Abstract

The Virtual World Region Agent Protocol (VWRAP) defines interactions between hosts collaborating to create a shared, internet scale virtual world experience. This document introduces the protocol, the objectives and requirements it imposes on hosts implementing the protocol. To the extent it affects protocol interactions, this document describes the model assumed by the protocol. A document roadmap is included that briefly describes the contents of other documents produced by the VWRAP Working Group.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 6, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.
Internet-Draft        VWRAP: Introduction and Goals            July 2010

Table of Contents

1.  Introduction and Motivation .......................... 4
   1.1.  Requirements Language .......................... 4
2.  Introducing the Virtual World Region Agent Protocol ........ 4
   2.1.  A Brief Introduction to Virtual Worlds ........... 4
   2.2.  Architectural Objectives and Requirements .......... 6
3.  Virtual World Region Agent Protocol Architecture ........ 8
   3.1.  Protocol Objectives ................................ 8
   3.2.  Structural Architecture and Services ............... 11
      3.2.1.  The Client Application ....................... 12
      3.2.2.  Services ................................... 12
      3.2.3.  Deployment Patterns .......................... 13
   3.3.  Architectural Elements ............................ 14
      3.3.1.  Communicating Application State Using REST-Like
              Resource Accesses ............................. 15
      3.3.2.  Bi-Directional Messaging with the VWRAP Event
              Queue ....................................... 15
      3.3.3.  Using Capabilities to Simplify Access Control
              Between Trust Domains .......................... 16
      3.3.4.  Using Flexible Format Descriptions to Avoid
              Version Skew .................................... 17
4.  Services Defined by the Virtual World Region Agent Protocol .18
   4.1.  User Authentication ................................ 18
   4.2.  Presence in the Virtual World ....................... 20
      4.2.1.  Establishing Presence with the Region Domain .... 20
      4.2.2.  Moving Presence ................................ 21
   4.3.  User and Group Messaging ............................ 22
      4.3.1.  Spatial Messaging .............................. 22
      4.3.2.  User to User and User to Group Messaging ........ 23
   4.4.  Digital Asset Access and Manipulation ............... 23
      4.4.1.  Manipulating Digital Assets .................... 24
      4.4.2.  Establishing Presence for Digital Assets ....... 25
5.  Document Roadmap ........................................ 26
6.  IANA Considerations ...................................... 29
7.  Security Considerations ................................ 29
   7.1.  Capabilities ...................................... 29
   7.2.  User Authentication ............................... 30
   7.3.  Agent Domain to Region Domain Authentication ....... 30
   7.4.  Access Control for Digital Assets .................. 30
8.  References .............................................. 31
   8.1.  Normative References .............................. 31
8.2. Informative References . . . . . . . . . . . . . . . . . . . . . 31
Appendix A. Definitions of Important Terms . . . . . . . . 32
Appendix B. Acknowledgments . . . . . . . . . . . . . . . . . . . . 33
Authors’ Addresses . . . . . . . . . . . . . . . . . . . . . . . . . 33
1. Introduction and Motivation

Virtual Worlds are of increasing interest to the internet community. Innumerable examples of virtual world implementations exist; most using proprietary protocols. With roots in games and social interaction, virtual worlds are now finding use in business, education and information exchange. This document introduces the Virtual World Region Agent Protocol (VWRAP) suite. This protocol suite is intended to carry information about a virtual world: its shape, its residents and objects existing within it. VWRAP’s goal is to define an extensible set of messages for carrying state and state change information between hosts participating in the simulation of the virtual world.

Virtual worlds differ in their capabilities and architectural features. The VWRAP protocol is not appropriate for all massively multi-participant experiences. This document provides an introduction to virtual worlds and defines characteristics of virtual worlds VWRAP explicitly supports. It also describes the specific objectives of the protocol suite. An overview of the protocol is included, including an architectural description and list of services defined in a typical virtual world deployment. A guide to the documents proposed by the VWRAP working group is provided in an effort to describe which protocol features are defined in which working group deliverable.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Introducing the Virtual World Region Agent Protocol

2.1. A Brief Introduction to Virtual Worlds

At its most basic level, the virtual world mirrors the reified world. It is inhabited by people and contains objects. Objects and people have a distinct place in the world and respond to external forces. The social construction of the virtual world is also similar to the reified world. People meet and interact with others to complete work tasks or for simple enjoyment. People converse, share media, and even sing to each other. A virtual world may allow commerce or enable building of virtual assets. Nearly the complete range of human interaction can be replicated in the virtual world. Properly rendered, an experience in a virtual world can carry the same impact as one in consensus reality.
To be relevant to the participant’s experience, virtual worlds must retain characteristics of the "real world." Objects in the virtual world are models of familiar shapes and textures, represented using a common data format so they may be easily processed by the human visual cortex. At the same time they must carry sufficient information to be consumed by participants with visual impairments or to be processed by automated systems. Though the virtual world’s state is maintained by abstract collections of data, it is rendered as recognizable (though occasionally fantastical) physical objects.

But virtual worlds are not completely faithful mirrors of the world our physical bodies inhabit. Virtual worlds are not limited by distance. With appropriate network connectivity, users may interact even if they are on opposite sides of the earth. Virtual worlds also allow participants to "play" with physical constraints. They provide the subjective experience of things not possible in consensus reality: participants can fly, the effects of "death" are temporary, users may call items into existence, examine the International Space Station with friends, examine DNA codon by codon with coworkers, or experience with a works of interactive art.

Virtual worlds inherit the intellectual property characteristics of other digital media. Implementers wishing to replicate the economy of scarcity present in the reified world must take special measures to avoid or limit the ramifications of unauthorized content duplication.

The VWRAP suite assumes network hosts, likely operated by distinct organizations will collaborate to simulate the virtual world. It also assumes that services originally defined for other environments (like the world wide web) will enhance the experience of the virtual world. The virtual world is expected to be simulated using software from multiple sources; interoperability standards are essential for delivering a robust services and compelling user experiences. VWRAP provides these interoperability standards. It may be used with large arrays of hosts simulating a vast virtual world, or small worlds operated for the benefit a few persons. It defines a trust model so hosts from different organizations may limit access to sensitive information.

