Overlay Networks for Usage Management

by

Christopher C. Lamb

B.S., Mechanical Engineering, New Mexico State University, 1994
M.S., Computer Science, University of New Mexico, 2002

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Computer Engineering

The University of New Mexico

Albuquerque, New Mexico

April, 2012

Dedication

This thesis is dedicated my wife Elisa and our two daughters Evelyn and Katherine.

Acknowledgments

I would like to thank my advisor, Professor Gregory Heileman, for his support.

Overlay Networks for Usage Management

by

Christopher C. Lamb

ABSTRACT OF DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Computer Engineering

The University of New Mexico

Albuquerque, New Mexico

April, 2012

Overlay Networks for Usage Management

by

Christopher C. Lamb

B.S., Mechanical Engineering, New Mexico State University, 1994
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

Glossary

RDFa Resource Description Framework – in – attributes

XDM Extensible Metadata Platform

XML eXtensible Markup Language

Chapter 1

Introduction

1.1 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.

1.2 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 [2]. Likewise, the NSA is focused on exploiting cloud-centric systems to facilitate information dissemination and sharing [5].

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 [26].

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 [3]. 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*.

Chapter 2

Cross Domain Examples

2.0.1 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.

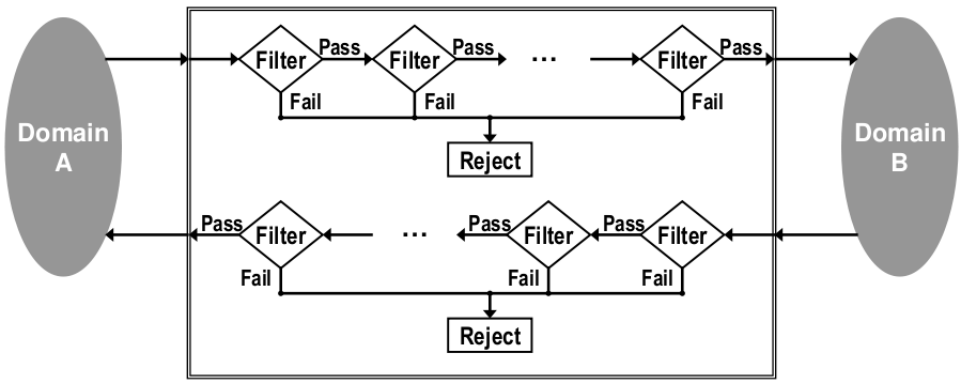

Figure 2.1: NSA Legacy Notional Architecture Model

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 [22].

In the scenario portrayed in Figure 1.1, *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.

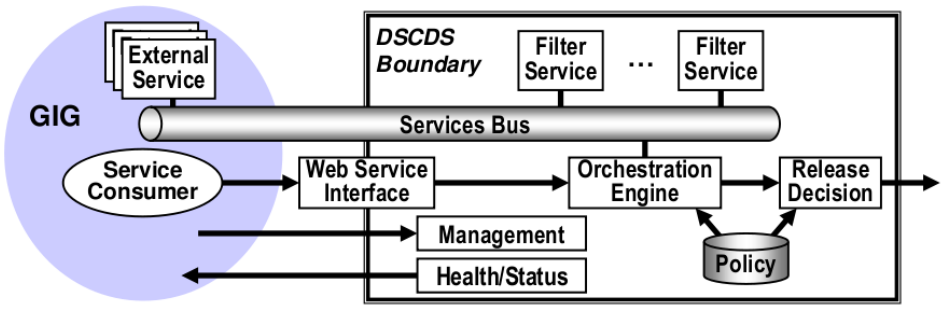

Figure 2.2: NSA Service-Oriented Model

NSA, Services

In recent years, the NSA has extended the legacy system architecture for cross-domain information sharing to exploit service-oriented computing styles [22]. Visualized in Figure 1.2, this model incorporates more modern conceptual elements and componentry.

