SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Ing. Radovan Semančík

Revised World Wide Web Architecture

Dissertation

submitted in partial satisfaction of the requirements for the degree of

philosophiae doctor

in

25-21-9 Computer Tools and Systems

Bratislava, December 2008

DEDICATION

To Katka Stanovská, my love and partner, for incredible support, care and encouragement. Without her this work would be far from possible.

Also to doc. Ing. Margaréta Kotočová, PhD. who took the role of supervisor during my study. I appreciate the knowledge and experience that guided me in my research and helped to significantly improve quality of this work.

Finally to all my colleagues at nLight for maintaining a creative and supportive environment and for engaging in discussions that helped to refine my thoughts.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Introduction

The interaction with computer system is an important part of our lives. The computers frequently store and use data that describe characteristics of physical human beings. Yet only a few systems consider the specific nature of personal information in their architecture.

Today's World Wide Web was built on assumptions that most of the shared information is public. It also expects that only a very limited number of subjects will publish and modify the information while many subjects will read it. However, such situation will not hold for much longer. Many applications build on top of the World Wide Web are actually making it *writable*. Wiki applications make it possible for a large community to modify the same resource. Blogging, photo and video sharing applications allow to add comments to the resource, therefore influencing its state. The original World Wide Web architecture has not anticipated that most of the resources will be dynamic and many of them non-public or semi-public.

This paper provides a basic structure of a model that may provide insight into some of the areas that touch personal data. Some concepts in the problem area are understood only intuitively and this work attempts to provide more formalized definitions and it attempts to put these concepts in broader perspective. The work was motivated by a desire to design an “identity layer” on top of current World Wide Web architecture. However, that approach was changed in the process of modeling the system and analyzing the basic structure of the World Wide Web. We have discovered that basic identity-related mechanism must be integral part of the World Wide Web architecture to be effective. This work describes the motivation and the reasons for such conclusion and it provides a proposal of the necessary changes to the architecture of the World Wide Web.

The famous cartoon [1] provides a good description of the current status quo: “On the Internet, nobody knows you are a dog”. But the situation is more complicated: on the Internet nobody knows that you are *not* a dog, therefore you should not be forced to live in a doghouse. Efficient cooperation requires trust, which in the case of our analogy translates to the confidence that the other party is not a dog. Currently, such a confidence can be achieved on the Internet only by a long-term relationship. Considering a short attention span of current society, reliance on a long-term relationship to build up trust is hardly the ideal path for the Internet. Challenged by this problem, one of the important goals of this work was to propose an improvement of the effectiveness of ad-hoc interactions on the Internet.

This works starts by describing the state of the art related to World Wide Web architecture, identity management and Internet security in chapter 1. The problems of current World Wide Web and related technologies are also described in this chapter. Chapter 2 follows up with summarization of the problem, definition of dissertation objectives and non-objectives. The primary objective of the work is to correct the inconsistencies in World Wide Web architecture and improve the architecture to better suit today's needs.

No meaningful architecture can be conceived without understanding of the basic concepts of the problem space and reflecting the needs and habits of people that seek to harvest the benefits of the architecture. Therefore a model that described the basic undercurrents of the architecture is provided in chapter 3. The model describes interaction between two worlds: the physical world of *realspace* where all the tangible objects and subjects exist and the virtual world of *cyberspace* where intangible software components interact. The interactions between these two worlds are discussed, especially focusing on the relevance and reliability of the exchanged data.

Design methods and desired environment is discussed in chapter 4. Focus of the discussion is on the environment that can support efficient cooperation and privacy of the users. The result is an outline of desired environment and a reputation-based mechanism for supporting cooperation while maintaining acceptable level of privacy.

The model of realspace-cyberspace interactions and results of the environment discussion are applied to current architectural concepts of the World Wide Web. The internal inconsistencies of WWW architecture as well as the problems uncovered by the application of our model are discussed in chapter 5.

Chapter 6 describes architectural principles to guide the design of World Wide Web. A new architectural style named RRSS is described. This style is combined with other architectural styles to create a revised World Wide Web architecture. The proposed architecture is divided to several levels of abstraction. It reflects the implications of the model described in chapter 3 and attempt to avoid the problems identified in chapter 5. The focus of the work is on the basic architectural principles, while only an outline of the proposed architecture on protocol specifications is provided. However a specific proposal for improvement of the current World Wide Web architecture and a migration path outline is provided at the end of the chapter. The architecture is validated at the end of the chapter using a scenario-based validation approach and the known drawbacks of the proposals are discussed.

The dissertation concludes by discussing the results and summarizing the contributions to the field of Computer Science in chapter 7. The subjectivity on the information on the World Wide Web is a guiding principle in this work. We consider the information that originate in realspace and are used in cyberspace to be subjective. Therefore a source of such information must always be considered as part of evaluation of relevance of the information. This principle is implied by the model, it is later used for evaluation of current World Wide Web architecture problems and finally it is reflected to a proposed architecture.

This work makes the following contributions to the research within the field of Information and Computer Science:

- A model that describes interactions between realspace and cyberspace, especially focused on representation of personal data in the cyberspace.
- A mechanism for evaluation of anonymity and identity based on the proposed model.
- Assessment of architectural inconsistencies of the World Wide Web architecture. Both internal inconsistencies and problems uncovered by the application of the proposed model are described.

- A definition of RRSS, a new architectural style for representing information in cyberspace while taking into consideration their potential source and target in realspace.
- Application of the RRSS architectural style together with other previously described styles to a World Wide Web architecture, resulting in a proposal of improved architecture for World Wide Web.

1 State of the Art

This chapter provides outline of a current state of the art on the field of Internet identity management, World Wide Web architecture and related areas. The outline follows an organization roughly based on the level of abstraction, starting with the simplest method of Internet security and concluding by discussion of World Wide Web architecture and its problems.

1.1 Password-Based Systems

Password authentication is a traditional method of authentication. It was used in computer systems even before computer networks evolved and it is still very popular. Password authentication has many security drawbacks, but its simplicity is the primary reason for its wide usage. Passwords sent in cleartext over the network are subject to eavesdropping and replays, passwords can be stolen either from the end user or from a server database and most of them can be easily guessed [2]. Eavesdropping and replay attacks on the communication layer can be prevented by employing a good encryption mechanism, but passwords can still be compromised in client or server operating systems, before they can be encrypted. Some operating systems provide password-caching features that can be abused by an attacker to read a password cache and get all stored passwords.

It is not possible to use an encrypted channel to protect passwords in every circumstances and the static character of passwords causes different kinds of problems nevertheless. To overcome these issues, several One Time Password (OTP) schemes were proposed. The most popular one time password scheme was the S/Key scheme, developed at Belcore [3] and later standardized by IETF [4]. This scheme is based on the repeated application of the one-way function to the secret value to get the one time password sequence. Passwords from this sequence are used in a reverse order for authentication, each used only once. Password sequence contains fixed amount of one-time passwords and must be restarted when all of them have been used.

The challenge-response authentication scheme [5] has similar properties to one-time password schemes. Strictly speaking, one-time password schemes are only special instances of the challenge response schemes with fixed challenge information (sequence number, time instant, etc). The challenge-response scheme client takes some information (challenge) from the authentication server, processes it with the user-supplied secret value (password, shared secret) and returns the processed information (response) to the server. By considering the sent challenge value and received response the server can determine if the user knows the correct secret value. The most widely used challenge-response method is CHAP [6], used as a part of PPP protocol. Several token-based commercial one-time password schemes are present on commercial market. These schemes are operating on challenge-response principles, varying in key size, algorithms and the type of challenge (e.g. implicit challenge based on absolute real time).

Both one-time password schemes and challenge-response schemes in general have some common security drawbacks. When used in a plain TCP/IP environment, connection data can be manipulated after a successful authentication takes place. In this case the attacker does not need to attack the authentication scheme directly. If the authentication scheme accepts the secret value (seed) as a plain password that is chosen by the user, such a scheme is vulnerable to the dictionary attacks. Most of these schemes which are in practice are vulnerable to the man-in-the-middle attacks (Figure 1). These attacks can be prevented only by pre-authenticating the server to the client by using other independent methods and explicitly authenticating the transported data.

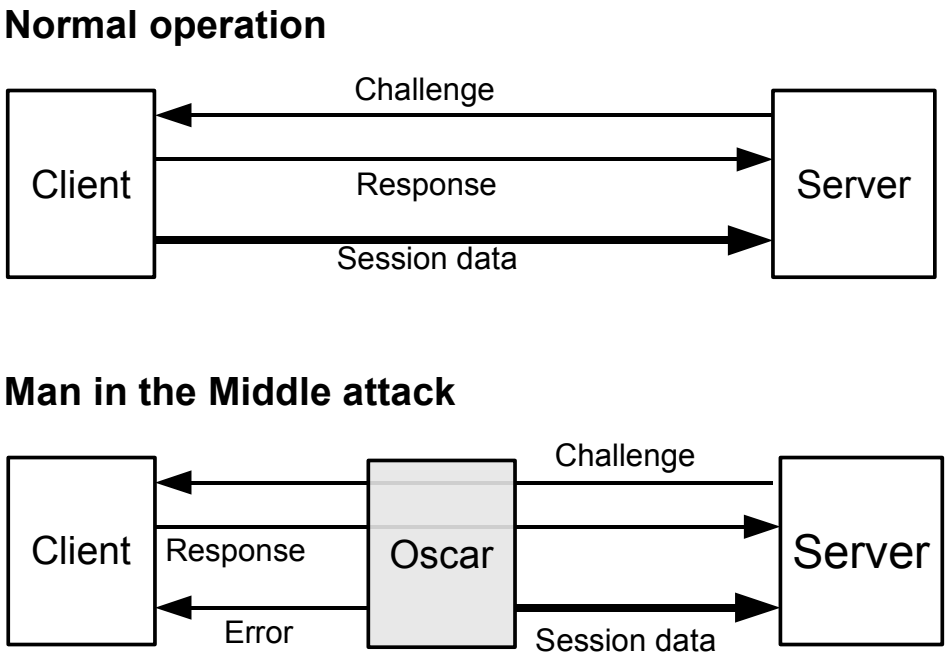

Figure 1: Man in the middle attack on challenge-response scheme

The common way of securing any password scheme is the use of an encrypted channel with an authenticated server. The Transport Layer Security (TLS) and Secure Shell (SSH) are the two most widely used methods to achieve this goal. Both of these protocols provide an encrypted channel to secure the password authentication from eavesdropping. Server authentication is achieved by employing asymmetric cryptography, either by ad-hoc methods in SSH or by using a X.509-based PKI in TLS.

1.2 Public Key Infrastructure

The Public Key Infrastructure (PKI) is a set of methods and formats for the management of public keys and all related data. PKI uses asymmetric cryptography methods to achieve its goals. Each entity in PKI has at least one asymmetric key pair. The identity of the entity is bound to the entity's public key by the Public Key Certificate (PKC). This certificate is a data structure that contains the entity's identification, certificate validity period, the identification of the certificate

issuer and any other data describing the certificate and its use. The PKC also contains the entity's public key and the entire PKC is signed by the certificate issuer's private key.

There are some infrastructures that do not impose any limits on which entity may issue certificates and which may not. These infrastructures are usually called 'Web of Trust' and their trust structure forms a generic directed graph. A good example of this kind of structure is PGP. Other infrastructures limit certificate issue privileges only to selected entities. These entities are called *certificate authorities* (CA) and their role is the management of certificates issued to *end entities* (EE). Certificate authorities may issue certificates to each other, expressing trust relationships. Trust structure of this PKI is hybrid: the trust relationship between the end entities and their certificate authorities forms a directed tree, but the relationships of different certificate authorities can form any generic structure.

1.2.1 Public Key Infrastructure Based on X.509

The international standard for the public key certificate format is based on ITU-T X.509 recommendation [7] and is widely used in both enterprise and Internet environments. Other formats evolved as internal parts of specific applications, but use of these proprietary formats yields in favor of X.509 certificates.

The certificate authority affirms certificate validity by its signature. However there are situations that the certificate validity must be terminated, for example due to private key being compromised. Each X.509 certificate contains a fixed validity period, but in the practice it is not possible to wait until the end of the validity period to invalidate the certificate. A mechanism must exist to revoke the certificate anytime during its validity period. The certificate authority publishes a list of certificates for this purpose that had to be revoked before the end of their validity period. This list is called the Certificate Revocation List (CRL) and is published at regular intervals or anytime it is needed. CRL is protected by the certificate authority signature so it can be stored in an insecure environment. The entity that is checking the certificate validity should locate the appropriate CRL and check if the certificate is not listed there. However the CRL method of the certificate validity verification may be insufficient for certain applications that require on-line certificate validation. These applications could use on-line validation protocols such as Online Certificate Status Protocol (OCSP) [8].

Version 3 of the X.509 recommendation allows the use of certificate and CRL extensions. These extensions can be used to specify an additional information, such as the certificate use constraints, subject and issuer alternative names or almost any other information. Use of the extensions considerably enhances flexibility, but different, non-interoperable, implementations of the same basic mechanisms appeared. Each of these implementations understands a different set of extensions and even if they agree on a common set, they interpret the extension values in a different way. To promote interoperability of certificate processing systems, the national organizations and standard bodies publish X.509 certificate profile documents. These certificate profiles specify the exact meaning of certificate extensions, rules for certificate processing and so on. The IETF published the X.509 profile for use in the Internet environment [9].

1.2.2 Transport Layer Security

Public key certificates can be used in many communication systems on the Internet. The most common communication protocol in use today that employs X.509 public key certificates is the Transport Layer Security (TLS) Protocol [10]. It was originally developed at the Netscape Communications corp. in 1994 as the Secure Socket Layer (SSL) protocol. The first publicly available version of SSL was version 2 [11], which suffered from several major problems [12]. That version was later updated to version 3 [13] with most of the SSv2 security problems eliminated. The SSL protocol version 3 was taken by IETF as the base for the TLS protocol version 1.0. The TLS and SSLv3 specifications have some minor differences that implementers should take care of to assure compatibility. Compatibility is not straightforward, but protocol versions could be correctly detected by examining initial protocol messages. The SSLv2 is not compatible with SSLv3 nor TLS, but implementation could support all these three protocol on the same TCP port. Nevertheless implementers are encouraged to use the TLS protocol or at least the SSLv3 protocol instead of SSLv2 whenever possible.

The TLS protocol does not depend on any specific cryptographic algorithm. The communicating parties can negotiate the best common cryptographic algorithm suite for secure communication. The protocol is asymmetric, the client is the connecting party while the server passively listens for connections. TLS supports several authentication modes:

- **Total anonymity**. Neither the server nor the client is authenticated, no key material origin is assured. This authentication mode is possible with TLS but its use is strongly discouraged.
- **Authenticated server.** The server authenticates to the client by presenting its public key certificate and providing proof of possession of the appropriate private key. Client is still anonymous, but the key exchange can be accomplished securely.
- **Mutual authentication.** Both server and client are authenticated to each other by using their respective public key certificates and appropriate proofs of possession of the private keys.

The most frequently used TLS mode today is the authenticated server mode. The server has an X.509 certificate for its fully-qualified domain name (FQDN) and the client (web browser) has a list of trusted certificate authorities. The client authenticates the server using its X.509 public key certificate and both the server and the client negotiate a session key. When the secure communication channel is set up, the client authenticates using a password, challenge-response system or whatever mechanism is appropriate. This client authentication must be processed on the application layer, the TLS layer is not aware of it happening.

The total anonymity mode is included for backward compatibility only and for any obscure application that may require it. The key material origin is not authenticated in this mode and the Man-in-the-Middle attack is possible. Use of this mode for whatever reason is not recommended.

If TLS is used in any authentication mode, it provides only short-term security for the transported data - TLS protects and authenticates data on the communication channel only. If the data is stored on the target system, they are no longer protected. Even if the

data block was received by the server in a mutually authenticated TLS connection, server is not able to provide any proof of data origin to the third party. Use of standalone digital signatures is recommended in addition to TLS to achieve such long-term protection requirements.

The most common practical use of the TLS protocol is to secure a WWW communication. It is called HTTPS [14] and it is essentially HTTP protocol communicating over a TLS-secured channel. Most of the connection-oriented protocols can be modified in a similar way to use the TLS layer for protection. Use of TLS to secure IMAP, LDAP and other TCP-based protocols is becoming quite common on the Internet.

1.2.3 Public Key Infrastructure Issues

Public key infrastructure is not well-suited to express dynamic trust relationship. The digital certificates used by public key infrastructures are relatively long-lived, they are difficult to update and revoke. Therefore the public key infrastructure is used to express relations that are quite static, such as binding name to an entity or to express contractual relations between organizations.

Attempts to use client-side certificates haven't gained any considerable user adoption. A low penetration of qualified certificates as authentication method for government services in European countries is a clear indication of this phenomenon [15]. High cost of the certificates and low benefits may be one of the reasons of low PKI adoption.

Except for low adoption there are privacy concerns of PKI usage [16]. Once the certificate is presented, all the attributes and data in the certificate are disclosed to the verifying party. That means that the certificates should contain only minimal amount of information to conserver user's privacy. However, such minimal certificates have limited usability, as they does not contain any additional data (such as user's name or address). The certificates can solve authentication problems, however they will not help in maintenance of user's profile. In addition to that the public key in the certificate is (by definition) globally unique value. Therefore the public key (or its hashed value) can be used as a globally-unique identifier of the user. Such value can be used to correlate user's activities at different sites, revealing more data about the user than the user chooses to disclose.

Digital credentials [17] is a cryptographic system based on secret-key certificates that was designed to overcome some PKI problems. Although the basic principles of the system are documented, the communication protocols and the details of usage are not. Therefore it is not possible to thoroughly evaluate digital credentials at the time of writing this document.

1.3 Internet Identity Systems

The World Wide Web (WWW) and WWW applications are the most popular methods of Internet usage. World Wide Web evolved from a simple distributed hypertext to a complex system of hypermedia applications. The common approach is to implement WWW applications in a single WWW *site*, using internet and WWW only as a communication medium between user and the application. This approach results in application (or site) maintaining its own instances of support subsystems, such

as security and user management. The result is that users are forced to create *user accounts* on most of the WWW sites. This leads to the duplication of effort, risks of data inconsistencies, lost and forgotten passwords and so on. There are also security risks. Users typically maintain few tens or hundreds of accounts. It is unrealistic to assume that the user will choose different password at each site and remember them all. Usual approach is to choose the same approach on most of the sites, giving the sites to impersonate the user on other sites. Most passwords are also transferred to the application in the clear, motivated by the desire to avoid cryptography overhead by site owners.

The goal of Internet identity systems is to move the maintenance of user accounts to dedicated and shared sites, therefore lowering the number of accounts a user has to maintain. The mechanism used by such systems is a secure transfer of user's identifiers, attributes and current authentication status of a user from source site (Identity Provider) to the destination site (Service Provider). For example, user may have established authentication session with source site. The Internet identity system is used to transfer that session status to the target site, so that that site can establish similar session with the user (Figure 2). The source and target sessions need not to be the same, they may differ in session tracking mechanism (cookies, URL parameters, etc.) and there may be different user identifiers, policies or any other session parameters. The session at source site may not even exist at the start of the process, it may be created by source site on demand.

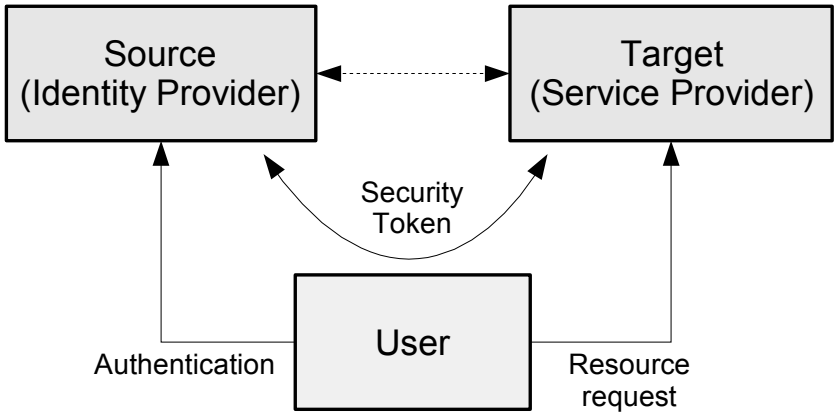

Figure 2: Internet identity system structure

The trust relationship must be established between source and target sites for a source site to trust the target site's requests and for a target site to trust the source site's identity statements. Establishing and maintaining this trust relationship is out-of-band for most Internet identity systems.

The Internet identity systems support one or both of the following mechanisms:

- **Browser-based mechanism:** the system uses only the features of a stock WWW browser, without a need for any additional client software.

- **Client-based mechanism:** the system uses client software installed on user's workstation.

Some Internet identity systems distinguish themselves as either *user-centric* or *federation* systems. The user-centric systems are focused on a common Internet user that is not affiliated with any major organization, e.g. a user that is not presenting himself as an employee of a commercial company. The federation systems focus on users that are already affiliated with a major organization, allowing them to seamlessly use services in other organizations. However, such distinction is only conceptual and does not imply any major technical differences in the systems. Therefore we will not explicitly consider the distinction between user-centric and federation systems.

The described Internet identity systems follow the proxy-based true SSO model according to Pashalidis and Mitchell [18].

1.3.1 Terminology

The individual Internet identity systems use different terminology to describe their operation. For the purpose of consistency we first define common terminology:

- **Source site** or Identity Provider is a system that maintains information regarding user (authenticated session, attributes, etc.) and is willing to transfer that information to other sites.
- **Target site** or Service Provider is a system that is willing to receive information about user. Receiving the information does not necessarily imply trust in that information.
- **User** refers to the physical person who is interacting with the computer system. The User is a virtual entity for a computer system that is represented by persona (or personae).
- **Persona** is a digital representation of user's characteristics. User may maintain several personas that may be more or less related to each other. The characteristics of persona are represented in the form of attributes.
- **Account** is a data structure that is usually kept in the computer system databases. The account is used for access control purposes, storing attributes, credentials, etc. Account is usually used as a persistent storage for (partial) persona attributes, but it also may be unrelated to any physical user or persona (e.g. system account).
- **Persona identifier** is a value of persona attribute that uniquely identify a persona in a given scope or context. In the most common case these identifiers are in form of short strings representing a username, but may have hierarchical structure of LDAP distinguished name or may be just a random binary value.
- **Pseudonym** is an alternate identifier for a persona. Pseudonyms belong to specific persona and are typically long-term identifiers. The persona to pseudonym mapping is in most cases private information, the persona to which a specific pseudonym belongs is not publicly known.

Simple Subject-Persona-Account structure is depicted on Figure 3. This is just an example of tree-like persona structure. The data stored in accounts may itself act as a

persona (or personas) and the structure may get much more complicated. Example of such structure is illustrated on Figure 3. Note that persona is an abstract representation of characteristics and that in practical implementation any persistent persona will need an account-like structure (or structures), where the data will be physically stored.

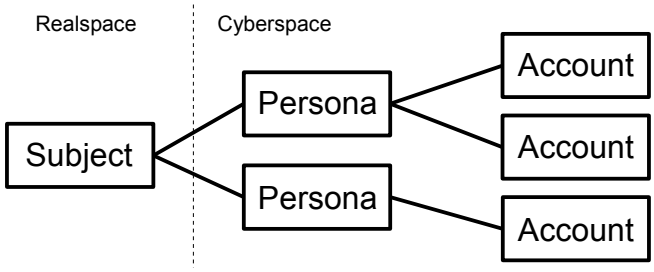

Figure 3 Simple Subject - Persona - Account structure

1.3.2 Browser-Based Message Exchange

Internet identity systems that use browser-based mechanism employ similar mechanism to transfer authentication status from source to destination site. In all these cases, browser redirection or form processing capabilities are used to transfer security tokens between sites. The process is illustrated in Figure 4 and it consists of following steps:

1) The user **requests resource** on target system (service provider) using his WWW browser. For this example we assume that there was no prior user interaction with the target site.
2) Target site does not recognize the user (has no valid session for the user/persona). The target site constructs the **authentication request** and returns it to the user's browser in the response. The response is returned in the

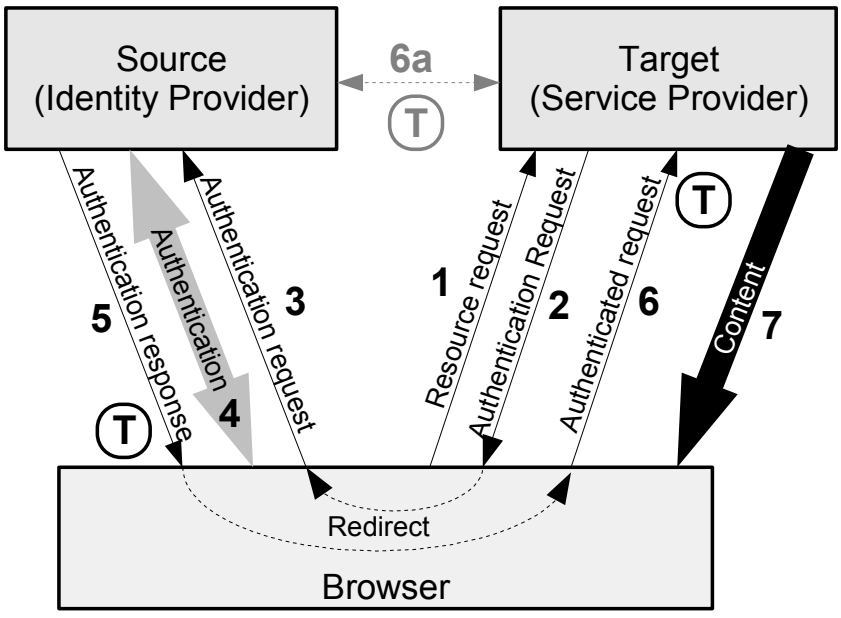

Figure 4: Browser-based SSO message exchange

form of HTTP redirect or HTML form that will redirect user interaction to the source (identity provider) site.

3) The **authentication request** is received by source site (identity provider). The source site processes the request, and applies any relevant policy.
4) The source site may **authenticate** the user, if not already authenticated or if any policy or the request requires re-authentication.
5) The source site constructs the **authentication response**, which contains the results of persona identity evaluation. The authentication response may contain a security token that will prove persona identifier and/or attributes to the target site. The authentication response is returned in the HTTP response to the user's browser in a form of HTTP redirect or HTML form that will redirect user interaction back to the target (service provider) site.
6) The authentication response is received by the target site. The response is processed and the security token is evaluated. For the response and token processing it may be necessary to contact source site directly (6a), for example to resolve references in the response. Note that the token itself may be passed by reference in the authentication response and it may be needed to dereference it by direct communication to the source site. After the response and any related security tokens and processed the persona identifier and/or attributes are determined.
7) The target site applies any relevant policies to the original access request (step 1) combined with the information determined in step 6. If the request is allowed, the target site will in most cases establish a local session with the user's browser. The local session will help avoid quite significant overhead of future re-authentications.

1.3.3 Client-Based Message Exchange

Internet identity systems that use client-based mechanism employ similar mechanism to transfer authentication status from source to destination site. The processing and transfer of security tokens is controlled by identity client installed on user's workstation. The process is illustrated in rough details in Figure 5. It consists of following steps:

1) The user **requests resource** on target system (service provider) using his WWW browser. For this example we assume that there was no prior user interaction with the target site.
2) Target site does not recognize the user (has no valid session for the user/persona). The target site constructs the **authentication request** and returns it to the user's browser in the HTTP response. The HTTP response is returned in a special form understandable by the identity client. The browser will recognize the format and pass it to the identity client.
3) Identity client evaluates the authentication requests, selecting appropriate source site (this may include interaction with user). It will then forward the authentication request to the source site. The **authentication request** is received by source site (identity provider). The source site processes the request, and applies any relevant policy.

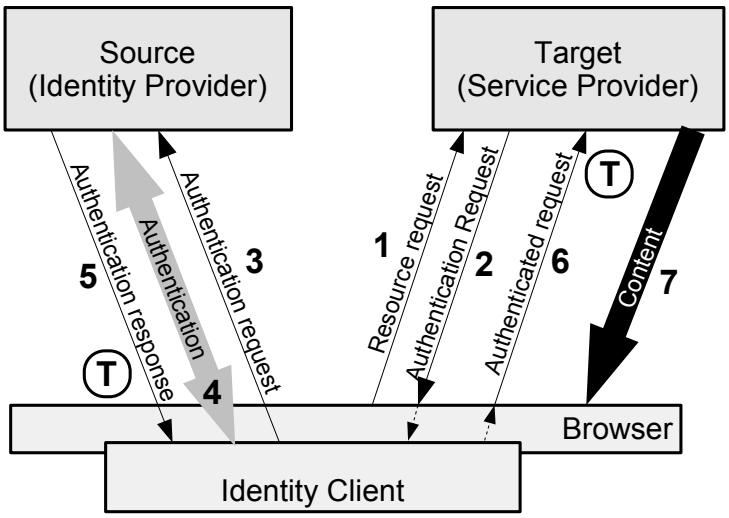

Figure 5: Client-based SSO message exchange

4) The source site may **authenticate** the user, if not already authenticated or if any policy or the request requires re-authentication.
5) The source site constructs the **authentication response**, which contains the results of persona identity evaluation. The authentication response may contain a security token that will prove persona identifier and/or attributes to the target site. The authentication response is returned in the response to the identity client. Identity client will process the response, extract security token and optionally cache it for later use. The client will add the security token to the HTTP request for target site, creating authenticated request for a resource.
6) The authenticated request is received by the target site. The request is processed and the security token is evaluated. After the security tokens and processed the persona identifier and/or attributes are determined.
7) The target site applies any relevant policies to the original access request (step 1) combined with the information determined in step 6. If the request is allowed, the target site will in most cases establish a local session with the user's browser. The local session will help avoid quite significant overhead of future re-authentications.

1.3.4 Internet Identity Systems Overview

The Internet identity systems considered in this document are summarized in Table 1. Following sections describe principles and methods that different Internet identity systems employ and how it fulfills the requirements defined earlier. Only an overview of the systems architecture is given and it is focused on authentication and single sign-on features only. Attribute services are considered only marginally. Persona linking methods and overall fitness of the system for the Internet environment is evaluated.

System	Origin	Version	Date	Status	Type
Liberty ID-FF	Liberty Alliance Project	1.2	November 2003	Operational	Browser Client
WS-Federation	BEA, IBM, Microsoft, RSA Security, Verisign	1.1	December 2006	Operational	Browser Client
"Identity Metasystem"	Microsoft	1.0 (profile)	April 2007	Operational	Client
OpenID	OpenID Foundation	2.0	December 2007	Operational	Browser
Shibboleth	Internet2 Project	1.3	August 2005	Operational	Browser
LID	NetMesh	3 Jan 2005	January 2005	Discontinued	Browser
SXIP	SXIP Identity	1.0.4	September 2004	Discontinued	Browser

Table 1: Overview of Internet identity systems

1.3.4.1 Liberty Identity Federation Framework (Liberty ID-FF)

The Liberty Alliance Project is a group of industry and non-commercial organizations whose objective is to prepare an open standard for the network identity systems. Decentralization and openness are the main goals of the alliance, their effort aims at providing federated identity. The Liberty Identity Federation Framework (ID-FF) specification set [19] [20] is a product of the first phase of the Liberty Alliance Project. It defines a Single Sign-On system with support for federated identities, including a mechanism for transfer of persona attributes.

The Liberty ID-FF follow the generic model described is sections 1.3.2 and 1.3.3. It uses Security Assertion Markup Language (SAML) [21] assertions as security tokens. Assertions are either passed directly in the HTML forms or are referenced by smaller artifacts. Liberty ID-FF specifications define several profiles for different combinations of these techniques, including a client-based mechanism (Liberty-enabled client proxy).

The Liberty ID-FF supports pseudonymity as a default behavior. When creating persona identifiers for target systems (`NameIdentifier` tag in SAML assertions), it is required to be a pseudo-random value that have no discernible correspondence with the original persona identifier. The linking of personas is carried out by associating pseudonyms, not primary persona identifiers, as illustrated on Figure 6. It is also possible to make linking chains by linking accounts managed by different Identity

Providers, which will enable to form a higher-level structure similar to Certificate Authority cross-signing in X.509 based PKIs. The relations are always implemented by linking a pair of identifiers (pseudonyms) combined with the provider identifiers, so there is no need for a global persona identifier space.

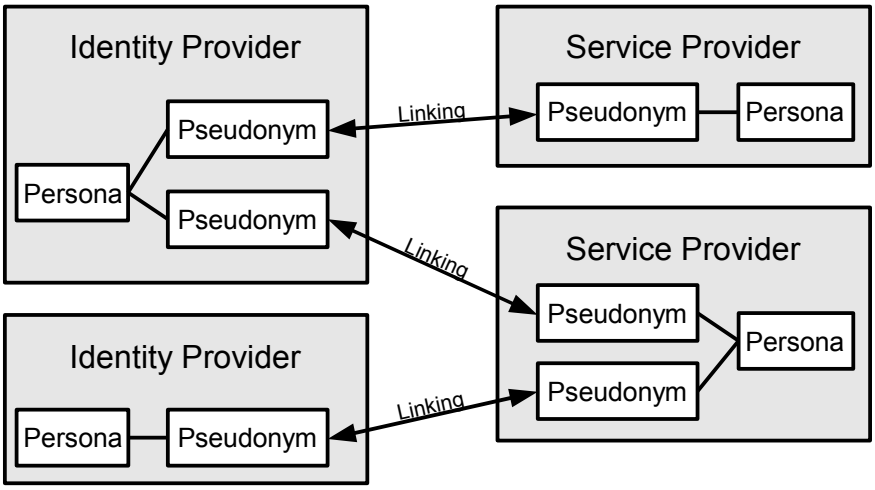

Figure 6 Liberty persona linking

1.3.4.2 Web Services Federation Language (WS-Federation)

The Web Services Federation Language (WS-Federation) specification [22] defines a mechanism for identity federation. Its primary focus is on federating services (computer-to-computer interactions), but it also addresses Single Sign-On issues in WS-Federation: Passive Requestor Profile specification [23]. The WS-Federation specifications builds on other documents, especially on WS-Trust [24] and WS-Security [25] and are published as a public drafts.

The WS-Federation Passive Requestor Profile uses the general mechanisms described in section 1.3.2. The exchanged messages are XML formatted according to the WS-Trust specification, with several additional service attributes. The sites may use URL references instead of direct message exchange.