VWRAP presupposes a virtual world with the following characteristics:

1. The virtual world exists independent of the participating clients.

   This is in contrast to systems that dynamically create virtual environments for specific social or task-oriented simulation. VWRAP assumes the state virtual world is "always on" and does not
require a specific protocol to establish new virtual worlds.

2. Avatars have a single, unique presence in the virtual world.

Users are represented in the virtual world by a digital representation called an avatar. A user’s avatar has an existence that mirrors the common physical world. Like people, avatars exist in only one place at one time. Avatars have a single identity that persists between user sessions. This identity may be used to render user-specific avatar shape or as the basis for access control.

3. The virtual world contains persistent objects.

Objects in the virtual world are governed by a "rational" life-cycle. They are created, persist and are (optionally) destroyed.

2.2. Architectural Objectives and Requirements

The overall objective of the VWRAP suite is to enable a stable, extensible and secure virtual world. The protocols defined by this working group do not mandate a specific system or network architecture. However, previous implementations of large distributed systems can yield clues as to how VWRAP compliant systems will be deployed. Documents produced by this working group reflect the consensus view of best common practice for producing large-scale, loosely-coupled distributed systems. VWRAP does not require a specific architecture, but its protocols are defined with a specific series of architectural patterns in mind.

These architectural patterns have the following objectives in mind:

1. Systems implementing virtual world services must be distributed.

Virtual worlds are inherently distributed systems. They are collaborative tools intended to reduce the effect of distance on meaningful social, educational and commercial interaction. Typical virtual worlds are envisioned as drawing services from network hosts operated by a wide array of organizational entities. Virtual worlds may also be large enough that a single system or cluster of systems is not capable of providing all services to each client.

Experience with previous virtual world technologies suggests systems implementing the virtual world assume services they depend on are provided by distinct hosts in different administrative domains. The virtual world, its user base and the constellation of virtual objects they use are simply too vast and
varied to assume a single system or a single provider.

This does not mean, however, that virtual worlds simulated by the VWRAP suite must exist on more than one system. Quite the contrary, all VWRAP components (including the user agent) could operate on a single system. But there is little reason to believe the protocol used to exchange information between virtual world services must differ between small and large systems.

Large virtual worlds with many users are certain to implement virtual world services differently than small ones. But the protocol used to communicate between systems implementing the virtual world can be the same. To be sure, optimizations may be possible for smaller deployments. Implementers of these "small" systems may find it useful to utilize standard, unoptimised protocols to allow for the possibility of growth.

But however large (or small) a virtual world deployment is, or how many distinct organizations contribute to its operation, software implementing virtual world services MUST assume resources required to perform its function are distributed amongst multiple hosts.

2. Services supporting collaboration are hosted on ‘central’ systems.

The "authoritative state" of the virtual world is stored on ‘central’ servers. Clients use protocols described in the VWRAP suite to access this state information (or to notified of state changes.) This is in contrast to some virtual world and Massively Multi-Participant systems in which state may be distributed. This is in contrast to ‘client authoritative’ or ‘co-simulation’ architectures in which each client is required to perform physics simulation for objects within view of the user’s avatar.

3. Virtual world services default to being ‘open’.

The VWRAP suite does not describe protocols for exclusive use within a "walled garden." Implementers and deployers are given the freedom to identify information sources and to determine which entities on the network they will trust. Service deployers may limit which users may access their services to a small set of individuals, employees of a particular company, residents of a particular country, users of a particular website. Or they may impose no restrictions whatsoever. It is entirely up to the service operator.
These same operators may require virtual assets to be hosted on their own servers, or they may identify a list of trusted asset servers operated by other organizations. Or they may impose no restrictions whatsoever. The decision is entirely up to the service operator.

The VWRAP suite describes protocols that are by default, "open." There is no requirement in these specifications that arbitrary bindings between services be enforced. In other words, requiring two services (e.g. - physics simulation and asset storage) be managed by the same organization’s servers is an issue of local policy, not of protocol.

3. Virtual World Region Agent Protocol Architecture

3.1. Protocol Objectives

The primary objective of the Virtual World Region Agent Protocol is to provide a stable, extensible, secure system for virtual world information interchange with the following characteristics:

1. Identity is universal.

Many network services are provided anonymously; the nature of the service does not require identity authentication prior to its use. But with the increasing deployment of customizable services delivered on the internet, identity is increasingly important. Even services that contain information that might not be considered "sensitive" require a representation of digital identity if for no other reason than to match service requests with user preferences. For example, a web page presenting current weather information may be enhanced by remembering locations of interest to each user. Recent work with "web mash ups," where multiple personalized or v sensitive resources are used in concert with one another points to the utility of a "universal" identity. The representation of this universal identity enables independent services to cooperate to present the facade of a unified application to the service consumer. This allows service aggregators to more easily integrate "best of breed" services into a consistent solution.

Universal identity is critical to the virtual world. To achieve an internet scale virtual world, user services must be distributed amongst multiple hosts. To achieve a compelling experience, it must be easy for service providers to deliver their services in the virtual world. To facilitate a compelling social experience in the virtual world, all users must have the
ability evaluate identity information of other users. Domains responsible for virtual world simulation MUST use a consistent representation of identity across all their hosts; simulation would otherwise be uncoordinated. Service providers who deliver content into the virtual world MUST use a consistent representation of identity to maintain the persistence of the virtual manifestation of their service; virtual objects used in conjunction with these services might otherwise appear to change state without apparent cause. Users depend on the persistent, universal identity of other users; if an avatar’s identity changed unexpectedly, the result would be a suboptimal virtual world experience.

