

The Virtual World Ecosystem Framework

Richard Boyd
Lockheed Martin
Cary, NC

Richard.j.Boyd@lmco.com

David Smith
Lockheed Martin
Cary, NC

d.alan.Smith@lmco.com

ABSTRACT

Virtual World Technology is transforming many domains and disciplines with its enabling technology and is ready to mature into the Virtual World Ecosystem Framework. A perfect Virtual World Ecosystem is one where users can all routinely and naturally accomplish individual and group missions and tasks collaboratively and efficiently at higher levels of performance with superior results. As it develops further, and the full ecosystem emerges, an open and accessible architecture approach will allow users and third parties to add customized content, behaviors, and technologies to create new applications and create a virtuous cycle of value directly benefiting all users of the system. Such an environment will support the complexities of the real world with adequate yet accurate representations of the real world; to include physical world models, as well as the human social, cultural, and behavioral models needed. The environment should be designed from the beginning to allow evolution to support future/emerging mission needs and technology developments.

In 2010, the authors were directed by the Office of the Secretary of Defense for Personnel and Readiness to study the current state of virtual world practice within the DOD, and to design an architectural framework that meets all of the goals below. Our intent with this paper is to present those results and the design to the IITSEC community, as well as the larger DOD community, in 2011.

What if we could provide a multiuser virtual training environment that was...

- Scalable • Interoperable • Future-proof • Secure - with 0 client install
- With an incentive-aligned business model that encourages ecosystem growth
- That provides a consistent user interface from handheld, tablet, laptop, or desktop; to the fully immersive holodeck?

ABOUT THE AUTHORS

Richard Boyd As one of the creators of the Lockheed Martin Virtual World Labs, Richard leads a group of innovative engineers and designers across all mission areas for Lockheed Martin to harness cutting edge computer gaming and virtual world technologies to improve human performance. Richard joined Lockheed Martin in 2007 with the acquisition of 3Dsolve, a North Carolina based computer game technology firm where he was founder and CEO. Prior to that, Richard was General Manager and VP of Sales for Virtus Corporation, where he served on the management team with David Smith that created several pioneering computer gaming companies including Red Storm Entertainment, with author Tom Clancy; and Timeline Computer Entertainment, with author Michael Crichton. In 1995 David Smith and Richard Boyd also co-wrote an industry-leading book on VRML technologies, called *The Virtus VRML Toolkit*, that was widely distributed and translated into three foreign languages

David Smith serves as the chief innovation officer of Lockheed Martin Global Training and Logistics. He has been focused on interactive 3D and using 3D as a basis for new user interfaces, collaborative environments, design tools, and entertainment for twenty-five years. He has developed and shipped world class systems and applications including The Colony, the first real-time 3D adventure game, Virtus Walkthrough, the first real-time 3D design tool for personal computers, OpenSpace, the first 3D component-based dynamically extensible architecture, the Croquet Project, an open source collaboration platform, and Qwaq Forums, based on the Croquet Project, a collaboration platform developed specifically for enterprises.

The Virtual World Ecosystem Framework

Richard Boyd
Lockheed Martin
Cary, NC

Richard.j.Boyd@lmco.com

David Smith
Lockheed Martin
Cary, NC

d.alan.Smith@lmco.com

Overview of the OUSD vision

In the summer of 2010, the Office of the Under Secretary of Defense for Strategy, Technology and Readiness released a request for proposal focused on resolving the interoperability problems of training efforts in virtual worlds being pursued by multiple services of the Department of Defense. The resulting effort has enjoyed serendipitous timing with the Internet adoption of WebGL and HTML5. As a result, a new opportunity for unprecedented interoperability and re-use, with much less friction and more accessibility, is afforded to a simulation industry pressured by retreating resources. The timing could not have been more provident.

