Self-service Cloud Computing

Vinod Ganapathy
vinodg@cs.rutgers.edu

Department of Computer Science
Rutgers University

The modern computing spectrum

The Cloud

Web browsers and other apps

Smartphones and tablets

Security concerns are everywhere!

Can I trust Gmail with my personal conversations?

Can I trust my browser with my saved passwords?

Is that gaming app compromising my privacy?

Today's talk

The Cloud

Web browsers and other apps

Smartphones and tablets

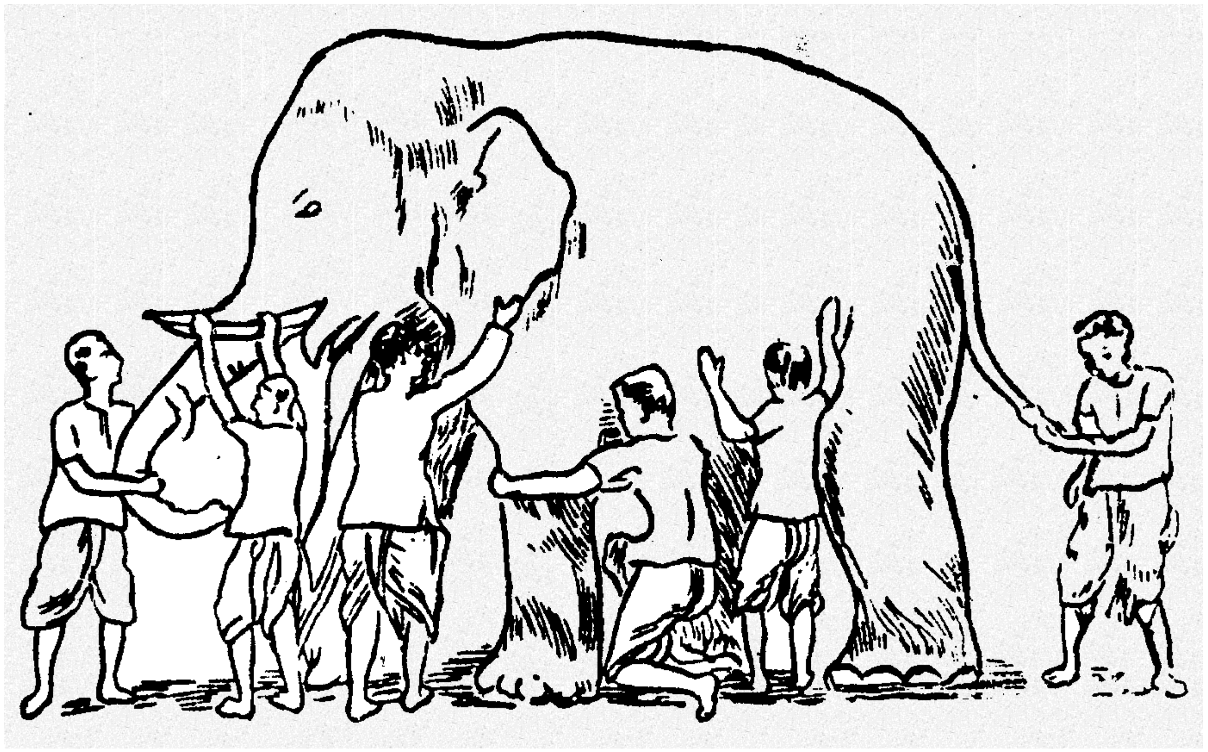
Self-service Cloud Computing

Shakeel Butt
Vinod Ganapathy

H. Andres Lagar-Cavilla
Abhinav Srivastava

What is the Cloud?

A distributed computing infrastructure, managed by 3rd-parties, with which we entrust our code and data.



What is the Cloud?

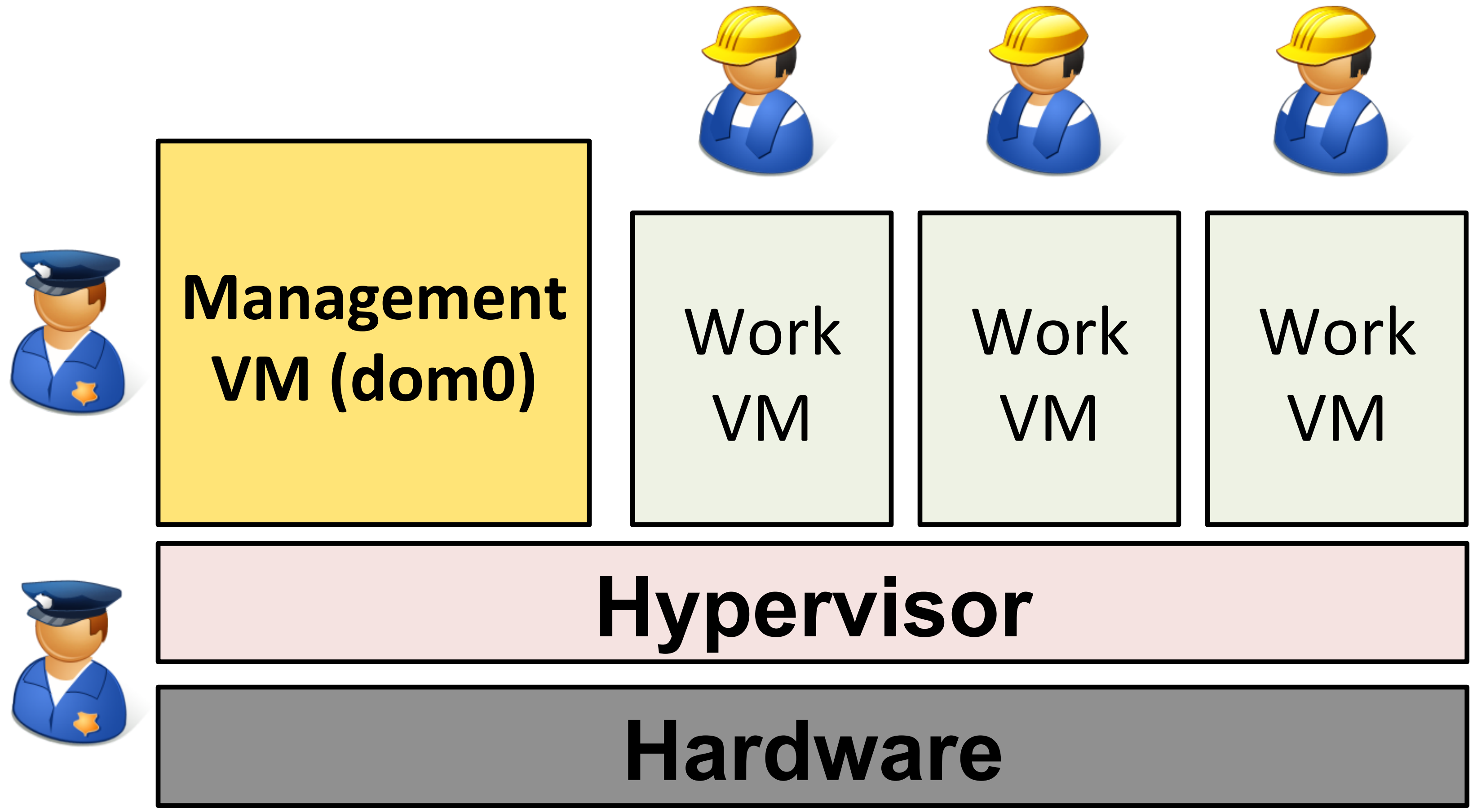
A distributed computing infrastructure, managed by 3rd-parties, with which we entrust our code and data.

- Comes in many flavours: ***-aaS**
  - Infrastructure-aaS, Platform-aaS, Software-aaS, Database-aaS, Storage-aaS, Security-aaS, Desktop-aaS, API-aaS, *etc.*
- Many economic benefits
  - No hardware acquisition/maintainence costs
  - Elasticity of resources
  - Very affordable: a few ¢/hour

I HIRED A CONSULTANT TO HELP US EVOLVE OUR PRODUCTS TO CLOUD COMPUTING.
BLAH BLAH CLOUD. BLAH BLAH CLOUD. BLAH BLAH CLOUD. BLAH BLAH CLOUD.
IT'S AS IF YOU'RE A TECHNOLOGIST AND A PHILOSOPHER ALL IN ONE!
BLAH BLAH PLATFORM.
Dilbert.com DilbertCartoonist@gmail.com
1-7-11 © 2011 Scott Adams, Inc./Dist. by UFS, Inc.

