Overlay Networks for Usage Management

by

Christopher C. Lamb

B.S. Mechanical Engineering, New Mexico State University, 1993
M.S. Computer Science, University of New Mexico, 2002
Ph.D. Computer Engineering, University of New Mexico, 2012

Abstract

Overlay networks have become a widely used technology with examples ranging from consumer focused distribution systems like BitTorrent to commercial content distribution systems like Akamai. These kinds of systems, with the appropriate policy-centric content management components, can address looming problems in information distribution that both companies and federal agencies are beginning to face with respect to sensitive content. This work addresses the current state of the art in these kinds of cross-domain systems, reviewing current example system architectures from the Unified Cross Domain Management Office (UCDMO), a federal organization specifically tasked with addressing these issues. It then covers other related work, introduces a taxonomy of types of policy-centric usage managed overlay network systems and an associated methodology for evaluating the individual taxonomic elements. It then delves into experimental evaluation of the various defined architectural options and finally presents results of comparing experimental evaluation with anticipated results.

Contents

List of Figures

List of Tables

1 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 re-imagined to take advantage of radical market shifts in computational provisioning.

2 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 [**?**]. Likewise, the NSA is focused on exploiting cloud-centric systems to facilitate information dissemination and sharing [**?**].

Cloud systems certainly exhibit economic incentives for use, providing cost savings and flexibility, but they also have distinct disadvantages as well. Specifically, the 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 [**?**].

To begin with, cloud technology is not currently as private as some organizations would like:

- *User Data Control* — In virtually any given Software-as-a-Service (SaaS) scenario, user data controls are sadly lacking. Once data has been committed to a specific provider, that data is completely out of the original

data owners control. Furthermore, as we will see below, that data my not even be solely owned by the original owner anymore either.

- *Secondary Use* — Most consumer facing social systems extensively mine user provided data for additional business advantages. This is a common and well known secondary use for supplied data. SaaS providers again have strong incentives to examine user provided information.

- *Offshore Development* — Service users have no real control over who actually develops the systems a given service deploys. Organizations have attempted to contractually limit development and support functions companies pursue to, say, the continental United States but have had very poor results with these kinds of unsupportable arrangements.

- *Data Routing* — Both system providers and system users in fact have little control over routing issues. Prohibiting data routing through sensitive countries is a difficult task for a single organization.

- *Secondary Storage* — Most large-scale systems expect to use Content Delivery Networks (CDNs) to help manage content, and that expectation is heavily reflected in their physical system architectures. They simply cannot divorce use of CDNs from their systems for a single organization.

- *Bankruptcy and Data Ownership* — Ownership and obligation to main-

tain expected data arrangements for a given company is not established under bankruptcy [?, ?, ?].

Security issues also emerge from utility computing infrastructures:

- *Data Access* — System users have very little control over who, in the system provider's organization, is able to access their data and systems.

- *Data Deletion* — Most savvy organizations have procedures in place to sanitize old storage elements like disk drives or backup tapes. System users have very little control over if and how this is done when computing services are treated as a utility.

- *Backup Data Storage* — Backup media is very difficult to encrypt, and most system providers still use tape systems as preferred media solutions for backup and storage needs. These tapes, or copies of them, are generally stored offsite to support disaster recovery scenarios. Security of these types of systems has been spotty to date [?, ?, ?].

- *Intercloud Standardization* — Cloud computing systems do not have any standardized way to transfer computational units or data between systems. Any protocols used for this kind of thing must be developed by customers themselves. Due to the desire of providers to lock-in customers, this will likely not change as any standard development is strongly counter-incentiveized.

- *Multi-tenancy and Side-Channels* — Multi-tenant architectures in which multiple customers simultaneously use the same systems open those customers to covert side-channel attacks.

- *Logging and Auditing* — Logging and auditing structures, especially for inter-cloud systems, are non-existent.

Finally, such systems suffer from internal and external trust issues:

- *Trust Relationships* — Trust is difficult to establish between individual cloud providers long-term.

- *Consumer Trust* — Service users are still not entirely trusting of cloud system providers.

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 [**?**]. 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*.

2.1 Current Solutions

Current and near-future proposed solutions endorsed by the UCDMO include system architectures assembled by the NSA, Raytheon, and Booz | Allen | Hamilton (BAH). The NSA has been active in this area for decades as a logical extension of their role in signals intelligence collection and processing. Raytheon and BAH have been engaged over the past few years to provide an alternative voice and design approach to these kinds of systems, an effort met with limited success.

These cross-domain solutions are intended to enable sensitive information to easily flow both from a higher sensitivity domain to a lower sensitivity domain, and from lower to higher as well. They generally act over both primary data (say, a document) and metadata over that primary data as well. Note that in these system, in most cases, human intervention is still required to adequately review data prior to passing into lower security domains.

