

Quantitative Evaluation of Usage Management Overlay Network Architectures

Christopher C. Lamb

*Department of Electrical and Computer Engineering
University of New Mexico
Albuquerque, NM, USA
cclamb@ece.unm.edu*

Gregory L. Heileman

*Department of Electrical and Computer Engineering
University of New Mexico
Albuquerque, NM, USA
heileman@ece.unm.edu*

Abstract—Herein, we establish a model for quantitatively examining system architectures and then use that model to evaluate a taxonomy of proposed architectural configurations of overlay networks for usage management. We first cover the need for overlay networks supporting usage management by highlighting the pending migration to utility computing models and the current shortcomings of those kinds of systems from a variety of perspectives. We also briefly cover the state of usage management within these kinds of systems, also examining the known state of the art in proposed cross domain solutions. We then cover a method for evaluating these kinds of systems and then use this method to evaluate a taxonomy of proposed overlay systems for usage management. We close with our current and future directions of research.

Keywords—usage management; overlay networks; quantitative architecture evaluation;

I. INTRODUCTION

Current enterprise computing systems are facing a troubling future. As things stand today, they are too expensive, unreliable, and information dissemination procedures are just too slow.

Generally, such systems still do not use current commercial resources as well as they could and use costly data partitioning schemes. Most of these kinds of systems use some combination of systems managed in house by the enterprise itself rather than exploiting lower cost cloud-enabled services. Furthermore, many of these systems have large maintenance loads imposed on them as a result of internal infrastructural requirements like data and database management or systems administration. In many cases networks containing sensitive data are separated from other internal networks to enhance data security at the expense of productivity, leading to decreased working efficiencies and increased costs.

These kinds of large distributed systems suffer from a lack of stability and reliability as a direct result of their inflated provisioning and support costs. Simply put, the large cost and effort burden of these systems precludes the ability to implement the appropriate redundancy and fault tolerance in any but the absolutely most critical systems. Justifying the costs associated with standard reliability practices like diverse entry or geographically separated hot spares is more

and more difficult to do unless forced by draconian legal policy or similarly dire business conditions.

Finally, the length of time between when a sensitive document or other type of data artifact is requested and when it can be delivered to a requester with acceptable need to view that artifact is prohibitively long. These kinds of sensitive artifacts, usually maintained on partitioned networks or systems, require large amounts of review by specially trained reviewers prior to release to data requesters. In cases where acquisition of this data is under hard time constraints like sudden market shifts or other unexpected conditional changes this long review time can result in consequences ranging from financial losses to loss of life.

Federal computer systems are prime examples of these kinds of problematic distributed systems, and demonstrate the difficulty inherent in implementing new technical solutions. They, like other similar systems, need to be reimagined to take advantage of radical market shifts in computational provisioning.

II. MOTIVATION

Current policy-centric systems are being forced to move to cloud environments and incorporate much more open systems. Some of these environments will be private or hybrid cloud systems, where private clouds are infrastructure that is completely run and operated by a single organization for use and provisioning, while hybrid clouds are combinations of private and public cloud systems. Driven by both cost savings and efficiency requirements, this migration will result in a loss of control of computing resources by involved organizations as they attempt to exploit economies of scale and utility computing.

Robust usage management will become an even more important issue in these environments. Federal organizations poised to benefit from this migration include agencies like the National Security Agency (NSA) and the Department of Defense (DoD), both of whom have large installed bases of compartmentalized and classified data. The DoD realizes the scope of this effort, understanding that such technical change must incorporate effectively sharing needed data with other federal agencies, foreign governments, and international organizations [1]. Likewise, the NSA is focused

on exploiting cloud-centric systems to facilitate information dissemination and sharing [2].

Cloud systems certainly exhibit economic incentives for use, providing cost savings and flexibility, but they also have distinct disadvantages as well. Specifically, they are not intrinsically as private as some current systems, generally can be less secure than department-level solutions, and have the kinds of trust issues that the best of therapists cannot adequately address [3].

