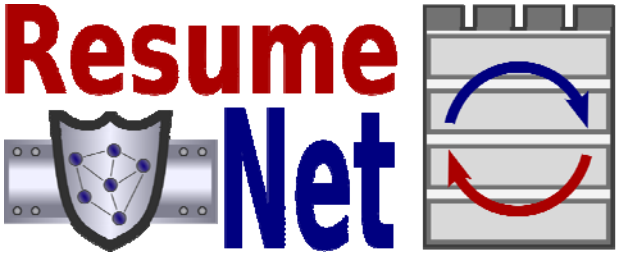
Resilience and Survivability for future networking: framework, mechanisms, and experimental evaluation



Deliverable number:	D4.1b
Deliverable name:	Federation requirements
WP number:	4
Delivery date:	28/02/2010
Date of preparation:	26/03/2010
Editor:	M. Karaliopoulos (ETHZ)
Contributor(s):	M. Karaliopoulos (ETHZ), G. Popa (ETHZ), C. Lac (FT), A. Fessi (TUM), A. Fischer (UP), C. Rohner (UU)
Internal reviewer:	P. Smith (ULANC)

Summary

The deliverable 4.1 is one of the project “light” deliverables requested by European Commission in the context of the project’s commitment to close interaction with the FIREworks Coordination Action. The deliverable aims at providing inputs to FIREworks for the compilation of a deliverable on federation requirements, which will aggregate contributions from all FIRE projects.

This is the second release of the document, which updates and revises the first version as issued in February 2009.

The deliverable consists of three main sections: an introduction briefly summarizing the scope of the experimentation within ResumeNet and the current thinking regarding the kind of experimentation facilities to be used. The main part of the experimentation activities will be carried out within WP4, officially launched in M18 of the project (March 2010); yet, there has been considerable interaction on experimentation issues in the project. The second section outlines the four experimental scenarios in the project and the testbeds that will be used to realize them. The level of detail in this description varies inline with the progress made so far in each experimentation scenario. Finally, the last section brings together some thoughts on the, rather limited, federation requirements coming out of ResumeNet.

Contents

1. Introduction

The experimentation work in ResumeNet is carried out in the project WP4 and its main aim is to evaluate the resilience framework aspects defined in WP1 and the mechanisms realizing this framework in the project WPs 2 and 3.

Four experimental scenarios are defined with the aim to address a wide variety of resilience issues in the context of different types of networks and service provision settings. Therefore, these four experimental scenarios have been selected to be complementary with respect to:

- network type: one scenario is considering wired networks and three scenarios consider wireless networks. The first of them is related to wireless multihop networks, the second evolves around self-organizing, opportunistic networks, and the third addresses smart environments, i.e., spaces filled with sensors and actuators within which devices are moving.
- types of network service faults that will be considered: they range from node misbehavior at different layers (MAC, routing), to software misconfigurations, and DDoS attacks.
- network functions: routing, wireless medium sharing, transport, but also signaling functionality provision will be addressed in the experimental scenarios.

The actual experimentation facilities to be used in these four scenarios are mainly in-house test beds. This choice minimizes interdependencies that might pose high overhead in the experimentation process while still satisfying the objectives of the experimentation tasks in the project. Examples of project in-house testbeds are the Wireless Mesh Network test bed of the ETHZ TIK group and the Haggle test bed maintained by the Uppsala University. Nevertheless, there are scenarios that would definitely benefit from larger-scale experimental facilities that may become available from FIRE or other EU R&D activities. An example of this second alternative regarding experimentation facilities is PlanetLab (Europe), currently under the responsibility of OneLab2 project and/or the test bed that will come out of the EC FP6 ANA (Autonomic Network Architecture) project.

The exact dependence on these external testbeds has been under assessment since the beginning of the project; in the case of OneLab2, for the example, project representatives have attended the kick-off meeting and there have been discussions with the groups involved there regarding the progress of work. The facilities offered by national initiatives such as the G-Lab in Germany, have also been explored, mainly in relation to the third experimentation scenario envisaged in ResumeNet. The baseline at the moment is to largely rely on the in-house testbed experimental facilities, while constantly assessing the possibility to use larger-scale experimentation facilities from FIRE or other EU R&D projects. In the context of the project experimentation, these test beds will be enhanced to facilitate the four scenarios envisaged in the project.

An additional experimentation facility that is used in the project is the Wireless Multihop Network operating in the village of Wray, situated approximately ten miles from the city of Lancaster in the north-west of England. The facility was initially used in the task of assessing challenges and their impact on network resilience [ResD1.1] and later for the interference detection and mitigation activities in WP2; it may accommodate further experimentation in the course of time.

2. Description of ResumeNet experimentation scenarios and related testbeds

1.1.Experimentation scenario: “Wireless Mesh Networking”

1.1.1.Scenario description

Ad-Hoc wireless networks are suitable for a vast array of applications where central control cannot be provided due to various reasons. However, the lack of central control also deprives ad-hoc networks of important properties. For instance, cooperation cannot be taken for granted when this assumption does not hold, that is, in situations each of these nodes is owned by a different entity. Therefore, a number of specific challenges stem from this essential characteristic of ad-hoc networks. Generally speaking, misbehavior can be classified as being either selfishness or maliciousness.

