FHV

Trusting your Cloud Provider (’s System)

Protecting Private Virtual Machines Hosted by a Cloud Provider Using Mandatory Access Control, Trusted Boot and Attestation

Armin Simma

deepsec 2014, Vienna

About me (Armin Simma)

- passion for "Sicherheit" (german for safety and security)

- security comes with my job
- teaching and researching IT security at Vorarlberg University of Applied Sciences, Austria (FHV)

About me (Armin Simma)

- passion for “Sicherheit”

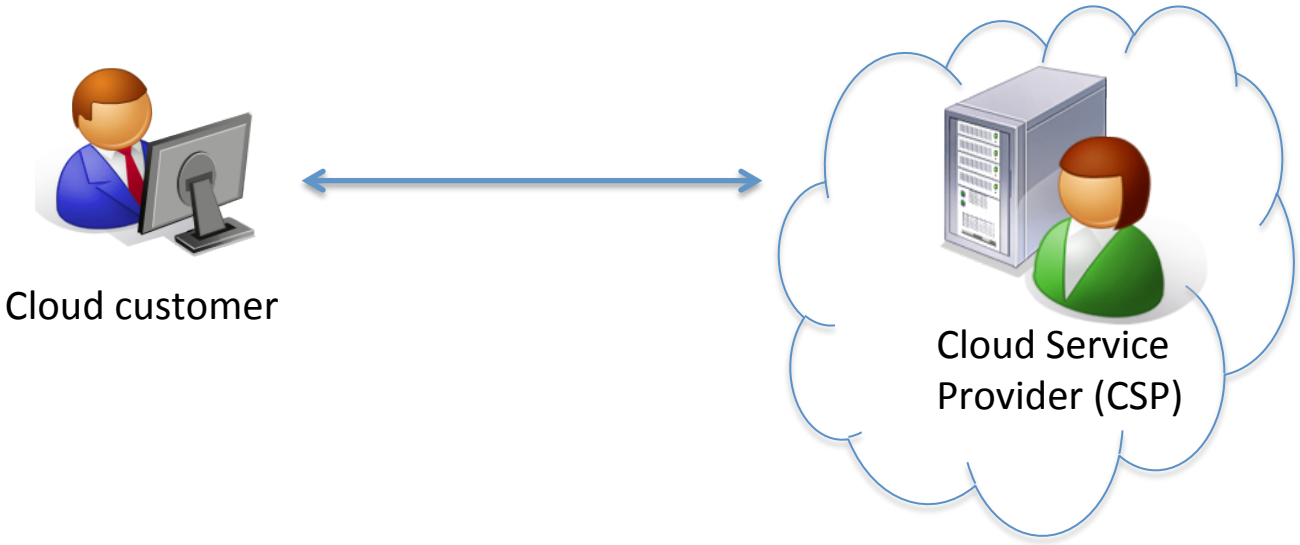
- safety (and trust) comes with my hobby
- mountain climbing…

Outline/ Agenda

- Motivation
- Problem (statement): insider attacks within the cloud
- Solution:
  - Mandatory Access Control (MAC)
  - Trusted Computing / Trusted Platform Module (TPM)
    - Integrity Measurement
    - Attestation
- Architecture of the system
- Ecosystem
- Related work
- Conclusion

Motivation: Cloud Security and Privacy

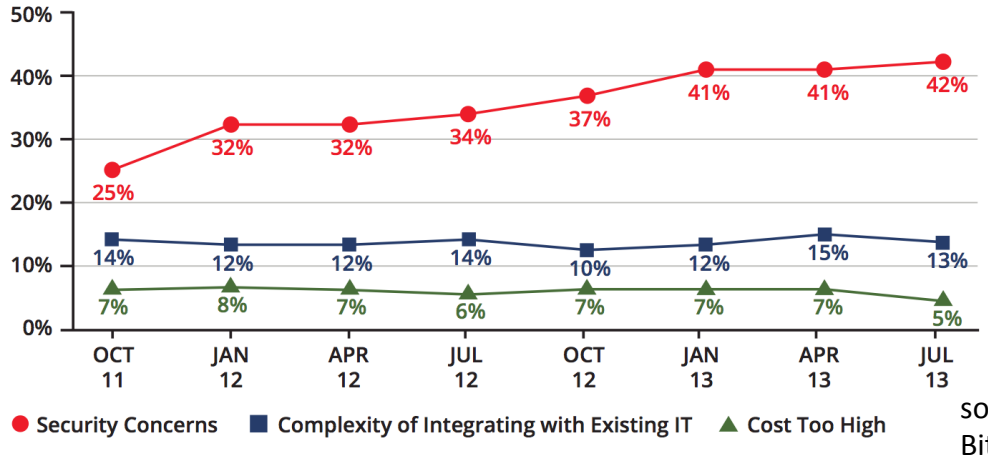
- Two main entities in a cloud scenario:
- Both have valid security goals
- We mainly focus on the **customer's security goals**



Motivation: Cloud Security and Privacy

- **Security is still top inhibitor** for cloud adoption
  - Alert Logic: “cloud security remains a major concern“

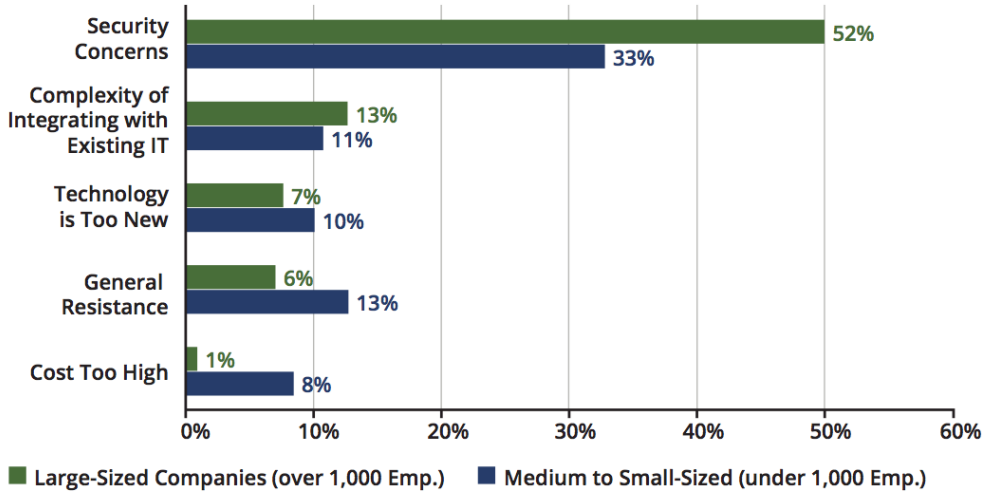
    source: B. Rahul, Maureen, and S. Scott, “Alert Logic Cloud Security Report,” Alert Logic, Houston, TX, 2014.



source: “Cloud Adoption Report,” Bitglass.com, Campbell, CA, Apr. 2014.

Motivation: Cloud Security and Privacy

- **Security is still top inhibitor** for cloud adoption



"Future of Cloud Computing Survey," northbridge.com, 2014.

Motivation: Cloud Security and Privacy

- **Security is still top inhibitor** for cloud adoption

Cloud Adoption Hindered by Security Concerns, Survey Reveals

A survey from Internap Network Services Corporation gives insight into the cloud business

By Gabriela Vatu on January 15th, 2014 15:43 GMT

Cloud services continue to face perception problems at a global level, as most companies have security concerns when it comes to these types of products.

