the U-nodes as well as the willingness of the latter to participate in data collection. This sampling pattern plays a role in determining the data availability at the U-nodes layer playing a crucial role in the effectiveness of and being inextricably linked with the applied data dissemination mechanism.

5.3.2 Relaying and Filtering

The performance of data dissemination may be measured by the "quality" of the information that becomes available to the interested users (e.g. availability and freshness that is affected by the data propagation speed) and the overhead spent (e.g. wireless bandwidth consumption or number of nodes involved, no matter if they are interested in it or not). The topology characteristics play an important role in the performance of the dissemination mechanism; for instance, the density of the U- or T-nodes is inextricably linked with the delay of accessing the information and delivering it to the interested group of nodes. At the same time, the speed of the U-nodes may affect the information propagation speed among the U-nodes interacting with each other. More specifically, mobility is the main means for achieving both data collection and data dissemination in BIONETS and demands that the store-and-forward policy of connected wireless networking paradigms be replaced by the store-carry-and-forward one ([16], [17], [18], [19]). While mobility constitutes one of the main detrimental factors regarding the performance of data dissemination in most traditional wireless networks, it is the necessary component that may guarantee data delivery in highly partitioned networking environments. The dissemination mechanism in BIONETS is characterized by

the opportunistic relaying of data between U-nodes that move around the area collecting opportunistically the data that is generated at the T-nodes' layer; the collected data needs to be stored, carried and forwarded upon U-nodes' encounters. Both data collection and data dissemination are opportunistic (there is no guaranteed connectivity between the U-nodes or between the U-node and the data it seeks for in the T-nodes layer), mobility-assisted and interrupted processes. This is an inherent characteristic of BIONETS implying that even in the ideal case where no bandwidth or storage constraints existed, the setting of this new environment would unavoidably impose a number of restrictions and limitations on the communication services that could be supported (mainly with respect to the availability of data and the speed at which it is propagated).

Data dissemination includes a number of design choices that may be categorized as relaying choices and filtering (processing) choices; the former correspond to a message-centric approach (and determine which nodes are supposed to act as relays as well as the number of copies that are spread in the network), while the latter to an information-centric approach (and determine the amount of information to be relayed leading to time- and space-configured data). At this point, it should be mentioned that this distinction might not always be clear; for example, a timestamp that determines when to drop a piece of data might be determined on the basis of the expected number of copies present in the network (message-centric approach) or the decrease in the importance of the specific piece of data (information-centric approach).

Message-centric approach: The number of data copies that are allowed to be spread in the network as well as the number of disseminators/forwarders that should be employed in the dissemination procedure (along with their allowed forwarding actions) should be considered as the main tuneable parameters of the applied dissemination mechanism in order to achieve the desired performance goals. The fact that, in the general case, there are multiple expected U-nodes interested in the data that is going to be disseminated and data may have time (and space) limitations leads to the conclusion that single-copy dissemination policies are not expected to achieve satisfying performance. What is more preferable within the BIONETS framework is a flooding-based mechanism where all (or part) of the U-nodes assume the role of a carrier and a forwarder of data they receive by their interaction with either other U-nodes or the T-nodes. In order to control the induced overhead, only a specific number of data copies (say K) should be propagated among the U-nodes in the network and each disseminator should have the right to make one or multiple forwarding actions until this number of data copies is reached ([20], [21]).

Information-centric approach: Since the bulk of data that may be available at the U-nodes' layer may be extravagant, a process that complements data dissemination and is considered to be inextricably linked with it is data filtering. The filtering process aims at reducing the size of data that is needed to be stored and carried by the U-nodes. The various techniques that could be implemented for this purpose should take into account the characteristics of the data, such as the length of the useful lifetime of the data (i.e., the period over which the data is expected to be useful to a potential recipient) or the area out of which data dissemination would be meaningless, in order to provide more efficient data storing and forwarding. Based on the above characteristics, the data copies could be stamped with a TTL (Time-To-Live) field indicating both/either the (useful) lifetime of the packet (a time counter) and/or the dissemination area (a hop counter) [22].

One key requirement for engineering BIONETS systems is the robustness to nodes' behaviour. In centralized controlled networks, indeed, all the users have a common goal and the presence of a centralized entity controlling all the operations guarantees that they operate according to pre-established rules for the benefit of the overall system. Therefore, cooperation among nodes may be assumed. Based on the collected information, the authority makes its own decisions aiming at reaching an equilibrium to maximize the overall system performance. On the contrary, the nature of BIONETS – large-scale system with distributed network functions and services operations – may motivate a possible tendency of users to misbehave; users act independently and, as a consequence, decisions are mainly driven by their personal interests. Cooperation here cannot directly be assumed, neither at network nor at application level. Still, even though BIONETS may be considered as an open non-cooperative system, equilibrium may be reached. To alleviate the effects of non-cooperation appropriate mechanisms able to tolerate the presence of uncooperative entities and cope with them effectively may be designed and implemented [23]. For example, enhancements can be obtained both at the design phase when the parameters are tuned and/or at the run time phase (when users run it) making the network/service mechanisms responsive and adaptive to the changes on the degree of nodes' cooperation.

