



## **BIONETS**

### **WP 4 – SECURITY**

#### **UNIVERSITY OF TRENTO**

### **D4.1 Trust and Reputation Management System Definition**

<b>Reference:</b>	BIONETS/UNITN/wp4/1.3
<b>Category:</b>	Deliverable
<b>Editors:</b>	Roberto Cascella, Anurag Garg (CN-UNITN)
<b>Authors:</b>	Anurag Garg, Roberto Battiti, Roberto Cascella, Alberto Montresor, Mauro Brunato (CN-UNITN)
<b>Verification:</b>	Fabio Martinelli (CN-CNR), Ioannis Stavrakakis (NKUA), Daniel Schreckling (UNIHH)
<b>Date:</b>	June 18, 2007
<b>Status:</b>	Final
<b>Availability:</b>	Public

---

## SUMMARY

---

### Contents

<b>1</b>	<b>Executive Summary</b>	<b>5</b>
<b>2</b>	<b>Introduction</b>	<b>6</b>
2.1	Motivation of the Deliverable . . . . .	6
2.2	Structure of the document . . . . .	7
<b>3</b>	<b>Problem Statement</b>	<b>7</b>
<b>4</b>	<b>State-of-the-Art</b>	<b>8</b>
<b>5</b>	<b>Breakdown of reputation management systems</b>	<b>9</b>
5.1	Reputation Aggregation Locations . . . . .	10
5.2	Feedback Collection . . . . .	11
5.3	Reputation Aggregation functions . . . . .	12
5.4	Reputation dissemination . . . . .	12
<b>6</b>	<b>BIONETS Architecture</b>	<b>13</b>
6.1	Disconnected Operation . . . . .	14
6.2	Mobility . . . . .	14
<b>7</b>	<b>Reputation System Definition in BIONETS</b>	<b>15</b>
7.1	Collection of Feedback . . . . .	16
7.2	Reputation Aggregation . . . . .	18
7.3	Dissemination of Trust Information . . . . .	19
7.4	Types of Reputation . . . . .	19
7.5	Reputation Aggregation Algorithms . . . . .	22
7.5.1	Age-Weighting of Trust Information . . . . .	22
<b>8</b>	<b>Applications of Reputation Management in BIONETS</b>	<b>24</b>
<b>9</b>	<b>Threat Analysis</b>	<b>27</b>
9.1	Identity and Trust . . . . .	27
9.2	Feedback and Reporting of Opinions . . . . .	28

9.3 Collusions in an <i>island</i> . . . . .	28
9.4 Inconsistent behaviour . . . . .	29
9.5 Denial of Service . . . . .	29
<b>10 Conclusions</b>	<b>29</b>

## DOCUMENT HISTORY

### Version History

Version	Status	Date	Author(s)
0.1	Draft	October 31, 2006	Anurag Garg, UNITN
0.2	Draft	November 10, 2006	Roberto Cascella, UNITN
0.3	Draft	November 13, 2006	Anurag Garg, UNITN
0.4	Draft	December 10, 2006	Roberto Cascella, UNITN
0.5	Draft	December 11, 2006	Authors, UNITN
0.6	Draft	December 20, 2006	Authors, UNITN
0.7	Complete Draft	January 4th, 2007	Authors, UNITN
1.0	Final	January 10th, 2007	A. Garg, R. Cascella, R. Battiti, UNITN
1.1	Revision	February 2nd, 2007	Authors, UNITN
1.2	Revision	June 8th, 2007	Authors, UNITN
1.3	Revision	June 18th, 2007	Roberto Cascella, UNITN

### Summary of Changes

Version	Section(s)	Synopsis of Change
0.1	Structure and Draft of Sections	Inclusion
0.2	Draft of Sections 1, 2, 3	Inclusion
0.3	Draft of Section 7	Inclusion
0.4	All Sections	Revision
0.5	Final version of Sections 6, 7	Inclusion
0.6	Draft of Section 8, 9, 10	Inclusion
0.7	All Sections	Inclusion of final version
1.0	All Sections	Revision and typos fix
1.1	All Sections	Revision from internal verification
1.2	All Sections	Document structure, Inclusion and Editing
1.3	All Sections	Revision, typos fix

## 1 Executive Summary

This document defines the BIONETS trust and reputation management system and summarizes the work performed in Task 4.2 to identify the security challenges unique to the BIONETS architecture and to define solutions to these challenges through the trust and reputation management system that will be used to provide “soft” security management of resources in BIONETS. As such, trust and reputation management is not a substitute for “hard” security mechanisms that will be used to protect BIONETS from external and internal threats but is a complementary technique that encourages the “right” behaviour by BIONETS components.

This document is not intended to provide detailed algorithms or implementation details of the trust and reputation management systems. Instead, it looks at the BIONETS node and system architecture from the point of view of trust and reputation management, examines the suitability (or lack thereof) of existing trust and reputation management techniques in this context and finally presents a system definition incorporating existing know-how and novel techniques that are necessitated by the unique evolutionary (and revolutionary) nature of BIONETS.

In defining the trust and reputation management system, we take into account the heterogeneous capabilities of the nodes that participate in the system. As the types of the nodes are different, including T-Nodes, U-Nodes and AP-nodes, the reputation system reflects this heterogeneity by defining different types of reputation associated with each type of node. In addition, services in BIONETS also have reputation values that are an indicator of the service demand, quality and reliability. BIONETS services are autonomic and evolve to adapt to the surrounding environment, like living organisms evolve by natural selection. One of the primary mechanisms through which this natural selection takes place is service reputation. A service that has a low reputation will not find any clients and would gradually disappear from the network. A service that meets demand in a satisfactory manner will find its reputation increase and more resources will be allocated to it as a result.

The BIONETS reputation management system will also be used to provide flexible soft-management mechanisms that are appropriate for a highly disconnected network. Disconnected operation will be common in BIONETS and there will often be isolated “clouds” of nodes that are only able to communicate with each other and are unable to communicate with anyone outside the cloud. In this situation, a transient trust network will be setup which will allow nodes to store all local transactions values or to calculate temporary reputations of other nodes until reconnection to the rest of the network, when reputation values for the nodes can be synchronized and updated.

To show how the reputation management architecture defined in this document will respond to attacks we present a threat analysis. In this analysis, many typical attacks on autonomic systems are listed and we show how the BIONETS reputation management system reacts to them. It must be noted that this response may not be adequate to secure the network as a full response to all of these attacks will necessarily involve the BIONETS security architecture and it is out of scope of this document. Nevertheless, this document provides an important component of the overall BIONETS security architecture.

## 2 Introduction

BIONETS is a novel network architecture that overcomes device heterogeneity and achieves scalability via an autonomic and localized peer-to-peer communication paradigm. Significantly, BIONETS will achieve these goals in a resource-constrained, dynamic and heterogeneous “disappearing network” environment, where disconnected operation is common, the population of nodes is vast, and node cooperation cannot be assumed. Biologically-inspired concepts permeate the network and its services, blending them together. The barrier between service providers and consumers (or users) is broken and services can “mushroom” spontaneously as required, paving the way for a service-centric future.

One method of achieving the spontaneous creation and evolution of services is by injecting user feedback and reputation information as the fitness criteria for service adaptation. At a later stage of the project, user feedback and reputation information will also be coupled with biologically-inspired solutions to self-organization and self-management to build user- and environment-aware services that are also robust, secure, able to deploy, manage and protect themselves, and continue to evolve and maintain themselves after deployment, without requiring human supervision.

However, the scope of this document is limited to the definition of the trust and reputation management system that will exist in BIONETS to manage user feedback and reputation information. The document identifies the challenges presented by the unique network structure of BIONETS and defines the solutions through the reputation management system presented.

### 2.1 Motivation of the Deliverable

This deliverable defines parameters and concepts concerning trust and reputation that pave the way for the deployment of a trust and reputation scheme suitable for the BIONETS system. The framework analyzed in this document presents the peculiarities of the BIONETS system and issues and challenges that we face in the definition of an incentive scheme for nodes’ cooperation. The concepts discussed in the document are of importance for other tasks and work packages that exploit nodes’ cooperation and they present the initial effort toward a soft-managed security of the BIONETS network.

This deliverable is a preliminary step in the design of a trust and reputation scheme integrated into the BIONETS network. This work is part of Task 4.2 which should:

- provide a preliminary framework for monitoring the performance of individual nodes and how they will honour their contractual obligations;
- define a mechanism that exploits positive feedbacks in past interactions;

The purpose of this deliverable is to detail the results accomplished by Task 4.2 during the first year of the project and it is supposed to provide the basic knowledge in terms of security measures and threats to other tasks and work packages, that should consider the results obtained as possible guideline to conduct their tasks. The document includes a general overview of the BIONETS architecture and its requirements in terms of reputation and trust management.

## 2.2 Structure of the document

The remainder of the document is organized as follows. Section 3 describes the problem we tackle in this deliverable situating it in the context of BIONETS. This is followed by a review of the state-of-the-art in Section 4 which discusses existing trust and reputation management systems. Section 6 discusses the features of BIONETS that are relevant to trust and reputation management. It discusses the assumptions we make in our system definition and the constraints that come from the BIONETS architecture itself. In particular we discuss the impact of disconnected operations and mobility on reputation management and section 6.1 discusses architectural designs to allow the system to function in the presence of disconnected islands of nodes.

