Security and the Cloud: Cloud Trust Brokers

Ravi Sandhu
Executive Director and Founder
Institute for Cyber Security
Endowed Professor of Cyber Securiy
University of Texas at San Antonio
+1.210.458.6081
ravi.sandhu@utsa.edu
www.ics.utsa.edu
www.profsandhu.com

Ravi Ganesan*
Founder, SafeMashups
+1.415.680.5746
ravi@safemashups.com ravi@findravi.com
www.safemashups.com www.findravi.com

Todd Wolff
Assoc. Dir. and Chief Architect
Institute for Cyber Security
University of Texas at San Antonio
+1.210.458.6998
todd.wolff@utsa.edu
www.ics.utsa.edu

*This work was performed when Ravi Ganesan was a Research Professor at the Institute for Cyber Security at The University of Texas at San Antonio which incubated SafeMashups Inc.

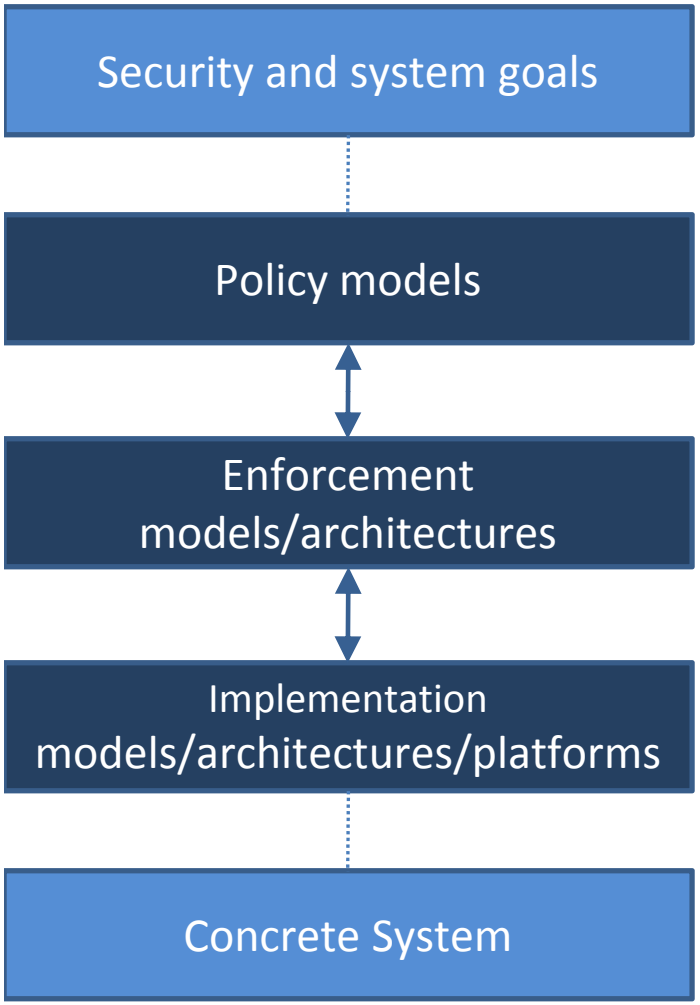
Institute for Cyber Security (ICS)

- **Mission: World-leading research with real-world impact!**
- Founded June 2007: young and agile in start-up style
- World-leading security modeling and analysis research
  - Role-Based Access Control (RBAC) Model: Commercially dominant model today
  - Usage Control (UCON) Model: Attribute-Based Access Control on Steroids
  - PEI layers: Policy (what), Enforcement (how), Implementation (how exactly)
  - Group-Centric Information Sharing: Sharing metaphor of meeting room
  - Security for Social Networks
  - Botnet Analysis, Detection and Mitigation
  - Multilevel Secure Architectures
  - Secure Cloud Computing
- World-leading research infrastructure
  - FlexCloud
  - FlexFarm

The Big Cyber (Security) Trend

EICE	→	ATCE
Enterprise/Infrastructure-Centric Era (Orange/Rainbow Era, Post-Orange Era)		**Application/Technology-Centric Era**
Applications are cyber analogs of previously existing enterprise-centric applications		**Future applications and application layer tecnologies will be fundamentally different**
• on-line banking • brokerage • e-retail • auctions • search engines • …. • payroll • inventory control • accounting • ….		• ? • ? • ? • ? • ? • ? • ? • ? • ?