The ensuing study described a host of challenges for an interoperable virtual world framework for training. The study determined that a virtual world framework must:

- Make training available 24/7, via the global information grid across the spectrum of training audiences; from individual home station users and small units to large force Combatant Command users.
- Possess a high level of realism. The platform must take advantage of modern graphics capabilities, and offer a clear path to ensure that the system can integrate new advances in the area as they become available. Further, the platform must provide realistic object interactions including physical models and artificial intelligent characters, and societies.
- Use common applications, references, and operational capabilities. The platform must allow integration of existing key applications as well as new applications that will be developed. This includes both 2D and 3D applications and support for both server-based and client-based execution. A common shared ontological/behavioral model will need to be explored.
- Be rapidly scalable and composable by the training user without the need for specialized skills. The

platform must be accessible to and extensible by a wide range of users and third-party developers, and there must be a clear model for developing and integrating easy-to-use tools.

- Be rapidly modifiable to replicate new operational capabilities or changes in the real operating environment. Quickly support mission rehearsals. The platform must have a clear content pipeline that ensures a simple path for creation of new content, importing of existing content, and a clear path for integration of real-time data acquisition and deployment.
- Provide a well understood path toward enabling a two-way interface between live and virtual training systems and their virtual representations in the federated virtual world. Operations in the federated virtual world and live and virtual training systems will need to be synchronized in real time so as to enable stimulation of sensors, visual replications, and interactions between platforms operating within and outside of the federated virtual world across the spectrum of training environments or systems securely. Support for future augmented reality systems must be considered.
- Support information operations, cyberspace, nuclear or catastrophic warfare, space, civil affairs, language and culture, and other soft skills training requirements across the globe. The platform must be open and extensible in virtually every dimension required for the vast complexity of training the next generations of soldiers.
- Be interoperable with interagency partners and multinational capabilities in order to train to a comprehensive approach. The platform must enable a number of orthogonal missions while maintaining a common extensible framework. In short, if two groups can interoperate in the real world, they must be able to interoperate in the virtual domain.

There are a number of key technical capabilities that the platform architecture must address from the start. The study determined that these capabilities are virtually impossible to add to an existing platform as they are deeply integral to how the system operates.

- Support for detailed after action review. “Key-frame” world states and all significant state modifications and data interchange (user actions, audio, and video) must be archivable and streamable. This includes both 3D world interactions, but should also include document management, and document and world versioning (storage is basically free). A “rewind” capability would be optimal, but likely computationally expensive, as most world modifications are “lossy” and cannot be easily revoked.
- Support for replicated computation and simulation. This allows for very complex direct user and physical interactions that would otherwise place a significant load on a server.
- Support for client/server interactions. These are typical “game” interactions where precise interactions are not required.
- Support for bi-directional data streaming interactions (audio, video, VNC). This allows sharing of legacy applications, as well as enhanced user-to-user communication modalities.

The Virtual World Framework (VWF) is a fast, light-weight, web-based architecture for creating and distributing secure, scalable, component-based collaborative virtual spaces. It leverages existing web-based standards, infrastructure, and emerging technologies with the intent of establishing a powerful yet simple-to-use platform that is built on top of the next generation of web browsers. These technologies include HTML 5 – a significant upgrade in expressive power for the web; WebGL – an integrated 3D graphics capability; WebSockets – providing a full TCP/IP connection between the client and server; JavaScript – the programming language of the web; and XMPP – the Extensible Messaging and Presence Protocol (Jabber).

The VWF will be particularly focused on portable and mobile platforms, as well as scalable, ad-hoc network infrastructure such as cloud computing. It is a zero-install platform, with additional software components added dynamically as required. VWF spaces can be embedded in virtually any application including web pages and emails. Further, VWF spaces can embed existing applications and browsers.

The VWF will be deployed as an open platform to ensure worldwide adoption. An ecosystem will be designed around it to ensure that government, industry and academia can finally take full advantage of the network effects of our Web 2.0 era, as well as the well known benefits of collaborative simulation training.