How to address these issues is an open research question. Organizations ranging from cloud service providers to the military are exploring how to engineer solutions to these problems, and to more clearly understand the trade-offs required between selected system architectures [4]. The problems themselves are wide ranging, appearing in a variety of different systems. Military and other government systems are clearly impacted by these kinds of trust and security issues, and they also have clear information sensitivity problems. This, coupled with the fact that these organizations have been dealing with these issues in one form or another for decades make them very well suited for prototypical implementation and study.

Current federal standards in place to deal with these issues in this environment are managed by the Unified Cross Domain Management Office (UCDMO). UCDMO stakeholders range from the DoD to the NSA. The current standard architectural model in place and governed by the UCDMO to deal with this kinds of issues are *guard-centric cross domain architectures*.

1) *Shortcomings of Current Systems:* Having reviewed the current state of the art of these kinds of cross domain solutions, they still have clear similarities, and in fact have not progressed far beyond the initial notions of how these kinds of systems should work. They still, for example, all use some kind of filter chaining mechanism to evaluate whether a given data item can be moved from a classified to an unclassified network. Both NSA models used filters explicitly, as did the BAH model. They all use a single guard as well, a sole point of security and enforcement, providing perimeter data security, but nothing else. In each of these current system architectures, users are only allowed to exchange one type of information per domain. The physical instantiations of these models are locked by operational policy to a single classification level limit. Users cannot, for example, have Top Secret material on a network accredited for Secret material. Finally, these models violate the end-to-end principle in large service network design, centralizing intelligence rather than pushing that intelligence down to the ends of the system [5].

2) *Characteristics of Future Systems:* Future systems will generally demonstrate decentralized policy management capabilities, infrastructural reuse, the ability to integrate with cloud systems, and security in depth. Policy management is decentralized and integrated within the fabric of the

system. The system is both more secure and resilient as a result, better able to control information and operate under stressful conditions. Multi-tenancy can lower costs and increase reliability and is furthermore a common attribute of cloud systems. An appropriately secured system facilitates integration of computing resources into multi-tenant environments. The ability to handle multi-tenant environments and to reliably secure both data at rest and data in motion leads to computational environments deployable in cloud systems. Finally, systems must operate under *all* conditions, including when they are under attack or compromise [6]. Ergo, they must provide protection to sensitive data in depth.

A. Other Related Work

This work introduces the notion of usage management embedded in a delivery network itself. It also provides an in-depth analysis of the challenges and principles involved in the design of an open, interoperable usage management framework that operates over this kind of environment. Besides referencing the material we have covered in depth to portray the current state of the art, the analysis includes application of well-known principles of system design and standards [7], [8], [9], research developments in the areas of usage control [10], [11], policy languages design principles [12], digital rights management (DRM) systems [13], and interoperability [14], [15], [16], [17], [18] towards the development of supporting frameworks.

While a large body of work exists on how overlay networks can use policies for *network* management, very little work has been done on using usage policies for *content* management. The primary contribution in this area focuses on dividing a given system into specific *security domains* which are governed by individual policies [19]. This system fits into our proposed taxonomy as an α -type system as it has domains with single separating guards.

A large body of work currently exists with respect to security in and securing overlay networks. These kinds of techniques and this area of study is vital to the production development and delivery of overlay systems, but is outside the scope of this work.

B. Cross Domain Solutions

The Unified Cross Domain Management Office (UCDMO) supports efforts to develop other specific solutions that have been presented over the past few years within the government space to handle this kind of information management. The National Security Agency set the standard in this area initially. In 2009, at a conference sponsored by the UCDMO, Booz | Allen | Hamilton and Raytheon presented alternative notional architectures contrasting with current NSA-influenced approaches [20], [21], [22], [23].

III. TAXONOMIES OF USAGE MANAGEMENT OVERLAY

A clear taxonomic organization of potential steps in approaching finer grained policy based usage management helps in describing the difficulties inherent in developing potential solutions as well as aiding in planning system evolution over time. Here, we have five distinct types of integrated policy-centric usage management systems, as shown in Table I. Of these five, only the first two levels are represented in current system model.

In this taxonomy, it is not required that systems pass through lower levels to reach higher ones. This taxonomy represents a continuum of integration of usage management controls. Systems can very well be designed to fit into higher taxonomic categories without addressing lower categories. That said however, many of the supporting infrastructural services, like identification management or logging and tracing systems, are common between multiple levels.