No specific format for security token is mandated by WS-Federation specifications, the security token formats are specified by WSS: SOAP Message Security (WS-Security) profiles. At the time of writing this document, the security tokens described in Table 2 were defined.

The source site (Requestor IP/STS) may also provide attribute and pseudonym services. However, the use of pseudonyms is not mandatory and no strict pseudonym models are defined. Some of the possible persona linking scenarios are illustrated in Figure 7.

Security Token	Description
Username	Provided for simple username/password authentication
X509	Provides authentication by X.509 certificate
REL	Provides support for Rights Expression Language
SAML	Provides support for SAML assertions
Kerberos	Provides support for Kerberos tickets

Table 2: WS-Security security token profiles

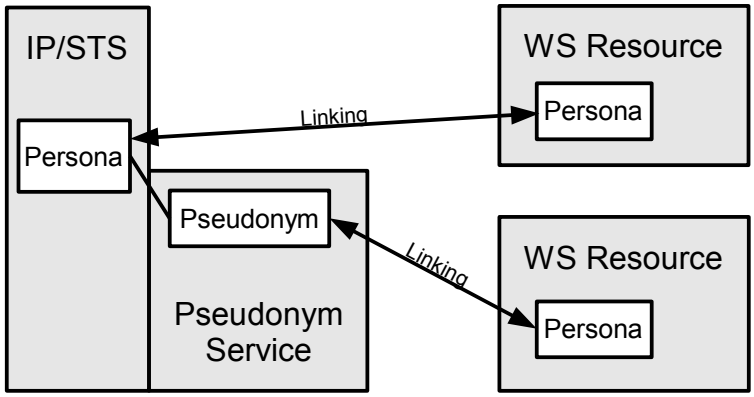

Figure 7 WS-Federation persona linking

The WS-Security specifications leave a lot of details to the implementer's decisions and to be defined by the service policy. Although that is good for flexibility, it brings additional degree of uncertainty to the system. The implementing systems may not be interoperable by implementing different subsets of specifications and/or using non-compatible policies.

The privacy decisions (e.g. use of pseudonyms) is left to the implementers. This may in practice lead to the implementations, that will not adhere to the best practice and the level of privacy in WS-Federation-compliant systems may be lower (in average).

1.3.4.3 Identity Metasystem

Identity metasystem is a name for a Microsoft initiative to create interoperable Internet identity system. It is an architecture based on client-side identity agent called *identity selector*. Microsoft provides implementation of identity selector in its Windows operating system, marketed under the CardSpace name. The whole technology is sometimes referred to as *information cards*.

There is no formal specification of the identity metasystem. Few informal articles exists, however they fall to provide the complete picture of the system. The only document that can be considered a specification of the identity metasystem is Identity Selector Interoperability Profile [26] published by Microsoft.

Identity metasystem is based on WS-Security [25], WS-Trust [24], WS-MetadataExchange [27], and WS-SecurityPolicy [28] specifications. It follows the general principle of client-based Internet identity system described is section 1.3.3. It assumes that the implementation of identity selector will interact with the user, giving user chance to select appropriate persona for the interaction.

Identity metasystem does not mandate any specific structure of personas and/or accounts.

1.3.4.4 OpenID

OpenID is a simple Internet digital identity system. It provides single sign-on services that can be used in WWW applications that do not require strong security. OpenID authentication protocol [29] roughly follows the browser-based mechanism described in section 1.3.2. However it deviates in firsts steps, adding communication between target site and source site prior to the main redirect sequence.

OpenID does not mandate any protection against man-in-the-middle attacks. It defines a mechanism based on Diffie-Hellman key agreement [30], however the mechanism as used in OpenID is useless. Because the public keys of Diffie-Hellman key agreement method are not authenticated, the protocol is open to a simple man-in-the-middle attack. The only practical countermeasure available for OpenID-compliant system is the use of HTTPS [31] protocol. The use of HTTPS is recommended by OpenID specifications, however it is still only optional part of the protocol. The use of HTTPS with OpenID is in itself sufficient to secure the OpenID exchange at the same level as would be provided by Diffie-Hellman key agreement method, rendering the Diffie-Hellman mechanism redundant.

OpenID operates with URL [32] or XRI [33] identifiers as globally-unique user identifiers. It assumes that the identifier can be dereferenced and that the controller of the identifier is authoritative data source for the object that the identifier identifies. While this assumption usually holds in the current WWW environment, it may not hold under all circumstances. For example the corner cases, vulnerabilities of identifier dereferencing and the high susceptibility of users to be confused in WWW navigation and trust are exploited by so called *phishing* attacks on OpenID [34]. The architecture, user interface and design decisions in OpenID design make OpenID-compliant systems very susceptible to this type of attacks [35].

OpenID does not mandate any specific structure of personas and/or accounts. However, it is obviously expected by OpenID designers that the primary way of OpenID usage is the use of single (or few) globally-unique user identifiers. However such approach will endanger user's privacy if several of target sites collude to gain information about the user.

1.3.4.5 Shibboleth

Shibboleth [36] is a web single sign-on and attribute exchange system built on SAML specifications [21]. It follows the browser-based mechanism described in section 1.3.2. It uses a modified SAML protocol for communications and SAML assertions as security tokens. Shibboleth adds optional WAYF service for identity provider selection.

The specifications do not limit the use of NameIdentifier types in the SAML assertion, but defines Shibboleth-specific transient name identifier. Shibboleth

specification recommends that if such transient identifier is used, it should be used only once. Transient identifiers can be used for transient persona linking, and may be used for subsequent attribute exchange using shibboleth attribute services. The structure of persona linking in Shibboleth is depicted in Figure 8.

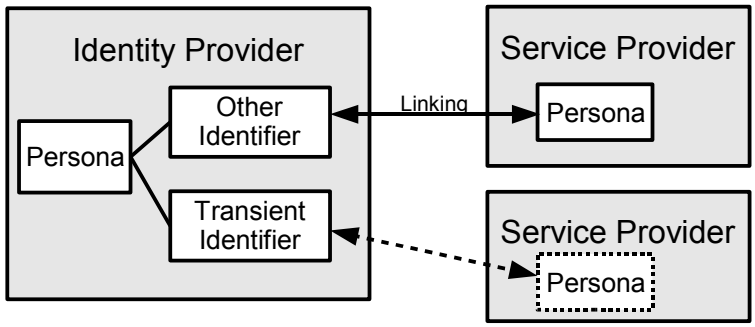

Figure 8 Shibboleth persona linking

The system implemented using Shibboleth specifications will need to specify a lot of local details, e.g. name identifier types, linking policies, etc. This type of flexibility may lead to situation that two shibboleth-compliant implementations will not interoperate. The shibboleth will depend on other specification to define persistent persona linking, if such will be needed. The use of transient name identifiers allows good degree of privacy, but for any practical purpose it will require a solid attribute service.

1.3.4.6 SXIP Network

SXIP Network is open identity network that is based on the Simple eXtensible Identity Protocol (SXIP) [37]. The SXIP Network provides single sign-on and attribute services to the participating sites. SXIP follows the principle similar to the browser-based mechanism described in section 1.3.2, but it introduces a central Rootsite that manages global persona identifiers. The SXIP protocol defines two communication options:

- *Simple Commands* provide simple, parameter-based method. The protocol data are transferred in HTTP parameters or HTML form fields. This method does not include any security token or any other form of intra-protocol security measure.
- *XML Commands* provide richer, XML-based interface. The exchanged messages are represented in SXIP Markup Language (SxipML) [38]. This method uses XML Digital Signature [39] element as a security token, but it is not included in all messages.

Personae are identified by 64-bit Globally Unique Persona Identifier (GUPI) assigned by the Rootsite during persona registration process. The GUPI is used as an universal identifier for a persona at all target sites (membersites). This model is illustrated on Figure 9.

Protocol variation using simple commands provides only minimal security. No intra-protocol security token is passed, the security is left to the external means. Even the use of HTTPS does not add any real security to the simple command exchange. The simple command exchange should be considered dangerous for most deployments and all simple (non-xml) SXIP commands should be disabled in production deployments.

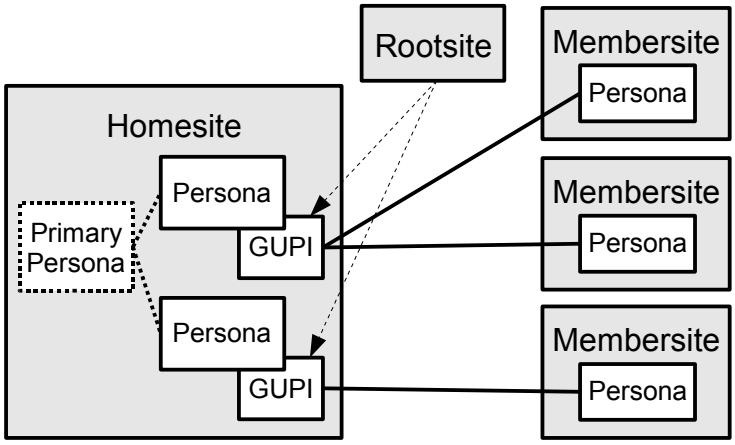

Figure 9 SXIP persona linking

Some of the SXIP XML commands include XML digital signature element that should protect the integrity of SxipML message. However, no specific methods or guidelines are documented for creation and validation of these signatures.

The `storex` and `fetchx` command messages are not authenticated, which may lead to the implementations that may allow anyone reading or setting arbitrary persona attributes.

The use of globally unique GUPI at several membersites makes it easy for the membersites to collude and correlate persona activities at several sites. This is partially mitigated by the use of different personae for different membersites. However, in the extreme case each membersite will require separate persona (and GUPI) to avoid possibility of collusion. As the GUPIs are assigned by rootsite and the assignment is governed by the rootsite policies, this approach may be inconvenient or maybe even unfeasible.

The GUPI is assigned to the persona on a specific homesite by rootsite. This assignment is claimed by `authDelegation` SxipML element. The `authDelegation` element is signed by Rootsite and includes an expiration time. The correct setting of expiration time it is the only way of limiting the validity GUPI delegation. If the expiration time interval is long (more that few hours), it will considerably limit the ability to migrate persona from one homesite to the other. If the expiration time interval is short (minutes or hours), it may reveal some GUPI usage patterns to the Rootsite.

SXIP is considered a deprecated mechanism. SXIP development efforts were merged with OpenID development.

1.3.4.7 Lightweight Identity (LID)

Lightweight Identity (LID) [40] has started as a web-based single sign-on system, that also allowed sharing of additional data. LID was implemented using an approach similar to the browser-based mechanism described in the section 1.3.2. The GNU Privacy Guard (GPG) [41] signatures on message parameters were used as a security tokens.

The credential supplied in single sign-on approval message was a GPG signature of response parameters. The credential was verified by the target site by getting corresponding public key using LID URL and validating the signature.

The LID documentation mentions pseudonyms, but these are in fact separate personae that may have been linked by unspecified means. The model of LID persona linking is depicted on Figure 10. LID URL as an identifier may leak information, especially in self-hosting scenario as proposed by LID documentation. For detailed explanations see section 1.3.5.3.

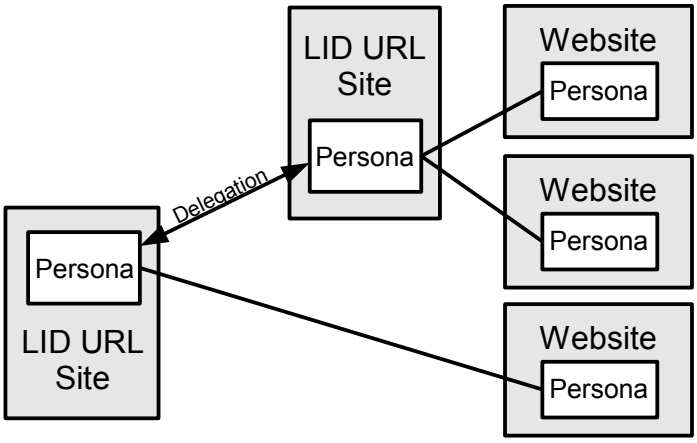

Figure 10 LID persona linking

GPG public key validation was left on simple "callback" method. No other method is mandated by the LID documentation (although it is allowed). The described simple method can be dangerous when using HTTP protocol, for example due to the DNS attacks [42] (note that DNS attacks may interfere with other LID functions, if LID URL specifies `http` scheme). While using HTTPS method to get public key, using SSL/TLS brings a dependency on X.509 PKI. The result is that LID uses two different PKI systems (X.509 and GPG), that are in principle and features very similar, but not compatible. Both of these systems must work properly for a safe LID operation: a problem in either one may result in the compromise of LID security as a whole. Validation of GPG public key using the GPG web-of-trust feature might be more desirable in this case, but such a validation may not be trivial and is not specified in the LID documentation.

LID pseudonyms are created as different LID URLs. Getting a pseudonym that is indistinguishable from primary LID URL may not be easy, as it will frequently lead to getting a new DNS domain or hosting space in existing domain and installing LID software. This process is difficult to automate and for that case it may be difficult to implement good pseudonymity or anonymity functions using LID specifications.

All user attributes has to be stored under the LID URL control. That may be undesirable as this means that there exists a centralized point that has all the information about user (or at least a persona). The LID documentation mentions, that the attribute processing may be delegated from pseudonym to the primary URL, but such a delegation will result in the pseudonym LID URL management software to see the

attribute value anyway. There are no mechanisms to refer requesting website to other URL for the attribute value retrieval.

Light-Weight Identity (LID) is no longer developed as Internet identity system. The effort on network protocols and communication principles were joined with OpenID project. LID is now an identity management system that uses OpenID protocols.

1.3.5 Common Issues

The Internet identity systems that follow the described method usually use short-lived security tokens based on public-key cryptography. The common drawback of these systems is the ability of source site to track the user's log-ons on service provider's site. While the source site cannot track the user's activity at these sites, the log-ons itself may provide sufficient information to potentially violate user's privacy.

1.3.5.1 Impersonation

The general property of Internet identity systems is the ability of source site to trivially impersonate the user [18]. The source site that exploits this ability may use the target's site interface to read the target site persona attributes.

1.3.5.2 Persona Identifiers

The Internet identity systems need a way to link several personas. The linking is implemented by associating persona identifiers on different systems. There are two approaches to the management of persona identifiers:

- Global persona identifiers. The persona identifiers are allocated by central authority that guarantees global uniqueness of the identifier. The examples of these identifiers are OpenID URL or SXIP GUPI. The global uniqueness of the identifier allows direct linking of personae on the global (Internet) scale.
- Local persona identifiers. The persona identifiers are allocated by the system, where the persona originated. These identifiers may be unique only in the scope of the source system. For the purposes of persona or account linking, the target site must accept the identifier in this form or (more frequently) apply appropriate identifier mapping.

While the direct linking and global persona identifiers may be the easiest scenario, global identifiers shared by many sites may be used to correlate user activities on several systems and thus reveal personal information without user consent. To overcome this problem, lower level virtual personae (with different identifiers) may be used as pseudonyms. If this approach is deployed in the Internet scale, the persona management may become difficult and may need automation. The automatic pseudonym persona management is technically close to the indirect linking using pseudonym identifiers. Indirect linking reveals less information about the user, therefore it may be considered a better approach for the Internet environment.

Global persona identifier is not required in indirect linking scenarios. The link is expressed as a pair of persona identifiers, each unique in the context of the system that created it. The identifiers combined with the peer system identifier are sufficient to uniquely define a link on both source and target system.

1.3.5.3 Self Hosting of Source Sites

One way of storing identity information is to host a source site on a system, that is under user's sole control. As this concept may seem attractive from the privacy point of view, it in fact may be undesirable in the practice:

- URLs of self-hosted source site may leak information. Parts of personal information may become part of the URL, for example domain name or path prefix may contain user's name. Additional information may be leaked by DNS records (e.g. SOA record), IP addresses or DNS and IP address databases (e.g. Internet Registries).
- The maintaining of site security on the operating system and application level is a difficult and never ending process. It is not likely that common user will have necessary knowledge and skills to implement and maintain all required security controls. In the case of outsourcing the system administration or security maintenance of the hosting system, the user is no longer in sole control of the data.
- It may be required that the trust relation between source site and target site may be bidirectional. For example in electronic banking scenarios the bank may be made responsible to enforce sufficient authentication level. For this reason, the bank (target site) must trust the authentication provider (source site) to authenticate user according to the agreed regulations. The trust to the source site in the self-hosting scenario may be questionable, and the high number of self-hosting sites with whom to establish trust may make the process unfeasible.

The self-hosting scenario is technically proxy-based true single sign-on system [18], but may be regarded a local true single sign-on system from the organizational control point of view. The self-hosting of source sites brings only one advantage: control over the stored data. But the control over data is lost when transmitted to other sites and even control of the stored data itself may be questionable. The self-hosting scenario will in most cases likely lower the privacy of the user and/or overall security level of the system.

1.3.5.4 Trust

Considered Internet identity systems take two principal approaches to the problem of trust relationship between source and destination sites:

- **Direct trust** between source site and target site is assumed. The sites must have direct relationship, assess each other ability to perform their tasks. This assumes a first-hand experience with other party and may also include business agreement or contract between the parties. The authenticated cryptographic material is expected to be exchanged prior to first interaction and continually maintained. The management of trust and cryptographic material is usually assumed to be done out-of-band and is not directly addressed by the specifications.

  The direct approach provides good abilities to manage risks, however it has very limited scalability. It is obvious that it will not be possible to establish direct relationship between all communicating parties on the internet. It is

questionable whether it is possible to establish direct trust between all target sites and all (or most) source sites.

- **Implied trust** between parties. The trust is usually implied from generic Internet infrastructure system such as DNS or the public key infrastructure of HTTPS. Most Internet infrastructure systems were not designed to support trust between parties, therefore depending on systems such as Internet Domain Name System (DNS) is not appropriate. Relying on such systems may even be dangerous, especially if the infrastructure was not designed with security in mind and several classes of attacks exists. Security deficiencies of DNS system is frequently compensated by public key infrastructure of HTTPS. However, HTTPS only binds DNS name to the keys used for securing data transmission. It does not express anything about the trustworthiness of the parties. The certificates usable for HTTPS purposes are equally well available to legitimate sites as to the attackers. Therefore it is a fallacy to assume that presence of valid HTTPS certificate implies that the party presenting it is trustworthy.

Neither direct trust nor implied trust is well-suited for Internet environment. Direct trust has scalability limits while implied trust is not conceptually feasible at this time. The issue of trust is a major obstacle for practical implementation of Internet identity systems. Even though most Internet identity systems claim that the issues of trust are out not in the scope of the project, the trust will be needed for deployment of practical systems on the Internet scale.

1.4 Identity, Anonymity, Privacy and Reputation

This section outlines the current state of the art related to the aspects of identity, anonymity, privacy and reputation. The efforts in this area are broadly spread, reaching even outside the field of computer science, into the fields of psychology, sociology and legislation. It is not feasible to survey the area in sufficient depth and breadth, therefore only the works that relate to the objectives of this work are surveyed.

Abelson and Lessig [43] note that there is no generic system for identification in cyberspace. Identity is considered to be a collection of characteristics which are either inherent or are assigned to the entity. They also note that the there is an inherent difference between cyberspace and realspace, seen as *unbundling* of identity information from transaction in the cyberspace. Therefore a form of verifiable identity information needs to be explicitly included in cyberspace transactions. The presence or lack of traceability is discussed in the paper, especially from the point of view of law enforcement. They argue that social and other non-technical aspects have to be considered when designing a system supporting identity in the cyberspace.

Pfitzmann and Kōhntopp [44] provide terminology for anonymity, unobservability, pseudonymity, and identity management. Anonymity is defined as inability to identify subject within a set of subjects. Anonymity is considered to be a boolean value, where subject can either be anonymous or non-anonymous. In the revised version of the document [45] it is admitted that for some application a thresholds needs to be defined. However, such a subjective definition may lead to inconsistent perception of anonymity between applications. The paper later operates with global anonymity as a measurable quantity, admitting that maximum anonymity may not be possible

to reach. Identity is defined as a set of attributes of person or subject that sufficiently identify such entity within any set of similar entities. Similarly to the anonymity, identity is regarded as a boolean value. It also defines the *linkability* and *unlinkability*, however the definition is only applicable to the items of interest, not to subjects themselves. A pseudonym is defined as a subject identifier that is different from the real name, however admitting that the pseudonym and real name may be indistinguishable. The paper does not distinguish between human actors and computer actors in the system.

Privacy was described by Warren and Brandeis [46] as "the right to be let alone" in the late 19th century. Solove [47] describes that traditionally view on privacy as an invasion paradigm and he argues that it is no longer efficient to address the privacy risks of 21st century. He is using the metaphor of Franz Kafka's novel The Trial to describe the current situation. Solove argues that the current situation is "architecture of vulnerability" because ordinary people are vulnerable to the mishandling of private-sector organizations. He proposes architecture based on Fair Information Practices [48] which was a one of the sources for OECD guidelines for the protection of privacy, that are similar to the EU personal data protection directive [49].

Resnick et al. [50] are discussing ad-hoc interactions on the Internet between parties that have no previous relationship. They propose a distributed reputation mechanisms a solution for problems inherent in such interactions. The positive effects of reputation systems on ad-hoc interactions are also described by Axelrod [51], proposing evaluation of past actions for the purpose to be used in future decisions. He describes it as "shadow of the future" that can motivate users to a better behavior in the present by threatening to punish bad behavior in the future. However, Windley et al. [52] note that there is a natural trade-off between reputation and privacy. They argue that as reputation is calculated from the record of past interaction, revealing such record means that part of the subject's privacy is lost. Sabater and Sierra [53] provide a review of computational trust and reputation systems, classifying them by a variety of criteria. They note that the reputation can be seen as global or subjective.

A game-theoretic approach to model Internet interactions based on examination of the record of past interactions is provided by Friedman and Resnick [54]. It is shown that the availability of cheap pseudonyms is harmful to the level of overall cooperation in the network. Friedman and Resnick propose the creation of once-in-lifetime certificates to mitigate the effects of cheap pseudonyms.

1.5 World Wide Web

World Wide Web (WWW) is a distributed global hypermedia system that originated as simple hypertext system in early 1990s. At the time of the World Wide Web inception there was no architectural or protocol documentation available, except for a short proposal [55]. The principles and protocols of current World Wide Web architecture have evolved during late 1990s, resulting in definition of URI [32] and HTTP 1.1 [56]. The architecture was guided by the Representational State Transfer (REST) architectural style described by Fielding [57] in 2000. However, there was no architectural document for World Wide Web until 2004, when the *Architecture of the World Wide Web, Volume One* [58] was published by World Wide Web Consortium (W3C). The WWW

architecture document retrospectively documented the thinking behind the development of World Wide Web. However according to Fielding [57] the architecture and protocols of World Wide Web we only roughly aligned with the REST architectural style and inconsistencies still remain.

World Wide Web Consortium (W3C) Technical Architecture Group (TAG) assumed coordination of WWW architecture development. W3C TAG influences the WWW architecture by publishing TAG findings [59], explaining and deciding questionable aspects of WWW architecture. However the findings are not always consistent with WWW architecture as described in details in section 5.2.

The only practical security and trust mechanism for the World Wide Web is HTTPS [31]. It is a protocol providing channel security (confidentiality and integrity protection) and authentication of the connection endpoints. The only way how to evaluate the confidence in the key material is to use X.509 public key infrastructure. However the "trust" structure of WWW is based on several certificate authorities that are either pre-configured or user-configured into web browser software. The browser software will accept any data coming from sites certified by any of these "trusted" certificate authorities as authentic. These certificate authorities only check that the DNS name that appears in the certificate belongs to the organization that requests the certificate, therefore they cannot be used for any evaluation of trust beyond this single naming aspect.

Current WWW browsers do not indicate the trustworthiness of displayed information. The information that browser typically display is limited to URL of the page and indication is HTTPS usage. It is difficult for a user to check whether the URL is correct and users frequently ignore the URL bar. This leads to a success of *phishing* attacks [60] because users cannot distinguish authoritative authentication page from a fake page with the same design. The use of HTTPS adds a possibility to get information about the page source from server certificate. However, users usually does not know how to display such information and what it means [60]. But even if such information would be always displayed, it would only provide information about the owner of DNS name, which may be different than the entity providing the data, e.g. in case of hosting the service or providing it in a *Software as a Service* (SaaS) manner. The usability study accomplished by Dhamija, Tygar and Hearst [60] demonstrated that good phishing site deceived 90% of participants. Users have no information to base an informed decision about trustworthiness of the site, such information is provided in wrong way or it is very easy to fake. This leads to a conclusion that the information provided by the browser is insufficient for a user to make an informed decision about the quality of the displayed information.

Current architecture of World Wide Web assumes that information always comes from its authoritative source or a trusted proxy. The HTTPS mechanism is designed to be effective for protection of information under such assumption. However, the usability of HTTPS is limited when such assumption does not hold, for example in case of on-demand data replication and migration.

There are two authentication mechanisms defined for the use with HTTP protocol [61]: Basic and Digest authentication. Both authentication mechanisms are fixed to username/password credentials and are not designed as extensible. The Basic authentication is susceptible to eavesdropping and replay attacks. The Digest

authentication improves on that, but it requires a state (nonce value) to be kept at the server between requests, thus violating the statelessness principle of REST architectural style. The Digest authentication is fixed to MD5 mechanism, which must be considered a weakness according the design principles set for this work. Both methods are susceptible to man-in-the-middle attacks, as there is no authentication of the HTTP server. No other widely-used mechanism for WWW authentication exists, except for application-level authentication (described in section 1.1) and HTTPS mutual authentication (described in section 1.2).

1.6 Discussion

Internet is a world-wide communication medium. It connects almost 1.5 billion of users [62]. It is unrealistic to assume that any significant part of such a huge population will maintain long-term relationships with all the people they interact. It must be expected that significant part of the interactions will be carried out between people and organizations that does not know each other. Therefore a method is needed that helps Internet users do evaluate the trustworthiness of interaction partners and generally any information available on the Internet.

Traditional methods of password-based authentication do not provide any benefit. These method assume establishing a relationship (user account) that is inefficient to maintain for short-term interactions. It may even become infeasible in a dynamic environment where the relationships change rapidly.

The methods based on public keys may be more appropriate, however they have failed for scenarios where users have to manage their own key pairs. If the public-key methods are feasible at all, the proper way of deployment is not known. However it is obvious that traditional hierarchical public key infrastructure based on X.509 specification is not feasible for Internet environment.

Variety of Internet identity systems appeared recently. These systems attempt to address the problems of multiple user accounts using practical solutions for decoupling authentication from the services. Although these systems may alleviate the pain of maintaining multitude of account, they do not address the issue of trustworthiness or evaluation of characteristics of communication parties.

The core of the problem lies in the architecture of World Wide Web. World Wide Web mechanisms does not provide any way that can assist users with evaluating trustworthiness of communication party or reliability of information discovered on the Internet. It does only provide a mechanism how to bind part of the resource identifier (DNS name) of the source that provides resource representation (Web server). However, this mechanism is inefficient to stop phishing attacks.

It is expected that the reputation systems may improve that situation by imposing “shadow of the future” on the users. By threatening the users to retaliate bad behavior in the future may motivate them to behave in a reasonable way. The reputation may provide a valuable input and a vital cue to users for evaluation of trustworthiness of the communication parties and reliability of information available on the Internet. However, there is no mechanism on the Internet in general and in World Wide Web in particular to support that could support inclusion of reputation and resource rating in day-to-day Internet interactions.

2 Dissertation Objectives

This section defines the problem that is being solved, the objectives for this work and statement about non-objectives.

2.1 Problem Definition

Following item summarize the problems that were described in details in chapter 1. The problems are described in a form suitable for the purposes of this document.

- The only practical method for the “trustworthiness” indicators is a binding of DNS name to the web server that controls that portion of URL space using mechanisms of HTTPS protocol. However, such mechanisms are insufficient to provide any information about trustworthiness of the certificate holder.
- It is assumed that user can evaluate the reliability of information provided on the Internet. However, the information comes from a variety of sources, many of them being fraudulent. It is questionable whether a user can evaluate reliability of information provided by a persona that he haven't ever interacted with and there is a only a small chance that he will interact with it again.
- The indicators that current World Wed Web software provides to the users are not able to stop phishing attacks. This indicates that the current methods of trustworthiness evaluation are insufficient for users to make correct informed decision about reliability of information on the Internet. We acknowledge that this is partially failure of user interface design, especially the form and placement of visual cues. However we posit that this failure is also caused by insufficient quality of information provided by the user interface.
- Even if we assume that the user can evaluate the trustworthiness of information source, there is no way in current World Wide Web to identify the information source. The HTTPS way of source identification has limited usability, as it provides only a DNS name of web server. There is no way how to gain more information (such as reputation score) from the DNS name. And the whole HTTPS approach fails in scenarios that feature dynamic replication and migration of information across multiple sites.
- The architecture of World Wide Web is outdated for needs of current dynamic Internet. While it was appropriate few years ago when dynamic content prevailed, it is being pushed beyond its limits in current dynamic and ever-changing world. We suspect that the inadequacies and inconsistencies of World Wide Web architecture render solutions to the above problems difficult.

2.2 Objectives

The work is focused on the improvement of World Wide Web architecture, especially on conceptual layers of the architecture. The objectives of this work are defined as follows:

- Define the goals and expectations for the World Wide Web architecture and design for current needs and for a foreseeable future.
- Evaluate the state, consistency and appropriateness of World Wide Web architecture according to specified goals.
- Identify and discuss fundamental problems of WWW architecture, especially focused on handling of user identities and evaluating trustworthiness of commutation parties and reliability of information in WWW environment.
- Propose architectural improvements in World Wide Web architecture on a conceptual level, leading to improved support for evaluation of information reliability.
- Validate the proposed architecture and demonstrate that the proposal can handle situations that are problematic in current World Wide Web.

We acknowledge that the current World Wide Web exists and that it will be very difficult to replace it. Therefore our goal is not a radical and fundamental change of World Wide Web. We rather seek to propose improvements that can be implemented gradually during the course of natural World Wide Web evolution. The work is motivated by a desire to define goals for future development of World Wide Web – a reference architecture that could be used to judge design decisions that will be necessary during development of World Wide Web.

One of the primary implicit goals of this dissertation was to keep the efforts of designing the improvements of World Wide Web architecture feasible. We sought to provide complete results at the conceptual level that can be reviewed and evaluated rather than incomplete and therefore non-conceptional solution of a single World Wide Web aspect. The review and improvement of World Wide Web architecture is definitely not an easy task, therefore we opted to split it to several steps. This document describes the results of the first step, improving the mistakes of World Wide Web architecture on the conceptual layer. Therefore the objectives of this dissertation are set specifically for that purpose.

2.3 Non-Objectives

Following items are not considered to be objectives of the work, they not addressed by this dissertation.

- User interface details are considered to be out of scope of architectural work. The details of user interface should follow from the usability design and they are naturally constrained by system architecture.
- Reputation computation algorithms and mechanisms are not analyzed in detail. However existence of such mechanism (or rather several such mechanisms) is taken into account.
- Protocol specifications and specific improvements are only outlined, without any detailed specification. The objectives are focused on conceptual layer, leaving specific component design and protocol specifications for follow-up work.

3 The Model

This chapter provides description of a model that will guide the rest of this work. The model is based on the interaction of two worlds: the world of human beings and the world of computers. We discuss how people deal with computers and the implications on the reliability of information provided by computers. We also deal with anonymity and identity of people in regard to data processing.

The model is a by-product of the architectural work. No practical architecture can be conceived without first understanding the concepts that underlay the architecture and the needs of the people that seek to gather the benefits of the architecture. Therefore we provide a model that attempts to describe and explain the undercurrents of realspace-cyberspace interactions. The architecture that is described later in this document builds on the ideas introduced in this chapter.

3.1 Definitions

Realspace: The world that an average human can directly observe with his senses.

Cyberspace: The world of interactions of computer components. It cannot be directly observable by humans. It can only be accessed indirectly using a terminal device.

Terminal Device: Device that allows interaction between realspace and cyberspace, usually a personal computer, mobile phone, but also a network camera, autonomous network sensor, measurement tool, network printer, etc.

Mobility: Ability of the user (realspace person) to move freely between locations and devices. The user may access the system from two different places using two different terminal devices at two different times, still getting the same or equivalent service. The service may only be limited by the capabilities of the device user is using and by the access control restrictions.

Network node (**node**): A self-contained cyberspace entity that interacts with other entities on a computer network. It is usually a server, active network device, personal computer, mobile phone, etc.

Site: A group of network nodes that are under a control of a single realspace entity. The physical location of the network nodes does not matter for a concept of a site, however the network-wise distance may be important. Single realspace entity may control several sites. We will use the term site to generally refer to a group of nodes that has some designated function, for example web server farm or enterprise data center.

Identity: The term *identity* is used as an adjective to specify systems and concepts that deal with personas, personal information or other concepts related to person. We use *identity* rather than *personal*, as the word *personal* implies private or sensitive character or something belonging to a person, which is not appropriate meaning for the purpose of this work. For example the term *personal agent* would imply that the agent belongs to a specific person, which is not the case as such agent just maintains the data about many persons. Term *identity agent* has neutral meaning, which is more

appropriate in this context. We do not assume any special meaning or definition for the noun *identity*, we use it freely to mean any kind of information about a person.

Identity agent: An entity that maintains information about personas. It may be a specialized entity (identity providers), usual business organization that has first-hand relationship with the user (e.g. telecommunication companies or employers). In the case of self-maintained persona we consider the software system that maintains the persona to be identity agent.

Trust: Is a realspace relationship. We will define trust according to Dasgupta [63]: "the expectation of one person about the actions of others that affects the first person's choice when an action must be taken before the actions of others are known". In addition to the trust as person-to-person relationship we do not exclude the person-to-organization or organization-to-organization meaning of trust, provided that the organizations may exercise free will. We will assume that trust is a deliberate choice of sentient beings and that it is not automatically implied by any action, sensory input, indication in the user interface, etc. Such approach is also supported by Friedman, Kahn and Howe [64] stating that "People trust people, not technology".

Identifier: A data unit that can be used to distinguish one entity among other similar entities. Identifier does not imply that the entity can be located, does not specify where it is located and does not even mean that the identified entity exists.

Name: A human-readable identifier. However, the term *name* is used in a relaxed meaning in this document. Especially in context that refer to the traditional meaning of *name* or the meaning implied by other specifications (such as in *DNS name*).

