Deficiencies of World Wide Web Architecture

Research Report

Author:
Radovan Semančík

Date:
November 2009

Version:
1.0

Abstract:
The World Wide Web is becoming global communication medium. However it was originally designed as a hyper-media system for static content. The original architecture of World Wide Web exhibits problems when applied to the current dynamic environment. This paper presents a model for describing human-computer interactions, which is used to define a new goals for World Wide Web architecture. The model and the definition of goals is used to evaluate current architecture of World Wide Web. Discovered inconsistencies and deficiencies of the architecture are described in detail. Identified problems originate from the vague meaning of WWW resource, the limited ability to express opinion about resource and to use such opinions to evaluate trustworthiness of information provided by the resource. The solution outline is provided is rough details, describing the conceptual ideas of one possible solution for described problems.

nLight, s.r.o.
Vendelínska 109
Lozorno, Slovakia
www.nLight.eu

Table of Contents

1 Introduction....1
2 Persona Model....2
3 Desired Environment....3
4 World Wide Web Architecture....5
4.1 REST Architectural Style....5
4.2 Resources and Identifiers....6
4.3 The Meaning of Resource....8
4.4 URI Aliases and QNames....9
4.5 Persistence....10
4.5.1 HTTP URIs....11
4.6 Security and Trust....12
4.7 Semantic Web....13
4.8 Hidden Assumptions of World Wide Web Architecture....14
5 Solution Outline....16
6 Conclusion....18
Bibliography....19

1 Introduction

World Wide Web (WWW) is the primary mechanisms for global-scale interactions of humans and computers. Since its spontaneous inception in 1990s it was continually improved to reflect both needs of the users and the desire for a clean architecture. The first major activity on the architectural grounds was a definition of REST architectural style [1] that was supposed to guide the evolution of World Wide Web. The second important milestone was a compilation of WWW architectural principles into a single document [2]. However, the current architecture of World Wide Web still exhibits several inconsistencies and deficiencies.

The purpose of this paper is to describe the problems of WWW architecture, especially the problems related to the nature of human-computer interactions. We examine the positioning of computer users as sources of information on World Wide Web, especially with the goal to understand how user reflect themselves in the environment of global network.

Basic outline of a model for evaluation of human-computer interaction is described in this paper. The model is focused on the correspondence of data records stored in computer systems and the real-world entities that they represent. The model is used to set the goals for future World Wide Web architecture, to evaluate the current architecture of World Wide Web and to identify problematic areas.

The core of this paper contains the description of architectural inconsistencies and deficiencies of World Wide Web architecture. These problems were discovered by reflection of new requirements and application of the model on the current World Wide Web principles. The evaluation is aimed at identification of basic problems that preclude the use of World Wide Web as a person-to-person communication medium, instead of using it only as a world-wide repository of static information (as it was originally designed).

Outline of a solution for the some of the identified architectural problems is provided. The outline roughly describe the conceptual changes to current architecture of World Wide Web that need to be made to at least partially solve describe architectural problems. However the solution proposal is far from complete and it is included in this paper only as an example, one possible approach to address the described problems.

2 Persona Model

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. The persona model was introduced in [3]. This paper provides brief introduction, updated terminology and extension of the persona model.

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. Computer monitor, keyboard, camera or independent sensor are examples of terminal devices. Neither realspace nor cyberspace entities are sure whether the terminal device operates as expected, as they cannot perceive the other world directly. Correlation of information from several terminal devices may increase the confidence in the information, however the reliability of the information always disputable.

Based on the discussion above we can formulate following statement:

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

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. 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.

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.

3 Desired Environment

The architecture of a system must take into consideration two principal aspects: the environment that it needs to be deployed to and the environment that it wants to create. Each system is created for the purpose of changing the current environment. The goal of the system might by to speed up business processes, support communication, increase efficiency. The desired environment, the environment that we seek to create must be understood before appropriate system architecture could be conceived. World Wide Web is yet another system and therefore it is important to discuss the effect that World Wide Web should induce before we can evaluate and improve its architecture.