When referring to *selfishness*, one of the essential challenges is cooperation between nodes. The simplest form of cooperation is represented by forwarding data on behalf of other stations. Nevertheless, since cooperation is a global objective that is not recognized by individual nodes, a mechanism should be provided in the network to determine the nodes to collaborate or, in other words, to make them aware of the individual and global advantages that collaboration can provide.

In practice, the most encountered situation is that of a wireless mesh network, which mostly contains static nodes. The number of mobile stations that are participating in these networks is expected to be rather small. These are usually community wireless networks whose nodes are controlled by individual participants. Because of this, selfish forwarding strategies may be a reasonable (rational) choice for some devices. In particular, if the owner of a station considers that it cannot obtain a proper quality of experience due to the forwarding tasks its station also has, it may instruct the software to start dropping packets which it should in fact forward. For mobile devices, such as laptops or WiFi-equipped phones, the selfish behavior can be triggered, more obviously, by the low battery levels. In both cases, selfishness occurs not as a malicious behavior, but rather as a result of limited resources. Consequently, a device would rather choose to address its own requirements, rather than helping others to achieve their communication goals. Analyzing the reasons that could generate selfishness, as well as the static nature of the nodes which mostly compose wireless mesh networks, we infer that the number of nodes displaying selfish behavior should be relatively small, under reasonable traffic conditions.

In any case, we want to minimize selfish behavior in the network. Having noticed that currency-based mechanisms have a number of drawbacks, we have turned our attention to analyzing the dependencies between nodes created by the corresponding data streams. Our theoretical study[Pop10] shows that even in a network where all nodes are implicitly selfish (a node does not help any other as long as it is not helped), the cooperation levels can be brought to reasonable levels (above 90%) if certain conditions are met. This refers mainly to the fact that nodes need to be made aware of the dependencies that exist between them. However, no existing routing protocol for wireless mesh networks is providing this information directly.

- Therefore, our experimental study will need the following elements to fulfill its purpose, namely that of offering a protocol for selfishness avoidance:

- A software module residing on every station whose purpose is to compute based on traffic requirements and power level when to apply a selfish strategy and when not. Note that the number of selfish node should remain fairly low as long as the throughput requirements from other stations remain also low.
- A set of modifications to a known shortest path routing protocol (such as OLSR), whose role would be to provide information about dependencies created by data streams between nodes to all networked stations.

A software module running on each individual machine for computing the communities of cooperating nodes and responding to changes in the level of cooperation.

Initially, all communication will be routed based on a shortest path policy, as prescribed by the initial routing protocol. Note that usually selfish nodes conceal their existence during the routing and start manifesting as such during the packet forwarding phase. Upon detecting selfish behavior, nodes that have previously used for forwarding relays outside their own community and which see the quality of their communication decreasing will decide to reroute. The new route will be with complete certainty free from selfish nodes as long as the new relays are members of the same community of mutually cooperating nodes.

1.1.2.Testbed description

The experiments will take place on the wireless multihop network testbed of the Computer Engineering and Networks laboratory (TiKNet), at the G floor of the building housing the Department of Information Technology and Electrical Engineering in Swiss Federal Institute of Technology (ETH Zürich). The testbed is readily available, having already been used in prior research work.

TiKNet consists of approximately 20 nodes (PCs) split in two categories:

Dell PCs with 2GHz processor and 512MB RAM memory;

PCs with 866MHz processor and 512MB RAM memory.

All the nodes are currently equipped with D-Link DWL-AG530, 108/54Mbit Tri-Mode Dualband WLAN Adapters. The testbed is currently being upgraded through the addition of new nodes.

Each node is running GNU/Linux and is connected also to the local wired network in order to ensure a safe way for the configuration and maintenance operations. Thus, such operations are usually performed out of band (via the wired network). The aforementioned tasks can be performed by using a standard web interface installed on every node [webmin] or by using ssh.

In-house software for emulation of various types of attacks is currently being developed. In addition, readily available traffic generation tools such as tcpdump, mgen, iperf, tudp shall be employed.

The testbed can be accessed by members of ETH Zürich involved in the ResumeNet project and by master students that help with the testbed development.

1.2.Experimentation scenario: “Opportunistic Networking”

Opportunistic networks consist of typically mobile nodes that are intermittently connected to each other using short range communication. In contrast to most other networks, no end-to-end

path between communicating nodes is assumed. Nodes store, carry, and forward messages upon encounter. A store-carry-forward transport service allows the forwarding of data to a node that is not connected to the source at the time the data is sent, if there exists a 'temporal path' between source and destination. Node mobility is thus important for data dissemination, in that it causes contact opportunities between different nodes and also allows nodes to physically transport data to bridge areas where no connectivity might be available.