According to a global survey from the Internap Network Services Corporation,

source: http://news.softpedia.com

Mark Russinovich: RSA 2014

Motivation: Cloud Security and Privacy

- Cloud Security Alliance lists malicious **insider attacks** in their "Notorious Nine: Cloud Computing Top Threats in 2013" as number 6
- https://cloudsecurityalliance.org/download/the-notorious-nine-cloud-computing-top-threats-in-2013/

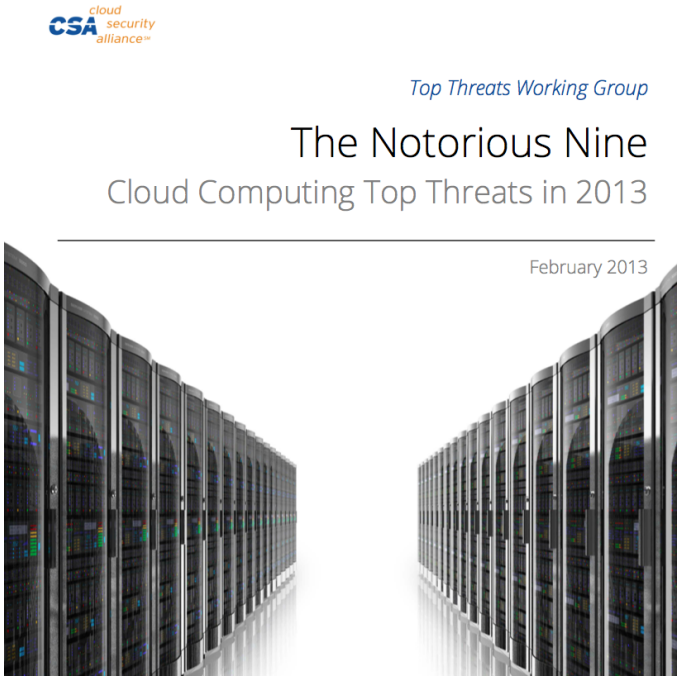
Real-World examples of insider/ rogue administrator attacks

- CERT.org (2012)
- administrator had access to file with password info

http://www.cert.org/blogs/insider-threat/post.cfm?EntryID=100

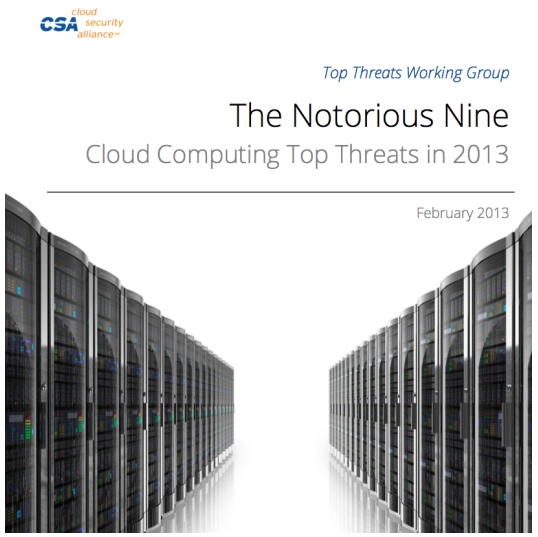
- Salesforce.com large-scale leak of confidential user data 2007
- insider (employee of salesforce) revealed customer list to phisher

D. Thilakanathan, S. Chen, S. Nepal and RA Calvo "Secure Data Sharing in the Cloud"



Motivation: Cloud Security and Privacy

- (malicious) **insider attacks**
- see also keynote today morning from Alexander Hutton
- classification of attacks: insider attacks are 18%