The Internet and World Wide Web should support an environment of effective cooperation. Such environment should induce the positive network effect [4]. 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. 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 of the World Wide Web.

We believe that these features can be satisfied by creating an 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. It attempts to find the equilibrium between colliding forces and keep the system in that equilibrium by self-balancing mechanisms.

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.

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. We cannot expect that all Internet users will maintain a long-term relationship with all other Internet users. Therefore an 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 [5]. 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” [6] and it can influence current behavior of people by threatening to punish bad behavior in the future.

Some information on the Web may be in a self-validating, such as a set of instructions that can be immediately tested. But substantial fraction of the Web contains information in form of claims that cannot be instantly verified, opinions and promises. It is obvious that only a human user can determine the trustworthiness of information on the Web. However it is a substantial difference if user has to evaluate the trustworthiness of the information without any assistance as compared to evaluation in presence of appropriate visual clues. Computer system could provide such clues, for example visual indicator based on the reputation of information source.

We seek to design an architecture that will support positive network effect by creating an environment of responsibility. We expect that distributed reputation system might be a key mechanism to create such environment. We seek to design a system in which the user is ultimately responsible for evaluation of provided information. However the system should provide appropriate assistance (in the form of visual clues) to aid the user in evaluation of information that comes from unknown sources.

4 World Wide Web Architecture

World Wide Web originated in early 1990s as an distributed hyper-text system, based on Hyper Text Transfer Protocol (HTTP) and 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” [2]:

> *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).*

Following sections contain description of deficiencies of the architectural style, architecture, design and specifications that form current World Wide Web.

4.1 REST Architectural Style

The principles and protocols of current World Wide Web architecture have evolved during late 1990s. The architecture of World Wide Web was guided by the Representational State Transfer (REST) architectural style described by Fielding [1]. 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 [7] and AJAX [8].

The server component of the REST architecture is supposed to be stateless [1]:

> *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 it fails if the resources 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 [9]. 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 themselves. 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 [10], HTTP [9] and HTML [11] 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 [12]. 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.

4.2 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 [2]:

> *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 [2] 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. However according to the persona model such representation must be considered 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 even be harmful.

The only wide-spread security mechanism for the Web is HTTPS [12]. 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 of 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 [2]:

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

This problem was recognized [13] and a solution was proposed by the W3C Technical Architecture Group [14] 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 [10]:

> *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 [9]). 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. Such approach may provide cleaner separation of concerns and also a space for technology innovation.

From the orthogonality principles proposed by the WWW architecture document [2] it follows that the data formats used for resource representations should be independent from the URIs. However, the fragment segment of the URI (segment following the hash character) by definition depends on specific resource 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.

4.3 The Meaning of Resource

The definition of a resource is very vague. It is essentially defined as “whatever might be identified by a URI” [2]. This may lead to almost anything to be considered a resource. Even resource representations may 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 the URI 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 [14] 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 [2] 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 [15].

The situation may be partially addressed by using HTTP redirects [9], 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 [15].

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.

4.4 URI Aliases and QNames

The World Wide Web Architecture [2] 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 [2], however it is deemed acceptable by at least some members of W3C TAG [16]:

> *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 automated reputation system could not determine that the opinions about such URIs apply to the same resource. The method (protocol) used to access the resource 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) [17] 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 of 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 [2]:

> *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, because it is a complication of basic concepts. 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. A less generic URI format could provide universal method how to combine URI with a free-form name or even how to combine several URIs into a single URI. An example of such mechanism is a cross-reference used in eXtensible Resource Identifier (XRI) [18].

4.5 Persistence

World Wide Web Consortium (W3C) Technical Architecture Group (TAG) claims that the URIs using the *http* scheme support persistence. The draft finding of the TAG [16] contains following statement 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 [19] 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 the practice proposed by W3C TAG actually inhibits persistence of URIs.

W3C published a document [20] 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 [20] 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 [2]:

> *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 long-term persistent identifiers is not practical.