PEI Layers World-View



- Necessarily informal

- Specified in terms of users, subjects, objects, administrators, labels, roles, groups, etc. in an idealized setting.
- Security analysis (e.g. security objectives, security properties, etc.)

- Approximated policy realized using system architecture with trusted servers, secure protocols, etc. in a real-world setting
- Enforcement level security analysis (e.g. safe approximations with respect to network latency, protocol proofs, security properties, etc.)

- Technologies and standards such as SOA, Cloud, SaaS, TCG/TPM, MILS, X.509, SAML, XACML, Oath, Oauth, etc.
- Implementation level security analysis (e.g. vulnerability analysis, penetration testing, protocol proofs, security properties, etc.)

- Layered software stacks executing on Hardware

Cloud Security: Myths and Reality

- **Myths**
  - Same old, same old

    MULTICS did it all in 1970s
  - It's all new, it's all different

    Let's re-invent all the basics
- **Reality**
  - Cloud Technology intrinsically changes existing security problems

    e.g. "What hardware does your system run on?", a typical security evaluation question becomes irrelevant
  - Cloud Technology enables new applications which bring new security challenges

    e.g. multi-party applications running on a multi-party platform
- **New fundamental problems arise including**
  - How to broker trust across multi-party applications running on multi-party platforms

Cloud Computing: The Broader Context

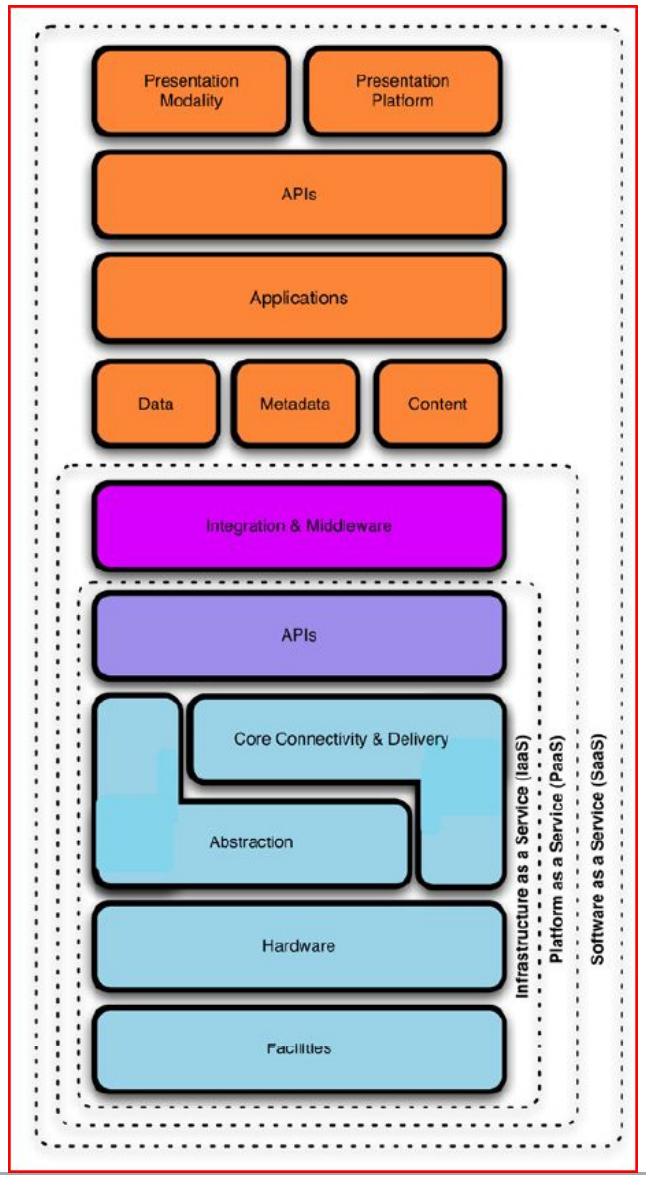
- Macro Trend: We are moving to a new multi-party Internet.
  - Old Paradigm: SSL lock /Green bar gives Alice hints as to authenticity of site