Opportunistic networks have mechanisms built in to make the network resilient to intermittent connectivity. Delay tolerance is possible due to late binding of, for example, addresses and progressive forwarding decisions on the nodes. Resilience is increased by introducing redundancy and spreading multiple copies of the same data distributed to different nodes. By having multiple copies, chances that at least one of the copies is on a (non-predictable) temporal path are improved. However, limited storage capacity and battery require a careful trade-off increasing redundancy to maximize delivery, and limiting redundancy to avoid storage overflow.

Our experimentation is two-fold. We investigate the impact of node misbehaviors on opportunistic networks and aspects related to congestion management. Experimentation is performed on the in- house Haggle testbed that runs on both mobile phone and virtual machines, as well the ONE emulation framework from Helsinki University.

1.2.1.Scenario description

Impact of node misbehavior

We analytically assessed the vulnerability of two popular data relaying alternatives, the unrestricted and two-hop relay schemes, to node selfishness [KAR09a]. The model can be used to quantify the vulnerability of the two relaying schemes to node selfishness but also drive remediation actions against it. We experimentally validate the model and extend the results with jamming, buffer constraints and finite contact duration.

Congestion control

We intend to investigate aspects related to congestion management, that is dissemination strategies, resource management (e.g., data ageing), and resilience to attacks. The goal is to investigate the influence of different strategies on system behavior and performance to finally improve resilience. We use the Haggle architecture for implementing the different strategies.

1.2.2.Testbed Description

The Haggle testbed allows emulating a mobile opportunistic network and conducting repeatable tests in a controlled and easy to manage environment. The Xen virtual machine monitor is at the core of the testbed. Xen supports execution of multiple guest operating systems (or emulated Haggle devices), on a single physical machine, that are monitored by a host system.

Opportunistic network connectivity is emulated over a virtual Ethernet bridge. Network topology changes are performed by controlling connectivity with traffic filtering. This is done by setting rules in the iptable on each node in the testbed and thereby blocking traffic from certain nodes.

The order in which the rules are configured is specified in a scenario file. These scenario files can automatically be generated from either real-world traces or statistical models. The Ethernet bridge also allows interconnecting with other networks, for example another testbed (scalability), or a wireless network with real-world Haggle nodes running on mobile phones.

A graphical management console allows starting/stopping nodes, controlling connectivity, and visualizing the internal state of the diverse nodes and their interaction. After an experiment, an collection of analysis scripts automatically generates statistics and graphs from the logfiles of the nodes (e.g., about delay, delivery ratio, dissemination topology, etc.)

The testbed runs at a dual-core desktop computer (3GHz CPU, 4GB RAM), using Linux for both host and guest operating systems. Up to 30 Haggle devices are supported at the moment. Further scaling is planned although not critical to produce interesting results.

1.3.Experimentation scenario “Service-Level Resilience”

1.3.1.Scenario description

In this experimentation scenario, two concepts for service-level resilience will be evaluated; P2P-based services, and virtualization.

Experimentation with P2P overlays for service resilience

Services will be classified in two categories:

Services that are carried out by a large scale P2P network where all end points are contributing to the service. Examples of these services are distributed rendez-vous points, for example, for establishing a VoIP phone call. This example is depicted in Figure 1.

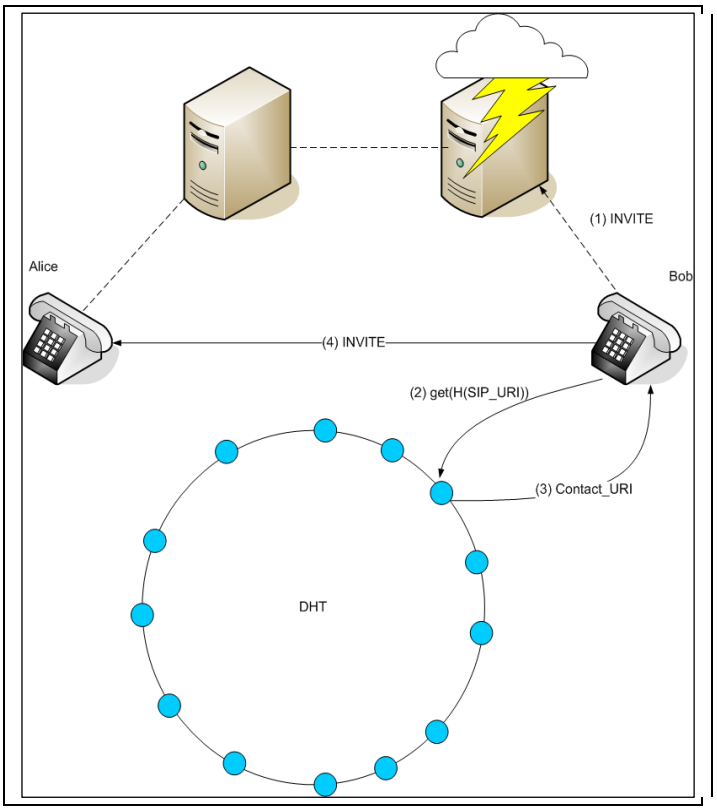


Figure 1. P2P-Session Establishment in case of server failure

Services provided by a service provider, where the nodes offering the services are part of the provider’s infrastructure. In this case, servers and clients are organized in a P2P network.