Focusing on the network level only, the performance of the applied dissemination mechanism is linked with the expected U-nodes' behavior. This behavior may refer to the intentional or non-intentional decision of a U-node to participate or not in the dissemination process. The intentional non-participation in the dissemination process refers to the cases of misbehavior, selfishness or non-cooperation; the non-intentional, non-participation in the dissemination process refers to the cases of buffer limitations or energy constraints. The design of the data dissemination mechanism should encounter the possibility of a non-cooperative environment or the technology limitations of the U-nodes in order to fine-tune the parameters of data dissemination accordingly. The lack of the U-nodes' cooperation may delay the spreading of data within the network, since some data copies may be discarded and not further relayed. To alleviate such inefficiencies, the increase on the number of data copies to be propagated as well as the uniform distribution of the relaying activities among the set of the U-nodes present in the network would improve the performance of data dissemination (potentially aided by an estimation of the existing degree of cooperation in the network)[24].

5.4 Information Filtering and Data Management

One of the key issues for enabling BIONETS-like systems is to devise efficient mechanisms for coping with the scalability issues related to the traffic carrying contextual information. In BIONETS, such information is gathered by T-Nodes and passed to U-Nodes in proximity, either in a proactive fashion (“push” mechanisms) or in a reactive one (“pull” mechanisms). Due to the expected high density of T-Nodes devices embedded in the environment, the system may generate extremely large amounts of contextual information, which may disrupt the network if not adequately managed. It is therefore mandatory to envision a mechanism able to limit the diffusion of data generated by T-Nodes. In [19], such mechanism has been termed *Information Filtering*. The basic concept is that contextual data loses its significance and usefulness (i.e., its information content) when moving

away (in space and time) from the area where it has been generated. In turn, the information content of a message can be related, through standard information-theory tools, to the number of bits necessary to encode it [25]. Ideally, we would therefore have a mechanism which shrinks the size of the messages carrying contextual data as they travel within the network. The mechanisms supporting information filtering should be distributed, i.e., implemented by each U-Node. The local decisions on the level of resolution at which data needs to be kept should be done only based on the *metadata* describing the data type and attributes. Various mechanisms can be encompassed, ranging, e.g., from the simple threshold-based technique presented in [4] to more complex schemes based on the multi-resolution properties of wavelet transforms or similar techniques.

Such mechanisms have to be coupled with means for effectively retrieving stored data. In general, data will be stored in U-Nodes in the form of messages. Each message will be characterized by a set of $\langle \textit{attribute}, \textit{value} \rangle$ pairs describing various metadata associated with the message, e.g., the type of data contained, the source node etc. Messages will be filtered based on their metadata and stored in an internal database, present at each U-Node. The filters may be used, e.g., for blocking spam messages or for enabling communications only with trusted peers. Filters are based on a set of matching criteria on the $\langle \textit{attribute}, \textit{value} \rangle$ pairs associated to the message. A U-Node can also access and modify a subset of the message metadata. This includes, e.g., the number of hops traversed by a message or the number of its copy already disseminated in the system.

5.5 Interworking with Legacy Networks

The two-tier network architecture of the BIONETS project aims to enable adaptive and evolving services on top of two major classes of devices: T- and U-Nodes. BIONETS systems are meant to work without requiring the presence of any infrastructure. Further, they do not assume backward compatibility with legacy IP systems. Notwithstanding, we aim at providing the necessary features for ensuring BIONETS systems can leverage existing IP infrastructure for achieving enhanced service. At the same time, such mechanisms should enable IP-based services to access BIONETS islands to gather environmental data. This represents a partial form of interoperability, which may be enhanced in the future.

Summarizing:

- BIONETS systems do not rely on any infrastructure for functioning and performing the expected tasks;
- BIONETS systems can *opportunistically* exploit the presence of infrastructure-based IP networks for enhancing the quality and range of offered services;
- Legacy services can leverage BIONETS networks for collecting environmental data.

The internetworking capabilities with infrastructured IP networks and services are provided by BIONETS Access Points. A BIONETS access point presents the following set of features:

- it shall enable internetworking between BIONETS and legacy IP networks;

- it shall work on the U-node plane in BIONETS network as the U-nodes carry and forward the traffic generated by the services running;
- it shall have at least the same networking capabilities as the U-nodes in order to be interoperable with the U-nodes;
- it may have methods for advertising its presence to U-nodes (e.g., by means of beacon messages);
- it shall have a modular design to support different protocols running in the IP networks;
- it should not create too much extra traffic in the BIONETS network because the U-nodes are battery-wise more restricted than the AP;
- it shall have a fairly large amount of persistent memory for data buffering over long periods of time;
- it shall have the necessary CPU power to cope with the traffic flow without congesting or dropping information.