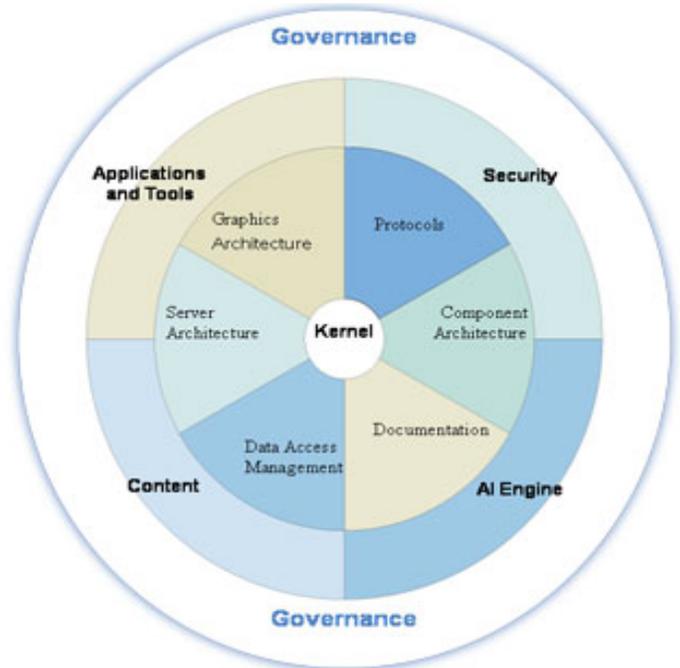


Figure 1 Elements of the VWF, Governance of the Ecosystem is the key to success.

Architecture

The diagram above is a high-level overview of the system. We will approach this discussion from the center of the diagram and work our way out to the edge. The kernel and other elements that are closer to the center are extremely well defined. As we move out from the center, the variability and flexibility increases dramatically. This culminates with the ultimate variable, which is the social experiment of governance that will be required to develop and sustain this project.

Kernel

The VWF system kernel will be a JavaScript library that implements the majority of the communication, collaboration, and object/data management for a replicated world. It will be built using other existing frameworks including an extended object model, such as that provided by prototype.js and a rendering platform, such as that provided by the DOE Kuda project: <http://code.google.com/p/kuda/>.

The kernel of the VWF can be thought of as the main switch for the system. It manages the synchronization of new users, communication between the various components, communication with the web-server/reflector, and communication between the users. The main role here is to manage replicated behaviors and state.

Component Architecture

The component architecture defines the nature of individual objects that can be manipulated, demonstrate behavior, and their state defines the persistence model of the virtual world. This effort will define both the software architecture and the file structure for component definitions.

A VWF space consists of a set of dynamically loaded components. A component is a late-bound object that includes a combination of both the object's behavior and its visual properties. These properties are considered to be "orthogonal" and it is the intent of the design to allow a great deal of flexibility in the visual properties of a component, but an extremely rigorous restriction on the definition of the behavior model.

The component behavior includes response to direct user interactions, response to interactions with other components, and changes of the component state over time. The kernel's main responsibility is to manage the components and their various behavior interactions that make up a particular virtual world. This behavior needs to be identical on all systems that the replicated instance of the component is running on. We refer to this aspect of the component as its "model".

The visual properties of a component will usually be determined by the target platform and its capabilities. This means that when a component is loaded, though its behaviors must be well-defined and identical, its visual representation must be flexible. For example, the system should allow a component to be represented as a fully rendered 3D object on one system and perhaps a 2D top-view image on another. Both would have the same behaviors and interactions, just their visual representation will be different. The visual properties of a component are referred to as its "view".

Protocols

The VWF protocols are simply the vehicle of communication between the various sub-systems and components that make up a VWF space. These include higher-level standard communication protocols such as Jabber/XMPP, to specific VWF event-based message passing protocols that enable the replicated global interaction and computation that defines the system.

Data Access Management Framework

All objects in the VWF are accessed via a URL-type reference inside the world. That is, just as a web page does not include the actual images that may be used with it, but simply a reference to it, the same thing applies to all content in a collaborative VWF virtual world. Many objects that make up the virtual world can actually be accessed dynamically from various content repositories or can even be dynamically constructed as required. For example, one user may be visiting a virtual world using a high-end desktop machine, while another may be using a mobile device like an iPhone. The requests for the content that make up the virtual world would also include information about the target platform, hence the high-end desktop will receive a maxed out file, while the iPhone will receive a decimated version of the same object – much smaller both because of the smaller bandwidth of the delivery channel available to the iPhone and because of its limited space and less capable graphics engine.