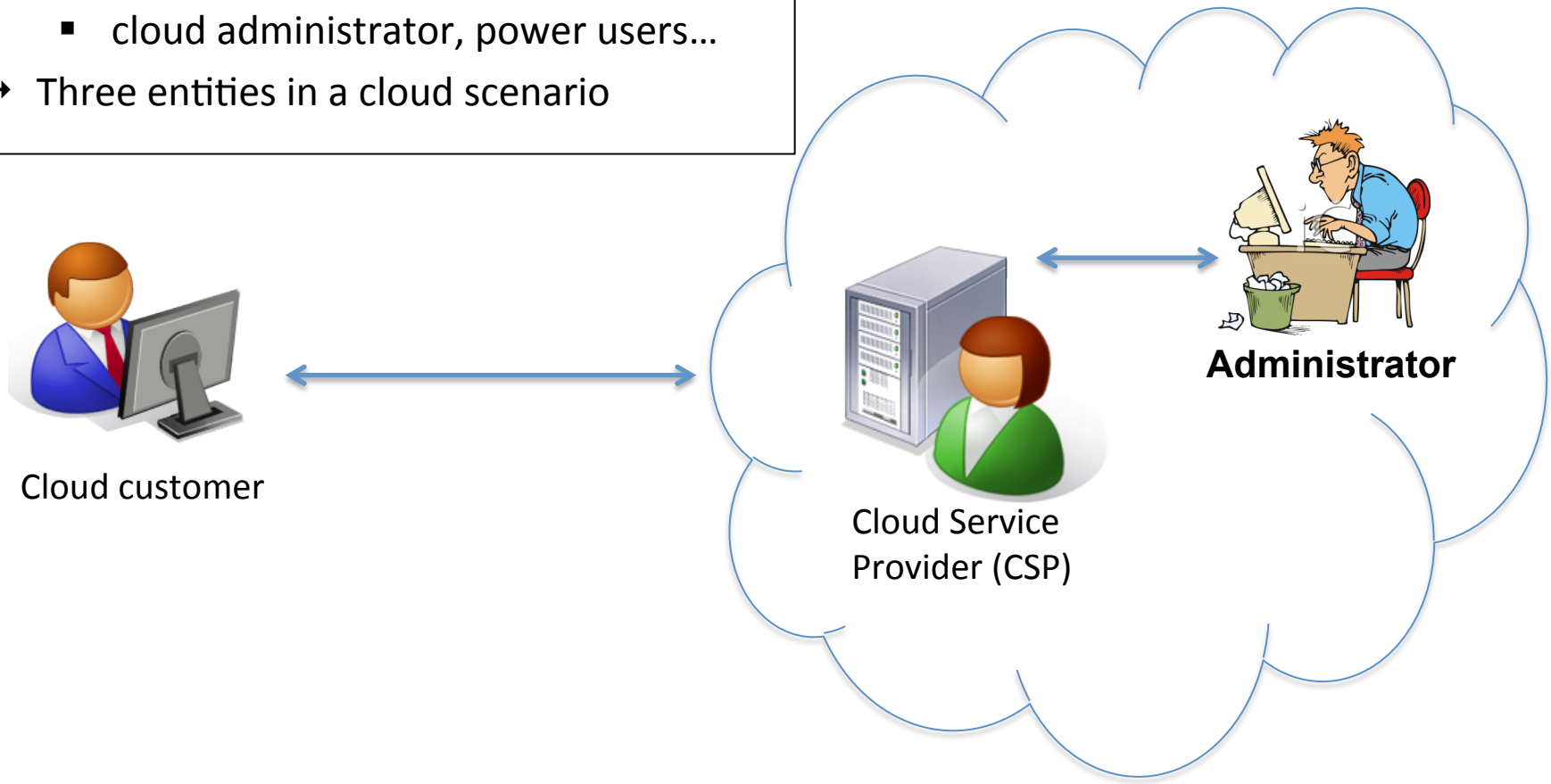


Motivation: Cloud Security and Privacy

insider of the cloud

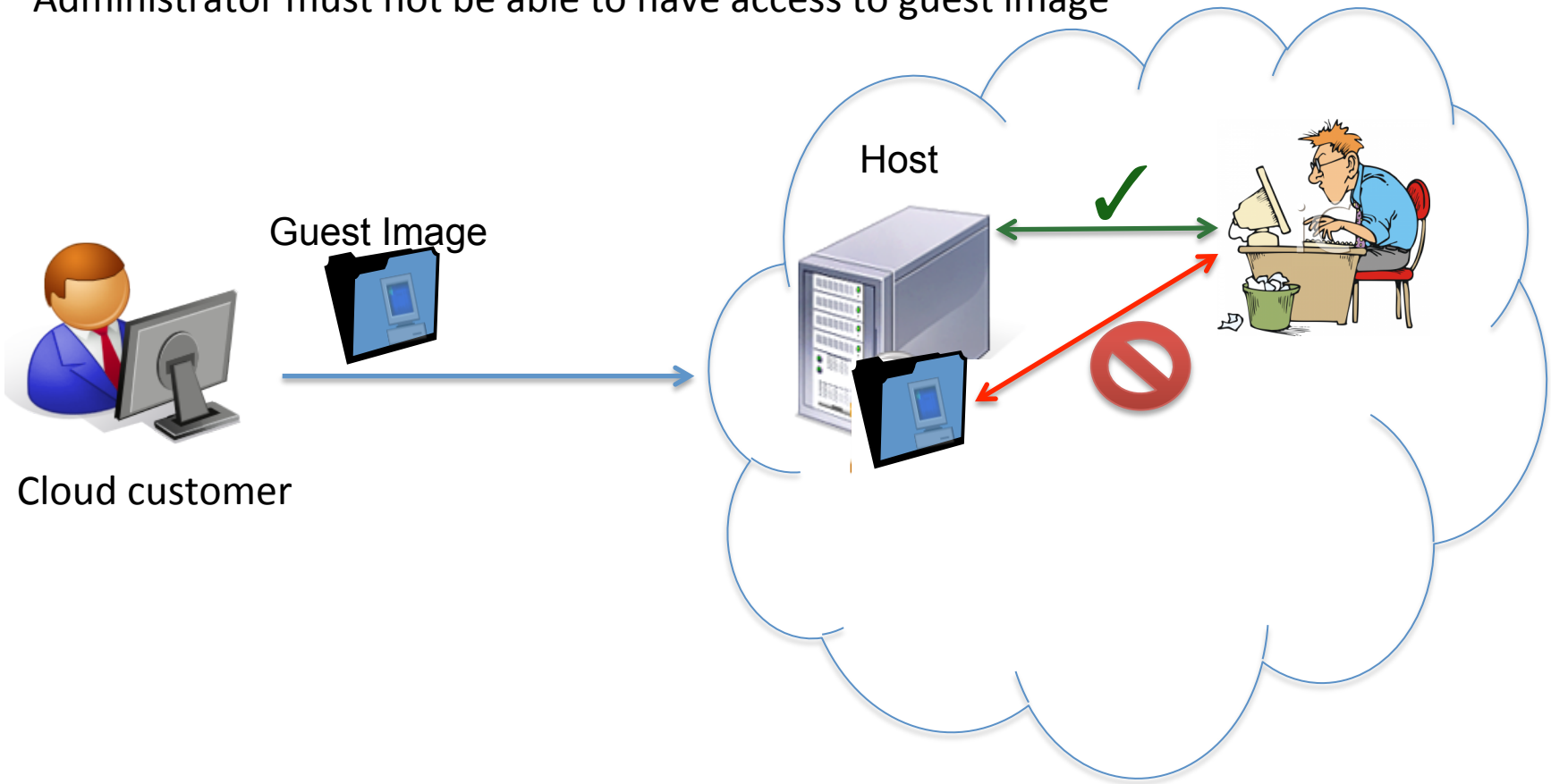
- cloud administrator, power users…

→ Three entities in a cloud scenario



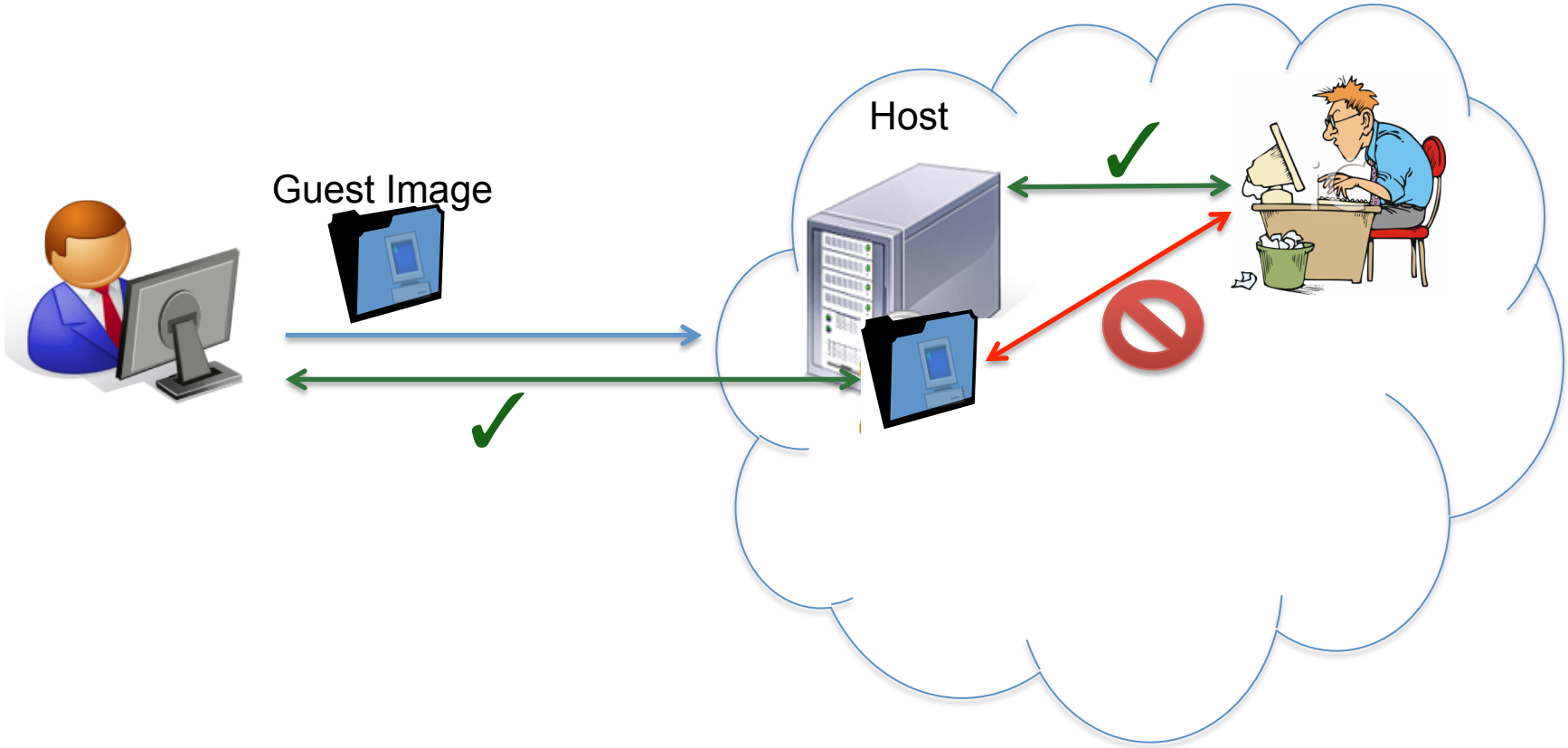
Motivation: Cloud Security and Privacy. Confining the insider

- Administrator must be able to administer/control the host but
- Administrator must not be able to have access to guest image



Motivation: Cloud Security and Privacy. Confining the insider

- Customer (and nobody else) must have access to his guest image
- *"Retaining Control over Private Virtual Machines Hosted by a Cloud Provider "*



