Trusted Virtualization:

Challenges and Opportunities

MWSG@LBNL - Dec 7

Frank Siebenlist (franks@mcs.anl.gov)

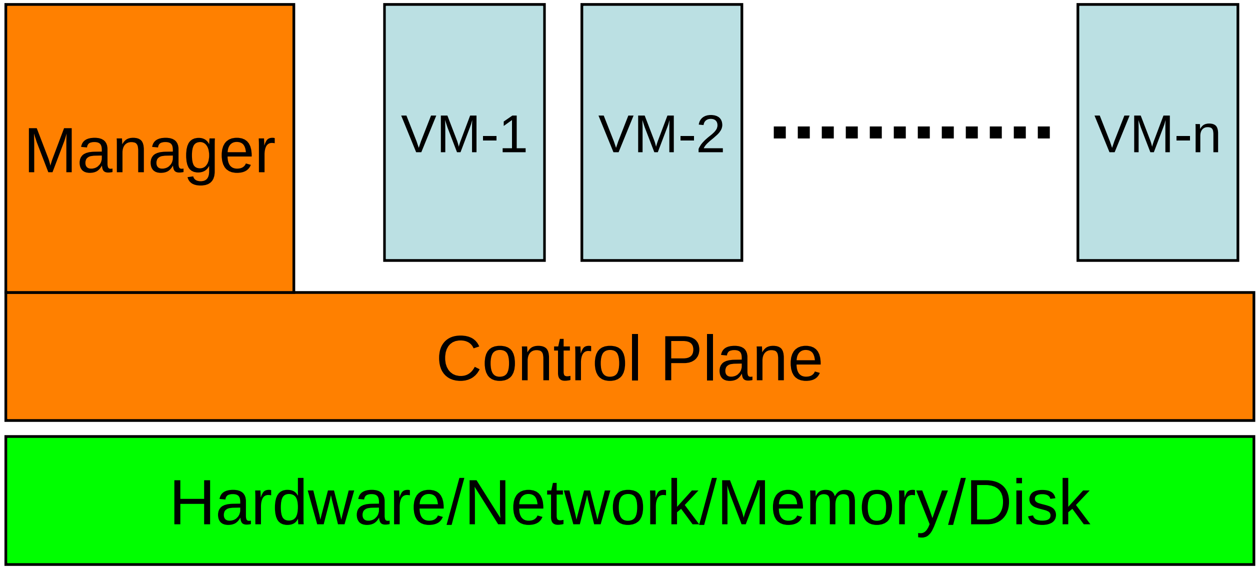
Abstract

- Unless you haven't been drinking Kate's VM-Kool-Aid, you will know that Virtual Machine technologies will be taking over the world over the next 5-10 years...
- According to all the marketing hype, VMs can make deployments more secure, but there are a number of challenges associated with the extra level of indirection that virtualization represents.
- Furthermore, there are also a number of new and novel opportunities to leverage VM-technologies to enhance the security.
- The purpose of the talk is to bounce off ideas and observations, and Frank is looking for an audience that isn't too shy...