Graphics Architecture

As noted in the section describing components, it is expected that the VWF will be required to support a range of graphics capabilities; however, it is important to establish at least one or two approaches to rendering the components in a VWF space. Most important is a 3D scene manager. There are a number of very good WebGL focused graphics platforms available today, and we will most certainly focus our efforts on one of these.

Server Architecture

We envision a direct extension to existing web-server platforms for the VWF. We will initially be using a system based upon node.js, which is a very simple yet powerful server framework. This allows us to define the entire VWF in a single language. This is not a requirement of the server framework, as any system that implements the reflector protocols will work.

The server of the system has a number of responsibilities, including the traditional web-page definition and management. In addition, the server must provide a time-stamped message reflector service, content management, and any additional side-band communication and services (video, audio, VNC) that might be required.

Documentation

One of the most overlooked and essential aspects in the development of successful platforms. is the availability of high quality documentation. Though our intent is to provide a very clean and relatively simple-to-use platform for developers and users, it is essential that a

set of guides be provided to them that enables them to quickly understand and properly leverage the system. Given the diverse set of users that we hope will be engaging with the system, we will need to provide documentation that addresses their diverse requirements. At a minimum, the following documents should be created and made available via the web:

- **System Developer Guide**

This provides an overview of the architecture and is intended for the core development team, as well as groups that may need to extend or modify the core system for a particular non-standard application. This will include a very thorough overview of the core architecture of the key components of the system, as well as the requirements that informed this architecture, how the system is constructed, alternatives that might be taken, and detailed implementation issues.

- **Application Developer Guide and Cookbook**

This is the most important document, in that it is intended for the developer community. This focuses primarily on the development of components, but also includes information on how to develop and deliver new tools and applications on top of the system, and how to create new user interface models. It further describes how a developer can create a new virtual world and server with the VWF, how new components are constructed, and how to manage key aspects of the system, such as the rendering API and the model/view approach.

- **Standard User Guide**

Though the VWF will enable a large degree of flexibility in terms of user experience and interaction models, it is expected that the vast majority of users will likely utilize the system in a standard way similar to that available from traditional multi-user virtual world environments today. This document will provide an introduction to standard use of the system, including how to create and add new content, how to create new behaviors, and how to leverage the tools that will be provided to the core community.

- **Installation and Operations Guide**

Standard installation packages and virtual machine images will allow VWF to be deployed easily in common configurations, but larger and private installations will require additional attention. This document will provide information for system administrators for capacity planning, performance tuning, integration with enterprise authentication systems, and monitoring.

Content

The content that makes up a VWF space is entirely made up of components defined above. The components will include the component behaviors, which are simply defined by executable code that is included with the component and the visual representation of the component that can be made up of a number of different kinds of data. Our intent is to allow the visual representation of this content to be extremely flexible. This includes:

- Traditional 3D polygonal representations. This is usually defined by 3D meshes – points that define triangles, which define the surfaces of an object.
- Texture images. These are the images that are wrapped onto the polygonal (or other) representation of the object.
- Procedurally defined content. This includes simple shaped objects such as spheres and cubes, defined by simple parameters such as radius, length, width, height, etc. Additionally, procedurally defined objects can include spline-based objects and generalized extrusion objects. This kind of content can be quite small when compared to polygonal representations and can be dynamically rendered by a GPU-based shader.
- 2D representations. Certain situations may require that a component provide a 2D visual representation of an object. This can be provided using a 2D image or a dynamically rendered object.

AI Engine

It is clear that a successful VWF targeted to complex training and simulation exercises will require significant object intelligence. In particular, the VWF components are intended to be “plug-and-play” such that a component created for one virtual world exercise should be transferable to a new situation while still exhibiting reasonable and valuable interactions in this new context. One possible approach to this is to define a standard ontology across the components that will enable them to query other components for their capabilities and thereby negotiate the nature of their interactions.

Tools and Applications

Though tools can be thought of as a form of application, we differentiate between them here. Tools are software used to create, modify, and edit the content, including behaviors that make up a particular virtual world. Tools are intended for the developers of a virtual world. An example of this might be a tool that allows new content to be placed into the virtual world and edit the component behaviors.

Applications are a specific focused use of a given virtual world. An example of this might be a virtual world designed to train on a specific set of maintenance tasks. We use tools to construct applications. Applications are intended for the end users of a virtual world.