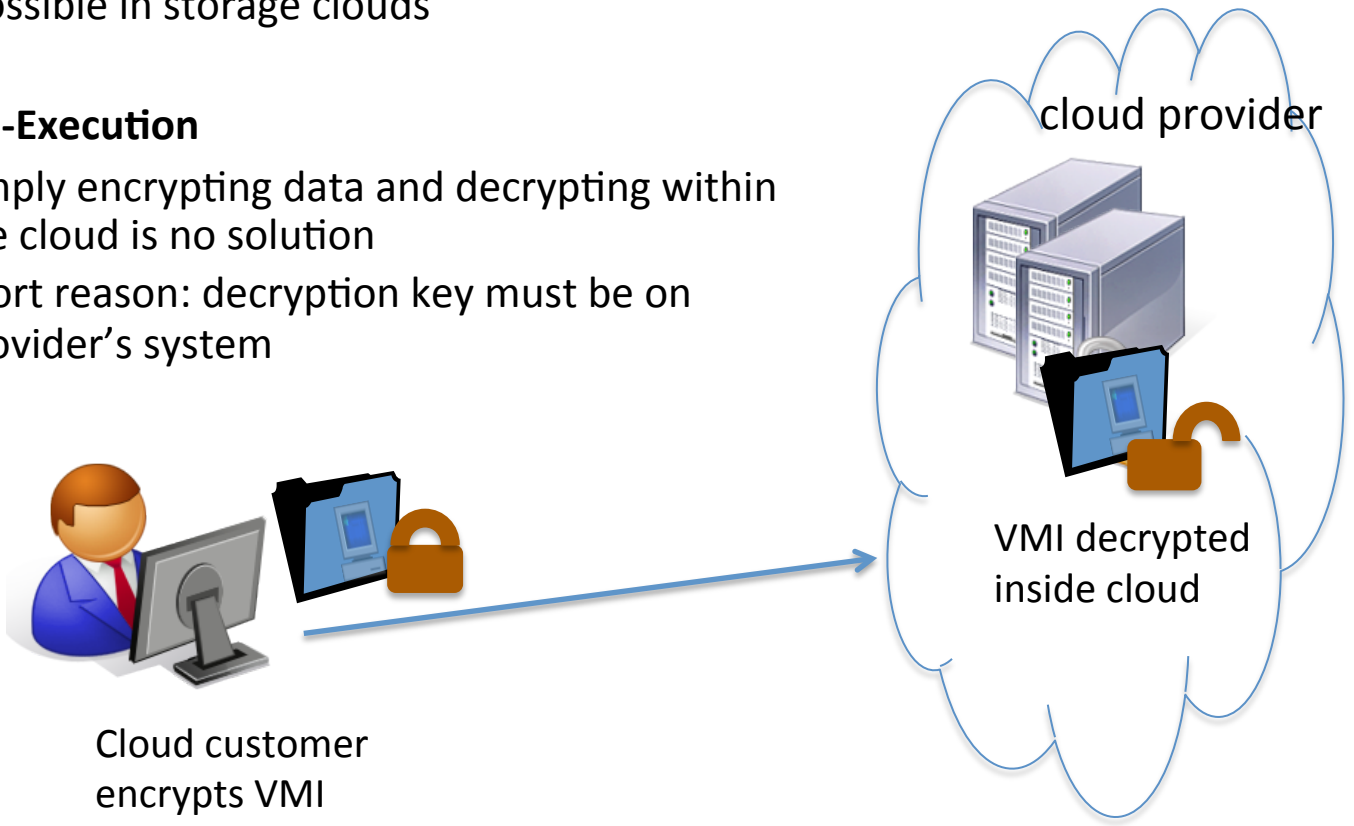
Problem Statement

- How to assure, that (possibly malicious) insider at the provider's side has no access to customers VM?

- Assumptions:
  - insider has no physical access to host machine
  - cloud provider is "partly" trusted
    - provider is not malicious but
    - provider could be careless (about security controls)

Simple Solution: Crypto

- **encryption of VMI (virtual machine image)?**
- **Problem:**
  - Only possible in storage clouds
  - **Data-in-Execution**
    - simply encrypting data and decrypting within the cloud is no solution
    - short reason: decryption key must be on provider's system



Security based on Trust

- Trust: "**assured** reliance on the character, ability, strength, or truth of someone or something "

  - merriam-webster dictionary

- assurance needs kind of a "proof"
  - compare common criteria EAL7
  - assurance through e.g. formal verification/ proof

- How can a cloud customer get proof?

Research (and practical) question:

- How to assure, that (possibly malicious) insider at the provider's side has no access to customers VM?
- How can cloud customers trust in the cloud infrastructure..
  - …that provider protects their sensitive information (=the VMI)?
  - Customers would like a "**proof"** …
    - …that the cloud infrastructure is "secure" = trustworthy
    - …that provider has taken measures/ controls against (insider) threats

Assurance/ “proof”: Service Level Agreements

- Customers would like a “proof”/ guarantees …
  - …that provider has taken measures/ controls against (insider) threats

- in the past and currently: **Service Level Agreements** (SLA) between provider and customer
- most cloud SLAs do not fully cover security

  [Per Hakon Meland et al., Expressing cloud security requirements for SLAs in deontic contract languages for cloud brokers]

- If any “proof” exists at all, it is an external audit/ certificates

Solution in a nutshell

1. **Mandatory Access Control (MAC)** for confining (root) user
2. **Trusted Computing** technology for preventing low level attacks e.g. boot attacks and for the trustworthiness proof
   - integrity measurement
   - attestation

MAC Mandatory Access Control

- Discretionary Access Control (DAC) gives owner of resource full access to this resource
  - typically on file level (rwx)
  - no least privilege (see e.g. passwd process and setudid)
- Mandatory Access Control (MAC) enforces system-wide policies
- MAC allows more fine-grained policies
  - on system level
- MAC is considered more secure than DAC

MAC Mandatory Access Control: SELinux

- SELinux is an implementation of MAC for linux
- Past projects have shown the strength of Selinux

- Dan Walsh gives many examples where SELinux would have prevented known vulnerbilities

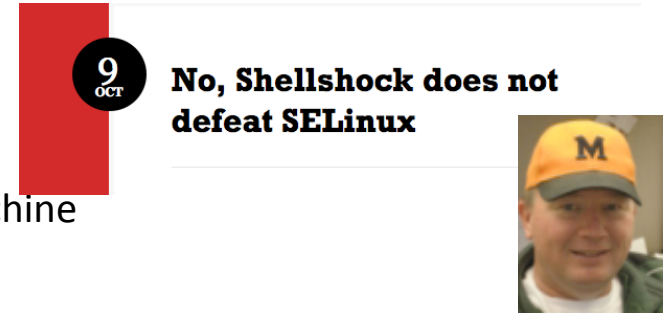
  examples:

  - shellshock

  http://cybermatters.info/2014/10/09/shellshock-selinux/

  - CVE-2011-1751 Break out of virtual machine

  http://danwalsh.livejournal.com/45194.html



- examples of (old but) real systems that restrict root user
  - Russell Coker's Play SELinux Machine
    - http://www.coker.com.au/selinux/play.html
  - Hardened Gentoo SELinux
    - http://www.linuxsecurity.com/content/view/114169/169/

MAC Mandatory Access Control: SELinux

- We use SELinux in our prototype to restrict root
  - from copying virtual images and snapshots
  - from doing other (malicious) things

```
# ls –Z
# system_u:object_r:virt_image_t:s0  fedora20.qcow2
# chcat -- +s2:c1 fedora20.qcow2
# ls –Z
# system_u:object_r:virt_image_t:s0:s2:c1 fedora20.qcow2
```

- We use SELinux to create tamper-resistant log files
  - even for root
  - restrict root from /var/log/audit.log

- SELinux policy can become **complex**
  - our (simple v1) prototype has ~40 complex rules

MAC Mandatory Access Control

- Conceptual problem
  - provider wants (and needs) a root user for several administrative tasks:
    - e.g. for updates

- Proposal
- Principle of **least privilege**

  → two root users:
    - default root
      - strong authentication required
        - only physical login (no remote login)
        - 4-eyes principle/ dual password
        - multi-factor authentication (HW token)
    - confined “root” (power user)
      - for day-to-day operations

