Towards Semantic Web: Seamless Integration of Services and Devices for the FRUCT Community

Valery Kirkizh
Saint-Petersburg State University of Aerospace Instrumentation
Saint-Petersburg, Russia
vkirkizh@vu.spb.ru

Maria Komar
Yaroslavl State University
Yaroslavl, Russia
maria.s.komar@gmail.com

Vitaly Petrov
Tampere University of Technology
Tampere, Finland
vitaly.petrov@tut.fi

Abstract

Semantic Web presents a generalized approach of applications and services integration that enables direct communication between different entities without any user interaction. Besides obvious advantages of Semantic Web usage, not all of proposed methods can work in a highly-diverse heterogeneous environment, the community or company intranet system usually looks like. In this paper we focus on FRUCT community online services integration under Semantic Web paradigm, aiming to provide an applicability assessment of particular service-to-service communication methods. Also we investigate the Semantic Web usefulness for Machine-to-Machine and Machine-to-Service communications, aiming to proceed with secure and convenient way of user authentication to a number of services. Finally, we conclude and highlight possible research issues that have to be solved during the evolution of Semantic Web.

Index Terms: Semantic Web, Internet of Things, Machine-to-Machine Communications, Social Networks, Bluetooth, Wireless Authorization, Smart Spaces.

I. INTRODUCTION

In the past few years, the amount of applications people use in their daily life grew up dramatically. Moreover, standardized methods of user-to-service interaction and harmonization of common used file types and protocols made it possible for the user to communicate with multiple services simultaneously.

So now the automated environment could be considered as a network that has star topology with the user being a central node. Such system works fine to some extend, however, the extensive growth can lead up to a scalability problem: the amount of information flows the user can handle is limited. As such, being a central node for all the services, he/she cannot efficiently maintain the information exchange after some point. So, even if considering an interactive system that communicates with the user, forwarding all the traffic “through” him is not always the best option. Therefore, a number of approaches were proposed to enable direct communications between heterogeneous services and applications without user interaction. Despite the fact, most of these approaches are straight forward; there are some difficulties in appropriate technology selection for a particular task, with respect to the number of involved nodes, types and amount of traffic between them, etc. So there are some well-known practices rather than the “silver bullet”. One of the commonly used titles for highly-integrated systems with Service-to-Service communications being enabled is Semantic Web.

The Semantic Web is an attempt to combine a lot of the disparate technologies and provide a possibility for automatic integration of the heterogeneous services and applications [1]. Generally speaking, the Semantic web is not an independent technology; it is the set of the ideas, approaches and technologies for applications and services connection, data representation, static and dynamic information provisioning and ontology combination. It is a collaborative movement led by the World Wide Web Consortium (W3C [2]) – an organization which develops different standards connected with the Internet. The standard of the Semantic Web promotes common data formats, the semantic markup in web pages, unified interaction between applications [3]. The main aim of the Semantic Web is converting the current unstructured web into a “web of data”. According to the W3C, “The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries” [4]. The concept of the Semantic Network Model was formed in the early sixties by the cognitive scientist Allan M. Collins, linguist M. Ross Quillian and psychologist Elizabeth F. Loftus in various publications [5, 6], as a form to represent semantically structured knowledge. The term “Semantic Web” was coined by Tim Berners-Lee [1], the inventor and director of the World Wide Web. The semantic web is a vision of information that can be easily interpreted by machines, and machines can perform more of the boring work including finding, combining, and acting with information located in the web [7].

In this paper, summarizing the previous results in the field of study, we discuss the applicability aspects of Semantic Web technologies on a concrete example: online services integration for the FRUCT Community. Focusing on a set of most commonly used approaches for service-to-service communications, we highlight their features and make a one by one mapping with the set of common integration task, which the developer usually solve. The goal of this work is to assist engineers making a motivated choice of a particular technology to go with and also to investigate possible bottle-necks of well-known solutions.

Moreover, in the second half of the paper Semantic Web paradigm is extended to allow Machine-to-Machine and Machine-to-Service communications. This rapidly increases the amount of possible applications, but on the same time brings new, previously negligible in the field, research challenges, like capacity, delay and power consumption boundaries.