Perhaps the key differentiator between the success and failure of a new platform is the strength and capabilities of the tools that exist to support it. A good tool chain will enable very quick turn-around between creation of content and experimentation of that content in the context of a virtual world.

Sample Tools Needed

One of the major goals of the VWF project is to provide a foundation for the development or porting of tools that will allow even relatively naïve users to create and modify the content that makes up a space that will be used for training. The DARPA Realworld project has made great strides in this direction, and a host of tools on the market today could potentially be modified to support an open VWF art and development pipeline.

SECURITY

The VWF can be made as secure as any other web experience. Although the VWF does not explicitly address issues around security, it takes advantage of the full infrastructure around secure web browsing and

secure client-server models already supported by IT organizations. This familiarity should remove some of the uncertainty that exists around virtual world adoption today; however, it also shares many of the same challenges that all other web-based solutions share. There are a number of additional issues around security that will need to be considered,

- **ECMAScript Enhancements.** ECMAScript (JavaScript) 5 has a number of key enhancements that dramatically improve the security landscape. These include such features as tamper-proof objects (can't assign a new behavior to a “frozen” object), strict access modes, and confining potentially offensive code that may be part of a VWF component. This is not a complete solution by any means, but is a major step forward in creating more useful and secure infrastructure for the web and for the VWF in particular.
- **Content Sourcing.** As a given web page today is made up of a number of components with content from a diverse set of sources (images, YouTube videos, sounds, etc.), a VWF component can also be sourced from virtually any location. This is particularly critical when we consider that a VWF world will be made up of intelligent components that can both query and interoperate with other existing components that are in a space.
- **Access Control.** With the content delivered over http, web servers can impose access restrictions using mechanisms that are already in use with other content. These include password control and limited access systems, such as OAuth. Complex installations may connect these mechanisms to a backend security database for centralized control. https can also be used to ensure that data is not exposed in transit. With scenes potentially composed of data from many sources and protected by a variety of access controls, VWF will need to be aware of the user's identity and manage access on behalf of the user.

- **Malicious Content.** VWF wishes to treat components as abstract entities and run their attached JavaScript in a sandbox. This is possible to a degree, but with the browser's JavaScript engine running both the component scripts and the scripts that create VWF, the components cannot be completely isolated from the browser. If components can be loaded from arbitrary locations, the user will be exposed to attacks from malicious scripts. To mitigate this risk, some combination of the following will be needed:
 - Enforce a same-origin policy that only allows scripts to be loaded from the domain that published the scene,
 - Require that components be signed so that they may be traced to their origin and optionally blacklisted or whitelisted, or
 - Script components using a reduced-complexity language that is interpreted by VWF rather than being executed alongside it.

GOVERNANCE

THE DOD VIRTUAL WORLD FRAMEWORK ECOSYSTEM

The central organizing idea of the DOD Virtual World Framework is that we now have the unique opportunity to harness the major forces behind the success of the Internet as a communications medium in order to create an interoperable, standardized way to build and deploy 3-D simulation learning in a virtual world. We often speak of technology trends and forces like Moore's Law for exponential growth in processor price performance and Metcalfe's Law for networking effects when discussing how the Internet came to be so pervasively useful. But we know now that the social, cultural, regulatory, economic, and business forces that served as boundary constraints and guides for early Internet growth deserve the most credit and attention when planning for new ecosystem development.

As daunting as the technical challenges of the VWF are, the governance challenges are clearly the lynch pin. With an open model governed by the proper incentives, we should be able to cultivate a **massively parallel contribution system** that creates value for all

stakeholders and rapidly builds a powerful VWF capability.

Aligning incentives

Human and other biological systems populated by goal-seeking autonomous actors are driven and governed by incentive systems. Risk, reward, and penalties, whether natural or man-made, have profound influence on how a system evolves and in which direction it grows. Anyone who has raised a family, run a business, or governed a population of citizens understands that incentive and punishment systems must be carefully designed in order to achieve the desired optimal results.

A technology ecosystem describes the structure and behavior of a network of organizations that share a key technological platform and the ways individual actors can flourish in such an environment. The design of the incentive system is critical to the success of a technology ecosystem. .