2. Presentation of protocol data is in a flexible fashion.

While the primary purpose of the virtual world is to simulate a physical or social space, the tools used to access objects in the virtual world may be varied. Using a "3d viewer" is the primary mode of interaction with the virtual world, but other tools may be better suited for some tasks. For instance, it may be easier for a user to use a web browser to review avatar profile information, or to change details of virtual objects. Further, virtual world "mash ups" may prove to be important to some communities. To support the web (where XML [XML2008] and JSON [ECMA262] are the lingua franca of information exchange) while also supporting tools where binary encodings are more appropriate, VWRAP was designed to be "presentation neutral."

VWRAP resources are intended to be accessed via HTTP [RFC2616] or HTTPS [RFC2818]. In some circumstances, however, there may be advantages to accessing resources using other transports. HTTP(S) is well suited when request-response plus meta-data semantics are desired. For instance, event streams may be better suited by using XMPP [RFC3920] while time sensitive messages may be better carried over RTP [RFC3550]. Except where noted, implementers should assume that resources are accessed using HTTP or HTTPS.

VWRAP protocol exchanges are described in terms of an abstract type system [I-D.ietf-vwrap-type-system] describing the content of virtual world resources in a language and transport neutral fashion. An interface description notation allows implementers and service designers to describe the flow of abstract data between protocol participants. Abstract structured data types and information flows are reified using serialization rules most appropriate for the application. Web-based applications may choose to use JSON or XML. Server-to-server interactions may use the VWRAP specific binary serialization scheme if implementers...
and deployers view binary encoding to be advantageous. The
decision of which serialization scheme to use is ultimately that
of the system implementer. VWRAP has been designed to provide
this flexibility to system implementers and those tasked with
deploying VWRAP compatible systems.

3. Separation of concerns and easy extension is the norm.

VWRAP has been designed to allow meaningful separation of
concerns. In other words, changes in one part of the protocol
should not appreciably affect other parts.

For example, the authentication portion of the protocol is
independent of the part of the protocol that deals with instant
messaging or instantiating objects in the virtual world. In
addition to defining messages for communicating application
state, the specification also defines pre- and post-conditions.
Should one particular authentication scheme be found to be
lacking, it can be modified or replaced without affecting other
systems.

This type of separation of concerns in the protocol specification
also makes it easy to deploy "related solutions." While VWRAP
was designed primarily to communicate the state of the virtual
world between servers and client applications, a number of
related applications also exist. E-Commerce web sites related to
the virtual world and mobile chat clients allowing instant
messaging between mobile networks and virtual world participants
are just two examples of such applications. Proper separation of
concerns allows new services to be specified and deployed without
the need to redefine existing protocol.

4. The system exhibits resilience in the face of version skew.

Core to the VWRAP protocol is the idea that different components
and services may be operated by different administrative
entities; identity management services might be operated one
business while simulation services are operated by another. In
environments where many different organizations participate,
version skew can be an important concern. VWRAP was designed to
"degrade gracefully" when two systems running different versions
of the protocol attempt to communicate.

VWRAP uses a flexible abstract type system and interface
description notation to describe the structure and type semantics
of elements in messages sent between systems. Because this type
system makes extensive use of variable width, clearly delineated
data fields, services which consume protocol messages may
identify and extract only those message elements they know how to handle. While this is not a guarantee that message semantics may be preserved in all version skew situations, it does eliminate one important cause of interoperability failures.

3.2. Structural Architecture and Services

The Virtual World Region Agent Protocol assumes services on many different network hosts will collaborate to simulate the virtual world. Each service is responsible for a particular task: maintaining agent presence information, physics simulation, relaying text and voice chat information, etc. Services themselves are comprised of resources whose attributes are accessed by way of RESTful patterns at unique addresses. Because individual resources are accessed using their address, they are often referred to as "protocol endpoints."

Figure 1 below is intended to communicate the following concepts regarding the structural architecture of VWRAP:

- VWRAP uses a client-server architecture.
- There are likely to be a multitude of user agents.
- There are likely to be several instances of each defined service (i.e. there is no single, "blessed" authentication or asset service.)
- Services may be clients of other services (i.e. it’s not only user agents that are clients.)
- There is no obligation that each client establish a relationship with each type of service. (e.g. it’s okay to authenticate, then communicate exclusively with a chat server or an asset server.)
3.2.1. The Client Application

VWRAP presumes the virtual world is simulated for the benefit of human users. Whether that human is operating a "viewer" application to render the virtual world, or using a web interface to perform routine maintenance tasks, the user is expected to be operating software outside the administrative control of organizations offering virtual worlds services. VWRAP makes no assumptions about client software save it adheres to the described protocol.

It is important to remember that there are likely to be more types of client applications than just 3d virtual world renderers. Some client applications, termed "user agents," will be intended to be operated directly by human beings for their benefit. Other client applications may be automated services: search and indexing, backup and restore, etc.

3.2.2. Services

"Services" are loosely-coupled collections of RESTful protocol endpoints, each providing access to the internal state of applications simulating the virtual world. Services are operated by an "End Entity," who is responsible for it’s proper operation. End entities will become more important when discussing the VWRAP trust model. There is no requirement that the protocol endpoints that comprise a service be meaningfully related, but it does tend to make the system more understandable. This is to say, the VWRAP suite makes no attempt to enforce a policy that services be comprised of related protocol endpoints.

Implementers and deployers should also note that there is no
requirement that all endpoints in a service be operated on network hosts administered by the same end entity. In other words, it is perfectly acceptable for a "service" to be comprised of resources whose access addresses map to hosts from multiple organizations. Implementers should never assume that one trusted endpoint in a service implies that all endpoints in that service are trusted.

3.2.3. Deployment Patterns

The VWRAP suite assumes that the services that simulate the virtual world will be hosted on multiple network hosts. Client applications that render the virtual world for end users are assumed to be on distinct hosts as well.

Figure 2: Protocol Flows Between Organizations

Figure 2 above demonstrates a typical virtual world deployment. In it we see multiple client applications connecting to several servers operated by multiple organizations. Each "service host" in this diagram exposes an interface to a collection of resources representing the state of objects in the virtual world. It is traditional to discuss related resources as exposing "an interface" while related interfaces comprise "a service." Typical services are listed in a different section.