As detailed in §4.1, APs operations will rely on the presence of a proxy server able to decouple the operations on BIONETS networks (which will be handled through an interface which allows communications with U-Nodes, as shown in Fig. 3) from the ones on IP networks.

The following two additional assumptions are made:

- APs are static;
- APs have always-on IP connectivity to the Internet.

While these two assumptions may appear to be limiting (and, in particular, they do not enable the use of APs for internetworking with legacy DTNs, see Appendix A), the main focus is here on providing BIONETS the possibility of leveraging existing IP infrastructure to provide a better service.

APs may advertise their presence by broadcasting beacon messages. A U-Node passing in proximity of a BIONETS AP can decide to register with it, generating an address which will be bound to its name. A registered U-Node will be able to communicate with the IP world only when within communication range of the AP.¹ If the U-Node moves out of the AP range, communications will be broken. Conversely, IP-based servers can access local data by sending a query to the AP, which will then translate the query and send an analogous one to the U-Nodes passing by. The IP infrastructure can also be used for building shortcuts (i.e., BIONETS tunnels) for connecting different islands of nodes. In order to achieve the latter functionality, a service for location management of the APs is needed. In particular, we may assume that all BIONETS APs shall run a service, supporting query resolution for location management and providing the

¹For the moment, we decided to avoid the possibility of exploiting spreading of AP-generated or AP-destined messages on the U-Nodes plane. This is in-line with the BIONETS philosophy, which encompasses APs as optional elements, which provide “virtual contacts” among disconnected away islands.

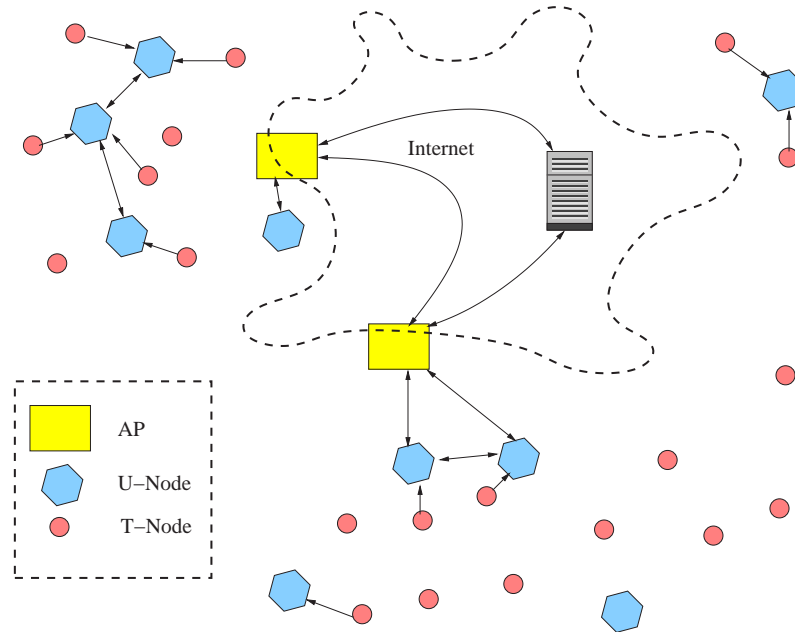


Figure 7: Graphical representation of the BIONETS network architecture, encompassing AP for leveraging IP infrastructure.

necessary means for establishing a connection among two BIONETS APs. Such service will build an overlay network, which will be used to route message among BIONETS islands. More details on the interworking with infrastructured IP networks can be found in [26].

6 Integration of Security Framework

This section reviews the emerging BIONETS high-level architecture and the security requirements we identified in deliverable D1.1.1 [4]. For completeness we summarize identified security requirements and briefly discuss how the current architecture complies with them.

6.1 Security Requirements

This section distinguishes the following three main security principals in the BIONETS network architecture: T-Nodes, U-Nodes and Access points.

The following subsections enumerate the security requirements of the above listed principals. For a detailed discussion of these requirement and for details of their derivation please refer to deliverable D4.2 [27].

6.1.1 T-Nodes

This subsection lists and explains relevant security requirements for T-nodes.

Identification: In some application scenarios, identification mechanisms to enable specific security capabilities (e.g., authentication or reputation collection) are inevitable and are thus required.

Authentication: Some application scenarios require some proof of identity. These include the application of possible device updates, the accumulation of reputation data, or the collection of mission critical T-node readings.