2.1.1 NSA, Filtered

The NSA conducted initial work in this area. Their standard-setting efforts culminated in a reasonable conceptual system architecture, using groups of filters dedicated to specific delineated tasks to process sensitive information

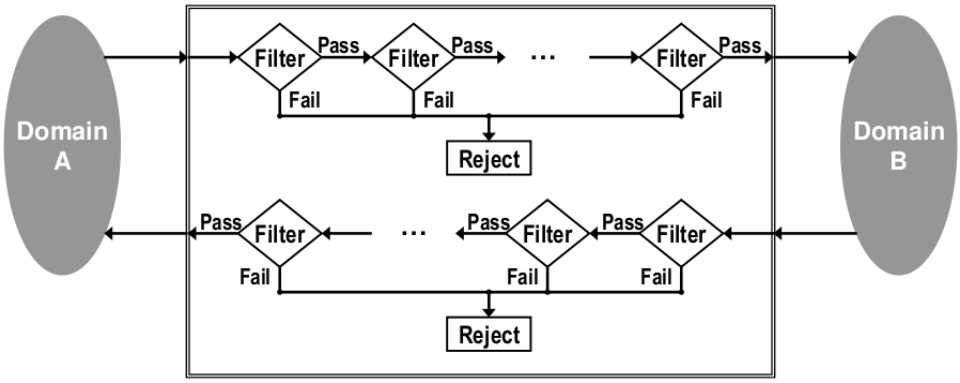

Figure 1: NSA Legacy Notional Architecture Model

[**?**].

In the scenario portrayed in Figure 1, *Domain A* could very well be a private cloud managed by the U.S. Air Force, while *Domain B* is a public operational network of some kind shared by coalition partners in a joint operation.

A system user attempts to send a *data package* consisting of a primary document and associated metadata from *Domain A* to *Domain B*. At some point, that submission reaches a *guard*, which contains at least one *filter chain*. Each filter chain then contains at least one *filter*. Individual filters can execute arbitrary actions over a submitted data package and have access to any number of external resources as required. At any point, a filter can examine the data package and reject it, at which point it will frequently wait for human review. If a filter does not reject a data package, it passes that package onto the next filter or submits it for delivery to Domain B.

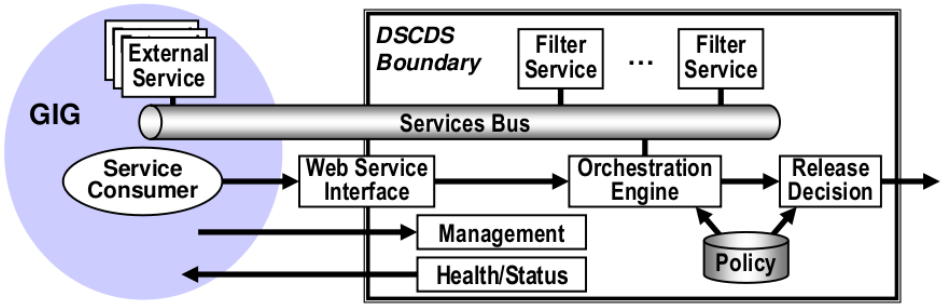

Figure 2: NSA Service-Oriented Model

2.1.2 NSA, Services

In recent years, the NSA has extended the legacy system architecture for cross-domain information sharing to exploit service-oriented computing styles [?]. Visualized in Figure 2, this model incorporates more modern conceptual elements and componentry.

In the view in Figure 2, we see on the left the *Global Information Grid*, or *GIG*. On the right, we have the *Distributed Service-oriented Cross Domain Solution*, or *DSCDS*. The GIG is not a truly open system — rather, it is a loosely coupled collection of computational services handing data at a variety of levels of sensitivity, federated to provide stakeholders timely access to relevant information [?]. The DSCDS is essentially the embodiment of the NSA's cross-domain vision applied to service oriented computing. This model fuses various technology choices with previous cross-domain thinking.

Indicative of this more modern system design thinking, we have a variety of services and service consumers attached to a common service bus within the GIG. Within the DSCDS, we have groups of filters implemented as ser-

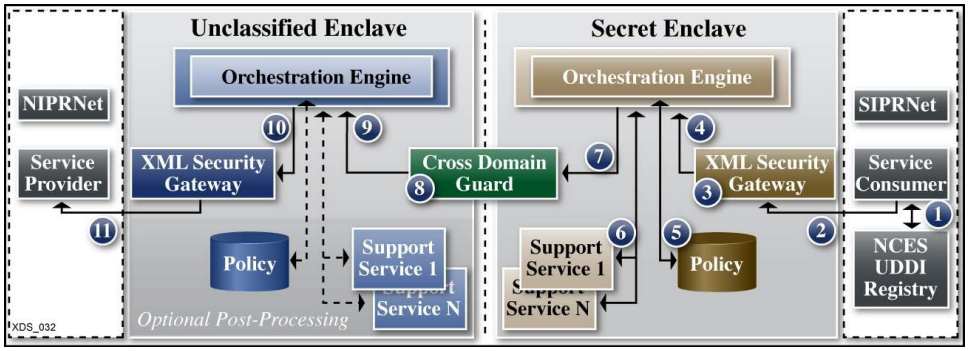

Figure 3: Ratheon Model

vices inspecting transferred data when moved over the bus. Finally, all of this interaction is managed by a management interface and controlled by an orchestration engine accessing a centralized group of policies.