There is no restriction on how services map to hosts. Deployments where all resource interfaces are hosted on a single host is just as
valid as one where resources are spread between a wide array of systems. Client applications should support both. Resource interfaces are addressed using URLs. Clients should treat the URLs for resources as opaque resource locations.

The VWRAP suite makes extensive use of capabilities. Provisioning of the initial "seed capability" is described in the document describing the VWRAP trust model and user authentication techniques [I-D.ietf-vwrap-authentication]. The standard technique for "logging in" involves querying the authentication service for the location of other services. Implementers should note there are no restrictions on the location of these subordinate services. They may be implemented on the same host as the authentication service, or in the same domain, or they may be hosted by a completely different organization.

Operators of authentication services SHOULD configure their systems to only provide clients with service locations operated by trusted entities.

Implementers should note that client applications may be configured to allow end users to ignore the authentication service and directly provision their client applications to use explicit service locations. Client application developers may wish to consider offering such an option. Developers and operators of virtual world services MUST assume the client may provide them with data from untrusted services.

3.3. Architectural Elements

VWRAP utilizes a number of "architectural motifs" or recurring design patterns. Most notably they include:

- exposing application state via RESTful resources
- using URIs [RFC3986] to represent the address of application resources
- using HTTP(S) to transport message oriented protocol data
- using XMPP to transport data appropriately modeled as event streams
- using (S)RTP to transport time-sensitive messages
- defining application resources using reified formats for abstract type and interface description notations
o using an abstract type system to define access semantics of fields in protocol messages

o using multiple reification schemes to "serialize" abstract structured data (defined serializations include XML, JSON and Binary.)

3.3.1. Communicating Application State Using REST-Like Resource Accesses

Not all virtual world interactions must be real-time exchanges. Many common activities like user authentication and texture and object transfer do not require "real time" semantics in the same way that applications like video-conferencing and Voice Over IP (VOIP) do. While it is generally a better experience if textures download quickly, if they are delayed, it does not have the same ramifications as if a voice packet in a VOIP system were delayed. Additionally, some interactions with the virtual world are strongly reminiscent of the request / response semantics used by popular protocols (like HTTP, POP3, etc.)

Because many protocol exchanges in the virtual world may be represented as non-real-time request / response interactions, VWRAP "reuses" the messaging semantics of HTTP. The justification for this is simple. Were VWRAP to not use HTTP, many of the features of HTTP would need to be re-invented or at least re-specified. Features like the use of mime types to identify payload structure; the use of message headers to modify the request or response and the use of URIs to address and identify resources. HTTP also has the benefit of being well supported by tools vendors and well understood by manufacturers of networking equipment.

VWRAP messages that communicate request / response style messages flow between clients and servers, use HTTP(S) as a message transport. Message flows that are better modeled using an event stream MAY use XMPP as a transport. Time sensitive messages MAY use RTP as a transport. In each case, application objects representing the application state expose a RESTful interface and are addressed unambiguously using URIs.

3.3.2. Bi-Directional Messaging with the VWRAP Event Queue

Not all protocol interactions are easily represented by HTTP’s request / response semantics. When the server has a message for the client, there is no widely deployed technique for the server to initiate a HTTP request to the client. It is interesting to note that this is the same problem developers of "rich web applications" see when deploying their applications. Though VWRAP is not targeted
for implementation exclusively in web browsers, we can utilize some of the techniques common in COMET applications.

Work is ongoing to define a general solution for "reverse HTTP," but many of these solutions require the definition of new protocol and deploying new code to web servers. The current best practice for COMET-style interaction is the use of the "long poll."

To avoid "technology lock in," VWRAP defines an Event Queue abstraction that represents the flow of messages from the server to the client. The Event Queue is expected to be implemented using the long poll technique. When additional options such as Reverse HTTP or web sockets are specified and in general deployment, the Event Queue may be re-implemented using these techniques.

### 3.3.3. Using Capabilities to Simplify Access Control Between Trust Domains

Simulated objects and services delivered by VWRAP compliant systems will require some level of access control. Distributed access control is a notoriously difficult problem, however. VWRAP seeks to minimize the drawbacks of distributed access control by use of capabilities. In this context, a capability is an opaque URL, some portion of which contains a securely generated, cryptographically unguessable sequence of digits. Capabilities are used to define service endpoints and are intended to only be in the possession of trusted parties.

For example, a system may export the capability:


This URL defines the protocol endpoint used to communicate application state changes (or query application state) for a specific application object by a specific user (or delegate.)

Capabilities are required to be effectively unguessable as they represent the right to perform a set of operations on a particular resource. Additionally, they must be kept "secret." While the task of maintaining the confidentiality of a number of web resource addresses may be a burden, it does have the advantage of simplifying access delegation. If a subject wishes to delegate access to a third party, they simply communicate the capability.

To reduce the likelihood of successful guessing attacks, inadvertent disclosure of a capability and "time of check, time of use" attacks, capabilities in VWRAP have a fixed lifetime, after which they expire. Systems SHOULD pick capability lifetimes commensurate with their
security requirements and MUST NOT respond to protocol requests
directed at a capability’s URL after it has expired. Additionally,
VWRAP capabilities may be "single use" or "one shot," meaning that
they may only be used once before expiring.

Because capabilities are randomly generated with a short lifetime,
VWRAP defines a mechanism for securely communicating capabilities and
re-requesting expired capabilities.

It is important to note that capabilities do not completely replace
traditional access control models. Systems may still use traditional
Subject-Permission-Object schemes to represent access control for
objects under their control. Capabilities provide a mechanism for
communicating access control decisions among loosely coupled trusted
systems.