Section 7 presents the main result of this deliverable: the trust and reputation system definition that will be used to self-manage trust in the BIONETS network. This includes a discussion on the three services that will be the components of the BIONETS reputation system: collection, aggregation and dissemination of trust information. In this context, Section 7.4 discusses the types of reputation that will be available in BIONETS due to the different classes of nodes and services. The algorithms that will be used to calculate the trustworthiness of the nodes are discussed in Section 7.5.

Section 8 then presents the applications of the reputation management scheme defined in BIONETS. This includes a discussion of the typical application of reputation management as used in BIONETS as well as applications that are unique to BIONETS. Section 9 performs a threat analysis pointing out the attacks that can be made targeting the BIONETS reputation management architecture. This is useful for the security work package as a whole since not all attacks can be dealt with by the reputation management system alone. We conclude in section 10.

## 3 Problem Statement

The world envisioned in the BIONETS project is populated by heterogeneous devices that both compete and cooperate in providing and consuming services available in the network. BIONETS exploits these properties of an autonomic system to achieve scalability and to maintain survivability of the system which evolves continuously. Nodes acquire sufficient reasoning to make decisions, to form opinions on other nodes and to create the feedback mechanisms that enable context-awareness of services.

However, the heterogeneity, the autonomic adaptation and self-evolution of services introduce new challenges and security issues in the BIONETS network. The feedback mechanism and the evolution of the service as aggregated components require nodes' cooperation and willingness to implement the protocols without modification to the fairness mechanisms defined in them. Furthermore, the service-centred network envisioned in BIONETS is defined in such a way that services can evolve and can adapt to the surrounding environment. This property implies the exploitation of the knowledge available in the network and of the cooperation between nodes. It is not possible to assume *a priori* cooperation of the nodes as their goals may differ from those of the system as a whole. In many cases nodes may act selfishly by not propagating information or by denying access to information or by injecting false data in the network. In this potentially non-cooperative environment nodes' trust and reputation become key components to sustain the stability of the system. For instance, trust and reputation can be used to decide whether to interact with a node so that

network resilience is increased by reducing attacks based on nodes' misbehaviour.

Traditional strategies for ensuring node cooperation and incentivizing good behaviour are helpful in BIONETS but are inadequate because BIONETS nodes have different capabilities in terms of mobility, processing and storage, the population of nodes is vast and interactions between nodes can be sporadic and discontinuous. Hence, BIONETS should ideally use self-management schemes that exploit the heterogeneous capabilities of the nodes themselves. Given the decentralized nature of the communication, the solution is to have an adaptive scheme based on user feedbacks and on reputation information with an emphasis on using *locally* available information and caching reputation information to survive periods of disconnectedness.

## 4 State-of-the-Art

The traditional network architecture has been that of a client-server model where a client program, running on one host, requests services and receives a response from a server program, running on another host. In such architecture, the mechanisms that are appropriate for securing the network can be classified as *hard security* [21] mechanisms because their purpose is to secure the network from intruders and malicious users and, at the same time, allow access only to authorized users. It is assumed that a user that is properly authorized and authenticated as such will not take any steps that are harmful to the network and its actions can be tracked. This model, known as the AAA model, relies on *authentication*, *authorization* and *accounting*.

In recent years there has been a growth of distributed systems of a new kind where the member nodes are autonomous, i.e. not under the direct administrative control of a single entity, such as peer-to-peer networks, mobile ad hoc networks (MANETs) and autonomic networks. These networks generate new security challenges that cannot be tackled by traditional measures. Nodes may be properly authenticated and authorized and yet behave in a manner that is detrimental to the overall network goals. This is due to the nature of these networks that harness the excess resources of nodes in the network to provide services to the nodes themselves. A selfish node that does not want to expend any of its own resources can thus consume resources from the system but withholds its own excess resources, decreasing the level of service provided to other nodes.

Hence, there needs to be a means to ensure that member nodes will behave in a manner that contributes toward overall system goals. Nodes that do not cooperate in a way that contributes toward system goals are called **misbehaving**. Misbehaviour can be classified as harmful behaviour or behaviour arising out of self-interest when non-cooperation results in greater benefit to the node than cooperative behaviour. The juxtaposition of self-interest and overall system gains has been the subject of much study in the past, particularly in the area of game theory and mechanism design [15, 3, 20] and it is best illustrated by well-known problems such as the Prisoners' Dilemma and its generalization [5]. Therefore, it becomes necessary to create a rewards or incentives system such that the maximization of their own gains by "selfish" or "rational" nodes also maximizes the overall gains to all members. Nodes that do not share their own resources with other nodes receive proportionately poorer service in return to the point of being excluded from the network.

The system also needs to protect itself from misbehaving nodes that are acting in a way that is harmful to it or to other nodes. This can result in several well-known security problems such as Denial-of-Service

(DoS) attacks, routing disruption attacks (where a node sends forged routing packets to create routing loop or to partition the network) and resource consumption attacks (where a node injects extra data packets in the network in order to consume precious resources). This misbehaviour may be due to a malfunction in a node or it may be due to deliberate maliciousness. However, from the point of view of the system, malfunctioning and deliberate maliciousness are the same as both can damage it. Both are labelled as examples of “Byzantine” behaviour [2] as opposed to “Rational” behaviour exhibited by selfish nodes.

In this context, reputation-based trust management systems are an invaluable tool for measuring the trustworthiness of a node. Previous work in this area has drawn on existing research from several disciplines including evolutionary biology [27, 14, 10], sociology (social network theory [4, 25, 16]) and a number of fields within computer science (e-commerce, peer-to-peer systems, email filtering etc.). Examples of existing reputation-based trust management systems include XREP [8], NICE [18], P-Grid [1], PeerTrust [29], EigenTrust [17], ROCQ [12, 13] (and several others such as CORE, CONFIDANT, SPORAS, HISSTOS, DCRC/CORC, Beta etc.)

## 5 Breakdown of reputation management systems

Reputation management systems monitor the behaviour of member nodes during transactions with other nodes and assist member nodes in selecting interacting nodes. To provide this service, information on nodes’ behaviour is collected and then aggregated to compute the trust value of a node. Trust values are shared with other nodes who take appropriate action such as quarantining a malicious node or ignoring information received from a malfunctioning node. Hence a reputation management system performs three distinct functions: 1) collection, 2) aggregation and 3) dissemination of trust information. The first and the third function require communication among the nodes of the network whereas the second function, that of aggregating reputation information, may be performed locally at a single node or at multiple nodes depending on the reputation system architecture (shown in Table 1).

	Centralized Systems	Distributed Systems
<b>Collection</b>	Send report to Trusted Third Entity	Send report to aggregation locations
<b>Aggregation</b>	Centralized computation	Distributed or local computation
<b>Dissemination</b>	Ask reputation value to Trusted Third Entity	Request reputation value of transacting node from aggregation locations

Table 1: Architectural generalization of the reputation mechanism

The architectural and implementation details of the reputation mechanism depend on the underlying network on which the online community is based. When the community is built on top of a traditional client-server network, a trusted third party exists to collect and aggregate opinions to form a global view. e-Bay feedback, Amazon customer review and the Slashdot distributed moderation systems are all examples where feedback from users is stored in a centralized trust database. In this case all nodes can report their observations of other nodes to this central node. The aggregation is performed in this centralized database and all users have access to the global reputations thus computed. Nodes may request this value from the trust database for a node/service with which they are about to interact and decide to go ahead with the interaction if the returned trust value is satisfactory.

When the community is built on top of a P2P network (such as Gnutella or Kazaa), the challenges of managing feedback become much harder. There is no centralized, trusted, reliable, always-on database and the collection, storage, aggregation and dispersal of trust information must be done in a distributed way. One possibility is to let each node contact every other node in the network periodically to exchange information about all the nodes either of them interacted with. This solution is impractical for BIONETS as not only is the communication overhead very high and not scalable but BIONETS cannot guarantee that all nodes will be able to contact all the other nodes in the network all the time.

Another approach relies on choosing one or more designated agents for each peer in the network. All trust information related to the given peer is sent to these agents which have a complete view of the given peer's transaction history. Reputation management systems that have used such agents (variously called agents, score managers, trust managers etc.) include P-Grid [1], PeerTrust [29], EigenTrust [17] and ROCQ [12]. Relying on third parties for storage and dissemination makes the system vulnerable to tampering and falsification. The system must also provide redundancy because peers may drop out of the network at any time.

## 5.1 Reputation Aggregation Locations

In this section we analyze in details possible aggregation and storage location for the reputation value. Depending on the reputation system architecture and the feedback collection mechanism, aggregation of the collected feedback can be performed at a number of places in order to form the reputation value. A number of aggregation locations have been discussed in the literature [19, 11].

- *Transacting Node*: In this case a node aggregates only its personal experience with the node whose reputation value is being computed. Nodes rely on first-hand experience only and do not share experiences with other nodes. There is no need for reputation information collection and dissemination over the network as nodes produce and consume all the reputation information by themselves.
- *All Nodes*: In this scenario all nodes share information about all other nodes. This method generates a large amount of network traffic and scales quadratically with the number of nodes. It is only practical in small networks and when transactions are rare. Since nodes may share false information, this method requires a node to weigh collected information according to the credibility it places in the reporting node.
- *Central Database*: As the name suggests, this method requires all nodes to report feedback to a central trusted database which aggregates the reputation information of all nodes and disseminates it to nodes that require this information.
- *One-hop Neighbours*: A node shares first-hand experience with all nodes when it comes into contact with them. Hence, a node may have the reputation information furnished to it by all nodes it has interacted with. A node may wish to increase the number of one-hop neighbours in order to increase the amount of information it collects. Again, a node may wish to weigh the information it receives by the credibility of the reporting node.
- *Multi-hop Neighbours*: A node asks the nodes it comes into contact with to give it all the information they have and information that themselves receive from their neighbours. In this way, a transitive

chain of trust can be created between nodes that have a trust relationship. This method is popular among schemes that use trust currencies or credits [28, 7].