In the view in Figure 1.2, 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 [4]. 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 services 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.

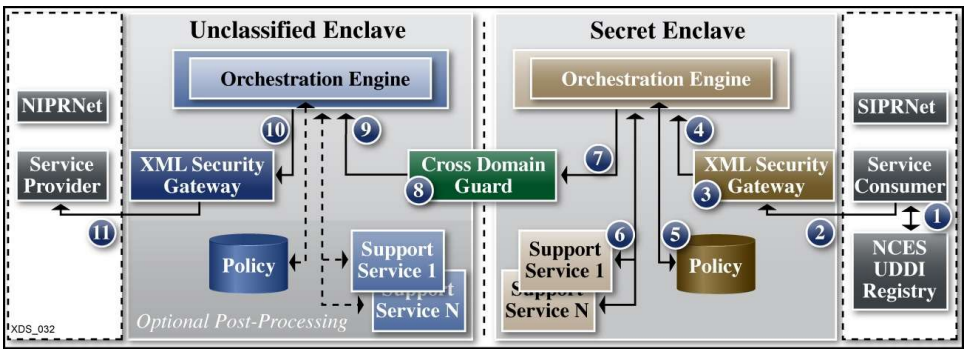

Figure 2.3: Ratheon Model

Raytheon

In the past few years, Raytheon has offered a new model for cross domain use influenced by the NSA service-oriented model [24].

The model in Figure 1.3 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.

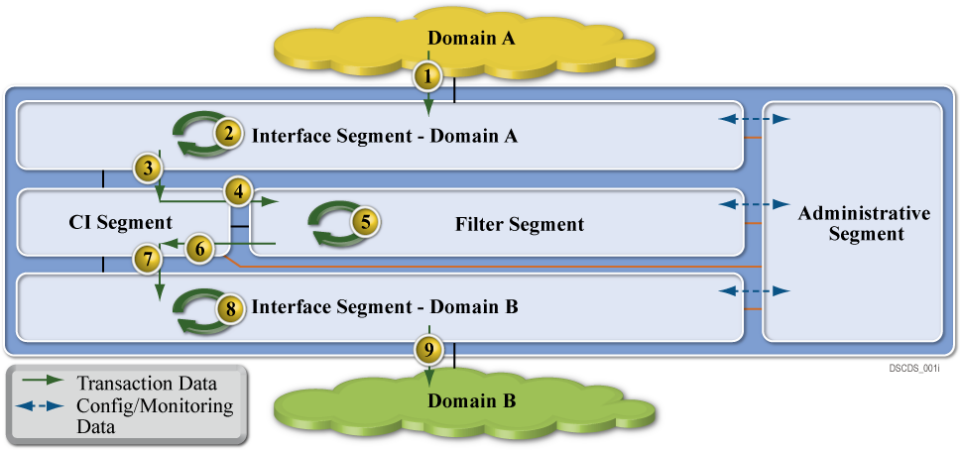

Figure 2.4: Booz | Allen | Hamilton Model

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.

Booz | Allen | Hamilton

BAH submitted a competing model, also in 2009 [11]. 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 1.4 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 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.

Chapter 3

Current Systems

3.0.2 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 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 (BAH) and Raytheon presented alternative notional architectures contrasting with current NSA-influenced approaches [22, 4, 11, 24].

These kinds of cross-domain solutions 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. Users cannot, for example, have Top Secret material on a network accredited for Secret material. Finally, these models violate end-to-end principles in large service network design, centralizing intelligence rather than pushing that intelligence down to the ends of the system [13].

End-to-end principles are generally considered core to the development of extreme scale, distributed systems. Essentially, one of the key design decisions with respect to the early internet was to move any significant processing to system end nodes, keeping the core of the network fast and simple. Known as the end-to-end principles, this design has served the internet well, allowing it to scale to sizes unconceived when originally built. Current cross domain sytems are placed at key routing points between sensitive networks. These locations are core to information transfer between systems and ergo violate the initial design principles upon which the internet was founded. There does exist some belief that end-to-end principles need to be modified to support future networks, but nevertheless, current cross domain systems still violate the basic ideas behind large, scalable networks [9].