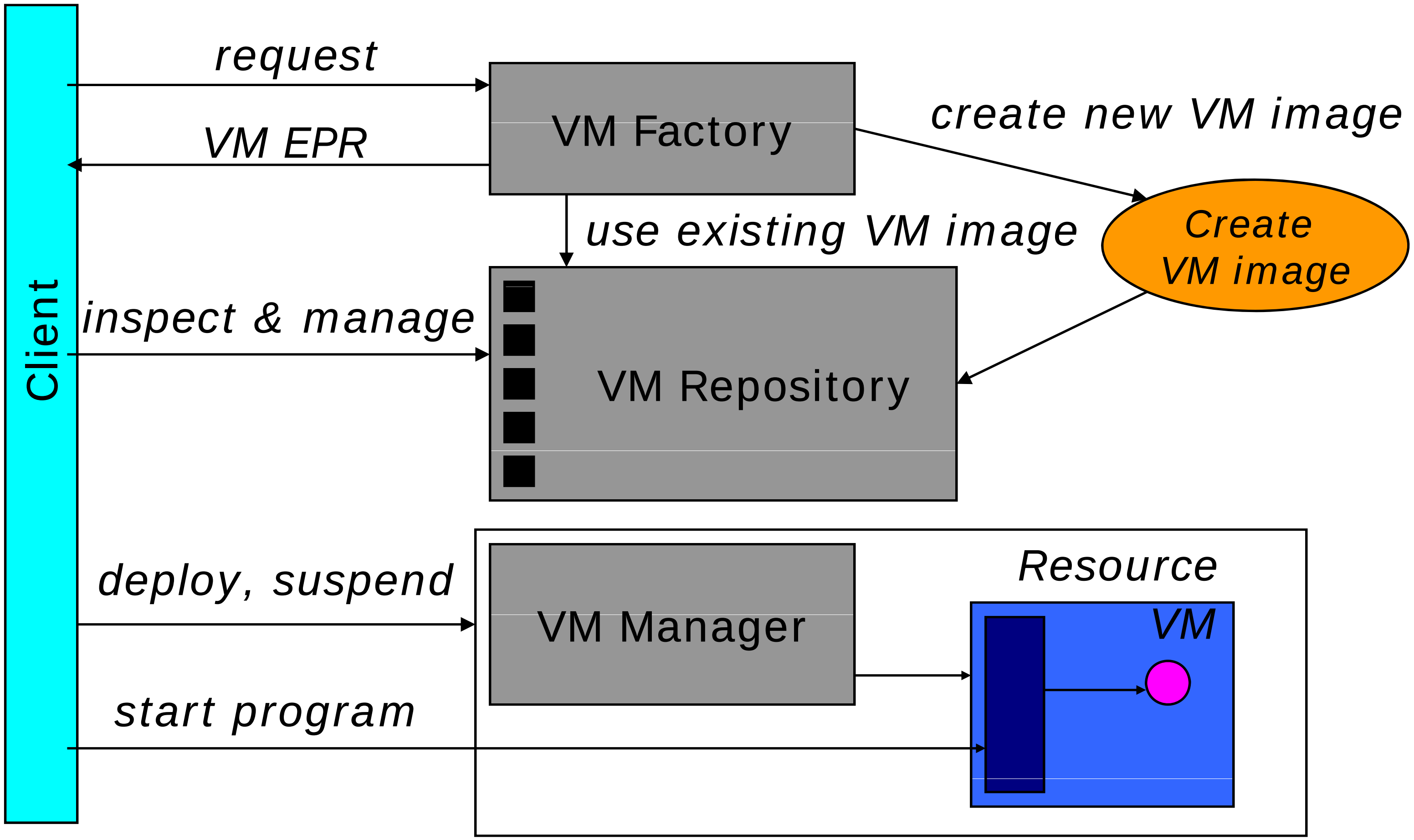
3/5/10 Year Prediction: VM Deployment Everywhere

- Every Network Service runs on a VM
  - 1 Service/VM if possible
- 10s-100s-1000s of VMs per physical Server
  - 10s-100s of cores/CPU, multiple CPUs/board
- All desktop/laptops/PDAs/cellphones/??? everything runs their OSs/apps in VMs
  - VMM/Hypervisor is pushed into the BIOS
- Commercial IT-world, data centers, clusters, all have fully adopted VM-technologies

