Overlay Architectures enabling Cloud Computing for Multi-Level Security Environments

Christopher C. Lamb, Gregory L. Heileman

Department of Electrical and Computer Engineering
University of New Mexico

June 24, 2012

THE UNIVERSITY of
NEW MEXICO

Outline

1. Introduction
2. Motivation — Cloud-centric Usage Management
3. System Architecture
4. Progress
5. Conclusions
6. References

Our Group and Current Work

Our group has roots in digital rights management, machine learning and neural networks, and semantic analysis.

We are working on how to manage sensitive content in computer networks. We are looking at three areas at this point:

- *Routing* — How can we control how sensitive information is routed over the internet? When and why is this appropriate, how can we do it, and where?
- *Redaction* — If we dynamically redact information from transmitted content, what are the implications? how can we go about doing this?
- *Usage Management* — After content has been redacted, how can we control content after delivery? when do we want to do this? why is it appropriate, and how can we implement this?

Simulations ⇒ Cloud ⇒ SDN (Openflow)

What We're Doing

This is a sample of our preliminary work. The accepted paper is six months old, and we've progressed since then.

Today I'm going to cover the submitted paper and touch on current progress as well. This will involve:

- *Covering Motivation and Current Work*
- *Describing Shortcomings of Current State*
- *Discussing Characteristics of Future Solutions*
- *Reviewing our Solution Taxonomy*
- *Discussing Simulations and Current Directions*

Please, any comments with respect to alternative avenues, collaboration, or resource pooling is more than welcome! That's why I'm here.

Introduction

Distributed enterprise computing systems are facing a troubling future. They are:

- *Expensive* — They do not use current commercial resources and use costly partitioning schemes
- *Unreliable* — Too reliant on outmoded security approaches
- *Slow* — Sensitive information is manually reviewed too often leading to the right people being unable to get the right information in time

They need to be re-imagined to take advantage of radical shifts in computational provisioning.

Federal computer systems are a prime example of these kind of problematic distributed systems, and demonstrate the difficulty in implementing new technical solutions.

The Problems — Customer Perspectives

Current policy-centric systems are being forced to move to cloud environments and build much more open systems. Usage management is a key problem in this domain — information needs to be delivered to those who need it as soon as possible:

”...It is imperative to effectively exchange information among components, Federal agencies, coalition partners, foreign governments and international organizations as a critical element of our efforts to defend the nation and execute national strategy...”[1]
— *DoD Information Sharing Strategy*

”...The CIO of the National Security Agency is focusing on IT architecture and a cloud-centric approach to sharing information...”[4]
— *Informationweek*

The Problem — Characteristics

Cloud systems may save money, provide more flexibility, but they also [8]:

- *Are Not Private* — User data control in SaaS is lacking, causing policy concerns for agencies; Data owners have no technical control over secondary use; providers may use offshore development; data can be routed across sensitive countries or secondarily stored on CDNs; data privacy on bankruptcy is ill-defined
- *Are Less Secure* — Controlling data access, data may not be wiped in all XaaS scenarios, availability/backup leads to possible data proliferation, lack of standardization in intercloud communication and data transfer, multi-tenancy and side-channel attacks, difficult logging/auditing
- *Cannot Be Trusted* — Trust relationships, consumer trust

Current Solutions

How are these problems being addressed by impacted organizations?

They're just starting to be actively addressed and are an open research question [2].

Cross-domain architectures are currently the standard for monitoring and information dissemination in an effort lead by the *Unified Cross Domain Management Office*, associated with the Department of Defense (DoD) and the National Security Agency (NSA).