Future systems will generally demonstrate decentralized policy management capabilities, infrastructural reuse, the ability to integrate with cloud systems, and security in depth. Policy management will need to be 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 [28]. Ergo, they must provide protection to sensitive data in depth.

3.0.3 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, inter-operable 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 [10, 12, 14], research developments in the areas of usage control [25, 19], policy languages design principles [20], digital rights management (DRM) systems [18], and interoperability [17, 16, 21, 15, 1] 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 [27]. 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 over 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.

Chapter 4

Proposed Taxonomy

4.1 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 1.1. 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

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 4.1: Proposed Usage Management Taxonomy

into the routing fabric, then the computational nodes, and finally by incorporating evaluation directly into content.

The UCDMO has specific goals, with the idea end state described as a flat network architecture with usage management injected into the system at the object level. This is exactly the final δ architecture described within this paper [23, 7]. The UCDMO also has specific goals outlined within it's founding charter, including:

- **Optimize Capabilities** — Drive robust and extensible cross domain capabilities to support a secure and integrated information enterprise.
- **Oversee Resources** — Maximize return on CD cross domain investments, reduce duplication of effort, and increase efficiency of CD activities.
- **Mitigate Risk** — Support risk-based decisions by enabling global awareness of cross domain operational connections.
- **Provide Leadership** — Provide leadership across the interagency spectrum to ensure coordinated cross domain governance, oversight and community reciprocity.

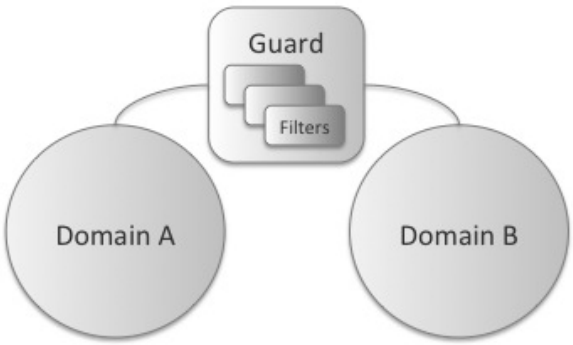

Figure 4.1: Taxonomy (ϕ)

Our work here certainly contributes to these goals, providing robust cross-domain capabilities, helping mitigate risk, and contributing toward advancing the state of the art in this kind of multi-level security environment.

4.1.1 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 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

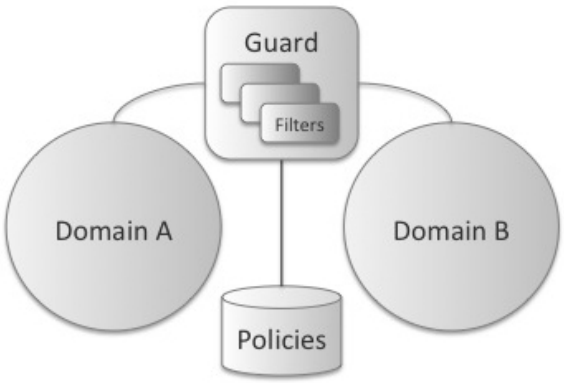

Figure 4.2: Taxonomy (α)

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 those domains. 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.

4.1.2 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 rather than specific content semantics.

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.

4.1.3 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 [6].