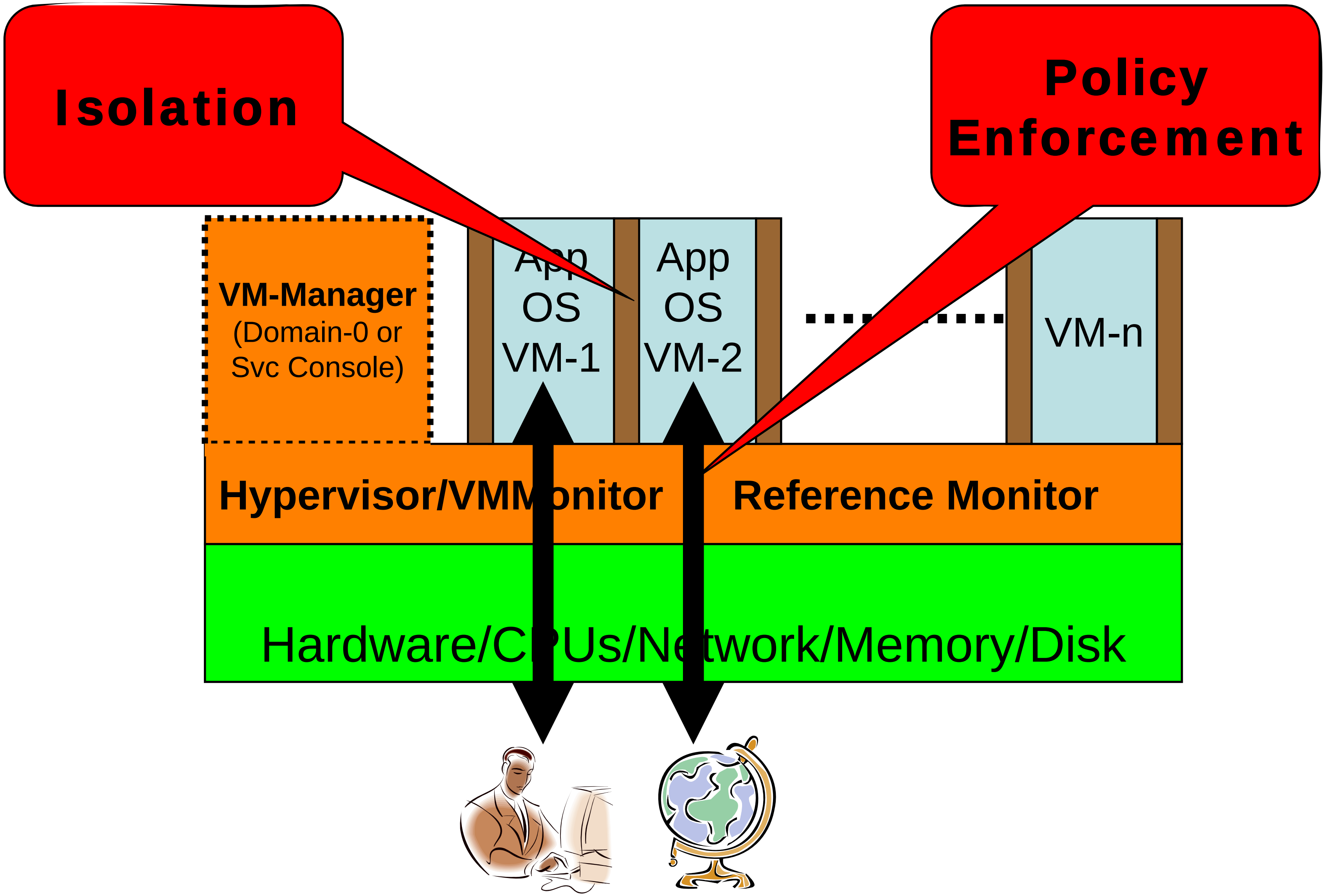
VMs & VMM


Manager
VM-1
VM-2
VM-n
Control Plane
Hardware/Network/Memory/Disk

How do Grids and VMs Play Together?



Virtual Machine



VMs and Security

- VM Insulation/Isolation/Compartmentalized
  - VMs don’t “see” each other
  - Limited consequences of compromise (single VM)
- Hypervisor/VMMonitor “transparent” control/monitoring
  - Real-time policy enforcement of network/memory/disk/cpu access
  - Monitor bandwidth/memory/disk/cpu usage
  - Throttle bandwidth/memory/disk/cpu usage
- Freeze, Migrate, Replicate VM-images
  - Forensic evidence frozen
  - “Menu-svc” to prepare commodity/custom-made configs
- Security policy becomes part of the SLA between the VM-host and VM-owner
  - Service Level Agreement about use of ports, network, libs, cpu, external access, “behavior”, etc. (includes security components)
  - Enforce Least Privilege Model
    could limit bot-net/army capabilities

Challenges because of Virtualization

Challenge:
Assurance of VM's Hosting Environment

- The virtualization of resources introduces an additional abstraction that complicates the policy enforcement for a VM-user who requires assurances about the location, type, or kind of hardware that hosts the hypervisor
- The use of secure hardware components, like integrated TPM, could help to attest the trust chain from the application service running on a VM running on a hypervisor running on a specific machine that has an embedded TPM

Where does my Service run?