    New Paradigm: 3rd Party Trust Broker advises Alice through browser (e.g. McAfee Site Advisor toolbar) as to quality of site
  - Old Paradigm: Everyone is an Identity Provider (IP) and a Relying Party (RP)

    New Paradigm: Few 3rd Party Trust Broker IPs, everyone else is an RP.
  - Old Paradigm: User gets service from a single web app, perhaps with behind the scenes collaboration from other services

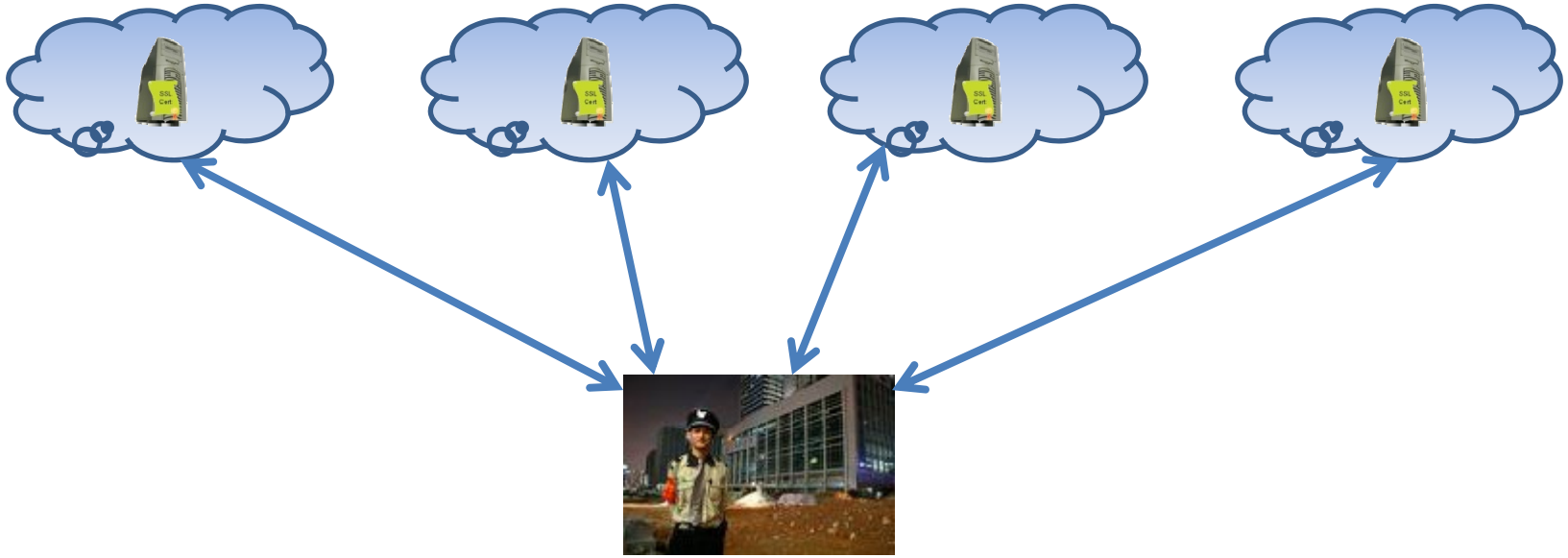
    New Paradigm: User's service is a mashup being served from multiple (often cross domain) web apps, provisioned by the digital-age end user
- Virtualization and clouds are themselves an example of this trend
  - Applications and data itself no longer reside in one permanent location but live in multiple locations at different times
- It's all about multi-party applications on multi-party platforms engineered on-the-fly by innovative twenty-somethings. Self-service on steroids.

Cloud Security Alliance Reference Model



- Provides an excellent reference model covering the major 'cloud' services being offered today.
- But the model does not explicitly identify the multiple parties involved. Lets take a (of course fictitious!) e.g.:
  - I mashup Google Maps with SalesForce.Com to get visual picture of where my customers are.
  - Lets say Google Maps runs on a Free BSD guest OS running on Vmware ESXi and SalesForce.Com runs on a Linux guest OS running on Amazon's EC2 cloud using XEN.
  - Assume Google uses SUN hardware using the SPARC architecture and SF.com uses INTEL PCs.
  - And Google leases data centers in multiple countries from Savvis and SF.com does the same from Cable and Wireless.