Address: A data unit that specifies location of a specific entity. Can be applied to both: realspace (e.g. postal address) and cyberspace (e.g. IP address).

3.2 Realspace and Cyberspace

The basic principle of the model used for the purposes of this work is that there are two distinct spaces:

Realspace is the world that we live in. The world that we can see, feel, hear, smell or taste. The world that can be understood by using just senses and mind of an ordinary human being.

Cyberspace is a world of computer-to-computer interactions. It cannot be directly observed by human beings, as we cannot directly measure electrical currents and voltage with sufficient precisions, we cannot directly decode the information from optical fibers and we cannot directly detect the magnetic fields of data stored on disk drives.

The interaction between realspace and cyberspace is made possible by *terminal devices*. These devices are entities that are part of both spaces and they convert information from a form perceivable in one space to the form suitable for the other space. For example computer monitor can convert the electrical signals that humans cannot observe to the light signals suitable for human observation. Computer keyboard converts movements of human hands to the electrical signals that can be understood by computers.

The boundary between the spaces is somehow fuzzy. For example it is difficult to objectively decide if the matrix of the LCD display is in realspace, cyberspace or both. The actual distinction always depends on the point of observer's view and on the level of detail. Therefore we acknowledge fuzziness in the model and we deliberately cover it under the concept of terminal device. Terminal device is an entity that can interact in both spaces. However the concept of terminal device may still vary, depending on the level of detail. For example display and keyboard of a personal computer may be seen as terminal devices, while the rest of the computer may be seen as an entity of cyberspace for one application of the model, while another application may consider that the whole personal computer to be a terminal device.

As this work is based on a considerably high level of abstraction, we will operate with the concept of terminal devices on a coarse-grained level. We will consider user personal computers, printers, mobile phones, autonomous cameras and sensors and similar devices to be terminal devices for the purposes of this document.

Realspace entities cannot be sure whether the terminal device operates as expected, as the realspace entities cannot observe the cyberspace interactions. The realspace entities may use one terminal device (e.g. network protocol analyzer device) to check the behavior of another terminal device. But even this approach will not provide definitive results, as both devices may be faulty or they may be modified to cover up the malicious behavior.

Cyberspace entities cannot make sure that the terminal device operates as expected either. Cyberspace entities cannot observe the behavior of realspace entities in any other way than using a terminal device. Cyberspace entities may try to check the data provided by one terminal device (e.g. fingerprint reader) by correlating it to the data provided by another terminal device (e.g. digital camera). But the cyberspace entity cannot make sure that the information provided by these two terminal devices is related, for example that it refers to the same physical (realspace) location. The camera may be placed in a different location than the fingerprint reader, in which case correlation of the data provided by these two devices is most probably pointless. Even if the cyberspace entity has information that these two devices are located at the same place, that information was inevitably introduced to the cyberspace by another terminal device and the cyberspace entity has no way how to check reliability of such information.

Based on the discussion above we can formulate following statement:

> Crossing the boundary of realspace and cyberspace is always subjective.

The entities in cyberspace have no way how to make sure that the collected data really belongs to the realspace object. The data collected by a computer system about specific person may be true or false, the computer system cannot distinguish that. Even biometric data collected by computer system may be false, they may belong to another person, may be entirely fake or may be spoofed by replacing the sensor with a device that will replay spoofed data.

The entity receiving data from other space using a terminal device must make its own assumptions about the relevance of the data. It has to (implicitly or explicitly) evaluate a level of belief that the data describe what they are supposed to describe. This level of belief is usually *predetermined* for a specific terminal device, pattern of usage, etc. For example a personal computers (which will be considered a cyberspace entity for this

example) are usually programmed in such a way that they assume that the pressure on the keys of a keyboard (terminal device) are caused by fingers of a human being and that it is an indication to enter the letter denoted on each individual key to the computer system. This belief is predetermined and programmed into the computer system. However, it may in fact be a paw of a domestic cat that caused the pressure instead of human finger and the meaning of that action may be far from a desire to enter a letter to the computer system.

Therefore we consider realspace-cyberspace interactions to be *subjective* (as opposed to being objective). The interaction depends on the interpretation of the information by both realspace and cyberspace entities, on their preconceptions, predetermined behavior and believes, on the presentation and detection capabilities of terminal devices, on the environment and overall situation of the interaction.

As any information that resides in the cyberspace originated in the realspace and had to pass realspace-cyberspace boundary, we may formulate following statement:

> Any information coming from the cyberspace is subjective.

Therefore the trustworthiness of the information cannot be reliably evaluated unless its source is known. The credibility of the information source must always be considered to determine the likeness that the information is true. Therefore we can formulate following statement:

> The source of the information in the cyberspace is equally important as the information content.

The amount of information that needs to be known about the source depends on a situation. For example the source itself may be anonymous, but we may be inclined to believe the information from that source if a known trusted entity vouched from the trustworthiness of the source. On the other hand the knowledge of the source does not automatically imply confidence in the information that it provides. For example information from a well-known liar is not likely to be considered true.

Based on the reasoning above, we do not require the user of information should have complete knowledge about the realspace identity of the source of information. We rather propose that appropriate information about the source should always be conveyed with the information content. We also propose that the source is always taken into consideration when the information is used, while the actual mechanisms of consideration may vary:

> **Recommendation:** The source of the data in cyberspace should always be taken into consideration. No data should be regarded as true or authentic without the evaluation of the realspace source of the data and the method of transferring them to and from cyberspace.

The decision about the quality of the information may be based on information provided by a third party. Such entity has to be believed to truthfully evaluate the source and provide acceptable results. This evaluation can be transitive: mediating mediated information or correlating information from several sources. However, even if the evaluation of the source is mediated, the decision whether to believe provided information is has always to be done by the (realspace) person that is using that

information. Cyberspace entities can only process and provide data to support such decision.

3.3 Persona

Persons (realspace entities) acting as users of computer systems are represented in the cyberspace by data structures. As explained in the previous sections, the computer systems that are interacting with realspace persons have limited capabilities of determining the person characteristics directly. The data structures that represent users are assembled from subjective information which is commonly entered to the system by the users themselves. This data is seldom verified in any way, although in some cases personally-identifying information about a subject is collected and stored by a third party such as a certification authority. Third party may also verify the data and make the information about this verification available in the cyberspace, however such verification information is also subjective.

The data structure maintained in the computer system is in most cases incomplete representation of realspace person, with variable reliability. The data structure may not even be unique: the realspace person may create a similar data structure in the same or different computer system. The person may also create several distinct data structures to represent different roles or personalities for different purposes (Figure 11).

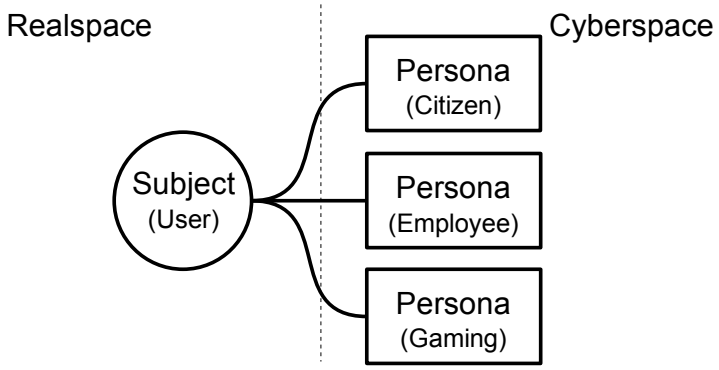

Figure 11 Subject and the personas

We will use the term *Subject* to represent the realspace person and the term *Persona* to represent the cyberspace data structure maintained in the computer system that is related to the person:

> **Subject** is a conscious realspace entity that is guided by a free will.
>
> **Persona** is a cyberspace data structure that represents some aspects of (realspace) subject or a cyberspace entity that is governed by realspace subject. The aspects are represented as a collection of properties with machine-readable values.

Subjects represent users of the computer systems, living persons or persons that once lived but are already dead and it may also represent organized groups of people such as companies, governments or other organizations. This model does not reflect realspace organization of the world and does not consider distinction between physical persons

and legal persons or any other type of realspace entities. All conscious realspace entities that can exercise free will are considered subjects for the purpose of this model.

The correspondence of the persona properties with the characteristics of the realspace subject may vary. One extreme situation may be a government-created persona to represent subject as a citizen, while the correctness of the persona attributes is verified by the official authorities. On the other end of the spectrum may be a persona created for the purpose of a computer game that will represent completely fictional character. The completeness of persona data may vary as well. Some systems will maintain only minimum information needed for correct operation, to conform with privacy legislation. Other systems may collect rather rich data records.

Persona can also represent computer program or similar cyberspace entity. However, we assume that the computer system cannot act by itself (it is not conscious) and that it follows that each cyberspace entity represents a subject, is controlled (programmed) by a subject or acts on behalf of a subject. Therefore persona always directly or indirectly describes a single subject.

> The cyberspace data structures of persona can be only a subjective representation of realspace subject.

As explained in the previous section the crossing of realspace and cyberspace boundaries is always subjective, therefore the data in a persona must always be regarded as subjective. Whether a persona can provide complete description of a subject is still an open question, as it is not known whether all the characteristics of a person can be observer and transferred to the cyberspace. Therefore we will assume that persona is usually a partial and incomplete representation of the subject, however we will not rule out the possibility of a complete representation.

3.4 Anonymity and Identity

The persona usually originates as an data structure describing one realspace person (subject). But after the original creation, the link between the persona (cyberspace entity) and subject (realspace entity) may not be apparent. In privacy-enhancing applications the inability to distinguish a subject given a persona may even be crucial requirement. According to this, a mechanism is needed that will allow evaluation of relations between personas and subjects in a running system, long after the creation of personas. The following paragraphs provide means that can be used for this purpose.

We define a concept of a world set that is a set of all candidate subjects that could be used as a source for personas in the system we consider.

World set is a set of all possible subjects, denoted as follows.

$$W = \{S_1, S_2, S_3, \ldots, S_n\}$$

In the broadest possible model the world set will contain all subjects that can, even with negligible probability, describe any possible persona. This definition of a world set would be greatly impractical, as it would include all the living persons and also all the persons that ever lived and all the groups of persons. In the practical model applications the world set would be chosen with respect to some baseline knowledge base. For

example if we know that the personas in our system represent only living realspace persons, we can choose the world set that contains all living realspace subjects. If our implementation allows only Internet-based access and thus our personas describe only the people that can access the Internet, we can choose the world set to include realspace persons that are Internet users only.

> **Identity probability** is a (Bayesian) probability that given persona describes given subject. Denoted
>
> $$i_{P,S}$$
>
> whereas P is a persona and S is a subject.

Identity probability of zero (0) means that persona P cannot describe subject S. Identity probability of one (1) means that the observer is sure that the persona describes the subject. The values between zero and one may be interpreted as different degrees of belief that the persona describes the subject.

The identity probability value, as well as all other probabilistic metrics defined in the document, are subjective to a specific observer and depend on his knowledge. Different observers may assign different values to identity probabilities. The following statements are only expected to hold for a specific observer.

While persona describes exactly one subject (by definition), the sum of identity probabilities for specific persona and all subjects in the world set must be 1. The following holds:

$$\sum_{k=1}^{n} i_{P,S_k} = 1$$

We can define a random variable I_P with the world set W as the collection of source states and with individual probabilities being equal to identity probabilities of a single persona:

$$P(I_P = S) = i_{P,S}\ , \quad \forall S \in W$$

The random variable I_P represents possible subjects that the persona P may describe.

> Given a specific persona, **anonymity set** (denoted AS_P) is a set of all possible subjects from the world set for which holds that the identity probability of the persona and the subject is greater than zero. Can be denoted as:
>
> $$AS_P = \{S:\ S \in W \ \wedge\ i_{P,S} > 0\}$$

In other words, anonymity set is a set of all possible subjects that the persona may describe. In contrast to the world set, the anonymity set is based on identity probability and therefore is subjective to the observer. This definition of anonymity set is a probabilistic extension of the definition used by Pfitzmann and Kōhntopp [44].

> **Anonymity ratio** of a persona with respect to world set and observer describes the relative uncertainty in persona correspondence to a subject. Defined as

$$ar(P) = \frac{H(I_P)}{H_{max}},$$

whereas $H(I_P)$ is an entropy [65] of random variable I_P, defined as

$$H(I_P) = -\sum_{k=1}^{n} i_{P,S_k} \log_2(i_{P,S_k}),$$

and H_{max} is a maximum entropy for a random variable with n states, that is

$$H_{max} = \log_2(n).$$

Anonymity ratio of 1 means total anonymity: the persona may describe any subject from the world set with equal probability. The anonymity ratio will be 1 if the observer cannot in any way distinguish the subject that the persona describes (the probability distribution of random variable I_P is uniform).

Anonymity ratio of 0 means no anonymity: The persona describes a single specific subject. The anonymity ratio will be 0 if the observer is sure that the persona describes a specific subject.

Identity ratio of a persona with respect to world set is a probabilistic inverse of the anonymity ratio. Defined as

$$ir(P) = 1 - ar(P).$$

The identity ratio describes the degree of observer's belief that persona P describes some specific subject. Identity ratio of 0 means no identity: the observer cannot infer any useful information about the subject that persona describes. Identity ration 1 means total identity: the observer is sure about the subject that the persona describes.

Exact values of anonymity ratio and identity ratio may be very difficult (if even possible at all) to compute in the practice. Only the estimated values of these metrics can usually be determined. In a common case, it will be beyond the power of any observer to gather up-to-date information about all subjects in the world set. As such data is required for identity probability computation, the anonymity and identity ratios should be seen as theoretical (and subjective) values that, in practice, can only be estimated.

3.5 Analogy and Heterology

While anonymity and identity cannot be easily determined in computer applications, we may use other metrics that can be computed in computer environment and may provide estimations of anonymity and identity. Following paragraphs provide description of analogy and heterology relations and analogy probability value that may be used to approximate anonymity and identity in practical scenarios.

The personas are **analogous** if and only if they describe the same subject.

Analogous personas describe the same subject. These personas may be two accounts in different systems that belong to the same user or two database records describing the same person.

Note that the persona properties for the analogous personas may have different values. For example one persona will provide the subject's office phone number as a point-of-contact and another may provide his mobile phone number. Also note that it may not be necessary to identify the subject to resolve the analogy of personas - it may be apparent that the personas describe the same subject without actually knowing which specific subject they describe.

The analogy defined in this deterministic manner may not be very useful in practice. It is usually difficult (if possible at all) to reliably decide if two personas are analogous, but it is usually feasible to provide estimation on how likely it is that the personas describe the same subject. For this reason we define the probabilistic version of analogy:

> **Analogy probability** is a (Bayesian) probability that two personas are analogous. Denoted
>
> l_{P_1, P_2},
>
> whereas P_1 and P_2 are personas.

Analogy probability can be seen as a degree of observer's belief that two personas describe the same subject. For example if two personas share the same values of first name and last name attributes, the observer may believe that these personas describe the same subject with 60% probability. Therefore the analogy probability of these personas would be 0.6.

> The personas are **heterologous** if and only if they describe different subjects.

Heterology has features similar to analogy. For example, two personas may have the same attribute values, but yet may be heterologous (e.g. if the personas has only sex, age and city attributes) and it may not be necessary to identify the subjects to resolve heterology of personas. Similarly to the concept of analogy we define a probabilistic version of heterology:

> **Heterology probability** is a (Bayesian) probability that two personas are heterologous. Denoted
>
> h_{P_1, P_2},
>
> whereas P_1 and P_2 are personas.

Heterology probability can be seen as a degree of observer's belief that two personas describe different subjects. For example if two personas have different values for first name and last name attributes, the observer may believe that the personas describe different subject with 90% probability. Therefore the heterology probability of these personas would be 0.9.

Two personas can only describe the same subject or two different subjects. Therefore it follows that

$$l_{P_1,P_2} + h_{P_1,P_2} = 1 \ .$$

Figure 12 shows an example of anonymity, identity, analogy and heterology applied to simple structure of two subjects and several personas. The thin arrows denote data origin. For example, persona P_3 originated as a (partial) copy of P_2 that in turn originated as a (partial) description of subject S_1 characteristics. The thick arrows denote anonymity/identity, analogy and heterology relations. For example personas P_1 and P_2 are analogous, because they describe the same subject S_1. Also the personas P_1 and P_3 are analogous. Note that it is neither required to determine relation between P_1 and S_1 nor relation between P_2 and S_1 to evaluate the analogy.

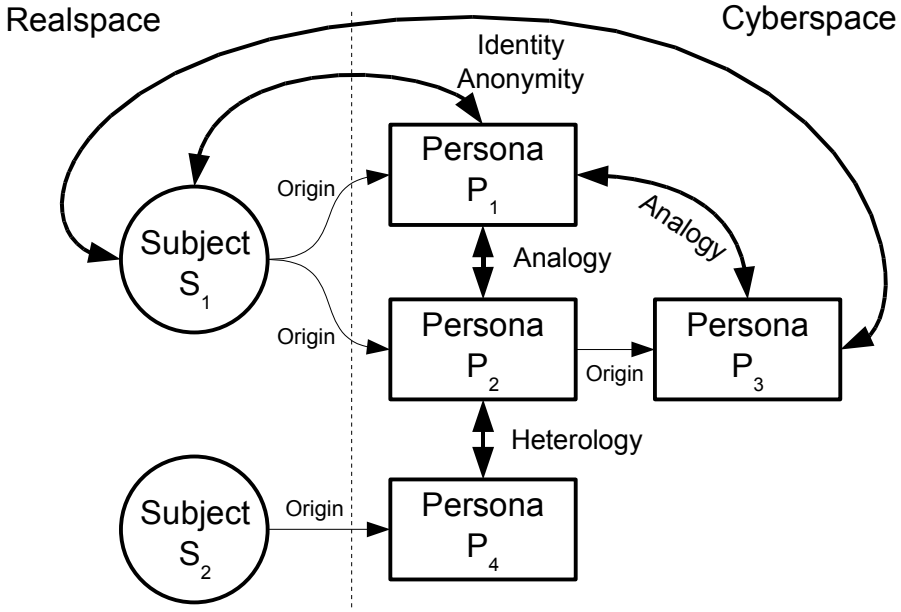

Figure 12: Subject-Persona Relations

Analogy probability and a world set based on persona databases may provide an estimation of anonymity/identity ratio values in practice. Consider that we have a description of a specific person, specified in a cyberspace form of "test persona". We want to estimate the anonymity ratio of this persona. We cannot do this by comparing the data in persona directly to every realspace person, as that may become infeasible even for medium-sized world sets. We need to provide anonymity ratio estimation by a different method, which can be more easily automated.

Let us suppose that we have access to a government database that contains records of all registered citizens of a country. These records are personas, as they are digital representations of realspace person characteristics. Instead of comparing our "test persona" to every physical person, we will compare it to the records in the database. We are in fact determining analogy probabilities of the "test persona" and the personas in the government database. We may use these analogy probabilities as an estimation of identity probability. And we may use the set of government database entries as an approximation of the world set.

We will use the terms identity, anonymity, analogy and heterology in the rest of this document to refer to both the deterministic and the probabilistic versions of these metrics. The choice of the specific approach (deterministic or probabilistic) may vary for each application of the model.

3.6 Addresses and Identifiers

Personas are regular cyberspace objects. Assuming that a suitable mechanism exists, any object can be identified and addressed. In case of globally-distributed information systems the identification and addressing of objects becomes a crucial aspect of the system.

The purpose of *identifier* is to refer to an object. The identifier is used to distinguish one specific instance of an object from other instances. The quality of the identification may vary. Some identifiers may be globally unique and unambiguous, other identifiers may be usable only under some circumstances and in limited scope. The ideal identifier is globally unique and unambiguously and permanently refers to the same object under all circumstances.

The purpose of *address* is to locate an object. The address will inevitably become invalid if the object moves. For example if a node in the network moves to a different part of the network, it address will change. If a person moves to another city his postal address will change. The address may not be unique, several objects may share the same address. For example several nodes may share the same multicast address on the network. Several people may live in the same place and therefore share an address. The addressing may also be hierarchical or may be "proxied". For example many hosts may be hidden behind a single address if network address and port translation mechanisms are used. In this case the node executing the translation is forwarding data based on its internal state tables. Many people may be hidden behind a single company address. In that case an internal mail department in the company is forwarding the mail based on employee location directory.

The purpose of identifier is to distinguish particular object, while the purpose of address is to locate such object. While these two goals may seem to be the same or very similar, they are in fact considerably different. The differences are summarized in Table 3.

The identifiers and addresses have conflicting goals that are very difficult to align. We consider the use of identifiers as addresses or vice versa as harmful and non-productive. The examples of some problems caused by mixing the concepts of address and identifiers are described in chapter 5.

The Domain Name System (DNS) [66] is an example of indirect addressing mechanism. Although DNS allows to use human-readable names instead of numeric addresses, the primary purpose is to identify locations. Therefore DNS as it is being currently used cannot be considered an identifier scheme. However, DNS might be extended to a general purpose identification mechanism, if the use of DNS would not be limited just to identification of Internet hosts.

Identifiers allow *trivial linkability* of personas. If two or more personas are known to describe subject identified with the same identifier, then these personas are usually

analogous (with high probability). We will refer to such personas as *trivially linkable*, as it is trivial to determine the analogy of the personas.

	Identifier	**Address**
Purpose	Distinguish an object from other objects.	Locate an object.
Lifetime	Maximize lifetime, keep identifier unchanged as long as possible, so the relations established by the identifier and maintained without a need for an explicit action.	Minimize lifetime to keep the address fresh. The address may get quickly invalid if the object moves rapidly (e.g. mobile object, object replicated on demand, etc.)
Content	Do not include location-related data in the identifier, as location may change and identifiers should stay valid as long as possible. Do not include time-related data, data that depends on any mutable structure or anything that is likely to change.	Include location in easy-to-use form. The most usable form may include routing information or organizational aspects to speed-up the lookup of an object.
Examples	Name of a person, ISBN, ASN.1 Object Identifier (OID), U.S. Social Security Number, UUID	Postal address, IP address, URL, DNS names
Translation and Modification	The identifier should be bound to the message or other data. The attempt to change an identifier should be detectable (e.g. by signing the message containing the identifier).	It should easily support address translation, forwarding, dynamic routing, etc. The manipulation of address should not affect the message or transferred data. The dynamics of transfer (routing) should not cause invalidation of the message.

Table 3: Comparison of identifier and address

4 Design Goals and Methods

The architecture is not created on a green field. Successful architecture must consider both existing and desired environments. Our architectural proposal is based on existing design of the Internet and World Wide Web, proposing extensions and improvements. Therefore it is natural that our proposal is inclined to reuse existing concepts and principles wherever possible. The proposed architecture is also guided by the desire to create a better environment for cooperation, as described in the rest of this chapter.

Architectural model similar to the model proposed by Fielding [57] is used as a guiding principle in this work. The behavior of the designed system is described in forms or constraints: what the system must do and what it must not. Architectural style is formed as a named set of architectural constraints. But in contrary to Fielding we do not strictly distinguish between architecture, architectural style, specification or interface. We consider all of them to be a named set of constraints on different levels of system architecture and design.

Simple diagrams are used to illustrate the architectural concepts. The diagrams use arrows to describe that one concept is constraining another concept. For better illustration we use solid-line arrows to denote that one concept is constraining most parts of other concepts, while we use dashed-line arrow if only some aspects are constrained. Different levels of abstraction are expressed by rounded corners of rectangles in the diagrams. More generic concepts are displayed as rectangles with corners that are rounded more than those of specific concepts. An example of such diagram is provided in Figure 13.

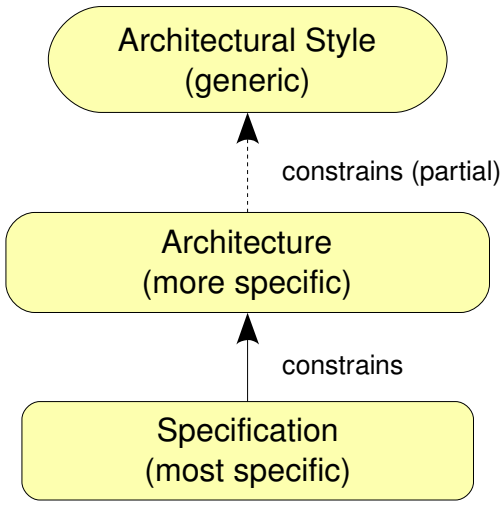

Figure 13: Example of architectural diagram

4.1 Desired Environment

For any solution to be successful it has to fit into the environment and support it. The solution should support the desired properties of the environment, while inhibiting unwanted effects. In this chapter we discuss the environment that we want to create and support.

Our goal is to create an environment that will support effective cooperation of people. Such environment should induce the positive network effect [67]. It should encourage the cooperation of any two entities in the network. The cooperation should not be limited to channel-oriented interactions, where few strong entities mediate most of the interactions on the network. Such a centralized approach would limit the value induced by the network effect. While the environment should encourage the pattern of any entity interacting with any other entity without a need for mediation, mediators may still be needed to initiate the interaction (rendezvous) and to manage the risk. However, the need for mediation should be kept to a necessary minimum.

It is expected that the environment will change as the society changes. The designed system must address such a dynamic nature of the environment. No information should be regarded as permanent. The dynamics of the information must be taken into consideration and be reflected in the architecture.

Three environment settings are considered in following sections: anarchy, authoritarianism and environment of responsibility. The list of environments is by no means comprehensive. These three environments are provided as examples. They considered a thought experiment to support the discussion about the properties of desired environment. The possibility of existence of described environments in their pure forms as described is questionable. We rather expect that the resulting environment will be a mix of different aspects of the described environments, possibly supplemented by properties that we haven't envisioned.

4.1.1 Anarchy

There are no explicit rules in the anarchical environment. Anybody can do anything, nothing is forbidden or denied. All the rules are set by individuals, and enforced by individuals. The anarchical environment is based on secrecy, anonymity and decentralization that are built as defenses of individuals against the intrusions from other individuals. Revealing information is a one-way operation, as the individual does not have any control over the information after it is revealed. It cannot be taken back. The information is regarded an asset of highest value.

Anarchical environment has following properties:

- The individuals do not disclose information that is not required to be disclosed.
- The mechanisms should be designed in such a way that communication parties should not be able to learn any additional information about the other party except the information that is explicitly revealed.
- The actions of the individual should not be traceable to a particular person, as such a tracing may reveal information that should not be revealed.

- Two or more actions of the same person should not be linkable. Correlating these actions may reveal more information than is revealed by the individual actions.
- The system must not be able to determine location of the user.
- The users should avoid keeping of information and policies on centralized location, as it lowers the exposure of information in case of security incident.
- User identifiers must be temporary, single-use only to limit the traceability of users.
- User's credentials must not leak any information.
- User's physical identity must not be known.
- There is no way to avoid misbehaving users as anybody has the freedom to do anything.
- Misbehaving owners of information systems cannot cause any significant damage, as they are not trusted by definition. Only the most essential data is revealed to them.
- There is no way how to compensate for the damage caused by another user of the system.
- No single entity (person, organization) must control the whole system or any of its essential parts.
- The privacy rules are not set by anybody. The privacy is based on secrecy (not revealing the information).
- Free speech prevails over privacy and control. There is no way how to stop gossip and misinformation.

Although the anarchic environment may look like the ultimate incarnation of freedom, it is not. There is no way for an individual to stop others invading his liberties. The goodness of one individual is frequently an evil of other and the anarchic environment lacks any mechanism for balancing. There is no way how to correct errors, for example how to take back the information that was revealed by mistake. Individuals does not have sufficient power to enforce their rules on others, therefore the rules that are set by an individual can constrain only these individuals that willingly follow them. But there is little motivation to follow any rules willingly.

The effect of anarchical environment will be a wilderness where few strong individuals will abuse the majority of others, making them even weaker and more susceptible to domination. The business is ineffective, as nobody can be trusted. Long-term relationships are necessary to build-up relationships of trust and no long-term relationships could be established if the identity of participants is actively hidden.

The anarchical environment can be seen in some parts of the “wild” Internet, but it is slowly disappearing due to the effect of legislation that finds its way to the cyberspace.

4.1.2 Authoritarianism

Authoritarian environment is completely controlled by a single entity. That entity sets and enforces all the rules. What is not explicitly allowed is denied. The environment is

focused on security, enforcement and centralization. Everything is subjected to interests of the controlling entity.

Authoritarian environment has following properties:

- The controlling entity will consider any information that can help better evaluate the risks. No information of importance must remain hidden from it.
- All actions should be accountable and traceable to a particular person, for the purpose of punishing the individuals that does not follow the rules.
- The controlling entity can limit the access to resources based on location, terminal device, time, protocol, etc.
- Communicating parties should know each other to evaluate reliance on the provided information.
- The system must be able to enforce use of secure access devices.
- The system must be able to determine the physical location of the user.
- Important information and policies are kept on centralized location for easier management and validation.
- User's identifiers must be permanent, as that supports accountability.
- User's credentials must be kept under the direct control of the user at all times, they should be (forcibly) replaced regularly to guarantee their quality.
- User's physical identity must be re-checked regularly.
- Rely on physical enforcement to deny bad actions of users.
- Rely on certification of trusted information system owners.
- No way to avoid misbehaving information system owner, if that is approved by the controlling entity.
- Seek damage compensation by legal persecution of the misbehaving users or system owners, usually by escalating the problems to controlling entity.
- Whole system must be under a control of a single entity (person, organization) or appropriate legal arrangements must be in place to ensure enforcement of the policies.
- Privacy rules are strictly mandated by a centralized entity (e.g. government) that has the power to enforce them. There is no privacy of individuals with respect to controlling entity.
- Control prevails over free speech. Privacy may or may not be provided.

The authoritarian rule is very difficult to scale, large authoritarian environments tend to become bureaucracies. Even if the initial regime cared about the well-being of users, that motivation goal tends to be lost in the maze of bureaucratic labyrinth. The authoritarian environments (especially bureaucracies) suffer from a high risk of corruption, as the gains of a corruption are irresistible.

Although a small authoritarian environment can work efficiently, the effect of large authoritarian environment will be a bureaucratic system that is difficult to change and balance. The free speech and innovation will be diminished, the environment will stagnate. The effect of corruption will bring down efficiency of the whole system and the users will suffer. The business will be effectively mediated by the controlling entity,

as the controlling entity will determine who is trusted and who is not. This leads to a centralized business model that is very difficult to efficiently scale.

The authoritarian environment can be usually seen in large enterprises and government organizations.

4.1.3 Environment of Responsibility

The environment of responsibility empowers users to exercise their free will, but still makes them accountable for the consequence of their actions. This environment is focused on economic efficiency, convenience, availability of resources and mobility. The environment of responsibility is based on the business environment of partially regulated free market. It is left to self-balance itself, intervening only on the conceptual level balancing the responsibilities of entities (e.g. for the case of privacy rules) and for resolution of disputes.

The environment of responsibility has following properties:

- The owners of information systems consider any information that can help provide better service to the user. Better service to the user means better economic results.
- The system should be able to detect a returning user to provide personalization. This does not necessarily mean that the identity of user is known to the system owner.
- The system should help tracing the action to a physical person if necessary for the purposes of law enforcement.
- The system must not enforce the use of specific client software or device. The market should not be monopolized and the innovation should be possible.
- The system must allow the access from any device, any place at any time to promote the availability of services to users and allow mobility (better efficiency).
- The system should be able to determine location of the user to provide location-based personalization.
- The location of the user should not be a limiting factor. User should be able to get the same (personalized) service regardless of the location.
- Replicate information and policies at multiple locations to promote availability.
- User's credentials should be permanent. They should be replicated on multiple locations, always accessible, easy to replace in case of loss.
- User's physical identity should not matter. Users should be able to use aliases and present alternative personas to overcome social limitations.
- Rely on "vox populi" (reputation, recommendations, popularity) to judge other user's credibility and to avoid misbehaving users. Reputation feedback should be allowed to demonstrate positive or negative opinion on other users.
- Rely on business incentives and legal regulations to discourage bad behavior of system owners.

- Some damage compensation may be possible, but the court trial may be long an the evidence may not be sufficient. Users should invoke such mechanisms only as a last resort and rather try to avoid problems based on evaluation of reputation.
- The system should not be under a control of a single entity, but parts of the system could be centrally controlled. Appropriate legal arrangements should be in place in case the controlling entity is not behaving as expected.
- Some privacy rules are mandated by a centralized entity (e.g. government) that has the power to enforce them. These rules should set a basic framework and responsibilities of system owners instead of regulating every aspect of data protection.
- The global rules should be used to balance free speech, privacy and control.

The environment of responsibility is the approximation of the realspace organization of most modern societies. It attempts to find the equilibrium between colliding forces and keep the system in that equilibrium by self-balancing mechanisms. However the intervention of a regulation force may be necessary to prevent the environment to shift toward anarchy or authoritarianism.

The environment of responsibility differs from the environment of unregulated free market in the aspect of regulation of key concepts. We expect that the unregulated free market will create entities that eventually become too strong to enforce their own rules and therefore deliberately shifting the environment towards anarchy or authoritarianism as they can get better advantage in these situations.

The challenge is to avoid the creation of channels and trust silos that limit the scalability of the relationships in this environment. We hope that the reputation system can help enhancing the ad-hoc interactions.

4.2 Privacy

The privacy is traditionally defined as “the right to be let alone”. The concept of privacy protection was described in the late 19th century by Samuel Warren and Louis Brandeis [46]. The privacy protection was based on a tort law, the right to seek compensations of privacy invasions. The movement to protect privacy in the late 19th century was partially a reaction to a technology shift: a development of instant photography and the rise of sensationalistic reporting in journalism.

Now we are facing similar technological shifts. The developments in data processing and networking technology make it increasingly easy to gather and process any kind of information, including personal information. The personal data are gathered and processed by both public and private sector organizations. The processing is done often without consent or even a knowledge of the person that the processed data describe. Errors and incorrect information is occasionally introduced in the data flows and it is very difficult for any single person to correct the information in all the replicas of the data, especially if such a person is not aware of the data existence. The erroneous data in the databases may have severe impact on the lives of ordinary persons, being denied a credit or loan, experiencing difficulties finding a job, etc.

Daniel Solove [47] is using the metaphor of Franz Kafka's novel The Trial to describe the current situation. The protagonist of the novel is arrested without being informed about the reason and without being set to the jail. A trial is being held against him, while he does not know anything about it, he has no contact with the judges, he does not know the details of the conviction and he cannot defend himself. He is executed at the end of the novel without any explanation.

