Policy Overlay Networks

Christopher C. Lamb

Department of Electrical and Computer Engineering
University of New Mexico

June 10, 2011

THE UNIVERSITY of NEW MEXICO

Outline

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.

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.

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.

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.

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.

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 [11]:

- *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

The Problem — Characteristics

Cloud systems may save money, provide more flexibility, but they also [11]:

- *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

The Problem — Characteristics

Cloud systems may save money, provide more flexibility, but they also [11]:

- *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

The Problem — Characteristics

Cloud systems may save money, provide more flexibility, but they also [11]:

- *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

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 [9]

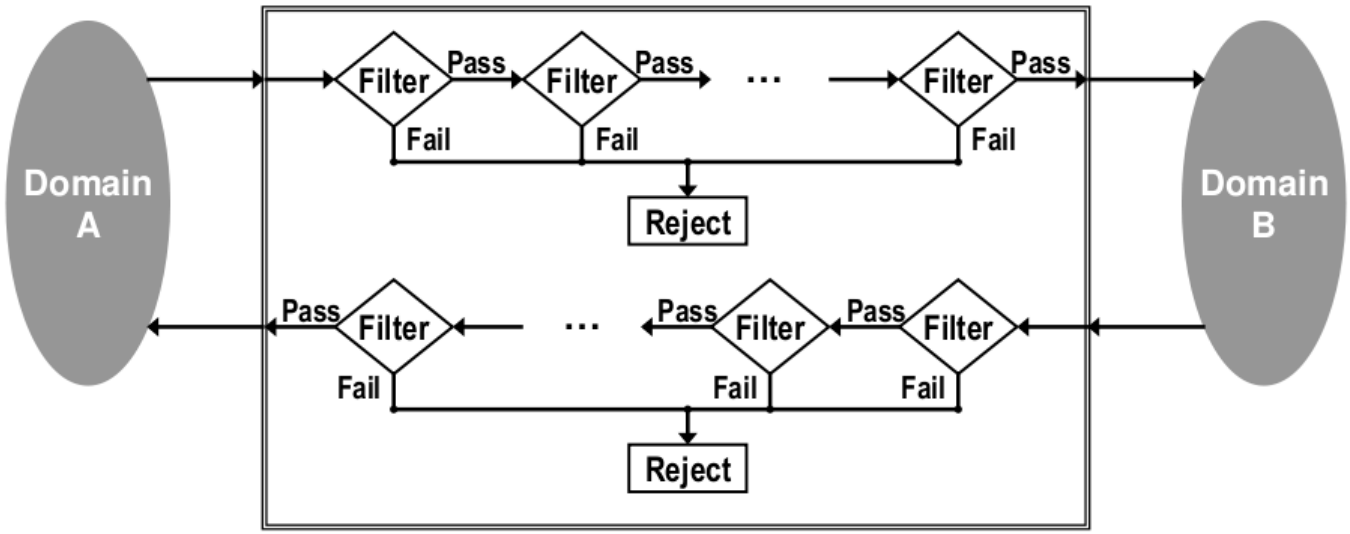


Figure: NSA Legacy Model

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

Current Solutions — NSA (SoA)

Future cross-domain notional architecture [9]