Access Control: Access control on T-nodes is an optional requirement, applicable to scenarios in which access to resources of the T-node or the T-node itself should or must be restricted. However, as soon as a node supports any update functionalities access control is required. Thus, we currently consider access control as an optional security feature of a T-node.

Integrity: The discovery of node manipulation is almost impossible and to implement tamper proof hardware is very costly. Although we may think of T-nodes which have similar characteristics as the secure application module (SAM) and hardware security module (HSM) on currently available smart-cards [28], we see hardware integrity as an optional requirement.

It may however be possible to secure the integrity of some parts of a T-node such as data stored on the node [29]. For a more elaborate discussion we refer to deliverable D4.2 [27].

Cryptography: Integrity and authorization mechanisms will require some cryptographic means to accomplish their tasks. Thus, cryptographic mechanisms are already a requirement for nodes which need to comply with the requirements discussed in the last subsections. Here, it has to be emphasized that cryptographic algorithms will be an optional feature of T-nodes and are not required a priori.

Key Exchange and Distribution Mechanisms: For secure communication with resource restricted devices symmetric key cryptographic algorithms are currently of first choice, because of their simplicity and execution speed. Thus, it is nearly inevitable to have key distribution mechanisms in place or to use key exchange mechanisms to establish session keys.

Bi-Directional Communication: For almost all the security requirements mentioned so far, bi-directional communication is needed and thus becomes a requirement for T-nodes with enabled security mechanisms.

Communication of Security Capabilities: As technology is expected to advance rapidly, we can soon expect T-nodes with a vast variety of capabilities and applications. As a consequence, T-nodes will be able to perform very different security functionalities. Therefore requirement for a T-node is the provision of mechanisms which can communicate the security capabilities to U-nodes.

Update Capabilities: The past has shown that updates for security mechanisms are crucial. Hence, an update capability is a requirement for T-nodes. However, this T-node requirement should depend on the quantity and task of the specific T-node type to be employed.

Availability Despite all the security measures which may be implemented in a T-node, the data delivered by them must still be available to eligible nodes. Additionally, security mechanisms in T-nodes should not cause high latencies on the processing and transmission of data and on the access to the node itself. In the presence of active T-nodes it is also very important that security features do not impose large overheads.

6.1.2 U-nodes

This subsection lists and explains relevant security requirement for U-nodes in BIONETS.

Identification: U-nodes will be subject to trust and reputation systems, they will serve to share resources with other connected U-nodes, initiate secure connections, etc. Most of these tasks will require some globally unique identifier to achieve secure operation.

Authentication: There are several scenarios in which authentication with/of a U-node is required: a user may need to prove its identity to be able to use the node with a specific role. In BIONETS U-nodes themselves may also need some means to authenticate to other U-nodes. Services may require U-node authentication.

Access Control: As discussed in the last section a U-node will require access control mechanisms for the following four logical entities in BIONETS:

Users. The U-node will provide an interface for regular user-node interaction ²

Services. Services executed on a U-node may need access to specific resources of a U-node, e.g., memory.

T-nodes. Depending on the security policy under which a U-node operates it may also be required for a U-node to control access of or to T-nodes.

U-nodes. One concept in BIONETS is cooperation between U-nodes. Many scenarios are possible in which nodes may interact to achieve a certain goal. Sharing resources between U-nodes is one of these examples and requires access control.

Integrity: If talking about node integrity at the U-node level we can think of host based intrusion detection systems. Light weight intrusion detection systems, which are at least able to discover the unauthorized modification of system files, may be required for some U-nodes, e.g., when processing highly sensitive data. We consider this requirement to be weak and optional for nodes with specialized functionalities. The integrity of data stored on a node (including services) has another focus and will be discussed in [27].

Cryptography: Also for U-nodes cryptographic primitives are required as they will be needed for an implementation of authentication, access control or integrity functionalities.

Key Exchange and Distribution Mechanisms: May be required if U-nodes decide to set up efficient secure channels with other U-nodes, APs, or T-nodes and do not want to use locally stored credentials.

Again, in this case we assume the use of symmetric encryption algorithms for sessions.

Bi-Directional Communication: Comparable to subsection 6.1.1 almost all the security requirements above require bi-directional communication. Only difference to T-nodes is the fact that we do not consider two way communication to be optional.

²Please note that such interface is not shown in the U-Node architecture in Sec. 4, where the focus is on the components relevant from the networking point of view. Same applies to the AP architecture as well.

Secure Execution Framework for Services: A secure execution framework for services is a strong requirement for U-nodes as they face evolving services, services which may show arbitrary behavior, combinations of locally running previously unknown instances of services, etc.

The secure execution framework for services is required to

- protect the U-node from malicious services
- protect other services from malicious services
- ensure secure service combination

Provide Security Functionalities: Every U-node is required to have a set of primitives and protocols isolated from the general service architecture to ensure a permanent availability of correctly working primitives and protocols. Consequently, the U-node must provide an interface which offers access to these functionalities.