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

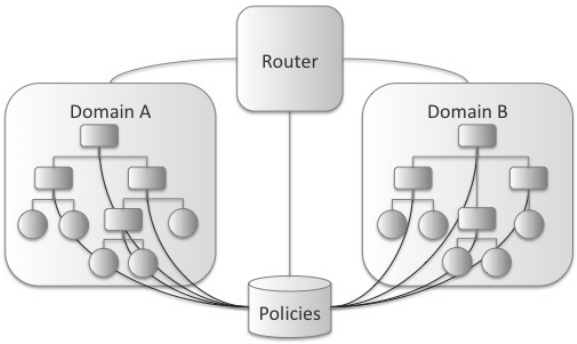

Figure 4.3: Taxonomy (β)

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 [8].

Certain types of traffic are easier to evaluate than others however. For example, HTTP requests and responses are easier to examine that 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 at different need-to-know levels to be created ad-hoc under different routing segments. In cases of network compromise, this kind of dynamic

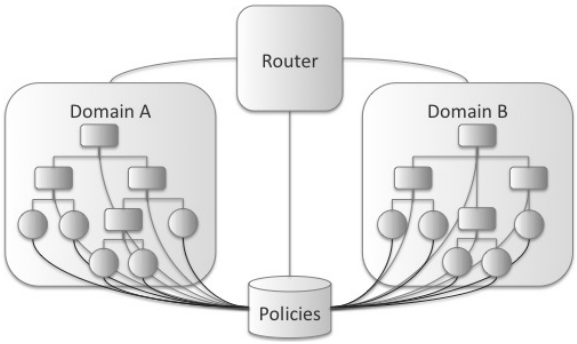

Figure 4.4: Taxonomy (γ)

policy enforcement can also allow for quick node excision as well.

4.1.4 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 suoply mis-information.

4.2 Taxonomic Analysis

The various levels of the taxonomy vary primarily with respect to the inclusion of policy-based usage management and overlay structure. ϕ type systems are not structured with overlay use in mind, nor do they use policy-centric management. Conversely, γ type systems are both purely policy oriented and completely overlay structured.

As systems move through the various levels of the taxonomy they gradually move from one side of the spectrum to another. Overlay structures, hierarchical or otherwise, gradually migrate into the network beginning with β systems. Policy orientation is injected into the architectures starting with α systems and moving into the network fabric in parallel with overlay inclusion.

4.2.1 Characteristics of Policy-centricity

In these systems policy based management supplies distinct advantages over filter-centric information control. This kind of policy-centric usage management is more content specific than filters, more flexible, and is more expressive than filter-centric systems.

Content Specific

Filters, in filter-based systems, are not coupled to the content passing through the system. Rather, they are usually tied to the characteristics of attached networks. For some filters, that is not problematic. Mal-ware filters, for example, are very general and do not need to have an understanding of filtered content and are not sensitive to that content at all, though they can be very sensitive to specific context. This limitation does however prohibit filters from doing anything content specific. Due to their deployment limitations, in that they are deployed to such a system via a process distinct from processing content, they are unable

to use presented content or current dynamic context to influence information processing decisions.

Consider content c impacted by a dynamic context d where d is defined in terms of the content itself, the person or system requesting that content, and the environment in which that request is made. Here, only under certain specific environmental conditions is that requesting agent allowed access to the requested content. Ergo, the decision to pass the content to the requester is based upon characteristics of the content related to dynamic changes within the environment. A filter-centric solution contained within the ϕ level of the taxonomy is unable to change filter rules based on changes like new content or environmental alteration. A policy-based system, on the other hand, is able to express the content specific policy easily for more dynamic evaluation.

For example, if c contains information that can only be accessed for a specific time period, a static filter simply cannot determine that the information in c is no longer appropriate for dissemination after that time period ends. That kind of evaluation requires meta-data associated with c that specifically describes these time bounds and a dynamic contextual evaluator able to determine when that window of access has closed.