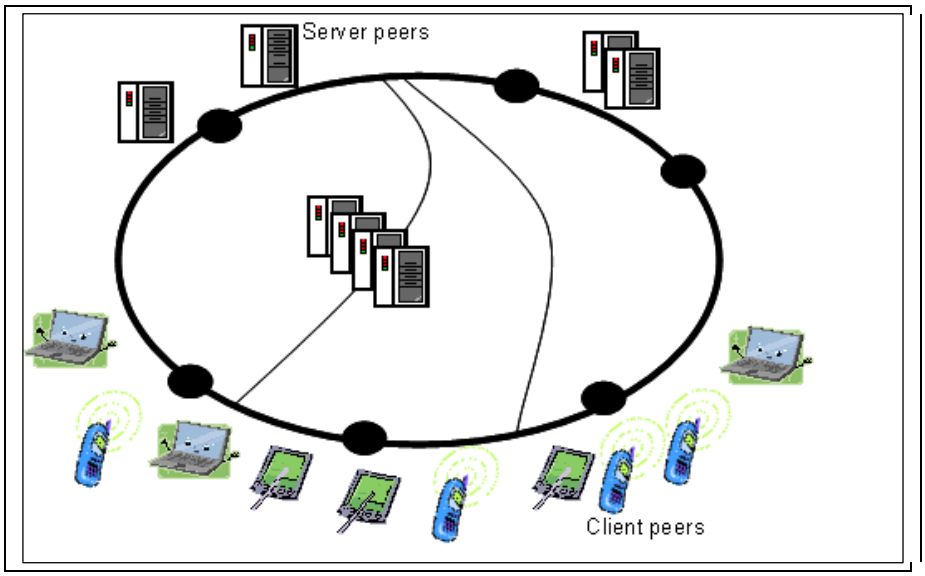


Figure 2. Hybrid P2P Overlay Networks with Clients and Servers Peers

The concepts that will be developed by TUM in WP3 for P2P-based service resilience and overlay-assisted service discovery will be evaluated. The overlay-assisted service discovery will involve the client peers actively in the remediation process when a challenge occurs, so they will be able to find other server peers, and re-establish connectivity in the overlay. Furthermore, this concept will facilitate locating servers when they change their location. This is the case, for example, when virtual servers have been migrated. Migration of services is a remediation mechanism anticipated by UP (see below). Therefore, it is planned that TUM will cooperate with UP for this purpose.

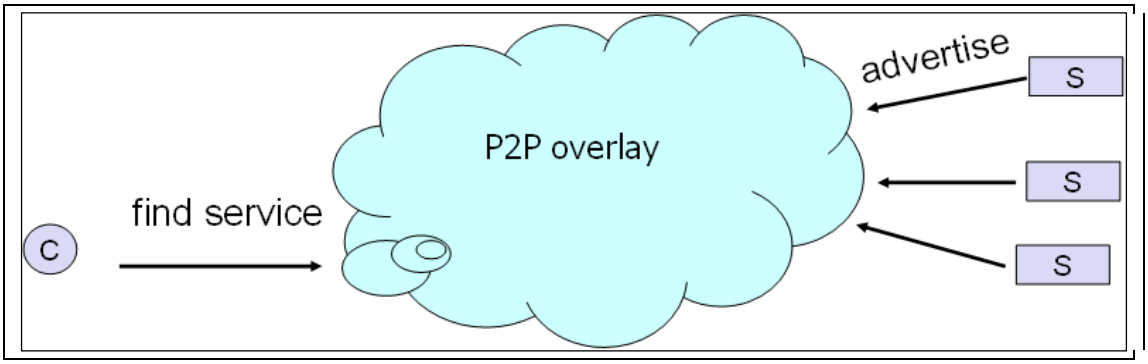


Figure 3. Overlay-based Service Discovery

The experimentation scenario will include the performance degradation up to complete failures of some major components in the service overlay as well as the underlying network. The failure of servers can be emulated by shutting down the server. The signaling for session establishment should remain possible. The failure of routers can be emulated by shutting down a subset of the peers, which are supposed to be connected behind the router. The overlay should be able to recover from this kind of challenge.

Performance degradation could be emulated by activating a higher CPU load at the server. It might be evaluated whether this could introduce additional complexity, or whether session establishment at the peers' side can still be performed smoothly.

It is worth mentioning that all these objectives for experimentation, challenge emulation and remediation mechanisms remain subject for changes. Potential changes in the experimentation plan would be mainly motivated by two reasons: better integration with other experiments; and increasing the scientific value of the experiments.