Communication of Security Capabilities: Comparable to T-nodes, U-nodes must be able to communicate their security capabilities to other U-nodes. It is also required that a U-node can communicate the number and type of security mechanisms available to the services running on the node.

Update Capabilities: A U-node must offer the possibility to update its built-in services and credentials. Otherwise outdated or compromised security mechanisms may prevent secure interaction.

Availability: Security functionalities implemented on U-nodes must ensure that they do not restrict eligible nodes from using the offered services. This implies that these mechanisms do not become a single point of failure and turn whole collections of nodes or services inaccessible. Security features are also required to not impose large overheads as far as power consumption, storage requirements, and bandwidth are concerned.

6.1.3 Access Points

Identification: Access points will be used to connect several BIONETS islands, to provide environmental data to IP based services, and to access IP based services from BIONETS nodes. Additionally, they will support the security framework in BIONETS. For all these applications authentication is needed and thus identification mechanisms are required.

Authentication: As mentioned above authentication will be required. It is especially relevant to enhance the security framework.

Access Control: Access points will require access control mechanisms for the following five logical entities in BIONETS:

Users. APs will provide an interface for user interaction (e.g. super user access for configuration).

Services. Although it is not foreseen that BIONETS services run on a AP we allow for their access control. This is not an additional burden as APs will need to control the access of the BIONETS proxy server anyway but APs will also provide dedicated services, e.g., for the trust and reputation framework. In order to ensure their secure operation we consider access control for services on APs as a requirement.

T-nodes. Depending on the security policy under which an AP operates it may also be required for an AP to control access of or to T-nodes.

U-nodes. Depending on the security policy under which an AP operates it may also be required for an AP to control access of or to U-nodes.

Access points. To control access to or from the BIONETS island from or to another BIONETS island (specified by the security policy), the AP may need to support access control.

Other IP hosts. Although not being part of the BIONETS architecture, access of other IP hosts to the AP as well as to the BIONETS island the AP is connected to has to be controlled.

Integrity: Light weight or regular host-based intrusion detection systems may be required for APs especially if they constitute a backbone for the support of security framework. In case of a detected compromise they should disable the BIONETS proxy and do not support connectivity anymore.

Cryptography: For APs cryptographic primitives are definitely required. APs are foreseen to support the security framework and they will need at least some means for authentication.

Key Exchange and Distribution Mechanisms: Will required if APs decide to set up efficient secure channels with U-nodes or T-nodes and do not want to use pre-distributed keys.

Bi-Directional Communication: As APs are one level higher in the logical BIONETS hierarchy than U-nodes and in order to be able to support all security mechanisms that U-nodes can offer, APs also have to support bi-directional communication.

Secure Execution Framework for Services: A secure execution framework for services is only a weak requirement for APs. Although it may be possible to execute dedicated BIONETS services on APs (e.g. reputation aggregation services) they will definitely not be part of the service evolution, combination, etc.

Communication of Security Capabilities: As in the T-node and U-nodes case, APs must be able to communicate their security capabilities to other BIONETS nodes.

Update Capabilities: APs must offer the possibility to update their built-in services and credentials. Otherwise outdated or compromised security mechanisms may prevent secure interaction.

Availability: As APs are an logical entity in the BIONETS architecture which basically complements the general functionality, availability of APs is not crucial.

In any case, security functionalities implemented on APs should not become a single point of failure and turn whole collections of nodes or services inaccessible.

Security features running on APs are also required to not impose large overheads for the nodes it is communicating with.

6.2 High Level Security Architecture

Following the collection of the security requirements for nodes in the BIONETS architecture, this section describes how the security framework integrates and how it accounts for the above requirements.

Please note that this section only tries to reflect the basic security framework which is elaborated in deliverable D4.2 [27].

6.2.1 T-nodes

In this section we refer to T-node classes. For their elaborate description and the explanation of the meaning of these classes and their security capabilities we refer the reader to [30, 26].

Identifiers for T-nodes as described in section 5.2 are neither unique nor globally valid. Thus, they are neither useful for authentication purposes nor for applications such as trust and reputation system. However, we assume that for higher T-node classes, such as class B.y to D.y T-nodes, there will be nodes, which can be associated with a unique identifier. This identifier will be stored in some memory on the T-node. The data security unit (DSU) together with the communication security unit (CSU) will enforce a security policy (see figure 1). For example, you may imagine that an unauthenticated U-node tries to access the identifier of a T-node with a read operation. The security policy on the node, however, can require that only authenticated and authorized nodes are able to read this information. In this example, the privacy of the user would be protected. Accordingly, the node would deny access if the security requirements are violated.