4.5.1 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. The syntax of HTTP URIs does not follow the same consistent set of rules. 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. 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 [16] 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 [2]

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 [16]. The specification of URI [10] 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 [9] 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. We consider any such mechanism to be an *addressing* mechanism rather than *identification* mechanism.

4.6 Security and Trust

The only practical security and trust mechanism for the World Wide Web is currently HTTPS [12]. It is a protocol providing channel security (confidentiality and integrity protection) and authentication of the connection endpoints. HTTPS is based on SSL/TLS [21] 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 practical 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 provides definition of information source (according to the persona model). But it does not provide any information about the reputation of the entity that published the information. It does not indicate 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). Therefore the user will likely be able to determine trustworthiness of an organization with which he maintains a long-term first-hand relationship, such as his bank. But the user will not be able determine a trustworthiness of information provided by arbitrary network site. However, to support the network effect and our target environment, the interactions should not be constrained to a handful of organizations that have close relationship with the user. Therefore a practical security and trust management system for the Internet should not operate on a direct black/white approach. It should indicate finer scale of “trust”, not just the two extremes of secure/trusted or insecure/untrusted.

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 paradigm of the “static Web” no longer applies. 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 [22]: 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. 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. The assignment of URIs to users may also be an advantage for deployment of distributed reputation system on the Internet scale.

4.7 Semantic Web

The semantic web [23] 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 [24]. 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 [13]. 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 [25] 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 and therefore the problems does not fully manifest themselves. The computers cannot judge the reliability of information just by themselves, therefore the problems are magnified in the environment of computer-to-computer interactions of semantic web. However the problems of World Wide Web and Semantic Web are the same or at least similar, therefore the solutions proposed for World Wide Web may also be applicable to Semantic Web.

4.8 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.

The WWW architecture assumes that the information on the Web will be used be intelligent subjects that can judge the reliability of the information without any assistance. While this may be true for a part of Internet users, the spread of hoax mails and mis-information indicates that it definitely does not apply to all the users. The situation is even more difficult if we assume existence of semi-intelligent software agents, for example for a purpose of

semantic web. Such agents will probably not be able to judge trustworthiness of the information without additional meta-data.