The taxonomy itself starts with the current state, integrating policy evaluation systems into the network fabric gradually, moving away from filters, then by adding policy evaluation into the routing fabric, then the computational nodes, and finally by incorporating evaluation directly into content.

A. Evaluation Methodology and Model

In order to successfully evaluate the elements of our overlay taxonomy, we must first establish a model against which to measure the presented architectures. The current standard for evaluating software quality is ISO/IEC 25020 and this, along with other related standards from other service delivery organizations has begun to be integrated into both academia and industry as a tractable way to measure system quality [24], [25].

This particular model must address quality attributes specific to the presented architectures rather than the functional domain. The goal of this model is to allow for architectural evaluation of policy evaluating architectures regardless of the specific functional domain. Ergo, injecting a specific functional domain into the evaluation or the evaluating model is unacceptable. Acceptable attributes are those which directly target quality attributes of the architectures in question.

$$E = \{f_e, f_r, f_u, f_p, f_m, f_f, f_s, f_c\} \quad (1)$$

$$W = \{w_e, w_r, w_u, w_p, w_m, w_f, w_s, w_c\} \quad (2)$$

$$s = \sum_{W, E} w_i f_i \quad (3)$$

We are specifically interested in evaluating architectures for policy evaluation functional suitability, reliability, usability, possible performance efficiencies, maintainability, portability, security, and compatibility, specifically neglecting any kind of domain functional suitability. Each area will be associated with an evaluation function. The suitability of

a given architectural option will be evaluated by a tuple of these functions, which can then be converted into a weighted sum leading to a single quantitative metric representing suitability under evaluated conditions.

Also important to note, certain attributes may not be able to be evaluated using specific architectural models. For example, conceptual or notional architecture models are intended to convey specific ideas prevalent in a given system architecture rather than ways that architecture will be realized. As such, these kinds of models generally cannot be evaluated for things like portability or performance efficiency, as these qualities usually manifest based on specific standard and technology selections, respectively. In these cases, the evaluation functions representing those attributes will be weighted at zero.

Unevaluated Attributes: Functional suitability in this context reflects the ability of the system to accurately manage artifacts based on policy and context. As the functional domain is the same for all examined systems, we will neglect this and set $w_e = 0$.

These are purely notional models, and as a result cannot be evaluated using generally accepted usability metrics like the Systems Usability Scale (SUS) [26]. Ergo, for this analysis we will set the weight $w_u = 0$.

Performance efficiency is generally a characteristic of logical and physical system architectures. Specific logical architectures can certainly decrease the performance of a given system through poor design, insufficient caching, badly considered state management, or inferior scalability. Physical architectures clearly impact performance if processing power, storage, communication bandwidth, or other attributes are insufficient to process apparent loads. We will therefore set $w_p = 0$ in this analysis.

One way to evaluate system portability is via standard compliance [24]. Proposed architectures herein will not be evaluated to the level at which proposed standards have an impact. Therefore, in this analysis, $w_f = 0$.

Compatibility with existing systems requires either accepted system standards or other systems with which to be compatible. In this analysis, we have neither accepted standards or other systems, so we will set $w_c = 0$.

Evaluated Attributes: In the scope of this paper, we will evaluate system reliability by evaluating the information entropy of a given system in which failure is modeled using a Weibull distribution over selected systems elements emulating a burn-in model with faults decreasing over time[27]. The function f_r for a given architecture will be a functional representation of the entropy of the component distributions.

Information entropy here is used to succinctly evaluate the stability of a given system component. In this context, the higher the entropy, the more unstable a given system is as the randomness inherent in that system leads to a higher required communication bandwidth to enumerate a given possible state [28]. This approach is also more succinct

<i>Name</i>	<i>Description</i>
ϕ	The initial level of this taxonomy, ϕ classified systems have a single guard without policy-based control
α	α classified systems have a single guard by have begun to integrate policy-based control
β	Systems that have begun to integrate policy-based control with router elements are in the β category
γ	Systems that have integrated policy-based control with routing and computational elements
δ	Continuous policy-based control with <i>smart licensed</i> artifacts

Table I
PROPOSED USAGE MANAGEMENT TAXONOMY

for this context as it requires less specific knowledge with respect to precise component behavior under load when compared to other approaches like Bayesian Networks and is more quantitative than fault tree analysis [29]. Other software specific models require either *a priori* knowledge of components or neglect the risk associated with higher complexity imparted by additional system componentry [30], [31].