Any *authentication* task of the T-node will be conducted by the communication security unit (CSU) which is the second and last component which has direct access to the credentials of the T-node, including its identifier.

The *access control* concept for data has been explained above. The access of other entities, for example to update the T-node with new primitives or credentials or to query data from the T-node through its network interface, is controlled by the CSU.

It is hard to illustrate the realization of hardware *integrity* in a picture. However, in deliverable D4.2 [27] you can see, that data, security primitives and protocols are stored in different modules of the node. Optionally, such modules may be placed in SAMs or HSMs as described in [28].

As we have just explained, *cryptographic primitives* and *protocols* will be stored on the node. Their type and variety are application dependent. They will be used by the CSU and DSU to comply with the T-node and application specific security requirements. Consequently, for each

scenario, it is required to determine a minimal set of these primitives and protocols. Clearly, it is impossible to design cheap T-nodes which comply with all these sets. Additionally, there may be services which have not been foreseen in the design phase and which require totally different security mechanisms on a T-node. In work package four, task 4.3, we are currently developing a model which tries to overcome this problem. It aims at a set of a fixed number of low-level primitives (such as permutation boxes, addition in $GF(2)$, substitution boxes, etc.) able to realize a large variety of higher-level security primitives (such as RC5, DES, AES, etc.) and protocols (such as key exchange or authentication protocols).

Using these cryptographic primitives and protocols the CSU of a T-node is also capable to run *key exchange* protocols. Again, the type of this protocol is dependent on the protocol and primitive suite stored on the node.

Non-volatile memory units on a T-node, which may also be secure unit such as SAMs or HSMs, will allow for *key distribution mechanisms* which store a key or a set of credentials in the deployment phase of the T-node. T-node classes A.0 and A.1 do not offer a *bi-directional communication* interface. However, neither of these classes support the above mentioned security protocols. Thus the T-node architecture accounts for bi-directional communication.

The *communication of security capabilities* is accomplished by the CSU which uses data stored on the node, containing information about the available security primitives and protocols.

An update interface which is controlled by the CSU enables *update capabilities* of the node.

Availability can not be illustrated but is a design feature which should ensure smooth operation. As an example, this requirement has to be kept in mind when selecting a protocol suite for the T-node. The same holds for all other node types.

6.2.2 U-nodes

As discussed in section 5.2 U-nodes will possess an identity, as they will be subject to a trust and reputation system. As in T-nodes, this identifier will be stored in non-volatile memory which is secured by the so called *Security Enforcement Unit* (SEU). Protection in this context means that the unit controls access the data of the node.

The SEU (also see Fig. 2) is basically a logical high-level component which also contains a communication security subunit (CSSU) as we find it in T-nodes. For consistency this subunit also conducts *authentication* tasks. By means of the SEU, the CSSU has access to the required credentials and security primitives and protocols of the node.

The implementation of *access control* mechanisms in a U-node is more complex as more interacting entities are involved. However, to keep this description as generic as possible we can indicate that the data security subunit, the communication security subunit (CSSU), and the service security subunit (SSSU) in the SEU are responsible for policy decision and the policy enforcement subunit (PESU) for policy enforcement. These are the basic building blocks of access control. For more details refer to [27].

In the near future we can not realize a U-node in which we can ensure full hardware *integrity* as this would be far too expensive. Parts of a node may be protected however. Some mechanisms

were already mentioned in the last subsection. To detect compromises of a U-node or U-node manipulation through malicious services, users, or connection from other nodes a host based intrusion detection system may be employed. This could be implemented as a special type of service.

Implementations of *cryptographic primitives* and *protocols* will be stored on the U-node in non-volatile memory which is accessible by the *security enforcement unit* to accomplish miscellaneous tasks. As the node will also have to *provide security functionalities* to services running on the node the SSSU allows for the enhancement of existing services with security primitives or protocols. Security protocols and security primitives are not subject to service evolution. This allows the use of functionalities with specific security characteristics.

As in T-nodes and as described above, U-nodes are endowed with the required functionalities to run *key exchange* protocols and to make use of *key distribution mechanisms*.

Not only the latter functionality will require interfaces capable to provide *bi-directional communication*. As described in section 4.1 U-nodes are equipped with such interfaces.

The control over the execution of services is eminent in BIONETS. Services are similar to active code, they are modified autonomously and they combine with other services from the network to name just some characteristics. Thus, a *secure execution framework for services* is realized by running code in an isolated environment which has to forward any access to data, network, other services, etc. through the SEU. Even service composition is controlled by the SEU. In this way we have a static as well as dynamic mechanisms in place to ensure execution.

Also U-nodes store the information about security suites installed. This data enables U-nodes to *communicate their security capabilities* to other nodes.

Likewise, U-nodes offer *update capabilities* via an update interface. All update efforts are controlled by the SEU to avoid unauthorized access.