Flexibility

Policy-centric systems are more flexible than filter-based counterparts. In a filter-based solution, the type of content that can be evaluated is tightly coupled to the filters installed. If a given piece of content is new to a given filter-centric solution, that content cannot be appropriately examined and must be submitted for human review. A policy-based system is designed to be more general. Based upon a common ontology [19], the evaluation system can be very general with respect to it's evaluation of a given policy. A general policy engine can handle a great variety of different content as long as the policies associated with that content correspond to known domain ontologies. This generality leads to a greater amount

of flexibility with respect to what can be expressed in a specific policy, and leads to a more flexible maintainable system.

A filter is going to have a specific responsibility, like redacting sensitive words from a document, for instance. In order for that filter to redact those sensitive words, it must have access to some kind of list of what those sensitive words are. Remember, ϕ level systems use static filters, so that filter can only be updated when the filter itself is updated. Now a policy-centric system on the other hand can have a policy associating sensitivity with various areas of content in a specific document. In this case, all the system must do is understand the sensitivity described in the policy associated with the content, and can then redact that content if needed. The ontology describing the areas of sensitivity will change more slowly that the possible content itself, leading to a more flexible maintainable system.

This is of course a simple example solvable by creating a dynamic list; the key point of the above example is that the specificity of the filters requires additional complexity in the filter system itself. The generality of the policy-centric system allows the complexity to be more clearly expressed and contained within the policy file.

Expressiveness

While filters can process content at specific perimeter points, it's lack of reach into a given network fabric limits the power a given filter can actually have over transmitted content. A policy associated with content, when transmitted with content, can reference much more than the semantics of the protected content. That policy can describe specifically, in detail, how that content can be used. Filters simply cannot exercise that level of control.

Assume a distributed system with multiple filter points. In this kind of system, information distribution can be controlled via deployed filters at a relatively fine level of granularity. This kind of distribution control cannot influence the use of protected content however — one that content is distributed, possessors are accorded full access.

Policy enabled systems are not limited in this way. Policies, when coupled with policy evaluation tools, can exercise control not only over distribution and routing, but also over use of distributed content at endpoints.

These advantages accrue in usage management systems as policy capabilities are propagated through the overlay fabric. Some of these advantages, like expressiveness, appear simply by beginning to use policies instead of filters. The remaining two have more of an impact as additional policy-centric nodes combine to form an overlay system suitable for cloud deployment increasing their impact as they move from α to δ types of systems.

4.2.2 Overlay Structure

Overlay structure integration exhibits clear advantages over single point perimeter systems as well. Specifically, overlay systems are more partition-able than perimeter solutions, enable content throttling, provide capabilities for dynamic content control, and allow content to be more traceable.

Characteristics of Partition-ability

Administrators typically deploy filter-based perimeter protection at strategic routing points on secure networks. These kinds of networks are designed with specific regions of enhanced sensitivity separated by cross domain management systems regulating information flows [22, 24, 11]. While sensible from the perspective of each protected region as a secure domain, this design thinking begins to fall apart when exposed to the very real threat of the malicious insider. Boundary-centric information flow control is impossible to realistically achieve when the actual boundaries between malicious actors and system users is constantly in flux. When a malicious actor can be anywhere within a system, boundaries are simply too dynamic to be realistically recognized. In order to surmount this fluid system posture,

designers must adopt a security in depth mindset.

Application layer overlay networks enable this kind of defense in depth via the possibility of partitioning. A given overlay system depending on the level of overlay inclusion can partition the user space and by doing so decrease the attack surface available to a malicious insider. ϕ and α level systems based on perimeter filters simply do not present this ability. Systems beginning with β provide the potential to create need-to-know cells of finer granularity up to δ type systems in which cells can be created at the level of specific content. These need-to-know cells serve to help quarantine possible intrusion into the sensitive distribution fabric if that fabric is compromised by helping isolate that system failure within the compromised cell.

For example, assume a hypothetical system with nine nodes connected along a single data plane within an prototypical secure network. With perimeter defenses, if one of those nodes is compromised, a malicious actor can begin to monitor communications traffic between all network nodes, effectively compromising the entire network. In this same network, if designers partition the system into three overlay cells of three nodes, a similar intrusion in one of those cells will effectively only compromise that cell, leaving the other two cells unaffected. This decrease in possible targets for compromise effectively decreased the network attack surface from any give node by $\frac{2}{3}$, correspondingly increasing the security posture of the system.