Maintainability can be measured via examination of a given system with an eye toward areas prone to change. Loosely coupled components directly contribute to the ability to change a specific system component. With this in mind, we can measure develop a simple ratio to indicate the maintainability of a given system — the total number of components with high change impact to the number of components with high change impact that have been decoupled from the system via an interface

$$f_m = c_i / c_\delta \quad (4)$$

We will evaluate security of a given system via attack surface analysis of the system itself as well as the protected domains.

Any weights not explicitly set to zero previously will be set to one, giving us:

$$W = \{0, 1, 0, 0, 1, 0, 1, 0\} \quad (5)$$

or, in essence:

$$s = f_r + f_m + f_s \quad (6)$$

B. ϕ -level Overlay Systems

The ϕ classification consists of systems like the initial NSA and BAH notional models. These systems consist of two distinct domains, separated by a filter-centric single guard. The initial NSA system model is clearly of this type, separating two domains with a guard using filter chains. The

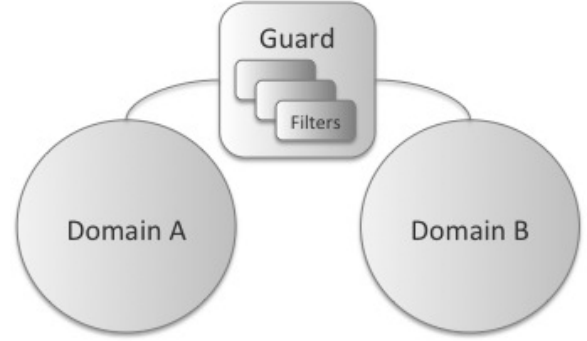


Figure 1. Taxonomy (ϕ)

BAH model is also of this type, using a Filter Segment to evaluate data packages transmitted between interface segments attached to specific domains.

Generally one of the domains supports more sensitive information than the other, but that is not always the case. In the models we have examined this has certainly been true, but classified information for example is commonly stored in *compartments* which are separated by clear *need-to-know* policies enforced by access lists and classification guides. These kinds of compartments contain information at similar levels of classification, but contain distinct informational elements that should not be combined.

In these kinds of systems, specific rules regarding information transfer and domain characterization are tightly bound to individual filter implementations. They are based on *a priori* knowledge of the domains the guard connects, and therefore are tightly coupled to the domains they connect. Furthermore, the filter elements are standalone within the system, in this classification, not availing themselves of external resources. Rather, they examining information transiting through the filter based purely on the content of that information.

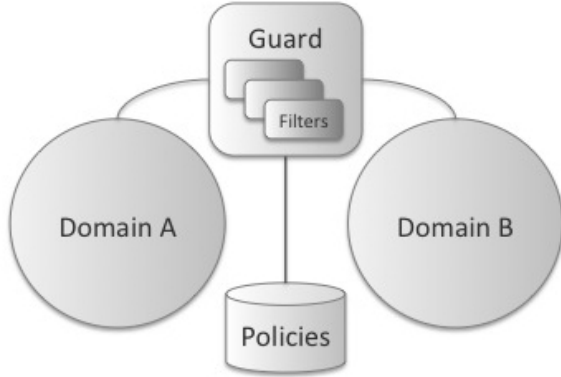


Figure 2. Taxonomy (α)

C. α -level Overlay Systems

The α overlay classification contains systems that have begun to integrate policy-centric usage management.

Both policies and contexts are dynamically delivered to the system.

The dynamic delivery of context and policies allows these kinds of systems more flexibility with policy evaluation.

The α category begins to integrate policy-centric management rather than using strict content filtering.