  → tamper-proof log files
    - for all users (including root)

Privileged Identity Management

- The usual recommendations for controls/ processes
  - see e.g. CSA Cloud Controls Matrix
  - or ISO 27002, NIST 800-53, HIPPA…

  - least privilege
  - separation of duty
  - removing unused accounts
  - remove account immediately when employee leaves company
  - no “generic” accounts e.g. “Administrator”
    - instead use personalized account
    - for liability, accountability
  - logging and monitoring
    - **especially for privilege account**
    - when accessing sensitive data
    - **protect log files**
  - …

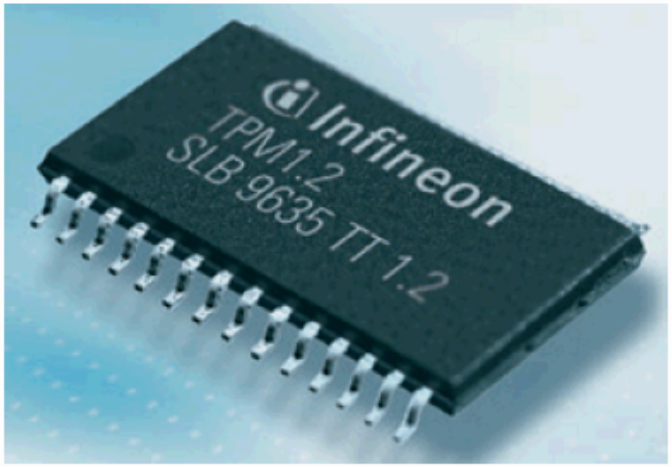
What is Trusted Platform Module (TPM)?

In a nutshell

- Security chip (microcontroller)
- keys are protected against external attacks and theft
- tamper-resistant

- Cost: mass production: ~ 1-2$
- currently sold 600 million+ TPMs

- Spec created by
  Trusted Computing Group TCG



TPM is Root of Trust

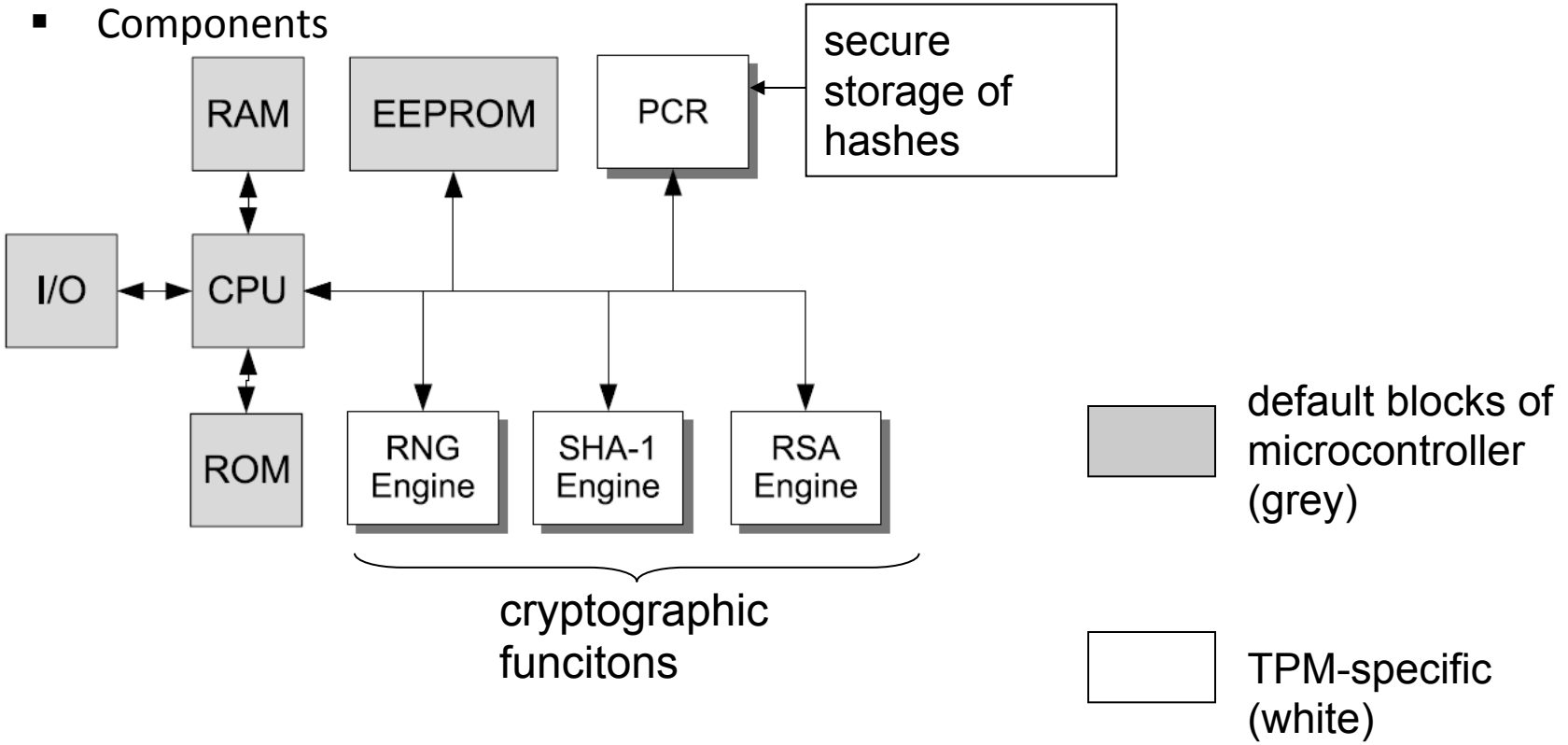
- **Root of Trust** (RoT) gives the initial anchor for a security system

- TPM is Root of Trust for
  1. integrity and measurement
  2. attestation
  3. storage

What is TPM?

The Details (v1.2)

- Components



- Authentication/ Identification of the machine (= the TPM plus platform)
  - Attestation Identitiy Key (AIK)
- Storage for hash values
  - stored hash value can not be overwritten
  - PCR
- Integrity Measurement
  - using hashes of the system
- Attestation:
  - vouching for the integrity of the configuration/status of the system