A day at the office



Mr. CSO

CIO: “The VP of Sales wants to use this cool new mashup between Google Maps and SalesForce.Com. It is going to increase our revenue, and the whole company will benefit. Surely you are OK with it?”

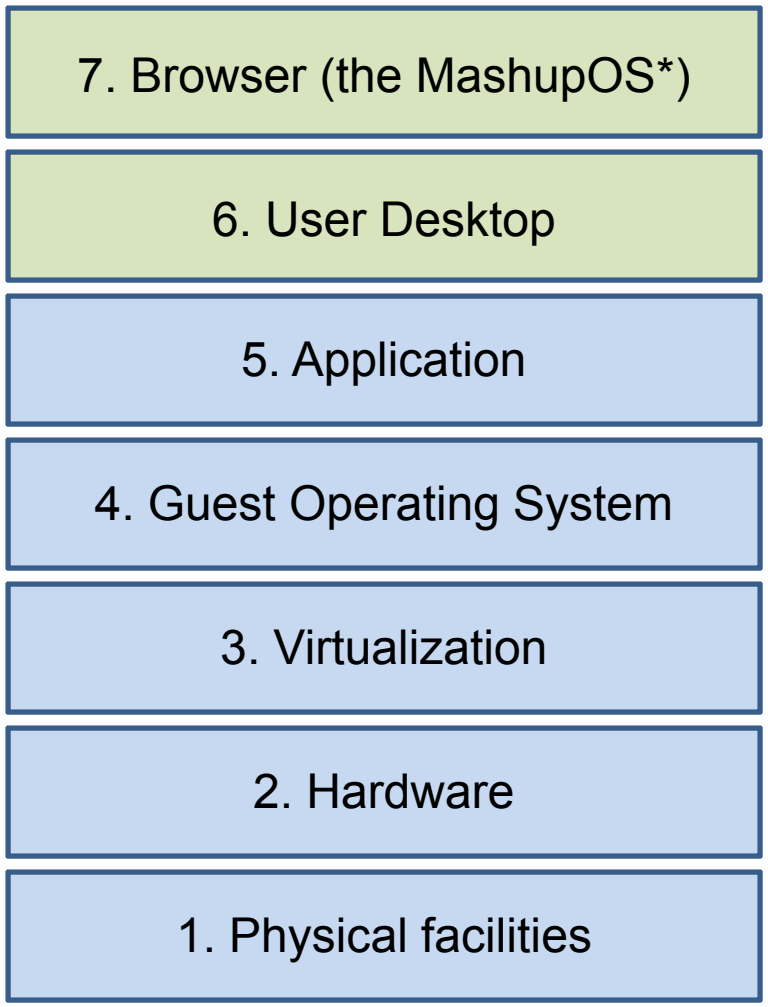
CSO contemplates: “What a fine conundrum!”

Soon Enough: New multi-party applications are popping up via spontaneous innovation by the twenty-something generation with no bandwidth for contemplation. Self-service on steroids.

4 step enforcement model/architecture

1. Identify the 'parties' using seven tangible layers.
   - Why seven? (homage to OSI stack; also considered a lucky number!).
2. Divide each layer into 'security units (SUs)' with security profile.
   - E.g. each cage in a shared physical data center is an SU or each guest OS on a hypervisor is an SU.
   - The Cloud Security Alliance document is an excellent starting point for what information each SU security profile (e.g. XML based) should have.
3. Add agents to each SU which contain all relevant security information (with appropriate certification).
4. When SU's need to interact, they can only do so if an automated Cloud Trust Broker (CTB) controlled by the CSO team permits it and enables it.
   - SU-Agent to CTB to SU-Agent connection is initiated.
   - If CTB allows connection, it brokers session keys and then steps aside.