- “Somehow” I received an EPR for a Service
  - Through broker/discovery/directory svc
- Policy: Only run on DOE-approved Compute Facilities
- Where and how do I get the assurance that my Service conforms?
  - Virtualization adds additional level of abstraction/indirection
- How can we “anchor” the trust on the HW?
  - Compute resource users have similar interest as the DRM-folks of the movie/music industry
  - Trusted Computing Platform (TCP) may/can help…

TCP-HW=>VMM=>VM-image=>OS=>app…=>user

Challenge:
Correctness of Hypervisor Security Execution

- The overall protection of the VM’s from the outside world as well as from the other hosted VMs relies on the integrity of the hosting system, i.e. the integrity of the hypervisor software and correctness of the policy enforced by its reference monitor.
- In order to limit the number of bugs in the hypervisor code, the code base must remain as small as possible and must be formally proven secure where possible.
- The correct and unambiguous enforcement of the policy by the reference monitor as it is derived from the SLAs and higher-level site-policies is another concern.

Privileged Domains/Partitions

- The Hypervisor may be “small” and “proven secure”, but…
- The VM-Manager (Dom-0) is not
  - Equivalent of “root”
  - Compromised Dom-0 => All VMs are Compromised
- VM-Manager often facing internet…

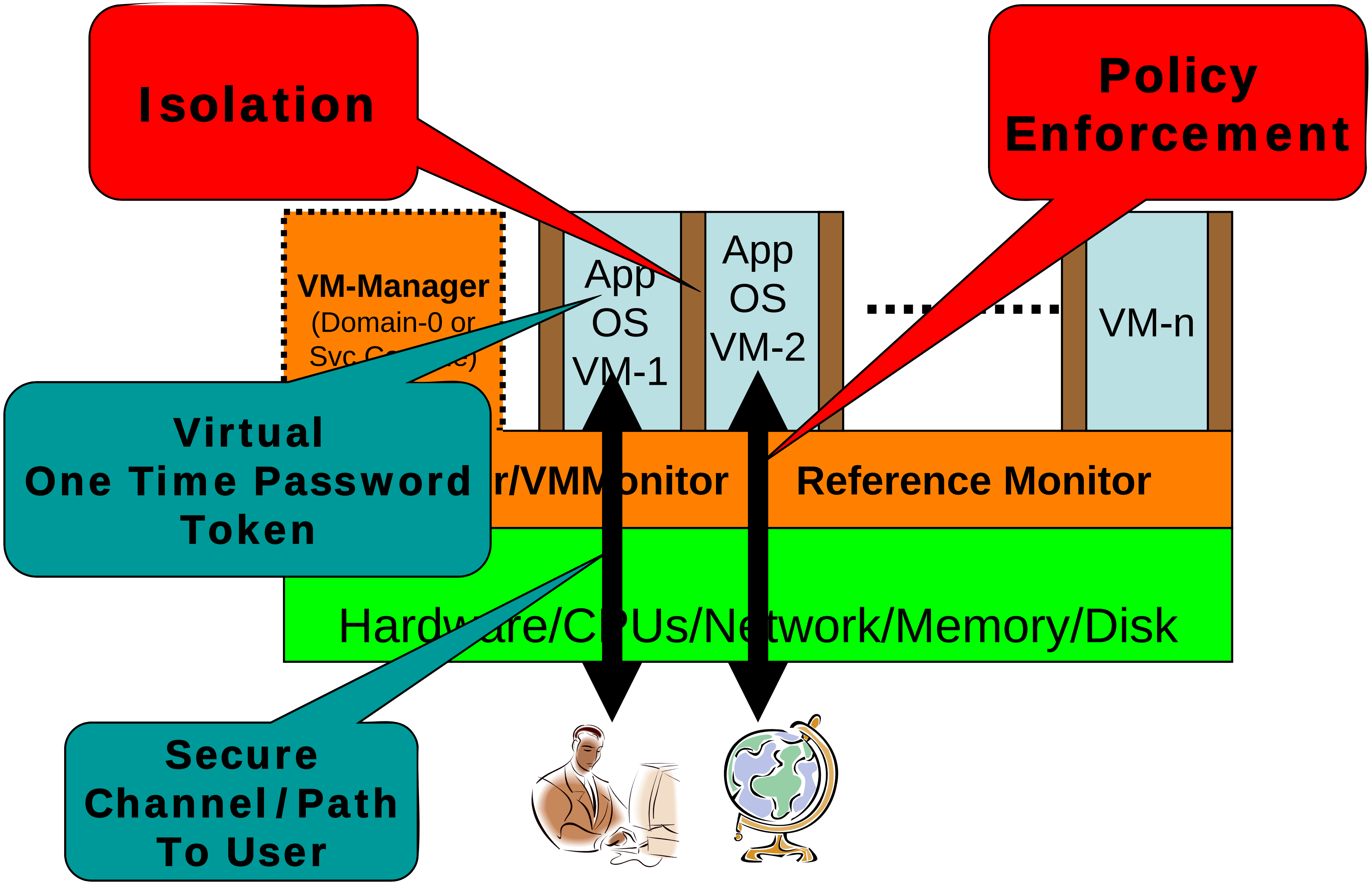
- Need ways for compartmentalize or split responsibilities among multiple privileged VMs
  - Not trivial… weakest link