3.3.4. Using Flexible Format Descriptions to Avoid Version Skew

It is a common practice in large, complicated software systems to
divide the system into smaller, more manageable pieces. The precise
nature of this partitioning is beyond the scope of this protocol.
Practical experience has demonstrated that services distributed
across multiple cooperating hosts must contend with the issue of
version skew. Simply stated, version skew is the condition where
multiple versions of a service are interoperating simultaneously.

There are many reasons why version skew may be introduced. In VWRAP,
services may be operated by different organizations with different
deployment schedules. Or perhaps one particular service operator is
required to support an obsolete version of a particular service
endpoint for a small number of customers. Whatever the cause of
version skew, it has, in the past introduced difficulties in
deploying distributed services.

VWRAP does not seek to eliminate version skew, but attempts to reduce
its impact. VWRAP services are defined in using an abstract
interface description notation. It defines the type semantics of
fields inside a protocol message using an abstract type system. Each
of the types defined in its abstract type system has a default value
and common conversions between conformable types are defined. The
protocol also defines systems for serializing protocol messages prior
to transmission across the network. Each of serialization scheme
renders protocol messages into a collection of variable length
fields. Protocol content is identified by JSON syntax, binary tags
or XML element semantics, not by its position in the message. The
abstract type system and interface notation does not support the
concept of a "required field." If a field defined in a protocol
interaction is not present in the serialized message, it is
semantically equivalent to the field being present and containing the default value for the field’s type.

Careful construction of application resources allows them to operate without fear version skew induced format differences may cause the semantics of the message to be unclear. If a message arrives at a service endpoint with extra fields (fields defined in a later revision of the protocol exchange), the consumer can still extract those fields it understands. If a message arrives lacking a field described in the protocol exchange, the service endpoint SHOULD interpret it as if the field was present and contained the default value for its type. This implies the message consumer cannot depend on the format of the message to determine validity, but must examine the contents of the message, converting missing fields to present fields with default values, and then determine if sufficient information is present to imply semantics about the protocol exchange.

This technique will not eliminate all ramifications of version skew, but carefully constructed service descriptions should be able to avoid the most common problems found when services interoperate with minor revision differences. While the Virtual World Region Agent Protocol itself does not mandate this style of message interpretation, it does require that messages be constructed so that service endpoints may do so.

4. Services Defined by the Virtual World Region Agent Protocol

4.1. User Authentication

User Authentication in the Virtual World Region Agent Protocol is intended to verify the user’s authorization to control their avatar in the virtual world and associated services. VWRAP currently defines three methods for authenticating a user, as well as recommendations for integrating some third party authentication schemes. The inputs to authentication are an avatar or account identifier and a related authentication token. Assuming the token is successfully authenticated, the output of authentication is a seed capability or "seed cap."

Like many VWRAP protocol exchanges, authentication protocol data is represented as serialized data carried over a secure HTTPS transport. The use of TLS with VWRAP authentication is recommended for all deployers who do not employ some other network security scheme (IPSec, link encryption, etc.) Implementers are advised that in addition to user passwords and other credentials, the seed capability returned after successful authentication is also considered
"sensitive" and should be protected with appropriate network security measures.

Authentication schemes defined in the VWRAP Trust Model and User Authentication [I-D.ietf-vwrap-authentication] specification use a cryptographic hash to demonstrate the user is in possession of the shared secret associated with their account. Recommendations also exist for using transport authentication mechanisms (such as TLS client certificates) in place of shared secrets.

The authentication mechanisms described above are believed to be sufficient at the time of this writing. It is an unfortunate truth, however, that cryptographic primitives are occasionally shown to be less secure than originally believed. For this reason, VWRAP Authentication was designed to be extensible; allowing future users to define new authentication schemes without invalidating other authentication components. A further benefit of flexibility is the ability to integrate other authentication schemes into an VWRAP context. OpenID and SAML, for instance, are popular identity and user authentication technologies that are defined outside the IETF. VWRAP’s flexible authentication system allows organizations responsible for these standards to define their use with VWRAP without having to change the text of the VWRAP Authentication standard.

A typical flow of events for user authentication follows. This is a simplified version; readers with an interest in authentication are referred to the VWRAP Trust Model and User Authentication [I-D.ietf-vwrap-authentication] specification.

1. The end user presents their account identifier and an authenticator to the authentication services of the agent domain. Endpoints for user authentication protocol messages are typically well defined, public URLs.

2. The authentication service authenticates the authenticator. If the credentials cannot be authenticated, an error condition is returned.

3. The authentication service generates a seed capability and returns it to the user.

4. The user queries the "seed cap," requesting capabilities for other services the user is authorized to use.

It is important to note that in the last step listed above, the client is free to request a subset of services offered by the agent domain. This allows the same authentication service to be used by
restricted clients (for instance, a group-chat only client) as well as traditional 3d viewers.

4.2. Presence in the Virtual World

"Presence" in VWRAP refers to at least two related concepts: account presence and avatar presence. "Account Presence" describes the readiness for interaction between a user and an agent domain. A client applications signals the user’s readiness for interaction with an agent domain’s services by initiating (and completing) user authentication. Once authenticated, the user is "present." But an agent domain may export more services than interacting with the virtual world. It is conceivable a user may simply wish to manipulate their profile data, reorganize their digital assets, or make use of messaging services exported by the agent domain. Interacting with these services requires only "account presence." This type of presence implies only a client application presented legitimate credentials to the agent domain’s authentication service.

When a user wishes to interact with the virtual world, their avatar must be placed or "rezzed" there. Placing an avatar requires the cooperation between the agent domain and the region domain controlling the system with authority for the target virtual location. The quality of the system describing this interaction is "avatar presence."

4.2.1. Establishing Presence with the Region Domain

Once authenticated with the agent domain, the client application has established "account presence." Once in possession of a valid seed capability, the client application may request a set of capabilities representing services offered by the agent domain: digital asset management, instant message and voice chat support as well as placing the user’s avatar into the virtual world.

Placing an avatar in the virtual world begins with the client exercising the "place my avatar in a region" capability. As part of this transaction, the client provides the URI representing a region. Upon receipt of this request, the agent domain determines the validity of the URL provided, and if the URL resolves to a trusted region domain begins the protocol between the agent domain and the region domain to place the user’s avatar in the region.