- By 2015, 90% of government agencies and large companies will use the cloud **[Gartner, "Market Trends: Application Development Software, Worldwide, 2012-2016," 2012]**
- Many new companies & services rely exclusively on the cloud, *e.g.*, Instagram, MIT/Harvard EdX **[NYTimes, "Active in Cloud, Amazon Reshapes Computing," Aug 28, 2012]**

Virtualized cloud platforms



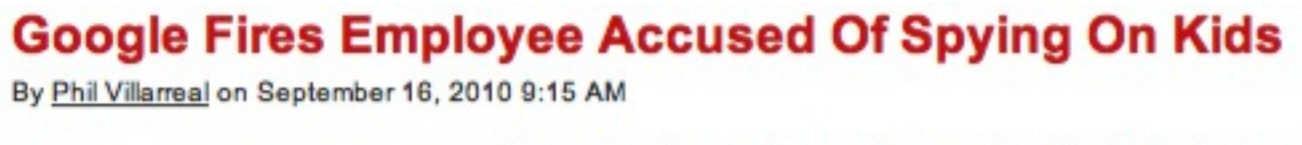
<u>Examples</u>: Amazon EC2, Microsoft Azure, OpenStack, RackSpace Hosting

Embracing the cloud

Embracing the cloud

Problem #1

Client code & data secrecy and integrity vulnerable to attack



Google Fires Employee Accused Of Spying On Kids

By Phil Villarreal on September 16, 2010 9:15 AM

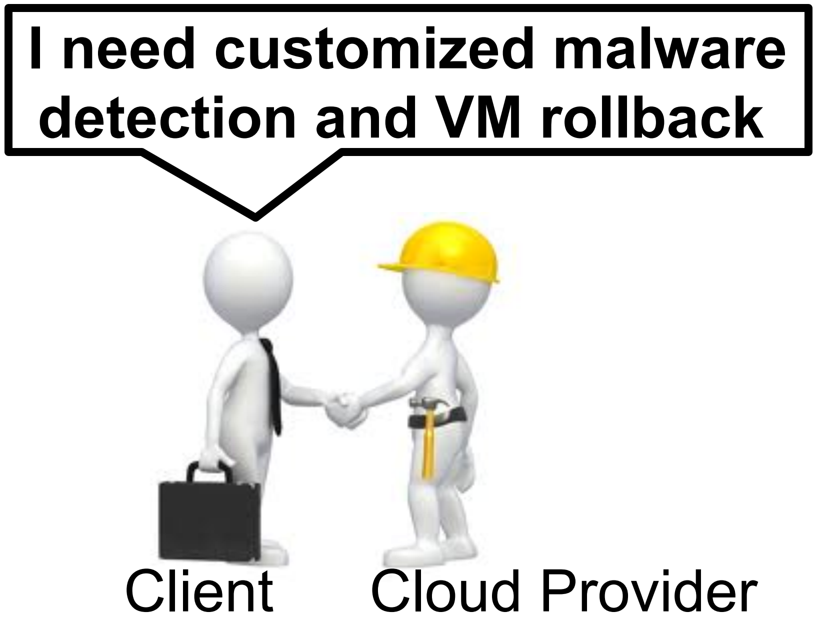
Embracing the cloud

CLOUD COMPUTING IS NO GOOD BECAUSE STRANGERS WOULD HAVE ACCESS TO OUR DATA.

Problem #1

Client code & data secrecy and integrity vulnerable to attack

Embracing the cloud

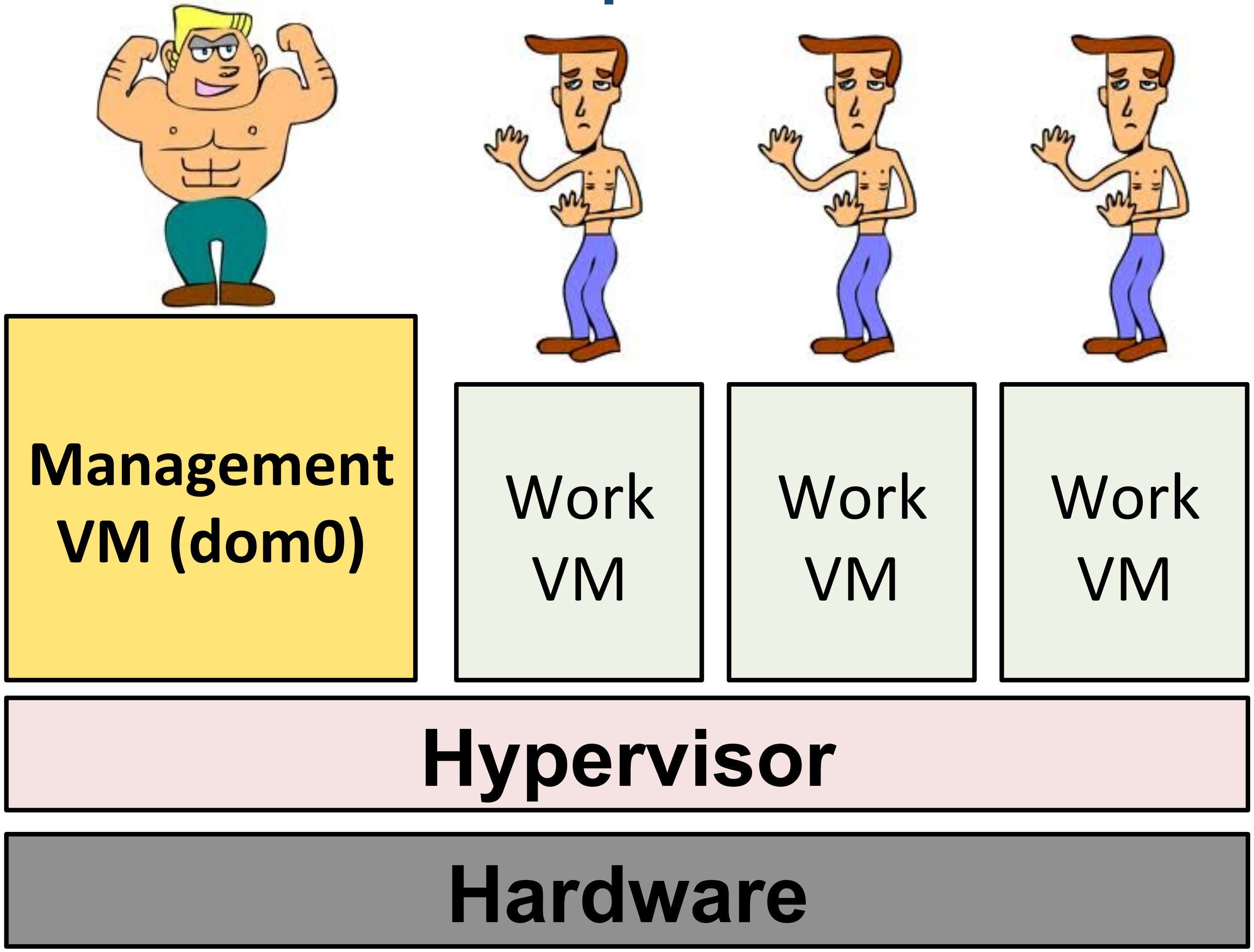


For now just have checkpointing …
Client
Cloud Provider

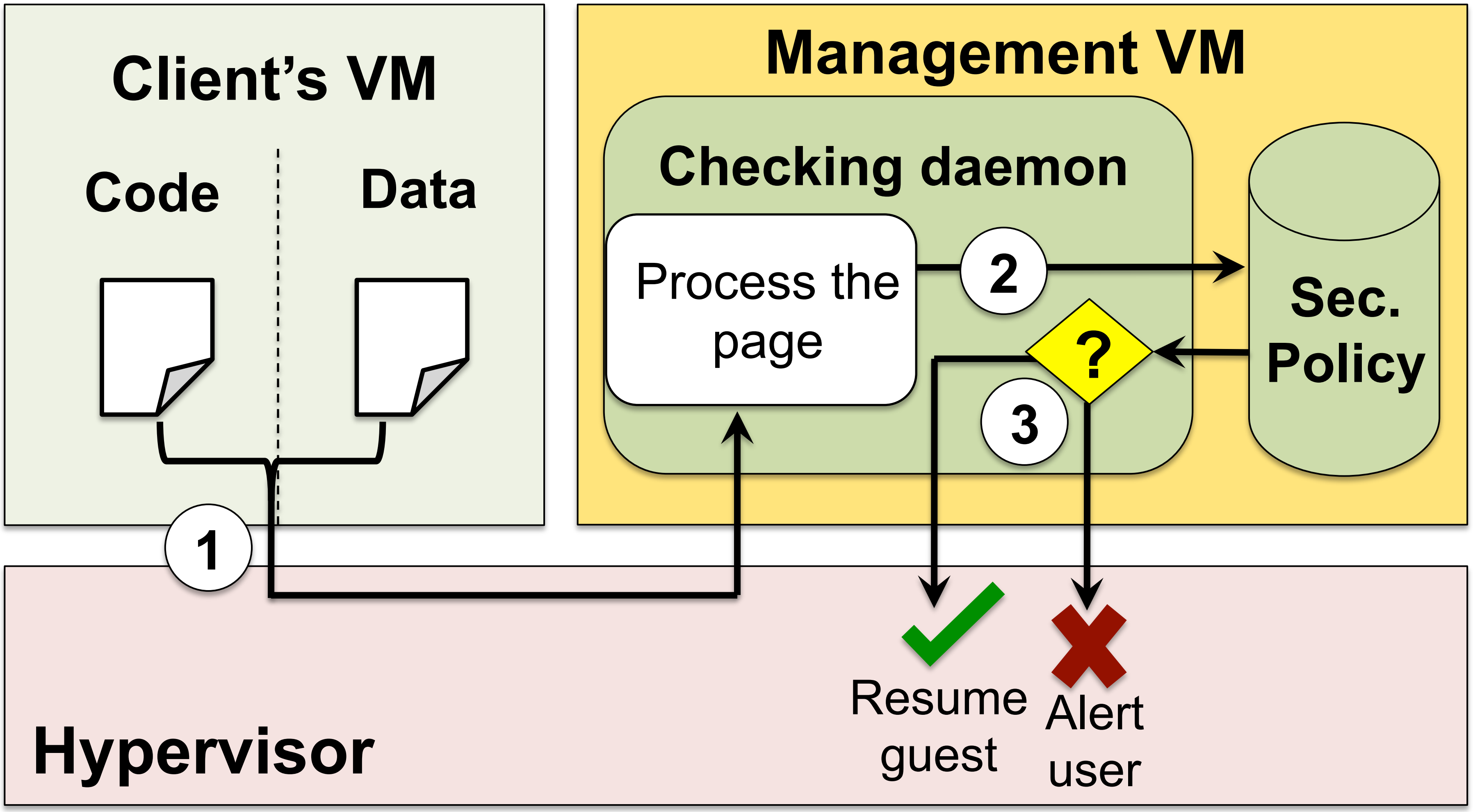
Problem #2

Clients must rely on provider to deploy customized services

Why do these problems arise?