For decades, we in the IITSEC community have struggled with the special problems inherent in getting government, industry and academia to cooperate efficiently around a host of problems that emerge as our country's needs grow more complex. Whether it is better weapons systems, a more reliable and efficient energy infrastructure, better education, or better training through simulation, the problems of interoperability, cooperation and reusability continue.

Overcoming the "Coase Cost"

In 1937, Ronald Coase wrote a brief but very influential article called "The Nature of the Firm" (Coase) in which he sought to explain and account for the fact that we are not ants or bees; and that any organization comprised of humans must endure some transaction cost for keeping everyone in unison on a project. The reason governments and firms grow large, he maintained, is that the cost of cooperating with others in unison and harmony is larger than doing it internally. These external costs include search and information costs, bargaining costs, keeping trade secrets and policing and enforcement costs.

The Internet has changed this calculus. The telephone and high speed air travel had already started to reduce the cost of cooperation from when he wrote that article, but the Internet affords a new opportunity for reducing friction and increasing cooperation. Online auctions and marketplaces are reducing transaction costs for artists and developers. Companies like Google have created tremendous value for themselves by reducing search costs. Social media applications have reduced the costs for many-to-many communication and sharing of ideas and media.

The VWF Ecosystem must take advantage of these network 2.0+ effects.

All of these friction reducing capabilities are growing increasingly mobile, thanks to the inexorable march of Moore's law and Metcalfe's law.

Moore's law (especially when combined with Mead's) implies that the price performance of computing power is doubling roughly every eighteen months. Metcalfe's law states that the value of a network is the square of its nodes. Every time the network doubles in size it quadruples in value.

As seen in the chart below, the number of users accessing the Internet via mobile devices is expected to eclipse that of desktop devices in 2014.

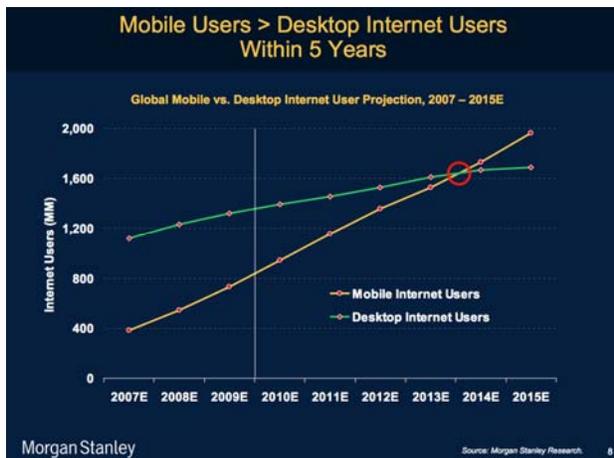


Figure 2 Mobile Internet access eclipses Desktop access in 2014

With these new mobile powers to organize and cooperate we are now seeing dictatorships in the middle east tumble and small firms reach large audiences with new capabilities.

Brokerage on the Cloud

On July 10, 2008 Apple launched the App store. Building on lessons learned by companies like NTT Docomo in Japan, the brilliance of the App store model was its complete transparency. A developer could create an application for an iOS device and place it in the store and expect to receive 70% of the purchase price collected by Apple (a price that the developer set). As of June, 2011, there are more than 425,000 third party apps in the App store. The app store is strong testament to the fact that clear incentives encourage positive ecosystem behavior, even when the ecosystem governor (in this case, Apple) exerts control over the ecosystem.

The government has also harnessed this trend with its own app marketplace. Federal CIO Vivek Kundra has

urged all federal agencies to explore cloud computing as a way to increase efficiency and interoperability. The resulting Govcloud initiative (Ballmer) caused the birth of the government's own app store for cloud IT services, productivity apps and even social media apps, and is designed to dramatically reduce the \$76 billion the government spends on IT applications and services every year.

The evolution of the VWF will also take advantage of Cloud Computing. Because it is a zero install model, the only thing required of the end user is a modern web browser and an Internet connection. Streaming game services such as Onlive and Gaikai have demonstrated that detailed and graphically intense multiplayer first person shooter games and even massively multiplayer online games can be delivered to fairly dumb terminals with all of the assets and rendering residing on servers.