Experimentation with virtualization for service resilience

The second major concept for service-level resilience is virtualization. Virtualization provides mechanisms to abstract a service from the underlying hardware, enabling service mobility in order to enhance service resilience. An experimentation scenario will examine the requirements and benefit of dynamic service mobility, taking different migration strategies into account as a remediation mechanism. Each of the available migration strategies comes with a cost attached to it, in terms of bandwidth used, service downtime, total time to recovery, session stability, etc. The respective cost of these strategies has to be measured and compared to each other in order to compare the benefits and drawbacks of all migration strategies in the context of different challenge scenarios. Restrictions on mobility, both on the service side and on the virtualization side, have to be identified. Additionally, appropriate control mechanisms to trigger a service migration will be defined.

1.3.2.Testbed description

Testbed for P2P Services

In order to perform an extensive functional and quantitative evaluation of the service-level resilience with P2P, a large-scale testbed would be required.

Software: TUM will use a self-implemented prototype for a solution for highly-reliable signaling for VoIP, which is called Cooperative SIP (CoSIP) [Fessi07]. The prototype is implemented in Python programming language. It currently makes use of an implementation of the Kademlia DHT algorithm called "Entangled"[1], (also in Python). The implementation will include a control framework for establishing (and re-establishing) a large-scale distributed testbed and collecting measurement data. For example tools such as "parallel ssh" will be used in order to automate the setup procedure of the testbed. A lightweight web server will be running on each P2P node in order to better monitor the system.

Testbed Platform: TUM has been working on initial experiments with PlanetLab. Current experiments involve a number of peers up to 500. Future experiments with several thousands of peers or even more would be useful for better evaluation. The possibility for evaluation with mobile nodes would open new opportunities for experimentation. The concept of "slices" used by PlanetLab has been sufficient since we were able to install Python on it and run our P2P software.

Hardware: Nodes equipped with a minimum of 2 GBytes RAM and an up-to-date CPU power would be useful[2].

[1] http://entangled.sourceforge.net/

[2] Our experiments with PlanetLab up to now showed that we had problems with PlanetLab nodes with 1GBytes of RAM.

Bandwidth: The P2P system is used for signaling only which is a small portion of the traffic. The bandwidth requirements are rather minimal.

Access policies: TUM is currently already running initial experiments on PlanetLab. The access is provided to students at TUM, based on the access policies of PlanetLab. (An account is needed to login to an assigned PlanetLab "slice"). The experiments involve emulating phone calls with SIP UAs that are running on the PlanetLab nodes.

It might be possible to provide a demo where users can perform an "echo" call to one of these nodes, or use the running testbed on PlanetLab to initiate phone calls peer-to-peer between each other. However, it is not clear yet, how such facilities could contribute to the evaluation of the anticipated resilience goals.

Testbed for virtualization

In order to evaluate the requirements and options of Virtual Machine migration, a large-scale testbed consisting of virtualized servers in different subnets is required.

Software: UP research currently focuses on system virtualization, using the XEN hypervisor with additional management primitives developed in the JAVA programming language.

Hardware: The hardware should be able to host several XEN virtual machines per physical host. The most significant requirements are put on CPU and RAM. Current test-runs use hardware in the range of 2-8 multi-Gigahertz cores and 4-32 Gigabyte RAM per physical host.

Bandwidth: Since Virtual Machine migration includes the movement of large amounts of data (Virtual Machine images), a high amount of available bandwidth is necessary in a testbed.

Testbed Platform: A large-scale testbed with multiple subnets, consisting of machines virtualized with XEN would be useful to evaluate the implications of service mobility. In order to carry out tests, it would be necessary to get full root access to the machines in question, being fully able to create and move any number of Virtual Machines. The current access policies in PlanetLab, for example, would not be sufficient to perform these experiments.

1.4.Experimentation scenario “Communicating objects' data platform”

The generalization of Internet usage, the rise of service offers for mobile terminals, and the increasing use of tags such as RFID (allowing the communication with, and about, objects - through naming systems such as Object Naming Services - and data search/collection systems in compliance with EPCglobal standards, for instance) induce great changes in commerce businesses. The ubiquitous access to product/service offers, i.e., wherever/whenever these are needed through multiple available communication means, thus tends to become impossible to circumvent in the 2010s.

In the framework of a French national project (ICOM[3]), a technical platform was built, allowing exchanges between applications through heterogeneous hardware and software. This intra - or inter - enterprise infrastructure links various identified objects (RFID, 1D/2D bar codes, NFC, ...) to the company information systems and fixed/mobile terminals and/or, to a lesser extent, the objects to each other.

This multi-networks and multi-communicating objects middleware will allow real-time processing of large volumes of information, i.e., event exchanges related to the objects with the applications and their information systems located locally, or not, and belonging, or not, to the distributor company. This integration of the value chain of trade and distribution associates different stakeholders (design, production, logistics) in order to generate new business models within the “Internet of Things” framework. The numerous applications are related to logistics (parcels follow-up, inventory), sales functions (items traceability), or marketing (merchandising).