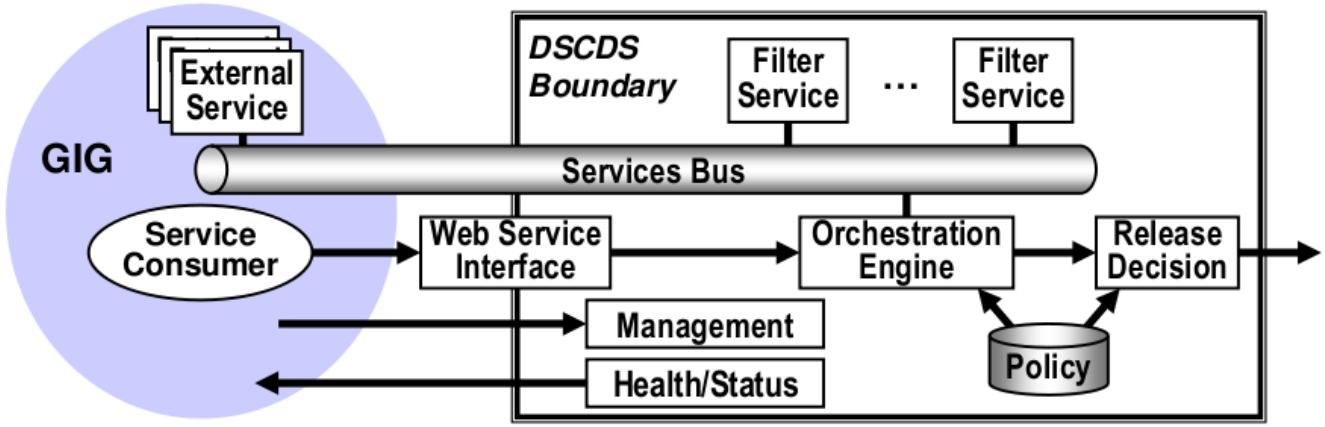


Figure: NSA Service-Oriented Model

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 [10]:

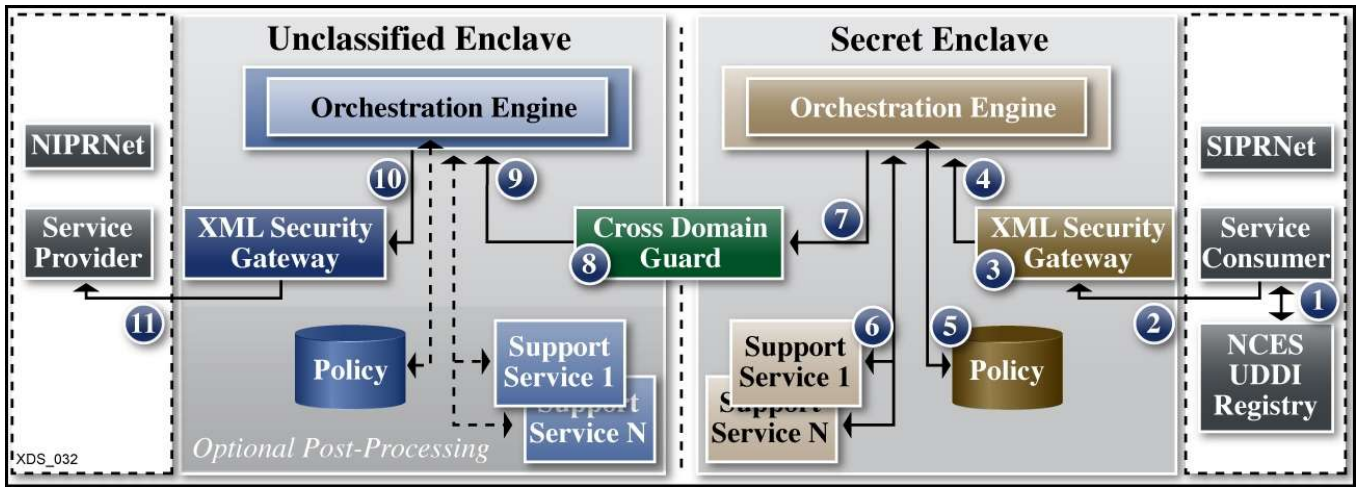


Figure: Raytheon Model

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

Current Solutions — BAH

Booz—Allen—Hamilton presented a service-centric cross domain solution in 2009 [8]:

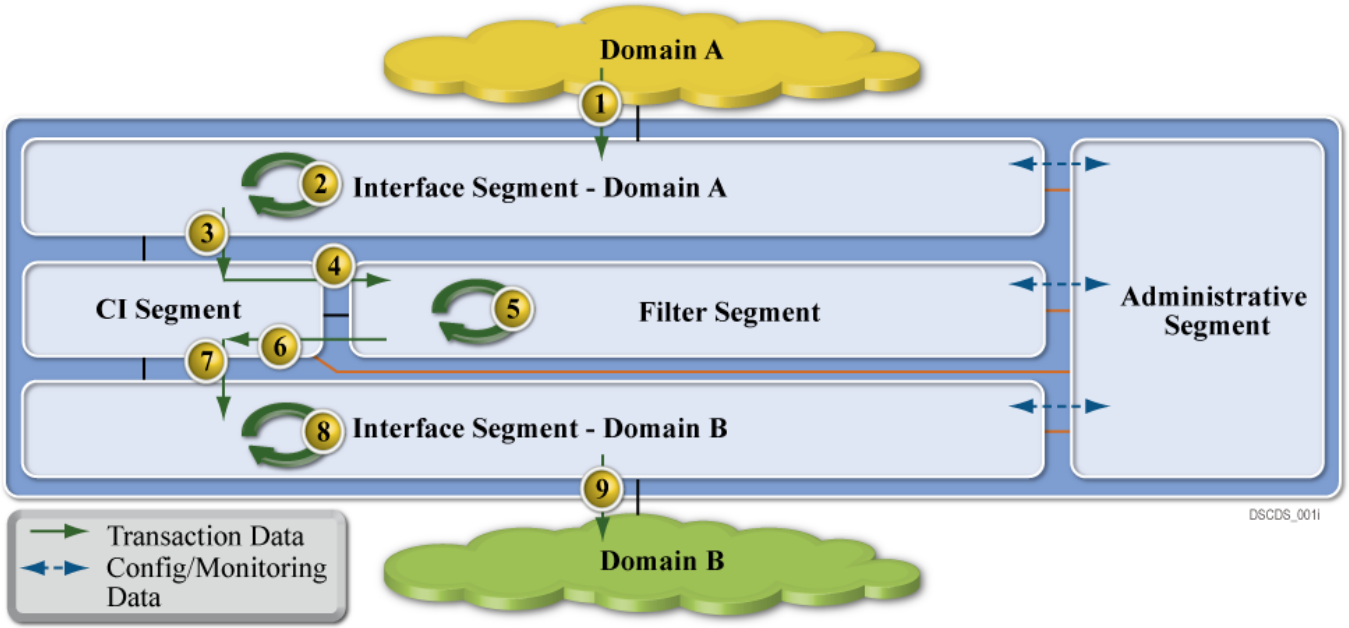


Figure: Booz—Allen—Hamilton Model

...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.