II. Interaction of Software Services Through Semantic Web

FRUCT Association is a cooperation of academy and industry institutes involving top universities and companies from Russia and European Union. The aim of the program is to develop the international collaboration between company engineers, university students and stuff [8]. FRUCT is not a classic commercial company, nor a traditional open community. This fact leads to several problems: no clear internal structure and community members diversity. These issues create serious difficulties for community management. That’s why special instrument named FRUCT Social Network has been developed [9, 10].

Before social network was established, there were some FRUCT web resources, such as main website www.fruct.org [8], E-WeREST website [11] and others. Although all these resources were build on different platforms and were standalone services which could not change any information with each other, they did not need to integrate, because all these resources except www.fruct.org were information sites without any user interaction. With social network development the necessity to integrate existed resources and combine the same entities such as members, universities, laboratories, etc. appeared and became a significant issue, because it is senselessly to keep equal data and equal entities at different sites. So Semantic Web way seems being able to solve this problem.

Currently, the FRUCT web resources structure presents a simplified version of general Semantic Web architecture [12] with a set of information providers, a number of applications and websites for user interaction and an ontology repository service (SQL database) for information storage and representation.

Below, using the FRUCT social network as an example, the application of different Semantic Web technologies is described. In particular, we will focus on four methods that Semantic Web provides for web services integration.

Case 1. Usage of central storage

If you have some web sites or services located on a single server, you can use the first method which is concluded in creation of a common central node. Usually it is shared database and communication with it is carried out by formulated rules. Thus different web services get the possibility to have access to joint data.

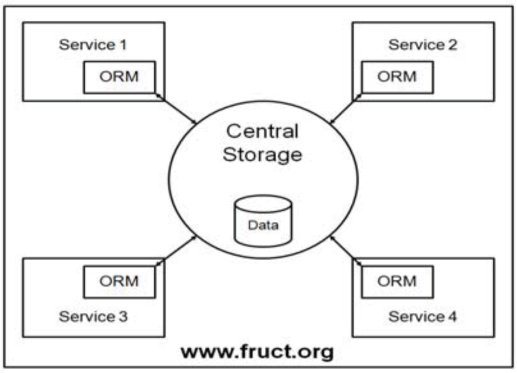

Fig. 1. Central node usage

For example, FRUCT community contains information about FRUCT members, laboratories, working groups, research and development projects, etc. This information can be used both main web site (www.fruct.org) and social network (social.fruct.org). Records in database are saved in unified standard and can not be broken by some other service, but good practice is to use object-relational mapping. Object-relational mapping (ORM) [13] in computer software is a programming technique for converting data between incompatible type systems in object-oriented programming languages. This creates, in effect, a “virtual object database” that can be used from within the programming language. Thereby instead working with database records directly you work with habitual objects in your preferable programming language.

Case 2. Real-time remote data access

If your services are located on remote servers or you cannot use common central node, you have to choose the second method which is concluded in conventional web services transformation to semantic web services. Semantic web services are the server end of a client-server system for machine-to-machine interaction via the World Wide Web.

Semantic services are components of the semantic web because they use markup which makes data machine-readable in a detailed and sophisticated way (as compared with human-readable HTML which is usually not easily “understand” by computer programs). Semantic web services are built around universal standards for the interchange of semantic data, which makes it easy for programmers to combine data from different sources and services without losing meaning. Web services can be activated “behind the scenes” when a web browser makes a request to a web server, which then uses various web services to construct a more sophisticated reply than it would have been able to do on its own. Semantic web services can also be used by automatic programs that run without any connection to a web browser.