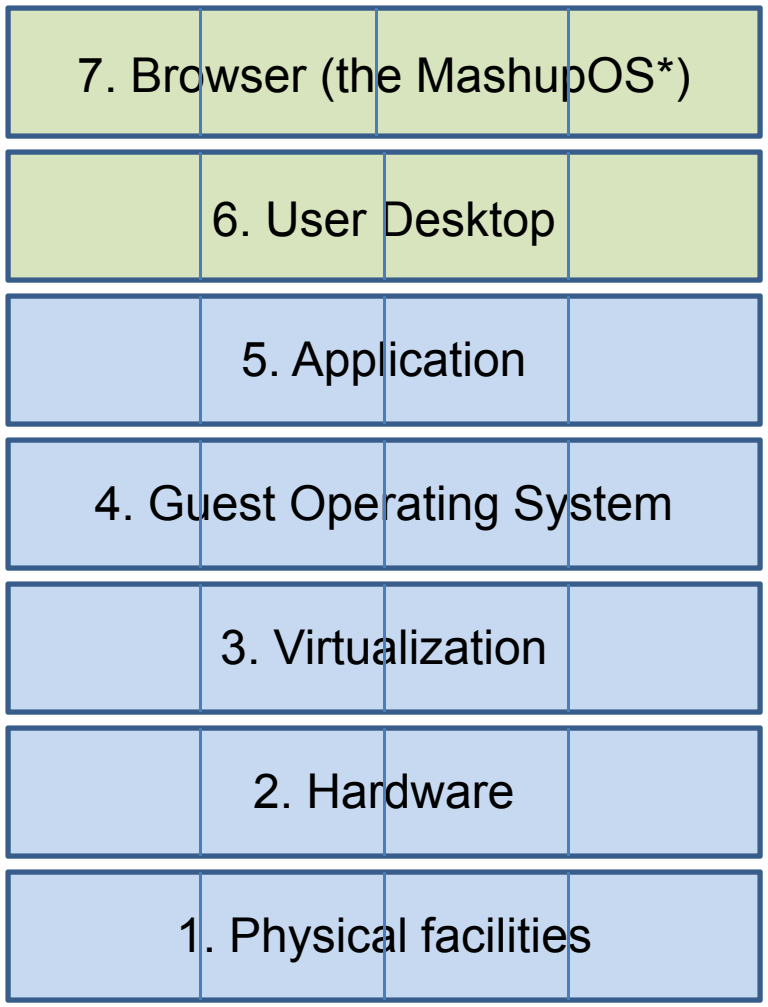
Step 1: Seven Layer Clouds



Specify the generic capabilities at each layer using a standards-based short XML profile.
e.g. Which browser? Is app a mashup from different sources?
e.g. Is the desktop imaged by an enterprise? Does it have a personal firewall/AV built in?
e.g. What sort of vulnerability analysis and scanning was done on the application?
e.g. Does guest OS have common criteria/NIAP certification?
e.g. Is privileged user access allowed remotely? Are dual controls required.
e.g. Do motherboards have TPM/other crypto hardware? Does the SAN use encryption?
e.g. What is policy for physical security? Does network layer 2 have encryption?

*Note: MashupOS is a term created by Microsoft Research (Ref. Howell, et al)

Step 2:Divide into security units (SUs)



Add/Modify generic layer security profile with SU specific profile.
e.g. Different widgets in a mashup might come from sources at different levels of trust.
e.g. Different user accounts could have different privileges.
e.g. In shared web hosting on common OS the application security profiles will differ.
e.g. Obviously could be different OSes, but even same OS could widely differ in security.
e.g. Different administrators could have different privileges wrt different hypervisors.
e.g. Each blade or server or SAN could be separate SU.
e.g. Each cage in shared facility could be a separate SU.