- *Designated Agents*: All trust information pertaining to a particular node is stored at (possibly trusted) agent or agents. An algorithm is used to determine the agent(s) designated to store information about each node. All nodes that interact with a node send their feedback to the designated agent(s) for that node using this publicly known algorithm. A designated agent may not be trustworthy or reliable, i.e., it may leave the network or lose this information. Multiple agents can be used to ensure a measure of redundancy. A designated agent does not have to know the identity of the nodes whose information it is storing. A one-way hash function can be used to obscure the identity of the node.

The most common method of choosing the designated agents is through distributed hash tables (DHT). DHTs provide an addressing and routing mechanism for nodes in distributed systems. They also ensure that designated agents are randomly chosen from the entire node population. They also provide some protection through the anonymity offered by their inbuilt routing mechanisms as the designated agents do not know whose trust information they are storing. DHT-based solutions have the advantage of being scalable and yet providing a complete view of the information relevant to constructing the reputation of a peer.

## 5.2 Feedback Collection

This section deals with possible approaches to collect information that pertains to the behaviour of a user in all of its previous interactions. The gathered information represents the input to the reputation aggregation function and can be collected in a proactive, reactive or hybrid way.

If the information is collected reactively a node will only send its feedback when it is requested to do so. This will require the reputation gathering service to periodically poll each node and ask them what feedback they have on nodes with whom they have interacted since the previous round of feedback collection. This mode has the advantage of being more efficient as long as the polling frequency is not too high. Each transaction between any two nodes will not trigger additional network load due to the gathering of reputation information. However, this mode can also result in a delay in the detection of node misbehaviour as reputation information is not reported immediately.

If reputation information gathering is done proactively, a node will send the feedback on its transaction partners after every transaction. This may incur additional network overhead but reduces delays in the detection of node misbehaviour. A transaction in this context is defined as a single, self-contained interaction between the two nodes, usually involving the provision of data or computation. The precise definition of a transaction will depend on the application context but examples of transactions include transferring a file or a portion of a file, forwarding a message on behalf of another node etc.

The hybrid approach consists in implementing a reactive and proactive scheme. A node periodically polls other nodes to gather information. However, if a node consider the importance of the feedback time-critical it might decide to disseminate this information after a transaction.

### 5.3 Reputation Aggregation functions

The reputation value of a node is calculated by aggregating the information pertinent the history of the node itself. Several functions can be used to aggregate reputation values starting from a simple average of the feedbacks to a majority rule approach. The implementation of the aggregation function is dependent on the application context and on the desired output.

The aggregated reputation value must be presented in a way that is useful to the consumers of this information. The aggregation service may output a binary value (trusted or not trusted), values on a discrete scale (say from 1 to 5 or  $[-1, 0, 1]$ ) or on a continuous scale (0 to 1). Along with the feedback, the nodes may send to the aggregating service node a second value that describe the confidence they have in their reporting opinions or the importance of the transaction they are reporting [13]. This “quality value” is useful to weigh the feedback received.

### 5.4 Reputation dissemination

Once the reputation information has been aggregated to form a reputation value, it needs to be disseminated to other nodes. Once again, similar to reputation information collection, reputation dissemination can happen both proactively and reactively. In proactive dissemination, reputation values for all nodes are pushed out to all nodes in the network. No dissemination is necessary if all nodes compute the trust values of all other nodes as in EigenTrust [17]. In reactive dissemination, a node has to explicitly request the reputation value of another node which it receives in response. When a number of aggregators respond, as in the case of multiple designated agents, the node that receives these multiple responses must further operate on these responses to determine whether it will interact with the node in question. A node may elect to take a simple majority, that is, if a majority of designated agents call the node trustworthy then the node is assumed to be trustworthy. The node may also choose to weigh the responses with the confidence it has in the designated agents themselves.

The aggregation service may also attach a second value to the trust value it reports and send a tuple instead of a single value. The second value of the tuple may indicate the number of transactions on which the trust value is based or the importance of those transactions [13]. This can be useful to distinguish trust values that are based on very small number of transactions or on very minor transactions as trust values thus computed may not be an adequate indicator in mission-critical tasks. For example, on e-Bay, seller ratings also indicate the number of feedbacks the seller has received. A seller with lots of positive feedback and no negative feedback can be seen trustworthy than a seller with some positive and no negative feedback. A single reputation value cannot capture this information and the second “quality value” can capture the weight of the transactions on which the reputation is based. Using a quality value also protects the system from “milking” agents that build up their reputation by behaving honestly in many minor unimportant transactions and then behave dishonestly in a few large transactions. If all transactions have the same weight, an agent can successfully milk the system by choosing a few large transactions to behave dishonestly in. The concept of transaction quality has been discussed in literature [29, 13]. The disseminated reputation may also be timestamped so that old information is appropriately given less weight if desired.

When a node is interested in receiving a service it may be less interested in the trustworthiness of a particular node than a list of trusted node who can provide it with that service. In this case, the reputation

aggregation system may disseminate a list of nodes providing the requested service that have a trust value above a given threshold. The requesting node can then contact these nodes with a request for service. If the first node cannot be contacted or refuses to provide the service, the requesting node can go down the list till it finds a node that will provide the service.

## 6 BIONETS Architecture

The BIONETS architecture lacks any centralized administrative control. This makes BIONETS a good candidate for the application of a reputation management system. However, there are many aspects of the BIONETS design that make a straightforward application of existing reputation management architectures impossible. We now present these features of BIONETS that make conventional reputation management architectures inapplicable and make the task of reputation architecture design non-trivial.

Unlike existing systems that use reputation management, all nodes in BIONETS are not equal from the reputation management perspective. There are three main actors in terms of devices in the BIONETS networks: T-Nodes, U-Nodes and APs. Within T-Nodes themselves, at least five different classes have been identified in the BIONETS security architecture document [26]. Moreover, T-Nodes can only act as data sources and no direct T-Node-to-T-Node communication is possible in BIONETS networks. Hence, no reputation aggregation or dissemination is possible at T-Nodes and all such activities can only take place at U-Nodes as U-Nodes can act as both data sources and data consumers. Further, most T-Node classes cannot act on any reputation information that may be provided to them.

The main node types in BIONETS are as follows:

- **T-Nodes** are simple, inexpensive devices with sensing/identifying capabilities. T-Nodes act as an interface with the environment and are needed to provide context-awareness to BIONETS services. T-Nodes may be almost-passive devices such as TAGs or RFIDs. T-Nodes do not communicate among themselves but are just “read” by U-Nodes passing by. They present minimal capabilities in terms of processing/storage/communications.
- **U-Nodes** are complex, powerful devices. No stringent limitations on requirements are encompassed for U-Nodes. PDAs, laptops and smartphones represent examples of a U-Node. U-Nodes are carried around by users (hence they are “mobile”) and run services. They interact with the environment through T-Nodes, from which they gather information to run context-aware services. U-Nodes may communicate among themselves to exchange information, whether environmental data or service-specific code (in order to enable service evolution).
- **Access Points** are complex powerful devices that act as gateways with the IP world. APs do not run services but just enable BIONETS and IP-based services to talk to each other. The architecture and functionalities of APs will be dealt with during a later stage of the BIONETS project. It should be noted that APs typically have more computational power at their disposal than U-Nodes and T-Nodes, have an AC power source that allows them to stay on indefinitely (without needing to conserve battery power) and have a permanent connection to the Internet. Therefore, APs may be used to exchange information with some trusted third party to verify, update or get reputation information from the Internet.

In addition to these we can add **users** and **services** as entities in their own right that have security and trust implications. In this deliverable, we take the position that users can be closely identified with U-Nodes (User-nodes) and thus need not be treated as separate entities. On the other hand, services are an important separate entity and must be monitored for performance and behaviour. Services are capable of misbehaving in their own right, either through malicious intent of one or more providers of the service or through as yet unknown emergent behaviours that result from adaptive service creation and evolution. We shall come to the treatment of services in trust and reputation management later in the deliverable.

## 6.1 Disconnected Operation

One of the salient features of BIONETS networks is their ability to operate as disconnected clouds of nodes that are unable to communicate with any entity outside the cloud for extended periods of time. For example, in a wireless ad hoc network, when a group of nodes moves too far away from the rest of the nodes they are disconnected from the main network even though they are connected to each other.

Disconnected operation poses certain problems for the reputation management system. Nodes are unable to access reputation information about other nodes in the cloud if this information is stored outside the cloud. Reputation information already in the cloud can be used as normal. Disconnection may also affect provisioning of services thus resulting in negative feedback for a node that was supposed to be providing the service but is unable to do so now due to disconnection.