6.2.3 Access Points

As mentioned in section 4, APs will basically offer the same functionalities as U-nodes. Main architectural difference is the nonexistent service execution unit. Thus, if you need more details than provided by the explanations below, please refer to the last subsection or to [27].

Comparable to U-nodes, access points will store *identifiers*. They can be equipped with more than one identifier which can be used in different contexts, e.g. authentication with IP hosts, other APs, U-nodes, and T-nodes, respectively.

The *authentication* tasks required in APs will be conducted in the SEU of the AP (also see figure 3).

Also the *access control* mechanisms will be conducted by the SEU. Only difference to the U-node case are the entities we provide access control for (e.g. U-node do not have to provide access control mechanisms for IP hosts).

Similar *integrity* mechanisms as in U-nodes can be employed. In fact, one may think of employing more rigorous intrusion detection and prevention mechanisms as the node is directly exposed to Internet traffic.

Cryptographic primitives and *protocols* will be stored on the AP in non-volatile memory. Access to this memory, including updates, will be controlled by the SEU.

APs will support public key cryptosystems and *key exchange* and *distribution mechanisms*. Appropriate protocols are stored on the U-node or are applied in the deployment phase.

As described in section 4.1 APs are equipped with *bi-directional communication* interfaces for U-node and T-node interaction as well as for communication with other IP hosts.

We foresee the execution of dedicated services on the AP, e.g. services supporting the trust and reputation system. As these are dedicated services we ensure *secure execution for services* by applying the same checks as provided by U-nodes before we deploy the service in the AP.

Also an AP will be able to *communicate security capabilities* to other BIONETS entities by using the information about its capabilities stored on the AP.

Update capabilities of APs are provided in the same way as in U-nodes.

6.3 Conclusion

Based on the requirement collection which was extended to access points we developed a security architecture which complies with these requirements. This section discussed why specific requirements are relevant for specific node types. We outlined how these requirements could be implemented on a node level. Given that the network is still subject to research and as we expect more refinements to occur the security framework can not be a final structure either. It will rather develop with the network architecture in parallel.

7 Outlook and Next Steps

In this deliverable, we have presented an enhanced version of the system requirements and disappearing network architecture presented in [4]. Particular attention has been paid to the integration in the architecture design of the required security features.

The proposed network architecture is aligned with the BIONETS service architecture detailed in [5]. The next logical step shall be the merging of the two architectures into an integrated *SerWorks* (i.e., Services + Networks) architecture. This involves the design of a flexible, lightweight architecture able to support (self-evolving) service-oriented operations in the BIONETS framework, characterized by features such as node heterogeneity, disconnected operations and nodes mobility.

References

- [1] I. Carreras, I. Chlamtac, H. Woesner, and C. Kiraly, “BIONETS: BIO-inspired NExt generation networkS,” in *Proc. of WAC*, Berlin, DE, 2004.
- [2] “BIONETS project.” [Online]. Available: <http://www.bionets.eu>
- [3] M. Weiser, “The computer for the 21st century,” *ACM Mob. Comput. Commun. Rev.*, vol. 3, no. 3, pp. 3–11, 1999.
- [4] D. Miorandi (editor), “Requirements and Architectural Principles: Application scenario analysis, network architecture requirements and high-level specification,” BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D1.1.1), June 2006.
- [5] J. Huusko (editor), “Service architecture: requirement specification and concept definition,” BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D3.1.1), June 2006.
- [6] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, “Delay-tolerant network architecture,” 2007, IETF RFC 4838.
- [7] F. D. Pellegrini (editor), “Disappearing Network Infrastructure and Design: Functionality and Challenges,” BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D1.2.1), June 2006.
- [8] S. Steglich (editor), “Specification of service life-cycle,” BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D3.2.1), January 2007.
- [9] J. Huusko (editor), “Refinement of Service Architecture and Requirements,” BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D3.1.2), July 2007.
- [10] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley, “The design and implementation of an intentional naming system,” in *Proc. of ACM SOSP*, Kiawah Island, SC, 1999, pp. 186–201.
- [11] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, “Building efficient wireless sensor networks with low-level naming,” *ACM SIGOPS Operating Systems Review*, vol. 35, no. 5, Dec. 2001.
- [12] V. Bharghavan, “A dynamic addressing scheme for wireless media access,” in *Proc. IEEE ICC*, Seattle, WA, 1995, pp. 756–760.
- [13] C. Schungers, G. Kulkarni, and M. B. Srivastava, “Distributed on-demand address assignment in wireless sensor networks,” *IEEE Trans. on Parallel and Distributed Systems*, vol. 13, no. 10, October. 2002.
- [14] J. Elson and D. Estrin, “Random, ephemeral transaction identifiers in dynamic sensor networks,” in *Proc. of ICDCS*, Phoenix, AZ, 2001.