The Kafka metaphor well describes the modus operandi of organizations that collect and use personal data without the consent of the persons that the data describe. Clients are being denied a credit or loan without any further explanation, while the decision was based on the data collected by a credit reporting agency. The applicants for a job are refused because of data from a background check that was conducted by a third-party organization, while the applicant is not informed about the data collected. Many small trials are being held against us without giving us a change to object and to defend ourselves.

Solove argues that the current situation is “architecture of vulnerability” because ordinary people are vulnerable to the mishandling of private-sector organizations. He also explains that the free market mechanism alone will not make the situation considerably better as there is a power imbalance between the people and the organizations and the people cannot negotiate the fair terms for handling of personal data. He argues that an architectural approach is needed to remedy the problem of privacy in the information age. He proposes architecture based on Fair Information Practices [48] which was a one of the sources for OECD guidelines for the protection of privacy. Fair Information Practices recommend following rules for personal data processing:

- There must be no personal-data record-keeping systems whose very existence is secret.
- There must be a way for an individual to find out what information about him is in a record and how it is used.
- There must be a way for an individual to prevent information about him obtained for one purpose from being used or made available for other purposes without his consent.
- There must be a way for an individual to correct or amend a record of identifiable information about him.
- Any organization creating, maintaining, using, or disseminating records of identifiable personal data must assure the reliability of the data for their intended use and must take reasonable precautions to prevent misuse of the data.

These rules are partially reflected in the EU personal data protection directive [49]. In addition to these rules Solove proposes practices to protect the rights of individuals. Based on this proposal we define a set of rules that supplement those proposed by Fair Information Practices:

- Opt-in for processing of personal data instead of opt-out. The personal data may be processed only if the user explicitly expressed consent with the processing. The user must always have a choice to refuse processing of his personal data and still be able to get the service (if possible).

- Liability. The companies collecting the data should be held responsible for the accuracy of the data, maintaining proper security of the data and honoring the data processing policies agreed with the users. The companies should compensate the users in the case of breaking those rules even if no direct harm was caused to the users.
- Enforcement of the rules be government. The basic personal data processing rules should be enforced by the legislation and there should not exist any possibility to contract around them.
- Let users participate. The companies gathering the information should inform the users about the state of their information. The users should be informed on a regular basis as well as on as-needed basis if there is an unusual change in the information. The users should have an easy way how to challenge correctness and completeness of the gathered and stored information.

We will take these rules as well as the rules based on Fair Information Practices as requirements for data protection and privacy properties of the architecture proposed in this document. We consider these rules as important tool to balance the trade-off between privacy and free speech. Such balance is needed for effective cooperation and it is an important aspect of the environment of responsibility.

4.3 Trust and Reputation

In usual realspace interactions the behavior of person is guided by the previous experiences. When two persons are interacting, their attitudes will be determined by the previous experience with the interaction partner or by the information about the partner acquired from other entities. Realspace persons are summarizing the experience and available information to form an opinion about the interaction partner. Such opinion determines of the information provided by the partner will be believed, to estimate how likely will the partner keep his promises and to determine the overall risk in dealing with the partner. The situation in the realspace is seldom black-and-white, an individual will rarely trust the partner completely or do not trust a single word. The level of confidence in the partner usually oscillates between the extremes.

However the current situation in the cyberspace is different in nature. The user of the computer is usually completely trusted (e.g. system administrator), not trusted at all (e.g. unauthenticated user) or falls in one of the few preset confidence groups determined by system permissions (e.g. user roles). The level of trust in the user is usually determined by an authoritarian entity (system owner, certification authority), not the interacting parties using that system for communication.

An approach similar to the mechanism of realspace interactions is also needed for cyberspace interactions. However, a long-term relationship is necessary to build up a trust between entities. The system that is based on direct trust between entities would have tendency to centralization. The interactions would be mediated by several well-known entities that are able to attract long-term relationship with many users. These entities may eventually become too powerful to dictate their own rules and decide who should be trusted and who should not, which would lead to an authoritarian environment and it can limit the network effect.

The ideal mechanism must allow reliable interactions without the need of a prior long-term relationship to build up a trust between parties. However, it should be able to enforce balance in the interactions, so all involved parties can appropriately assess the risk of a specific interaction. Distributed reputation mechanism seems to be appropriate to fulfill these goals [50]. Reputation is a mechanism that evaluates past actions for the purpose to be used in future decisions. Such effect is described as "shadow of the future" [51] and it can influence current behavior of people by threatening to punish bad behavior in the future.

A game-theoretic approach to model Internet interactions based on examination of the record of past interactions is provided by Friedman and Resnick [54]. It is shown that the availability of cheap pseudonyms is harmful to the level of overall cooperation in the network. Friedman and Resnick propose the creation of once-in-lifetime certificates to mitigate the effects of cheap pseudonyms. However, the paper is not specific about feasibility of such certificate system in the Internet environment. Also, problems like the loss or abuse of once-in-a-lifetime certificate are not considered in the paper. We consider these problems as practical obstacles for the usability of such system in the Internet environment.

The actual mechanism to maintain the reputation values is outside the scope of this work, however we specify following requirements, assumptions and ideas about the properties of a practical reputation mechanism for the use in globally-distributed information systems:

- Any persona can express opinion on the reputation of any other persona.
- It is expected that the personas of end users are maintained by specialized entities (identity providers), by usual businesses that has first-hand relationship with the user (e.g. telecommunication companies) or similar organizations. However we do not put any constraint on who can maintain personas. The extreme case of self-maintained persona should always be possible, however we consider it non-practical. We will refer to the entities that maintain other personas as *identity agents*.
- We expect that the reputation of a persona is primarily expressed by the identity agents. Identity agents will gather feedback about the behavior of personas that it maintains and summarize that feedback into an integral value that is expressed to other interested parties.
- Identity agents represent themselves in a form of a persona and they have their reputation as well. The opinions that influence reputation of identity agents are expressed by other entities, usually the businesses that provide services to end users, independent rating organizations, etc. The opinions about identity agents should reflect whether the reputation values provided by the agent matched the actual behavior of the person.
- The reputation of the source of reputation feedback should be taken into consideration. The feedback from the source with good reputation should have much higher weight than a feedback from a source with bad reputation. That also applies to the reputation provided by the identity agents. The reputation of the persona maintained by the identity agent should be combined by the reputation of the identity agent before use.

- Identity agents are motivated to objectively maintain reputation of the personas and properly handle the reputation feedback. The mishandling of reputation feedback may negatively affect identity agent's ability to predict the behavior of its users and therefore also his own reputation. Bad reputation of identity agent affects the relevance of reputation of all personas maintained by that agent. That will mean that the identity agent must lower the price on his services (e.g. maintaining a persona), which is a business disadvantage. Therefore agents are motivated to avoid such situation.
- Providing the reputation feedback must be easy, ideally it should be part of the client software (e.g. web browser) for end users and be integrated with business software (e.g. Client Relationship Management system) for organizations.
- The interacting entities will risk fraction of their reputation in each interaction. In case of successful interaction the reputation of both entities will likely get better. In case of unsuccessful interaction the reputation of one or both entities will likely get worse.
- Each entity will choose its own set of sources to evaluate the reputation of parties it interacts with. We expect that the entities will choose sources that are close to them culturally or business-wise, sources that share common values and world-view.
- To introduce new persona to the system may be difficult. If the persona would have low or neutral reputation, it will be very difficult to get the appropriate level of service, as we expect that the best service will be reserved for entities with good reputation. This can be solved by setting reputation of new personas provided by identity agents to a good value and charging the person appropriate price to address the risk of bad reputation imposed on identity agent in case of misbehavior of a new persona. If the subject controlling new persona would behave badly, the reputation of the persona will soon get bad and the persona will be useless. The misbehaving user will need to invest in a new persona. This approach may eventually get very expensive for misbehaving user, thus motivating him to stop such behavior. This may be emphasized if there will be only a few identity agents with high reputation and these agents will charge extra fees for additional personas belonging to users that already have low-reputation personas.

We believe that the reputation infrastructure based on the specification above would provide a framework that can utilize the network effect while still allowing interaction parties to appropriately address the risk of cyberspace interactions. We believe that proposed approach will be effective to stop spam mail and similar low-cost damage that is being done on a large scale. However, we expect that it will not be effective to stop highly motivated targeted fraud and criminal behavior, therefore a legal system must still be in place to address such behavior. We do not consider that a problem, as such high-profile fraud and criminal acts are not frequent and it is effective to resolve them by legal means.

The description of approach described above cannot be considered by any means precise or complete, it is just an illustration of a concept. We acknowledge that the exact mechanism for processing reputation is not known and that it is still an open problem. However we believe in the correctness of the approach and henceforth we will assume

that an effective solution to this problem could be found and that the solution will have the properties described above.

Windley et al. [52] note that there is a natural trade-off between reputation and privacy. They argue that as reputation is calculated from the record of past interaction, revealing such record means that part of the subject's privacy is lost. Therefore it would be desirable not to reveal the record of past interaction to more parties than necessary. Considering the proposal above only the identity agents need to examine the record of past interactions. Other entities will rely on the summation of reputation provided by the identity agents.

Trust can apply only to realspace subjects. Trust is person-to-person relation with psychological background. The concept of trust is not applicable to computer systems, as computers have no psyche, no consciousness. Computers behave as programmed. Therefore we can discuss the belief that they are programmed according to implied expectations. We can assess the assurance level that the system behaves as specified, based on software development practice and testing results. We even can evaluate the trust to the authors of the system that they did their best during system implementation. We can trust system administrator that the system is well maintained and properly configured. But it does not make sense to trust the computer system as a cyberspace entity. However, it may prove useful in practice to assign a reputation to the computer systems and similar personas. However, such reputation must be considered as a reputation that results from the actions of system developers, administrators and owners – the realspace entities that control the behavior of such system. Therefore it is useful for any persona to feature reputation.

The reputation can be seen as global or subjective according to Sabater and Sierra [53]. The reputation value can be globally announced by some entity and used by other entities or any entity can compute its own value of reputation. We consider this distinction purely technical. We argue that any entity will evaluate the trustworthiness of other party based on available input values, while global reputation of other party may be one of these values. Therefore we consider the use of the reputation to be subjective in any case, while we admit that the publishing and distribution of reputation might be global.

An appropriate level of trust is necessary for efficient cooperation. The cooperation of the parties that do not trust each other is burdened with high overhead of contractual constraints, management controls and continual checks. The lower level of information about the trustworthiness of the other party means higher risk for the transaction and therefore higher cost to the controlling and remediation mechanisms rather than investing to the core subject of the interaction. We consider reputation as a key mechanism how to evaluate trustworthiness of the subjects. Interaction partners may user reputation values to guide trust-related decisions about the other partner without requirement for a long-term relationship. Such a method will be necessary to maintain efficiency in globally distributed environment.

4.4 Conclusion

The goal of the architecture proposal provided in this thesis is to encourage creation of environment of responsibility, as described in section 4.1.3. Partial bias from that

environment should be possible, to allow for flexibility of changing needs of Internet users. The proposed architecture must be able to support the privacy of the users as described in section 4.2 and it should follow the basic principles of trust and reputation as stated in section 4.3.

5 Architecture of the Internet

The development of Internet is an effort of a large and diverse group. Computer scientists, application developers, users and many others, all are taking their part in forming the Internet. The development of protocols and applications is not controlled by any central authority. Several standard bodies are strongly influencing the design of the Internet, but even those organizations are not a position of force and control. Such environment contributed to the evolutionary approach to the architecture of the Internet [68]:

> *The Internet and its architecture have grown in evolutionary fashion from modest beginnings, rather than from a Grand Plan.*

There is no document that describes the invariable architectural principles of the Internet, as there are no such principles [68]:

> *Principles that seemed inviolable a few years ago are deprecated today. Principles that seem sacred today will be deprecated tomorrow. The principle of constant change is perhaps the only principle of the Internet that should survive indefinitely.*

As the design of Internet is guided by evolution and constant change, the occurrence of design problems and architectural inconsistencies is inevitable. Following sections describe the current state of the Internet and World Wide Web architecture and identify the most obvious design and architectural issues.

5.1 TCP/IP Protocols

The TCP/IP protocol family was design as a simple communication protocol for interconnecting networks. The development of the Internet protocols based on the TCP/IP protocol family was guided by the continual evolution, rather than visionary plan. The evolutionary approach is described in RFC1958 [68]:

> *"Engineering feed-back from real implementations is more important than any architectural principles."*

Since the popularization of the Internet in 1990s and the start of massive commercial utilization of “The Network” the design of the Internet was heavily influenced by commercial companies. The design of the network products that were introduced on the market was guided by the market demand, rather than by any desire to keep the Internet properly architected. An example of such approach could be a typical firewall device which deliberately allowed network site to break compliance with TCP/IP specifications by dropping essential network packets (such as ICMP packets).

One of the most relevant problems of the Internet protocols is the use of IP address on higher system layers. The meaning of IP address is to identify the network endpoints (network interfaces) on the network layer. However the IP address is used for many different purposes for whose it is not suitable:

- IP address is used to determine user's physical or network-wise location of user's terminal equipment. The location of a user cannot be reliably determined if application-level gateways (proxy servers), Network Address Translation or similar mechanism is used. In these cases the IP address of the connection source (proxy server) does may not correspond with actual location of the user. Example of the manifestation of this problem is a WWW site that displays the text in Hebrew, while the user is actually located in Slovakia and he is using a company proxy server located in Israel. The user's realspace location or the network-wise cyberspace location of end user's terminal device cannot be reliably determined from the source IP address of the connection. The use of the IP address to estimate network-wise cost of communication with the user for optimization purposes may still be correct.
- IP addresses are used to track user sessions. Such approach will fail for mobile users. The IP address of user's device is changing frequently as mobile user changes networks. Example of manifestation of this problem is the interruption of video streaming for a user traveling in a car. User's device is switching from one mobile hot-spot to the other as the can moves across the city, loosing streaming session with each IP address change. Similar problem appears with firewalls that identity user based on his IP address. Such identification is invalidated if the user moves through the network.
- IP address is used to define endpoints of application-layer services. If a single IP address is used to address the endpoint, it fails for multi-homed or replicated services. The IP address can only address one network endpoint. A service that is provided on several network endpoints (several network interfaces, servers, sites) cannot be addresses by a single IP address in a satisfactory manner. This problem can be partially amended by the use of DNS names instead of IP addresses. But the mapping of a DNS name to an IP addresses is usually static and does not reflect the context of the end user. Therefore if a service is provided by two network-wise distant sites, there is no widely-deployed mechanism to determine which one is better for the user to access.

The use of the IP address as described above by application layers should be considered a layering violation. Application-layer mechanisms should be used to provide information about user's location, track user sessions, etc.

5.2 World Wide Web

World Wide Web originated in early 1990s as an distributed hyper-text system, based on TCP-based Hyper Text Transfer Protocol (HTTP) and SGML-based Hyper Text Markup Language (HTML). It later evolved to a generic delivery mechanism for information objects. At the time of this writing is World Wide Web perceived as a general-purpose “information space” [58]:

> *The World Wide Web (WWW, or simply Web) is an information space in which the items of interest, referred to as resources, are identified by global identifiers called Uniform Resource Identifiers (URI).*

The principles and protocols of current World Wide Web architecture have evolved during late 1990s. The architecture was guided by the Representational State Transfer (REST) architectural style described by Fielding [57]. The basic architectural elements of REST architectural style and their effects on World Wide Web architecture are summarized in following paragraphs.

The REST style is based on the Client-Cache-Stateless-Server style. All interactions are asymmetric, with the roles of client and server clearly distinguished. The server is passive (reactive) and cannot initiate interaction. All interaction are limited to only those initiated by client, therefore asynchronous notifications of events from server to clients is not possible. This limitation applies equally to the WWW architecture and it limits development of applications whose the nature is distribution of asynchronous events and messages (e.g. instant messaging, news distribution, etc). This limitation is in practice addressed by polling mechanisms, such as RSS [69] and AJAX [70].

The server component of the REST architecture is supposed to be stateless [57]:

> *All REST interactions are stateless. That is, each request contains all of the information necessary for a connector to understand the request, independent of any requests that may have preceded it.*

The statelessness is endorsed as one of the key architectural principles of REST. When applied to the architecture of World Wide Web, the use of any state mechanism such as HTTP Cookies and frames is considered an architecture mismatch. However the assumption of statelessness can hold only if the resources are immutable and fail if any of them can be modified. The REST architecture allows for modification of resources, especially when applied to the WWW in a form of PUT, POST and DELETE methods of the HTTP protocol [56]. If one of the interactions changes state of the resource, all subsequent interactions depend on the result of the interaction that caused the state change. For that reason the interactions in the REST architecture cannot be considered stateless, as the state is present in the resources. Fielding does not address this problem in his description of REST, however he notes that the use of caching for resource representations may provide erroneous response. Such an error would not be possible if REST would be entirely stateless, in other words if the response would depend only on the information in request.

The architectural inconsistency in REST was not apparent in 1990s and early 2000s. Majority of the resources on the World Wide Web at that time were not highly dynamic and their frequency of change was very low as compared to the usual cache expiration intervals. The HTTP protocol provided controls for disabling the caches and these controls were used intensively for the purposes of web applications and dynamically-created web content as it gained popularity in 2000s.

The concept of “writable web” is expected to change the nature of web interactions even more. Current World Wide Web applications are most focused at distributing information from resource owners to consumers. However the concept of writable web assumes that much more user than just the owner of a resource will contribute to the information. Wiki applications, commenting on blog posts and social networking sites are examples of writable web approach. With proliferation of writable web applications it is expected that the state of web resources will be changed even more frequently and in a less coordinated fashion as it can be observed now. It is expected that the negative

effect of stale cache content may cause that most resource representations transferred by HTTP protocol will be marked as non-cacheable.

Uniform interface is another basic principle of REST architecture. However it was only partially reflected to the architecture of World Wide Web. The URI [32], HTTP [56] and HTML [71] specifications were supposed to define the uniform REST-like interface for the World Wide Web. However these definitions include a considerable degree of extensibility of the definition, focusing on the syntax of the interface and defining only the minimal semantic meaning when needed. While this approach allows to use the WWW mechanism for a broad range of applications, the definitions provided in URI and HTTP specifications are closer to definition of a network layer rather than application interface. HTTP provides means how to denote the metadata of the transferred data (e.g. media type), but it does not constrain the transferred data in any way. The URI specification defines the common syntax for identification of resource, while not precisely defining what is meant by “resource” and not constraining the semantics of the identification scheme.

The REST architecture mandates layered system approach, which will allow for intermediaries that can understand the unified interface. While this is a valid requirement and is well reflected in the design of open public World Wide Web, the situation is getting worse when security is applied. The only practical security mechanism for World Wide Web is HTTPS protocol [31]. This protocol provides channel-level protection for the entire HTTP interaction. As the details of the HTTP interaction are hidden from intermediaries, the layered design constraint cannot be applied. The solution of the WWW architecture is to allow tunneling of HTTPS protected communication through intermediaries by tunneling it in plain HTTP using HTTP CONNECT method. While this solution in practice provides a features similar to those of layered system, most of the advantages of layered design are lost.

The REST architecture includes the Code on Demand principle, which is expected to dynamically extend the capabilities of the client. However, the mechanisms for Code on Demand are not part of any basic WWW standard or interface definition. Few industry solutions for Code on Demand appeared “in the wild”, most notably Java Applets and JavaScript. However the virtual machine for these languages is not a part of the standard WWW interfaces and therefore their use must be considered an optional extension to the functionality of World Wide Web.

Following section are focused on the detailed description of the issues of current practical World Wide Web architecture.

5.2.1 Resources and Identifiers

The concepts of Resource and Uniform Resource Identifier (URI) are central concepts in the architecture of the World Wide Web. However only vague definitions of a resource are available [58]:

> *By design a URI identifies one resource. We do not limit the scope of what might be a resource. The term "resource" is used in a general sense for whatever might be identified by a URI. It is conventional on the hypertext Web to describe Web pages, images, product catalogs, etc. as “resources”. The distinguishing characteristic of these resources is that all of their*

> *essential characteristics can be conveyed in a message. We identify this set as "information resources."*
>
> *[...]*
>
> *However, our use of the term resource is intentionally more broad. Other things, such as cars and dogs (and, if you've printed this document on physical sheets of paper, the artifact that you are holding in your hand), are resources too. They are not information resources, however, because their essence is not information. Although it is possible to describe a great many things about a car or a dog in a sequence of bits, the sum of those things will invariably be an approximation of the essential character of the resource.*

It is obvious that *resource* may be a realspace object and that resources are identified by URIs. That implies that one of the intents of the WWW architecture is to identify realspace objects by URIs. The World Wide Web architecture document [58] mentions the concept of URI owners and it recommends a good practice for URI owners:

> *Available representation: A URI owner SHOULD provide representations of the resource it identifies.*

It follows that realspace objects should have representation in cyberspace maintained by the owner of the URI. Such a representation is always subjective. There is no assurance that the URI owner is also the owner of the realspace "resource", therefore the representation of the "resource" provided by the URI owner can be harmful.

The only wide-spread security mechanism for the Web is HTTPS [31]. This mechanism has provisions how to assure the user agent that the received content was transmitted by the controller of the DNS domain which was used to create the URI identifying the resource. This mechanism provides no other guarantees in regard to the origin of provided information, its trustworthiness or applicability.

As the representations of a realspace objects in cyberspace are always subjective, it is very important to distinguish between the reference to the realspace object and its cyberspace representation. For example if someone provides a harmful description of an organization, that organization would like to refer to that *description* as being "harmful" without the risk to referring to the *organization* as "harmful". From the definitions above it a obvious that URIs can identify both realspace objects and cyberspace descriptions of these objects. But to distinguish these concepts by simple examination of the URI should not be possible, as implied by the following guideline [58]:

> *Opacity: Agents making use of URIs SHOULD NOT attempt to infer properties of the referenced resource.*

This problem was recognized [72] and a solution was proposed by the W3C Technical Architecture Group [73] by not allowing to provide a representation of a resource that is not an information resource:

> *The W3C Technical Architecture group eventually decided to resolve the architectural problem that if an HTTP response code of 200 (a successful*

> *retrieval) was given, that indicated that the URI indeed was for an information resource, but with no such response, or with a different code, no such assumption could be made. This compromise resolved the issue, leaving a consistent architecture.*

Although it is claimed in the above citation that this decision leaves a consistent architecture, some issues still remain. The most obvious problem is that the above decision makes generic concept of URI dependent on the HTTP protocol definition. However URIs are supposed to be protocol independent identifiers [32]:

> *A common misunderstanding of URIs is that they are only used to refer to accessible resources. The URI itself only provides identification; access to the resource is neither guaranteed nor implied by the presence of a URI. Instead, any operation associated with a URI reference is defined by the protocol element, data format attribute, or natural language text in which it appears.*
>
> *[...]*
>
> *Although many URI schemes are named after protocols, this does not imply that use of these URIs will result in access to the resource via the named protocol. URIs are often used simply for the sake of identification.*

The described situation can be seen as the conflict in the web architecture and a source of potential deeper problems. The protocol-independent principle of URI design is often spoiled by assuming the ability to dereference the URI and get appropriate representation of the resource it identifies. For example the URIs with *http* scheme (HTTP URIs) cannot be both independent of HTTP protocol and being universally resolvable using HTTP protocol, as both HTTP protocol and HTTP URIs are obviously at the same architectural level (both defined in RFC2616 [56]). Possible solution might be definition of appropriate URI scheme on lower level and definition of HTTP protocol that will depend on that URI scheme. In that way other protocols can depend on the same URI scheme without the need to depend on HTTP protocol, which may provide a space for innovation.

From the orthogonality principles proposed by the WWW architecture document [58] it follows that the data formats used for resource representations should be independent from the URIs. However, the fragment segment of the URI by definition depends on the specific representation. Therefore the concept of URI can be seen as a leaky abstraction that leaks the details of both access protocol and representation data format.

5.2.2 The Meaning of Resource

The definition of a resource is very vague. It is essentially defined as “whatever might be identified by a URI” [58]. This may lead to almost anything to be considered a resource. Even resource representations may by themselves be resources (they are often identified by URIs already). Such a recursive principle gives great freedom of choice for system implementers, but it may become very confusing. For example if someone will receive an URI in mail message and opens it in his browser a picture of a woman will be displayed. If the user will reply to the mail message commenting that it is “terrible”, what could it mean? Does it mean that the picture that was displayed on his screen was

in low resolution and could be barely seen? Or does it mean that the photographer made a poor job and made a bad photograph of otherwise pretty woman? Or does the woman that was the model for the picture looks bad? Or does the user mean that the person on the picture might present good looks but she is not the kindest person on the face of earth?

This situation may be easy to resolve for humans, given a specific context of the URI in the message and the response. But if an automated reputation system would interpret the negative feedback of a user on a specific URI, it would be difficult to distinguish whether the image processing algorithm, photographer's skill, model's look or model's personality was meant as the target of the opinion.

The decision of W3C TAG [73] that only directly dereferencable URIs may be considered URIs of information resources have addressed the problem only partially. Given the previous example we still cannot distinguish if the user expressed his opinion about the low quality of displayed image or inappropriate lighting and composition of the photograph. This situation is made even more confusing by W3C recommending to avoid arbitrary URI aliases [58] for the same resource while at the same time recommending different URIs for something that can easily be considered different representations of a single resource [74].

The situation may be partially addressed by using HTTP redirects [56], especially the HTTP response code 302 (found). A resource may be identified by a primary URI, which will respond to dereferencing with HTTP 302 response code, indicating the URL of resource representation in the *Location* header. However, the redirects do not provide metadata about the meaning of the primary URI. Therefore the client still cannot distinguish whether the URI used was meant to identify the abstract concept, specific person or a specific photograph of the person.

Similar problem can be illustrated by the common situation of displaying a HTML form for user log-in when a user tries to access a protected resource. User enters resource URI to the browser, but instead of resource representation an authentication HTML form is displayed. However HTML form is definitely not a resource representation and returning the form in 2XX HTTP response may be misinterpreted as resource representation. A user could assume that the URI used in the browser was in fact URI of the authentication page. But even if HTTP redirects are used the situation of providing authentication page, which is not a representation of requested resource, cannot be distinguished from an alternative representations of the requested resource [74].

A new HTTP redirect response code or a change of definition of existing codes would be needed to indicate the distinction between the situations above. However, the HTTP response codes cannot resolve the problem at its core, which is fuzzy definition of resource and no standard way how to determine what is identified by URI.

5.2.3 URI Aliases and QNames

The World Wide Web Architecture [58] document proposes a practice to avoid URI aliases:

> *Avoiding URI aliases: A URI owner SHOULD NOT associate arbitrarily different URIs with the same resource.*

However URI aliasing is a common practice on the web today. It is a common practice that following URIs identify the same resource (filesystem directory):

```
http://example.com/dir
http://example.com/dir/
http://example.com/DIR/
```

Similarly the URIs in different schemes may represent the same resource:

```
http://example.com/myvideo
https://example.com/myvideo
rtsp://example.com/myvideo
```

This practice is clearly in conflict with the practice proposed by World Wide Web Architecture document [58], however it is deemed acceptable by at least some members of W3C TAG [75]:

> *It's appropriate to note here that in cases where the necessary form of client/server interaction for a particular kind of information resource, for example streaming video, cannot be provided by the protocols normally associated with existing URI schemes, new schemes may be appropriate.*

We consider the above practice of using URIs in different schemes to identify the same resource as harmful. The identification of access method should be determined by the client, it should not be a part of resource identifier.

We observe that the URI aliasing is considered harmful, because there is no practical way how to determine URI equivalence and/or canonical URI for a resource. If a mechanism for URI equivalence and the concept of canonical URIs would exist, the negative effects of URI aliases may be eliminated or at least kept to the minimum.

The eXtensible Markup Language (XML) [76] used for data representation on the World Wide Web introduced the concept of namespaces. The XML namespaces are identified by URIs and were originally designed to provide namespace separation for XML element and attribute names. However the XML namespace mechanism is used for identification other resources as well, for example for identification of services. The name in a XML namespace is called Qualified Name (QName) and it is composed from URI-formatted namespace name and free-form local part. The QNames are not URI. However the World Wide Web Architecture document strongly recommends the use of URIs for resource identification [58]:

> *Identify with URIs: To benefit from and increase the value of the World Wide Web, agents should provide URIs as identifiers for resources.*

This apparent inconsistency in the World Wide Web architecture is later addressed in the same document by mandating following practice:

> *QName Mapping: A specification in which QNames serve as resource identifiers MUST provide a mapping to URIs.*

However this practice is seldom followed, as there is no universal or recommended way how to map QNames to URIs. Such a universal mapping is difficult to design with sufficient universality, as the author of the specification using QNames should not constrain the format of URIs used for namespace definitions.

We see this duality in using QNames and URIs to identify the same concepts (resources) as harmful to the architecture of World Wide Web. We account the

difficulties in mapping between QNames and URIs to the unnecessary flexibility of generic URI format, which inhibits the attempts to design a universal mapping mechanism.

5.2.4 Persistence

The URIs using the *http* scheme are considered by World Wide Web Consortium (W3C) Technical Architecture Group (TAG) to support persistence. The draft finding of the TAG [75] claims the following statement to be a fact about URIs with *http* scheme:

> *http: URIs support persistence as well as it is in-practice possible to do so.*

However, the persistence of URIs with *http* scheme (HTTP URIs) depends on assignment of DNS name or IP address. The IP address assignment cannot be considered persistent, as IP address assignment is in many cases not controlled by the URI owner. The IP address can be changed as a reaction to events independent from the actions of URI owner (e.g. restructuring of service provider's network, migration to IPv6, change of service provider, etc). DNS name assignment can be made reasonably persistent in the mid-term scope (few years) for well-established organizations. However it is difficult for individuals to obtain a DNS domain under their control. Therefore it is difficult to implement persistence for HTTP URIs scheme for individual users.

A *hosting* of identifiers with well-established organization may be an alternative to provide some persistence for identifiers of individual users. An organization may assign a DNS subdomain or a portion of URI hierarchy for the use by individuals. For example identifier namespaces assigned to individual user jack may look like the following:

```
http://jack.examplehosting.com/...
http://examplehosting.com/jack/...
```

The drawback of this approach is that the hosting organization is in fact owner of the URI namespace. The portability of the identifiers from one hosting organization to the other is difficult and it requires cooperation of both hosting organizations. The situation may be compared to the situation of telephone number portability that had to be often mandated by law to make it possible.

The W3C TAG draft finding on the use of metadata in URIs [77] proposes following practice:

> *Good Practice: URIs intended for direct use by people should be easy to understand, and should be suggestive of the resource actually named.*

This practice may be interpreted to encourage the use of human-readable names in URIs. However, human readable-names are often subject to change. For example company names, department names, user names. Tracking numbers, ISBN numbers, product part numbers are usually quite stable, but people would seldom consider them as easy to understand and suggestive. Therefore such a practice inhibits persistence of URIs.

W3C published a document [78] summarizing good practices to improve the persistence of URIs by making them “Cool”. The document essentially proposes to leave out any redundant and unnecessary information from URIs, but still keeping them human-readable. However, this proposal does not address persistence in situations like change of owner's name, change of resource name, change of numbering scheme, etc.

Additionally the practice recommended by the document is seldom followed. Even the URI of the document itself [78] does not completely follow recommended rules.

The solution might be to provide a human-readable URIs and persistent URIs for the resources at the same time. The human-readable URIs would be intended for interactions with humans (e.g. seeing the URI on the billboard), while persistent URIs will be intended for the use by computer systems (e.g. bookmarking, hyperlinking, etc.) However, such a solution would make the evaluation of URI equivalence very difficult, and it may be considered in conflict in a practice recommended by the WWW Architecture document [58]:

> *Avoiding URI aliases: A URI owner SHOULD NOT associate arbitrarily different URIs with the same resource.*

The conclusion is that URIs with *http* scheme can support practical mid-term persistence for well-established organizations and hosting scenarios. But considering the current situation of DNS name assignment practice the use of HTTP URIs as general-purpose persistent identifiers is not practical.

5.2.5 HTTP URIs

The URIs have generic syntax for hierarchical names. The distributed namespace of HTTP URIs is hierarchical. Starting with scheme as the most significant segment followed by the DNS top level domain, second and other domain name levels followed by the path segments. However, the syntax of HTTP URIs does not cleanly reflect that hierarchy. All hierarchical parts of DNS names are collapsed to the non-hierarchical authority segment. This design decision made the HTTP URIs more readily usable at the time of the original World Wide Web design. Such a decision introduced artificial distinction between the hierarchical host name and hierarchical path. This limits the delegation capabilities of the identifiers. The decision which part of the naming hierarchy should be expressed in the host name and which part to express in path segment needs to be made at the time the identifier is assigned. It cannot be easily changed while the identifier is used and still maintain the resolvability of the identifier.

The W3C TAG is claiming [75] that it is not a practical limitation. In case of re-structuring the network, for example if a single host was distributed to several hosts, the original host can still accept the requests to dereference the URIs and respond with HTTP Redirect or proxy the request to new hosts. However such a solution may have operational consequences. The original host may become a bottleneck, especially if the motivation for network redesign was poor performance of the system.

The HTTP URIs cannot be considered pure identifiers, as they leak several implementation-specific details. They define the access protocol to use. Although it argued by W3C TAG [58] that the `http` scheme prefix should not be understood as definition of access protocols, the practice of distinguishing the access protocol from URI prefix is considered acceptable by the document published by the same organization [75]. The specification of URI [32] states that there is a distinction between URI and URL, but it fails to define a method to distinguish them. The specification of HTTP URIs [56] does not provide such mechanism either. Considering a practice common in the Internet today and the architectural inconsistencies stated above, we

must consider HTTP URIs to be addresses for a specific use with the HTTP protocol and not generic identifiers.

The HTTP URIs define location of the resource. This location is represented as DNS name or IP Address. Therefore HTTP URIs depends on the Internet addressing and naming infrastructure. Based on the definitions provided in chapter 3, we consider any such mechanism to be an *addressing* mechanism rather than *identification* mechanism.

The syntax of HTTP URIs does not follow the same consistent set of rules. For example both the authority and path sections of HTTP URIs have hierarchical structures. The authority section of HTTP URI has the most significant component at left-hand side and the hierarchy separator is a dot character. While the path segment has the most significant component on the right-hand side and the hierarchy separator is a slash character.