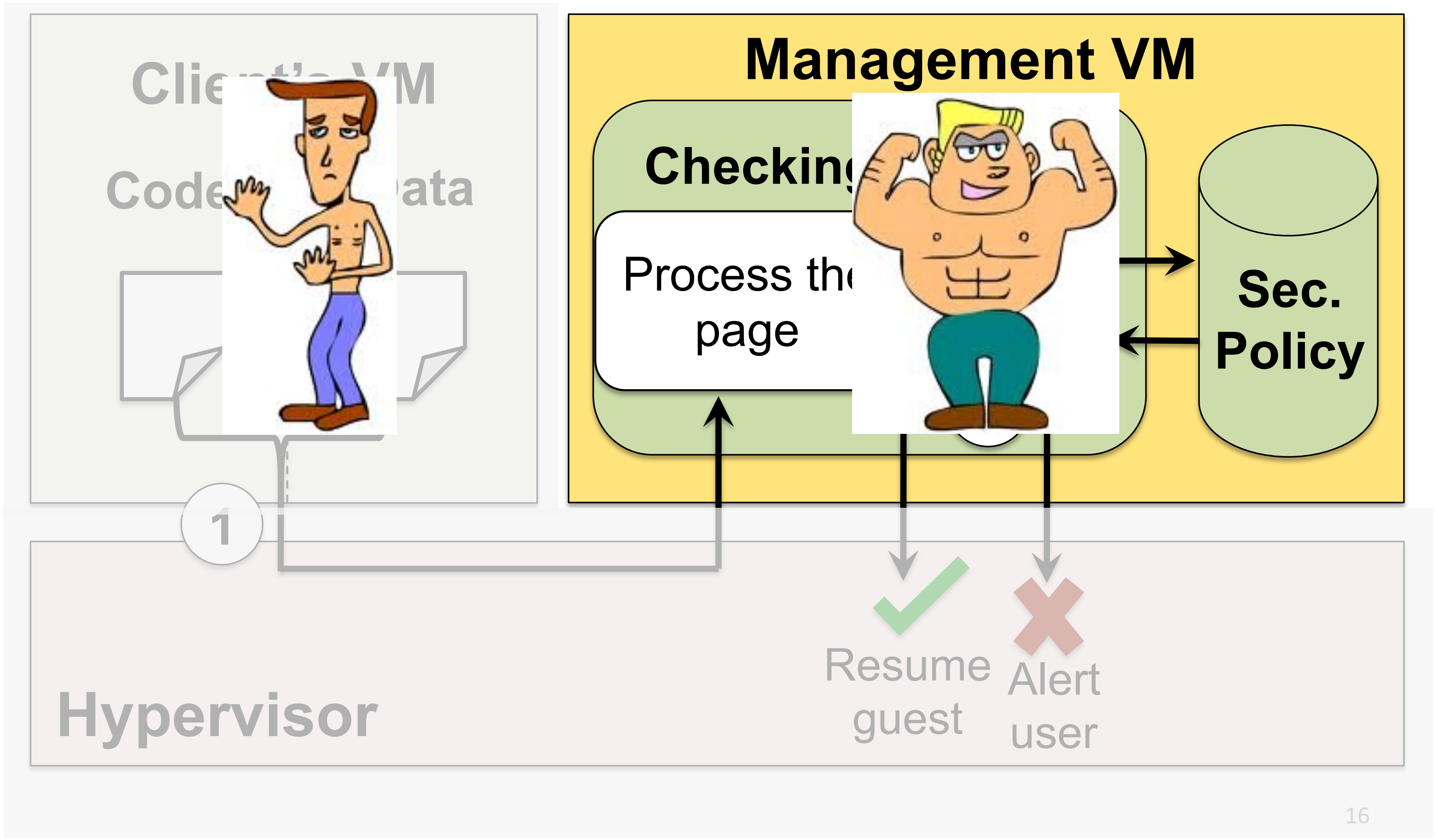
Example: Malware detection



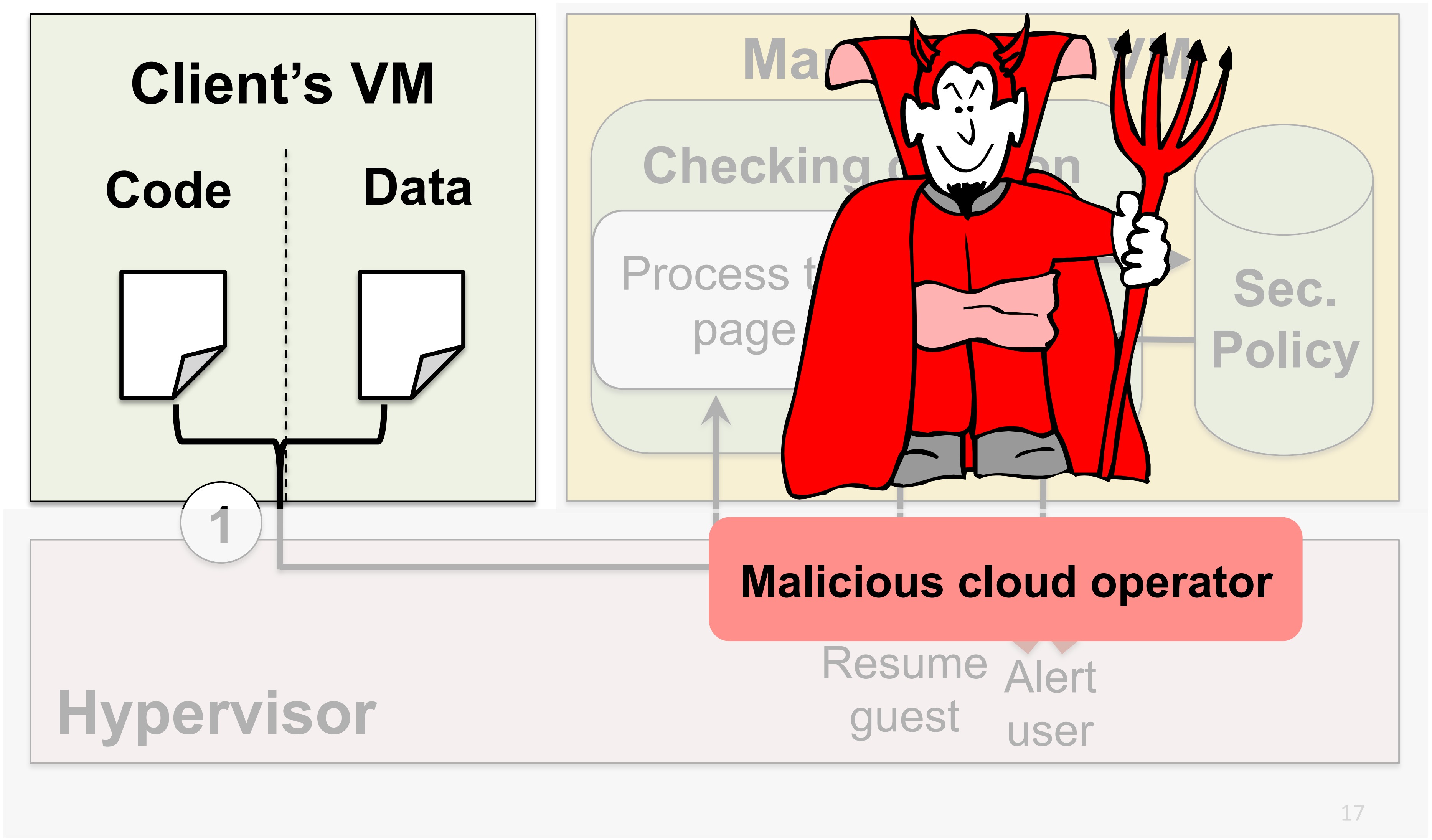
[Example: Gibraltar -- Baliga, Ganapathy, Iftode, ACSAC'08]