Current Solutions — NSA

Legacy cross-domain notional architecture [6]

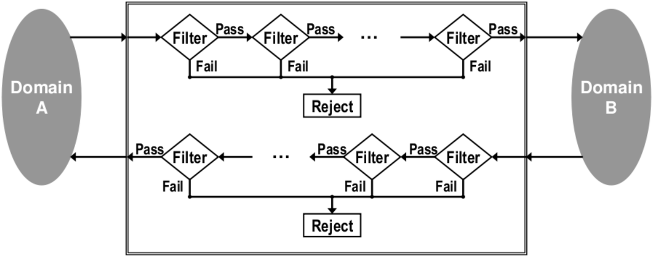


Figure: NSA Legacy Model

Domain A — Private cloud managed by the Air Force
Domain B — A public operational network

Current Solutions — NSA (SoA)

Future cross-domain notional architecture [6]

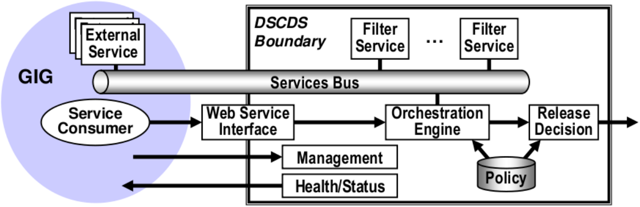


Figure: NSA Service-Oriented Model

GiG — Global Information Grid; a large public cloud operated by the DoD
DSCDS — Distributed Service-oriented Cross Domain Solution

Current Solutions — Raytheon

Raytheon's notional architecture supporting cross-domain information flow [7]:

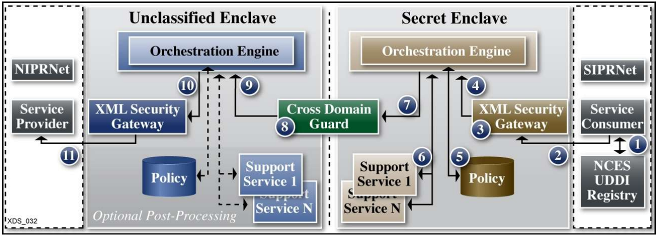


Figure: Raytheon Model

...still uses a single perimeter guard...

Current Solutions — BAH

Booz|Allen|Hamilton presented a service-centric cross domain solution in 2009 [5]:

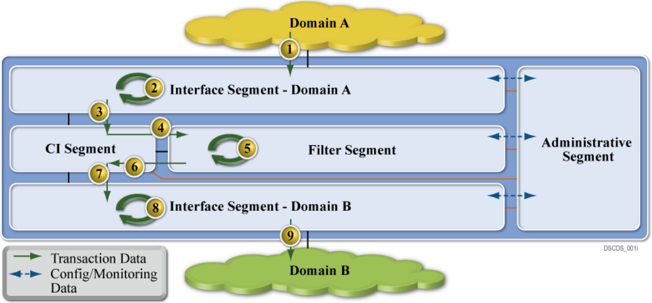


Figure: Booz—Allen—Hamilton Model

...still uses a single perimeter guard (called a filter segment)...

Future Solution

Organizations are falling back on what they know in the scope of new problems.

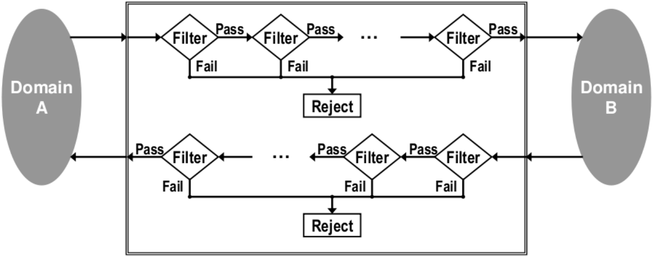


Figure: NSA Legacy Model

Even though we know they don't work [9].

Characteristics of Current Solutions

- *Centralized Policy* — They use centralized policy injection into communication flow. Note that in each sample model, policy is *only* evaluated at guard points.
- *Physical to Compartment Mapping* — In each of these cases, users are only allowed to exchange one type of information per domain. The physical domain systems 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.
- *Perimeter Protection* — The use of a single policy enforcement point at domain interconnects supplies a crunchy exterior to the creamy interior data filling.

What's Wrong with Current Solutions?