The VWF ecosystem will encourage a host of online exchanges for code examples (components), content pipeline tools, training, and complete applications. These small pieces, loosely joined, will be able to be combined into robust simulations with much less effort than working with the arcane tools of a traditional proprietary walled garden platform.

The essence of a the VWF technology business ecosystem is that, in a networked society, monolithic actors are at a disadvantage to networks of organizations working together towards a common goal. The Internetwork of "small pieces loosely joined" as David Weinberger put it, is what gives the Internet its resiliency. If we view our plans for the VWF ecosystem from a high conceptual level, it will be possible to influence that network activity in a desirable direction with less effort, energy, and capital than

would be required if we were attempting to act alone (as with so many of the point solution efforts we have abandoned). In order to create an ecosystem around the VWF, we need to create a set of positive sum relationships, a symbiosis, between stakeholders who work together around the technology platform. We must design the incentive system so that all of the actors in the VWF ecosystem see a clear benefit to participating in what we call a *massively parallel distributed contribution system* of effort.

After weighing the various options regarding how to give birth to a thriving VWF ecosystem, we found ourselves increasingly turning to an open multi-sided platform governed by a consortium of government, industry, and academic experts as the recommended design for the initial stage of the development of the ecosystem. This is a sort of hybrid design of the Google Android marketplace and the W3C consortium that presides over web standards.

A two sided platform is one where a developer creates an application, then sells it to the end user (Example Teleplace) A multi-sided platform (Hagiu) can include a manager of the platform, developers, end users and brokers who enable content transactions between the platform contributors. Varying levels of control can be exerted by the platform creator over the transactions among the parties, ranging from high control to high autonomy.

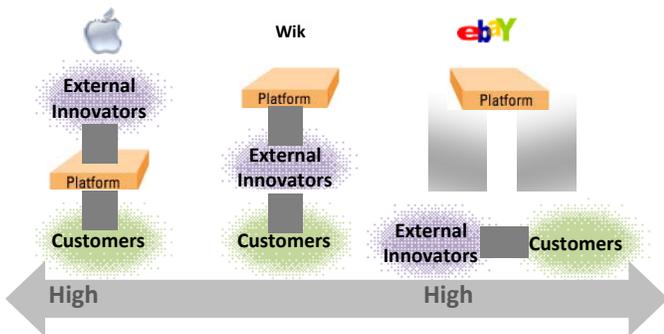


Figure 3 Ecosystem Platforms can range from high control to high autonomy.

Although the Apple iTunes App store is often cited as a model of ecosystem design, it is actually a High control model. Every application must be submitted directly to Apple for approval and distribution, and Apple exacts a 30% fee for each transaction. The Microsoft multi-sided platform model is actually more open. Microsoft creates an ecosystem for developer and controls them by managing a set of open application programmer interfaces (APIs). Developers are then free to develop for and transact with end users with no interference from Microsoft. Ebay is a more autonomous model, with ebay simply providing the infrastructure which

allows buyers and sellers to find each other efficiently (and thus lowering the Coase cost) and simply taking a transaction fee with no approvals required.

The Linux open source ecosystem is perhaps the most autonomous example of ecosystem management. Participants in this ecosystem develop content and are free to share it or sell it in the marketplace with no interference from any other body. The usefulness of the product is what determines success, not any intervening regulator. Even in this environment, though, Red Hat still enjoys a \$9 billion market cap packaging and selling products, services and training to this open ecosystem. There will likely be many opportunities like this within the VWF ecosystem.

Rather than prescribing a one size fits all model for the VWF ecosystem, we believe that it is likely that there will be multiple management and business models driven by the use cases and security requirements of the military, branches of civilian government and intelligence agencies.. It is clear at this point that, even if we do nothing, an ecosystem is already developing around the HTML5 and WebGL standards on the open Internet.

Large companies like Google, Zynga and Facebook are taking advantage of these standards, and small companies too are beginning to wade in. Very soon a very unique opportunity will develop for the DoD to create a governance model for managing a core, highly controlled virtual world environment (Core) that may include highly controlled APIs, security and content suitable for deployment on secure networks. This inner Core will be highly controlled by an agency or small group within the government.