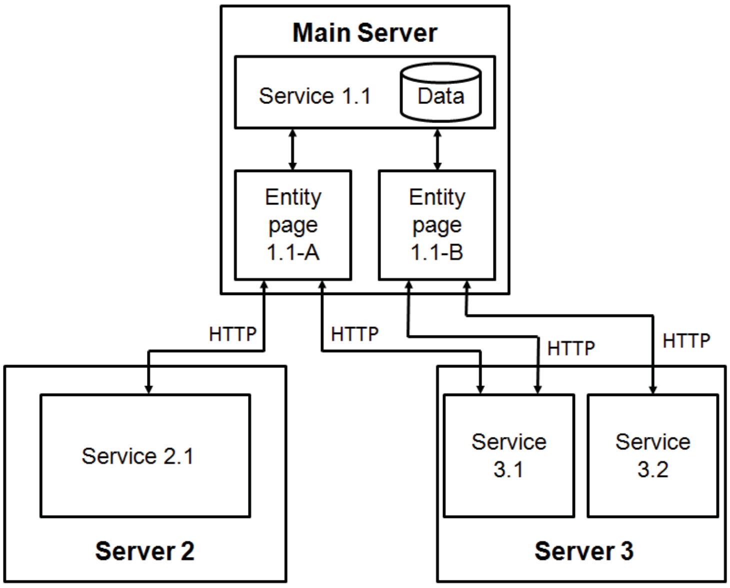

Fig. 2. Real-time remote data access

From the point of view of this method all database records described above such as members, laboratories, etc. become semantic entities. They are stored on the main server, for example, www.fruct.org. Other services such as social.fruct.org have to have access to this data. Every entity has to have its own page located at special URL on the main server. This page has to be marked up with certain html meta tags and has to be able to process inbox requests. Because of World Wide Web arranged in such a way that everybody can get access to webpage it is important to protect private information from prying eyes. Security is provided through authorization mechanism. Every service which needs to get private entity data has to send auth key. Also there are web pages contain lists of the entities of one kind, e.g. projects. Described method is used for providing united user authorization at FRUCT web resources. If you log in at www.fruct.org you will be logged in at social network and e-werest.org and vice versa [14]. It is provided by following method. User visits one of the websites and opens an authorization form, which he filled in with his login and password and sends it to the server. Firstly the server sends HTTP-request to the main server (www.fruct.org), which keeps all information about users. This request contains user login and the server secret key. The main server checks secret key and, if it is correct, sends answer, which contains information about user with received login. Than the first server sets cookies to remember user authorization session. Thus user becomes authorized at this website. Secondly the server sends requests with user login and password to the other FRUCT websites, which also checks their correctness and sets cookies. It is necessary to authorize user straight at all FRUCT web resources. So if you successfully logged in at one of the sites, other sites will recognize you.

Case 3. Recurrent remote data access

Two described above methods allow you to connect and integrate your own services, but if your want to integrate your service with external one, these methods will not work. The important feature of FRUCT social network is integration with external web services Google Scholar [15] and Google Calendar [16]. The main difference between these services is that Google Calendar has real-time API, but Google Scholar does not. Every FRUCT member can show his public events on his profile page if he connects his profile with Google Calendar. Of course, user events are private data by default and user must give his agreement to show some of them on his page in social network. So firstly user has to click button "Connect with Google Calendar" which redirect him to google.com and than he has to confirm integration with Google Calendar, after that he will be redirected back. During this process google.com gives to social network special key associated with current user. When user chooses calendar he wants to have an ability to import events from this calendar from Google servers to social network automatically, without user participation. Described authorization protocol is OAuth, which is an open standard for authorization in Semantic Web. OAuth provides a method for clients to access server resources on behalf of a resource owner (such as a different client or an end-user). It also provides a process for end-users to authorize third-party access to their server resources without sharing their credentials (typically, a username and password pair), using user-agent redirections [17].