5.2.6 Security and Trust

The only practical security and trust mechanism for the World Wide Web is currently HTTPS [31]. It is a protocol providing channel security (confidentiality and integrity protection) and authentication of the connection endpoints. HTTPS is based on SSL/TLS [79] security mechanism. Although the actual cryptosystems used by the SSL and TLS can be flexible, the options provided by the current implementation are quite limited. Several symmetric cryptosystems can be selected for bulk data protection and a few options for asymmetric key exchange are present as well. But the only way how to evaluate the confidence in the key material is to use X.509 public key infrastructure.

The current "trust" structure of WWW is based on several certificate authorities that are either pre-configured or user-configured into web browser software. The browser software will accept any data coming from sites certified by any of these "trusted" certificate authorities as authentic. The certificate authorities usually only check if the organization requesting a certificate owns the corresponding domain name. As the certificate authorities usually do not have any long-term business relationship with the certified organizations, they usually rely on the paper or electronic evidence. Especially in international environment with varied legislation and domain registration procedures, the evidence collected by the some certificate authorities is not difficult to fake. All certificate authorities are considered equal during certificate evaluation, therefore a single certificate authority with a weak certification procedures can ruin security of the whole system.

The fact that organization owns the DNS domain name that appears in the URI is not very helpful for a user to determine if he can trust the site owner or not. It does not provide any information about the reputation of the entity that published the information, whether the information provided by the entity is true or whether the entity can be expected to keep their promises (e.g. promises of privacy).

Current architecture of World Wide Web assumes that information always comes from its authoritative source or a trusted proxy. The HTTPS mechanism is designed to be effective for protection of information under such assumption. However, the usability of HTTPS is limited when a parading of the "static Web" do longer apply. For example if a massive replication and data migration mechanisms are used, there is no single place of data transmission. The requested information may come from any node in the

network that has a replica of that information. There is no single source of data transmission and there are no trusted proxies.

The authentication of users is another problematic aspect of World Wide Web architecture. There are two authentication mechanisms defined for the use with HTTP protocol [61]: Basic and Digest authentication. Both authentication mechanisms are fixed to username/password credentials and are not designed as extensible. The Basic authentication is susceptible to eavesdropping and replay attacks. The Digest authentication improves on that, but it requires a state (nonce value) to be kept at the server between requests, thus violating the statelessness principle of REST architectural style. The Digest authentication is fixed to MD5 mechanism, which must be considered a weakness according the design principles set for this work. Both methods are susceptible to man-in-the-middle attacks, as there is no authentication of the HTTP server.

The usual way for user authentication on the World Wide Web is the use of HTML forms to submit the appropriate type of authentication credentials to server, optionally protected by the use of HTTPS. The server validates the credentials using a local database and if the validation is successful, HTTP cookie containing a random session identifier is set in the HTTP response. The cookie is sent by the client in all subsequent requests. The session identifier can be matched with the session state maintained by the server. However this mechanism is frequently used and it is considered relatively secure for most web applications, it violates the statelessness principle of REST architectural style as it requires to keep session state on the server.

According to the principles of WWW architecture, any resource of relevance should be given an URI. The users of Internet can be seen as resources and they are definitely resources or relevance, therefore they should be given URIs. However, such practice is seldom used and there is no direct support for that in the World Wide Web standards or architecture.

5.2.7 Hidden Assumptions of World Wide Web Architecture

The World Wide Web architecture was knowingly or unintentionally based on a set of assumptions that limits the applicability of World Wide Web. These assumptions were not documented in any official W3C document. The following paragraphs attempt to reverse-engineer some of them and discuss possible problems.

The WWW architecture assumes that the Internet nodes are organized in sites that are controlled by well-established organizations. The sites are assumed to be reliable and operated by skilled staff. The organization that controls the site governs the assignment of URIs to resources and can exercise proper practices for URI consistency, persistence and other desired properties. However, many computers on the Internet belong to individual users. These computers may host resources that should be addressed by URIs. The maintenance of the URI namespace for resources on personal computers could be very difficult, as general public will probably not follow all the best practices of URI assignment. The computers are frequently mobile and are not always-on, which complicates any system that assumes the ability to dereference a URI.

The WWW architecture assumes that each site has assigned a DNS domain name and that the DNS domain name assignment is stable. This assumption fails for personal

computers of individual users, as they seldom have assigned stable DNS name. The assumption is only partially true for well-established organization. It is a usual practice for an organization name to change, for example in the case of re-branding, acquisitions and mergers. While it is usually feasible to maintain old DNS names as well for a short period of time, keeping them indefinitely if usually not desired. The human-readable character of DNS domain name will motivate namespace maintainers quickly migrate all references to a new name and drop the old one.

The WWW architecture assumes that each Internet node has (direct or indirect) connectivity to any other Internet site. Universal connectivity is required for global URI dereferencing. However, connectivity may be limited due to the effect of firewalls, dynamic network address translation or the target node may be mobile or may not be always-on.

The WWW architecture assumes that the information resources are statically located at the sites, they are neither migrated nor replicated between sites. The goal of direct dereferenceability of URIs and the use of DNS names in the URIs limit the ability for dynamic migration and massive replication of resource between sites. The clear distinction of authority and path in URI makes it very difficult to re-structure the site, let alone distributing the resources between sites.

The WWW architecture assumes that the source of transmission of document data is the source of the document content. The HTTPS, the only practical security mechanism for the Web, authenticates the site that transmits the resource representation data. However the site that transmits the data may not be the source of the document, especially in scenarios that include dynamic data replication and migration.

The WWW architecture assumes that each site can authenticate all the users of that site, if such authentication is necessary. It also assumes that no authentication or any kind of information about the user is needed in a vast majority of cases and that most of the World Wide Web content will be publicly available without any constraints.

The WWW architecture assumes that there is and always will be one universal protocol for the World Wide Web. This place is taken by HTTP now and it is assumed that this situation will not change in any foreseeable future.

5.3 Semantic Web

The semantic web [80] is a proposed concept that builds on top of World Wide Web principles. The goal of the semantic web is not a distribution and hyperlinking of human-readable documents, but it is rather focused on the computer-processable description of objects. The objects are supposed to be described in XML-based data languages, such as RDF [81]. The semantic web object descriptions are supposed to be ordinary WWW documents accessible using WWW protocols (usually HTTP).

The semantic web does not store realspace objects. A software system cannot store an apple or a car. It can only store information about the object (object description). The problems related to this subtle difference were already identified by Berners-Lee [72]. It may also be an incomplete claim that semantic web stores the cyberspace objects, as the semantic web itself may only reference them and the objects themselves could be obtained from other systems (using non-WWW protocols).

The semantic web is still under development and it is not yet widely deployed. The opponents [82] of the semantic web concept describe severe obstacles to the feasibility and practicality of the semantic web deployment. Most described problems are caused by the unreliable data in the semantic web. We consider the described problems as a consequence of the subjectivity of crossing the realspace-cyberspace boundary. We argue that the same problems apply to the conventional World Wide Web. However the human consumers of World Wide Web can judge the reliability of the content, while computers cannot.

5.4 Scenarios and Situations

This section provides description of several scenarios and situations that are problematic to implement in current World Wide Web. The scenarios define the actors, prerequisites and setup, describe the steps of the scenario and the problems. Some scenarios provide several variants, each described in its own series of steps.

The purpose of the scenarios is to demonstrate the deficiencies of Word Wide Web architecture in a language that is close to practical deployments. The same scenarios are also used for validation of proposed solution in section 6.6.

5.4.1 Private Photo Sharing Scenario

Actors:
user, friends

Prerequisites:
User has a photo in electronic form.

Summary:
User wants to a photo with friends and only with his friends.

Solution 1 Scenario:

1) User uploads his photo to his server, creating WWW resource from it, assigning a URL.
2) User sends the URL to his friends, using unspecified communication method (e.g. e-mail).
3) Friends will follow the URL to see the photo, using HTTP GET request to get a representation of the photo resource.

Solution 1 Problems:

- The list of friends is maintained at unspecified place, outside World Wide Web concepts.
- The URL has to be communicated to friends by unspecified method, outside World Wide Web concepts.
- There is no practical way how to limit access to the photo resource to the friends only, therefore the photo resource has to be publicly accessible. Only the minimal security can be provided by keeping the URL secret. However, sharing the URL with friends increases chance of disclosing the URL to

unauthorized users, for example by a friend unintentional forwarding the mail with URL.

- An access to the photo cannot be revoked from a friend that ceases to be a friend. The only practical way would be to delete the resource, denying access to the resource even to the authorized friends.

Solution 2 Scenario:

1) User creates accounts for all his friends on his server, distributing credentials to the friends using unspecified secure channel.
2) User uploads his photo to his server, creating WWW resource from it, assigning a URL.
3) User sends the URL to his friends, using unspecified communication method (e.g. e-mail).
4) Friends will follow the URL to see the photo, using HTTP GET request to get a representation of the photo resource. When challenged for authentication, they will provide credentials assigned to them by the user.

Solution 2 Problems:

- The URL has to be communicated to friends by unspecified method, outside World Wide Web concepts.
- The credentials of the friends needs to be maintained, which may prove a very laborious task. Especially considering security best practice of refreshing credentials regularly.
- Friends, having lots of other friends themselves, will need to remember a lot of credentials.

5.4.2 Shopping Collectibles Scenario

Actors:

buyer, seller

Prerequisites:

Seller may or may not own a precious collectible. Buyer and seller do not have any previous relationship or haven't been involved in any previous interaction.

Summary:

Buyer wants to buy a precious collectible that seller claims to own.

Solution 1 Scenario:

1) Buyer finds the offer of seller to sell a collectible that he desires.
2) Buyer contacts the seller. They agree on the price.
3) Buyer sends money to seller.
4) Seller receives the money. He decides to cheat, keep the money and does not send the collectible.

Solution 1 problems:

- The seller can trivially cheat. As buyer and seller do not have any previous relationship, it may be very difficult to avoid the misbehavior of seller.

Solution 2 Scenario:

1) Buyer finds the offer of seller to sell a collectible that he desires.
2) Buyer contacts the seller. They agree on the price.
3) Seller sends collectible to buyer.
4) Buyer receives the collectible. He decides to cheat, keep the item and does not send the money.

Solution 2 problems:

- The buyer can trivially cheat. As buyer and seller do not have any previous relationship, it may be very difficult to avoid the misbehavior of buyer.

Discussion:

It is possible to implement a system based on reputation even in the current World Wide Web, however it is only practical if buyer and seller operate on the same system. For example if both of them maintain an account on the same site, exhibiting good reputation. Such an approach however limits the scalability of the system to the capacity of a single system, which is plainly not suitable for the use in Internet environment. In some situations it may be possible to interact if only one of the actors has established good reputation with a specific site. The actor without good reputation has to send his part of the deal first. However, even if the deal went well, the actor without an reputation can harm the other party. Having nothing to lose he can submit a negative opinion about the deal. For example he can claim the he haven't received the collectible or money and requesting to cancel the deal. The motivation for such misbehavior will increase with the value of the item being traded.

5.4.3 Resource Rating Scenario

Actors:

reviewer, editor, reputation system, search engine

Prerequisites:

- A URI is assigned to represent an old house, requests to that URI will return the representation of the house in a form of digital photograph.
- Photograph of the house is taken only for documentation purposes, without any artistic merit. The composition and lighting were not considered important and it is even slightly blurred.
- Reviewer reviews a photograph of the house and has an opinion about the house age, location and aesthetic quality.
- Reviewer has established relationship with reputation service (e.g. by creating an account).

- Search engine is integrated with reputation service, ordering search results based on the reputation scores provided by reputation system.

Summary:

Reviewer wants to express that opinion to help potential buyers evaluate the quality of the house. Editor wants to find quality photographs of old houses to use as an illustration in his magazine.

Solution Scenario:

1) Reviewer logs in to the reputation service.
2) Reviewer will express his positive opinion about the house, rating the resource that represents the house with "excellent" rating.
3) The reputation service will process that rating, combining with previous rating or rating of other reviewers. Reputation service will expose the (positive) reputation score of the resource (identified by URI) to the public.
4) Search engine will index the resource representing the house. The search engine will check the reputation score of the resource (identified by URI) with the reputation system and associate that with the URI in the created index.
5) Editor queries search engine for the photographs of houses.
6) Search engine will use the index and the reputation scores to satisfy the query. It will present results to editor. The photograph considered in previous step will be among the first results, as it exhibits high reputation score.

Solution problems:

- The obvious problem is that editor looking for a quality photograph of houses will end up with a poor photograph of excellent house. That is not what the editor was looking for.
- If the reputation system would be bi-directional, the editor will most likely express strongly negative feedback about the poor photograph, not aware that the resource in question actually represented the house, not the photograph. This will harm the reputation score of the house, lowering its market value.
- There is no way how to affect individual actors in case they are doing poor job, as none of the actors is identified by URI. Therefore the motivation of reviewer to provide good review is weak, as well as motivation of a reputation system to correctly compute the scores.

5.4.4 Fake Breaking News Scenario

Actors:

readers, publisher

Prerequisites:

- Publisher makes up a story about an event that is likely to gain wide attention, may have severe impact on general public but it is very difficult to verify. For example a story that secret military technology has fallen into the hands of criminals and that government is covering that up to avoid negative publicity.

- Publisher controls a minor media channel (e.g. a blog) with a small group of readers.
- Publisher and readers does not have any strong a-priori relationship.
- Few of the readers follows publisher's channel.

Summary:
Publisher wants to popularize his channel by publishing a fake story. Reader wants to be informed about any dangers that may arise.

Solution Scenario:

1) Publisher publishes the fake story on his blog.
2) The readers will get to the story. Some will not believe it and silently dismiss it, but few readers will believe it and spread the word in a good faith to protect their friends from misuse of secret military technology.
3) The story will spread, triggering both positive and negative reactions. The popularity of publisher's channel will rise. For example with increased number of sites linking to the publisher's blog his page rank will grow, making his blog more likely to appear in web searches.
4) As the story cannot be proven or falsified the attention will cease in relatively
5) The automatic ranking engines that operate on the number of incoming links or number of blog posts cannot recognize positive and negative opinions. They will rate the channel as popular, satisfying the publisher's desire for increased popularity.

Solution problems:

- There is obvious problem for readers. They will face the decision to believe or not believe a non-verifiable story without any cues that can help them make informed decision about the reliability of the information.
- The publisher is likely to gain popularity using this strategy. The negative feedback has only negligible negative effect on the publisher. Any claims that the story is a fake will decrease publisher's credibility only temporarily. And they can even be silently censored by the publisher. However any feedback is likely to increase the rank of his channel, increasing his popularity without any strict time constraint.
- Publisher is not risking much. If the story should be proven to be a fake, the amount of negative feedback will increase. However any feedback is good for the publisher and in any case he is still able to censure the feedback or modify it from negative to positive.

5.4.5 On-Demand Content Distribution Scenario

Actors:
users, distribution nodes, seed node

Prerequisites:

- Content is stored in the form of big file on the seed node.

- Seed node is connected to the network by a relatively weak physical link. It can well handle transfer of one content copy to the rest of the network but it would be heavily overloaded for more than 10 simultaneous transfers.
- Distribution nodes are located in a physically separate locations worldwide
- Distribution nodes are willing to maintain a copy of the content, but only as long as it is needed.
- There is a large number of users at least in order of millions.
- Users are physically distributed across the world and their location is not static.

Summary:

Users want to get the content in a short time duration (several minutes or hours). Users want to get the content as quickly and efficiently as possible. They want to maintain appropriate security while getting the content. Seed node wants to make the content available to the network, however it wants to avoid overload of its link and resources. For that purpose the seed node makes a deal with the distribution nodes. Distribution nodes agree to distribute the content of seed node.

Comment:

This scenario well describes situation of a non-popular site quickly becoming popular, for example by publishing link to that site in a popular communication channel (widely known as *slashdotting* or *slashdot effect*).

Solution 1 Scenario:

1) Users pre-configure or discover the list of distribution nodes into their system.
2) When requesting any WWW content, users first try to ask one of the close distribution nodes. If the node responds with error, user tries to get content directly.
3) Distribution node receives a request from user, it will use internal distribution algorithm to locate a closest copy of content in the distribution network. If such copy is not available, it will download it from seed node and create the first copy itself.
4) Distribution node responds to the user with a copy of the content.

Solution 1 Problems:

- Client software needs to be modified to get a list of distribution nodes, to be able to decide which of the nodes is a close to the user and to try to request content from the distribution node before regular WWW request is attempted.
- HTTPS-based security is difficult to implement. User will need to trust distribution nodes to provide any content. However such relationship will not map to the realspace agreements, as the user has no contract with distribution nodes. It is the seed node that opts for content distribution.
- If there would more (competing) sets of distribution nodes it will be very difficult to maintain the lists of distribution nodes.
- The distribution nodes can trivially violate user's privacy as it will see all user's requests, regardless whether they distribute the content or not.

Solution 2 Scenario:

1) Seed site publishes the URI of the content in a form that forces users to use distribution nodes. For example it may use a form similar to `http://distribution.node.com/http%3A%2F%2Fseed.node.com%2FresourceName`.
2) User will use standard web browser to access that URI. It will result to HTTP GET operation on arbitrary distribution node.
3) Distribution node receives a request from user. It will use user's IP address to decide what is the best distribution node in the network for the user and use HTTP redirect to forward user's request to that node.
4) Distribution node with a better location receives a request from user, it will use internal distribution algorithm to locate a closest copy of content in the distribution network. If such copy is not available, it will download it from seed node and create the first copy itself.
5) Distribution node responds to the user with a copy of the content.

Solution 2 Problems:

- The content distribution will be fixed to a single distribution network. Once the URI is created in step 1, it cannot be changed to point to different distribution network. Change of the URI will invalidate all previous links.
- HTTPS-based security is difficult to implement. User will need to trust distribution nodes to provide any content. However such relationship will not map to the realspace agreements, as the user has no contract with distribution nodes. It is the seed node that opts for content distribution.
- Distribution node will be (incorrectly) considered to be the authoritative source of distributed data.
- Additional request is needed to redirect user to a better node, which means lower efficiency of the system.
- If the second distribution node happens to be inaccessible to the user, the user will end up in redirection “black hole” in step 4, having no way how to mitigate the situation. Even if the availability of the node is checked in step 3 some race conditions still remain.

Solution 3 Scenario:

1) Seed node delegates DNS resolution of its DNS domain to the system of distribution nodes.
2) User will use standard web browser to access the URI of a resource on seed node. First step of the process is a DNS resolution of seed node hostname. As the seed node delegates its DNS domain to the DNS server of distribution nodes, the request will end up on a distribution node DNS resolver.
3) DNS resolver of distribution nodes will use request source address to choose a good distribution node for the client. It will respond with IP address of the selected node.
4) User will use the resolved IP address as a target of HTTP GET operation on selected distribution node.

5) Distribution node receives a request from user. It will determine the resource name from the request. It will use internal distribution algorithm to locate a closest copy of content in the distribution network. If such copy is not available, it will download it from seed node and create the first copy itself.

6) Distribution node responds to the user with a copy of the content.

Solution 3 Problems:

- Seed node needs to delegate entire DNS domain to distribution nodes. Solutions that delegate only a subdomain are possible, however in this case it will be necessary to decide beforehand which resources will be distributed and which will be centralized, limiting the possibilities.
- DNS resolutions are frequently proxied and cached. It is questionable whether the IP address of DNS request will be a good approximation of actual user location.
- HTTPS-based security is not ideal for this situation. Seed node will need to trust distribution nodes with its HTTPS certificate and associated private key. That private key needs to be distributed on all distribution nodes, increasing the risk of private key exposure.

5.4.6 URI of the Sun Problem

Problem outline:

What is the URI of Sun, the star Sol of the Solar System?

Problem details:

According to the World Wide Web Architecture [58] anything that can be identified by URI can be a resource. URI definitely can be assigned an URI, for example `http://www.nlight.eu/sun`. The problem is that Sun may be identified by many URIs, each of that providing representations that may vary considerably. For example the information provided by Wikipedia will describe popular scientific information about the Sun as a start. However the page of religious community could describe Sun as deity and the site that focus on astrology will describe the impact of sun on our future. But even without going to the extremes, there are many pages describing the star Sol and its properties. Who is the authoritative source of information about Sun, the star Sol of the Solar System? According to the World Wide Web Architecture [58] assigning different URIs to the same resource should be avoided. Therefore who will be the single authoritative source for the Sun?

The mindset of current World Wide Web architecture does not provide any clean answer to this question. The core of the problem is that there is no authoritative source for a substantial part of the information that is expected to be available on the Internet. It is questionable whether such objective authoritative source exists at all.

5.4.7 Multi-Protocol Access Problem

Problem setup:

User has multi-protocol client software. This software can download the content using HTTP for later viewing, it can stream the content using more efficient

streaming protocol for immediate viewing or it can use HTTPS for secure download of sensitive content with when a performance penalty is acceptable. The user also has a hand-held device that can connect to the network only by a link with low throughput. That device does not have enough storage to store entire content, therefore common download technique is not possible. The device requires special-purpose optimized streaming protocol to operate.

Web server that hosts the resource (content object) can support all of the above protocols.

Problem outline:
What URI to use for the resource?

Problem details:
According to the World Wide Web Architecture [58] URI aliases should be avoided, therefore only a single URI should be assigned to a resource. However, different protocols usually mandate different URI formats. For example HTTP [56] protocol mandates `http` URI scheme while HTTPS protocols mandates `https` URI scheme for the same resource. The resource in this situation cannot be compliant both with WWW architecture and protocol specifications, therefore it is not possible to implement multi-protocol access to resources without violating the WWW architecture.

5.5 Conclusion

We have identified following major problems of the Internet architecture:

- IP addresses are used for identification purposes. Network addresses should be used only for addressing.
- Assumption that World Wide Web resources are static and therefore the World Wide Web interactions can be stateless. The nature of resources may be dynamic, therefore no such general assumption can be made.
- The definition of "World Wide Web unified interface" is weak. It defines only the very basic aspect to transfer data between nodes. However it does not define the minimal capabilities of World Wide Web components. Claim that two or more components are compliant to the World Wide Web Architecture does not guarantee interoperability between such components.
- Vague meaning of the resource. It is not clear what a resource represents, therefore it is difficult to implement meaningful rating and reputation mechanisms.
- URIs are used both for identification and addressing. The location of the resource and its identifier are interdependent, inhibiting the effectiveness of massive replication and data migration systems. Most identifiers depend on HTTP which depends on DNS which may be an obstacle to identifier persistence.
- There are multiple identifier formats where URIs should be used instead: QNames, Internet media types, domain names in digital certificates, etc. The

generic mapping of these identifiers to URIs is not provided, as there is no support for such mapping in URI syntax.

- WWW Architecture depends on HTTP, which ruins the protocol independence of World Wide Web. Protocol independence is important to support innovation and future development of World Wide Web.
- Security mechanism depends on the location of the resource, not on its origin.

The simplified architectural diagram of World Wide Web architecture is provided in figure Figure 14. It does show only the most essentials specification needed for understanding the architecture. For example the XML-related specifications are not shown. The diagram clearly illustrates that the World Wide Web architecture depends on HTTP and therefore in turn on the Internet protocols. This may introduce fragility to the system if the HTTP or Internet protocol specifications will need to be changed.

We have shown in this chapter that the architecture of the Internet technologies and the World Wide Web in particular are far from ideal. The evolutionary nature of both basic Internet principles and World Wide Web architecture is a good tool to guarantee adaptability and usability of the technology. However it fails to address some fundamental changes in the environment and user needs, rendering an ineffective and inconsistent system, although still a usable one.

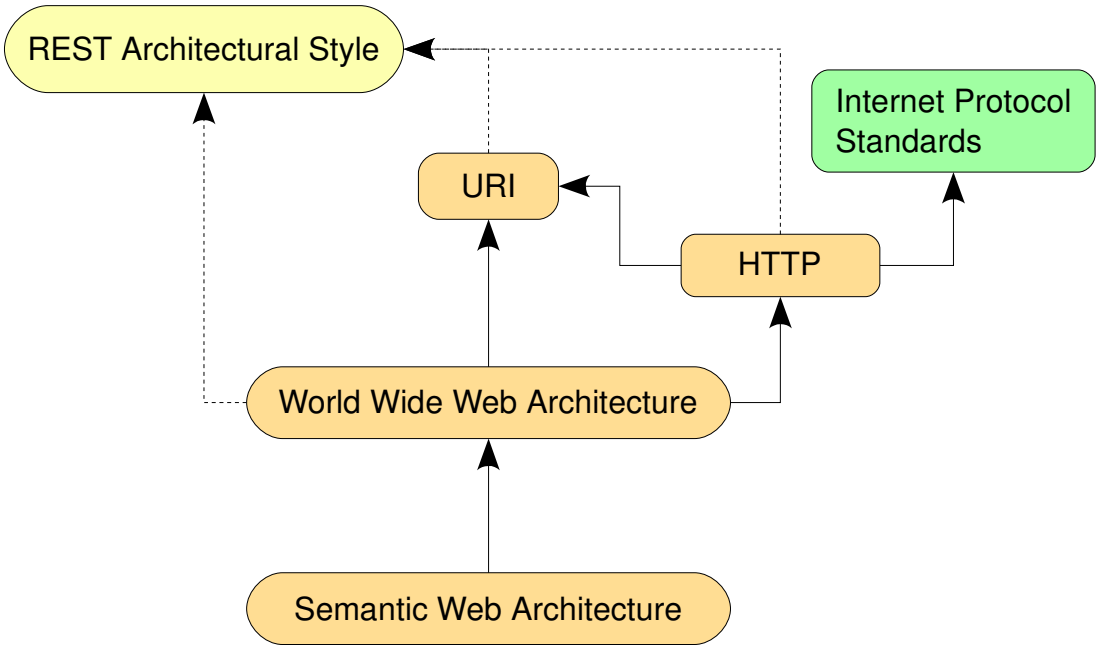

Figure 14: Current World Wide Web architecture diagram

6 Proposed Architecture

The current architecture of the World Wide Web is far from perfect. The World Wide Web architecture is supposed to be guided by the REST architectural style [57], but the style is only partially followed in practice. The REST style itself, especially the constraint of statelessness, has a doubtful application for a current World Wide Web. The problems of current World Wide Web architecture were described in detail in previous chapter.

This chapter introduces new architectural style called RRSS. It is heavily inspired by REST, building on the same essential ideas. However, the ideas are divided to finer-grain parts. New concepts are introduced in the RRSS style, based on the implications of the model described in chapter 3.

Later in this chapter we describe how the RRSS architectural style can be combined with other constraints to get a style applicable to the architecture of World Wide Web while avoiding some of the drawbacks of the REST style and current World Wide Web architecture that were described in chapter 5. Revised World Wide Web architecture is described based on these principles. A layered model of architectural abstraction is introduced to emphasize system stability in a case of changed environment or requirements. We conclude the chapter by proposing specific changes to current World Wide Web architecture that, as we believe, may help to build an environment of responsibility as described in section 4.1.

The proposed architecture is a theoretical work. We do not claim that the deployment of the proposed architecture is feasible or practical. The architecture should rather be used to guide the development of protocol specification and to help make the correct design decisions. The goal is to use the architecture to distinguish systemic solution from a short-term fix.

6.1 RRSS Architectural Style

A RRSS architectural style is described in this section. The style defines basic constrains of structuring and representing information in cyberspace, while taking into consideration that the origins of these information may be in realspace and also that it may be used by realspace entities. The RRSS is a conceptual architectural style. Direct applicability of RRSS architectural style to build applications is not the goal. It is not expected that applications will be build just by applying the RRSS style. It is rather expected that RRSS will be combined with other architectural styles. The resulting hybrid architectural styles are expected to be used to guide the design of practical applications.

The RRSS architectural style defines four basic elements: Resource, Representation, Source and Semantics. It also defines the relations and constrains for those elements. The RRSS architectural style derives directly from the empty (NULL) style. Following paragraphs define these elements and their place in the style.

6.1.1 Resource

Similarly to the authors of REST architectural style and the World Wide Web Architecture we find it is difficult to explicitly define a resource, as it can represent many different types objects and concepts. However unlike the authors of WWW architecture we want to be more specific about the meaning of the resource. Therefore we will define the resource indirectly by defining the properties of the resource. These properties are defined and explained in following paragraphs.

Resource has to be accessed by cyberspace components that make it available for cyberspace clients. Therefore we define resource as a cyberspace entity:

> **Resource** is a cyberspace entity.

While it is possible to define resource in such a way that it could be realspace or cyberspace entity, we are deliberately ruling out the realspace entities to simplify the model. Definition of resource as cyberspace entity can keep the part of the model that derives resource representation from resource and vice versa entirely in cyberspace. Well-know software modeling methods can be applied to such interactions. The limitation for a resource to be cyberspace entity is not considerably constraining the model, as the mapping between cyberspace and realspace entities can be applied at the conceptual level, as defined below.

> **Resource** can represent both realspace and cyberspace object or concept.

The resource is a cyberspace representation of cyberspace or realspace entity. Resource can represent a realspace object (e.g. the city of Bratislava), specific realspace concept (e.g. apple or house), abstract realspace concept (e.g. love or evil) or any cyberspace entity (e.g. file, directory, database record, etc.).

The resource should be self-contained representation of an entity. The state changes of the resource should be internally consistent. The cohesion of components inside resource should be high, while coupling between resources should be low. We define these requirements in a loose manner, as it is highly implementation-specific:

> Resource is a **complete, self-contained and consistent** representation of object or concept.

Resources are not constant in time. The state and even some aspects of the nature of the resource may change in time. Some resources are changing frequently, others are changing rarely, but all of them are changing. Therefore the client must not blindly assume any specific behavior of the resource state:

> Resource state is **dynamic and volatile**. It cannot be assumed that a resource follows any specific state-transition model, unless it is explicitly constrained by additional information.

For example it cannot be assume that a resource is constant without examination of resource meta-data. The representation of the resource should not be cached unless it is known that the state-transition model and the consistency characteristics of the resource allow caching.

We consider any attempt to constraint a resource to any fixed state-transition model as useless and non-productive. We argue that the nature of the realspace and cyberspace entities that the resources should represent cannot be constrained in this way, therefore the resource must follow the same principle. Pretending that by a mere definition of such constraint will change the reality is futile. Doing so will only introduce architectural inconsistencies of the implementation with the model, as the implementers will be strongly motivated to reflect reality and only slightly motivated to maintain the architectural consistency.

However, if it is known to the maintainer of a resource that it is has some specific dynamics, then it can be expressed explicitly and the implementations may take advantage of such knowledge. For example if it is known to the maintainer of a resource representing a printed book that it is unlikely to ever change, he could express the information that the resource representation may be cached.

6.1.2 Representation

The resource could be used for a variety of purposes in many different applications. It is not known whether it is possible to design a single mechanism to represent any resource in a form suitable for all current and future applications. Therefore we explicitly support polymorphism by allowing broad range of resource representations.

> Resource can be represented in the cyberspace by any number of **resource representations**.

Resource representation is a view of a resource that is limited by data format, language, target device properties, audience restrictions, etc. It may represent only some aspects of the resource. There is no requirement that resource representations must be complete. However the requirement on the resource to be consistent still holds, therefore the use of representation to present or modify the resource must not break the consistency of the resource itself.

Resource representation is a cyberspace entity. If a resource representation is complete, self-contained and consistent, it may be considered a resource. This possibility is introducing a recursion in the model. Resources have representations which may themselves have representation and so on. However the recursion is consistent with the proposed architecture. It may even be a good fit to describe some realspace concepts that tend to be recursive. For example specific book may be translated to several languages or have several versions. The book as a generic concept can be seen as a resource, represented by the specific translations. But each translation is a resource by itself that can have representations in specific data formats, such as PDF.

As the resource is a dynamic entity, so is the representation of the resource. The user should not assume that the information that he can see can be seen by all other users. For example if a user refers to a resource representing “today's news” on Monday, the news will be most probably much different when see on Friday. While the referred resource is still the same, the representations differ in time. Another example is a user seeing a personal page of another user. If there is an established relationship between these users, the provided information may be quite rich. For example it may contain private telephone numbers and show user's on-line status. However, if such information

is seen by an anonymous user, the provided information may be very sparse, limited only to basic details. The resource is still the same, but the representations differ by context.

The provision to allow any number of representation may be a concern for interoperability, unless a small set of mandatory representation formats is defined. This mandatory set may change slowly over time as the system evolves, but still assure interoperability of most of the software components.

> Small number of **well-defined and stable resource representation formats** should be standardized and mandated. Resource should provide at least one representation in the standard format.

The broad optional set of representation format may provide means for special-purpose application, for experimentation and innovation, but without any guarantees of interoperability.

6.1.3 Source

The information that forms a resource representing realspace concept is crossing the realspace-cyberspace boundary and therefore it must be considered subjective. We propose to generalize that principle and apply it to all resources:

> Resource is always **subjective**.

This generalization would allow to hide the detail whether the resource source is in realspace or cyberspace from the users of the resource. By mandating the subjectivity of resource the realspace-cyberspace distinction will no longer be important. This implies slight complication for resources that represent pure cyberspace entities, as they might not be considered subjective under some circumstances. However we believe that most useful cyberspace information are derived from realspace sources, and they are therefore indirectly subjective as well.

Subjectivity is an essential concept of the proposed architecture. By emphasizing subjectivity of resources, we induce an approach such as the resource will always be considered an opinion, rather than a truthful data. This is well aligned with the meaning of most information available on the World Wide Web. The information available from the resource identified by `http://www.whitehouse.gov/` should be considered an opinion of the resource owner (U.S. Government) about The White House. The opinion about the same object created by other bodies or nations may dramatically vary. None of these information should be considered an absolute truth just because they are accessible on the World Wide Web. All the information that are provided by World Wide Web should be considered subjective with respect to their sources.

As the resource is always subjective, a source of the resource must be considered to evaluate the qualities of the resource:

> The identification of a **resource source** is an integral part of the resource.

No resource should be used without considering its source. We expect that the relevance, meaningfulness, truthfulness and other qualities of the resource will be evaluated by the users of that resource. The users may express their opinion about the qualities of the resource, which also means that they are expressing the opinion about

the source of the resource. For example if an author of a document provides helpful and truthful information, the readers of such document may express a good opinion about the document and the author. The author's reputation would grow, expressing the expectation of the public that also other documents from the same author will have good qualities. Therefore it is expected that the reputation of the resource source will influence the evaluation of the resource as well as the results of the evaluation may influence the reputation.