- *Centralized Policy* — A centralized policy enforcement system simplifies infrastructural attacks. Adversaries know exactly where to focus efforts to compromise policy enforcement, lowering overall system trustworthiness and reliability.
- *Physical to Compartment Mapping* — The traditional model for multi-level security, enforced in this scheme, is that the network is classified at the level of the most sensitive data that transits it. Ergo, those that have clearances at a level to view sensitive data are unable to view that data generally without extensive swivel-chair integration.
- *Perimeter Protection* — Perimeter protection is a necessary but not sufficient security approach. By itself, it doesn't work [9].

Characteristics of Future Solutions

- *Decentralized Policy* — 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.
- *Infrastructure Reuse* — 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.

Characteristics of Future Solutions

- *Cloud Integration* — 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.
- *Security in Depth* — Systems must operate under *all* conditions, including when they are under attack or compromise [9]. Ergo, they must provide protection to sensitive data in depth.

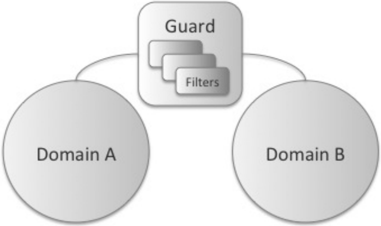
System Architecture

What would this kind of overlay system look like?

- *Meta-Model*
- *Non-Hierarchical Overlays*
- *Hierarchical Overlays*
- *Ontologies and Taxonomies*

...and what would the migration path to these systems look like?

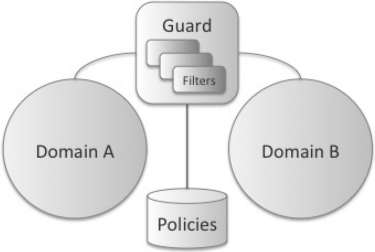
System Architecture - Level ϕ



This is a baseline cross-domain solution. It is filter based and does not have any external policy sources. These are primarily:

- *Filter-centric* — They use content filters of some sort against submitted information.
- *Blacklist-oriented* — They use hard-wired blacklists to filter and redact.

System Architecture - Level α

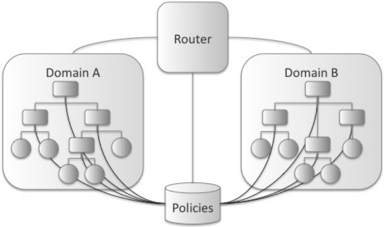


This is a more advanced cross-domain solution featuring distributed policy management. Characteristics include:

- Generalized Control — No longer required to use fixed blacklist-centric solutions, these kinds of systems process policies defined over a more general ontology [1].

[1] Ontological impedance now a problem

System Architecture - Level β

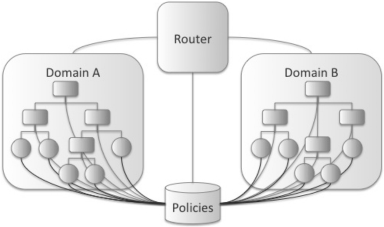


We are beginning to inject usage management into the fabric of the network, linking content routing elements to policy information [2].

- *Content-based Routing* — We can start to implement content based routing at this point with available policy information.
- *Dynamic Context* — We can also start to take advantage of changing network context with respect to routing data.

[2] Not really applicable to pure non-hierarchical systems

System Architecture - Level γ



We now are injecting usage management into every content node within the system. This gives us:

- *Extensive Control* — We can now effectively do things like *rights retraction*.
- *Pervasive Usage Management* — We have content monitoring at all nodes of the content network.

System Architecture - Level δ

Here, we introduce the concept of a *Smart License*:

- *Mobile* — Licenses are small programs that move along the overlay and are run at various policy enforcement points [3]
- *Integrated* — Content, Policies, Usage Management Mechanism all packaged in Smart License
- *Contained* — Content and Policies are never exposed, all access to content is through specific interfaces

Advantages

Potentially more **secure** for content, provides finest-grained **control**; **simpler** routers and nodes

Disadvantages

Mobile code requires **uniform execution environments**, which have their own **security** problems; **complex** license

Costs and Benefits

So we have integrated usage management into a content network taking a very information-centric perspective. At what cost?