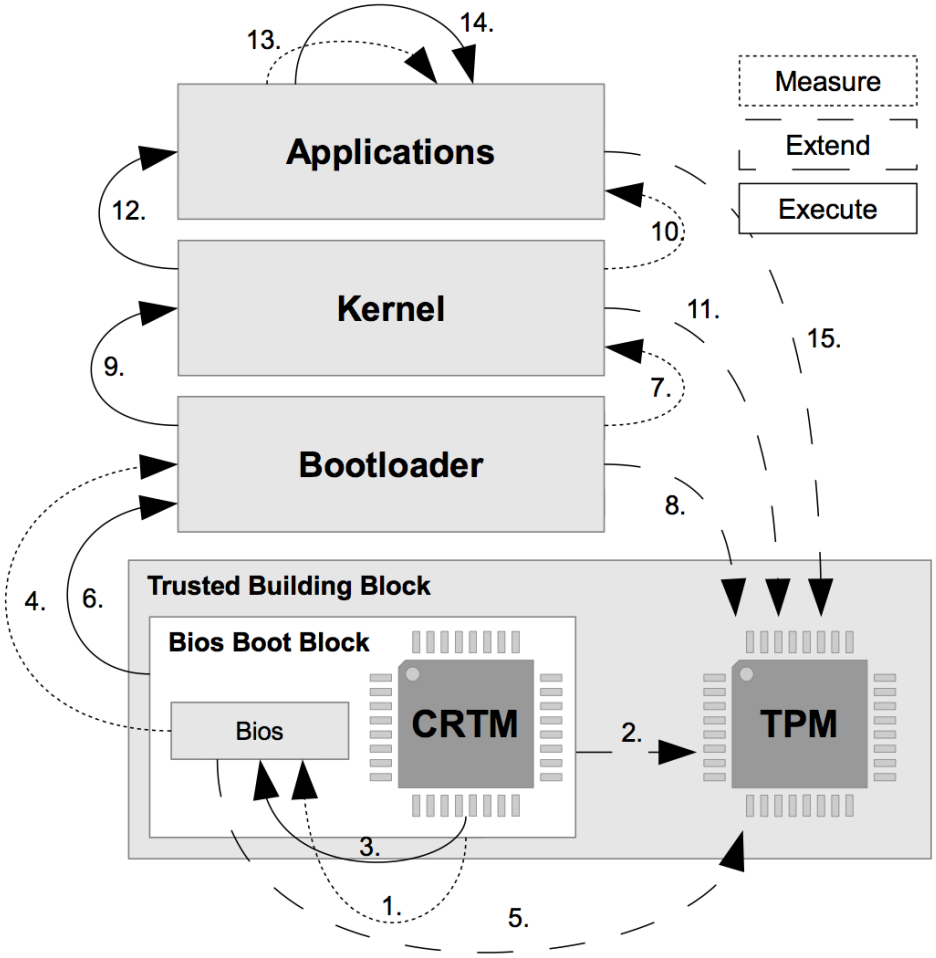
TCG Measured Boot: Details

- Before any component is executed, it is measured i.e. a hash is logged into the PCR
- For each component C_i do these steps:

1. **Measure** C_{i+1} : compute hash of code of next component C_{i+1}
2. **Extend** this hash into TPM's PCR using TPM command *PCR_Extend*
3. **Execute** C_{i+1}

- Somewhere it has to start: Core Root of Trust for Measurement CRTM

PCR_Extend operation ≈ store hash value to PCR of TPM and preserve old value



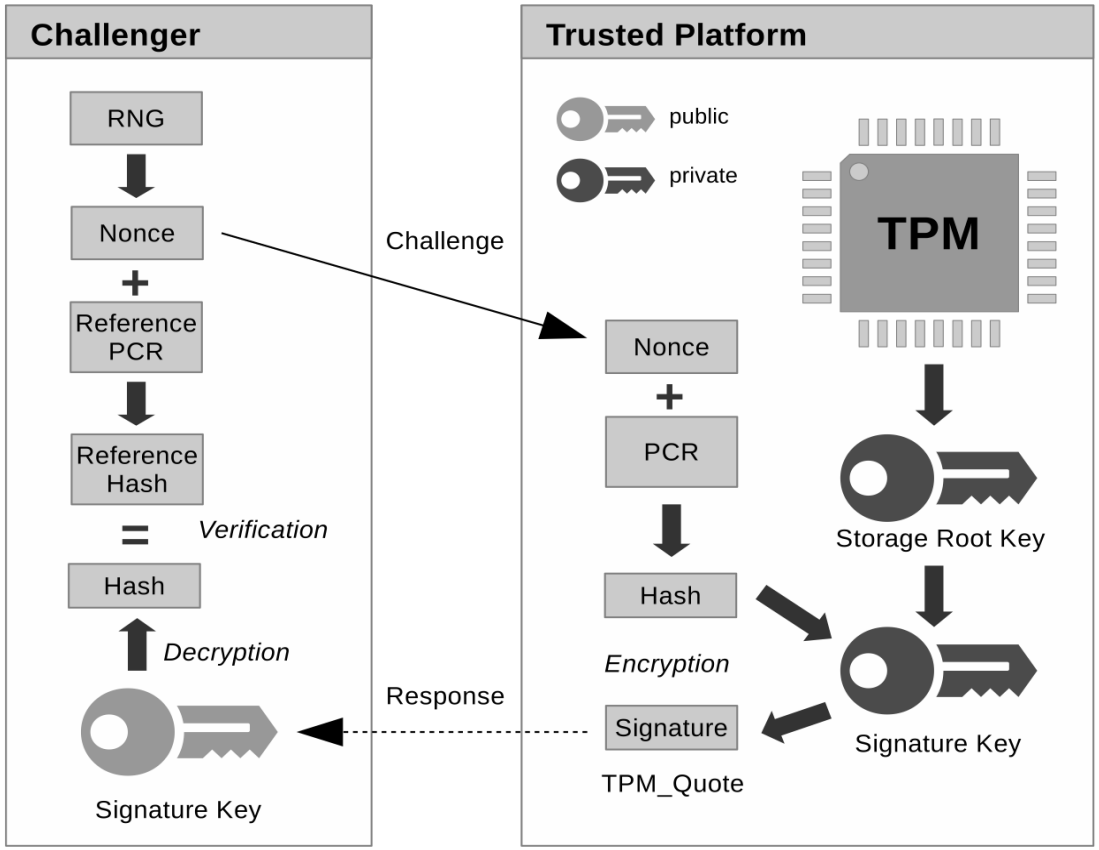
Reporting and Attestation

- After measurement and storage of hashes into PCR…
  …usually a (remote) entity wants to obtain these values
- This process is called **reporting**

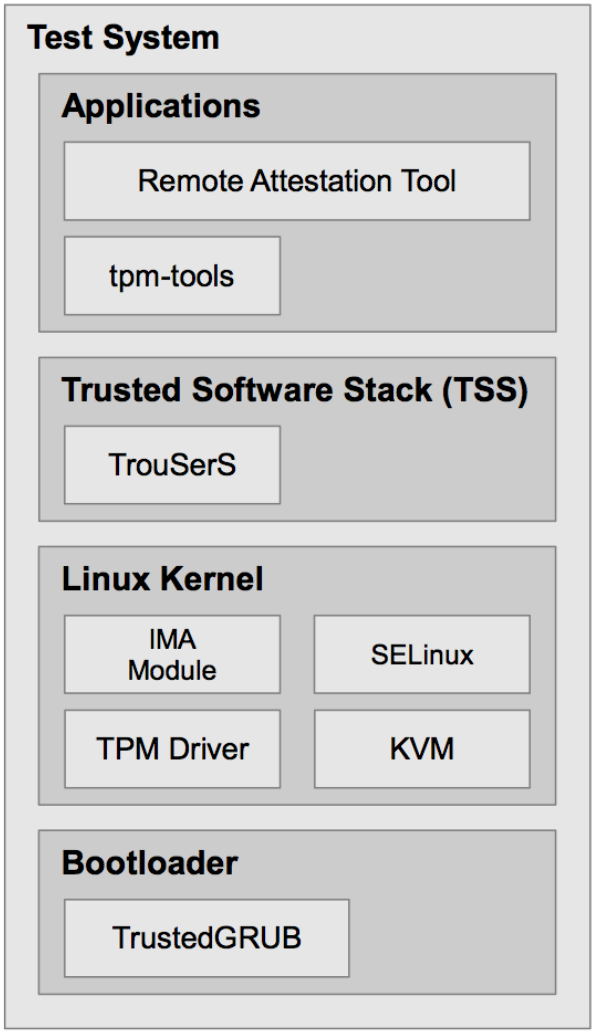
- The remote party uses the reporting for **attestation** of the platform , i.e. as a proof that a specific system (configuration) was loaded/booted
- The remote party needs **reference values** to compare the transferred PCR values with
- These reference values must have been created in advance by using an identical reference system
  - e.g. by a trusted third party/ tester = TTP/TTT
  - values are often called “golden” values or “whitelist”

Reporting and Attestation

- Nonce provides for freshness of response
- Digital Signature (done inside TPM): assurance of integrity of PCR values