Problem

Clients must rely on provider to deploy customized services

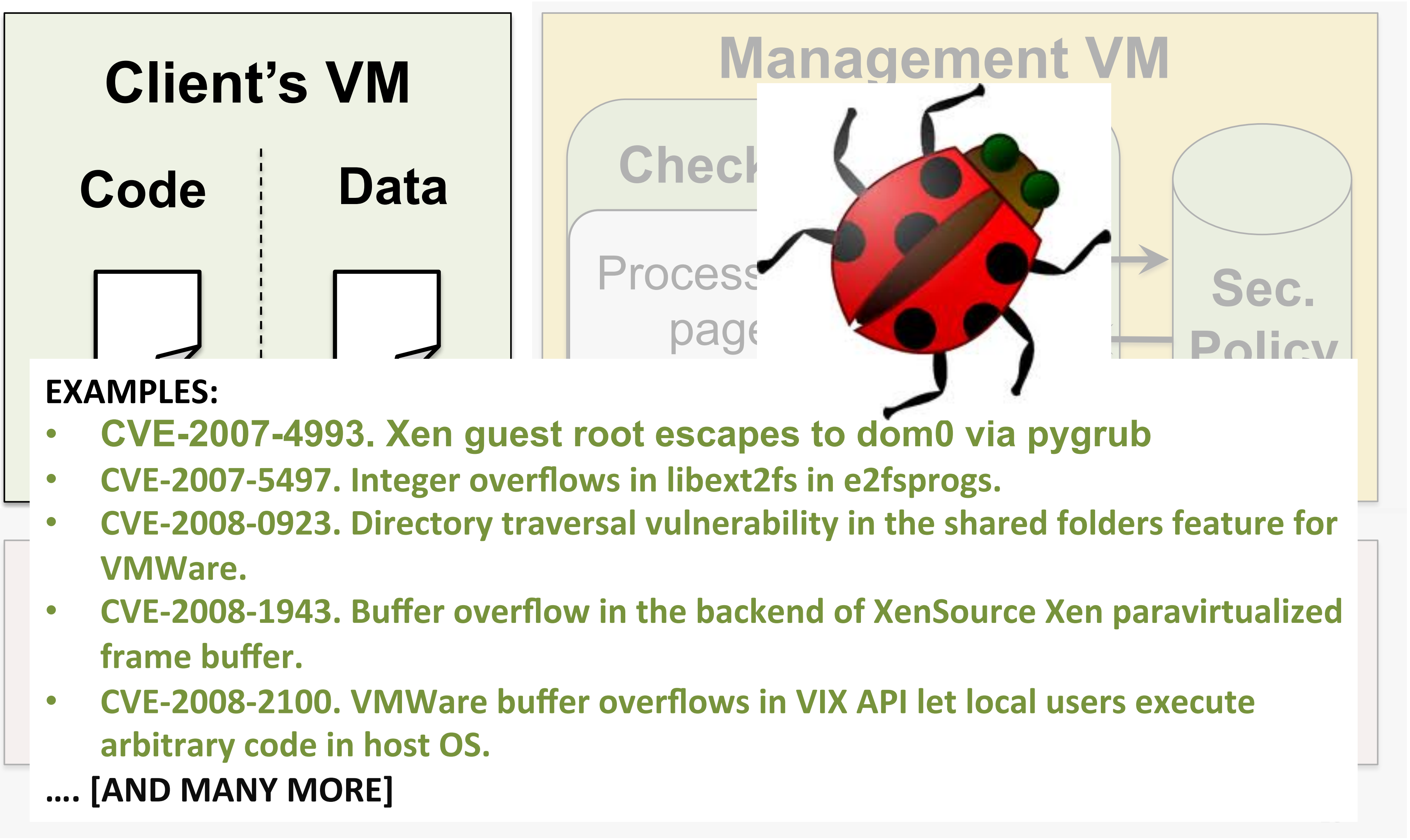


Problem Client code & data secrecy and integrity vulnerable to attack



Problem

Client code & data secrecy and integrity vulnerable to attack



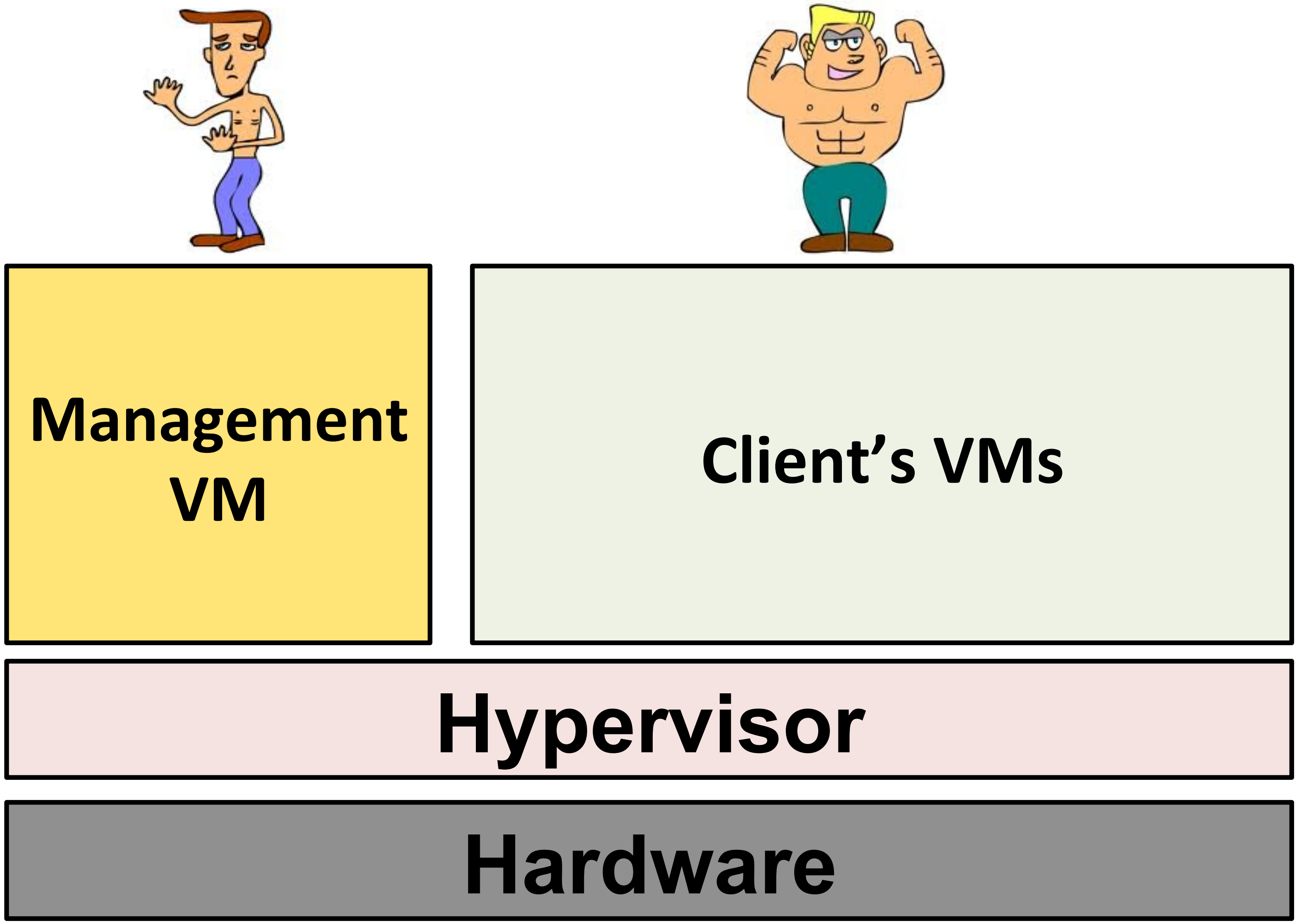
EXAMPLES:

- CVE-2007-4993. Xen guest root escapes to dom0 via pygrub
- CVE-2007-5497. Integer overflows in libext2fs in e2fsprogs.
- CVE-2008-0923. Directory traversal vulnerability in the shared folders feature for VMWare.
- CVE-2008-1943. Buffer overflow in the backend of XenSource Xen paravirtualized frame buffer.
- CVE-2008-2100. VMWare buffer overflows in VIX API let local users execute arbitrary code in host OS.