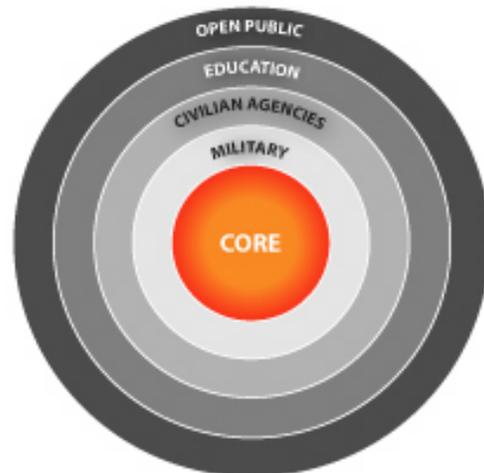


Figure 4 Each ring could have a different business model and level of control based on security needs.

As we move out from the core to other service and government uses we expect to see less control and more autonomy in the management of the ecosystems; all the way out to an open Internet environment where governance will look more like the Linux model. There will be an opportunity for each ring to borrow from the more autonomous rings on the outside of its position while interior, more highly controlled rings can manage content and applications using methods appropriate for their use cases.

CONCLUSION

The authors of this paper believe that, like the Internet, the VWF ecosystem will evolve in multiple, possibly unpredictable ways after the initial deployment stage as we witness the evolution of a more mobile semantic "Internet of things" with a variety of new user interfaces, sensors, and computer devices driving the evolutionary direction.

If the design and governance of the VWF are properly realized, we should see an explosion in the number of non-specialists empowered to create simulation learning and other kinds of content, and applications that are easily discoverable and procurable with proper remuneration and attribution to the authors, while also being secure and sliceable by permissions level, by aggregators and end users.

The business models of the Virtual World Framework are a departure from business as usual for the DoD, but the VWF massively parallel contribution system model should provide a place for every firm and every stakeholder. There should be as many business models as there are on the Internet, and possible some new models will emerge.

ACKNOWLEDGEMENTS

The authors would like to effusively thank Frank Di Giovanni, Carl Rosengrant, Ben Wash, Susan Malkus, Todd Sherman and Alan Kay for their contributions to this project and the great opportunity this represents to transform our readiness level for the challenges to come.

REFERENCES

- Ballmer, Steve, March 5, 2010 retrieved from: http://www.techflash.com/seattle/2010/03/notes_steve_ballmers_five_dimensions_of_the_cloud.html
- Boudreau, Kevin J. & Lakhani, Karim R., "How to Manage Outside Innovation", MIT Sloan Review, 2009
- Boyd, Richard, & Smith, David. (1995). *The Virtus VRML Toolkit*, Hayden Books
- Coase, Ronald., (1937). "The Nature of the Firm" *Economica*, 16, pages 385-405
- Fjeldstad, Øystein & Andersen, Espen, "Casting Off the Chains", Norwegian School of Management, 2003
- Hagiu, Andrei, "Multi-Sided Platforms: From Microfoundations to Design and Expansion Strategies" Working Paper, Harvard Business School, 2009
- Iansiti, Marco; Levien, Roy, "The Keystone Advantage", Harvard Business Press, 2004
- Moore, James M., "Predators and Prey: A New Ecology of Competition", *Harvard Business Review*, 1993
- Tapscott, Don & Williams, Anthony D., "Wikinomics: How Mass Collaborations Changes Everything", The Penguin Group, 2008
- <http://www.eweek.com/c/a/ApplicationDevelopment/W3C-HTML5-Spec-Due-in-2014-108529/>

WebGL Resources

- WebGL 64K competition
<http://gl64k.com/>
- Google Chrome native client
<http://code.google.com/p/nativeclient/>
- Shader Toy
<http://www.iquilezles.org/apps/shadertoy/>
- Google Body
<http://bodybrowser.googlelabs.com/>
- WebGL Camp
http://www.webglcamp.com/wiki/index.php?title=Main_Page
- GLGE WebGL library
<http://www.glge.org/>
- DOE KUDA
<http://code.google.com/p/kuda/>