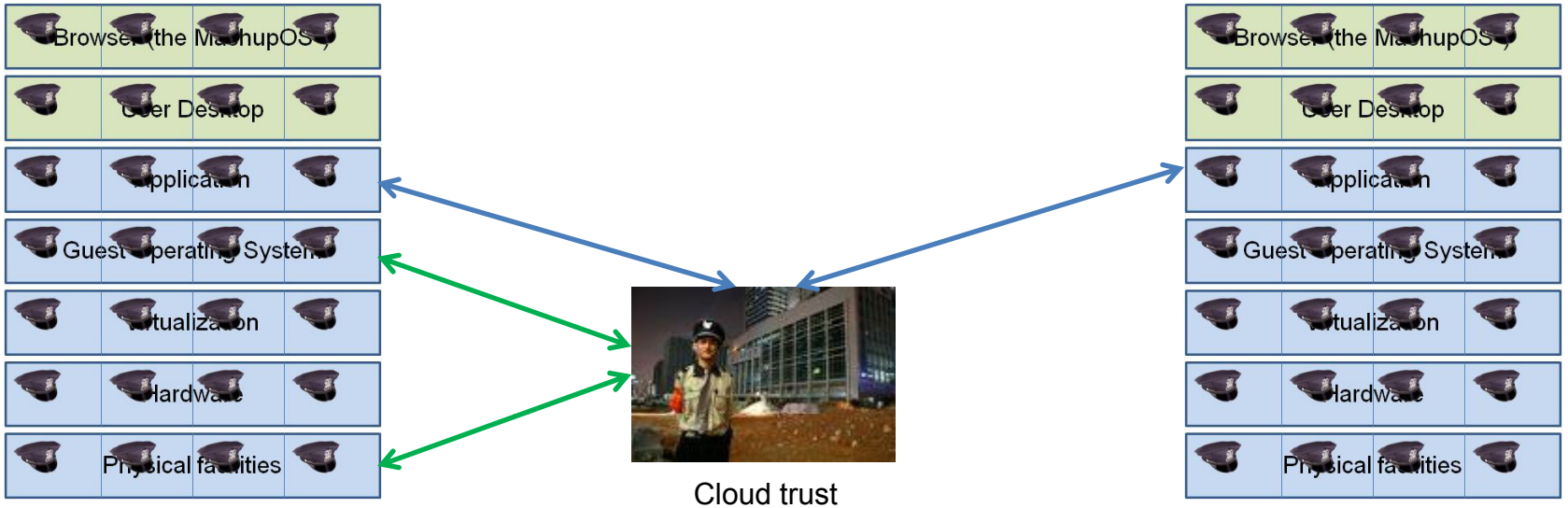
Assumptions (especially about Layers 1 -5)

- While not always true:
  - we in general assume that layer N-1 ***can*** generally compromise layer N.
  - we in general assume that N+1 ***should not be able*** to compromise layer N.
  - we assume that the 'Chinese Walls' between SUs are ***not*** porous.
- The reliability of all of the three above assumptions should be certified regularly and become a part of the SU profile.
- The SU profiles are cumulative upwards, in other words, the Layer 5 SU profile has the profiles for all the layers below.
  - Note: One does not have to wait for lower layers to implement functionality. E.g. a Guest OS can create a profile for the data center, if the data center does not as yet provide one. Obviously in the long run it would be preferable if each layer maintains its own up to date profile.

Step 3: Embed "security agents" into each SU

- The agents are used to establish connections between SUs:
  - Either to share SU security profile
  - And, potentially as precursor to moving data/apps from one SU to another.
- They have to have access to unique credentials (private key, cert) for mutual auth and session key exchange.
  - How keys are protected should be part of security profile.
  - Could potentially use TPMs at Layer 2 for key protection, and allow channel from Layer 5 to 2 limited for this purpose.
- Some layers, especially Layer 1 may not have ability to embed agents. Agent will have to reside on separate locked down computer.

Step 4: The Cloud Trust Broker



- SU-Agents are never allowed to communicate directly with each other.
- They have to:
  - Establish trust through a cloud trust broker (CTB) using a multi-party trust protocol.
- The CTB will validate the 'security profile' and use policy to decide whether to permit the establishment of the connection.
  - It can also compare the security profile presented to it, with its prior knowledge of what the security profile of the SU was.
- In most cases after setting up the connection, the CTB gets out of the way.