.... [AND MANY MORE]

Our solution

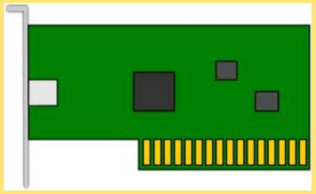
SSC: Self-service cloud computing



Outline

- Disaggregation and new privilege model
- Technical challenges:
  - Balancing provider's and client's goals
  - Secure bootstrap of client's VMs
- Experimental evaluation
- Future directions and other projects

Duties of the management VM



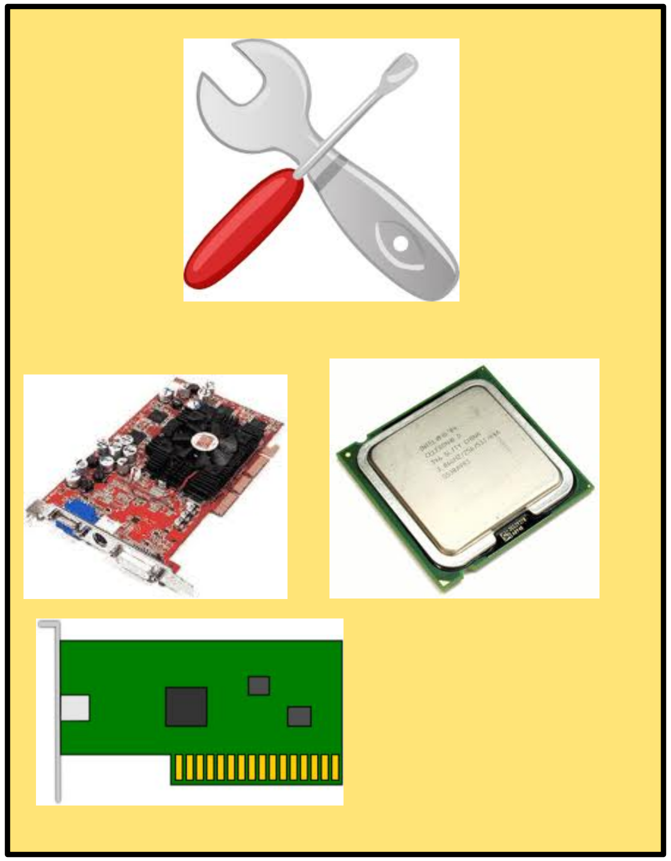
Manages and multiplexes hardware resources

Manages client virtual machines

Management VM (Dom0)

Main technique used by SSC

Disaggregate the management VM



System-wide Mgmt. VM (SDom0)

Per-Client Mgmt. VM (UDom0)

- **Manages client's VMs**
- **Allows clients to deploy new services**

Solves problem #2

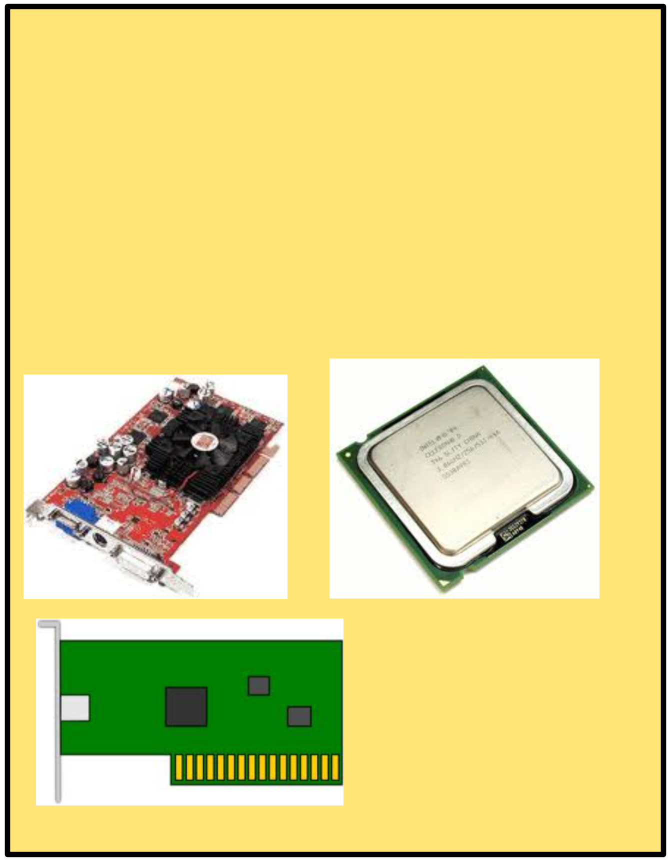
- **Manages hardware**
- **No access to clients VMs**

Solves problem #1

Embracing first principles

Principle of separation of privilege

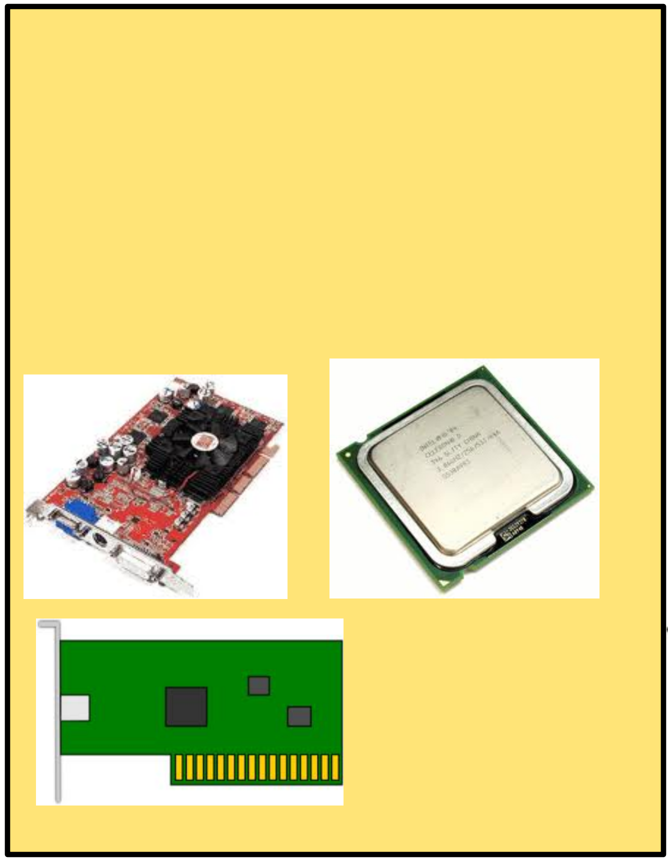
Per-Client Mgmt. VM (UDom0)



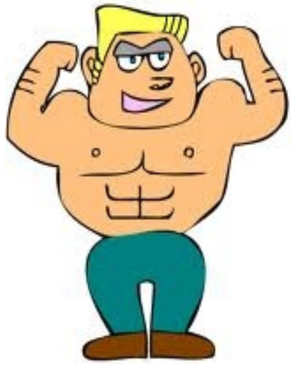
System-wide Mgmt. VM (SDom0)

Embracing first principles

Principle of least privilege

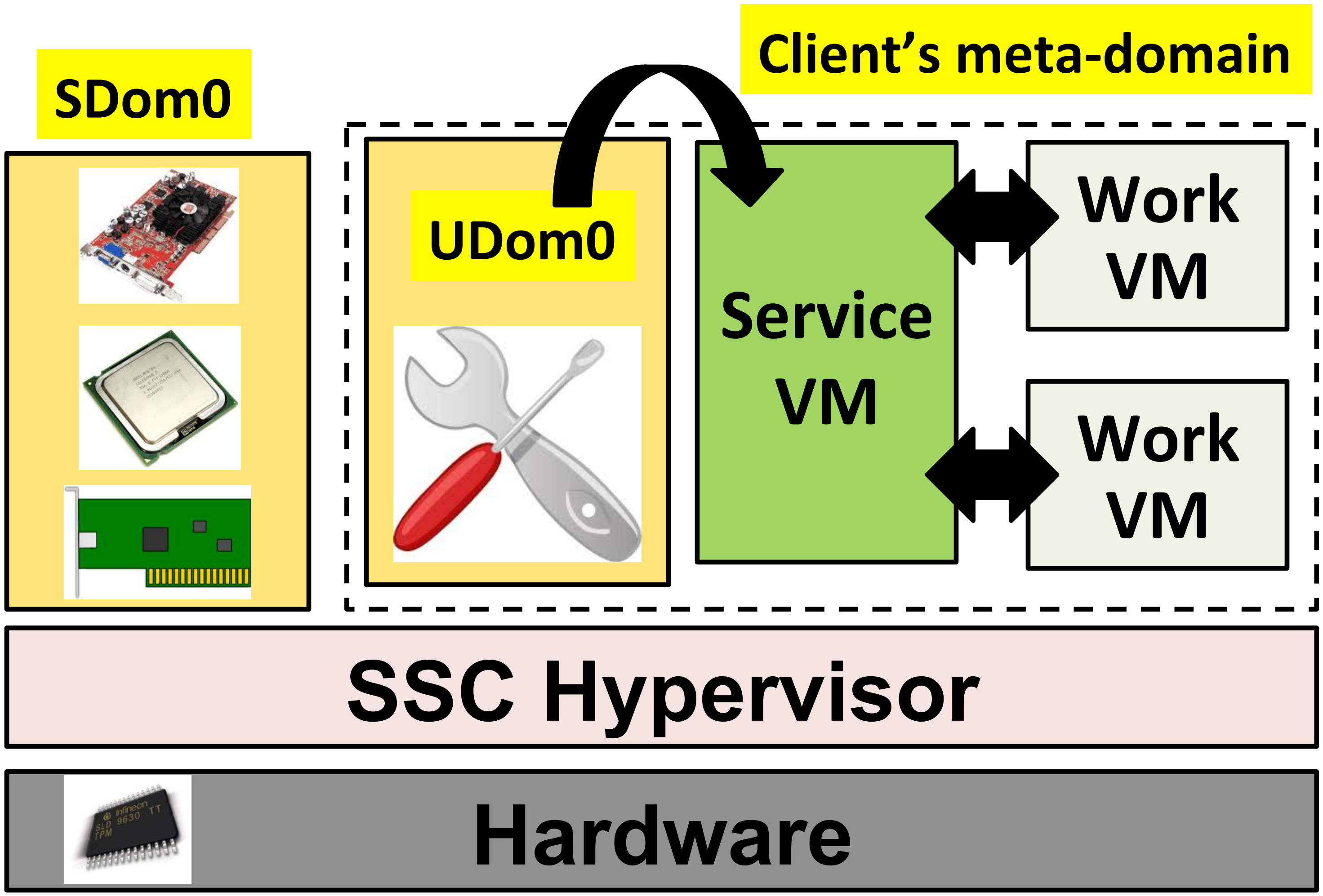


Per-Client Mgmt. VM (UDom0)