Content Throttling

Perimeter located filter systems only have the opportunity to control sensitive traffic at that initial boundary. Information located in repositories behind that boundary is not subject to control if it is retrieved by an agent also ensconced behind that same system boundary. Granted, control can be exerted at the repository level, but in a system with more than one repository, this is of limited impact.

A partitioned cell-oriented system, on the other hand, provides greater opportunity for information monitoring and control. The partitions applied atop the physical infrastructure provides additional potential control points requests must cross in order to access needed information. Furthermore, less random cell design provides the capability to unify repositories, providing tight control of information dissemination.

Our hypothetical nine-node system, for example, provides no control over information dispatched from one of the contained nodes to other contained nodes in its initial design form. There are simply no control points within that nine-node network at which to monitor and control information flow. Partitioning that space into three three-node cells provides at least two potential control points for inter-cell requests at which information flow can be monitored. In cases where a malicious insider is actively collecting and hoarding data for exfiltration, these additional control points give system administrators the ability to automatically throttle the rate at which sensitive material can be accessed by users to increase the cost of data collection and increase the likelihood of agent discovery.

Dynamic Content Control

Singular perimeter solutions due to their lack of internal control points also forgo the ability to provide dynamic content control. Once information has traversed a given perimeter access point, it is no longer under the control of that point and can no longer be retrieved, accessed, monitored, or modified. Overlay solutions with internal control points can provide the ability to continually monitor and control disseminated information.

Within a given overlay system, depending on that system structure, data can be more rigorously controlled. β, γ, and δ systems provide the ability to dynamically change information access via contextual changes at a finer grained level than perimeter solutions can. γ and δ systems can in fact provide the ability to retract information access on a per request basis.

This kind of control is especially useful in situations where external partners may temporarily need access to sensitive information for a specific short period of time, say during some kind of joint exercise or activity. γ and δ systems can provide that access only during the window of operation, and retract that access when that window closes. This kind of use is common in joint military operations with coalition partners, for example.

Traceability

The singular location of perimeter filter solutions also precludes easy information traceability. Data requests within a given network sans internal controls is more difficult to trace than an overlay solution with a partitioned cell structure that is tailored to the specific information requested (say, XML databases or semantic web content). The partitioned overlay requires requests to traverse multiple routing nodes at which request and response content can be examined and stored for later analysis and visualization. Perimeter solutions without this kind of structure simply cannot monitor flows at this finer-grained level.

The strengths of overlay systems over single perimeter points gradually increase as overlay structures increasingly permeate any given system. Some abilities, like content-centric access repudiation, can only occur with smart licensed artifacts at the δ level. Others, like traceability or throttling, become more effective as a system architecture traverses from lower to higher levels of capability within the proposed taxonomy.

References

[1] Marlin architecture overview. Technical report, 2006. http://www.marlin-community.com.

[2] DoD Information Sharing Strategy. http://cio-nii.defense.gov/docs/InfoSharingStrategy.pdf, May 2007.

[3] Assured Information Sharing in Clouds. http://www.zyn.com/sbir/sbres/sttr/dod/af/af11-bt30.htm, August 2011.

[4] Department of Defense Global Information Grid Architectural Vision. http://cio-nii.defense.gov/docs/GIGArchVision.pdf, 2011.

[5] NSA Pursues Intelligence-Sharing Architecture. http://www.informationweek.com/news/government/cloud-saas/229401646, April 2011.

[6] Openflow - Enabling Innovation in Your Network. http://www.openflow.org, November 2011.

[7] About the UCDMO. http://www.ucdmo.gov/about.html, January 2012.

[8] JBoss ESB. http://www.jboss.org/jbossesb, January 2012.