- [15] B. Ford, J. Strauss, C. Lesniewski-Laas, S. R. amd Frans Kaashoek, and R. Morris, “Persistent personal names for globally connected mobile devices,” in *Proc. of USENIX OSDI*, Seattle, Washington, 2006.
- [16] A. Lindgren, A. Doria, and O. Schelen, “Probabilistic routing in intermittently connected networks,” in *The First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR 2004)*, 2004.
- [17] T. Spyropoulos, K. Psounis, and C. Raghavendra, “Single-copy routing in intermittently connected networks,” in *IEEE SECON*, October 2004.
- [18] —, “Spray and wait: An efficient routing scheme for intermittently connected mobile networks,” in *Proceedings of SIGCOMM 2005*, August 2005.
- [19] I. Carreras, I. Chlamtac, F. De Pellegrini, and D. Miorandi, “Bionets: Bio-inspired networking for pervasive communication environments,” *IEEE Trans. Veh. Tech.*, vol. 56, pp. 218–229, Jan. 2007.
- [20] A. Panagakis, A. Vaios, and I. Stavrakakis, “Study of two-hop message spreading in DTNs,” in *5th Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, Limassol, Cyprus, 2007.
- [21] A. Spyropoulos, K. Psounis, and C. Raghavendra, “Performance analysis of mobility-assisted routing,” in *ACM MOBIHOC*, May 2006.
- [22] I. Carreras, D. Tacconi, D. Miorandi, and F. Pellegrini, “A multi-resolution data management scheme for opportunistic information diffusion,” CREATE-NET, Tech. Rep. 200600012, November, 2006.
- [23] Y. A. Korilis, A. A. Lazar, and A. Orda, “Architecting noncooperative networks,” *IEEE JSAC*, vol. 13, no. 7, pp. 1241–1255, Sept. 1995.
- [24] A. Panagakis, A. Vaios, and I. Stavrakakis, “On the effects of cooperation in DTNs,” in *The Second IEEE/Create-Net/ICST International Conference on COMMunication System softWARE and MiddlewaRE (COMSWARE)*, Bangalore, India, 2007.
- [25] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: John Wiley & Sons, 1991.
- [26] D. Raz (editor), “Disappearing Network Autonomic Operation and Evolution,” BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D1.2.2), June 2007.
- [27] D. Schreckling (editor), “Towards security in BIONETS,” BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D4.2), July 2007.
- [28] W. Rankl and W. Effing, *Handbuch der Chipkarten*, 4th ed. München, Wien: Carl Hanser, 2002, vol. ISBN 3-446-22036-4.

- [29] A. Seshadri, A. Perrig, L. van Doorn, and P. Khosla, “Swatt: Software-based attestation for embedded devices,” in *IEEE Symposium on Security and Privacy*, 2004. [Online]. Available: citeseer.ist.psu.edu/seshadri04swatt.html
- [30] D. Schreckling (editor), “BIONETS advanced security mechanisms,” BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D4.2), July 2007.

A Interworking with Legacy DTNs

APs were introduced to enable BIONETS to opportunistically leverage IP infrastructure. At the same time, we can imagine a situation in which BIONETS islands of U- and T-Nodes coexist with legacy DTNs [6]. Such networks employ the store, carry and forward paradigm, but are not compatible with the BIONETS communication paradigm with respect to data collection (communication with the T-nodes) and service-specific data dissemination (dissemination protocols employed by the U-nodes). It may be therefore of interest to address the interoperability issues arising in such framework. In particular, such possibility was not foreseen in the original project scope, and appeared only during the preparation of this report. It is indeed clear that APs, as defined in the deliverable at hand in terms of supported functionalities and architecture, are not suitable for performing such task.

In particular, when considering interworking with legacy DTNs, the following issues have to be accounted for:

- BIONETS systems could opportunistically exploit the presence of DTNs for enhancing the speed at which the message dissemination process takes place;
- BIONETS messages may be encapsulated in DTN bundles;
- Being address-centric, DTN architecture will not provide the filtering of information performed by U-Nodes;
- DTN security architecture may not be fully compatible with BIONETS security mechanisms;
- Legacy DTN systems could leverage BIONETS systems to efficiently gather context-augmenting data;
- Legacy DTN systems could leverage BIONETS systems to increase the diffusion speed of messages.

Adding to BIONETS another class of devices (probably some sort of enhanced U-Nodes) for handling interoperability with legacy DTNs may provide the system with the opportunity of obtaining a faster dissemination of messages. At the same time, it could enable DTNs to profit from the peculiarities of BIONETS systems. On the other hand, there are non-trivial technical issues to be addressed; therefore we decided for the moment to leave such interoperability issues out of the scope of the workpackage activities and focus on the more prominent aspects of the BIONETS network architecture.