Costs

Increasing **attack surface**; all these policy evaluation and injection points are nifty avenues for exploitation

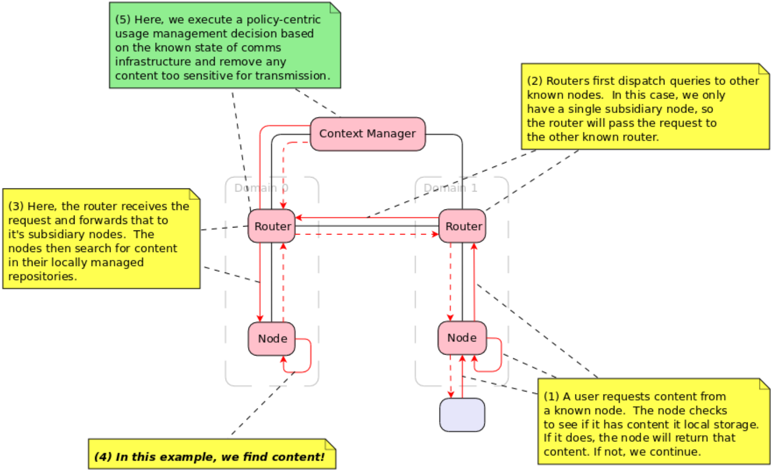
Increasing **complexity**; dynamic routing makes things more difficult and expensive to manage

Benefits

Additional **control** over sensitive content; both how it is **used** and how it is **distributed**

The ability to dynamically apply **new security controls** to transmitted information based on context (e.g. increase the strength of encryption)

Initial Prototype



(1) A user requests content from a known node. The node checks to see if it has content it local storage. If it does, the node will return that content. If not, we continue.

(2) Routers first dispatch queries to other known nodes. In this case, we only have a single subsidiary node, so the router will pass the request to the other known router.

(3) Here, the router receives the request and forwards that to it's subsidiary nodes. The nodes then search for content in their locally managed repositories.

(4) In this example, we find content!

(5) Here, we execute a policy-centric usage management decision based on the known state of comms infrastructure and remove any content too sensitive for transmission.

Current Work

We're currently extending the local simulation to a distributed content system. The environment we're using consists of roughly 40 nodes distributed over Rackspace (using Rackspace Servers), Amazon (using EC2), and our local Eucalyptus installation. We will use Heroku as well (We're deliberately building networks over providers and mixing IAAS and PAAS).

- *Ruby* — We use Ruby and the Ruby runtime as our base runtime engine. We use associated tools like RVM, Gem, and Bundler to manage our projects.
- *Sinatra* — Sintara is a simple but powerful HTTP engine.
- *Capistrano* — Capistrano simplifies deployment of software on large numbers of nodes.
- *YAML* — YAML is a simple data serialization language.

Future Work

We are in the process of configuring infrastructure to begin work with Openflow-enabled hardware. We will use Ruby and Ruby tools in this environment as well, with the addition of Trema, a Ruby and C based environment for building Openflow controllers.

Conclusions

Contribution of Work

The contribution of this work is a quantitative analysis of policy-centric overlay network options, associated taxonomies of use, and prototypical technology proofs-of-concept.

- *Network Control Options* — This includes various types networks and associated strengths and weaknesses addressing centralized and decentralized models.
- *Taxonomies of Use* — Depending on the specific usage management requirements and context, different overlays have different applicability; this work will provide guidance on suitability; it will eventually lead to how to manage data flow within SDN-capable infrastructure.
- *Prototypical Technologies* — Examples and proofs-of-concept will be required to appropriately analyze various architectural alternatives.

Questions? Comments?

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

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

[3] http://www.ietf.org/rfc/rfc3198. http://www.ietf.org/rfc/rfc3198, November 2011.

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

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

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

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

[8] 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.

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