[9] Marjory S. Blumenthal and David D. Clark. Rethinking the design of the internet: the end-to-end arguments vs. the brave new world. *ACM Trans. Internet Technol.*, 1:70–109, August 2001.

[10] Marjory S. Blumenthal and David D. Clark. Rethinking the design of the Internet: The end-to-end arguments vs. the brave new world. *ACM Transactions on Internet Technology*, 1(1):70–109, Aug. 2001.

[11] Booz, Allen, and Hamilton. Distributed service oriented architecture (soa) compatible cross domain service (dscds). Presented at the Unified Cross Domain Management Office Conference, 2009.

[12] David D. Clark. The design philosophy of the DARPA internet protocols. In *ACM SIGCOMM*, pages 106–114, Stanford, CA, Aug. 1988.

[13] David D. Clark. The design philosophy of the darpa internet protocols. *SIGCOMM Comput. Commun. Rev.*, 25:102–111, January 1995.

[14] David D. Clark, John Wroclawski, Karen R. Sollins, and Robert Braden. Tussle in cyberspace: Defining tomorrow's internet. In *SIGCOMM*, pages 347–356, Pittsburg, Pennsylvania, USA, Aug. 2002.

[15] Coral consortium whitepaper. Technical report, Feb. 2006. `www.coral-interop.org/main/news/Coral.whitepaper.pdf`.

[16] Gregory L. Heileman and Pramod A. Jamkhedkar. DRM interoperability analysis from the perspective of a layered framework. In *Proceedings of the Fifth ACM Workshop on Digital Rights Management*, pages 17–26, Alexandria, VA, Nov. 2005.

[17] Pramod A. Jamkhedkar and Gregory L. Heileman. DRM as a layered system. In *Proceedings of the Fourth ACM Workshop on Digital Rights Management*, pages 11–21, Washington, DC, Oct. 2004.

[18] Pramod A. Jamkhedkar and Gregory L. Heileman. Digital rights management architectures. *Computers Electrical Engineering*, 35(2):376–394, 2009.

[19] Pramod A. Jamkhedkar, Gregory L. Heileman, and Chris Lamb. An Interoperable Usage Management Framework. In *Proceedings of the Tenth ACM Workshop on Digital Rights Management*, Chicago, Oct. 2010.

[20] Pramod A. Jamkhedkar, Gregory L. Heileman, and Ivan Martinez-Ortiz. The problem with rights expression languages. In *Proceedings of the Sixth ACM Workshop on Digital Rights Management*, pages 59–67, Alexandria, VA, Nov. 2006.

[21] Rob H. Koenen, Jack Lacy, Michael MacKay, and Steve Mitchell. The long march to interoperable digital rights management. *Proceedings of the IEEE*, 92(6):883–897, 2004.

[22] NSA. Distributed service oriented architecture (soa)- compatible cross domain service (dscds) dscds overview. Presented at the Unified Cross Domain Management Office Conference, 2009.

[23] Unified Cross Domain Management Office. Cd101. Presented at the Unified Cross Domain Management Office Conference, 2009.

[24] Jason Ostermann. Raytheon dscds intro. Presented at the Unified Cross Domain Management Office Conference, 2009.

[25] Jaehong Park and Ravi Sandhu. The UCON$_{ABC}$ usage control model. *ACM Trans. Inf. Syst. Secur.*, 7(1):128–174, 2004.

[26] 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*, pages 693 –702, 30 2010-dec. 3 2010.

[27] G.M. Perez, F.J.G. Clemente, and A.F.G. Skarmeta. Building and managing policy-based secure overlay networks. In *Parallel, Distributed and Network-Based Processing, 2008. PDP 2008. 16th Euromicro Conference on*, pages 597 –603, feb. 2008.

[28] Ron Ross. Next generation risk management. Presented at the Unified Cross Domain Management Office Conference, 2009.