VM-Instance Identity

- VMs are ad-hoc created, snapshot'ed, replicated, cloned, versioned, mirrored, migrated, restarted, rolled-back, …
- Running VM's state changes
  - Different VM from the one that was started
- VM's state (memory/disk) can be frozen
  - but also modified before a restart
- We need ability to:
  - "Identify" VMs: "VM-instance"
  - bind "meta-data" and "properties" to VM-instances
  - express/apply/enforce policy to VM-instances
  - log audit entries about VM-instances
  - specify VM-instance provenance data

Opportunities to Improve Security

Virtual OTP Token



Isolation

Policy Enforcement

VM-Manager
(Domain-0 or
Svc C[illegible]e)

App
OS
VM-1

App
OS
VM-2

VM-n

r/VMMonitor

Reference Monitor

Hardware/CPUs/Network/Memory/Disk

Virtual
One Time Password
Token

Secure
Channel/Path
To User

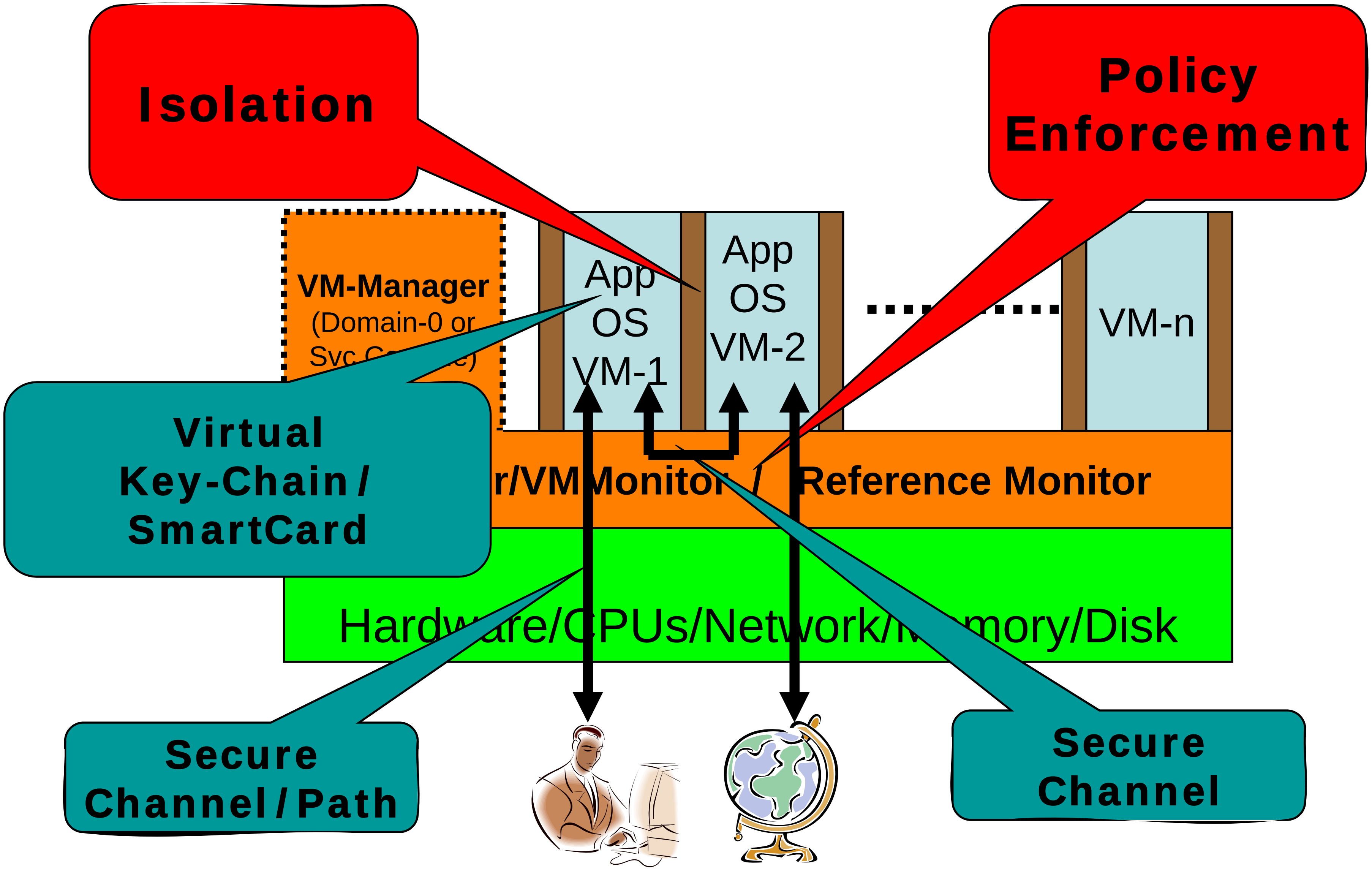
Secure Inter-VM Communication

- Inter-VM-Communication “managed” by Hypervisor
  - Connections and visibility of the communication are under Hypervisor’s control, i.e. are policy enforced.
- Inter-VM-Communications can be authenticated, and privacy and integrity protected without the need for any higher-level protocols like ipsec or SSL/TLS.
- Authentication on the VM-Id level
  - Similar to ipsec authN which is on the host-level

Trusted Security Service VMs

- Access to a VM can be restricted to only a single other VM managed by the same hypervisor and further restricted to a single communication mechanism and protocol.
- Off-load the secrets and crypto processing from a network attached VM to a non-network-accessible VM.
- Equivalent of using a VM as a smartcard or secure hardware device.
  - potential to limit the consequences of compromise but their feasibility requires further research.