Here, we again have at least two domains, Domain A and Domain B, though we could potentially have more. ϕ type systems require domain specific information to be tightly coupled to the filter implementations. Separating the permissions, obligations, and other constraints from the filters and incorporating them into a specific separate policy entity frees the Guard from this coupling and provides additional flexibility to the system.

The guard can continue to use filters to process data. These filters however are now more generic and decoupled from the specific domains it manages. The choice of using a specific filtering model rather than some other kind of construct is a design detail level to implementers. That said however, individual filters will be remarkably different and still need to understand the ontologies over which specific licenses are defined.

The policy repository is key to the implementation and differentiation of this taxonomy category. This repository can be implemented as a separate repository keyed into via a data artifact's unique URI, for example. It could also represent a policy sent in tandem with a data artifact in a data package.

The policy repository may be implemented as some kind of external service, and as such, represents the first such external service explicitly used in this taxonomy. Other external services may well exist and be used to adjudicate information transfer decisions as well.

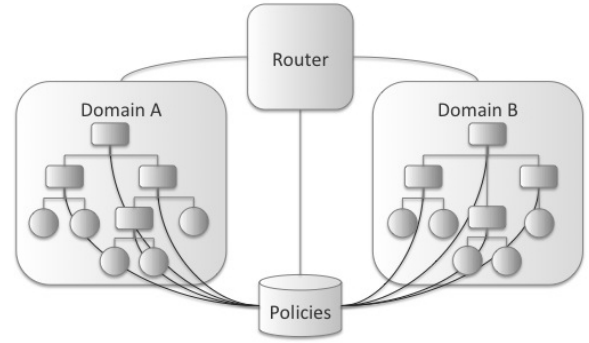


Figure 3. Taxonomy (β)

D. β -level Overlay Systems

The β taxonomic category begins to integrate policy-centric processing with router elements in a given network. While this work is centered on using overlay technology to illustrate and implement these concepts, it is important to note that this kind of distributed policy-centric processing could very well be distributed into the physical routing fabric of a given network as well by extending Software Defined Networking systems like OpenFlow [32].

In this model we can also host multiple domains as a result of flexible policy-based content examination. Each domain hosts a network of some kind, though that hosted network could very well be a degenerate network of a single system. Each network hosted in a domain is hierarchical, with specific computational nodes embodied by workstations, tablet computers or mobile devices, and routing points embodied by routers or switches of some kind.

Policy evaluation in this model has begun to penetrate into the routing elements of the specific domain networks. Here, note that we have started to penetrate into the routing fabric of the network by doing content evaluation at router points. Content-based switching networks have been successful in other domains, and such techniques can be used here to provide policy evaluation capabilities.

Certain types of traffic are easier to evaluate than others however. For example, HTTP requests and responses are easier to examine than TCP packets. When examining TCP packets, systems generally require additional context to select an appropriate packet window (e.g. the number of packets cached for examination). HTTP traffic does not usually require this kind of flexibility.

This migration of policy evaluation into the routing fabric provides for enhanced data security and better network management, especially if part of a network is compromised. Now that policy decisions can be made at the router level in a given network, we are starting to have network security in depth rather than simple perimeter protection. This not only provides the ability for additional information protection, but also allows for different compartments holding information

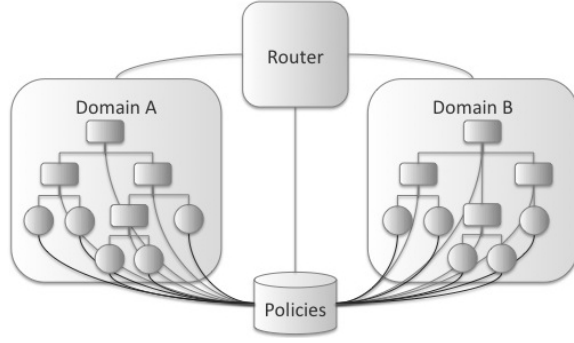


Figure 4. Taxonomy (γ)

at different need-to-know levels to be created ad-hoc under different routing segments. In cases of network compromise, this kind of dynamic policy enforcement can also allow for quick node excision as well.

E. γ -level Overlay Systems

The γ compartment has integrated policy evaluation with compute and routing nodes. Here, policies can be evaluated against content at all network levels — nodes emitting requests, nodes fielding requests, and all routing elements in between.