Architecture of core technical system

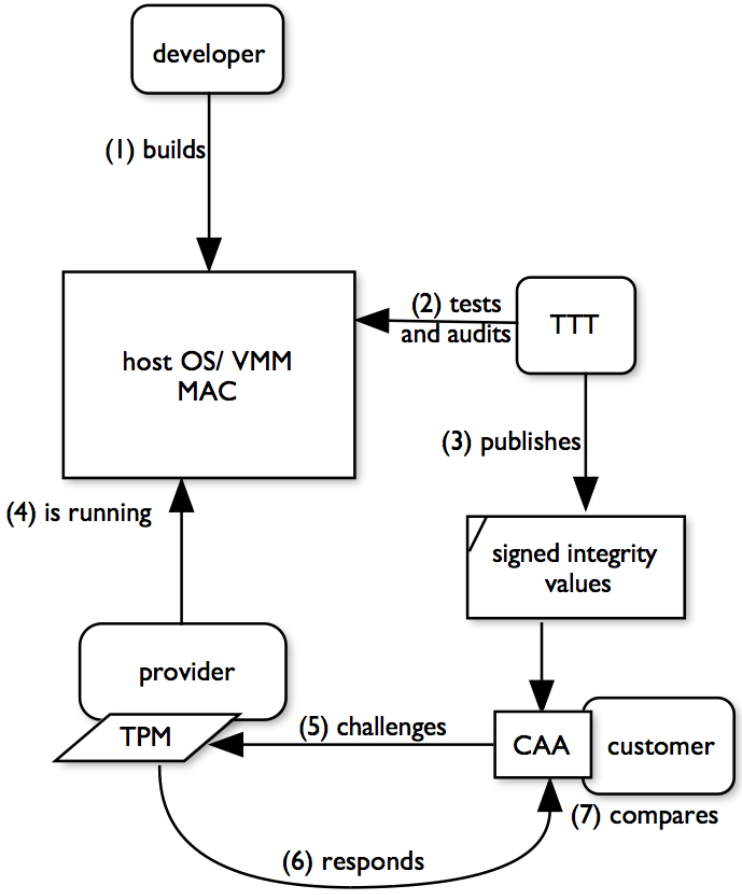


This was our first prototype.
We switched to **tboot** and **OpenAttestation** library (instead of TrustedGRUB and RemotAttestationTool, resp.) for the current version

The overall ecosystem

„Ecosystem“ is necessary for trust

- **Trusted Third Tester (TTT)**: external entity that should thoroughly test the reference system
  - publishes signed integrity values
- **Cloud Attestor and Advisor (CAA)**
  - software running on customers site
  - does the remote attestation



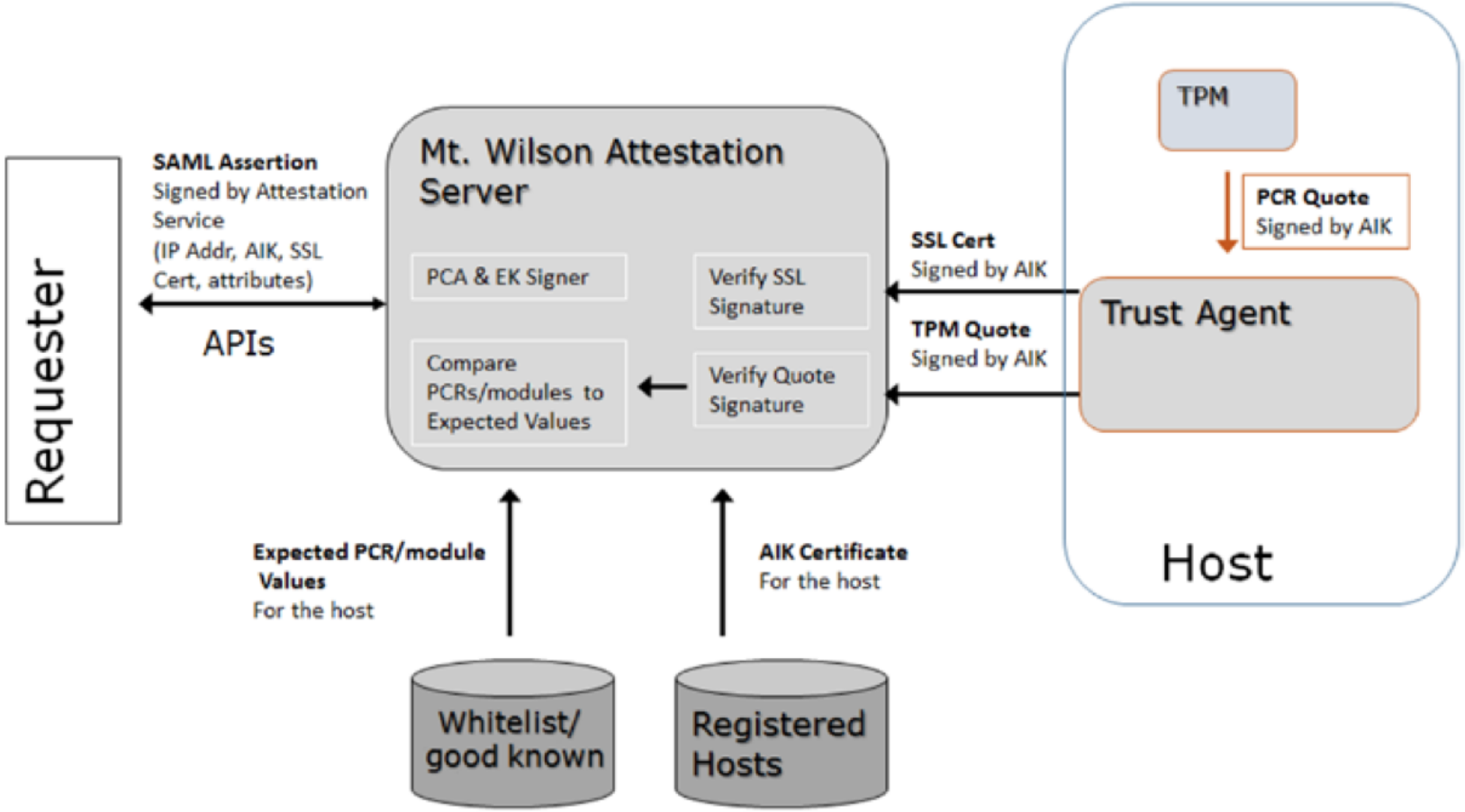
Related work/ technologies

- Intel:
  - Mt.Wilson
  - Mystery Hill
- Haven and SGX

- and the many scientific results (see deepsec proceedings)