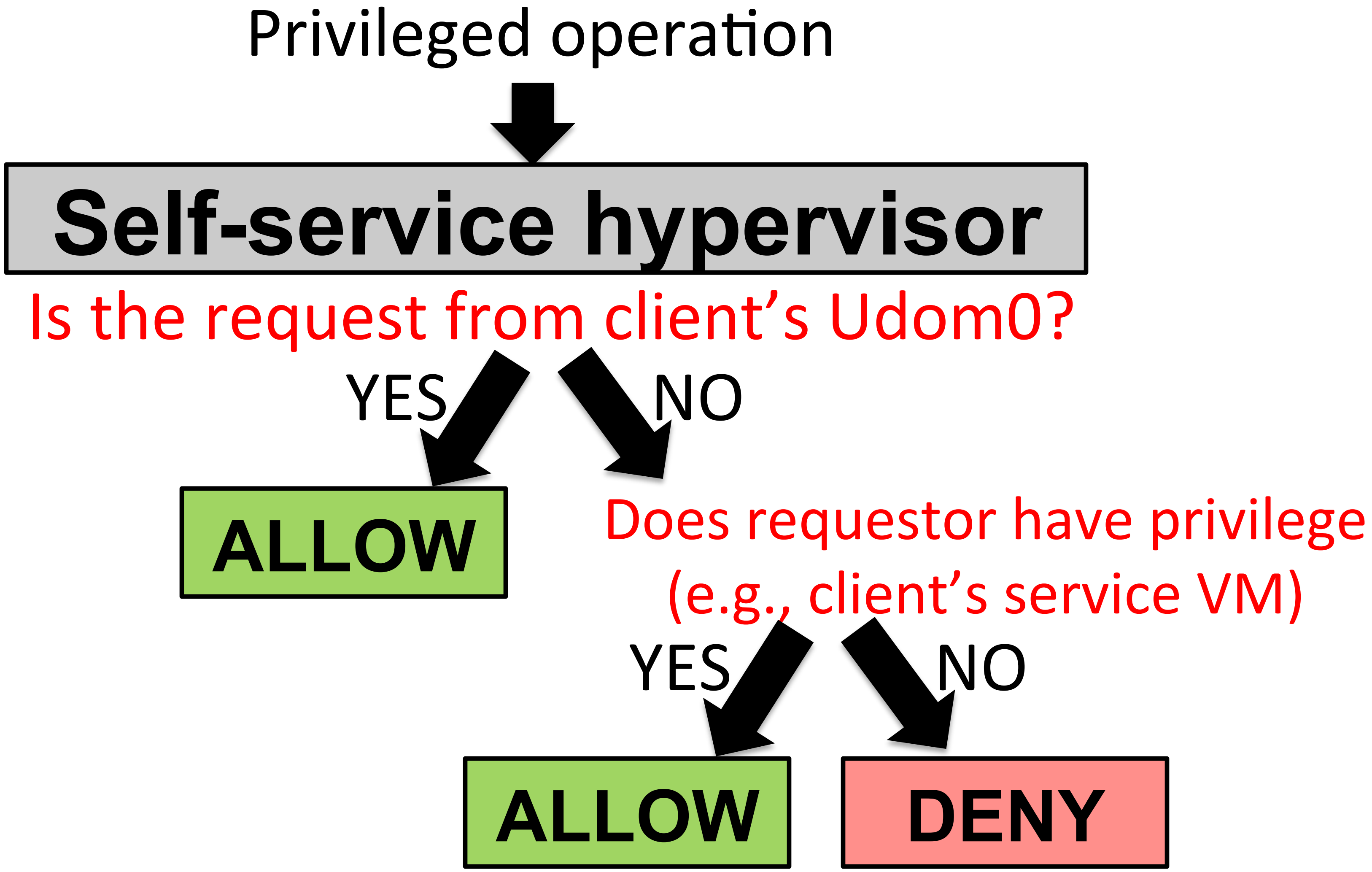
System-wide Mgmt. VM (SDom0)

An SSC platform



Equipped with a Trusted Platform Module (TPM) chip

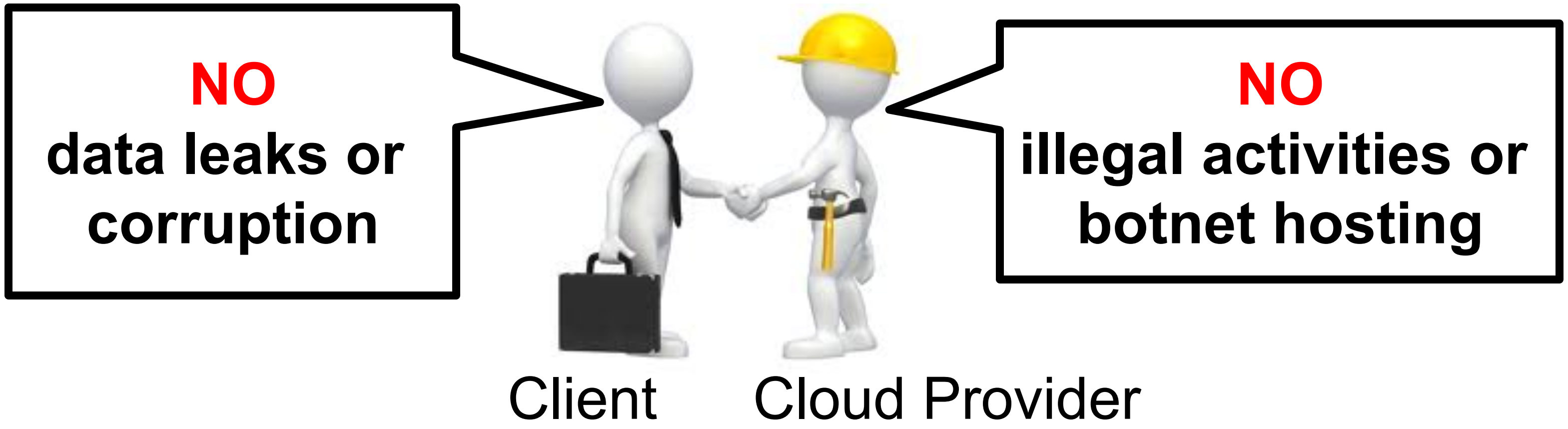
SSC's privilege model


Privileged operation
Self-service hypervisor
Is the request from client's Udom0?
YES
NO
ALLOW
Does requestor have privilege (e.g., client's service VM)
YES
NO
ALLOW
DENY