Note that here we have begun to access a common policy repository for various types of security metadata regarding primary data elements.

2.1.3 Raytheon

In the past few years, Raytheon has offered a new model for cross domain use influenced by the NSA service-oriented model [**?**].

The model in Figure 3 is more grounded in the actual technical environment this kind of solution would be embedded within. Here, we have the Non-secure Internet Protocol Router Network (NIPRNet) as one domain, and the Secret Internet Protocol Router Network (SIPRNet) as the other. Here, NIPRNet is the lower security domain (lowside), and SIPRNet the higher security domain (highside). This particular view shows the motion of data from the high side (SIPRNet) to the low side (NIPRNet).

Here, a data request is submitted from SIPRNet first two the *XML Security Gateway* which calls into the *Orchestration Engine* for policy validation. The Orchstration Engine then coordinates calls into a *Policy Repository* as well as to a collection of external *Support Services*. Once rectified against these elements, the request is passed into the *Cross Domain Guard* which routes the request into the *Unclassified Enclave* in NIPRNet. Here, the request is passed directly through the lowside *XML Security Gateway*, without rectification, onto the *Service Provider*. The response from the Service Provider is then passed back to the requester via the inverse path.

This model also begins to use a centralized policy repository, just as the NSA Service Model. It also uses a single cross domain guard to transfer information from both the highside to the lowside, and vice-versa.

2.1.4 Booz | Allen | Hamilton

BAH submitted a competing model, also in 2009 [**?**]. In fact, both Raytheon and BAH presented their models under competitive contract to the UCDMO at the same conference, so the domain application is not coincidental.

Figure 4 embodies BAH's thinking with respect to cross domain information management. We have a *Domain A* as a high security domain, and *Domain B* as a low security domain. Here, we again have dataflow from the highside to the lowside through the cross domain management system.

While not as detailed as the Raytheon proposal, this does have similar elements. Here, we data first travels from Domain A into the *Interface*

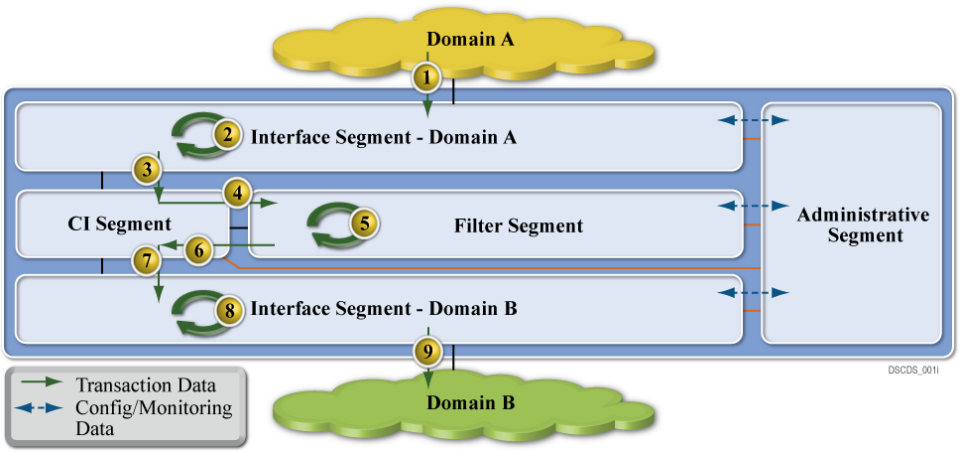

Figure 4: Booz | Allen | Hamilton Model

Segment for Domain A, similar to the secret enclave used in the Raytheon model. From there, it moves into the *CI Segment*, which in turn submits the transferring data into the *Filter Segment*. From there, the package is moved into the *Interface Segment for Domain B*, and then onto *Domain B*. The *Administrative Segment* provides managment and oversight of the system as a whole.

Note the absence of specific policy-centric elements. This system is reliant on specific policy-agnostic content filters as well.

2.1.5 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 [**?**].

2.1.6 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 environ-

ments deployable in cloud systems. Finally, systems must operate under *all* conditions, including when they are under attack or compromise [**?**]. Ergo, they must provide protection to sensitive data in depth.

2.2 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 envrionment. 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 [**?**, **?**, **?**], research developments in the areas of usage control [**?**, **?**], policy languages design principles [**?**], digital rights management (DRM) systems [**?**], and interoperability [**?**, **?**, **?**, **?**, **?**] 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 [**?**]. 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

Name	*Description*
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 *smart licensed* artifacts

Table 1: Proposed Usage Management Taxonomy

is vital to the production development and delivery of overlay systems, but is outside the scope of this work.

3 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 1. 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.

3.1 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 [**?**, **?**].

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.

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

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

$$s = \sum_{W,E} w_i f_i \tag{3}$$

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.

3.2 ϕ-level Overlay Systems

The ϕ classification consists of systems like the initial NSA and BAH notional models in Figures 1 and 4.

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 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.

3.3 α-level Overlay Systems

3.4 β-level Overlay Systems

3.5 γ-level Overlay Systems