The precise exchange of messages between each party is beyond the scope of this document, but is described in the VWRAP Teleport specification. But a few important points should be noted:
The protocol endpoint at the agent domain the client application uses to place the user’s avatar in a region is provided to the client as a capability following successful authentication. It is not a publicly defined, fixed URL.

The region the client wishes the agent domain to place their avatar in is represented as a URI. This URI may be a URN, in which case the agent domain SHOULD have the ability to convert the URN into a URL. If the target region is identified by a URL, it MUST use the HTTP [RFC2616] or HTTPS [RFC2817] URI schemes.

The agent domain MAY apply a local policy to the URI and reject the request before attempting to connect with the region domain. (a "behind the firewall" agent domain may limit clients connecting to it to systems known to be inside the local intranet, for instance.)

The agent domain MAY apply a local policy and reject the request after it makes an initial communication request with the remote region. (for example, if the region domain is operating servers with expired TLS certificates, or if those certificates are issued by a certifying authority the agent domain does not trust, it may reject the request.)

The process of placing the avatar in the region results in capabilities from the region being communicated back to the agent domain for controlling the avatar. The agent domain SHOULD forward these capabilities to the client application.

The process also results in the agent domain issuing capabilities to the region domain, allowing it limited access to information about the avatar such as the avatar’s shape and appearance.

After an avatar is "placed" in a region, the agent domain is responsible for maintaining its presence. That is to say, after the avatar has been successfully been placed in the region, the agent domain MUST refuse to allow a second region to "take" the avatar’s presence without removing the avatar from its current region.

4.2.2. Moving Presence

When an avatar moves between regions, special care must be taken that the agent domain and both the source and destination regions end the process with the same understanding as to the avatar’s location.

Moving between regions is typically initiated by the client. The process is largely the same as the initial avatar placement, but with the important added step of removing the avatar from its source
location before rezzing it in its destination. (In fact, the initial placement of an avatar can be thought of as a transfer from "nowhere.")

The process of moving between regions is described in the VWRAP Teleport specification, though implementers should keep the following important considerations in mind:

- The client signals to the agent domain its desire to move from one region to another by accessing the same capability as is used for initial placement of the avatar.
- The agent domain must again check that local policy allows movement to the new destination, and MUST receive a capability for placing the client into the new region before it removes the avatar from its current location.
- The agent domain MUST also remove the avatar from its current location before placing the avatar in the destination location. Capabilities granted to the current region MUST be revoked as part of this process.
- The location of the avatar MUST be unambiguous and the agent domain MUST NOT represent the avatars location as being in two places at once. If required, for the short period between removing the avatar from one region and placing it in another, the avatar’s location may be "in transit."

4.3. User and Group Messaging

4.3.1. Spatial Messaging

Besides the presence of a fully articulated 3-dimensional representation of the user, the most important feature of the virtual world is interaction. The virtual world is a social space; communication with other users is important. Because the virtual world simulates features of consensus reality, "proximity chat" or "spatial messaging" is an important function. This mode of interaction allows users to "hear" text messages that are spatially proximal to the user’s avatar, while ignoring other messages. The assumption being that avatar’s whose users share a common interest will congregate in specific locations in the virtual world. Or they may find their avatars in the company of other users’ avatars who are engaging in interesting conversation. Either use case is possible; emulating the consensus reality feature that people can hear conversations close to them, but not hear more distant conversations is an important feature of the virtual world.
Spatial messaging is managed by the region domain, and may be
initiated by users’ client applications or by the region itself. It
is associated with an object in the virtual world (either an avatar
or a "plain" object) and occurs at a particular location. The host
in the region domain responsible for managing spatial chat applies a
proximity algorithm to the chat to determine which avatars or objects
are close enough to hear it. Those objects are all sent messages
with the contents of the message.

Client initiated chat begins when the client application posts a
message to the capability created by the region for an avatar’s
outgoing chat messages. This capability is given to the client after
successfully establishing presence in the region. Incoming spatial
chat messages are posted to the event queue established between the
client and the region.

Complicating matters somewhat, spatial chat may occur near region
boundaries. When this occurs, the host managing a region’s messaging
must have a mechanism to communicate chat messages to its peers.
Hosts responsible for spatial chat in a region must establish event
queues with their peers in order to receive chat messages that
originated near the region’s borders.

4.3.2. User to User and User to Group Messaging

Instead of speaking on the "public" spatial chat channel (remember,
each avatar within a defined range will be able to hear these chat
messages,) users may send private user to user messages. These
messages are managed by the user’s agent domain. After
authentication, a client may request a capability for establishing a
instant messaging sessions. The client then accesses this
capability, providing a unique identifier for the target user. If
the agent domain is able to successfully establish a session with the
target user, the message originator is provided a capability to which
outgoing messages are posted.

User to Group messaging is similar, but groups are used as the target
for a message.

Incoming user to user or user to group messages will arrive in the
event queue shared by the client application and the agent domain.

4.4. Digital Asset Access and Manipulation

The virtual world contains multiple digital objects; they have a
position and an orientation as well as a shape and potentially a
texture and other features applied to them. VWRAP defines formats
for describing objects and avatar shapes, but more importantly it
describes the mechanism by which those digital asset descriptions are transferred between client applications, agent domains and region domains. VWRAP also defines a trust model and a basic permissions system, describing which users or groups have the ability to make changes to any given object.

Digital assets may be "at rest" or "in world." Objects "at rest" exist only as a description of the object, maintained by a network addressable server and accessible via a unique URL. When an object is "rezzed in world," its representation is transferred to a simulation host in a region domain and it becomes viewable by avatars and other objects in that region.