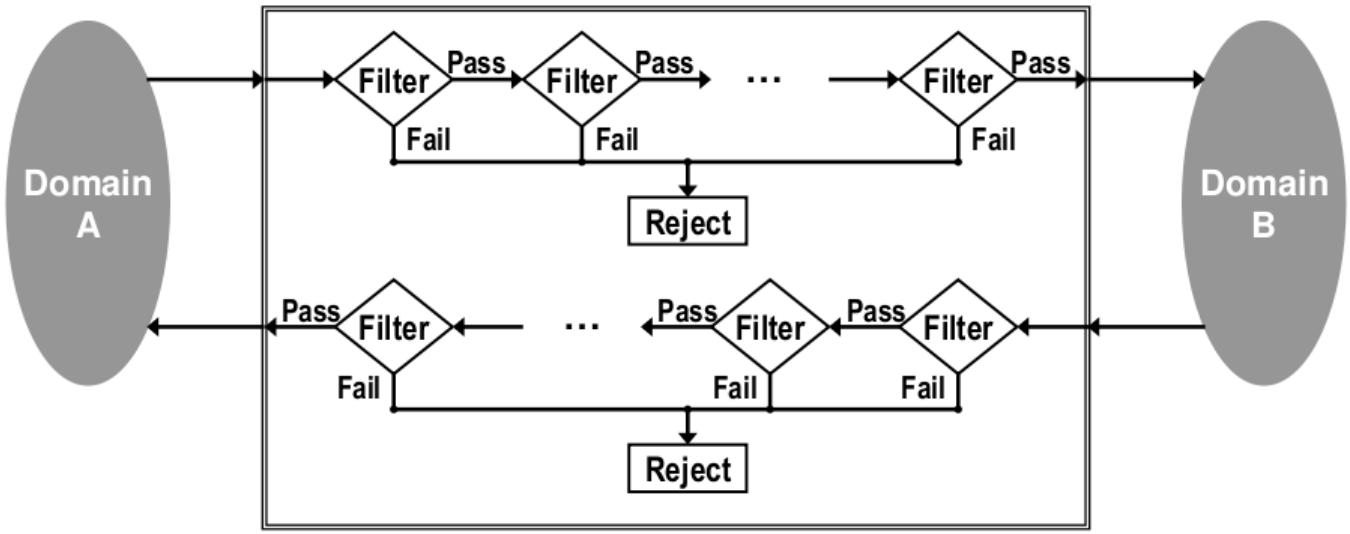


Figure: NSA Legacy Model

Even though we know they don't work [13].

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.

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.

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.

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 [13].

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 [13].

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 [13].

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 [13].

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

- *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

- *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 [13]. Ergo, they must provide protection to sensitive data in depth.

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 [13]. Ergo, they must provide protection to sensitive data in depth.

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 [13]. Ergo, they must provide protection to sensitive data in depth.

Closest Related Work

Protecting domains...

Domains exist below a specific overlay, and trusted secure paths between domains corresponding to a single overlay network are negotiated *a priori* and then used by the overlay [12]

May be useful in this work, but the authors are a bit obtuse on the application of their ideas

Using specific policies in a policy layer to protect the underlying network strata from abuse by overlays [7]

Doesn't really address content-centric policies

...not content.

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

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

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

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?

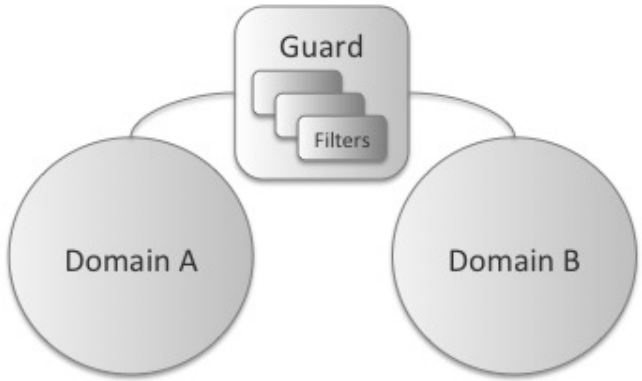
System Architecture

What would this kind of overlay system look like?

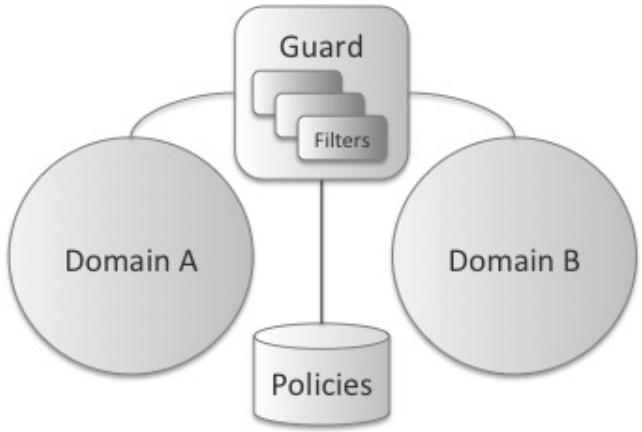
- *Meta-Model*
- *Non-Hierarchical Overlays*
- *Hierarchical Overlays*
- *Ontologies and Taxonomies*