1.4.1.Scenario description

The changing environment offered by the PubSub platform (multiple events from various sources, frequent access by new clients, ...) requires an adaptable and dynamic security policy. This need is also motivated by the diversity of threats that could face the platform. It has been seen in [Deba07] that the response to threats can be achieved by applying a security policy using contextual rules, allowing both to specify the normal system behaviour, and its nominal behaviour in the presence of threats. An extension to this approach was presented in [Khei09]: it offers a solution allowing dynamic configuration of contextual policy rules, and especially in multi-services environments comprising a large number of interdependent components.

We shall use this solution, i.e., contextual definition of the security policy, to secure the communicating objects' data platform. The process includes several steps:

- specification of the nominal system behaviour, i.e., the regular security policy standard applied to the platform in the absence of any external risk; it defines the permissions granted to each client, the type of data exchanged, the access rights for the publishers/subscribers, etc.
- specification of policy reaction following the threats, i.e., the set of security rules related to the challenge contexts; the specification of these rules will obey to the following procedure:

[3] Infrastructure pour le COMmerce du futur

- risk analysis in order to define a policy response after challenges detection, these challenges including: (i) confidentiality, e.g., passwords bypassing or publishers/subscribers data filter; (ii) integrity, e.g., forgery, at the PubSub service provider level, of data sent via XML routers (see below); (iii) availability, e.g., denial of service attacks against the platform routers.

- definition of challenge contexts characterizing identified risks, and activated when events related to such risks are detected (e.g., rejection of data published by a router);

- definition of actions, as contextual rules, in the case the challenge contexts are activated.

- definition of translation functions (mapping) connecting the challenge contexts to certain attributes of the recovered alert messages. These alerts specify the components of the platform related to the ongoing challenge, these components being the only ones concerned by the activated challenge context.
- disabling challenge contexts after a latency period which depends on the level of risk; a classification of the challenge contexts, according to the precariousness of the risk associated to each of them, will be established.
- deployment of the dynamic security policy, after a translation of this policy into a set of configurations applied to different control points (routers) of the platform.

1.4.2.Testbed description

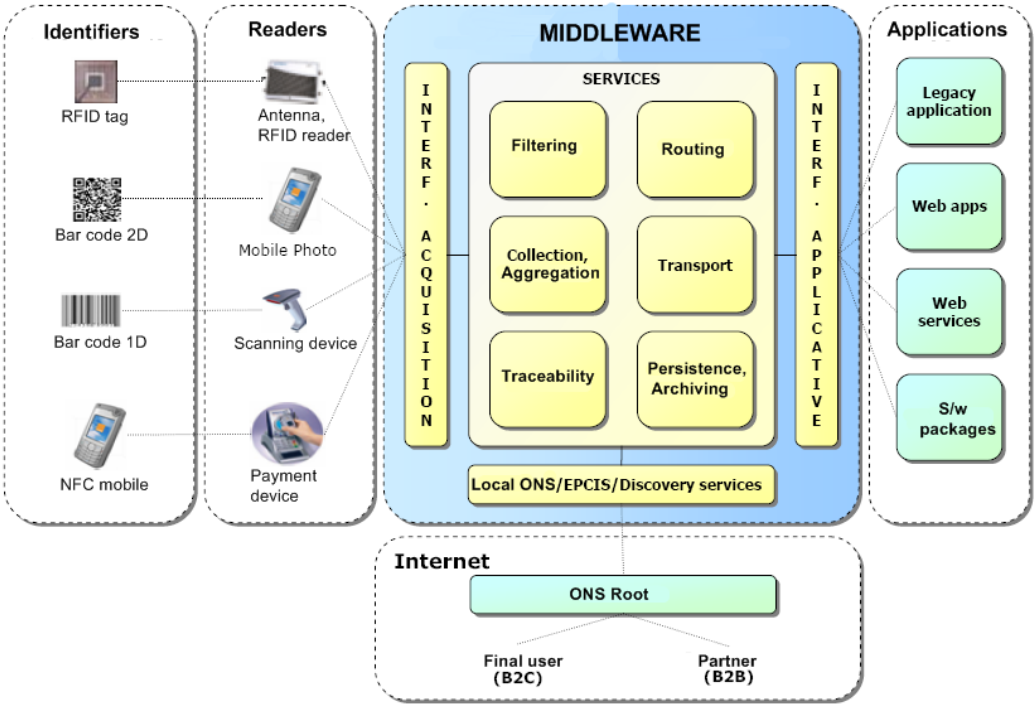


Figure 4. ICOM architecture

Figure 4 describes the two main parts of the architecture defined within the ICOM project: in the physical area, the objects are identified through their label (RFID, 1D/2D bar codes, NFC) by mass-market readers (cameras embedded in mobiles) or dedicated terminals (RFID antenna/reader, reader device, payment terminal) and the data are sent to the logical area. Based on an open source RFID software platform and EPCglobal standards (ALE[4], EPCIS[5]), the middleware consists of different modules:

- *collection* and treatment by *aggregation* → the acquisition/formatting of large data volumes allows the unification of different label types;
- *filtering*, *routing* and *transport* based on the contents in order to send only useful information and adapt them to the application requirements (see below);
- *traceability*/*persistence* for adding to the objects' observation events business information, process verification and *archiving* in databases;
- linked to an ONS Root, a distributed objects' name server (*local ONS*) processes the partners queries (B2B) or customers queries (B2C), guiding them to the requested services.