It is important to differentiate between negative feedback due to disconnection and negative feedback due to misbehaviour or else nodes that are disconnected will be wrongly identified as misbehaving nodes. It may not be possible to make this differentiation in all cases. For example, if a misbehaving node simply switches off its transmitting radio when it is expected to perform a service, neighbouring nodes cannot tell if the disconnection is intentional or not. However, if a node is selectively transmitting only some data - say it transmits its own packets but not those of others - it is possible to identify that the node is in fact misbehaving. Nodes may run a service that can heuristically determine which neighbours are switched off and which neighbours are not. This service can gather relevant data (such as signal strength) from all the neighbouring nodes of a node to determine whether the node has disconnected or not. The detailed description of such a service is outside the scope of this deliverable. In certain cases, it may also be possible for a node to explicitly tell its neighbours that it is going to disconnect for example when it is moving out of range or when it is about to run out of battery power.

During disconnection, nodes may not have access to the agents that are used normally to store feedback. Hence, nodes must store reputation information and feedback for all transactions locally until the cloud connects with the main network or another cloud again at which point the reputation information thus stored is propagated. There may also be a lack of synchronization in the reputation information stored at two sites if one of them becomes disconnected from the rest of the network. There will be a mechanism to synchronize this information upon reconnection.

## 6.2 Mobility

Another feature of BIONETS is the mobility of its nodes. Nodes may come together and form temporary coalitions to do useful tasks or provide each other services. When the nodes move out of range of each other

the coalitions cease to exist and new coalitions may be formed. In this scenario, we need to determine the extent to which it is possible to perform reputation management when the interactions are very short-lived. Even when it is possible to form reputation values, it may not be feasible to create a reputation management system in such an environment due to the overhead costs being too large. There are likely to be many scenarios when setting up a trust/reputation system is not feasible.

Thus, before setting up a reputation management system in a transient cloud a decision needs to be made on whether the overhead justifies the possible benefit. This decision may be static - i.e., it is decided *a priori* what types of “node clouds” and services will get a reputation infrastructure and what don’t or this decision may be taken on-the-fly for certain types of node clouds and services where it is not possible to decide this *a priori*.

For instance, a T-Node may move from within the range of one U-Node to another U-Node making the original U-Node an unsuitable point for further reputation management tasks pertaining to that T-Node. A T-Node may also be within the range of more than one U-Node simultaneously. In this case, both U-Nodes will store reputation information on that T-Node based on their interactions with it. If the interactions of both the U-nodes with the T-node are of a similar nature, it may be useful for the two U-nodes to exchange information on the T-node’s reputation and keep their reputation data on that particular T-node synchronized.

## 7 Reputation System Definition in BIONETS

The unique BIONETS architecture (see Section 6) and the features of the BIONETS systems presented in Sections 6.1 and 6.2 impose the definition of specific functions to define a suitable reputation management system. Moreover, the presence of heterogeneous entities and the communication constrains, locality and U-Node/T-Node interaction, impose the definition of new reputation types for these elements which are different from classical peer-to-peer systems. In addition, the BIONETS system is composed by services that must be considered an important separate entity and must be monitored for performance and behaviour.

Table 2 highlights new concepts introduced in the BIONETS system and the consideration made to design a reputation management system suitable for BIONETS. These new concepts make the use of reputation management systems already in literature impractical and impose the implementation of new functionalities. In this context we define proper communication model for gathering information from nodes and disseminate trust values in the system. The next sections consider the general properties (presented in Section 5) a reputation management system should implement and analyze them in the context of BIONETS by presenting solutions concepts and definitions which are proper of BIONETS.

In a reputation management system the reputation information needs to be 1) collected from the feedback providers, 2) aggregated to form a useful measure of trustworthiness and 3) disseminated to members requesting the reputation value of a particular node. Hence, a reputation management system needs to implement three distinct functions and the BIONETS trust and reputation management system is no exception. These functions may either be provided by three separate services or may be implemented by a single unified reputation service that takes care of gathering, aggregating and disseminating reputation information. In BIONETS, we choose the former alternative of having separate services performing each of the functions.

New concepts	BIONETS system constrains	Approach
Heterogeneous entities	Some T-nodes cannot act on reputation information <hr/> No reputation aggregation at T-Nodes <hr/> No reputation dissemination at T-Nodes	Different types of reputation
Services	Emergent behaviour from service adaptation and evolution	Reputation value defined for services
Communication paradigm	Disconnected operations <hr/> Mobility <hr/> T-Node/T-Node communication is not possible <hr/> APs are not always present	Distributed computation of reputation <hr/> Hybrid approach for collecting feedback <hr/> Synchronization upon reconnection <hr/> Locality for reputation

Table 2: BIONETS reputation system relevant features

## 7.1 Collection of Feedback

This function deals with the collection of information that pertains to the behaviour of a node in all of its previous interactions. This is essential as in BIONETS the trustworthiness of a node is solely dependant on how a node has behaved in the past. The gathered information represents the input to the reputation aggregation function and can be collected in a proactive, reactive or hybrid way, as discussed in Section 5.2).

If the information is collected reactively, as demonstrated in Fig. 1, a node will only send its feedback when it is requested to do so. In the BIONETS framework, nodes interact in the “island of connected” nodes and the designated agent periodically polls the other nodes to gather information on their transactions. The effectiveness of this approach is based on the frequency the designated agents query the other nodes. If the frequency is too high, nodes might have few interactions to report and the communication overhead introduced by the reputation management system is not compensated by an accurate measure of the reputation value. On the contrary, if the frequency is too low, this approach might suffer the high mobility of U-Nodes who might move outside the “island”. This approach has also the advantage of not triggering any report actions after each interaction at the cost of detection delay in case of nodes’ misbehaviour if the information is not immediately reported.

If reputation information gathering is done proactively (see Fig. 2), a node will send the feedback on its transaction partners after every transaction. This approach enables fast detection of nodes’ misbehaviour but may incur additional network overhead in BIONETS where nodes are battery powered and the communication might be difficult.

In BIONETS, we propose using a hybrid approach in which the reputation gathering service periodically polls all nodes to collect feedback on all of their transactions since the previous polling round. Additionally, a node may decide to send reputation information proactively if it feels that the feedback it has to send is time-critical and must be disseminated immediately.

It is undesirable for the reputation system if a node does not report feedback on transaction it takes part

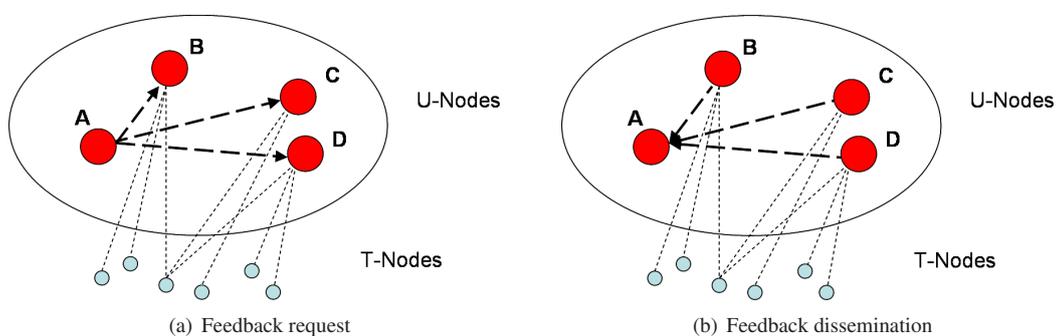


Figure 1: Reactive feedback collection. Nodes interact inside the island of connected nodes and U-Node A - designated node - periodically request feedbacks from other U-Nodes (a). Upon request, U-Nodes respond to U-Node A that is designated to collect and aggregate feedbacks (b)

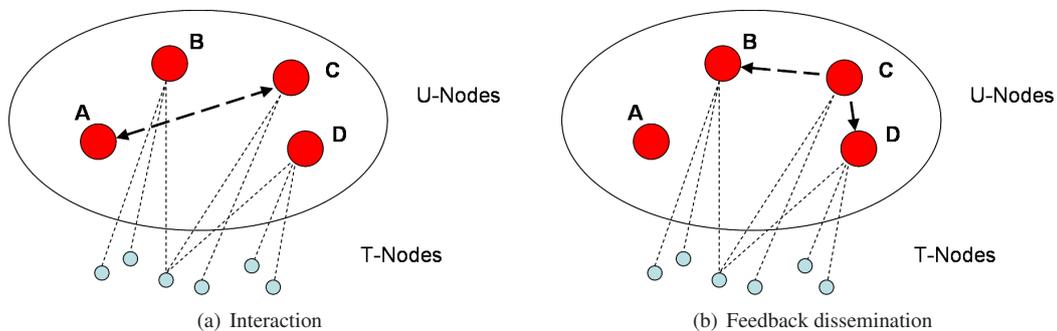


Figure 2: Proactive feedback collection. U-Node A and U-Node C interact (a) and U-Node C sends feedback to designated nodes in the same island of connected nodes (b)

in. Note that this is possible in both proactive and reactive scenarios as in a reactive scenario a node may simply ignore a request for feedback. If some nodes do not report their feedback, the reputation system has less information for aggregation and the computed trust values may be inaccurate.

Similarly a node may also report false feedback even when a transaction has not taken place. When the gathering service receives a report from only one participant of a transaction, it cannot tell if the reporting node has constructed false feedback or if the other node is not reporting feedback. To avoid this situation, the gathering service needs to keep track of all transactions that are taking place. During each polling round it can then check the feedback received from each node against the transactions it took part in and flag non-providers of feedback or providers of false feedback.