The specific algorithm for evaluation of the resources and maintaining reputation is outside of the scope of this document. However we argue that the binding between resource and resource source is essential for developing such mechanism. Most of the resources are not representing persons, therefore the concept of reputation does not directly apply to them. As discussed in section 4.3 the reputation is only meaningful if it can influence future behavior. Long-living personas and a relatively large amount of data about the persona behavior is necessary for efficient reputation system [50]. Resource source may be that long-living persona and resources produced by that source may for the data for reputation evaluation.

According to the model introduced in chapter 3, resource source is a persona. It is not required that the resource source is a persona that represents a realspace person. Resource source may be a persona representing a computer system, for example if the resource represents results of automatic summation of database statistics. However, we expect that many resource sources will represent realspace persons, as resources produced by such sources will be most meaningful and useful for the users. The resource source persona may not represent the realspace identity of the person. It may be a pseudonym or a persona that only partially reveals realspace information about the person. Such approach may allow to dynamically tune the trade-off between privacy and revealing of information for the purpose of inducing trust in the consumers of the information.

6.1.4 Semantics

The meaning of the resource may not be interpreted properly having just the resource representation. For example user that sees a picture on his screen cannot be sure whether the resource represents the picture or the object shown on the picture. Therefore an unambiguous semantic description of the resource is needed for a clear understanding of the resource meaning:

> Resource can be semantically described in a standardized computer-readable form. The **semantic description** is mandatory for all resources and it must be a part of any interface that is used to access the resource.

The ability to express the meaning of the resource is important for orderly operation of World Wide Web, as explained in sections 4.2 and 5.2.2. This problem is made worse by the sensitive nature and immaturity of the legislation concerning cyberspace. For example if someone puts a statement saying “John Doe is an idiot” on his company homepage, that will be probably seen by the court as defamation. However if someone asks a private question “What do you think about John Doe” and the response would be “I think he is an idiot”, that may be seen as expressing a private opinion that should be protected by the freedom of speech. However, both interactions are executed in the same

fashion on the World Wide Web, usually using HTTP GET request. Therefore it would be difficult to distinguish if the user asked to view company home page or asked for an opinion about John Doe. This problem could be at least partially addressed by mandating to provide semantic description of any resource, so the user can be aware of the context, not just the pure information content of resource representations.

The basic concepts guiding the use of resources in cyberspace should form a foundation for any of its implementation. Therefore these concepts should not depend on any implementation-specific detail. For example the resources and representations should not depend on any specific communication mechanisms, protocol or interface:

> The concepts of resource, resource representations, resource source and resource semantic description must be implementation-independent.

The implementation mechanisms should be build on top of the basic concepts specified above. The implementation should depend on the basic concepts, not vice versa. The implementation must provide all the basic concepts and keep the proper relations between them. This approach will allow to modify or replace the implementation without changing the basic principles. It is relatively easy to adapt applications to new communication mechanism, while it is very difficult to adapt applications when the basic operational principles change.

6.1.5 RRSS and REST

The RRSS architectural style is inspired by Representational State Transfer (REST) architectural style [57]. The purpose of RRSS and REST to explain and guide the architecture of World Wide Web is similar. RRSS builds on the concepts of resource and resource representation that are similar to those defined by REST. However, RRSS is an attempt to create a “pure” architectural style that is derived directly from the empty (NULL) style, while REST is a derived style. The motivation behind RRSS was to put the basic principles in the center and build the application of these principles by combing the RRSS with other architectural constraint to get an specific architectural style suitable for application in the cyberspace.

The REST architectural style is focused on distributed hypermedia systems. However the RRSS style is not limited to hypermedia. It is rather aimed as a generic model for building applications, where hypermedia is one of the applications. The support for distributed systems was in the author's mind while creating RRSS, it is not a strict requirement and we believe that the RRSS style is efficiently applicable to non-distributed systems as well.

Following sections describe how the RRSS architectural style can be combined with other constraints to get a style applicable to the architecture of World Wide Web while avoiding some of the drawbacks of the REST style and current World Wide Web architecture that were described in section 5.2.

6.2 Resource Identification Constraints

The RRSS architectural style does not contain any constraint concerning resource identification. This omission is deliberate, as we consider resource identification as non-essential part of the RRSS architectural style. Any implementation of the style may

provide their own means for distinguishing the resource, while some implementation may use hidden or implicit identification mechanisms (such as memory pointers). Therefore we will describe the aspects of resource identification as a separate architectural constraint that can be optionally applied to the RRSS architectural style.

The resource identification constraint is defined as follows:

> Each resource has assigned an **identifier** that can uniquely and consistently identify the resource within the whole system.

The definition of the resource identifier keeps in mind the potential use of the system in global-scale hypermedia applications. It is expected that the resource identifiers will be passed from application to application, cached or even remembered for a long time (bookmarked) and especially used as a hypermedia references. We expect that there will be no mechanism that will allow to force invalidation or updates of such stored identifiers. Therefore the requirement for unique and consistent identification is critical to address the needs of such applications.

This uniqueness requirement introduces globalism to the identification mechanism. Each identifier must be unique within the whole system. Therefore for systems that are expected to span the globe, such as World Wide Web, the requirement essentially mandates globally-unique identifiers.

The consistency requirement limits the assignment and use of the identifier. One identifier should not be assigned to two or more resources. The identifier assigned to a resource should be assigned to that resource indefinitely. However, more than one identifier may be assigned to a resource, allowing resource aliases.

The knowledge of the identifier does not imply the existence of a resource or accessibility of any representation of the resource it identifies. The resource that is identified may not yet exist, it may no longer exists, access to the resource may denied due to security constraints, it may be inaccessible due to network or identifier resolution problems, the client may not have appropriate software components to access the resource, etc. The ability to use the identifier for access to the resource does not affect validity of the identifier itself, it just affect its usability.

The implementations should not assume or assert anything based solely on the identifier. For example an implementation having an identifier should not assert that the resource that is identifier does not exist just because it cannot be accessed. The client should not express any opinion about the identifier itself, it should always express opinions about the resource and for that purpose the client needs to access the representations of the resource.

The resource identifier is meant to provide a way how to distinguish one resource in the haystack of many other resources. The goal of the identifier is to allow such ability to last very long and to be reliable. However the identifier is not equivalent with the resource and it even cannot be considered a representation of the resource.

For the purpose of reliable operation of the World Wide Web the identifiers must unambiguously identify the resources. This requirement is partially expressed by the Resource Identification constraint defined in previous sections, mandating that the identifier uniquely and consistently identifies the resource. This can be further specified in a form of more concrete constraints:

> Resource Identifier must identify at most one resource.

By requiring that resource identifier can identify no more than one resource we satisfy the constraint of uniqueness. Application that has a value of the resource identifier may reliably use it to distinguish one specific resource among other resources. The resource identifier may also identify no resource at all. The situation may happen if the resource does not yet exist or does not exist any longer. It is unrealistic to assume strict consistency requirements between different network nodes, therefore it would not be practical to prohibit the existence of identifier that does not identify a resource.

It is expected that the identifiers will be distributed thought the system with no explicit limitations. The identifiers may be used in the hypermedia documents to refer to the resources. The identifiers may be cached for a short time by the clients or stored for a long-term usage (e.g. bookmarking). There is no explicit mechanism how to assure consistency of the identifier between the source of the identifier and all the components that use it. Therefore we need to define following constraint to satisfy the requirement of consistent identification:

> Resource identifier that was assigned to a resource cannot be assigned to a different resource.

While this constraint may seem redundant, it is not so. Considering only the constraint of uniqueness above, an identifier that was used for one resource might be assigned to another resource after the end of life of the original resource. This action may cause problems for applications that assume that the identifier still refers to the original resource, leading to misbehavior of applications.

As the resource identifier may be stored for a long time on uncontrolled number of locations in the network, it is important to keep the period of validity of the identifier as long as possible. In the ideal situation it should the same or longer than the lifetime of the resource it identifies. However this may not hold if the resource identifier is changed (resource is “renamed”).

It cannot be assumed that any identifier will be resolvable and that it will stay resolvable indefinitely. Broken links are difficult to avoid in heavily distributed environments, such as World Wide Web. The question whether a strong consistency mechanism is feasible in such an environment is still open. Therefore our goal is to provide only a weak consistency guarantees: avoiding pointing the client to a different resource than originally intended, but not addressing resolution errors and broken links.

Resource identifier should not depend on the location of the resource or its representation. We expect that the location of the resource or its representations may change. Especially considering mechanisms such as dynamic on-demand replication in peer-to-peer networks. Any identifier scheme that reflects location of the resource would fail in environments where the location is rapidly changing. Based on this thinking we can formulate a more generic constraint for identifier schemes:

> Resource identifier scheme must not depend on any structure or concept that underlies the implementation.

Networking concepts such as IP address or DNS name are part of the World Wide Web implementation and not part of the architecture. We consider the use of network concepts in the resource identifier to be an abstraction that leaks the details of

underlying implementation beyond reasonable limits. The identifier must not depend on the IP address, it must not assume that the Internet is organized in sites, etc. Networking structures tend to change frequently, especially considering the lack of any strict architectural rules for the Internet (see chapter 5). Current hierarchical addressing and naming mechanism used for the Internet may change in the future, for example by introduction of flat namespace used by peer-to-peer networks. Resource identifier is meant to be identification mechanism, not addressing mechanism. The address (location) of the resource should be obtained indirectly, by resolving the identifier to an address.

However, the identifier scheme may assume the natural organization of realspace (e.g. companies, countries, etc.) and the natural organization of information, for example tendency to group information to folders or collections.

Even if the semantics of the identifier is well-described in the architecture, this semantics should not be visible to the client applications.

The semantics of the identifier must be opaque to the client applications.

Client applications should regard identifiers as opaques string or may understand only the very basic syntactic rules for the identifiers. The understanding of syntactic rules may be needed for combining several identifiers, e.g. in the case of URI to QName mapping in current World Wide Web (see section 5.2.3). Except for manipulation of the basic syntactic understanding the applications should not infer any other properties of the objects identified by the identifiers. For example the applications must not assume that the identifiers are related if they share a common prefix.

6.3 Revised World Wide Web Architecture

A proposal for revised and improved World Wide Web architecture is described in this section. The new architecture is based both on the architectural styles described by Fielding [57] and on the styles proposed in earlier sections. Proposed architecture is not created on the green field. It takes into consideration the existence of current World Wide Web and follows the basic ideas of the Web such as the goal to create global information space. It does attempts to judge if these ideas are good or bad, it just tries to improve the problems identified in the section 5.2. It is also not a goal to radically change the existing architecture or propose an ideal architecture that will last forever. The proposal should be understood as a next step in the evolution of the World Wide Web that we recommend to take. We also do not claim that the proposal is complete and can be immediately implemented. Some architectural elements are simply identified as missing (such as correct URI format or resource semantic description language). This work specifies only the desired properties of such elements and further work is required to design their specific definitions.

We propose to split the overall World Wide Web architecture to several levels of abstraction. The split may improve the understanding and visibility to the architectural concepts. Proper layering of the abstraction can also address different goals of dynamics and interoperability properties of the architecture, as explained below. We propose following four levels of abstraction (Figure 15):

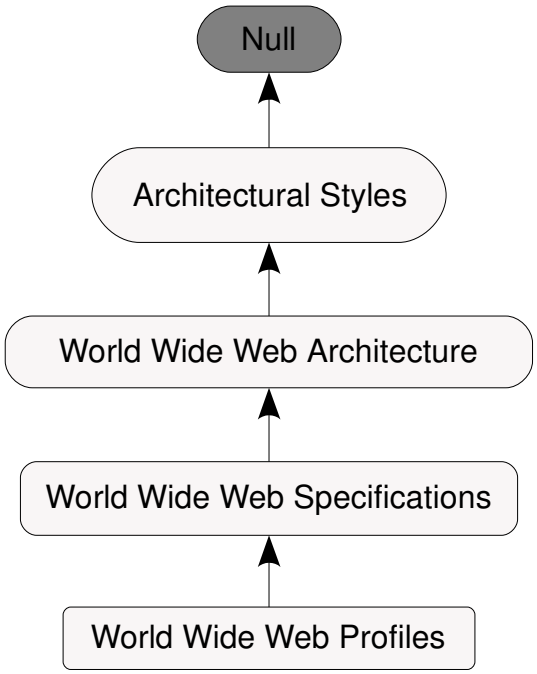

Figure 15: Abstraction layers of proposed WWW architecture

- **Architectural Styles** are the most abstract concepts. These form a set of architectural constrains that guide the creation of systems with appropriate properties and qualities. The architectural styles are not specific to the World Wide Web. The styles are rather generic and applicable to a wide range of applications. Architectural styles are used as a foundation and "best practice" to guide creation of WWW architecture. The architectural principles are considered extremely stable and the influence of changing world and requirements is considered negligible. It is expected that the architectural principles may be considerably changed only if an inconsistency is discovered or in the event of a major breakthrough in the state of the art in the field of computer science.
- **World Wide Web Architecture** is a set of architectural constraints, rules and recommendations that define basic principles of World Wide Web operation. These principles are considered fundamental and it is expected that they will be valid and applicable for a long time. Although the world is dynamically changing, the goal is to design these principles in such a way that they will change much less frequently. It is expected that the basic architectural principles may change only if an architectural inconsistency is discovered or the basic requirements for World Wide Web would dramatically change.
- **Protocol Specifications** provide specific definitions of communication protocols, data formats and interfaces. These specifications are based on basic principles, constraining them by specification of implementation details. It is expected that the protocol specification will be continually adapted to the implementation needs and that several protocols may exist at the same time for

the same purpose, with different characteristics. The protocol specification is a place for innovation and experimentation.

- **World Wide Web Profiles** define a set of protocols that are required for correct cooperation of all World Wide Web components. Profiles are mechanism for interoperability. Component that claims compliance with World Wide Web specification should comply to one of the interoperability profiles. For example World Wide Web Basic profile defines the minimum set of specification that a component must implement to be able to cooperate with other web components. Another example may be hypothetic World Wide Web Code on Demand Profile, which would be based on the Basic Profile and would incorporate the specification of virtual machine for downloadable code and other details of client-side code execution.

We believe that dividing the architecture to different levels of abstraction can provide well-controlled environment that still allows innovation and interoperability. The architectural styles are expected to provide the theoretical foundation. The World Wide Web Architecture should provide practical guidelines for protocol designers, thus maintaining consistency. The protocol specification layer should allow innovation, optimization and experimentation. While innovation is desired in essence, uncontrolled innovation may lead to non-compatible extensions that may limit the network effect of the World Wide Web. Therefore a layer of Profiles is proposed to define the interoperability constraints and requirements for different classes of applications.

6.3.1 Architectural Styles Layer

Architectural styles provide abstract guidelines for design of system architecture. The architectural styles constraint the system being designed to induce desired properties of the system. Following architectural styles are being considered to become a foundation of World Wide Web Architecture:

- RRSS architectural style forms a basic model of dealing with data. It defines the concepts of resource, resource representation, resource source and semantics. It constrains a generic data model to a limited set of concepts, therefore supporting simplicity and uniformity of the system.
- Resource Identification constraint is incorporated to allow referencing of resources in a globally distributed system. This style constrains the interface between nodes in the distributed system as well as data formats used for hypermedia applications. It supports simplicity and visibility.
- Layered Client Server is a composition of Layered System and Client-Server styles. This style constrains system to a hierarchical environment with clearly defined roles of client and server. The system can be composed in layers, dividing the internal parts of the system from external. This style supports scalability and evolvability.
- Cache is a derivation of Replicated Repository style. Cache allows to temporarily store the results of interactions and to use that stored results to improve the efficiency of future interactions. The cache is not a mandatory component of the architecture and therefore it cannot be considered a “pure constraint”. However the concept of potential existence of cache is needed in

the architecture to allow generic caching mechanism even if it is optional. This is reflected especially in the concept of expressing the dynamics (e.g. *cacheability*) of resource representations. Cache supports efficiency and scalability.

- Uniform Interface style constrains all components of the system to the same interface. This style is quite generic, as there may be many interfaces between the components of the system and the components may exist on different levels of system decomposition. In fact each architectural style somehow limits the system to a (more or less) uniform interface. Therefore for the purpose of this document we will understand the Uniform Interface style as a constraint that any and all component interactions must follow the same interface, while considering only the components on the same level of abstraction and system decomposition. The Uniform Interface style supports simplicity and visibility.

Current WWW architecture is guided by the REST architectural style. While this work is inspired by REST, it has identified some mismatches of REST constraints and the desired architecture of the World Wide Web. Following architectural styles are incorporated into REST but are not incorporated into our proposal for World Wide Web Architecture:

- Code on Demand and Virtual Machine styles are not incorporated to the World Wide Web architecture. The constraints imposed by these styles are not mandatory for WWW architecture. In other words the WWW architecture does not require that a Virtual Machine or Code or Demand environment must be a part of any World Wide Web system. Therefore these styles do not constraint the WWW architecture and can be left out. We propose to apply these constraints to appropriate World Wide Web profile rather than make it a core of the WWW Architecture.
- Stateless Client-Server mandates client-server operations that are unconditionally independent. As explained in section 5.2, this requirement cannot be universally satisfied in the World Wide Web. Therefore the architectural constraints of Stateless Client-Server style cannot be applied.

6.3.2 World Wide Web Architecture Layer

The World Wide Web Architecture is a composition of the architectural styles from the Architectural Styles layer supplemented by more specific architectural constraints. The goal of World Wide Web Architecture is to provide guidelines for developers of specifications that govern the basic operation of World Wide Web. It also specifies fundamental concepts of the World Wide Web, such as Unified Resource Identifier (URI).

Figure 16 illustrates the composition of World Wide Web architecture as well as the abstraction layers that derive from it. This figure follows the basic principles for conceptual diagrams as described in chapter 4. The colors used in the figure have following meaning:

- Blue: Generic architectural styles according to Fielding [57].
- Yellow: Architectural styles defined in this document.

- Green: Existing standards and specifications not directly related to World Wide Web.
- Orange: Standards, specifications and guidelines related to the World Wide Web.

Individual components of proposed World Wide Web architecture are described in details in following subsections.

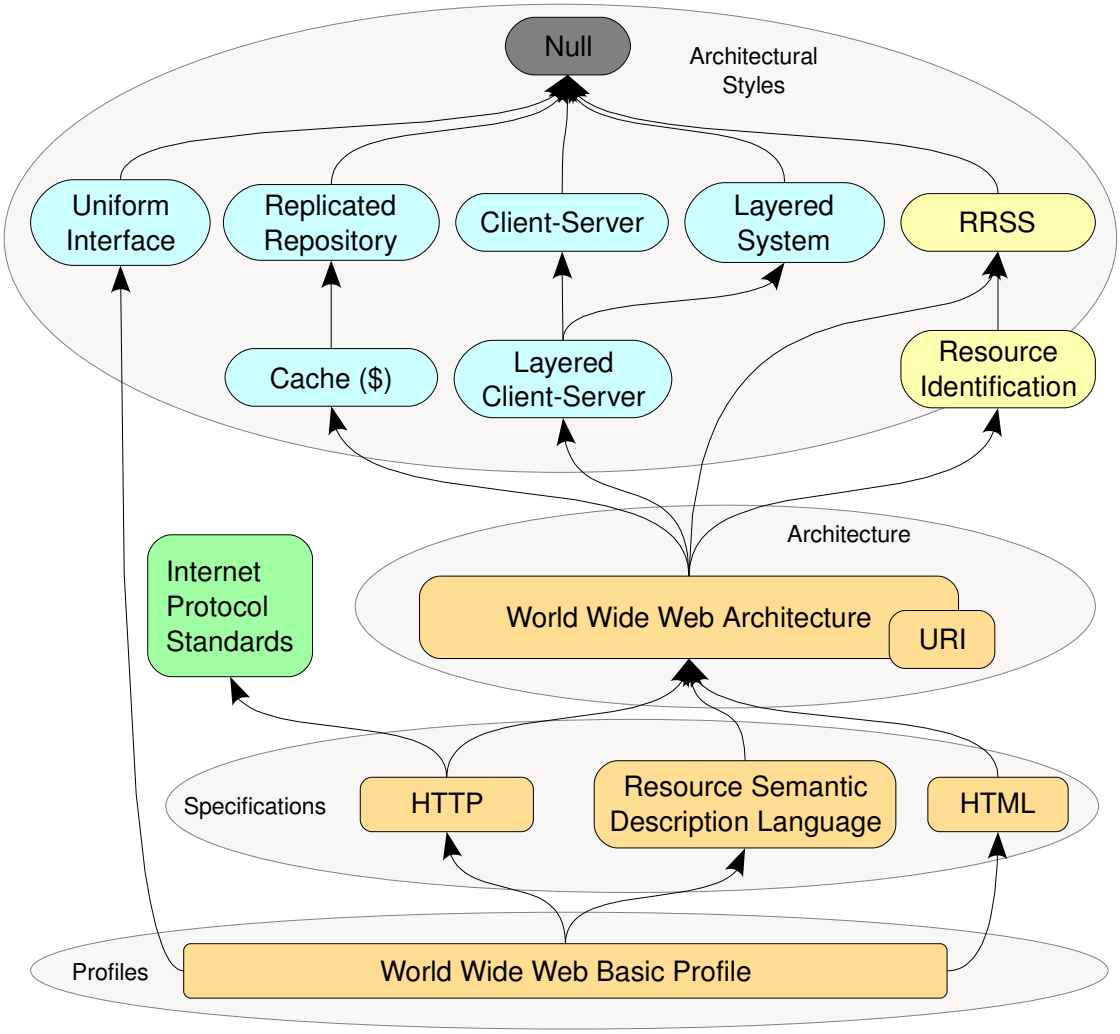

Figure 16: Proposed World Wide Web architecture diagram

6.3.2.1 Uniform Resource Identifier

The constraint that the resources must be uniquely and consistently identified is a one of the fundamentals of World Wide Web simple hyperlinking scheme. One document (resource representation) may refer to another resource using the identifier. We define Uniform Resource Identifier concept to fulfill such role:

> Resources are identified by identifiers with fixed syntactic rules, called **Uniform Resource Identifier** (URI).

The current state of URI definition is far from ideal, as described in section 5.2.1. To amend that situation, we propose that the complete definition of Uniform Resource

Identifier should be integral part of the World Wide Web Architecture for the following reasons:

- The URIs are used by numerous formats and protocols (e.g. HTML, XML, HTTP, RTSP, etc). The change of URI concept may influence the design of variety of components of the World Wide Web.
- The URIs are stored in numerous places. They are stored as anchors in HTML, namespaces in XML, stored in bookmark databases, pre-configured in software applications, etc. Change or the concept of URI may influence any and all of them.
- The concept of URI should stay valid even if the World Wide Web profiles change rapidly. For example if HTTP is replaced by other protocol, the migration path should include just the change of protocol, not the change of identification scheme. The identifiers used to express relations of resources and representations and the identifiers stored in databases should not be influenced by the change of protocol.
- The Uniform Resource Identifier (which is supposed to be an identifier) and the Uniform Resource Locator (which is supposed to be an address) should be easily distinguishable just by analyzing the identifier syntax, without requiring an attempt to resolve the identifier.

The complete syntax and semantics of Uniform Resource Identifiers should be part of the World Wide Web Architecture. The use of existing naming schemes for URIs should be deprecated and a single well-defined naming scheme should be defined. The naming scheme should conform to the Resource Identification architectural constraints defined in section 6.2. All entities conforming to current URI syntax and not following the new URI scheme should be considered Uniform Resource Locators instead.

We believe that such approach will provide numerous benefits for the architecture: By mandating fixed syntax and semantics for URI we allow the protocol-independence of World Wide Web Architecture. The architecture does not need to depend on HTTP any more, quite the contrary. The HTTP protocol would depend on the specific syntax and semantics of URI. This allows that the protocols like HTTP can evolve without the risk of affecting the World Wide Web Architecture with HTTP details. If the concept of URI would be defined in sufficient details at the level of World Wide Web Architecture, the protocol and data format specification may depend only on the World Wide Web Architecture and they need not to depend on each other. The protocol specifications are also not forced to define their own addressing scheme. The current situation is that new protocol specification needs to define their own URI naming scheme or they need to depend on a protocol that defines a naming scheme (e.g. HTTP).

Although the definition of specific syntax and semantics of URI naming scheme is out of scope of this work, in the following paragraphs we specify several suggestions that the suitable scheme should follow.

- The URI scheme should follow unified approach for all of its components. The drawbacks similar to the inconsistencies of HTTP URI syntax should be avoided (see section 5.2.5).
- The URI scheme should be fully hierarchical and should allow delegation of control at any level. There should be no explicit distinction of authority section

and local (path) section. Such details should be resolved at the time of identifier dereference. Keeping this distinction hidden will allow better flexibility of adapting the details of underlying implementation (e.g. network infrastructure) without affecting the validity of existing identifiers.

- The URI scheme should consider a method for incorporating flat namespaces, such as those used by system based on Distributed Hast Table [83] mechanisms (e.g. some peer-to-peer systems). For example flat namespace could be expressed as a shallow tree in the hierarchy.
- The URI scheme should support clearly defined rules for creating hierarchy and/or support embedded identifiers. For example the problem of URI-QName mapping (see section 5.2.5) may be solved by two methods:
  - Define a rule that the local part of the QName should be (hierarchically) appended to the namespace URI. For example a local value `foo` in namespace `/eu/nlight` would be mapped to URI `/eu/nlight/foo`.
  - Allow the expression of "URI within a URI", so URI could be used as a hierarchical component of any other URI. For example is a parenthesis characters would be used to denote the embedding, the local value `foo` in namespace `/eu/nlight` would be mapped to URI `/(/eu/nlight)/foo`.
- The URI syntax must make it easy to distinguish between URI and URL just by syntactically analyzing the identifier. This mechanism will be needed to avoid confusion and to avoid using URI instead of URL or vice versa (by mistake or ignorance). The requirement may appear to be in conflict with the constraint that client application should not understand semantics of URI. However, the URLs are not URIs and therefore it is possible to constraint the URI syntax and semantics in such a way that URIs and URLs are distinguishable at syntactic or semantic level.

There is no constraint that would limit the number of identifiers assigned to a resource. Therefore different identifiers identifying the same resource are possible. Such identifiers are sometime called aliases. The existence of aliases makes it impossible to evaluate the equality of resources by comparing their identifiers. This situation may be confusing for applications, as applications may refer to the same resource without knowing that it is in fact the same resource. To address this problem we introduce the concept of canonical identifier:

> Each resource must be assigned exactly one **canonical Uniform Resource Identifier** at any specific time.

Canonical URI can be used as normal URI to refer to the resource. There can be only one canonical identifier per resource and each identifier may identify only a single resource. Therefore it can be easily evaluated whether two identifiers refer to the same resource by resolving the identifiers to canonical identifiers and comparing resulting canonical identifiers.

Although it is not explicitly stated in the definition of canonical resource, it is recommendable that the lifetime of canonical identifiers should be as long as possible.

Such approach will make it easier to evaluate equality of resources in highly distributed environment such as World Wide Web. The canonical identifiers should not be directly exposed to end users, e.g. it should not be displayed in the address bar of the web browser. Therefore the motivation to include human-readable strings in the canonical identifiers should be minimized. The canonical identifiers should rather contain only persistently strings that for an identifier that could be persistently assigned to the resource.

Resource identifiers are translated to resource representation addresses and canonical identifiers during a resource identifier resolution process. This process is considered protocol specific and should be defined at the protocol specification architectural layer.

6.3.2.2 Resources and Representations

Basic principles of proposed World Wide Web Architecture are based on the RRSS architectural style described in section 6.1 and combined with other architectural styles. The constraints of RRSS style are made more specific and directly applicable to World Wide Web Architecture. The focus of this section is on specification of behavior of resource representations, as the resources themselves stay largely behind the scenes, in the area of implementation. We specify the concept of resource presentation in a form that can be used in protocol and data format definitions:

> **Resource representation** is an array of bytes that form the content of the representation together with metadata that describe representation data format and may describe other aspects of the resource representation.

For example resource representation may be a byte stream of a HTML file, a definition that the data format is HTML and (optionally) other metadata. Such definition mandates that the resource representation type definition and other metadata are integral part of the representation and that they must always be evaluated together. This places a common constraint on any access protocol or other mechanism to transfer and store resource representations.

If the state transition model of the resource is constrained, it should be expressed in the representation metadata. For example if a resource is updated once per day, the metadata may indicate that the provided representation will not change until the end of the day. That may allow clients to cache the representation and re-use it within the limits defined by metadata.

Based on the client-server architectural style, we define that resource representation is a data unit transferred between client and server:

> Resource representations can be directly retrieved from server to clients by using **access protocol**.

There should be no single access protocol mandated by the World Wide Web Architecture. The specification of the protocol should depend on the World Wide Web Architecture and the use of such protocol should be mandated by the World Wide Web Profiles. However, the existence of an access protocol as an abstract concept is assumed by the World Wide Web Architecture and common constraints for all World Wide Web access protocols are defined here.

Resources are identified by URI. However the URI should provide no information about the location of the resource or its representations. We also place no constraints regarding location of resources and resource representations, not even the constraint that the representations of one resource should be placed together. Therefore we consider it useful to define a concept that will represent address of resource representations:

> Resource representations are addressed by **resource representation locator**.

Resource representation locator is not an identifier, it is an addressing mechanism. It is not expected that the locator will be persistently stored by the applications. Therefore the locator would not be required to stay valid for much more than few minutes. Locators may be protocol-specific, may contain addressing or routing details, etc. The purpose of the locator is to allow efficient location of a resource representation in the network, therefore we assume that locator syntax and semantics will be adapted specifically for this task.

The usual World Wide Web interaction should be executed as follows:

7) Resolving URI to representation locators. The resolution may yield many locators, both for many types of representations and for many access protocols.
8) Choose appropriate locator for representation that the application can understand and protocol that it supports.
9) Use the chosen locator and protocol to access chosen resource representation.

These steps may be combined into less than three steps, if access protocol supports it. But in that case the access protocol needs to be chosen before the interaction, which limits the possibilities. For example if HTTP will be chosen to retrieve a resource representation and it will execute all these steps in one step, the representation will be retrieved by HTTP even if a more efficient way exists (e.g. streaming).

6.3.2.3 Resource Semantic Description

The RRSS architectural style mandates semantic description of the resource. Our proposal of WWW constrains that and defines semantic description of a resource to be resource representation. Therefore it becomes a mandatory representation of a resource:

> **Semantic description** of resource is a mandatory representation of that resource.

There must be a clean distinction between resource metadata and representation metadata. Our solution is to make resource metadata a mandatory representation of the resource. By constraining semantic description of a resource to the limits imposed by resource representation we are simplifying the architecture. No additional special mechanism is needed to handle semantic descriptions. Ordinary access protocol used for resource representations can be used. This approach also allows the existence of several data formats for resource semantic description, for example to address innovation (migration from one format to another) and experimental usage. However we expect that the World Wide Web Profiles will mandate single data format for resource semantic descriptions.

The details of the elements of resource semantic description are out of scope of this document. We expect that some specifications developed for the purpose of the

Semantic Web could be reused, but it was not considered in detail for the purpose of this work.

6.3.2.4 Resource Source

Based on the model introduced in chapter 3, we can consider resource source to be a persona. As persona is a cyberspace entity and it can be presented using a set of cyberspace representation, we will constraint it the concept of a resource source to be a resource:

> **Resource source** is a resource.

By mandating that resource source is a resource we solve several problems:

- The resource sources can be identifier in the same way as resources, using URIs.
- The representation of resource source can be accessed using the same mechanism as the resource itself, supporting uniformity and simplicity of the system.
- The mandatory representation of resource source is a semantic description, specifying the nature of the source.

Resource sources, being resources, must have definition of their sources as well. This introduces recursiveness to the model, which may be used to the advantage. Many personas have their sources. For example the approach to a reputation management described in section 4.3 implies that resource sources (personas) will be maintained by identity agents. Therefore the personas of identity agents are the sources for the personas of end users. There is no limitation that a will prohibit a resource to be its own source, as long as it makes sense semantically. Therefore the persona represented as resource can be its own source. This case is a *self-asserted* persona, which is means that the data provided by the representations of such persona originated from the persona itself.

According to the constraints of RRSS architectural style the identification of resource source is an integral part of the resource. Therefore a proper place in the architecture is needed to express the relation between resource and resource source. We consider the semantic description of the resource as an ideal place for that, as it is a mandatory element as its purpose is well aligned with the purpose of resource source identification. Therefore we mandate:

> **Identification of resource source** should be mandatory element in the resource semantic description.

The resources and their sources may be linked by the source URIs in the resource semantic descriptions. An example of such approach is provided on Figure 17. The figure illustrates an example of system based on identity agents that is outlined in section 4.3. A photograph is represented as a resource *Photo A*. It has a JPEG representation that contain the image data and semantic representation which defines that this resource is a photograph and that the source of the resource is a resource identified by an URI of *Photographer X*. *Photographer X* is a resource that represents the persona of photographer that took photo A. Such persona may represent civil realspace personality of the photographer, containing his real name. It may represent a realspace personality of the photographer as he is known in the artistic circles,

containing his artistic pseudonym. It may even represent a cyberspace account of the photographer on some kind of social networking site, featuring just his nickname. In any case, the persona is maintained by the *Identity Agent Y*. The information that *Identity Agent Y* maintains the persona of *Photographer X* is expressed by using the URI of resource *Identity Agent Y* as an identifier of a resource source in the semantic description of resource *Photographer X*. The resource *Identity Agent Y* is again a persona that represents the organization that maintains personas. It may be a social networking site, identity provider, the organization employing the photographer, visual arts professional organization, etc. In this case such organization acts as independent organization, therefore the source of the *Identity Agent Y* resource is its own source. It is a self-asserted resource.