The ICOM middleware is finally interfaced with various enterprise applications such as Web/existing applications, Web services.

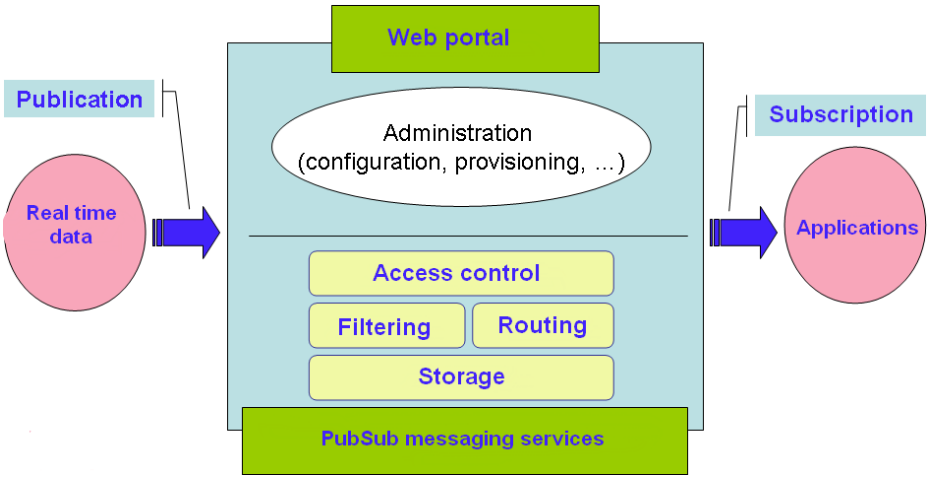


Figure 5. Principal functions of the data platform

The part of ICOM we are dealing with concerns the filtering functions and routing information with the use of a PubSub-based platform decoupling message senders and recipients (Fig. 5). This platform is based on a network of XML routers using hardware to process messages and allowing very high performance, the network covering itself a network of (traditional) IP routers. Through this platform:

- contents-based routing is done from selection filters (subscriptions) of XML documents through the shortest path to end nodes before distribution to the recipients;

[4] Application Level Events (http://www.epcglobalinc.org/standards/ale)
[5] EPC Information Services (http://www.epcglobalinc.org/standards/epcis)

- each company can manage two types of users (data publisher/sender, data subscriber/client) and closed user groups;
- publishers broadcast their messages to authorized recipients;
- subscribers only receive relevant data, thanks to contents filtering (subject, EPC code, ...), through chosen communication channels (Web service, Java Message Service, RSS, SMTP, SMS, ...);
- messages are stored until they have been issued;
- in addition to access control for both publishers and subscribers, a Web portal manages the administrative tasks (client configuration management, accounts provisioning, ...).

As of today, this testbed is accessible only internally. A larger scale testbed has not been considered up to now.

3. With respect to federation

As already mentioned in Section 2, ResumeNet is largely relying on in-house testbeds to carry out its experimentation activities. This is the case with three out of the four experimental scenarios in the project. On the contrary, the third experimental case will need a larger facility in the scale of PlanetLab but with additional features that do not seem to be available at the moment (see Appendix for a correspondence of the project with the OneLab management team on this).

In the longer term, leveraging the other three experimental scenarios to use larger experimental facilities is an opportunity, which is only viewed positively in the project. In general, federation of testbeds seems to make sense when interested in scalability and heterogeneity and these two aspects are relevant when assessing resilience. At this stage of research, however, simple scenarios with a few nodes might still be complex enough to fully understand the interactions and implications of all the involved mechanisms and therefore difficult to project on requirements for federation.

The governance model of the in-house testbeds is rather ad hoc, the responsibility being with the network administrators or nominated staff members in each lab. They are used mainly internally within research groups for research and academic purposes. There are usually two-three levels of access, e.g., {default, user, admin} and remote experiment execution is enabled via password sharing with the remote party.

The testbeds could eventually be made available to the FIRE and the broader research community.

References

[Aad04] Aad, I., Hubaux, J., and Knightly, E. W. 2004. Denial of service resilience in ad hoc networks. In Proceedings of the 10th Annual international Conference on Mobile Computing and Networking (Philadelphia, PA, USA, September 26 - October 01, 2004). MobiCom '04. ACM, New York, NY, pp. 202-215

[Ande03] Anderegg, L. and Eidenbenz, S. 2003. Ad hoc-VCG: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. In Proceedings of the 9th Annual international Conference on Mobile Computing and Networking (San Diego, CA, USA, September 14 - 19, 2003). MobiCom '03. ACM, New York, NY, pp. 245-259

[Deba07] H. Debar, Y. Thomas, F. Cuppens and N. Cuppens-Boulahia, "Enabling automated threat response through the use of a dynamic security policy", Journal in Computer Virology, Vol. 3, N° 3, August 2007