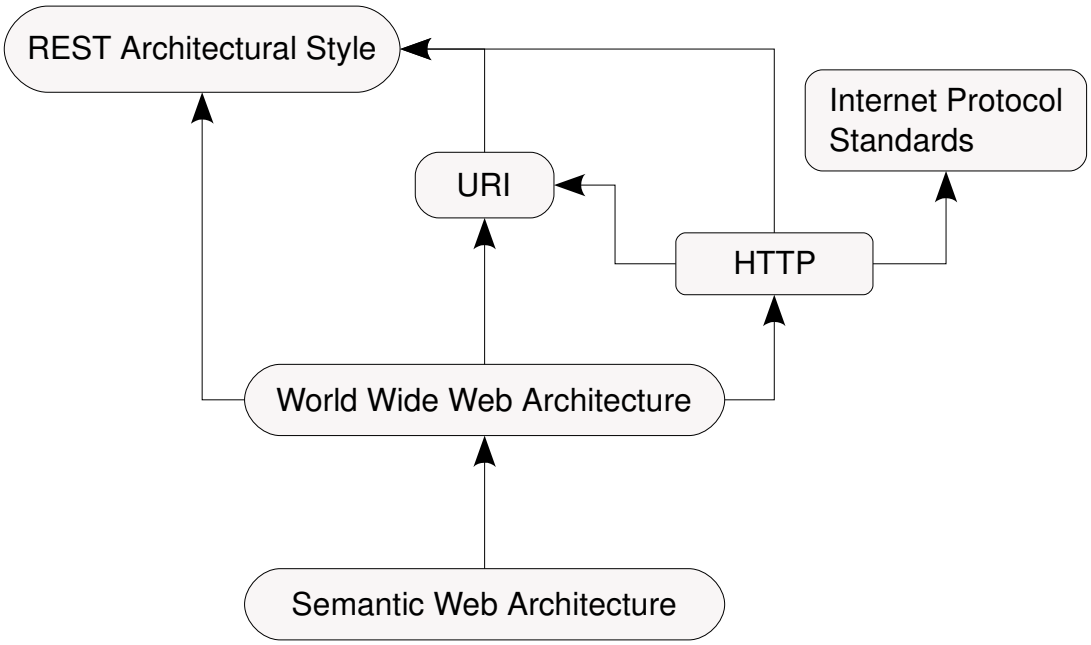

Figure 1: Current World Wide Web architecture diagram

5 Solution Outline

The current architecture of the World Wide Web is far from perfect. The simplified diagram of World Wide Web architecture is provided in figure Figure 1. The arrows on the diagram indicate dependency, the roundness of corners indicate relative level of abstraction. 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 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 2):

- **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. The architectural styles of REST or its alternative should belong to this layer.
- **World Wide Web Architecture** is a set of architectural constraints, rules and recommendations that define basic principles of World Wide Web operation. These

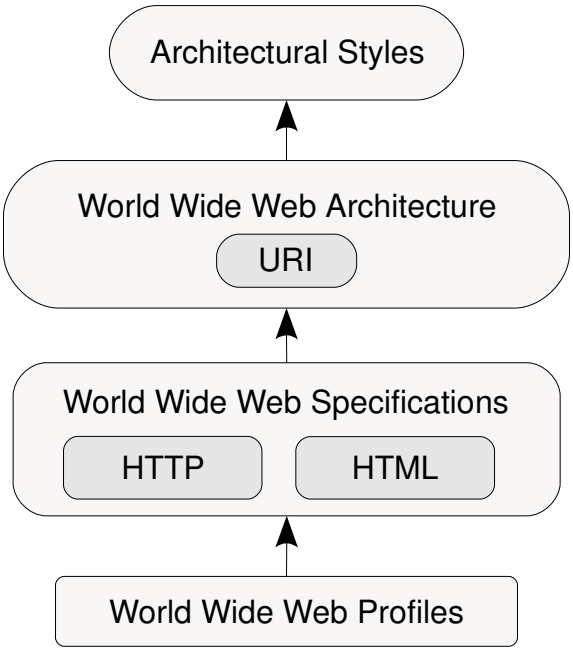

Figure 2: Abstraction layers of proposed WWW architecture

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. The fundamental concepts of World Wide Web such as URI belong to this layer.

- **Protocol Specifications** provide specific definitions of communication protocols, data formats and interfaces. These specifications are based on the architecture of World Wide Web, constraining the architectural principles 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. The specification of HTTP and HTML belong to this layer.
- **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 may define 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.

The World Wide Web architecture should consider any information on the Web to be just an subjective opinion. It should link all resources to their source (author). Therefore it may be an advantage to represent sources (authors) by resources, identified by URIs. Such “persona” resources could pride information about the author, reputation sources for determination of trustworthiness and so on. User's browser should include clients for distributed reputation system. The level of security for the content displayed in a browser should be combined with the reputation information of the content source. Therefore the user's will be provided with visual clues that can assist them with evaluation of trustworthiness of displayed information.

6 Conclusion

Basic model of realspace-cyberspace interactions was provided in this paper. It was observed that all realspace-cyberspace interactions are subjective. The evaluation of the source of cyberspace information is needed in order to decide whether it is useful or relevant. The model was used as a guideline to define new goals for World Wide Web architecture. The new World Wide Web should support ad-hoc interactions and positive network effect, without the need to constraint interactions to channels.

The model and the discussion of architectural goals were used for evaluation of current architecture of World Wide Web. We have identified following major problems:

- The subjectivity of resource representations is not addressed by the WWW architecture. It is assumed that the source of data transmission is also the source of data. Such assumption may not necessarily hold.
- 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.
- Vague meaning of the resource. It is not clear what a resource represents, therefore it may be difficult to implement mechanisms that rely on the meaning, such as rating and reputation mechanisms.
- 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.