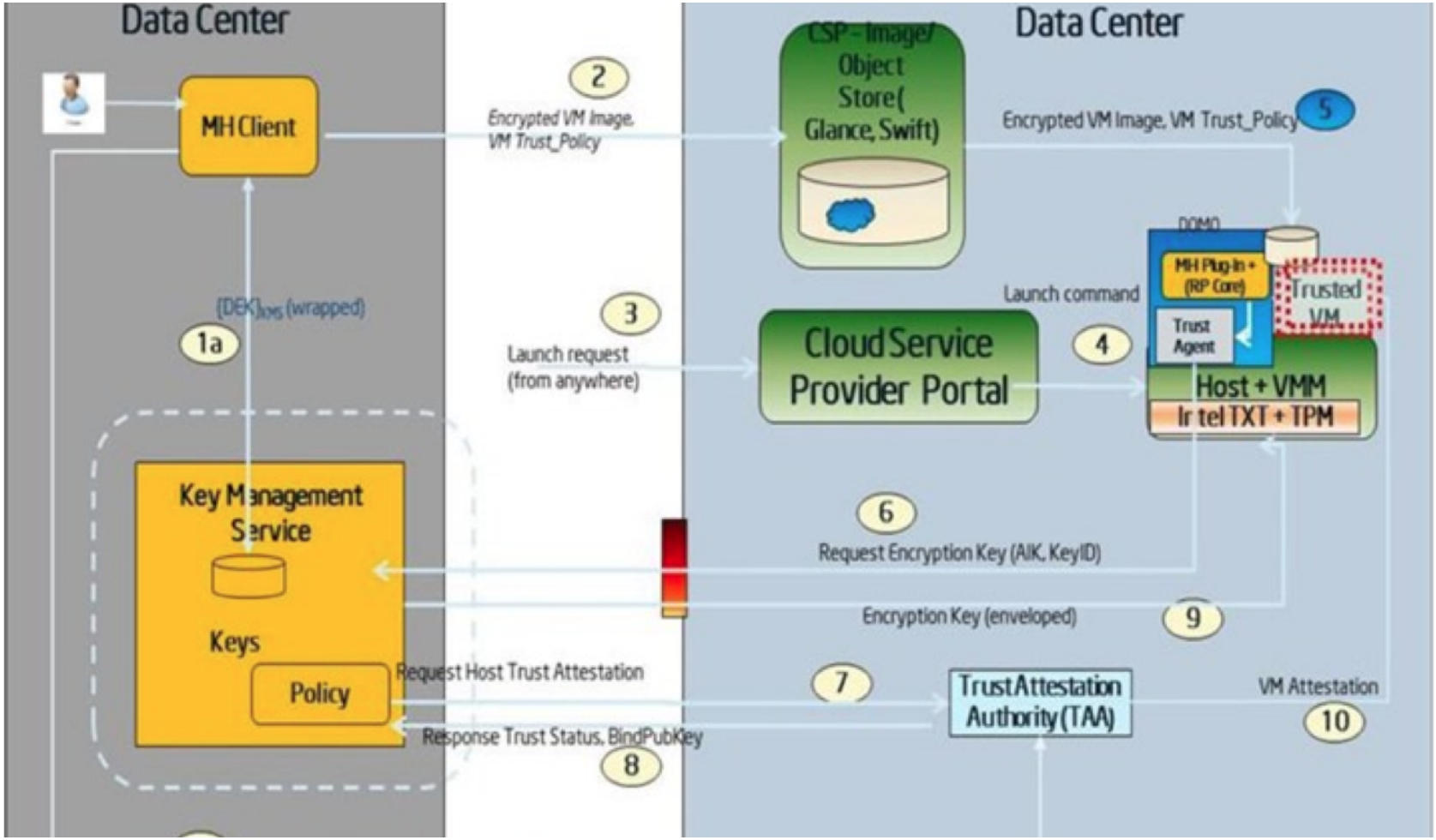
- Homomorphic encryption
  - the “holy grail” of cryptology

Related Work: Intel Mystery Hill and Mt.Wilson



Source: Raghu Yeluri: "Building the Infrastructure for Cloud Security:"

Related Work: Intel Mystery Hill and Mt.Wilson

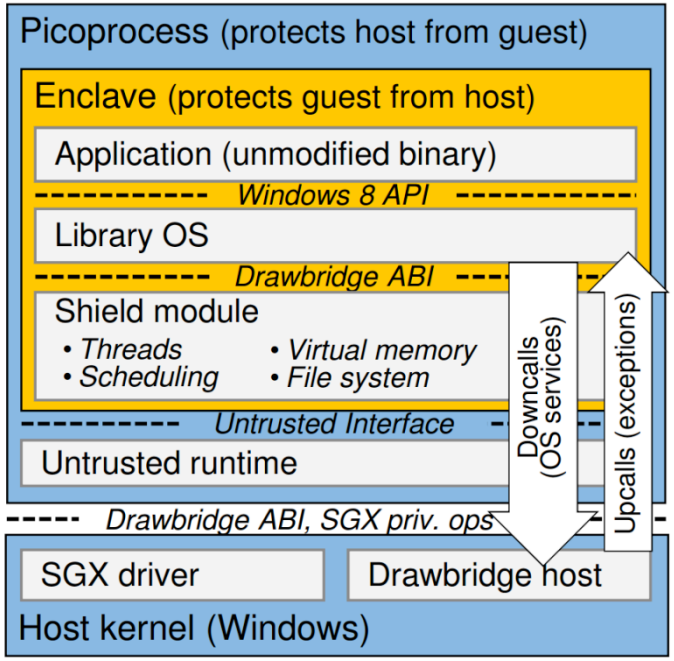


Source: Raghu Yeluri: “Building the Infrastructure for Cloud Security:”

Haven and SGX

- is an “inverted” sandbox
- code and data are protected from privileged code (e.g. hypervisor)

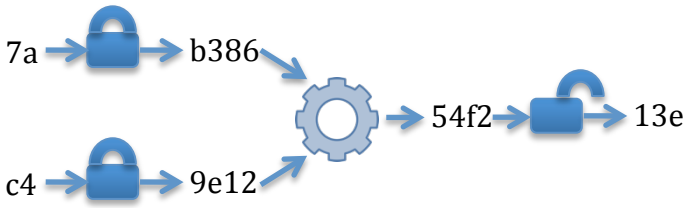
- Based on
  - Intel’s Software Guard Extensions (SGX)
  - Drawbridge



source: A.Baumann et al., Shielding Applications from an Untrusted Cloud with Haven

Homomorphic encryption

- Main idea:
  - perform operations on ciphertext
  - (encrypted) result must be decrypted
  - attacker without key does not learn anything, even if he knows ciphertext plus cipher-result



- Great Idea with many applications:
  - this talk:
    - sending encrypted code (VM) to an untrusted provider
    - provider can execute encrypted code
    - result (encrypted) sent back to cloud customer
  - big data: performing analytics on encrypted data

Issues with TPM-based solution

1. inefficiency
   - Measuring using TC-tech is slow
   - if TCB (trusted computing base) is big (in size), it is impractical

   - Solution:
     - minimize TCB: small hypervisor
     - D-RTM (Dynamic root of trust)

2. Whitelist ("golden measurements") management is complex
   - systems come in MANY variants: versions of OS, configuration, MAC policy…
   - reference system must be 100% the same

   - Solution:
     - modularize system
     - individual PCR/ reference values
     - D-RTM

Issues with TPM-based solution

3. Attacks on TPM
   - I am looking forward to hear the talk from Xenoh Kovah tomorrow at deepsec
   - John Butterworth, Xenoh Kovah et al.: “Problems with the Static Root of Trust for Measurement”, Blackhat 2013
   - Tarnovsky 2010 and 2013:
     - possible, but hard to do -> only for experts
   - Solutions:
     - due diligence in manufacturing process
     - physical access control to critical server (host) including TPM
4. TOCTOU
   - time of check – time of use
   - encryption of VMI using sealing

Issues with TPM-based solution

5. Is the cloud provider willing to disclose the details of his platform (OS, hypervisor, version, policy..)
   - Which is necessary for attestation!
   - Most providers use principle of “security by obscurity”

   - Solution: Property-Based attestation

   [Zhen-Hu Ning et al.: Property-Based Anonymous Attestation in Trusted Cloud Computing]

Conclusion

- Trusted Computing + MAC are very promising technologies to increase trust in cloud services/ providers.
  - complement each other
- Besides this technical base an ecosystem is needed:
  - **Whitelist** of trustworthy (tested) cloud platforms
  - entity that tests the overall system (OS, hypervisor, SELinux policy)
  - entity that publishes hashes of "good" systems

- Our research outcome:
  - We were able to build a prototype that
    - makes it hard for an insider to access virtual machines
    - makes it impossible to manipulate logs
  - Cloud customer get's a proof that such a system is running on the provider's site

My Vision

- trust in cloud is similar to
- trust in a mountain guide

Future work

- use OpenAttestation and OpenStack instead of IMA
- refinement of MAC policies
- encryption (sealing) of Virtual Images
- use D-RTM in addition to S-RTM
  - incorporate time-based attestation

Thank you for your *attestation* ;-)

Questions?

email: my.name@fhv.at