Hence, it is necessary for the reputation gathering service to have a record of all transactions. The participants in a transaction can agree on a string describing the transaction. The string can be simple and contain the IDs of the transacting partners and a timestamp. This string can then be digitally signed by both partners and submitted to the gathering service which will acknowledge receipt and give the nodes the go-ahead for the transaction. This task clearly requires additional resources and introduces additional delay in a transaction. Moreover, in the absence of a certification authority it may be impossible to verify the digital signatures on the transaction string. Therefore, this transaction approval step must be optional and can be skipped in resource constrained environments or when the transaction is time-critical. The participating nodes must agree to this when negotiating the terms of the transaction. They may also agree to send an unsigned report informing the gathering service that the transaction has taken place. In this case, the additional resources required are much smaller and there is no delay in the transaction.

## 7.2 Reputation Aggregation

Once reputation information is collected from feedback providers, it needs to be aggregated to form a useful measure of the trustworthiness of a node. In BIONETS the reputation computation service performs this function. This service runs on the reputation aggregators which may be U-Nodes or AP-nodes depending on the type of reputation information that has been collected. Details on the types of reputation information that are managed by the BIONETS trust and reputation system are given in Section 7.4.

In BIONETS, we use designated agents but these agents are not chosen randomly from all the nodes in the network. This is because not all nodes are capable of performing the functions associated with reputation management. Most T-Node classes do not have the storage and processing resources to perform these functions. So only U-Nodes (and possibly AP-nodes) will serve as designated agents. During normal, connected operation, all the U-nodes will form a DHT-based overlay. The designated agents will be chosen from among them by hashing the identifier of the node whose reputation must be stored. Multiple designated agents can be chosen by using multiple hash-functions [13].

The formation of short-lived networks and the lack of a guarantee of connectivity to other parts of the network mean that a conventional DHT architecture is not sufficient to store and disseminate reputation information in a distributed fashion. U-Nodes must opportunistically query other U-Nodes for information and cache reputation data that they think may be relevant in a future disconnected period. This is only feasible if U-Nodes have advance notice of impending disconnection. A service therefore needs to be implemented that will heuristically decide when a disconnection is imminent and thus cache reputation

information.

It may not be possible to choose designated agents purely randomly all the time even from all the U-Nodes. During disconnected operation, there may be a very small number (possibly one) U-Node in a disconnected cloud so that node automatically assumes the responsibility for being the designated agent. Finally, the choice of designated agents also depends on the type of reputation being aggregated.

### 7.3 Dissemination of Trust Information

The dissemination of trust information is the last step of a reputation management system in the framework of the communication model. The approach is similar to the collection of the feedback as in BIONETS we envision the presence of designated agents who aggregate, store and disseminate information. To cope with network resilience, multiple designated agents for the same node are present in the system. Thus, the retrieved reputation values must be aggregated to be useful. Like for the aggregation of feedback, a designated agent can send a tuple containing the reputation value itself and a value representing the confidence this node has in the disseminated reputation. This confidence value can be useful to distinguish between a reputation value derived from few or dissenting received feedbacks.

The disseminated reputation may also be timestamped so that old information is appropriately given less weight if desired. This is particularly important in the case of disconnected operations as a “networked island” has no new sources of information about nodes outside the island. When the island regains connectivity, it should not be relied on for information about nodes outside the island. Timestamps are also important when U-Nodes have storage constraints. When a U-Node finds that it cannot store any more reputation information, it can use these timestamps to determine the oldest information and discard it to make space for new information.

### 7.4 Types of Reputation

The BIONETS architecture envisages T-Nodes that may be severely limited in their storage and processing power. As a result, it has been decided that T-Nodes should not be used to store or aggregate any trust or reputation data. It is therefore reasonable to assume that T-Nodes cannot detect misbehaviour. U-Nodes will have to detect misbehaviour in T-Nodes as well as in other U-Nodes. Thus all reputation information for T-Nodes and for U-Nodes will be stored, aggregated and disseminated using U-Nodes only.

U-Nodes will be used to store several kinds of reputation information. They will store reputation information about other U-Nodes and about T-Nodes that they have been in contact with. These types of information need to be qualitatively different as U and T capabilities differ and the types of misbehaviour they can indulge in also differ significantly. In addition, U-Nodes will store information on the credibility of other U-Nodes, i.e. the reliability of the reputation information they provide. Finally, U-Nodes will also store reputation information about services.

AP-nodes may also store reputation information. We assume that AP nodes are under direct administrative control of the network and can thus be assumed to be trustworthy. For instance, strong security mechanisms, which allow for the authentication and connectivity of APs to a centralized security architecture, such as PKI, can be in place. Thus, it is not necessary to calculate the trustworthiness of an AP-node. At this point we also assume that AP-nodes do not need to compute the trustworthiness of any node in the

network since their role is primarily that of a resource provider and not resource consumer. AP-nodes also perform a mission-critical task of providing connectivity to the fixed network and it is undesirable to place them under excessive load by having them compute reputation values.

Hence, we envision AP-nodes to be the holders of reputation information on behalf of other U-Nodes. AP-nodes have far greater resources in terms of network bandwidth and battery power than U-Nodes and can hold and disseminate a lot more information. Hence, AP-nodes may be called upon to disseminate reputation values on behalf of U-Nodes. Moreover, AP-nodes are not mobile and thus offer a fixed point of reference for mobile U-Nodes. Note that AP-nodes are by no means necessary for the reputation management system to function. AP-nodes are supporting entities for U-nodes and can be called upon to hold and disseminate reputation information when available. However, this is a purely optional function envisioned to maximize efficiency. In the absence of AP-nodes, the U-nodes are capable of managing reputation by themselves.

U-Nodes may contain upto four different classes of reputation information. These are:

- **Trust Information on T-Nodes ( $T_t$ ):** Typically a U-Node will contain reputation information on the T-Nodes that are in its vicinity. The U-Node is responsible for data collection and performance monitoring of all these T-Nodes. It is envisioned that a T-Node may simultaneously be within the range of more than one U-Node, say, providing different types of data to the different U-Nodes. In this case, both U-Nodes will calculate trust information for that T-Node and they may exchange this information periodically to maintain synchronization.

T-Node reputations have a **local scope** as they are only valid for the use by the U-Nodes that are in its vicinity. While it is not strictly necessary for T-Nodes to have globally unique identifiers from the point of view of the reputation management system, an absence of unique T-Node identifiers will make it difficult, if not impossible, for U-Nodes to exchange information about T-Nodes. Moreover, T-Node reputations are relatively short-lived as the same T-Node is unlikely to be within the range of the same U-Node continuously for a very long time.

BIONETS defines 5 different classes of T-Nodes. The security capabilities differ for each class [22]. Thus, the reputation management system is applied differentially for each class.

- Class 0 T-Node: No reputation management is possible.
- Class 1 T-Node: Only proximate collection possible. i.e. reputation information should be collected and used locally, i.e. for transient systems. Long-term storage and forwarding of reputation information is not recommended.
- Class 2 and above: Support both transient and long-term reputations.

It should be noted that the decision on which reputation management system is going to be employed currently depends on the class of the T-Node in question. But it is intended that this decision will be taken dynamically in the future, depending on the T-Node characteristics which will be communicated dynamically.

- **Trust Information on U-Nodes ( $T_u$ ):** When a U-Node comes into contact with other U-Nodes, it will measure the trustworthiness of the other U-Node. It may store this information for future use

and it may also forward this information to other U-Nodes. U-Node reputations mirror traditional reputation management systems the closest. U-Node reputations are long-term and have a **global scope**. U-Nodes are similar in terms of node capabilities so it is not unreasonable for the reputation management system to treat them similarly. Finally, U-Nodes are capable of collecting, aggregating and disseminating reputation information and are therefore like peers in peer-to-peer networks for the purposes of reputation management.

- **Second-order Trust Information on U-Nodes ( $C$ ):** A U-Node may need to use reputation information that has been collected and aggregated by another U-Node. For example, when a T-Node passes from the influence domain of one U-Node to another U-Node, the second U-Node may request the trust value of the T-Node from the first U-Node. This information can then be used by the second U-Node in a number of ways. For example, the U-Node can use this information to adapt the security mechanisms involved in the interaction process.

In this scenario a U-Node cannot take the information provided by the other U-Node at face value. It is possible that the first U-Node is presenting falsified information about the T-Node in order to further its own goals. Hence each U-Node maintains a credibility rating (or second-order trust) for all U-Nodes from whom it has received information about other nodes. When it receives any reports from these nodes, it weighs the feedback according to the reporting node's second-order rating.

- **Trust Information on Services ( $T_s$ ):** A U-Node (or a BIONETS network as a whole) will also maintain reputation information about the services that exist in the system. A misbehaving service or a service that does not honour its commitments can thus be identified and other nodes be informed so that they do not get cheated by an untrustworthy service. It is important to guard against false information as a malicious U-Node may give incorrect feedback about a service in order to cheat other U-Nodes either by giving them the false impression that a service is unreliable or by making it appear that the U-Nodes providing that service are unreliable. Moreover, a U-Node may also advertize a malicious service (by rating it high) so that more nodes contact that service to get attacked.