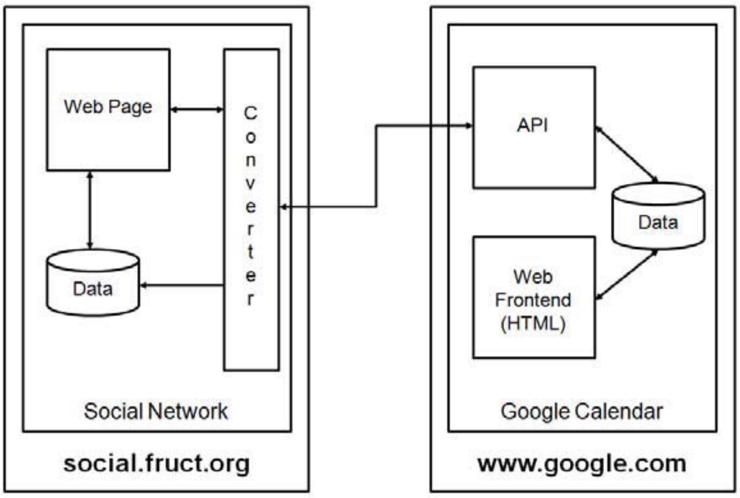

Fig. 3. Recurrent remote data access

After connecting user profile with his Google Calendar account social network service can send different requests to Google Calendar API server. For example, social network can get certain user events from calendar and than using special converter to show them on user page. It is worth noting that services can interact with each other without user intervention in real time. When you add or delete an event in your Google Calendar, this event automatically will be added or removed from your page.

Case 4. Reply parsing from remote server

Third described method is a good way to link your resource with external service, but it works only if external service has API. If it is not you have a problem which can be solved by the fourth method. Unfortunately, Google Scholar has not any API for developers, but FRUCT Social network is integrated with this service too. FRUCT social network can search its members' profiles in Google Scholar, import their h-index and i10-index values [18] and import their publications. If user has already filled in his publications in social network, it can get citation index of each publication from Google Scholar.

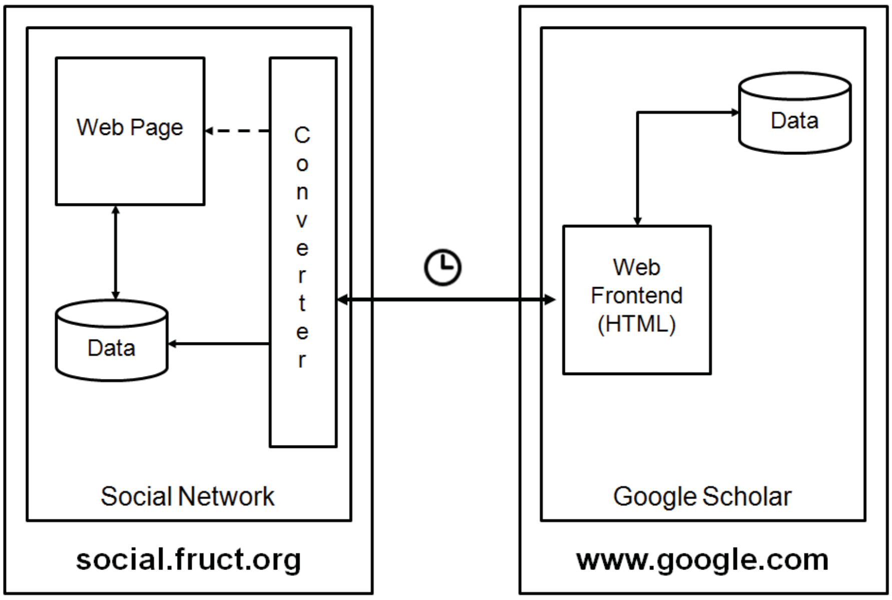

Fig. 4. Reply parsing from remote server

In contrast to Google Calendar, which has uniform interaction methods via API, Google Scholar provides only old-web methods based on html pages. So if you want to get some data from Google Scholar you need to parse HTML language using special converter base, for example, phpQuery lib [19]. It is not a very good way, because if Google Scholar pages markup is changed you will need to rewrite your code, but you don't have any another way. Because of there is no real time API you can run your converter once every day. For FRUCT social network it is not critically important problem because information in Google Scholars such as h-index is updated not very often.

So we have to admit, that Semantic Web technologies application can efficiently solve different types of connectivity problems with both internal and external resources. But the proposed approach could become even more useful when talking about communications between services distributed among the network of heterogeneous devices (like mobile phone communication with a smart card that forwards information to the desktop applications and web-based services in the cloud).