[Fessi07] Fessi A. et al., “CoSIP – a hybrid architecture for reliable and secure SIP services”, PIK - Praxis der Informationsverarbeitung und Kommunikation. Volume 30, Issue 4, Pages 206–212, Dezember 2007

[Khei09] N. Kheir, H. Debar, F. Cuppens, N. Cuppens-Boulahia, and J. Viinikka, “A service dependency modeling framework for policy-based response enforcement”, DIMVA, Milan, Italy, July 9-10, 2009

[Pop10] G. Popa, E. Gourdin, F. Legendre, and M. Karaliopoulos, "On Maximizing Collaboration in Wireless Mesh Networks Without Monetary Incentives", submitted to RAWNET 2010 Resource Allocation in Wireless Networks (with WiOpt 2010), Avignon, France, 4 June 2010

[ResD1.1] Understanding challenges and their impact on network resilience. ResumeNet Deliverable D1.1, March 2009

[tik] TIK-Net testbed, http://tiknet.ee.ethz.ch/doku.php

[webmin] http://webmin.com

[Zhon05] Zhong, S., Li, L., Liu, Y. G., and Yang, Y. 2005. On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks: an integrated approach using game theoretical and cryptographic techniques. In Proceedings of the 11th Annual international Conference on Mobile Computing and Networking (Cologne, Germany, August 28 - September 02, 2005). MobiCom '05. ACM, New York, NY, 117-131

[KAR09a] M. Karaliopoulos, Assessing the vulnerability of DTN data relaying schemes to node selfishness, IEEE Communications Letters, Vol. 13, No. 12, pages 923-925, December, 2009.

Appendix

Correspondence with OneLab testbed

ResumeNet- OneLab email

Dear Thierry, Walid and Timur,

we currently at the ResumeNet project (www.resumenet.eu) are looking for some testbed facilities. We have been running experiments with P2P networks on PlanetLab. However, now we would like to do some experiments where the testbed facilities at PlanetLab are not sufficient anymore for our needs. Due to the concept of "slices" at PlanetLab, we can not have privileged administrative control (root access) on the machines. The machines are shared by several users at the same time.

Since we would like to do experiments with virtualization technologies (e.g. Xen, KVM) we would need a pool of PCs, in the ideal case in a distributed manner, where we have full remote access to them.

Could you please tell us whether OneLab2 has been working on such as testbed facility? Any other information considering the access policy, and the time plan would be very useful.

Attached is a small description of what we need. Please feel free to let me know if you have any inquires.

Thanks you very much in advance.

Best regards,
Ali Fessi

Attachment:

Requirements for a ResumeNet testbed

The FP7 FIRE project ResumeNet includes different use cases for experimentation. The experimentation includes local testbeds as well as large scale testbeds. Some of the experiments include research on virtualization technologies, e.g., Xen, KVM and their management, e.g., migration from one physical host to another in the presence of challenges, such as attacks or natural disasters.

- A distributed testbed for running these experiments should be able to provide the following functionality:
- Ability to deploy Virtual Machine images specified in Open Virtualisation Format (OVF) files. OVF is a vendor-neutral format for virtualized operating systems supported by a number of system-level virtualization mechanisms

- Full access to the Hypervisor / Virtual Machine Monitor (VMM). In order to modify and enhance the management of virtual machines it is not sufficient to only have access to the virtual machines – access to the VMM is imperative. This includes specifically a usable execution environment within the VMM.
- Ability to conduct scalability and stability tests on the infrastructure. In order to evaluate both scalability and stability issues of the developed prototypes it is necessary that the testbed provides a significant number of physically different hosts, allowing for creating a large number of virtual machines on top of a realistic physical network.
- Ability for ResumeNet partners to have independent access to the deployed infrastructure (e.g., via ssh). Integration of components developed by different ResumeNet partners may require several partners to have concurrent access to the testbed. It would be highly impractical to have one partner relay all requests for testing software.
- High bandwidth connections. Current Virtual Machine management applications do not optimize migration of Virtual Machines with regard to size. Indeed, migration may consume a considerable amount of bandwidth. In order to not negatively influence any results acquired during the test, the available bandwidth between nodes should at least be on the order of 100MBit/s (GBit connections would be preferable).

Offering the additional feature for the testbed users to store our images on a server and recover them later to continue the experimentation at the same point would be also very useful.

Contact persons:

Ali Fessi (Technische Universität München) <fessi@net.in.tum.de>

Andreas Fischer (University of Passau) <fischean@fim.uni-passau.de>

Dr. Merkouris Karaliopoulos (ETH Zurich) <karaliopoulos@tik.ee.ethz.ch>

Onelab2 team reply

Hi,

- our platform is not designed for doing research on virtualization; we use virtualization as a technique, not as a goal
- we cannot open the root context (vmm) b/c of the necessity to control nodes, in particular wrt intrusion complaints

IMHO, PLE will not be suitable for these needs before a long time;
on the other hand, they might wish to run a privately owned myplc with Aki Nakao extensions towards Xen;