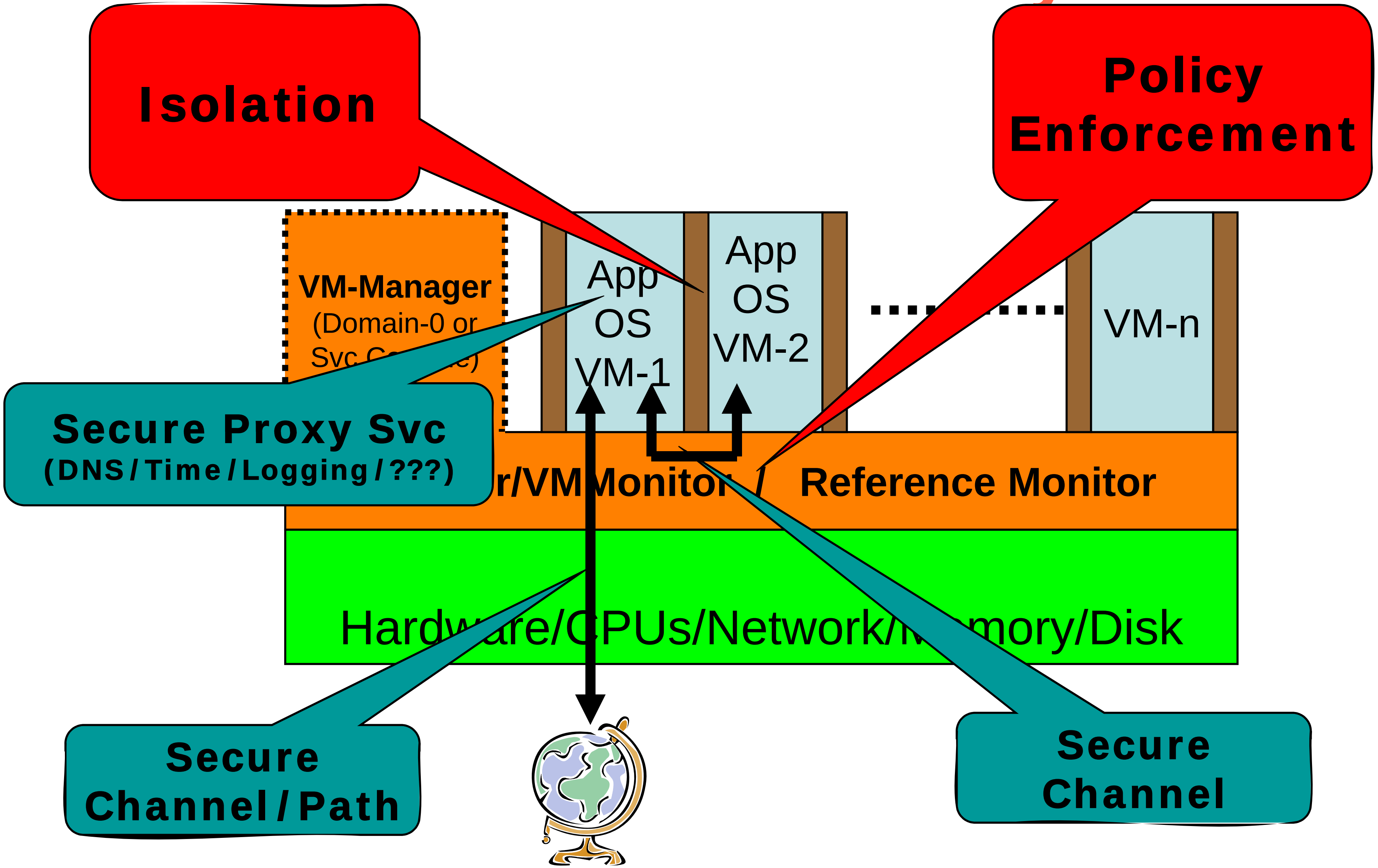
Virtual Smart Card



Secure proxy service VMs

- The inter-VM-communication subject to reference monitor’s policy enforcement and safe from snooping by other VMs or the outside world.
- Transparently provide security to insecure versions of protocols, like dns, snmp, smtp, by hosting a proxy service in a dedicated VM
  - Uses insecure protocols for the inter-VM communication
  - Communicating securely with the outside world through the secure versions of the protocols

Virtual Secure Proxy



Application Level Policy Enforcement

- Through transparent proxying of application-level protocols, incoming and outgoing communications can be monitored and policy can be enforced transparent from the applications.
  - WS-reverse proxy/firewall

Freezing the VM-State

- The execution of VMs differs from conventional computing environments in that applications can be stopped, frozen, serialized, replicated, migrated, and restarted/resumed on other hosting environments transparently.
- These features allow the higher-level ability to migrate, load-balance, and mirror resources based on demand and on deployment considerations.
- Unsuspecting applications may yield unintended results if application contexts are replayed.
  - Data-sets and memory-snapshots associated with such VM-images include long- and short-lived secrets that are used for authentication of the resource and the integrity of the communications which can be compromised if execution expectations are invalidated.