III. Using semantic web for hardware integration

Semantic Web allows to integrate different software services based on different architectures and platforms with each other. It becomes de-facto standard for modern systems. However nowadays not only humans use the Internet services but different independent devices too. The concept prescriptive exponential growth of these devices was named Internet of Things (IoT). The Internet of Things refers to uniquely identifiable objects (things) and their virtual representations in an Internet-like structure. The term “Internet of Things” was first time used by Kevin Ashton in 1999 [20]. The concept of the Internet of Things firstly became popular through the Auto-ID Center and related market analysts publications [21]. Radio-frequency identification (RFID) is often seen as a prerequisite for the Internet of Things. If all objects and people in daily life were equipped with radio tags, they could be identified and inventoried by computers [22]. However, unique identification of things may be achieved through other means such as barcodes or 2D-codes as well.

According to IoT, a lot of physical objects even very small sensors will be connected to the Internet [20]. Every device or sensor can be considered as a hardware service which can do something and exchange some information with outside. There by there is a new issue: how to connect and integrate hardware services with software ones. Semantic Web methods seemed able to solve this problem.

As can be seem from the description Semantic Web and IoT paradigms are very much relevant to each other, focusing on the same problem – service-to-service communication – from a bit different angles. The Semantic Web mainly provide application layer protocols for efficient data transfer, while IoT mostly discuss physical devices connectivity, with respect to the particular radio access network technology features. The combination of Semantic Web and IoT presents the system, where different services are distributed among the network and parameters of links between them are not equal to each other. As such, the Semantic Web technologies should be modified to be aware of possible physical, datalink and network layer problems, as well as new focus metrics, line energy-efficiency.

Thinking of example, current section presents the general architecture and technologies and protocols stack to provide alternative solution for user authentication in a heterogeneous network.

Every one well knows that almost every web resource or web service which is needed to authorize users uses text passwords. This way is obsolete, insecure and uncomfortable. Your need to think of complex passwords with big length, your passwords must be unique for different resources. Simple passwords are exposed to brute-forces attacks. So there is a problem how to remember all passwords. So there is necessity to provide new ways of user authentication. Today three ideas are offered: graphical passwords, biometric identification and wireless authentication via special device.

Graphical passwords are an alternative means of authentication for log-in intended to be used in place of conventional password; they use images, graphics or colors instead of letters, digits or special characters. One system requires users to select a series of faces as a password, utilizing the human brain's ability to recall faces easily. In some implementations the user is required to pick from a series of images in the correct sequence in order to gain access. Another graphical password solution creates a one-time password using a randomly generated grid of images. Each time the user is required to authenticate, they look for the images that fit their pre-chosen categories and enter the randomly generated alphanumeric character that appears in the image to form the one-time password. So far, graphical passwords are promising, but are not widely used. Studies on this subject have been made to determine its usability in the real world. While some believe that graphical passwords would be harder to crack, others suggest that people will be just as likely to pick common images or sequences as they are to pick common passwords [23].

Biometric methods promise authentication based on unalterable personal characteristics, but currently have high error rates and require additional hardware to scan, for example, fingerprints, irises, etc. They have proven easy to spoof in some famous incidents testing commercially available systems, for example, the gummie fingerprint spoof demonstration, and, because these characteristics are unalterable, they cannot be changed if compromised; this is a highly important consideration in access control as a compromised access token is necessarily insecure. Fingerprint identification, known as dactyloscopy, or hand print identification, is the process of comparing two instances of friction ridge skin impressions, from human fingers or toes, or even the palm of the hand or sole of the foot, to determine whether these impressions could have come from the same individual. The flexibility of friction ridge skin means that no two finger or palm prints are ever exactly alike in every detail; even two impressions recorded immediately after each other from the same hand may be slightly different [24].

Wireless authentication is a way based on some hardware device or chip card using for authorization in some system. It is a comfortable and secure method. For example, consider following situation with text passwords. User is being at his work and he needs to see something in social network. User has to open web site, fill in his name and password. If user wants to move away from his working place, he has to log off from web site. If he forget to log off, malefactor could get access to user’s private data. To prevention the same situations with FRUCT social network it is proposed to use some authorization device, for example, mobile phone as a simple device for programming. The main idea is concluded in distance between computer and mobile phone analyzing. If the distance became more, than some predetermined interval, user will be automatically logged off from social network. Bluetooth is one of the technologies for desktop and mobile phone communication [25]. In this case user does not need to care about logging off and user does not need to remember difficult passwords.