Services may require a different algorithm to calculate their reputation as opposed to nodes. A service will likely be provided by one or more nodes and the identity of these nodes may change as the service evolves and grows or due to service load balancing and optimization. The trustworthiness of a service can be initialized to a value based on the reputations of the node(s) that initially provide the service. However, as the service evolves and migrates to other nodes this initial value is no longer a good indicator of the service reputation. Hence, after the initialization of the service trust value - which is necessary to allow the service to find consumers initially - the service trust value should be calculated using the feedback received on the service. The providers of this feedback may be U-Nodes or other services. This feedback can be collected, aggregated and disseminated in the same way as the node reputations are managed.

The reputation of the service should not affect the reputation of the nodes that provide this service. This is because services are composed dynamically and evolve in unpredictable ways. That is to say, services have their own emergent behaviours. Hence, it is not appropriate for service reputations to influence node reputations. The feedback provided for a service is highly application dependent.

Parameter	Description
$T$	Computed trust value
$W(T)$	Age-weight of trust value $T$
$TS$	Timestamp of trust value of feedback
$F$	Feedback from transaction participant
$\alpha$	Aging constant

Table 3: Trust Parameters

Hence, the service description must also contain information that describes the type of data and attributes the service carries and how the feedback to the service can be described.

Services in BIONETS may either be short-lived or be longer-lasting. Service trust information ( $T_s$ ) will have the same life-span and scope that the service itself has. Once a service becomes extinct, there is no need to store reputation information about that service.

One of the goals of BIONETS is to allow evolution of services in an automatic need-based fashion. The BIONETS trust management architecture may also be harnessed to keep track of service utilization and service shortages in order to assist this evolution.

## 7.5 Reputation Aggregation Algorithms

All trust values in BIONETS whether they are trust values of U-Nodes, T-Nodes, services or second-order trust information about U-Nodes can be represented as a triplet. This triplet,  $(T, W(T), TS)$ , contains the trust value being disseminated  $T$ , the “weight” of the trust value  $W(T)$  and the timestamp  $TS$  which denotes when the trust value was computed.  $T$  is a value between 0 and 1 that signifies the trustworthiness of a node. A value of 0 means a node is not trustworthy and a value of 1 means a node is completely trustworthy.  $W(T)$  is the “weight” of the trust value which is essential for feedback aggregation as described below. A higher weight usually denotes that there have been recent and frequent updates to the trust value which make the trust value a more accurate representation of the trustworthiness of the node. Low weight indicates that the trust information was old at the time of aggregation.

The trust triplet contains the full information that is needed to update the trust value with additional feedback. Feedback is represented as a tuple  $(F, TS)$  where  $F$  is the feedback being reported, a number between 0 and 1 and  $TS$  is the timestamp for the feedback. Each transaction a U-Node has with a T-Node, service or another U-Node, results in the formation of an opinion on the trustworthiness of its transaction partner. This is then sent as feedback along with a timestamp that indicates when the transaction took place. Additionally, both the trust triplet and the feedback tuple must be encapsulated with an id of the node or service to which the trust value or feedback pertains.

### 7.5.1 Age-Weighting of Trust Information

An important feature of the BIONETS reputation management system is the age-weighting of all trust information including both the reported feedback and the aggregated trust values. The motivation behind this age-weighting is to ensure that stale reputation information is aged appropriately and given lower weight as it becomes older. In order to do this, all trust information is given a weight depending on its age. If the

information is  $t$  time units old, then the weight of that information is computed as  $e^{-\alpha t}$  where  $\alpha$  the aging constant denotes the rate at which information ages. Each instance of the reputation management system can choose an  $\alpha$  that is appropriate for the conditions of the network. Note that when the information is 0 time units old, i.e. the information is current, its weight is 1. A value of  $1/\text{hour}$  for  $\alpha$  would mean that the weight of a piece of information decays by approximately 63% per hour since  $e^{-1}$  is 0.3678. The value of  $\alpha$  will depend on the frequency of interactions in the BIONETS reputation management system. If transactions are infrequent, a low value of  $\alpha$  is desirable whereas when transactions are frequent, a high value is desirable.

When a node aggregates feedback to form a trust value it takes the age of the feedback into account. Hence, if a U-Node aggregates 3 pieces of feedback about a T-Node, say  $F_1$ ,  $F_2$  and  $F_3$  whose timestamps indicate that they are  $t_1$ ,  $t_2$  and  $t_3$  timeunits old, the trust value will be computed as follows:

$$T = \frac{F_1 e^{-\alpha t_1} + F_2 e^{-\alpha t_2} + F_3 e^{-\alpha t_3}}{e^{-\alpha t_1} + e^{-\alpha t_2} + e^{-\alpha t_3}} \quad (1)$$

The U-Node will store the trust tuple for the trust value where  $T$  will be the computed trust value,  $W(T)$  will be the denominator of the right hand side in the above equation and  $TS$  will be the time when the calculation was made. Storing  $W(T)$  has two functions. Along with the timestamp  $TS$ , it gives an indication of how old is the trust information when it was aggregated. In addition, it makes the incorporation of subsequent feedback into the trust value  $T$  very easy to compute.

**T-Node Trust Information:** A U-Node may store trust information on T-Nodes it used to be in communication with but is no longer communicating. This is done for two purposes 1) The U-Node may encounter the same T-Node again and 2) The U-Node can pass this information to other U-Nodes that may want to interact with that T-Node.

When a U-Node comes into contact with a T-Node it has never contacted before, it may request reputation information about the T-Node from another U-Node if the T-Node was in communication with that U-Node in the past. Hence, the trustworthiness of a T-Node may be computed using both first-hand information based on direct experience and on the basis of indirect feedback provided by other U-Nodes.

**U-Node Trust Information:** A U-Node will likely provide data or services to other U-Nodes so it is necessary for U-Nodes to know the reputation of other U-Nodes they are interacting with. The trustworthiness of a U-Node is always computed by combining feedback from U-Nodes it has interacted with. After each interaction with a given U-Node, that U-Node's transaction partners will send feedback to the designated agents for that U-Node. These agents are computed using distributed hashing on the global identifiers of the U-Node as described previously.

Since designated agents aggregate feedback from a number of different U-Nodes, they need to incorporate the credibility or second order trust values of the reporting U-Node in the aggregation algorithm. This is because, nodes may lie about the behaviour of other nodes in order to harm competitors or to mask their own misbehaviour. Hence, feedback should not be taken at face value and should be weighted according to the likelihood that a node will give false feedback.

Let a designated agent receive  $n$  feedbacks  $F_1, F_2 \dots F_n$  from  $n$  different U-Nodes that have credibility values  $C_1, C_2 \dots C_n$  with the feedback timestamps indicating that they are  $t_1, t_2 \dots t_n$  time units old. Then, the trust value will be computed as follows:

$$T = \frac{\sum_{i=1}^n F_i * C_i * e^{-\alpha t_i}}{\sum_{i=1}^n C_i * e^{-\alpha t_i}} \quad (2)$$

As before  $W(T)$  will be the denominator of the right hand side above.

When a U-Node wishes to know the trust value of another U-Node, it will send a message to all the designated agents of that U-Node requesting the trust value. It will then compute a weighted average of the responses weighting the trust values received with the corresponding age-weights ( $W(T)$ ) and the second-order trust values of the designated agents.

**Service Trust Information:** The trust values of services are based upon feedback received from U-Nodes and other services. The computation of service trust value proceeds in a manner that is analogous to the computation of U-Node trust values. Note that services will have globally unique identifiers which can be hashed to compute the designated agents for those services.

**Second Order U-Node Trust Information:** A U-Node performs several roles as not only it does provide and consume services, share resources with other U-Nodes, it also collects information from T-Nodes and forwards it to other U-Nodes after processing it. A U-Node also acts as a designated agent for storing trust values of other U-Nodes and services. It is therefore possible for a misbehaving U-Node to send falsified trust information about T-Nodes in its range or about U-Nodes and services for which it acts as a designated agent. If left undetected, this behaviour can have a compounded negative impact as it may cause U-Nodes to compute the incorrect trust values for other T-Nodes, U-Nodes and services.

Hence, U-Nodes must compute the second-order trust information for all U-Nodes they are interacting with. Second-order trust information is not shared with other nodes and is computed on a direct-experience basis only. The credibility of all nodes is set at 0.5 by default. Nodes can increase or decrease their credibility by reporting back accurate or inaccurate information. In BIONETS, there is no canonical information so the accuracy of reported information is estimated by comparing it with what other nodes report. A number of techniques exist that can be used to perform this calculation such as the Pearson correlation coefficient [23] or by using vector similarity like in Breese et al. [6].

## 8 Applications of Reputation Management in BIONETS

Reputation management is not a replacement of traditional security solutions and is instead a complementary strategy that works through establishing trust between participants of an autonomic system allowing them to collaborate so that they can provide each other with services that would otherwise not have been possible.

Reputation management systems can be used for the continual monitoring of U-Nodes and the evolution of services they provide. The BIONETS system will rely on the reputation management system for the collection and dissemination of trust information gathered during transactions between U-Nodes, between U-Nodes and T-Nodes and during service provision. This information can be used by the U-Nodes to build knowledge about the behaviour of other nodes/services including those with whom they have never interacted before.

Interacting and coordinating with other components is essential to the achievement of a component's individual and system goals. Through reputation management these interactions and coordination activities

can be mediated to further the overall goals of the system. The reputation of a component serves as a signal to other components and enables them to decide the level of cooperation they extend to the component.