Provenance

- The VM-instance allows us to capture better provenance data such that results can be made verifiable repeatable
  - Transparent of applications…
- Requires close integration of VM-Monitor with Provenance System
  - Needs R&D…

Goal: Limit Chance and Limit Consequences of Compromise

- State of networked clients & services:
  - “hacked” or “to be hacked soon”
  - All systems will be hacked:
    not “if” but “when” and maybe “already”
- Fact of Cybersecurity Life
  - get over it - live with it
- Goal: Limit Chance of Compromise
- Goal: Limit Consequences of Compromise
- Non-goal: “make systems unhackable”

Limit Compromise Chance

- Minimize VM-OS’s functionality
- If we move to a single application per VM, then we can strip the OS’ functionality to the bare minimum needed for that application. Proving correctness for such a small OS is easier and the number of exposed bugs will be less.
- Single-User OS
  - Also simplifies inter-VM authentication
    VM == OS == single-user

Limit Consequences of Compromise

- Limit damage of possible compromise
  - Least privilege operation
- Detection of compromise
  - “Abnormal” behavior
- Limit damage of detected compromise
  - Isolation
- Investigation
  - Forensic evidence
- Determination of result integrity
  - Provenance
- Fast recovery
  - Roll-back to well-known state

Least Privilege Operation

- Minimize VM's privileges to those required for correct operation… and no more
- Service Level Agreement (SLA) should determine the required use of resources (cpu/memory/disk/network)
- More details in the SLA =>
  - Finer-grained enforcement of resource usage
  - Increased ability to monitor for abnormalities
  - Lesser chance for compromise to occur
  - Lesser chance for compromise to spread

Compromise Detection

- The ability of the hypervisor to observe the detailed use of the physical resources by a VM in real-time, can be used to detect abnormal actions, like access to unknown outside IP-addresses, modification of critical disk files, calls to new libraries, and unexpected CPU-usage spikes.
- The issue becomes how to define “normal or expected behavior”:
  1) Let the VM-user identify expected behavior as part of the SLA with the hosting party, like the use of ports, external services, local library calls, etc.
  2) The hypervisor can observe a known non-compromised VM over time and deduce “normal” patterns of resource usage.
  3) Scan the source/binary code of the VM for resource access calls, like open().

Isolate compromise

- VMs hosted by a hypervisor have the nice property that they are isolated from each other such that a compromised VM will not be able to compromise another VM or the hypervisor directly, such as via a rootkit equivalent.
- A compromised VM could still attack other VMs through any of the communication mechanisms that the hypervisor allows it to use. By using well-defined access control policies over VM resources and integrity-protecting interfaces for communication, we could further isolate the VM and limit its ability to compromise others.

Investigation of compromises

- As intruders and compromises become more sophisticated, we need more advanced forensic analysis options.
- Hypervisors can freeze a complete VM-image that includes OS, application, memory and disk-data, which constitutes a substantial amount of forensic information.
- In addition, when a compromise is expected, the hypervisor with its reference monitor could change the running application's environment into a honey-pot configuration for real-time tracking of the intruder's actions.
- The hypervisor could record a VM's detailed actions such that one could literally rewind and playback through the VM's life, which could facilitate investigations.

Recovery from Compromise

- After detecting and studying a compromise, the affected environment has to be cleaned-up and restarted in a known safe state.
- The hypervisor’s ability to freeze a VM’s state can be used to snapshot VMs during their life-cycle.
- These snapshots provide safe recoverable images, which could potentially save substantially on the time and nuisance associated with recovery from security violations.

Conclusion

- Interesting challenges associated with VM-security (trust, identity, correctness)

- VM-technologies could substantially improve the secure deployment of clients and services
  - Isolation, resource usage policy enforcement, compromise detection/recovery, secure VM-Svc, etc.

- Many exciting research opportunities left
  - Many topics are researched now/already
  - …time-window is limited…