Wireless authentication is the easiest way among described above. Firstly it is reliable and secure method, because you can use very difficult auth keys, which user has not to remember (in contrast to graphical passwords method, where user has to remember images or images sequences). Secondly it does not require any additional hardware devices except widespread mobile phone (in contrast to biometric identification method, which requires hardware tool to read biometric data).

The basic scheme of this method is presented by Fig. 5 and Fig. 6. Fig. 5 reflects the system state formed after you bring your mobile phone to your computer. There are two conceptual blocks: computer and mobile phone. Mobile application keeps login and password for social network. The computer has three struct modules (programs): browser, desktop application (which monitors Bluetooth connection state) and proxy tunnel (which allows to transmit HTTP request from mobile application to social network and back).

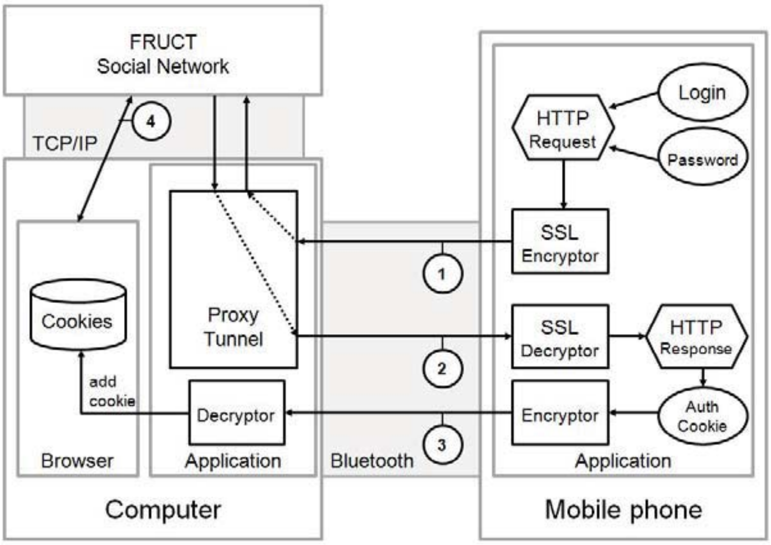

Fig. 5. Wireless key integration (connected state)

As soon as you bring your mobile phone near the computer, mobile application and desktop application established Bluetooth connection between each other. Than mobile application forms HTTP request for authorization in social network with login and password, crypts it with SSL and sends it to social network through proxy tunnel (1). Then social network returns HTTP response for mobile application (2). Mobile application pulls out an auth cookie from received response, than crypts this cookie and sends it to desktop application (3), which puts cookie into browser cookie storage. Thus browser can communicate with social network and user will be authorized.

The system state formed after you put off your mobile phone from your computer is described by Fig. 6.

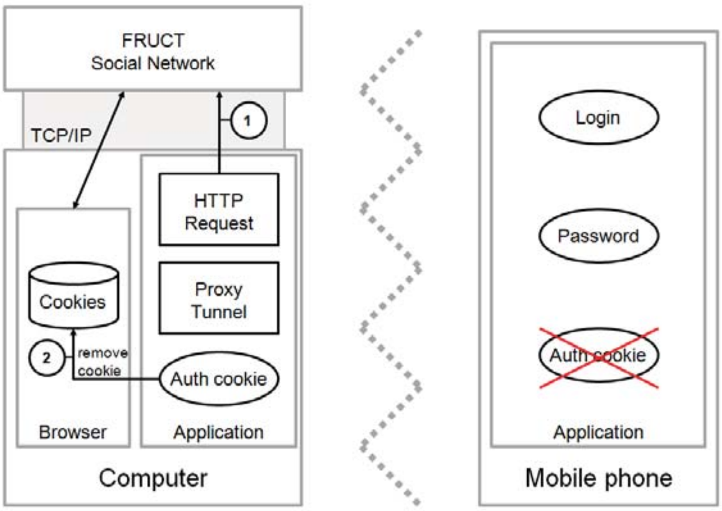

Fig. 6. Wireless key integration (disconnected state)

As soon as desktop application loses Bluetooth connection with mobile phone, it removes auth cookie from browser (1) and sends HTTP request for logging off to the social network (2). There by user can not be afraid about his account security.