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

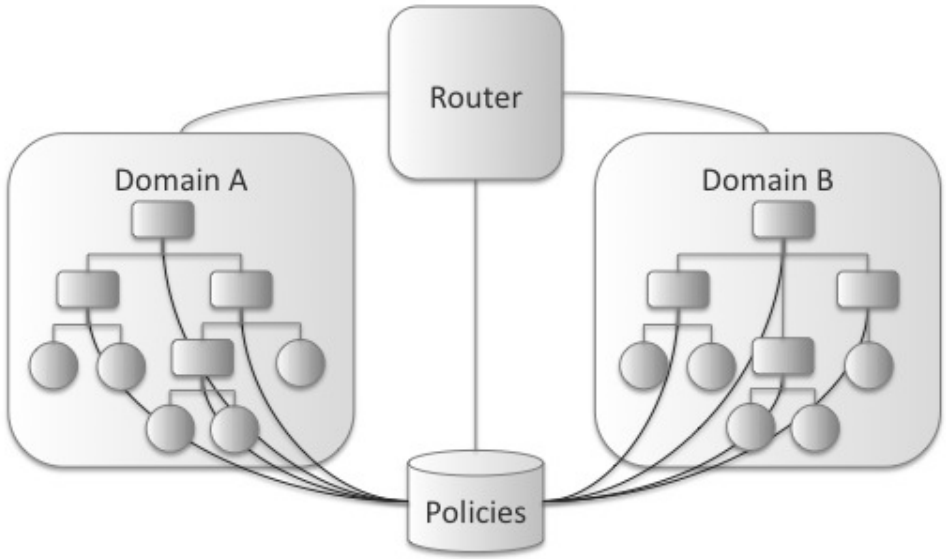
System Architecture - Level ϕ



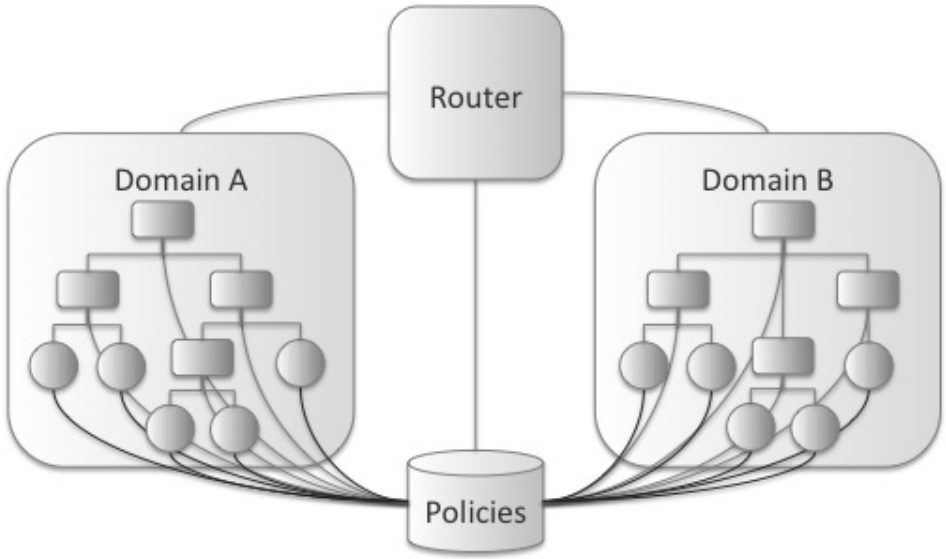
System Architecture - Level α



System Architecture - Level β



System Architecture - Level γ



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

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

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

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

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

Conclusions

Contribution of Work

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

- *Overlay Options* — This includes various types of overlay 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
- *Prototypical Technologies* — Examples and proofs-of-concept will be required to appropriately analyze various architectural alternatives

A Final Note

Software Defined Networking (SDN) is fast becoming an important research and development area [6, 5].

How does this work apply to SDN?

A Final Note

Software Defined Networking (SDN) is fast becoming an important research and development area [6, 5].

How does this work apply to SDN?

Questions?

[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] Open networking foundation. https://www.opennetworking.org/, November 2011.

[6] Openflow - enabling innovation in your network. http://www.openflow.org, November 2011.

[7] I. Al-Oqily and A. Karmouch. Policy-based context-aware overlay networks. In *Global Information Infrastructure Symposium, 2007. GIIS 2007. First International*, pages 85 –92, july 2007.

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

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

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

[11] 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.

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

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