Several classes of digital assets are defined: primitive shapes, textures, sound and animations for example. In addition to the data describing the asset, metadata may be applied to objects. Unique identifiers for creators, owners and affiliated groups may be maintained by an object. Permission metadata may be added to an object to limit its distribution to remote systems or to define the allowable operations by given users or classes of users. Object name, description and tag values may be applied and should help with indexing and searching for objects. Creation and modification dates may be applied to assist systems that cache assets. Recent discussions regarding open content licenses implies an interest in license metadata. Such metadata could be of use to consumers of digital assets; allowing them to more clearly interpret the creators intent with respect to sharing.

4.4.1. Manipulating Digital Assets

A number of useful manipulations of digital assets "at rest" are defined by VWRAP. Where appropriate, asset metadata may be altered by directly communicating with the network host with authority for that asset. This host may be part of the user’s agent domain, or in the case of region-specific assets, it could be associated with a region domain. It is important to note, however, that not all metadata is modifiable by all users, even the asset’s owner. Specifically, the semantics of the creator metadata do not allow the owner to change the creator's identity. Group membership may carry some rights like the ability to manipulate the size, shape and texture of an asset, but not an asset’s owner.

The ability to access or manipulate digital assets is based on the accessor’s identity. Accessing and manipulating digital assets is performed via capabilities which expose the state of the asset to an authorized client. This requires positive identification of the accessor prior to access. In the case where an asset server is owned by the same authority as the agent domain, this access may be as
simple as providing the proper capability after user authentication. In cases where the asset server is owned by a different authority, systems for deferred authentication may be necessary. Work is currently underway to integrate OAuth and SAML into VWRAP for this purpose.

At a gross level, the types of resources exposed by a digital asset server would include:

- a resource for searching an agent’s inventory
- a resource for iterating across an agent’s inventory
- a resource for accessing or manipulating a digital asset’s metadata
- a resource for uploading new digital assets, or changes to an existing asset.
- a resource for removing a digital asset from the authority of the asset server’s domain
- a resource for transferring the asset or a copy of the asset to a remote asset server
- a resource for instantiating an object "in world"

4.4.2. Establishing Presence for Digital Assets

Digital assets are intended to be used "in world," meaning there must be a way for a user to direct a simulation host to take an asset from an asset store and imbue it with presence in the virtual world. The separation between agent based services and region based services is fundamental to VWRAP and implies the authority for the system maintaining the asset "at rest" may be distinct from that which simulates the asset "in world." In practical terms, a region simulator may need to communicate with an asset server owned by a different person or company. In situations like this, trust is paramount. Because an asset’s metadata may limit the owner’s right to make copies of an asset, the agent domain MUST be able to trust the region domain will honor that metadata.

There are two levels of trust defined when working with digital assets: host-based trust and user-based trust. The former represents one system’s expectation that the other will honor the metadata regarding ownership, creatorship and rights and restrictions implied by these concepts. Host based trust is carried by X.509 / PKIX certificates and implies a managed PKI. User-based trust represents
the expectation the asset server will expose sensitive resources only
to users with the right to access such resources.

Provided trust is established between the asset server and a
simulation host, and the simulation host can demonstrate it is acting
on behalf of a user with rights to access a particular resource,
VWRAP defines a protocol for transferring a representation of the
digital asset for simulation. As part of this protocol, access to a
digital asset may be restricted while the object exists "in world."
This is the case for objects whose creators or owners specify that
only one copy of the asset may exist at a time.

5. Document Roadmap

This section provides a brief overview of the thirteen documents
expected to be delivered by the VWRAP working group. A brief
description of each document’s contents is provided. The documents
may be read in any order, but reading this document followed by
abstract type system specification, foundation document and
authentication document is recommended. The diagram below provides
recommended sequence.

```
+-----------------+
| Intro and Goals |
+-----------------+
  |                 |
  v                 v
+-----------------+
| Abstract Type System |
+-----------------+
  |                 |
  v                 v
+-----------------+ +-----------------+
| Foundations      | | Authentication |
+-----------------+ +-----------------+
  |                 |
  v                 v
  | Launch |
+-----------------+
* all other docs
```

Figure 3: Document Roadmap
1. VWRAP : Introduction and Goals (this document)

This document provides a detailed introduction to the working group’s efforts. It describes the characteristics of VWRAP virtual worlds and sets objectives for the virtual world experience and requirements for the protocol. Specific verbiage relating to the proposed architecture of VWRAP virtual worlds and deployment patterns are provided.

2. VWRAP : Abstract Type System for the Transmission of Dynamic Structured Data

The VWRAP protocols are intended to sit at the "application layer." This means that VWRAP protocol messages may be carried over HTTP, XMPP, RTP or raw UDP. Other specifications in the VWRAP series use an interface description language and an abstract type system to specify protocol messages in a language and operating system neutral manner. The abstract type system and interface description language are defined in this document. Guidelines for carrying VWRAP over HTTP and mime types for three serialization formats are provided as well.

3. VWRAP : Foundational Concepts and Transport Expectations

The "foundations" document provides a brief introduction and definition of certain fundamental technical concepts referenced in VWRAP. Important concepts described in this document include capabilities, a HTTP based event queue and important issues for implementers wishing to carry VWRAP messages over new transports.

4. VWRAP : Client Application Launch Message

This draft defines a MIME type and message format carrying virtual world session initiation information. It is intended to be used by web browsers to launch virtual world user agents. Virtual worlds that use web authentication technologies such as OpenID, OAuth and HTTP authentication may use this message to signal the web browser to launch a client which initiates a virtual world session.

5. VWRAP : Trust Model and User Authentication

The VWRAP architecture assumes a distributed constellation of servers cooperating to simulate a virtual world. This document describes a trust model hosts may use to determine whether the origin of a request is trustworthy. The specification does not require a host to discard messages without origin integrity, but it does specify how messages carry attestations of inter-domain
trust. This draft also defines the agent_login and optional maintenance resources for VWRAP authentication.

6. VWRAP : Voice and Text Communication Channel Establishment

VWRAP is expected to rely on external text and voice communications standards. XMPP is the presumed choice for text communication while RTP is the presumed choice for voice. Neither XMPP or RTP define virtual locations as addressable endpoints, so this document defines a technique for identifying them for spatial text and voice.