We see that the policy repository is supplying services to all computational elements in both domains. This gives us increased granularity with respect to data compartmentalization by integrating information security into each network element. At this point, the network can create compartments of single nodes, while previously in β level systems compartments could only be created under specific routing elements. At this level, we can also provide services revoking data access based on policy evaluation decisions when needed.

Furthermore, individual node exclusion is possible as well. β classified systems could excise network elements under specific routers by dynamic policy application. Now, we can apply the same functionality to individual compute nodes. For example, if a networked device like a smart phone is compromised, that device can be removed from access quickly or used to feed mis-information to adversaries.

IV. TAXONOMIC ANALYSIS

Now that we have developed a quality model and defined the taxonomic elements describing a migration from essentially no usage management to pervasive usage management, we can begin to apply that model to the elements themselves. We will begin by applying the model to ϕ elements, proceeding through the taxonomy to γ elements. In each case, we will describe the notional architecture including any inferred services required for application of our usage management model. Then we will apply the quality model from III-A. Finally, we will compare results

of the analysis and extract implications with respect to appropriate conditions for specific architecture application.

In each analysis, a failure is either not passing through acceptable content, or passing through unacceptable content. Furthermore, handing content to review of any kind is considered a failure if that content is acceptable for transmittal, and a success otherwise.

A. ϕ Classification Analysis

ϕ classified systems are remarkably simple and self-contained, but not invulnerable to failure, and can be very difficult to maintain.

A typical system in this classification receives content, with associated meta-data, passes that content to a chained group of filters, which then pass the content onto a dispatcher if it passes filter evaluation. If it does not pass filter evaluation, it is then held for future review.

From a maintainability perspective, immediate points of possible volatility include filter components, receptor components, and dispatcher components. Filters are generally expected to be the most vulnerable to required change as stakeholders create additional filter rules that must be deployed. Filters also require replacement as meta-data formats change. Content receivers and dispatchers may need to change as well based on content format changes or content dispatch rules.

In these kinds of systems, none of these components are designed with appropriate interfaces supporting easy upgrade at this level of evaluation. Ergo, in this case, $f_m = 0$.

B. α Classification Analysis

C. β Classification Analysis

D. γ Classification Analysis

V. CONCLUSION

The conclusion goes here. this is more of the conclusion

REFERENCES

- [1] "DoD Information Sharing Strategy," <http://cio-nii.defense.gov/docs/InfoSharingStrategy.pdf>, May 2007.
- [2] "NSA Pursues Intelligence-Sharing Architecture," <http://www.informationweek.com/news/government/cloud-saas/229401646>, April 2011.
- [3] S. Pearson and A. Benameur, "Privacy, security and trust issues arising from cloud computing," in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, 30 2010-dec. 3 2010, pp. 693–702.
- [4] "Assured Information Sharing in Clouds," <http://www.zyn.com/sbir/sbres/sttr/dod/af/af11-bt30.htm>, August 2011.

