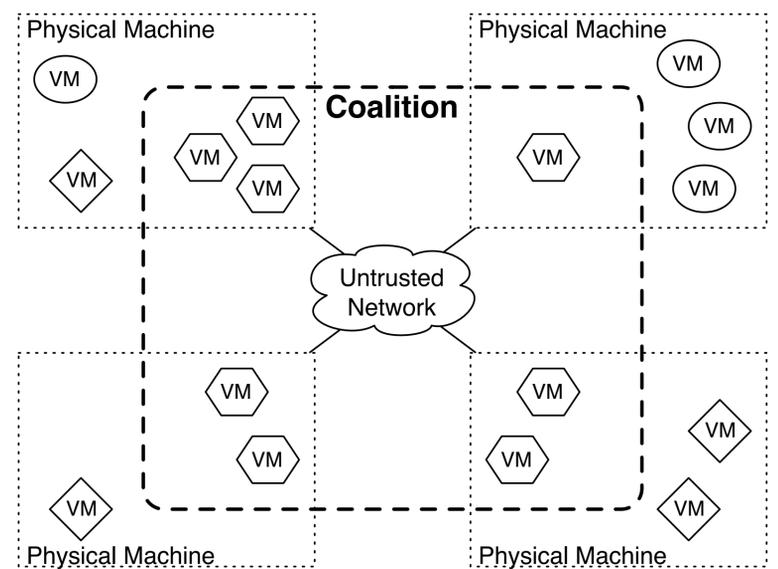


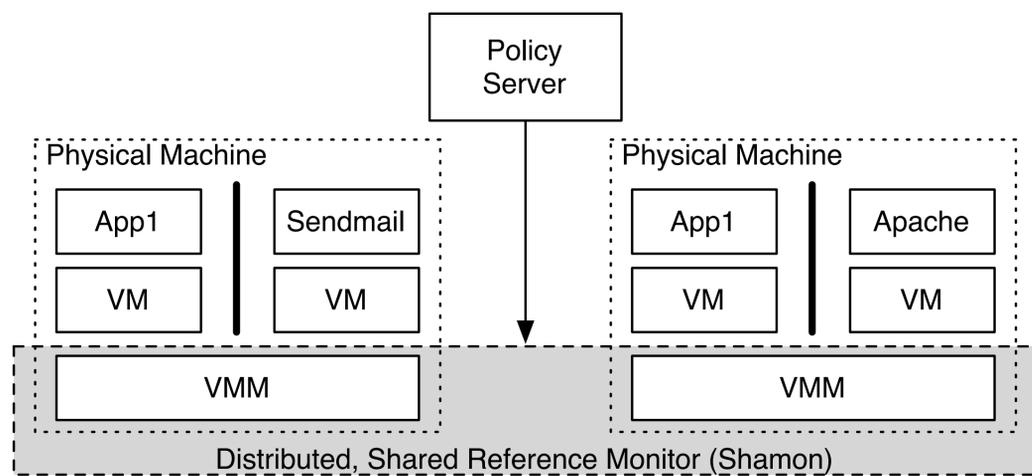
Problem:

We want to develop trust in the enforcement of security goals across many machines on an Internet scale, but fear of malicious administrators, compromised machines, and unwitting leaks of sensitive data make this difficult. Additionally, the complexity of operating systems makes it difficult to say anything meaningful about the security of another system in a traditional setting.

Our goal is to achieve the guarantees of a **reference monitor** (tamperproof, completely mediated, simple enough for verification) in a distributed setting. We would like to establish a **coalition** of virtual machines within which we can make some guarantees about the security of communication and the enforcement of policy. This coalition will be governed by a central authority, called a **Shamon** to enhance scalability and accommodate dynamic changes to the coalition.



Vision



A Distributed, Trusted, Reference Monitor (DTRM) is established to enforce policies on physical machines and virtual machines. This is the entity that is formed to control the coalitions of VMs.

Virtual machines run on the physical machines to provide applications. This allows us to reason at a much coarser granularity than the operating system level to govern sharing and access control.

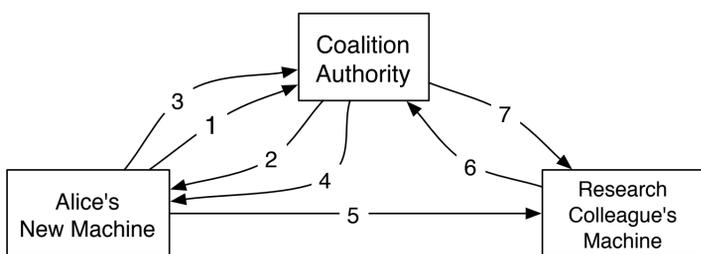
Policies then govern the relationships between VMs, forming coalitions. These policies govern which VMs are in which coalitions, which coalitions can talk to each other, and how they may communicate.

Sharing between VMs is governed by sHype, while SELinux governs how data is transferred from one VMM to another. SELinux policies govern this relationship, but they are much less complex than governing sharing between all the subjects and objects on an operating system.

For this to scale, we must have a way of managing attestations, membership, and trust.

Shamon

A central authority, or Shamon, will be used to manage attestations of the code that runs the VMM



However, this environment is not statically defined. The Shamon must also be able to manage the dynamic state of the system in view of the following state changes:

- Member join/leave
- Suspension/resumption
- Migration of coalition VMs
- MAC policy change at machine level
- MAC policy change at Shamon level
- Change in security goals

Current Work

First, we will create usable attestations that can easily be transmitted and checked by principals in the system. To do this, we will be using **Bloom filters** to reduce the space required for attestations and the complexity required to check them. This helps to make the size and possible values of an attestation manageable.

Then, we will combine these attestations with hooks added to IPsec's dynamic key negotiation to automatically pass **attestations over IPsec**. By doing so, we create trusted, labeled channels over which coalition members can pass information, hypervisors can negotiate trust, and all members can communicate with the Shamon to receive dynamic updates to the state of the system

Once this framework is in place, we will use this to create a **trusted VNC** system in which users can use any untrusted client machine and know that the only way they will be allowed to connect to their VNC session is if the client machine is running "good" software.

Finally, we will also be reasoning about the properties of the enforcement system. The enforcement policy will have to be distributed or reconciled to each machine in the coalition, and **trust relationships** will have to be established between hypervisors. For instance, we will show that the semantics of the trust logic are correctly enforced in the system policy. We also will consider the **consistency, soundness, and completeness** of the reference monitor in order to evaluate its effectiveness.