An outline of a solution to address described problems was provided. We recommend to make following changes to WWW principles:

- Divide World Wide Web architecture into four well-defined layers of abstraction.
- Regard resources to be cyberspace entities, consider them always subjective.
- Introduce concept of resource source, a cyberspace representation of the subject that authored the resource. Resource source can itself be a resource, identified by URI.

The provided solution is not complete and does not address all the mentioned problems. It is only an outline describing principles of a more complete solution. Future work need to focus on finishing the solution details. A solution of WWW architectural issues will most probably require change in the WWW principles, architecture and protocols. The green-field approach is obviously not feasible, therefore any practical solution must be based on the existing state of World Wide Web. Such solution will require cooperation of scientific community with standard bodies and will take considerable time to implement. Careful review and improvement of the WWW architectural principles is therefore a crucial part of the solution.

Bibliography

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

[2] *Architecture of the World Wide Web, Volume One*. 2004. http://www.w3.org/TR/2004/REC-webarch-20041215/

[3] Semančík, R.: *Basic Properties of the Persona Model*. Computing and Informatics, Vol. 26, 2007.

[4] Economides, N.: *The Economics of Networks*. Brazilian Electronic Journal of Economics, vol. 1(0), 1997.

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

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

[7] *RSS 2.0 Specification*. 2003. http://cyber.law.harvard.edu/rss/rss.html

[8] Garrett, J.: *Ajax: A New Approach to Web Applications*. 2005. http://www.adaptivepath.com/ideas/essays/archives/000385.php

[9] Fielding, R., et al.: *Hypertext Transfer Protocol -- HTTP/1.1*. RFC 2616, IETF, 1999.

[10] Berners-Lee, T., et al.: *Uniform Resource Identifier (URI): Generic Syntax*. RFC 3986, IETF, 2005.

[11] *HTML 4.01 Specification*. 1999. http://www.w3.org/TR/html401/

[12] Rescorla, E.: *HTTP Over TLS*. RFC 2818, , 2000.

[13] Berners-Lee, T.: *What do HTTP URIs Identify?*. 2002. http://www.w3.org/DesignIssues/HTTP-URI.html

[14] Berners-Lee, T.: *What HTTP URIs Identify*. 2005. http://www.w3.org/DesignIssues/HTTP-URI2.html

[15] Raman, T.V. (Editor): *On Linking Alternative Representations To Enable Discovery And Publishing*. 2006. http://www.w3.org/2001/tag/doc/alternatives-discovery.html

[16] Thompson, H.S., Orchard, D.: *URNs, Namespaces and Registries*. 2006. http://www.w3.org/2001/tag/doc/URNsAndRegistries-50

[17] *Extensible Markup Language (XML) 1.1 (Second Edition)*. 2006. http://www.w3.org/TR/xml11/

[18] Reed, D., et al.: *Extensible Resource Identifier (XRI) Syntax V2.0*. Committee Specification, OASIS, 2005. http://docs.oasis-open.org/xri/V2.0

[19] Mendelsohn, H., Williams, S. (Editors): *The use of Metadata in URIs*. 2006. http://www.w3.org/2001/tag/doc/metaDataInURI-31-20061204.html

[20] Berners-Lee, T.: *Cool URIs don't change*. 1998. http://www.w3.org/Provider/Style/URI.html

[21] Dierks, T., Rescorla, E.: *The Transport Layer Security (TLS) Protocol Version 1.2*. RFC 5246, IETF, 2008.

[22] Franks, J.: *HTTP Authentication: Basic and Digest Access Authentication*. RFC 2617, IETF, 1999.

[23] Berners-Lee, T., et al.: *The Semantic Web*. Scientific American Magazine, May 17, 2001.

[24] Klyne, G., Carroll, J. (Editors): *Resource Description Framework (RDF): Concepts*

and Abstract Syntax. . http://www.w3.org/TR/rdf-concepts/

[25] Doctorow, C.: *Metacrap: Putting the torch to seven straw-men of the meta-utopia*. . http://www.well.com/~doctorow/metacrap.htm