Key technical challenges

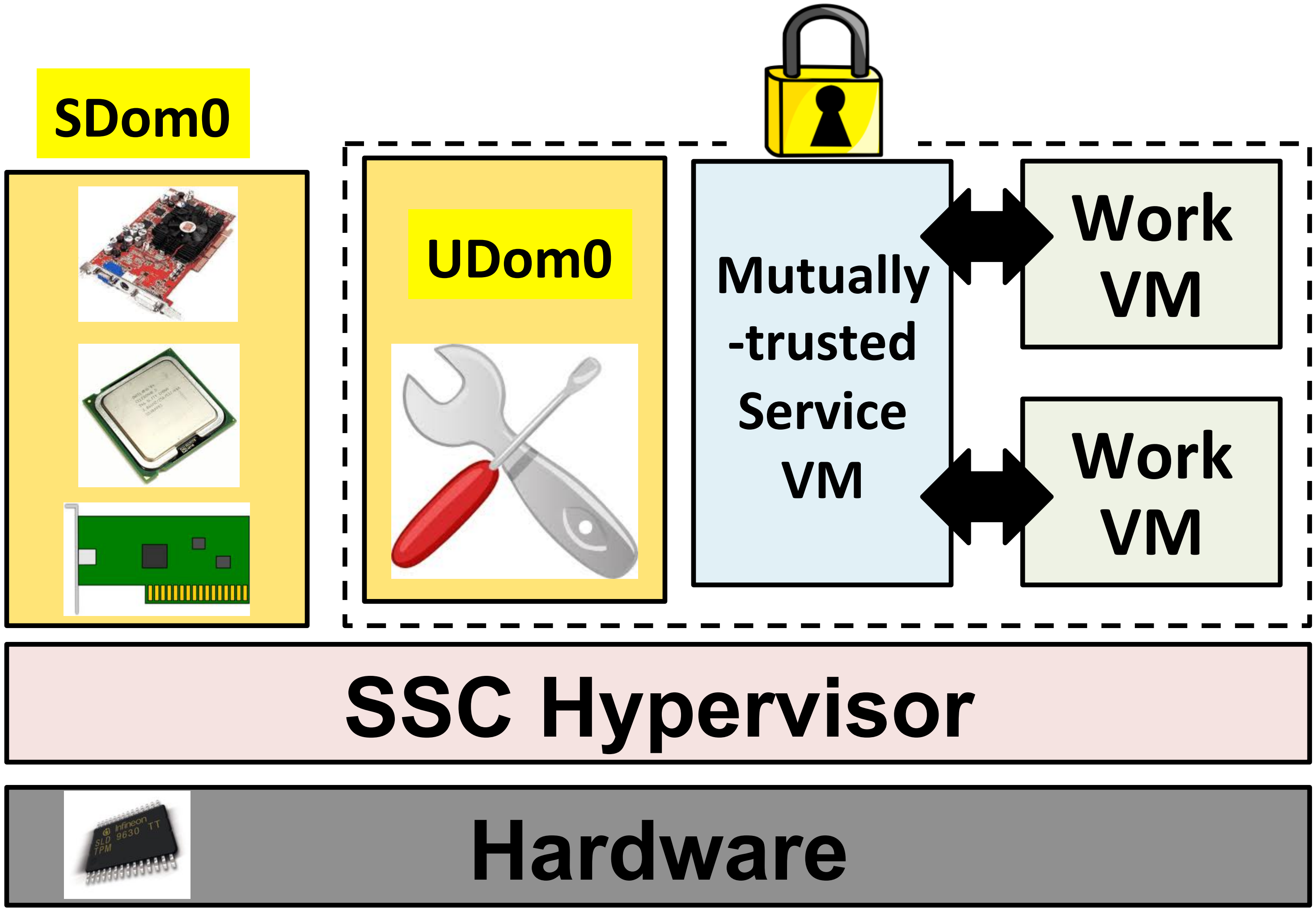
1. **Providers want *some* control**
   - To enforce regulatory compliance (SLAs, etc.)
   - **Solution**: Mutually-trusted service VMs
2. **Building domains in a trustworthy fashion**
   - Sdom0 is not trusted
   - **Solution**: the Domain Builder
3. **Establishing secure channel with client**
   - Sdom0 controls all the hardware!
   - **Solution**: Secure bootstrap protocol

Providers want *some* control

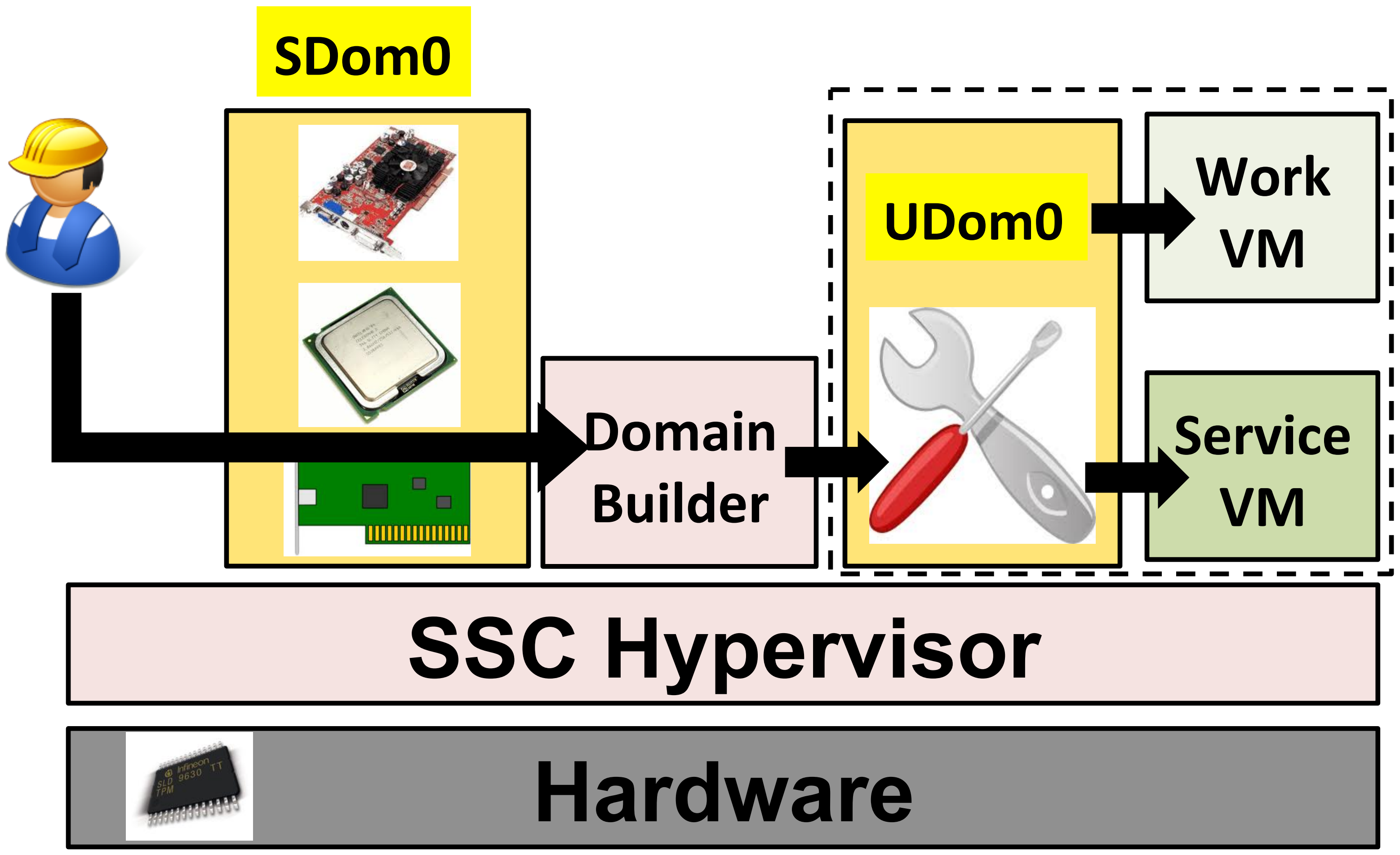


- Udom0 and service VMs put clients in control of their VMs
- Sdom0 cannot inspect these VMs
- Malicious clients can misuse privilege
- **Mutually-trusted service VMs**