As we have mentioned, the main uses of digital reputation management systems in autonomic systems are:

1. Identifying and excluding malicious entities
2. Incentivizing cooperation

Resnick [24] further defines three requirements for a reputation system: 1) to help people decide whom to trust, 2) to encourage trustworthy behaviour and appropriate effort and 3) to deter participation by those who are unskilled or dishonest. To this we can add the requirements that a reputation system 4) must preserve anonymity associating a peer's reputation with an opaque identifier and 5) have minimal overhead in terms of computation, storage and infrastructure.

The more acute threat to an autonomic system comes from participants who act in a malicious fashion with the intention of disrupting the system. In the framework of BIONETS, examples of such malicious behaviour include T-Nodes who propagate false data or U-Nodes who modify the service execution, nodes that refuse to collaborate and provide services, U-Nodes who hinder the creation of the "islands of connected devices". These types of behavioural attacks historically can be derived from peer-to-peer (P2P) systems and mobile ad hoc networks (MANET) as nodes tend to be selfish or to take control of the system itself. Thus, the lesson learned from the applicability of reputation management techniques to these systems is of help to understand the effect of the application of incentives mechanisms to the BIONETS environment.

The problem of motivating cooperation in P2P and MANETs is centred on the selfish and malicious behaviour of nodes who do not forward packets or introduce corrupted content such as in content distribution networks. The prevention of such attacks is particularly important as nodes (i) in the former case consume resources from the system without offering any resources of their own leading to the system being starved of resources or (ii) in the latter case the dissemination of services/data is compromised by corrupted content leading to the content being useless.

The solutions implemented to preserve the system, that has been presented in Section 4 cannot be applied in the BIONETS network as the assumptions and the system goals are different. Thus, we present in this section the applicability of reputation systems to the two-tier communication architecture envisioned in BIONETS (see Fig. 3) with a close focus on services.

Given the constraints imposed by the BIONETS network architecture, T-Nodes cannot communicate directly to each other and the only destination of a T-Node communication is a U-Node that is in close proximity. In this context, as T-Nodes are passive elements or data sources, they can provide inconsistent or incorrect data that can infer false information in the U-Node and reduce the accuracy of the service. The general idea behind the application of the reputation mechanism in T-Nodes/U-Nodes communication is based on the application of statistical analysis of data received from multiple T-Nodes in the proximity area. This analysis provides intrinsically a preliminary framework to evaluate the trustworthiness of the data a T-Node offers. Thus, the reputation value has only a local meaning and it is useful to filter the information sensed by T-Nodes.

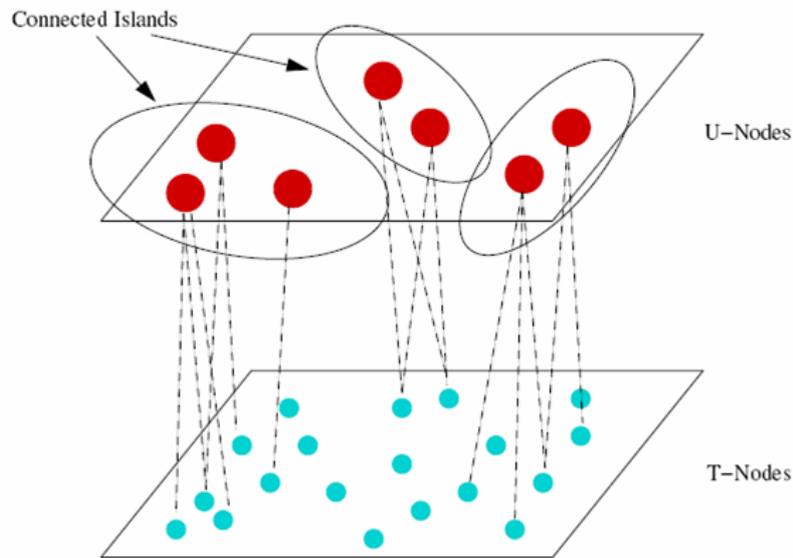


Figure 3: The two-tier BIONETS network architecture [22].

The enforcement of *cooperation strategies* acquires more importance in the communication paradigm envisioned between U-Nodes. U-Nodes are mobile entities that are both data sources and data consumers. Their peer role in the BIONETS system paves the way for the application of a trust framework. U-Nodes interact with other U-Nodes to exchange information on services and to exploit other nodes capabilities. During the message exchange U-Nodes can collect and disseminate their opinions on services, nodes and data. This mechanism can be exploited to provide feedback to U-Nodes and hence allowing them to compute the reputation estimates that are calculated on the past behaviour.

By considering a communication system characterized by islands of connected U-Nodes, or virtual community, we can assume that cooperation among nodes in a cloud is important to enable the connected component to survive and to work toward the system overall goal. Reputation management is an invaluable tool to identify faulty or malicious U-Nodes: A U-Node may become faulty or may come under the control of a malicious user who wishes to disrupt the network. The reputation value is an assessment of their cooperation level in the system. Malicious nodes can compromise the equilibrium reached inside the U-Nodes community as they can either use all resources available or they can introduce faulty information to take control of the “island”.

The use of a reputation framework does not enable only soft-protection from malicious nodes, who might overtake control of the system or exploit services and resources without contributing to the system goal, but it is also an important process to regulate the adaptation and evolution process of services. Adaptation to the environment and service evolution are driven by the feedback mechanism that is proper of a biological system. Reaction and mutation are mechanisms activated in response to new and different context information that are extracted and obtained during the dissemination process. The protection of such information with cryptographic tools might not be suitable for a system that is evolving fast and that is constituted by nodes with high mobility. Thus, reputation can be applied to filter context information based on the trustworthiness of the data and on the nodes providing such data.

Furthermore, context information and feedbacks have a cost for the nodes that generate the data and those that disseminate the knowledge in the system. Thus, nodes need incentives to participate in the feedback process. If this information is “sold” to other U-Nodes then the reputation of the providing U-Node can serve as an indicator of the price this information will command. This has the added benefit of encouraging U-Nodes to try and maintain a high reputation to maximize the returns for the information they provide.

To illustrate better the application of reputation during service evolution, let’s assume that a service is generated in a U-Node. This service will be propagated and disseminate in the system to other U-Nodes. The service will evolve to add new functionalities or simply adapt to the new environment or the context of the communication. If the new evolved service turns to be malicious the reputation system should be able to detect this malicious behaviour and eliminate the faulty service from the system.

## 9 Threat Analysis

This section discusses the attacks that can be made targeting the BIONETS reputation management architecture. The objective of this section is not necessarily to provide solutions to such attacks but instead to highlight attack scenarios so that the overall security work package can work towards minimizing the danger from such attacks.

### 9.1 Identity and Trust

The BIONETS two-tier communication system enables U-Nodes to make use of the resources of T-Nodes in close proximity. The mobility of U-Nodes, the ad-hoc formation of islands of connected nodes and the huge number of T-Nodes introduce in BIONETS several issues concerning addressing schemes or identity management [22]. In particular, the choice of an identification scheme reflects in particular the reputation system as nodes must be identified by a key, address or any other unique name so that the collection of the historical behaviour of a node and the estimation of its reputation can be accomplished in a correct and consistent manner.

If a proper identification scheme is not in place, nodes can easily forge their identities and appear in the network with a temporary identifier with the intention of disrupting the network or disseminate false opinions without being responsible for their actions. A node may also create a large number of identities and use these identities to attack the network. This attack is called the “Sybil” attack [9]. Its consequences have been identified and discussed at length in the literature of peer-to-peer networks.

Another attack related to identifiers is the whitewashing attack in which a peer behaves badly. When its reputation value becomes very low it exits from the system and enters again using a new identity. Other peers do not have any reputation information about this “new” peer and thus the peer can behave badly again till its reputation value becomes low again.

Another consequence of a flexible identification scheme and nodes’ mobility is that malicious nodes can impersonate other nodes and disseminate false information or bad services to keep the reputation value of honest nodes low. Alternatively, the nodes can consume resources from the system pretending to be another node.

In BIONETS we cannot expect to have unique and global identifiers for T-Nodes due to scalability issues. However, global identifiers for T-Nodes can be still in place if security reasons require a specific mechanism to address uniquely such nodes in the system, as discussed in [22]. For the sake of an extensive analysis of possible threats in BIONETS, we can assume that T-Nodes have local scope identifiers as their role of data providers has a local meaning. Thus, we can expect that their reputation value can be also local. This locality introduces several constraints in the reputation system and opens the door to several attacks that must be addressed. However, in this document we do not deal with identifiers and we limit our analysis to the reputation system.

The local scope of the reputation value for T-Nodes is a good solution if the U-Nodes utilizing the services of T-Nodes are in the same local scope<sup>1</sup>. However, as the connected components dynamically change membership, T-Nodes might exhibit different behaviours during their existence and escape being punished for their malicious behaviour. On the other hand, the reputation value of a T-Node might be highly dependent on the capability of a U-Node to judge correctly the service/data provided by the T-Node itself. Statistical methods or the credibility of the U-Node are useful techniques to weigh the T-Nodes reputation, but they do not help if the T-Node reputation is estimated by only one malicious U-Node.

These threats that are behaviourally dependent have limited scope and do not pose serious damage to the system as the data provided by the T-Nodes are filtered based on the context of the information such as proximity [22]. However, services are more exposed to “Sybil attacks” if there is no valid scheme to address them uniquely. A malicious node can create a very large number of services with false identifiers and create denial of service. Services might also evolve and change their identifier so that the reputation value associated to a service can be whitewashed and they can join the system as new services. Finally, since services themselves are evolving, the trust value of a service has limited applicability as the service may have completely changed its characteristics since the trust value was computed.