7. VWRAP : Agent Presence Establishment

Also known as "the teleport protocol," this specification defines how an authorized user’s agent establishes its avatar’s presence in the virtual world.

8. VWRAP : Region Description Format

This document defines the format of the virtual world’s scene graph and the protocol interaction used by clients to retrieve it.

9. VWRAP : Digital Asset Access

This document defines the protocol used to request and receive information about digital assets. It also defines asset types an implementer MUST support (primitive shapes, textures, etc.)

10. VWRAP : Primitive Object Format

This document defines MIME types and formats for objects in the virtual world: primitive shapes, NURBS and meshes.

11. VWRAP : Avatar Format

Avatars are expected to require more detailed information about shape and movement than primitive shapes. This specification describes how primitive objects and (optionally) bio-mechanical skeletal models may be requested by client applications.

12. VWRAP : Entity Identifiers

This document describes how avatars, primitive objects, regions, assets and locations in the virtual world are represented as URIs or URLs.
13. VWRAP : Time Sensitive Messages

Some state updates in the VWRAP protocol are considered "time sensitive." That is, they will be useless unless delivered within a proscribed period. Object updates and avatar control messages may fall in this category. This document describes the format and semantics of time sensitive messages within the VWRAP protocol.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

As mentioned previously, the concept of a persistent, ubiquitous identity in the virtual world is core to the user experience. Keeping agent based services distinct from region or object based services has advantages for scalability and flexibility. However, it does have ramifications for the security of the virtual world as a whole.

Most notably, this structure puts the agent domain in the role of a trust broker. That is, the agent domain is trusted both by the set of users who operate client applications and by the set of users who administer peer domains. A transitive trust relationship exists between the peer domains and end users by way of the agent domain. The administrators of the peer domain trusts the agent domain to properly identify end users, and potentially to ensure they are members of a particular class. The end users trust the agent domain to properly identify peer domains and to potentially limit the transfer of digital assets to only those domains that have explicitly agreed to honor asset permissions meta-data.

VWRAP does not REQUIRE domains to adhere to any preset policy, however. It instead provides a mechanism for communicating identity information so that such a policy MAY be enforced.

7.1. Capabilities

VWRAP makes extensive use of RESTful resources accessed via HTTP. Application state is communicated and changed by accessing web based resources. One characteristic of such resources is they have a well defined URL, many of which are formatted as URL-based capabilities. Capabilities have the characteristic that possession of the URL implies the right to access the resource it identifies. It is
important that capability URLs are shared only with trusted participants. The VWRAP Base document defines the characteristics of URL-based capabilities, including the requirement that they include an unpredictable random component in the URL. Implementers need also ensure that these URLs are protected using suitable mechanisms (such as TLS, IPSec or link encryption.)

7.2. User Authentication

Prior to granting an end user access to any agent domain managed sensitive resource, the agent domain MUST authenticate the end user. The VWRAP Authentication specification defines three techniques for using shared secrets to authenticate end users. The agent_login resource used for end user authentication provides an extensible mechanism, allowing the development and use of additional authentication techniques (SRP, TLS Client Certificates, SASL, etc.)

Again, it should be noted that VWRAP as currently defined does not REQUIRE an agent domain to support a particular authentication scheme (shared secret, public key, secure remote password, etc.) But it does define the mechanism for three shared secret options.

Once a user is successfully authenticated, their client application is passed a seed capability (as described in the VWRAP Base specification.) This seed capability is used by the client application to request access to resources and services managed by the agent domain (including services like "place my avatar in the virtual world.")

7.3. Agent Domain to Region Domain Authentication

Agent Domain authentication, or the process of authenticating an agent host to a region host uses a X.509 PKI. Prior to communicating, the agent domain generates a key pair for a particular agent host under their control and requests a certificate from each the region domain with which they wish to interact. The region domain returns a signed certificate to the agent domain which the agent domain uses in subsequent communication with the region.

7.4. Access Control for Digital Assets

In addition to security characteristics addressing traditional network and user security issues, the raison d’etre of VWRAP is to communicate state concerning items inhabiting a virtual world. Some of these items may have access control restrictions within the scope of the applications used to simulate and render the virtual world. VWRAP defines an extensible permissions model which allows permissions meta-data to be associated with virtual items.
8. References

8.1. Normative References


8.2. Informative References

Appendix A. Definitions of Important Terms

Agent

An "agent" is the abstract information-oriented representation of a user in the system. An agent differs from an "avatar" in that it is not a visual representation and intended to be manipulated and observed by computing systems (not people.)

Authentication ("Auth") Service

A service responsible for accepting user credentials and providing service endpoints for other services.

Avatar

The "avatar" is the user’s (mostly) visual representation in the virtual world. It is the entity that end users (humans) are likely interested in interacting with.

Client Application

A "client application" is any application that mostly consumes virtual world services. Client applications may or may not be "user agents."

Region

A "region" is the set of data maintained by a "simulator."

Simulator

A service whose primary responsibility is to simulate 3d spaces (including physics simulation and object presence maintenance) is termed a "Simulator."

User

The entity controlling an avatar in world is the "user."

User Agent

A "user agent" is a client application operated predominantly by an end user.
Appendix B. Acknowledgments

The author gratefully acknowledges the contributions of: Mark Lentczner, David Crocker, Larry Mastiner, Joshua Bell, Barry Leiba, Joe Hildebrand, Chris Newman, Katherine Mancuso and Jon Peterson.

Authors' Addresses

Meadhbh Siobhan Hamrick
P.O. Box 783
Boulder Creek, CA  95006
US
Phone: +1 650 283 0344
Email: OhMeadhbh@gmail.com

David W. Levine
IBM Thomas J. Watson Research Center
19 Skyline Drive
Hawthorn, NY  10532
US
Phone: +1 914 784 7427
Email: dwl@us.ibm.com