- [5] M. S. Blumenthal and D. D. Clark, "Rethinking the design of the internet: the end-to-end arguments vs. the brave new world," *ACM Trans. Internet Technol.*, vol. 1, pp. 70–109, August 2001. [Online]. Available: <http://doi.acm.org/10.1145/383034.383037>
- [6] R. Ross, "Next generation risk management." Presented at the Unified Cross Domain Management Office Conference, 2009.
- [7] M. S. Blumenthal and D. D. Clark, "Rethinking the design of the Internet: The end-to-end arguments vs. the brave new world," *ACM Transactions on Internet Technology*, vol. 1, no. 1, pp. 70–109, Aug. 2001.
- [8] D. D. Clark, "The design philosophy of the DARPA internet protocols," in *ACM SIGCOMM*, Stanford, CA, Aug. 1988, pp. 106–114.
- [9] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden, "Tussle in cyberspace: Defining tomorrow's internet," in *SIGCOMM*, Pittsburg, Pennsylvania, USA, Aug. 2002, pp. 347–356.
- [10] J. Park and R. Sandhu, "The UCON_{ABC} usage control model," *ACM Trans. Inf. Syst. Secur.*, vol. 7, no. 1, pp. 128–174, 2004.
- [11] P. A. Jamkhedkar, G. L. Heileman, and C. Lamb, "An Interoperable Usage Management Framework," in *Proceedings of the Tenth ACM Workshop on Digital Rights Management*, Chicago, Oct. 2010.
- [12] P. A. Jamkhedkar, G. L. Heileman, and I. Martinez-Ortiz, "The problem with rights expression languages," in *Proceedings of the Sixth ACM Workshop on Digital Rights Management*, Alexandria, VA, Nov. 2006, pp. 59–67.
- [13] P. A. Jamkhedkar and G. L. Heileman, "Digital rights management architectures," *Computers Electrical Engineering*, vol. 35, no. 2, pp. 376–394, 2009.
- [14] —, "DRM as a layered system," in *Proceedings of the Fourth ACM Workshop on Digital Rights Management*, Washington, DC, Oct. 2004, pp. 11–21.
- [15] G. L. Heileman and P. A. Jamkhedkar, "DRM interoperability analysis from the perspective of a layered framework," in *Proceedings of the Fifth ACM Workshop on Digital Rights Management*, Alexandria, VA, Nov. 2005, pp. 17–26.
- [16] R. H. Koenen, J. Lacy, M. MacKay, and S. Mitchell, "The long march to interoperable digital rights management," *Proceedings of the IEEE*, vol. 92, no. 6, pp. 883–897, 2004.
- [17] "Coral consortium whitepaper," Tech. Rep., Feb. 2006, www.coral-interop.org/main/news/Coral.whitepaper.pdf.
- [18] "Marlin architecture overview," Tech. Rep., 2006, <http://www.marlin-community.com>.
- [19] G. Perez, F. Clemente, and A. Skarmeta, "Building and managing policy-based secure overlay networks," in *Parallel, Distributed and Network-Based Processing, 2008. PDP 2008. 16th Euromicro Conference on*, feb. 2008, pp. 597–603.
- [20] NSA, "Distributed service oriented architecture (soa)- compatible cross domain service (dscds) dscds overview." Presented at the Unified Cross Domain Management Office Conference, 2009.
- [21] "Department of Defense Global Information Grid Architectural Vision," <http://cio-nii.defense.gov/docs/GIGArchVision.pdf>, 2011.
- [22] Booz, Allen, and Hamilton, "Distributed service oriented architecture (soa) compatible cross domain service (dscds)." Presented at the Unified Cross Domain Management Office Conference, 2009.
- [23] J. Ostermann, "Raytheon dscds intro." Presented at the Unified Cross Domain Management Office Conference, 2009.
- [24] M. Lepmets, E. Ras, and A. Renault, "A quality measurement framework for it services," in *SRII Global Conference (SRII), 2011 Annual*, 29 2011-april 2 2011, pp. 767–774.
- [25] "ISO/IEC 25020:2007," http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue.html, January 2012.
- [26] "Measuring Usability with the System Usability Scale (SUS)," <http://www.measuringusability.com/sus.php>, January 2012.
- [27] L. Fiondella and S. Gokhale, "Software reliability model with bathtub-shaped fault detection rate," in *Reliability and Maintainability Symposium (RAMS), 2011 Proceedings - Annual*, jan. 2011, pp. 1–6.
- [28] T. M. Cover and J. Thomas, *Elements of Information Theory*. Wiley, 1991.
- [29] M. Rausand and A. Høyland, *System Reliability Theory: Models, Statistical Methods and Applications Second Edition*. Wiley-Interscience, 2003.
- [30] Y. Wan and C. Wu, "Software reliability model based on stochastic theory," in *Biomedical Engineering and Informatics, 2009. BMEI '09. 2nd International Conference on*, oct. 2009, pp. 1–5.
- [31] X. Hu and S. Xu, "A synthesis software reliability model," in *Dependable Computing, 2009. PRDC '09. 15th IEEE Pacific Rim International Symposium on*, nov. 2009, pp. 31–36.
- [32] "Openflow - Enabling Innovation in Your Network," <http://www.openflow.org>, November 2011.