So, besides the complicated architecture, there are some research and development difficulties from the device-to-device communication part, such as insufficient energy efficiency, need of high security level of all the links, delay of the user interface, etc.

However, the proposed solution in general looks very promising, because it enables user authentication to any service from any computer without providing a single bit of credentials to untrusted node (e.g. you can stop changing your e-mail password every time after accessing the web interface for the Internet cafe).

IV. Conclusion

As shown in the paper, Semantic Web is a powerful tool to solve a variety of services integration problems. The paradigm extension for Machine-to-Machine and Machine-to-Service communications rapidly increases the usability level of complicated systems for end users. However, in this case, the implemented solution should be aware of possible bottle-necks in the network from physical, link and network layer perspective. In particular, existing technologies have to be redesigned in order to provide fast, secure and energy-efficient information exchange through untrusted and unreliable network.

References

[1] T. Berners-Lee, J. Hendler, O. Lassila, “The Semantic Web,” *Scientific American*, 2001.
[2] T. Berners-Lee, J. Jaffe, “World Wide Web Consortium,” http://www.w3.org/, 2013.
[3] I. Herman, “An Introduction to Semantic Web (Tutorial),” *17th International World Wide Web Conference*, 2008.
[4] N. Shadbolt, W. Hall, T. Berners-Lee, “The Semantic Web Revisited,” *IEEE Intelligent Systems*, 2006.
[5] A. Collins, M. Quillian, “Retrieval Time from Semantic Memory,” *Journal of verbal learning and verbal behavior*, 1969.
[6] A. Collins, E. Loftus, “A Spreading-activation Theory of Semantic Processing,” *Psychological Review 82*, 1975.
[7] T. Berners-Lee, M. Fischetti, “Weaving the Web,” *HarperSanFrancisco*, 1999.
[8] S. Balandin, FRUCT, “Finnish-Russian University Cooperation in Telecommunications,” http://fruct.org/, 2007.
[9] V. Kirkizh, E. Dashkova, V. Petrov, “Distributed Social Network Services for the FRUCT Community,” *10th FRUCT Conference Proceedings*, 2011.
[10] S. Balandin, FRUCT, “FRUCT Social Network,” http://social.fruct.org/, 2007.
[11] S. Balandin, E-WeREST, “East-West Research and Education Society on Telecommunications,” http://e-werest.org/, 2007.
[12] V. Benjamins, J. Contreras, O. Corcho, A. Gomez-Perez, “Six Challenges for the Semantic Web”, *Intelligent Software Components*, 2002.
[13] B. Douglas, S. Torsten, “Solving the Java Object Storage Problem,” Computer, vol. 31, no. 11, 1998.
[14] V. Kirkizh, M. Komar, V. Petrov, “Integration of the FRUCT Social Network with Other FRUCT Web Resources,” *12th FRUCT Conference Proceedings*, 2012.
[15] Google Inc., “Google Scholar,” http://scholar.google.com/, 2004.
[16] Google Inc., “Google Calendar,” http://google.com/calendar/, 2006.
[17] E. Hammer, “Introducing OAuth 2.0,” http://hueniverse.com/2010/05/introducing-oauth-2-0/, 2010.
[18] G. Woeginger, “An Axiomatic Characterization of the Hirsch-index,” *Mathematical Social Sciences*, 2008.
[19] The PHP Group, “The PhpQuery Library,” http://php.net/manual/en/pdo.query.php, 2013.
[20] K. Ashton, “That 'Internet of Things' Thing,” *RFID Journal*, 2009.
[21] G. Johnson, S. Bushell, “M-commerce key to ubiquitous internet,” *Computerworld*, 2000.
[22] P. Magrassi, T. Berg, “A World of Smart Objects,” Gartner research report R-17-2243, 2002.
[23] R. Morris, K. Thompson, “Password Security: A Case History,” *Bell Laboratories*, 1978.
[24] N. Quinche, P. Margot, “A Precursor in the History of Fingermark Detection and their Potential Use for Identifying their Source,” *Journal of Forensic Identification*, 2010.
[25] J Vainio, “Bluetooth Security,” *Helsinki University of Technology*, 2000.