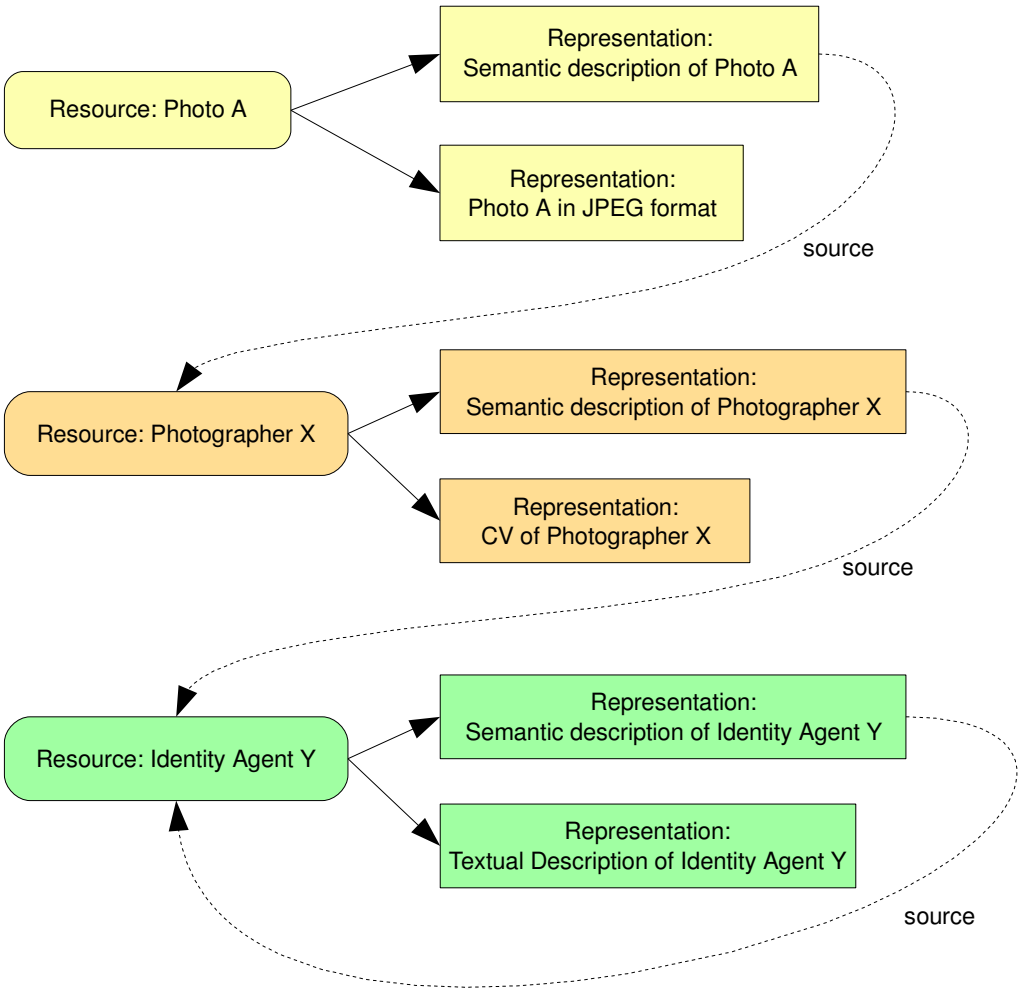

Figure 17: Resource source linking

Such a chain of sources may be used to augment security mechanisms of World Wide Web and to solve some of the problems described in section 5.2.6. We argue that the source of authority for the resource should originate in the resource source (persona), not the DNS domain of HTTP URL authority section. Therefore we propose that the certificate authorities of the World Wide Web should issue certificates to personas that represent resource sources. The certificate subjects may be identified by URIs, as the

personas that act as resource sources are resources by themselves. The certificates issued to resource source personas may be used to sign the content of semi-static resources. We envision that they may as well be used to delegate the right for creating of dynamic content to sites hosting the resources. However the specific security mechanism is not in the scope of this work.

The semantic description might also be optionally used to express a reputation of the resource. Such a method would only be applicable if the target of the reputation is a resource that represents persona and the source of the reputation is also the source of the persona resource. The reputation expressed in this way should be understood as the opinion about the qualities of the resource expressed by the entity that is identified as the source of the resource. This approach may be a simple way how to express reputation in strictly hierarchical persona structures. However, it is not yet clear whether such structures are feasible or practical. Therefore we expect that additional mechanism for expressing reputation will be needed.

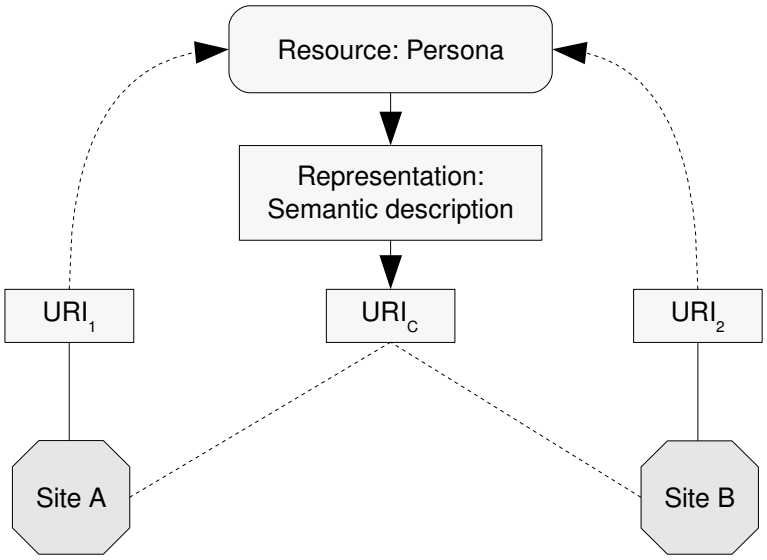

Figure 18: Direct linking to persona resource

The global nature of URIs may endanger privacy of personas, if used incorrectly. If a single global identifier for a persona is used, then all visible actions of the persona and all accessible resource published by the persona may be correlated and an attacker may gain more information that was consciously released by the controller of persona. A decent level of privacy may be maintained if pseudonyms are used instead of primary identifiers. A different pseudonym for the same persona may be used for each site. Therefore the amount of information available to the attacker is limited and the danger of exposing additional information is lower. A simple approach to address this problem would be to use different URIs that identity the same resource. However, such approach will be hindered by the mechanism for evaluating URI equivalence. Such URI pseudonyms can easily be correlated using canonical URI. Such situation is illustrated in Figure 18.

One possible solution is to create several resources that represent the persona. We can take advantage of the recursiveness of resource representation. We can model the original persona as a resource, while the pseudonyms will be representations of that resource. However, the pseudonyms are resources by themselves, therefore they have their own canonical URIs. The canonical URIs of the pseudonyms are different, therefore the pseudonyms are not trivially linkable. This approach is illustrated in Figure 19.

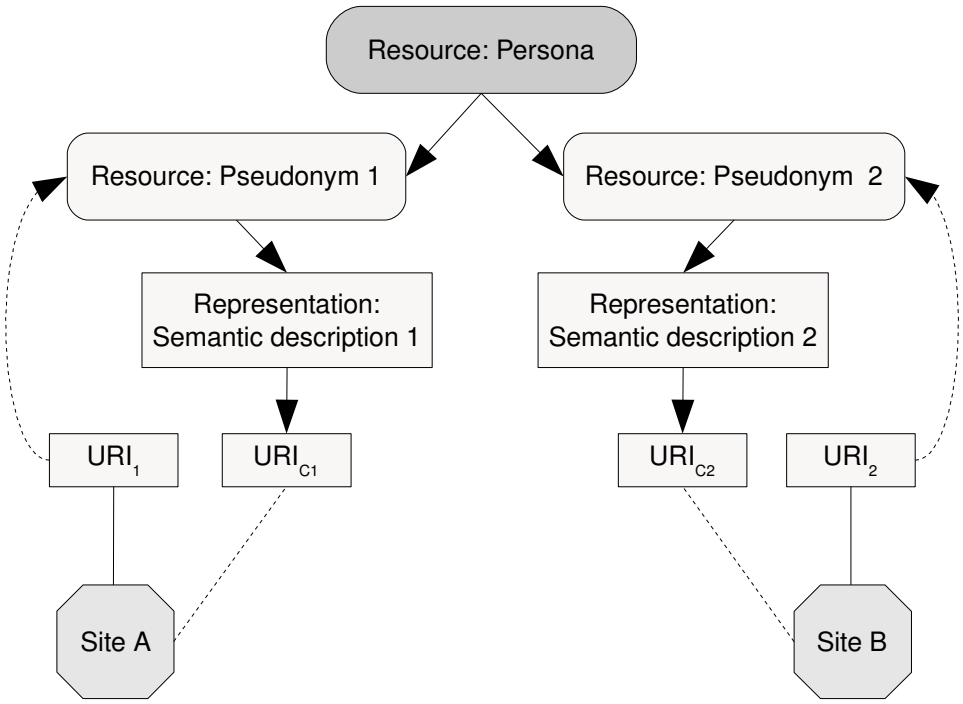

Figure 19: Linking to pseudonym resource

6.3.2.5 Security, Privacy, Time and Space

However may the idea be attractive, it is unreasonable to assume that the whole World Wide Web will contain only public, unconstrained information. In reality we must take into consideration that a flow of information will be limited. Some examples of the limitations are explained below:

- A user wants to share photos from a vacation only with his friends and family, but nobody else.
- A company wants to share business-sensitive data only with their business partners.
- A government agency will limit the access to the information only to secure devices, where the risk of information exposure is limited.
- A user is willing to reveal his e-mail address only to a business that has a good reputation and that will promise not to share such information with third-parties.

The implications of these limitations are that we cannot assume that a given representation of a resource that client have just accessed will be the same for different

user, different time, device or location. We cannot even assume that the any representation of the resource will be available at all.

It may seem that these limitations are in conflict with the global principles of the World Wide Web. We have specified an architectural constraint that the mapping of URI to a resource must be constant. But both the resource identification constraints and the access limitations can be satisfied at the same time. The URI will identify always the same resource. However, the access to the resource representation is not automatically granted. The access may be limited to authenticated entities, time and space constraints, etc.

If a user sends a URI in a message to another user, the receiver of the URI may or may be not able to access any representation of the identified resource. And even if he is able to access it, the representation provided for these two users may vary. However, the architecture mandates that any representation that is provided must be a representation of the same resource. Assuming that the resources are well-structured and maintained, the risk of misunderstanding of these two users is low.

6.3.3 World Wide Web Specifications Layer

The architectural layer of World Wide Web specification is not a primary target of this thesis. The details regarding protocol and data format improvements are rather left for a future work. However, the basic implication of proposed architecture and some basic ideas are presented in following sections.

6.3.3.1 Hyper-Text Transfer Protocol

Hyper-Text Transfer Protocol (HTTP) [56] is a primary access protocol for today's World Wide Web. It has originated as a simple TCP-based transfer protocol for hypermedia system in early 1990s. It was later amended to become general-purpose access protocol for global hyper-media system, what World Wide Web became several years later. The development of HTTP was guided by the architectural principles of REST, however even the author of REST admits [57] that there are architectural inconsistencies between HTTP version 1.1 and REST architectural style.

HTTP protocol requires HTTP URI to identify and locate the resource. It can execute several pre-defined operations on the resource, such as GET, PUT and POST. The protocol is designed and optimized to carry out all operations in a single request-response protocol exchange.

Although the identifiers in HTTP are described as HTTP URIs, these “URIs” are in fact used as locators. The authority part of the URI is resolved using DNS or parsed directly as IP address and then the connection to that IP address is made. Therefore HTTP URIs, if used with the HTTP protocol must be considered HTTP URLs instead.

The HTTP has a role of pure access protocol in our proposal of World Wide Web architecture. The HTTP operation takes URL as an address of a resource representation and executes the operation on the representation. For example the HTTP GET operation will fetch the resource representation from web server to a client. The HTTP protocol is limited to operate on HTTP URLs, it cannot process generic URIs. Therefore additional resolution mechanism is needed for a World Wide Architecture to be feature-complete.

The use of "Vary" header field in HTTP version 1.1 is not correct according to proposed architecture. It must not be assumed that a resource representation is static unless it is explicitly stated. However the "Vary" mechanisms assume that the resource representation is static, unless explicitly stated by the header. This mechanism can be made compatible with the architecture by mandating the "Vary" header in each exchange. However, we rather propose different mechanism: the "Vary" header should be replaced by "Invariant" header. The default state-change model of the resource should be undefined and no assumption about the dynamics of resource representation should be made unless explicitly stated by such header. And even if a state-change model of the resource is constrained, it should always have limits (e.g. time expiration).

6.3.3.2 HTML

Hyper-Text Markup Language [71] is a data format that describes the formatting properties of hyper-text pages. It is one of the basic standards for the World Wide Web. The support for HTML in web browsers is required for basic interoperability and usability of the World Wide Web.

We do not foresee any major changes in HTML that are imposed by our architecture proposal.

6.3.3.3 Semantic Description

According to our proposal of revised World Wide Web architecture, the semantic description is essential and mandatory representation of the each resource. Therefore at least one data format for resource semantic description needs to be part of any World Wide Web profile.

A semantic description based on Resource Description Framework (RDF) [81] could be a possible solution. RDF is a specification endorsed by World Wide Web Consortium for the purposes of the Semantic Web. The appropriate RDF vocabulary could be created to conform to the proposed architecture. The RDF description in such vocabulary could assume the role of resource semantic description, as mandated by our proposal.

The details of semantic descriptions are not part of this dissertation and are left for future work.

6.3.4 World Wide Web Profiles Layer

Profiles represent "wiring" of different specifications together. They allow the ability to develop the specification mostly independently of each other, just dependent on World Wide Web Architecture. The profiles also allow interoperability of World Wide Components: the components that are compliant with a specific profile can interoperate and use all the features implied by that profile. We envision that the profiles may be the base for WWW-enabled product evaluation or certification. The principle of profiles is based on the Uniform Interface architectural constraint according to Fielding [57]. Each profile defines a set of standards that must be implemented by the compliant component, effectively defining a complex (but uniform) interface.

A minimal profile would require only the functionality that is mandated by World Wide Web architecture. Such profile must include specification for the following elements at the very minimum:

- Data format for semantic resource representation. The specification of format for resource semantic description is mandated by proposed World Wide Web architecture.
- Access protocol. At least one access protocol must be defined for the profile to be practically usable.
- Resolution mechanism. A mechanism to resolve URIs to get resource addresses (e.g. URLs) usable by the access protocol.
- Data formats for resource source representations. At least one data format specification is needed for the profile to be practical. A maintainer of a resource that wants to be sure that his resource can be used by any component that complies with the profile should provide representation in this format. This does not mandate that all resources must have representations in this format nor does it precludes use of other formats.

The profiles define only the minimal requirements that are needed to implement desired functionality and guarantee interoperability. Any feature that is not prohibited by the specifications that are included in the profile is allowed. Only the mandatory protocols, data formats and other specifications are included in the profile. The implementation is free to support any additional specification, as long as they are not in conflict with the specifications of the profile. Therefore software component can comply to several profiles at the same time. It is also expected that the profile will be structured, for example that one profile will extend the functionality of another profile.

Two profiles are outlined in following sections: the legacy profile that describe current World Wide Web and basic profile that is a minimal practical set of features according to the proposed architecture. The provided descriptions of the profiles are just outlines. They are by no mean complete or precise. We expect that the specific profile definitions will be a result of a standardization work and broad consensus. We are not specific about the versions, dialects and variants of the specifications included in the profiles by purpose. The provided information should be sufficient for understanding the principle of profiles. It is also meant to initiate the discussion about the appropriate set of specifications for each profile.

Note: The proposed World Wide Web architecture mandates a single format for URI, however it does not define it. Therefore no profile could be complete until such format is defined and becomes part of the architecture.

6.3.4.1 Legacy World Wide Web Profile

The Legacy World Wide Web Profile describes the situation of the current World Wide Web. Strictly speaking, this is not World Wide Web profile according to the proposed architecture, as it is not consistent with our proposal. However, we consider a definition of such profile to be useful for the purpose of description of current state of World Wide Web.

The outline of Legacy World Wide Web Profile:

- The URIs mandated by the proposed World Wide Web architecture are not supported. There is no resolution mechanism for translating them to URLs. Therefore the profile is not consistent with the proposed architecture. However, once the appropriate URI format is defined in the World Wide Web

architecture, this profile might be amended to include appropriate resolution mechanism. If that happens, this profile could be aligned with the proposed architecture.

- HTTP access protocol must be supported by all clients. HTTP can work only with HTTP URLs, therefore application build on top of this profile will need to use URLs only.
- There is no explicit data format for resource semantic description. The semantic description is to be statically defined for all applications that comply to this profile. For example the description might define that all resources accessible using this profile are cyberspace data structures without any other explicit meaning. For example the resource accessed using this profile cannot be understood to represent a person or a place. It has to be understood as data structure that describes unspecified entity.
- As the semantic description is static, additional mechanism is needed to discover a source of the resource. The only practical mechanism available in current World Wide Web is to rely on the authority segment of HTTP URLs. That segment usually contains DNS name of the host maintaining the resource, however it may contain IP address as well. Therefore we could consider that the source of the resource in question is an entity addressed by the authority segment of HTTP URLs. However, the content of the authority segment of HTTP URLs is not URI and therefore it does not directly identify a resource that represents the resource source. Therefore using this mechanism will not be consistent with proposed World Wide Web architecture. Therefore we propose that due to a absence of any better mechanism any resource accessed using this profile should be considered its own source (self-asserted resource).
- HTML presentation language must be supported by all components to display resource representations in human-readable form.

The consistent implementation of legacy profile may be problematic. A method for clearly distinguishing interactions based on legacy profile from other profiles is needed. Legacy profile assumes default values for semantic meaning and source, which may be dangerous. For example a simple method to distinguish legacy profile might be defined as follows: if a resource semantic description is not available, then the use of legacy profile is assumed. However, such method may be dangerous if the semantic description would "not be available" due to the network failure. Then the values assumed by the legacy profile and the values defined by the (unavailable) semantic description might be different, resulting in an inconsistency. Therefore a method that asks for explicit confirmation of nonexistence of semantic description would be more appropriate. But even such a method may result in indeterministic state of the system in case of network failures.

The ideal solution would be to reuse existing World Wide Web standards both in legacy profile and in other profiles. For example it would be desirable for the ease of migration that both legacy and basic profiles mandate the use of HTTP 1.1. Such a goal may or may not be feasible. As the details of specifications and profiles are not the focus of this dissertation, we leave the answer to this question for future work.

6.3.4.2 Basic World Wide Web Profile

Basic World Wide Profile should provide the very minimum functionality to implement the proposed architecture and be practically usable. It is expected that this profile will be a base for most other profiles. Other profile will probably extend this profile to mandate additional functionality.

- The profile should support the URI format mandated by the proposed World Wide Web architecture. It should mandate appropriate resolution mechanism for mapping of URIs to HTTP URLs and (optionally) other addresses.
- HTTP access protocol must be supported by all clients.
- The definition of resource semantic description based on RDF should be part of the profile. The identification of resource source must be a mandatory element.
- HTML presentation language must be supported by all components to display resource representations in human-readable form.

The basic profile as described above (or one similar to it) may be a base for any World Wide Web client, server or agent. The adherence to that profile would improve interoperability. We expect that more complex profiles will be build on top of the basic profile to add support to specific features, such as code on demand, video streaming, etc.

6.4 Summary of Proposed Changes to WWW Architecture

The World Wide Web Architecture document [58] provides several formalized principles, constrains and good practice statements. The document provides a good overview of the architectural guidelines for current World Wide Web. This section summarizes the statements from the document and evaluates their validity with respect to proposed architecture. Appropriate replacements are proposed for the statements that are no longer valid. Validity of several statements is evaluated to being unknown. The proposed architecture does not directly address the matter of these statements, however the architecture may have implication on statement validity. Evaluation of these statements is not in the scope of this work.

Following statements are still valid:

> VALID: Global Identifiers (principle): Global naming leads to global network effects.
>
> VALID: Identify with URIs (good practice): To benefit from and increase the value of the World Wide Web, agents should provide URIs as identifiers for resources.
>
> VALID: URIs Identify a Single Resource (constraint): Assign distinct URIs to distinct resources.

As the URI aliases are incorporated the to the architecture and they are no longer harmful, following practice is no longer needed:

> DEPRECATED: Avoiding URI aliases (good practice): A URI owner SHOULD NOT associate arbitrarily different URIs with the same resource.

We rather propose a constraint that reflects the usage of canonical URIs:

> PROPOSED: Canonical URIs (constraint): A resource owner MUST assign exactly one canonical Uniform Resource Identifier to a resource at any specific time.

Following statement is still valid:

> VALID: Consistent URI usage (good practice): An agent that receives a URI SHOULD refer to the associated resource using the same URI, character-by-character.

The proposed URI architecture mandates a single and unified format for URIs, therefore following practice is no longer valid:

> DEPRECATED: Reuse URI schemes (good practice): A specification SHOULD reuse an existing URI scheme (rather than create a new one) when it provides the desired properties of identifiers and their relation to resources.

However, a constraint that reflects the unified URI format should be introduced:

> PROPOSED: URI Uniformity (constraint): All URIs must conform to the format and meaning defined by the WWW architecture.

Following statements are still valid and the reasoning provided by current World Wide Web architecture documents holds:

> VALID: URI opacity (good practice): Agents making use of URIs SHOULD NOT attempt to infer properties of the referenced resource.

> VALID: Reuse representation formats (good practice): New protocols created for the Web SHOULD transmit representations as octet streams typed by Internet media types.

> VALID: Data-metadata inconsistency (constraint): Agents MUST NOT ignore message metadata without the consent of the user.

As the proposed architecture mandates the presence of semantic description, we recommend to modify following statement:

> DEPRECATED: Metadata association (good practice): Server managers SHOULD allow representation creators to control the metadata associated with their representations.

We make to make the statement stronger:

> PROPOSED: Metadata (constraint): Server manager MUST provide ability to manage resource semantics and representation metadata. Client software MUST process these data.

We consider following principle useful, therefore we recommend to keep it a part of the architecture. However, the actual means and architectural constraint to implement this principle are missing. An introduction of generic principle of *operations* for access protocols will be necessary in the architecture. The implementation of these operations should be defined at the level of specifications, in the specifications of access protocols. The concept of *safe* or *obligation-free* operations needs to be defined on the architectural level as it needs to be shard concept among all protocols. This aspect is not addressed in the current architecture and it is left for future work. We recommend to define it as a principal guideline for now and to transform it to constraints during later revisions of the architecture:

> VALID: Safe retrieval (principle): Agents do not incur obligations by retrieving a representation.

Following statements is still valid:

> VALID: Available representation (good practice): A URI owner SHOULD provide representations of the resource it identifies.

However we propose new constraint that reflects the need for a semantic description:

> PROPOSED: Semantic description (constraint): A resource representation in a form of semantic description MUST be provided if any other resource representation is provided as well.

Such definition allows for a resource without any available representations. Such case may be valid in case of security constraints, network connectivity limitations, protocol mismatch, etc. However, if at least a single representation is provided, the semantic description must be provided as well.

Following statements are still valid:

> VALID: Reference does not imply dereference (principle): An application developer or specification author SHOULD NOT require networked retrieval of representations each time they are referenced.

> VALID: Consistent representation (good practice): A URI owner SHOULD provide representations of the identified resource consistently and predictably.

We propose to add another statement that will provide clarification of the apparent mismatch of the globalism of the World Wide Web and time, space, privacy, security and accessibility constraints:

> PROPOSED: Resource dynamics (constraint): The user of a resource representation MUST NOT assume that the representation presented to him will be the same if accessed at different time, by a different user or if a space location or context are different, unless an appropriate constraint is indicated in the metadata.

Following statements related to data formats are still valid. The validity of the statements related to XML is not evaluated, as the XML data format is not considered

by this dissertation. Therefore the validity status of these statements is indicated as UNKNOWN:

> VALID: Version information (good practice): A data format specification SHOULD provide for version information.
>
> UNKNOWN: Namespace policy (good practice): An XML format specification SHOULD include information about change policies for XML namespaces.
>
> VALID: Extensibility mechanisms (good practice): A specification SHOULD provide mechanisms that allow any party to create extensions.
>
> VALID: Extensibility conformance (good practice): Extensibility MUST NOT interfere with conformance to the original specification.
>
> VALID: Unknown extensions (good practice): A specification SHOULD specify agent behavior in the face of unrecognized extensions.
>
> VALID: Separation of content, presentation, interaction (good practice): A specification SHOULD allow authors to separate content from both presentation and interaction concerns.
>
> VALID: Link identification (good practice): A specification SHOULD provide ways to identify links to other resources, including to secondary resources (via fragment identifiers).
>
> VALID: Web linking (good practice): A specification SHOULD allow Web-wide linking, not just internal document linking.
>
> VALID: Hypertext links (good practice): A data format SHOULD incorporate hypertext links if hypertext is the expected user interface paradigm.
>
> UNKNOWN: Namespace adoption (good practice): A specification that establishes an XML vocabulary SHOULD place all element names and global attribute names in a namespace.
>
> UNKNOWN: Namespace documents (good practice): The owner of an XML namespace name SHOULD make available material intended for people to read and material optimized for software agents in order to meet the needs of those who will use the namespace vocabulary.
>
> UNKNOWN: QNames Indistinguishable from URIs (constraint): Do not allow both QNames and URIs in attribute values or element content where they are indistinguishable.
>
> UNKNOWN: QName Mapping (good practice): A specification in which QNames serve as resource identifiers MUST provide a mapping to URIs.

> UNKNOWN: XML and "text/*" (good practice): In general, a representation provider SHOULD NOT assign Internet media types beginning with "text/" to XML representations.
>
> UNKNOWN: XML and character encodings (good practice): In general, a representation provider SHOULD NOT specify the character encoding for XML data in protocol headers since the data is self-describing.

According to the proposed architecture a specific URI syntax should be mandated. Therefore following statement is no longer valid:

> DEPRECATED: Generic URIs (good practice): A specification SHOULD allow content authors to use URIs without constraining them to a limited set of URI schemes.

Following principles are still valid:

> VALID: Orthogonality (principle): Orthogonal abstractions benefit from orthogonal specifications.
>
> VALID: Error recovery (principle): Agents that recover from error by making a choice without the user's consent are not acting on the user's behalf.

6.5 Migration Outline

It is obvious that current World Wide Web cannot be changed to the proposed architecture in one day. The process how to implement the architectural will be most probably slow, adapting the specifications, modifying the software to comply the specification and finally deprecating old specifications. The whole process can take several years or maybe event decades. The basic steps required for the migration are outlined below:

1) Any sting that is currently considered to be URI should be considered to an URL instead. Update existing specifications to reflect the correct terminology to avoid confusion.
2) Define a unified URI format. It seems that the easiest way would be to define the syntax of new URIs to be backwards compatible with today's URL. In this case almost no changes in the data formats that support hyperlinking (such as HTML) will be needed. It may prove useful to include existing namespace into the URI namespace to ease the migration. For example, if existing DNS namespace would be a subset of proposed URI namespace, then anyone in a possession of a DNS domain would have already allocated part of the URI namespace.
3) Define a basic resolution mechanism how to translate URIs to HTTP URLs and other addresses. An existing naming mechanism might be used for that. For example a DNS specification could be extended with additional type of Resource Record that can specify pointers to URL space. Therefore the resolution mechanism could be based on DNS resolution until the pointer is

located and then appending unresolved part of the URI to the URL prefix in the pointer Resource Record.

4) An RDF vocabulary for resource semantic description should be defined. Alternatively other data format than RDF should be used for this purpose, but just one approach should be mandated.
5) HTTP should be used as access protocol. The HTTP specification should define a mandatory content type or URL format for accessing semantic description of a resource.
6) Mandate the use of all the extensions and specification above in a Basic World Wide Web Profile.
7) Define other profiles on top of the basic profile, such as the Code on Demand profile which will mandate support for a scripting language in the clients (browsers).
8) Improve the resolution mechanism and access protocols as needed. Reflect these changes in undated Basic World Wide Web Profile.
9) Deprecate old specifications and later deprecate all intermediary steps taken during the migration (such as the DNS resolution mechanism).

The steps described above are just an outline. It is expected that more than just the activities specified above will be needed. For example it could be expected that a generic authentication or security mechanism for the new World Wide Web architecture will be proposed. Such mechanism as well as other proposals and extensions have to be incorporated the migration path.

6.6 Validation of the Architecture

Proposed architecture resides mostly on conceptual level. It is considerably abstract, with only a few concrete proposals. Therefore usual architectural validation by prototyping key elements is not applicable for this proposal. The architecture is not defined in a sufficiently formal way to mount a formal proof of correctness. Therefore we have decided to use scenario-based validation of the architecture. Following sections describe the scenarios used for demonstrating efficiencies of current World Wide Web architecture in section 5.4. The solutions compliant with proposed architecture are provided for each of the scenarios and the solution properties and variants are discussed. Each solution also contains a list of essential architectural elements used in the solution.

Although the scenario-based validation cannot prove formal correctness of the architecture, it increases confidence in the architecture appropriateness and usability in scenarios that are close to practice. It is also used as a checking mechanism to uncover any obvious problems. The list of essential architectural elements in each scenario is used to make sure that all of proposed architectural constructs are used and therefore none of them is obviously redundant.

The goal of the validation is to show that the proposed architecture is an *improvement* over the original World Wide Web architecture. That is demonstrated on following scenarios. These scenarios are shown (in section 5.4) to fail under the current architecture. Following section describe how such scenarios can be implemented under proposed architecture, thus demonstrating an improvement.

6.6.1 Private Photo Sharing Scenario

Actors:
user, friends

Prerequisites:
User has a set of photos

Summary:
User wants to a photo with friends and only with his friends.

Solution Setup:

1) User and his friends create the representation of their personas as WWW resources.
2) User links his persona resource to the persona resources of his friends.

Solution Scenario:

1) User uploads his photo to his server, creating WWW resource from it, assigning a URI. Marking it as accessible only to the friends.
2) User will discover appropriate communication mechanism available to communicate with his friends by examining the persona resources of his friends (e.g. by using HTTP GET, requesting appropriate representation of the persona resource).
3) User sends the URI to his friends, using the communication mechanism and addresses discovered in previous step.
4) Friends will use the URI to see the photo, resolving it to URLs and using HTTP GET request to get a representation of the photo resource. The server that hosts the photo resource will detect that the resource is available only for friends, therefore it will challenge friends for authentication. It will use resource source link to obtain persona resource of user and check whether the accessing friend is in the list of user's friends.
5) When challenged for authentication, friends will present credentials proving their control of their persona resources (e.g. using suitable single sign-on mechanism).
6) User's web server will check validity of provided credentials (which may include communicating with friend's persona resources). If validation succeeds, the server will use resource source link to obtain persona resource of user and check whether the accessing friend is in the list of user's friends. If he is, the representation of the photo resource is provided.

Essential architectural constructs:

- Persona resources
- Resource source

Discussion:

- This scenario takes advantage of representing WWW users as resources, which is referred to as *resource personas* in the scenario. Resource personas are used

to maintain a list of friends, in a form of links to the resource personas of the friends.

- While the URL is communicated to friends using unspecified communication mechanism, the details of such mechanisms and the addresses are discovered from persona resources of friends. The system can be augmented by a communication mechanism that would operate on URIs and use WWW protocols. Such protocol may form example use HTTP POST method of the notification to the friend's persona resource.
- It is assumed that a suitable mechanism will exists that will allow any user to prove his control over his persona resource. An example of such mechanism could be a variant of one or more of the existing Internet identity systems, described in section 1.3.

6.6.2 Shopping Collectibles Scenario

Actors:

buyer, seller, maintainer of buyer's persona, maintainer of seller's persona

Prerequisites:

Seller may or may not own a precious collectible.

Buyer and seller do not have any previous relationship or haven't been involved in any previous interaction.

Summary:

Buyer wants to buy a precious collectible that seller claims to own.

Solution Setup:

1) Both buyer and seller create resources representing their personas. They slowly gain good reputation by taking part in regular day-to-day activities, interacting with other people, buying and selling non-expensive items in low-risk deals. The reputation is maintained by the entity that maintains their persona resources.

Solution Scenario:

1) Buyer finds the offer of seller to sell a collectible that he desires.
2) Buyer uses resource source link from the offer to contacts the seller. Both buyer and seller check each other reputation, to make sure that the other party is in good standing. They agree on the price.
3) Buyer sends money to seller.
4) Seller receives the money. He sends the collectible to the buyer.
5) Buyer receives the collectible. He checks it and finds it worthy. He will express his positive opinion about the deal to the maintainer of seller's persona. He discovers the maintainer by following the identifier of resource source of seller's persona.
6) Seeing positive feedback of the buyer, seller also expresses positive feedback about the deal and submits it to maintainer of buyer's persona. He discovers

the maintainer by following the identifier of resource source of buyer's persona.

Essential architectural constructs:

- Persona resources
- Resource source

Discussion:

- In case that one party decides to cheat, the other party will express negative feedback. As both parties present good reputation that was gained by a slow (and costly) process, they are motivated to maintain it. Neither party would be interested in engaging in a deal with a partner that presents low reputation. Alternatively the price can be adjusted to compensate for the risk indicated by lower reputation of one of the parties. Such approach is also discussed by Friedman and Resnick [54].
- As reputation is not centralized, the system may scale acceptably well. However there needs to be a way how each party can check the reputation of other party. It either needs a direct trust relationship with maintainer of other party's persona, or such relationship has to be mediated by his persona maintainer or other intermediaries.
- The scenario takes advantage from expressing personas of the actors in a form of resources that can be made accessible by the other party. It also takes advantage of the resource source mechanism that is used to locate the maintainer of the persona resources and therefore also maintainer of his reputation.

6.6.3 Resource Rating Scenario

Actors:

reviewer, editor, reputation system, search engine

Prerequisites:

- A URI is assigned to represent an old house, requests to that URI will return the representation of the house in a form of digital photograph. The URI will also present a mandatory semantic description, describing the resource as a representation of realspace object: a house.
- Photograph of the house is taken only for documentation purposes, without any artistic merit. The composition and lighting were not considered important and it is even slightly blurred.
- Reviewer reviews a photograph of the house and has an opinion about the house age, location and aesthetic quality.
- All the actors maintain personas in form of WWW resources.
- Search engine is integrated with reputation service, ordering search results based on the reputation scores provided by reputation system.

Summary:

Reviewer wants to express that opinion to help potential buyers evaluate the quality of the house.

Editor wants to find quality photographs of old houses to use as an illustration in his magazine.

Solution Scenario:

1) Reviewer logs in to the reputation service, e.g. by using single sign-on mechanism.
2) Reviewer will express his positive opinion about the house, rating the resource that represents the house with "excellent" rating.
3) The reputation service will process that rating, combining with previous rating or rating of other reviewers. Reputation service will expose the (positive) reputation score of the resource (identified by URI) to the public.
4) Search engine will index the resource representing the house. Search engine will query the semantic description of indexed resources, therefore it will be aware that this particular resource represents a house, not a photograph. It will associate the important data from semantic description with the resource in the index. The search engine will check the reputation score of the resource (identified by URI) with the reputation system and associate that with the URI in the created index.
5) Editor queries search engine for the photographs of houses. Editor will indicate that his interest is to get good quality photographs.
6) Search engine will use the index, the metadata in the index and the reputation scores to satisfy the query. It will present results to editor. The photograph considered in previous step will be among the last results or it will even be omitted. Even though it exhibits high reputation score, the type of the object does not match with query, therefore the search engine will be aware that the reputation score is not applicable.