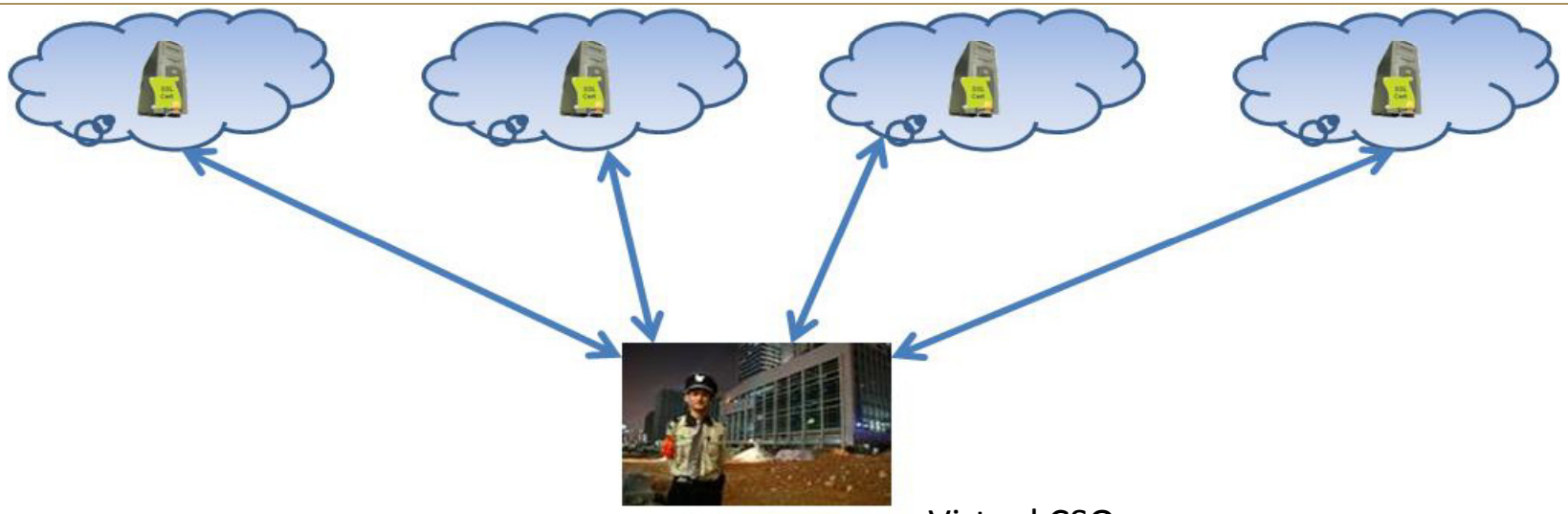
Two new ingredients are needed

- A multi party trust protocol that can run at any layer.
  - Preferably a single standard and demonstrably secure protocol.
- A Cloud Trust Broker
  - Could be hosted by enterprise
  - Could be a SaaS offering
  - Could be a public trust broker.
  - Eventually Cloud Trust Brokers may need to talk to each other.

SafeMashups Cloud Trust Broker

- SafeMashups is a company that was incubated at the Institute for CyberSecurity at the University of Texas at San Antonio and is currently being spun out. (see www.safemashups.com)
- It invented the concept of MashSSL, which is a way to make SSL a multi party protocol that can be run at any layer, including over (not under) HTTP.
- MashSSL is in the process of being made an open standard as part of an Alliance that includes almost every major Certificate Authority, and a variety of other security companies and universities, etc.
  - It allows for establishment of multi party trust through 3rd parties such as a cloud trust broker (or for two web apps communicating through an untrusted user).

SafeMashups Cloud Trust Broker



Cloud Trust Broker Provides

- Brokers session (using MashSSL) from one service and establishes temporary shared secret (the familiar SSL master secret).
- Will only allow connection establishment if the policy permits the services to interact.
- Once it establishes session, it gets out of the way allowing services to communicate directly.

Benefits of Cloud Trust Broker

- Enterprise retains control of flow of data/processes. E.g. only allow data and processes to migrate from less secure to more secure environments.
- Perfect point to enforce governance, regulations and compliance.
- Secure audit trail.

Summary

- Security and the Cloud
  - Cloud Technology intrinsically changes existing security problems
  - Cloud Technology enables new applications which bring new security challenges
    - It's all about multi-party applications on multi-party platforms engineered on-the-fly by innovative twenty-somethings. Self-service on steroids.
- New fundamental problems arise including
  - How to broker trust across multi-party applications running on multi-party platforms
- At a technical level this poses several challenges for which we propose the following solutions
  - A seven layer security enforcement architecture comprising Security Units at each layer with Security Agents that can communicate as brokered and enabled/prohibited by Cloud Trust Brokers
  - A demonstrably secure standard three party protocol to achieve this goal
- SafeMashups, a spin-out from UTSA's Institute for Cyber Security has
  - Invented MashSSL the necessary three party protocol which is in process of becoming an open standard
  - Implemented its first generation of Cloud Trust Brokers and is developing the next generation