## 9.2 Feedback and Reporting of Opinions

The reputation system bases its efficacy on estimating node behaviour through the reports or feedbacks from interactions. The feedback collection and trust dissemination mechanisms must deal with disconnected operations inside BIONETS. Mobility and temporary disconnections increase the difficulties of disseminating feedback as well as trust information.

Another threat comes from nodes that report incorrect feedback for their transaction partners. This results in the wrong trust values being computed. Existing reputation management systems use second-order reputations to evaluate the credibility of a node as a reporter of opinions. However, we still require traditional security mechanisms, that provide at least integrity, to protect opinions from modification while in transit and a non-repudiation scheme to ensure that nodes are kept responsible for their actions.

## 9.3 Collusions in an *island*

The creation of islands of connected nodes defines a new paradigm for the communication which exploits nodes proximity. Nodes interact inside the community predominantly and form opinions about other nodes. In this context, malicious nodes might collude in order to cause damage to a node in terms of services and

---

<sup>1</sup>No direct communication between T-Nodes is possible. T-Nodes interact only with U-Nodes

by sending bad opinions about the node itself. Since, the honest node does not have access to the outside world, it is much harder to identify a malicious collective.

The scope of this attack is limited to the islands of connected nodes. However, it can have severe consequences if false reputation values of nodes are transferred to other nodes once the cloud reconnects with the outside world. Moreover, colluding nodes can work to increase the reputation value of a malicious node.

## 9.4 Inconsistent behaviour

Another attack on the reputation system is through inconsistent behaviour of nodes that behave well for a period of time or in small transactions to increase their reputation value and then start behaving maliciously. This type of behaviour causes more problems in systems that give additional privileges to user with high reputation. In BIONETS, this attack has limited impact on T-Nodes as their services are provided locally. However, this attack might cause considerable damages in the system if it is related to services' reputation values. During its evolution and once it has gained a high reputation inside a community, the service might turn to be malicious and infect a large part of the system.

## 9.5 Denial of Service

Denial of Service attacks are typical of a client/server model where the attacker reserves or uses all resources available at the server to deny service to legitimate clients. This attack can seriously damage the reputation system in BIONETS as the key elements in the architecture are U-Nodes. They will store, collect and disseminate opinions, services and reputation values. If in a connected island U-Nodes are overloaded by service requests they cannot handle their job to report feedbacks thus limiting the consistency of the estimated reputation values.

To limit this attack, the reputation value should exploit the communication channel for service and data exchange to provide feedbacks with minimal overhead.

# 10 Conclusions

In this deliverable, we have defined the trust and reputation system that will be used in BIONETS. We have explored traditional reputation management approaches and examined their suitability in the BIONETS network. We have discussed the unique features of BIONETS such as high mobility and disconnected operation that make an application of an existing reputation management system impractical. With this in mind, we describe a reputation management system that uses three separate services to perform the tasks of collection, aggregation and dissemination of trust information.

The system handles node heterogeneity by describing four different kinds of reputation in BIONETS each of which is calculated using a different algorithm. Special emphasis is placed on the time-sensitivity of reputation information by incorporating mechanisms that age trust information. Trust values are age-weighted and the collection and dissemination protocols are defined to explicitly include timestamps and age-weighting information.

## References

- [1] Karl Aberer and Zoran Despotovic. Managing trust in a peer-2-peer information system. In *Proceedings on the Tenth International Conference on Information and Knowledge Management (CIKM-01)*, pages 310–317, New York, November 5-10 2001. ACM Press.
- [2] Amitanand S. Aiyer, Lorenzo Alvisi, Allen Clement, Michael Dahlin, and Jean-Philippe Martin and Carl Porth. Bar fault tolerance for cooperative services. In *20th ACM Symposium on Operating Systems Principles*, October 2005.
- [3] G.A. Akerlof. The market for ‘lemons’: Quality uncertainty and the market mechanism. *The Quarterly Journal of Economics*, 84(3):488–500, 1970. available at <http://ideas.repec.org/a/tpr/qjecon/v84y1970i3p488-500.html>.
- [4] C. Avery, P. Resnick, and R. Zeckhauser. The market for evaluations. *American Economic Review*, 89(3):564–584, 1999.
- [5] R. Axelrod. *The Evolution of Cooperation*. Basic Books, New York, 1984.
- [6] John Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Uncertainty in Artificial Intelligence. Proceedings of the Fourteenth Conference (1998)*, pages 43–52, San Francisco, 1998. Morgan Kaufman.
- [7] L. Cox and B. Noble. Samsara: Honor among thieves in peer-to-peer storage. In *Proceedings of the ACM Symposium on Operating Systems Principles*, October 2003.
- [8] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. Managing and sharing servants’ reputations in P2P systems. *IEEE Transactions on Data and Knowledge Engineering*, 15(4):840–854, July-Aug. 2003.
- [9] John Douceur. The Sybil Attack. In *Proceedings of the 1st International Peer To Peer Systems Workshop (IPTPS 2002)*, pages 251–260, Berlin/ Heidelberg, Germany, March 2002. Springer.
- [10] E. Fehr and S. Gächter. Altruistic punishment in humans. *Nature*, 415(6868):137–40, January 2002.
- [11] Anurag Garg and Roberto Battiti. *Digital Reputation Schemes for Virtual Communities*, chapter 85. CRC Press, 2006.
- [12] Anurag Garg, Roberto Battiti, and Roberto Cascella. Reputation management: Experiments on the Robustness of ROCQ. In *Proceedings of the 7th International Symposium on Autonomous Decentralized Systems (First International Workshop on Autonomic Communication for Evolvable Next Generation Networks)*, pages 725–730, April 2005.
- [13] Anurag Garg, Roberto Battiti, and Gianni Costanzi. Dynamic Self-management of Autonomic Systems: The Reputation, Quality and Credibility (RQC) scheme. In *The 1st IFIP TC6 WG6.6 International Workshop on Autonomic Communication (WAC 2004) (LNCS 3457)*, pages 165–176, Berlin/Heidelberg, Germany, October 2004. Springer.

- [14] H. Gintis, E.A. Smith, and S. Bowles. Costly signalling and cooperation. *Journal of Theoretical Biology*, 213:103–119, 2001.
- [15] M. Jackson. Mechanism theory. In *Optimization and Operations Research*, edited by Ulrich Derigs, in the *Encyclopaedia of Life Support Systems* [<http://www.eolss.net>]. EOLSS Publishers, Oxford UK, 2003.
- [16] Sham M. Kakade, Michael Kearns, Luis E. Ortiz, Robin Pemantle, and Siddharth Suri. Economic properties of social networks. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, Cambridge, MA, USA, 2005. MIT Press.
- [17] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the twelfth international conference on World Wide Web*, pages 640–651, New York, 2003. ACM Press.
- [18] Seungjoon Lee, Rob Sherwood, and Bobby Bhattacharjee. Cooperative peer groups in NICE. In *IEEE INFOCOM 2003*, volume 2, pages 1272–1282, San Francisco, CA, USA, April 2003.
- [19] Sergio Marti and Hector Garcia-Molina. Taxonomy of trust: Categorizing P2P reputation systems. *Computer Networks*, 50(4):472–484, 2006.
- [20] R. Morselli, J. Katz, and B. Bhattacharjee. A game-theoretic framework for analyzing trust-inference protocols. In *Proceedings of the Second Workshop on the Economics of Peer-to-Peer Systems*, 2004.
- [21] Lars Rasmusson and Sverker Jansson. Simulated social control for secure internet commerce. In *NSPW '96: Proceedings of the 1996 workshop on New security paradigms*, pages 18–25, Lake Arrowhead, California, United States, 1996. ACM Press.
- [22] WP 1.1 Requirements and Architectural Principles. D1.1.1 Application Scenario Analysis, Network Architecture Requirements and High-level Specifications. Deliverable D1.1.1, BIONETS Consortium, August 2006.
- [23] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, New York, NY, USA, 1994. ACM.
- [24] Paul Resnick, Richard Zeckhauser, Eric Friedman, and Ko Kuwabara. Reputation systems: Facilitating trust in internet interactions. *Communications of the ACM*, 43(12):45–48, 2000.
- [25] Paul Resnick, Richard Zeckhauser, John Swanson, and Kate Lockwood. The value of reputation on eBay: A controlled experiment. Working paper originally presented at the ESA conference, June 2002.
- [26] WP 4 Security. ID4.1 Security Architecture and Infrastructure Draft. Internal Deliverable ID4.1, BIONETS Consortium, November 2006.
- [27] R. L. Trivers. The evolution of reciprocal altruism. *Quarterly Journal of Biology*, 46:35–57, 1971.

- [28] V. Vishnumurthy, S. Chandrakumar, and E.G. Sirer. KARMA: A secure economic framework for p2p resource sharing. In *Proceedings of the Workshop on the Economics of Peer-to-Peer Systems*, Berkeley, California, June 2003.
- [29] Li Xiong and Ling Liu. PeerTrust: Supporting reputation-based trust in peer-to-peer communities. *IEEE Transactions on Data and Knowledge Engineering, Special Issue on Peer-to-Peer Based Data Management*, 16(7):843–857, July 2004.