Essential architectural constructs:

- Resource representation
- Persona resources
- Resource source
- Resource semantic description
- Subjectivity of resources

Discussion:

- The editor will get what he wants. The non-applicable reputation score will not deform the results.
- The search engine may provide controls for expressing feedback about the displayed item. Therefore if someone finds a way how to work around the reputation algorithm to artificially boost up reputation score, the score may be influenced to drop to the realistic value.

- Reputation system can be rated for quality of his predictions. Therefore if it rates a poor object with a low score, it will get positive reputation. If he rates poor object with high score, it will get negative reputation. Therefore the reputation system will be motivated to maintain his predictions accurate. That may mean that if someone finds a way to circumvent the system to gain high scores, the maintainers of the system will be motivated to fix the problem as soon as possible.
- The reviewer's opinion about the house may be published as a resource as well, maintaining a reputation score. If reputation system detects that a reviewer expressed an opinion that is not objective, it may express negative feedback about such opinion. As the opinion and the persona of the reviewer are linked (by a *resource source* relation), such negative feedback on the opinion may also influence the reputation of the reviewer.
- The weight of the house resource rating may be influenced by the reputation score of the reviewer in step 3. Therefore the opinion of the reviewers that do their job better will be more relevant for final reputation score.

6.6.4 Fake Breaking News Scenario

Actors:

readers, publisher, reputation systems

Prerequisites:

- Publisher makes up a story about an event that is likely to gain wide attention, may have severe impact on general public but it is very difficult to verify. For example a story that secret military technology has fallen into the hands of criminals and that government is covering that up to avoid negative publicity.
- Publisher controls a minor media channel (e.g. a blog) with a small group of readers.
- Publisher and readers have established personas represented as WWW resources.
- Publisher has a neutral reputation maintained by few reputation systems.
- Publisher and readers does not have any strong a-priori relationship.
- Few of the readers follows publisher's channel.

Summary:

Publisher wants to popularize his channel by publishing a fake story.

Reader wants to be informed about any dangers that may arise.

Solution Scenario:

1) Publisher publishes the fake story on his blog in a form of WWW resource.
2) The readers will get to the story. Reader's software (e.g. web browser) will follow a *resource source* link from the store to the publisher's persona. The software will check the reputation of the story and reputation of the publisher. The software will display appropriate visual cue to represent neutral reputation

of publisher and unknown reputation of the story. It may for example change the color of browser's border and chrome to gray.

3) Some readers will not believe it and silently dismiss it, but few readers will believe it and spread the word in a good faith to protect their friends from misuse of secret military technology.
4) The story will spread, triggering both positive and negative reactions. The popularity of publisher's channel will rise. People will express both positive and negative feedback about the story. That feedback will influence the reputation score of the story and also reputation of the publisher. Reputation of the publisher will not be considerably affected if the believers and non-believers are approximately equal. Publisher will not gain any substantial advantage.
5) As the story cannot be proven or falsified the attention will cease in relatively short time, without any substantial advantage for the publisher.

Essential architectural constructs:

- Persona resources
- Resource source
- Subjectivity of resources

Solution problems:

- The readers will still face the decision to believe or not believe a non-verifiable story, however they will provided with a cue to support their decision. The work of Friedman and Resnick [54] shows that the system can be improved by introducing a bias towards distrusting newcomers. Therefore the publisher's reputation could be initialized as bad. It is not likely that many people would believe a breaking story presented by a person with bad reputation. Therefore he would have to gain his reputation by publishing several true and verifiable stories that would gain him enough feedback to risk the fake story. This effort may be enough to motivate him not to risk the gained reputation by a fake story.
- The weight of the feedback from readers may depend on reader's reputation. Therefore a negative feedback from several important readers may compensate for a positive feedback of many non-reasonable readers. This may be enough to turn the scales and the risk may not be worth it. However, this is only a speculation without any results to support it.

6.6.5 On-Demand Content Distribution

Actors:

users, distribution nodes, seed node

Prerequisites:

- Content is stored in the form of big file on the seed node.

- Seed node is connected to the network by a relatively weak physical link. It can well handle transfer of one content copy to the rest of the network but it would be heavily overloaded for more than 10 simultaneous transfers.
- Distribution nodes are located in a physically separate locations worldwide
- Distribution nodes are willing to maintain a copy of the content, but only as long as it is needed.
- There is a large number of users at least in order of millions.
- Users are physically distributed across the world and their location is not static.

Summary:

Users want to get the content in a short time duration (several minutes or hours). Users want to get the content as quickly and efficiently as possible. They want to maintain appropriate security while getting the content. Seed node wants to make the content available to the network, however it wants to avoid overload of its link and resources. For that purpose the seed node makes a deal with the distribution nodes. Distribution nodes agree to distribute the content of seed node.

Comment:

This scenario well describes situation of a non-popular site quickly becoming popular, for example by publishing link to that site in a popular communication channel (widely known as *slashdotting* or *slashdot effect*).

Solution Scenario:

1) Seed node will sign the content with its private key and adds that signature to the resource semantic description. Alternatively the signature may be separate resource representation and only its presence indicated in the semantic description.
2) Seed node publishes abstract URI for the content. Seed node will delegate the last step of URI resolution to the distribution node network.
3) User's client software tries to resolve the URI to concrete locators (URLs). The resolution will be directed to a resolved on a distribution node.
4) Name resolver of distribution nodes will use request source address to choose a good distribution node for the client. It will respond with URL pointing to selected distribution node.
5) Client software will use the URL to reach distribution node.
6) Distribution node receives a request from user, it will use internal distribution algorithm to locate a closest copy of content in the distribution network. If such copy is not available, it will download it from seed node and create the first copy itself. It will also get a semantic description and signature from the seed node.
7) Distribution node responds to the user with a copy of the content.
8) Subsequent requests can be used by the client to get semantic description and signature of the resource.

Essential architectural constructs:

- Abstract URI
- Persona resources
- Resource source
- Semantic description

Discussion:

- Although additional requests are needed to resolve abstract URI to concrete URLs, they may well be worth the benefits of flexibility.
- Neither user nor seed node needs to trust distribution nodes.
- The resolution of URI may result in several URLs referring to several nodes. If one of the URLs fails, the client may try other URLs as a mean of fault resistance.

6.6.6 URI of the Sun Problem

Problem outline:

What is the URI of Sun, the star Sol of the Solar System?

Solution details:

There is no such thing as the URI of the Sun. Each information in cyberspace is subjective, it should be considered an opinion. Therefore there will be plenty of *opinions* about the Sun, each having its own URI. Some of the opinions will be broadly accepted, enjoying popularity and good reputations. Some of them may be controversial, less popular or otherwise marginal. However even such opinions have a right to exists and be presented.

The obvious problem is how to judge the multitude of opinions about realspace objects. The resource source link form the opinion about the Sun to its author can be used as a cue to judge the credibility of the information. If the information is provided by an astronomer in good standing, the information will likely be reliable. A reputation system can help to judge what means “in good standing”. However even the reputation may be subjective. For example astrologers may place good deal of trust for a reputation system that rates the resources according to their compliance with arcane arts rather than scientific practice. Therefore for a novice astrologer an information from a famous seer may be seen much more credible than from an established astronomer.

Essential architectural constructs:

- Persona resources
- Resource source
- Subjectivity of resource

6.6.7 Multi-Protocol Access Problem

Problem setup:

User has multi-protocol client software. This software can download the content using HTTP for later viewing, it can stream the content using more efficient streaming protocol for immediate viewing or it can use HTTPS for secure download of sensitive content with when a performance penalty is acceptable. The user also has a hand-held device that can connect to the network only by a link with low throughput. That device does not have enough storage to store entire content, therefore common download technique is not possible. The device requires special-purpose optimized streaming protocol to operate.

Web server that hosts the resource (content object) can support all of the above protocols.

Problem outline:

What URI to use for the resource?

Solution details:

Resource will be assigned an abstract URI that is protocol-independent. Name resolution of that URI results in a set of URLs that may be protocol dependent. The client will then select the appropriate URL for available access protocols or URL that is appropriate for the situation. This procedure may include examining resource semantic description and/or protocol-specific content negation for example to select the correct resolution and bit-rate of the stream.

Essential architectural constructs:

- Abstract URI

6.7 Drawbacks

Proposed architecture introduces a couple of drawbacks and trade-offs. The drawbacks that we are aware of are described in this section.

Multi-step name resolution is a direct consequence of URI abstractness. Location-independent identifiers naturally introduce additional level of indirection therefore increasing the overhead of name resolution. However we believe that such trade-off is well balanced with benefits gained by abstract identifiers, as can be seen on the scenarios described in section 6.6.

Mandating semantic description and resource source identification introduces additional complexity to web servers. Mandating semantic description does not influence simple clients as they are free not to use it. But the situation is different on the server side. Web server needs to be more complex and it also increases complexity of content management. However web applications are usually already aware of the nature of resources they present and they usually also know the source of the resources. Therefore the additional complexity is only in presenting that information along with the resources. Situation is a bit different for simple static web content. However in that case the web server may provide simple mechanism how to statically specify such information for each file or a set of files. In the worst case the web server may provide

default semantic description specifying the resource as *unspecified binary file* and specifying the server administrator as a source. However, such approach should be strongly discouraged if more information is available.

The proposed architecture is specified on conceptual level only. Although the architecture was validated by walking through a set of scenarios, it is expected that more drawbacks will surface when the architecture will be applied to the design of individual components, protocol specifications and profiles. We do not consider such event to be a failure of proposed architecture as long basic architectural principles holds. Such problems are expected and once they are uncovered they should be fed back to the architectural process, inducing changes in the proposed architecture, following a sound iterative development approach.

7 Contributions to the Field

The motivation of this work was to discuss what went wrong in the course of Internet evolution and to propose and improvements that could help change the Internet to a more desirable environment. The focus of this work was on the most important part of the Internet: the users. To lay the foundation for this work, we have examined the interactions of the physical personas with computer systems. In chapter 3 we have proposed a model that can explain some of the aspects of such interactions. It was demonstrated how the proposed model could be used to evaluate such properties as identity and anonymity. We consider the model still being in its infancy and we expect that follow-up work will substantially extend and improve it. However, we have demonstrated the usefulness of the model by applying it to the evaluation of current World Wide Web architecture, identification of architectural problems and proposals of improvements.

Any meaningful architecture must have a clear goal that it is meant to meet. The goals and the approach of our architectural proposals were discussed in chapter 4. The desired environment should support cooperation of people by the positive network effect. It should encouraged people to do business together, to cooperate on the projects, to re-use the results of each other work. We do not expect a Utopian community where everything will be available freely. A cost may be part of the cooperation and we seek an environment where that may be possible and efficient. Three principal environments are described: anarchy, authoritarianism and environment of responsibility. Properties of these environments are discussed and the environment of responsibility is evaluated as the best choice for cooperation. Privacy aspects of the desired environment are recognized as an essential part of the system. The privacy discussion emphasizes the proper balance of privacy and free speech. Trust is discussed as a fundamental enabler of efficient cooperation. A reputation mechanism is recognized as a suitable system to support trust-related decisions. An outline of the basic reputation mechanism for a global environment is provided.

This dissertation have identified major problems in the Internet architecture, especially the problems of the architecture of World Wide Web. The problems are described in detail in chapter 5. The major problem areas include:

- Addresses being used for identification purposes, including IP address and some URIs.
- Assumption that World Wide Web resources are static.
- Weak definition of World Wide Web interfaces and inconsistent application of architectural principles.
- Vague meaning of World Wide Web resources.
- Protocol-dependence of World Wide Web Architecture.
- Inappropriate security mechanisms.

Revised architecture of World Wide Web is proposed in chapter 6. The proposal is based on the evaluation of the problems of current World Wide Web architecture. It is attempt to amend the basic architectural principles and guidelines, especially in a way

that would reflect the implications of the model provided in chapter 3. We have proposed to divide the field into several layers of abstraction:

- Architectural Styles: abstract set of architectural constraints, not necessarily specific to World Wide Web.
- World Wide Web Architecture: combination of architectural styles and constraints specific to the environment of World Wide Web, yet still abstract enough to be protocol-independent.
- World Wide Web Specifications: a layer that contains concrete specification of World Wide Web protocols.
- World Wide Web Profiles: define a collection of specifications that form a unified interface for interoperability purposes.

The architectural styles layer defines RRSS, a new architectural style inspired by the REST style. The RRSS style builds on four basic components: Resource, Representation, Source and Semantics. While the concepts of resource and resource representations are adopted from the REST architectural style, the concepts of resource source and semantic description are our additions to the architectural style. Especially the concept of resource source is an implication of our model on the architecture of World Wide Web.

The RRSS architectural style, combined with other styles is applied to the World Wide Web architecture. The result is (still abstract) set of architectural guidelines that govern the basic principles of World Wide Web and provide foundation for protocol specifications. The World Wide Web architecture combines the architectural constraints of RRSS and other style to adapt them for the World Wide Web environment. A redefined concept of Uniform Resource Identifier (URI) is outlined in a form of ideas and requirements, while specific definition of the URI is left for future work. The concept of resource source is specified in a more concrete form. It is defined as a resource representing a persona of resource owner. Such recursiveness simplifies the principles of the model, while not constraining its flexibility. Security and privacy aspects of proposed architecture are shortly discussed as well.

The basic ideas of World Wide Web specifications and profiles are outlined, however they are not considered to be a focal point of this work. Principal implications on HTTP specification are shortly discussed. The description of World Wide Web Profiles layer outlines two profiles: Legacy and Basic. The legacy profile is aimed at description of compatibility with current World Wide Web and it is not strictly consistent with the proposed architecture. The basic profile described the basic set of features that World Wide Web software must implement to be usable and in accord with the proposed architecture. The end of chapter 6 summarizes a set of changes to current World Wide Web architecture document to make it compliant with proposed architecture. A short outline of a migration process from current World Wide Web to the proposed architecture is also provided.

The primary principle of this work is that the crossing of realspace-cyberspace boundary is subjective. Therefore all information in the cyberspace should be regarded as *opinions*, rather than unquestionable truths. The implication of that principle is the introduction of the concept of *source* to the RRSS architectural style. This concept is then reflected to the World Wide Web architecture in a form of resource source

identifier in a resource semantic description. Such an approach will effectively make the identity-related mechanisms an integral part of World Wide Web architecture.

This work is theoretical, its focus is on the conceptual principles of the architecture rather than protocol specifications. Although some ideas, constraints and proposals for specific protocols are part of this work, they are provided to demonstrate how the principles could be applied. Apart from that no other proof of correctness or architecture feasibility is provided. Although no claim about the feasibility of application of the proposed architecture is made, we believe that it can be a useful guideline to judge architectural decisions for ongoing evolution of World Wide Web.

7.1 Fulfillment of Dissertation Objectives

The objectives of the dissertation that were defined in chapter 2 were met:

- Goals and expectations for the World Wide Web architecture and design for a current needs and for a foreseeable future were defined in chapter 4.
- State, consistency and appropriateness of World Wide Web architecture were evaluated in chapter 5, based on the model introduced in chapter 3 and the goals defined in chapter 4. Current architecture of World Wide Web was found inconsistent and inappropriate for today's needs.
- Fundamental problems of WWW architecture were identified and discussed in chapter 5. These problems were described in detail and demonstrated on a set of scenarios.
- Architectural improvements in World Wide Web architecture were proposed on a conceptual level in chapter 6. Proposed architecture includes support for evaluation of information reliability as its integral part. Improved architecture of World Wide Web is the primary contribution of this work.
- Proposed architecture was validated using a set of scenarios in section 6.6. The same scenarios that were used for demonstrating deficiencies of current World Wide Web architecture were used, demonstrating the improvement of proposed architecture as compared to the current architecture.

7.2 Theoretical Contributions

This work makes the following contributions to the research within the field of Information and Computer Science:

- A model that describes interactions between realspace and cyberspace, especially focused on representation of personal data in the cyberspace.
- A mechanism for evaluation of anonymity and identity based on the proposed model.
- Assessment of architectural inconsistencies of World Wide Web architecture. Both internal inconsistencies and problems uncovered by the application of the proposed model are described.

- A definition of RRSS, a new architectural style for representing information in cyberspace while taking into consideration their potential source and target in realspace.
- Application of the RRSS architectural style together with other previously described styles to a World Wide Web architecture, resulting in a proposal of improved architecture for World Wide Web.

7.3 Practical Contributions

This work is focused on theoretical aspects of World Wide Web architecture and the proposed improvements are concentrated in the most abstract layers of the architecture. However, there are several practical contributions of this work:

- The model introduced in chapter 3 defines a concepts of analogy and heterology that can be used in computer system instead of identity and anonymity. As identity and anonymity are difficult (if possible at all) to determine by a computer system, the analogy and heterology can be wholly determined in the cyberspace. Therefore analogy and heterology may be used as more practical metrics, instead of identity and anonymity.
- Practical problems of current World Wide Web were identified and demonstrated on a series of scenarios in section 5.4. Several solutions for each of the scenarios was proposed and the problems of the these solutions were identified. However, all of the proposed solutions were found insufficient and it was concluded that architectural change is necessary to fully address the issues.
- Proposed World Wide Web architecture can be used as a reference architecture to evaluate and judge design decisions during the course of World Wide Web evolutions. The developers frequently face a decision where all the alternatives seems to bring the same short-term benefits. However, the long-term benefits of these solutions may not be apparent unless a long-term development goal is set. The architecture proposed in this work could be considered such a long-term goal. It could help developers, engineers, designers and standard bodies to evaluate impact of a particular design decision or approach to the long-term development of World Wide Web.
- A summary of changes necessary for current World Wide Web Architecture document [58] is provided in section 6.4. These changes are ready to use and incorporate into the architectural document, which can be used to guide future development.

7.4 Future Work

Although we achieved consistency and completeness at the conceptual level, we acknowledge that the proposal is not sufficient for practical implementation and that many technical details are missing. The future work should be focused on filling out the details:

- definition of appropriate URI syntax and semantics,
- specification of URI resolution protocol,

- details of resource semantic description,
- proper definition of World Wide Web profiles,
- the details of migration path and
- prototypes for validation of the architecture and design.

Conclusion

The Internet and World Wide Web are revolutionary technologies. They allow people to cooperate and share information. The Internet was not created in its current form, it has rather evolved in time. Similar evolution also applied to the World Wide Web, however that was partially guided by the architectural principles of REST. Such evolutionary approach worked perfectly to address simple needs of the environment and guarantee the survival of the Internet and World Wide Web. However, it failed to address more complex needs, such as privacy and data authenticity.

Up until recently most of the information available on the World Wide Web were public or intended for public usage. Therefore only minimal mechanisms evolved to protect the information, as protection was not necessary. The Internet was composed of sites, where one site was a consumer of information provided by other site. Apart from this simple interaction no other relationship existed. The users of the Internet were considered to be only consumers. All publishing and data sharing activities has to be done through the sites. The Web was designed and used under a “few publishers, many readers” paradigm, where the publishing of the information itself was regarded as “out of band” for the Web architecture.

This paradigm of the “static Internet” is changing. Large amount of non-public information is being transferred over the Internet. This information cannot be simply classified as private or public. The classification is of much finer grain and contains a degree of fuzziness: share information with my friends, with magazine subscribers, with business partners, with premium customers, etc. The organization of the Internet into strictly separate sites may also be challenged. The rise of peer-to-peer networks does not operate on the concept of a site. Replication and migration of data on demand is a modus operandi of these networks. Therefore the concept of “source of data transmission” is no longer useful for these networks. The parading of “few publishers, many readers” has transformed to “few publishers, many contributors, hordes of readers”. The World Wide Web is no longer read-only information system, it becomes writable. The rise of applications such as web forums, wiki, blogs, photography and video sharing has placed a very different set of requirements about the interaction capabilities of the web. Now we take for granted the ability to comment on a blog post or a photo from friend's vacation. However, these paradigm shifts are not well supported by current World Wide Web architecture.

We have proposed an improved architecture of the World Wide Web, adapting the basic architectural concepts to meet new requirements. We have introduced clean layering of World Wide Web architectural layers without complex interdependencies. We have defined the responsibilities and functions of individual layers, focusing on the most abstract layers because these have the most significant influence on the functions of World Wide Web. Our architectural work was guided by a model of realspace-cyberspace interactions that guided basic architectural ideas. The architectural form was shaped according to its desired function, seeking to induce good cooperation in the environment of responsibility and preserve privacy of the users.

The proposed architecture was validated on a series of scenarios that are difficult to implement in current World Wide Web. However, even if the concepts were validated, it does not necessarily mean that the architecture is appropriate. The goals for the architecture and the desired environment may not be specified correctly or the architecture may fail to support such environment due to unforeseen change in society. Sequential software development processes were shown to be flawed methods of software development, therefore we have no ambition to follow them. We rather see the correct approach in using iterative and incremental processes. The current World Wide Web may be seen as a first iteration in such a long-term process. We have identified the problems of current World Wide Web, identified the reasons of those problems and reflected gained knowledge back to the architecture. Thus we have closed the cycle of first large-scale iteration of World Wide Web development. Second iteration should start from this point, applying the conceptual changes to individual protocol, data format and profile specifications, creating a next generation World Wide Web. Implementation and use of the new World Wide Web will undoubtedly uncover problems that we have not foreseen. It may even uncover flaws in the proposed architecture similarly to the problems that we have uncovered in current World Wide Web architecture. However, it should not be considered a fallacy of this work but rather a normal course of iterative software development.

The effort to reshape the World Wide Web is far from complete, it is rather at its very beginning. This work lays a basic conceptual foundation for future works that can add more technical details, shape the specification and test the system in practice. The architecture of Internet-based system cannot grow on a green field. They need to coexist with currently deployed and widely used technologies, even if the deployed technologies are far from ideal. Our architectural proposal is formed as an extension and improvement of existing system: World Wide Web itself. We hope and believe that our work can help change the thinking behind the World Wide Web architectural to be more focused on the nature of the provided information, more supporting to the ad-hoc cooperation of people while preserving their privacy.

Bibliography

[1] STEINER, P.: *"On the Internet,Nobody Knows You're a Dog" cartoon*. The New Yorker, Vol.69 (LXIX) no. 20, 1993.

[2] SCHNEIER, B.: *Applied Cryptography, Protocols, Algorithms, and Source Code in C.* John Wiley & Sons, 1996. ISBN: 0-471-11709-9.

[3] HALLER, N.: *The S/Key One-Time Password System.* In.: ISOC Symposium on Network and Distributed Systems, 1994.

[4] HALLER, N. et al.: *A One-Time Password System.* RFC 2289, 1998.

[5] MENEZES, A., VAN OORSCHOT, P., VANSTONE, S.: *Handbook of Applied Cryptography.* CRC Press, 1996. ISBN: 0849385237.

[6] SIMPSON, W.: *PPP Challenge Handshake Authentication Protocol (CHAP).* RFC 1994, 1996.

[7] *Information Technology - Open Systems interconnection ....* ITU-T Recommendation X.509, 2000.

[8] MYERS, M. et al.: *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol.* RFC 2560, 1999.

[9] HOUSLEY, R. et al.: *Internet X.509 Public Key Infrastructure Certificate and CRL Profile.* RFC 2459, 1999.

[10] DIERKS, T., ALLEN, C.: *The TLS Protocol Version 1.0.* RFC 2246, 1999.

[11] HICKMAN, KIPP: *The SSL Protocol.* Netsdcape Communication Corp., 1995.

[12] WAGNER, D., SCHNEIER, B.: *Analysis of the SSL 3.0 protocol.* In.: The Second USENIX Workshop on Electronic Commerce, 1996.

[13] FRIER, A., KARLTON, P., KOCHER, P.: *The SSL 3.0 Protocol.* Netscape Communication Corp., 1996.

[14] RESCORLA, E.: *HTTP Over TLS.* RFC 2818, 2000.

[15] HOEPNER, P. (Ed.): *Study on PKI and Certificate Usage in Europe 2006.* Fraunhofer Institute FOKUS, 2006.

[16] CLARKE, R.: *The Fundamental Inadequacies of Conventional Public Key Infrastructure.* In: Proceedings of ECIS'2001 conference, Bled, Slovenia, 2001.

[17] BRANDS, S.: *Rethinking Public Key Infrastructures and Digital Certificates.* MIT Press, 2000. ISBN: 0-262-02491-8.

[18] PASHALIDIS, A., MITCHELL, C.: *A taxonomy of single sign-on systems.* In.: Information Security and Privacy, ACISP, 2003.

[19] CANTOR, S., KEMP, J.: *Liberty Bindings and Profiles Specification.* Liberty Alliance Project Specification, 2003.

[20] CANTOR, S., KEMP, J., CHAMPAGNE, D.: *Liberty Bindings and Profiles Specification.* Liberty Alliance Project Specification, 2003.

[21] MALER, E. et al. (Ed.): *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) v1.1.* OASIS, 2003.

[22] BAJAJ, S. et al.: *Web Services Federation Language (WS-Federation).* BEA, IBM, Microsoft, RSA Security, Verisign, 2003.

[23] BAJAJ, S. et al.: *WS-Federation: Passive Requestor Profile.* BEA, IBM, Microsoft, RSA Security, Verisign, 2003.

[24] ANDERSON, S. et al.: *Web Services Trust Language (WS-Trust).* 2005. http://specs.xmlsoap.org/ws/2005/02/trust/WS-Trust.pdf

[25] NADIN, A. (Ed.) et al.: *Web Services Security: SOAP Message Security 1.0.* OASIS, 2004.

[26] NANDA, A.: *Identity Selector Interoperability Profile V1.0.* Microsoft, 2007.

[27] BALLINGER, K. et al.: *Web Services Metadata Exchange (WS-MetadataExchange).* 2004. http://xml.coverpages.org/WS-MetadataExchange.pdf

[28] DELLA-LIBERA, G. et al.: *Web Services Security Policy Language (WS-SecurityPolicy).* 2005. http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-secpol/ws-secpol.pdf

[29] *OpenID Authentication 2.0 – Final*, 2007. http://openid.net/specs/openid-authentication-2_0.html

[30] RESCORLA, E.: *Diffie-Hellman Key Agreement Method.* RFC 2631, 1999.

[31] RESCORLA, E.: *HTTP Over TLS.* RFC 2818, 2000.

[32] BERNERS-LEE, T. et al.: *Uniform Resource Identifier (URI): Generic Syntax.* RFC 3986, 2005.

[33] REED, D. et al.: *Extensible Resource Identifier (XRI) Syntax V2.0.* OASIS, 2005.

[34] SLOT, M.: *Beginner's guide to OpenID phishing.* http://marcoslot.net/apps/openid/

[35] LAURIE, B.: *OpenID: Phishing Heaven*, 2007. http://www.links.org/?p=187

[36] CANTOR, S. et al.: *Shibboleth Architecture, Protocols and Profiles.* 2005. http://shibboleth.internet2.edu/shibboleth-documents.html

[37] HARDT, D.: *The Simple eXtensible Identity Protocol (SXIP) Reference.* 2004. https://sxip.net/archive/specs/sxip-reference.pdf

[38] HARDT, D.: *The Sxip Markup Language (SxipML).* 2004. https://sxip.net/archive/specs/sxipml-spec.pdf

[39] BARTEL, M. et al.: *XML-Signature Syntax and Processing.* W3C, 2002.

[40] ERNST, J.: *Light-Weight Identity*. NetMesh Inc., 2005. http://lid.netmesh.org/docs/NetMesh-LID.pdf

[41] *The GNU Privacy Guard.* 2005. http://www.gnupg.org/

[42] BELLOVIN, S. M.,: *Using the Domain Name System for System Breakin.* In.: 5th USENIX UNIX Security Symposium, 1995.

[43] ABELSON, H., LESSIG, L.: *Digital Identity in Cyberspace.* White Paper Submitted for 6.805/Law of Cyberspace: Social Protocols, 1998.

[44] PFITZMANN, A., KÖHNTOPP, M.,: *Anonymity, Unobservability, Pseudonymity, and Identity Management A Proposal for Terminology.* In.: Designing Privacy Enhancing Technologies, International Workshop on Design Issues in Anonymity and Unobservability, 2000.

[45] PFITZMANN, A., HANSEN, M.: *Anonymity, Unlinkability, Undetectability, Unobservability,Pseudonymity, and Identity Management –A Consolidated Proposal for Terminology v0.31.* 2008. http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.31.pdf

[46] WARREN, S., BRANDEIS, L.: *The Right to Privacy.* Harvard Law Review, Vol. IV, No. 5, 1890.

[47] SOLOVE, D.: *The Digital Person: Technology and Privacy in the Information Age.* NYU Press, 2004. ISBN: 0814798462.

[48] *Records, Computers and the Rights of Citizens.* Report of the Secretary's Advisory Committee on Automated Personal Data Systems, U.S. Department of Housing, Education, and Welfare, 1973.

[49] *Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data.* European Parliament and the Council of European Union, 1995.

[50] RESNICK, P. et al.: *Reputation Systems.* Communications of the ACM, 43(12), 2000.

[51] AXELROD, R.: *Evolution Of Cooperation.* Basic Books, New York, 1984. ISBN: 0465021220.

[52] WINDLEY, P. et al.: *Using reputation to augment explicit authorization.* In.: Proceedings of the 2007 ACM workshop on Digital identity management, 2007.

[53] SABATER, J., SIERRA, C.: *Review on computational trust and reputation models.* Artificial Intelligence Review, Vol. 24(1), 2005.

[54] FRIEDMAN, E., RESNICK, P.: *The Social Cost of Cheap Pseudonyms.* Journal of Economics and Management Strategy, Vol. 10, 2001.

[55] BERNERS-LEE, T.: *Information Management: A Proposal.* 1990. http://www.w3.org/History/1989/proposal.html

[56] FIELDING, R., et al.: *Hypertext Transfer Protocol – HTTP/1.1.* RFC 2616, 1999.

[57] FIELDING, R.: *Architectural Styles and the Design of Network-based Software Architectures.* University of California, Irvine, 2000. Dissertation.

[58] JACOBS, I., WALSH, N. (Ed.): *Architecture of the World Wide Web, Volume One.* W3C Recommendation, 2004. http://www.w3.org/TR/2004/REC-webarch-20041215/

[59] *Findings of the W3C Technical Architecture Group (TAG).* 2008. http://www.w3.org/2001/tag/findings

[60] DHAMIJA, R., TYGAR, J. D., HEARST, M.: *Why Phishing Works.* In.: Proceedings of CHI 2006 Conference on Human Factors in Computing Systems, 2006.

[61] FRANKS, J.: *HTTP Authentication: Basic and Digest Access Authentication.* RFC 2617, 1999.

[62] *Internet Usage Statistics, The Internet Big Picture.* 2008. http://www.internetworldstats.com/stats.htm

[63] DASGUPTA, P.: *Trust as a Commodity*. In: GAMBETTA, D. (Ed.): *Trust: Making and Breaking Cooperative Relations*, Blackwell Publishers, 1990. ISBN: 0631175873.

[64] FRIEDMAN, B., KAHN, P.H., HOWE, D.C.: *Trust Online.* Communications of the ACM, Vol. 43, No. 12, 2000.

[65] SHANNON, C.E.: *A Mathematical Theory of Communication.* Bell System Technical Journal, vol. 27, 1948.

[66] MOCKAPETRIS, P.: *Domain Names - Concepts and Facilities.* RFC 1034, 1987.

[67] ECONOMIDES, N.: *The Economics of Networks.* Brazilian Electronic Journal of Economics, vol. 1(0), 1997.

[68] CARPENTER, B. (Ed.): *Architectural Principles of the Internet.* RFC 1958, 1996.

[69] *RSS 2.0 Specification*. 2003.
http://cyber.law.harvard.edu/rss/rss.html

[70] GARRETT, J.: *Ajax: A New Approach to Web Applications.* 2005.
http://www.adaptivepath.com/ideas/essays/archives/000385.php

[71] RAGGETT, D., LE HORS, A., JACOBS, I. (Ed.): *HTML 4.01 Specification.* W3C Recommendation, 1999.
http://www.w3.org/TR/html401/

[72] BERNERS-LEE, T.: *What do HTTP URIs Identify?.* 2002.
http://www.w3.org/DesignIssues/HTTP-URI.html

[73] BERNERS-LEE, T.: *What HTTP URIs Identify.* 2005.
http://www.w3.org/DesignIssues/HTTP-URI2.html

[74] RAMAN, T.V. (Ed.): *On Linking Alternative Representations To Enable Discovery And Publishing.* 2006.
http://www.w3.org/2001/tag/doc/alternatives-discovery.html

[75] THOMPSON, H.S., ORCHARD, D.: *URNs, Namespaces and Registries.* 2006.
http://www.w3.org/2001/tag/doc/URNsAndRegistries-50

[76] BRAY, T. et al. (Ed.): *Extensible Markup Language (XML) 1.1 (Second Edition).* W3C Recommendation, 2006.
http://www.w3.org/TR/xml11/

[77] MENDELSOHN, H., WILLIAMS, S. (Ed.): *The use of Metadata in URIs.* 2006.
http://www.w3.org/2001/tag/doc/metaDataInURI-31-20061204.html

[78] BERNERS-LEE, T.: *Cool URIs don't change*. 1998.
http://www.w3.org/Provider/Style/URI.html

[79] DIERKS, T., RESCORLA, E.: *The Transport Layer Security (TLS) Protocol Version 1.2.* RFC 5246, 2008.

[80] BERNERS-LEE, T., et al.: *The Semantic Web.* Scientific American Magazine, May 17, 2001.

[81] KLYNE, G., CARROLL, J. (Ed.): *Resource Description Framework (RDF): Concepts and Abstract Syntax.* W3C Recommendation, 2004.
http://www.w3.org/TR/rdf-concepts/

[82] DOCTOROW, C.: *Metacrap: Putting the torch to seven straw-men of the meta-utopia.* 2001.
http://www.well.com/~doctorow/metacrap.htm

[83] BALAKRISHNAN, H. et al.: *Looking up data in P2P systems.* Communications of the ACM, Vol. 46, 2003.