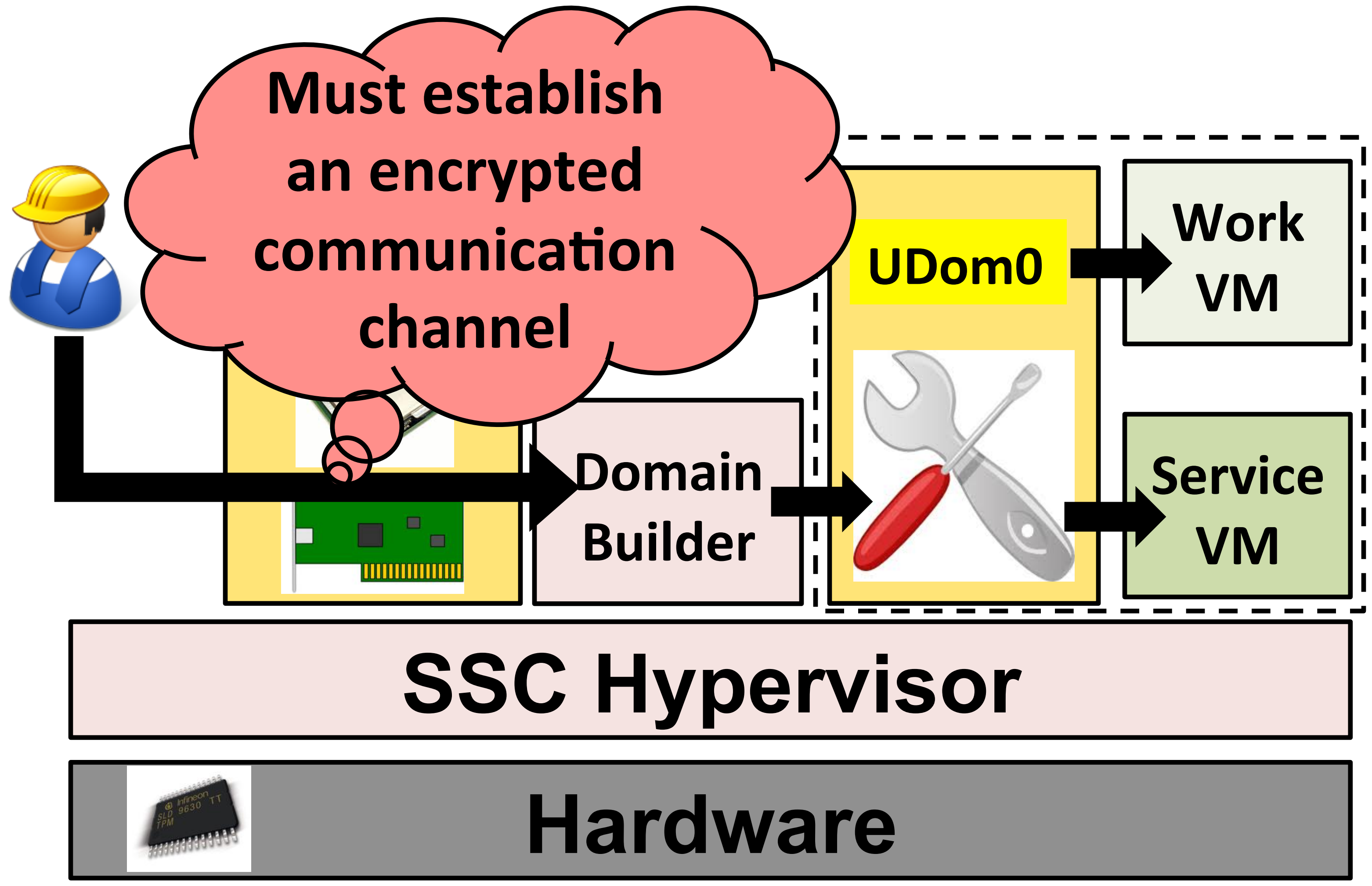
Trustworthy regulatory compliance



Bootstrap: the **Domain Builder**



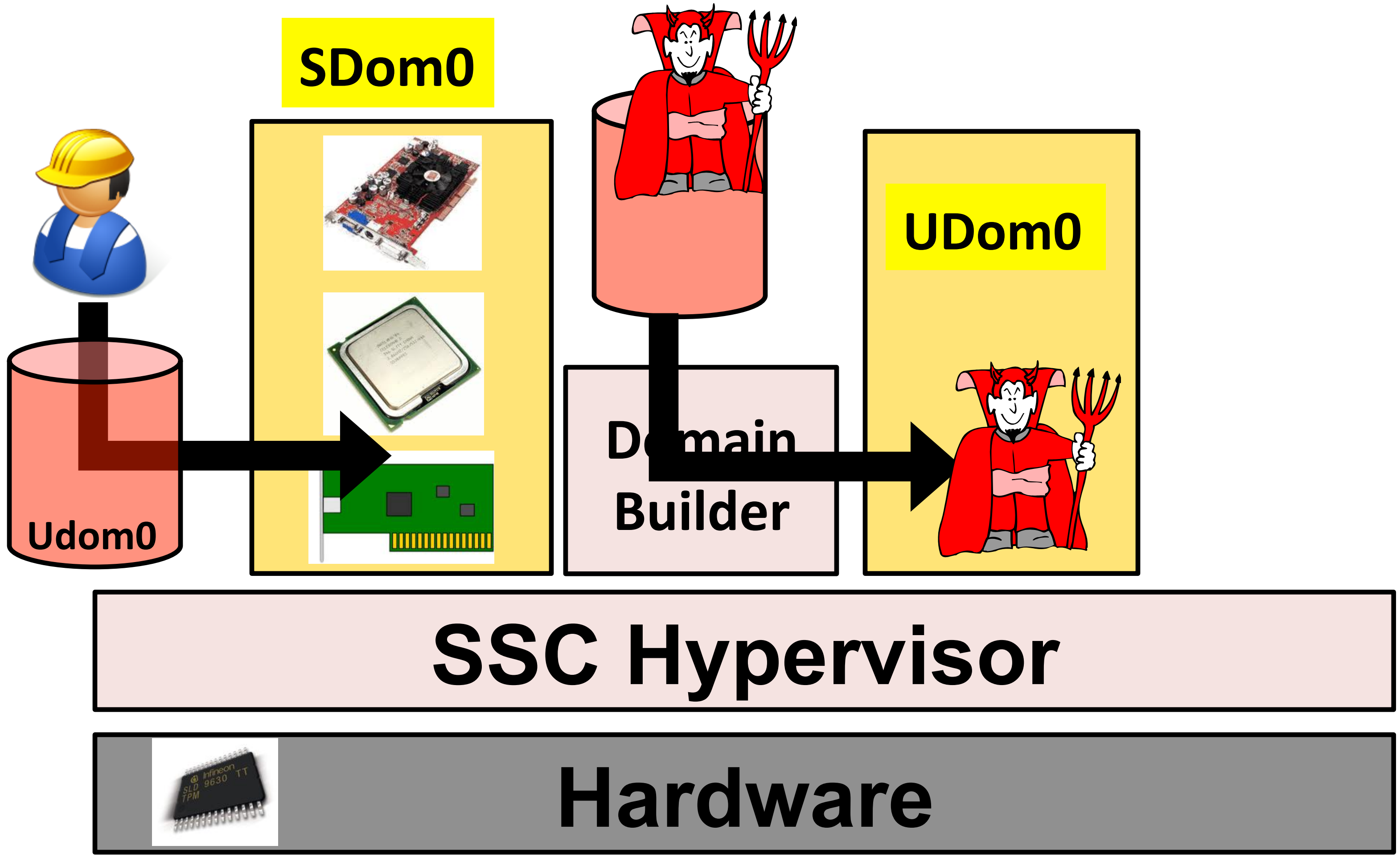
Bootstrap: the **Domain Builder**



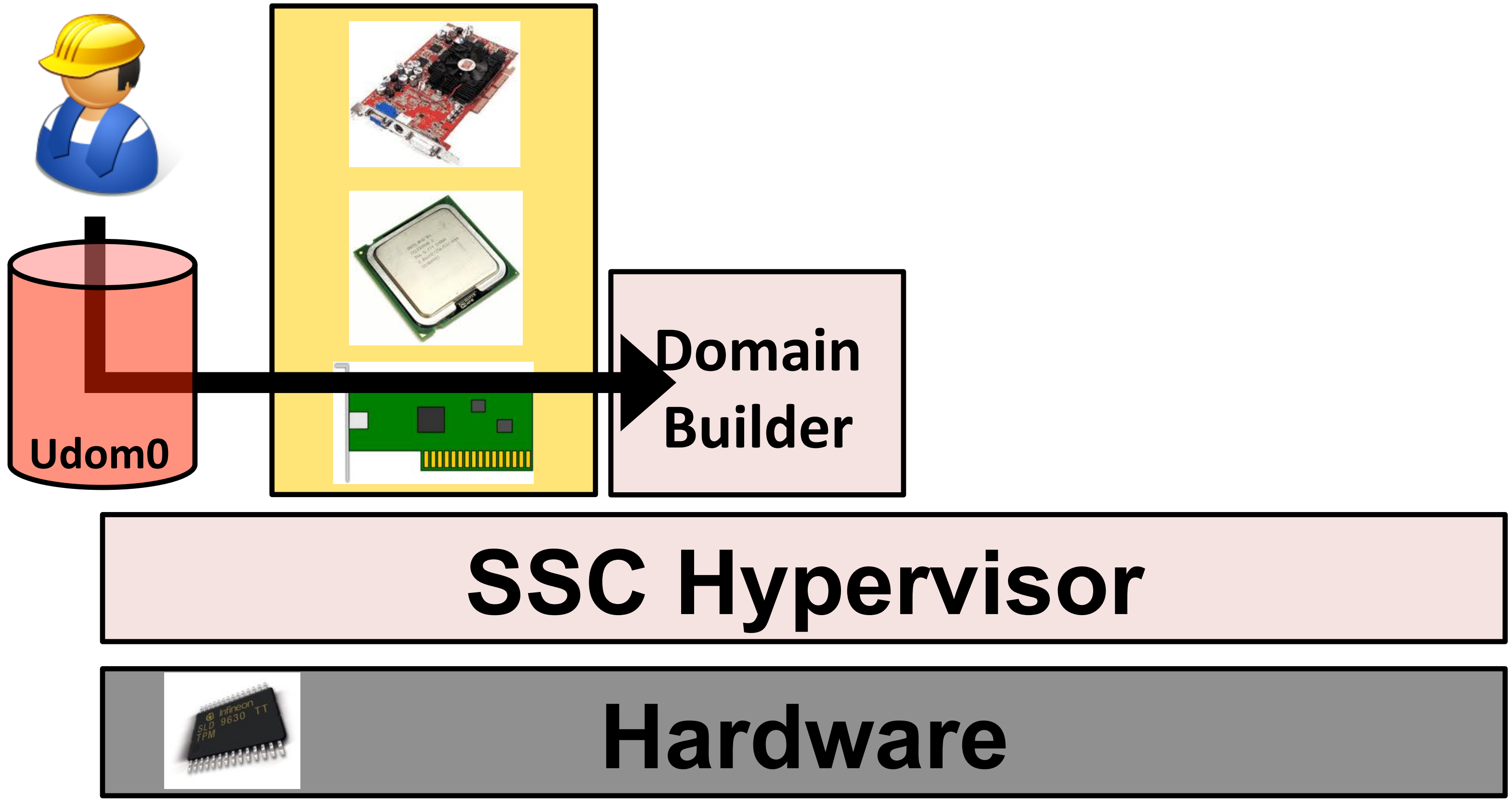
Secure bootstrap protocol

- **Goal:** Build Udom0, and establish an SSL channel with client
- **Challenge**: Sdom0 controls the network!
- **Implication**: Evil twin attack

An evil twin attack

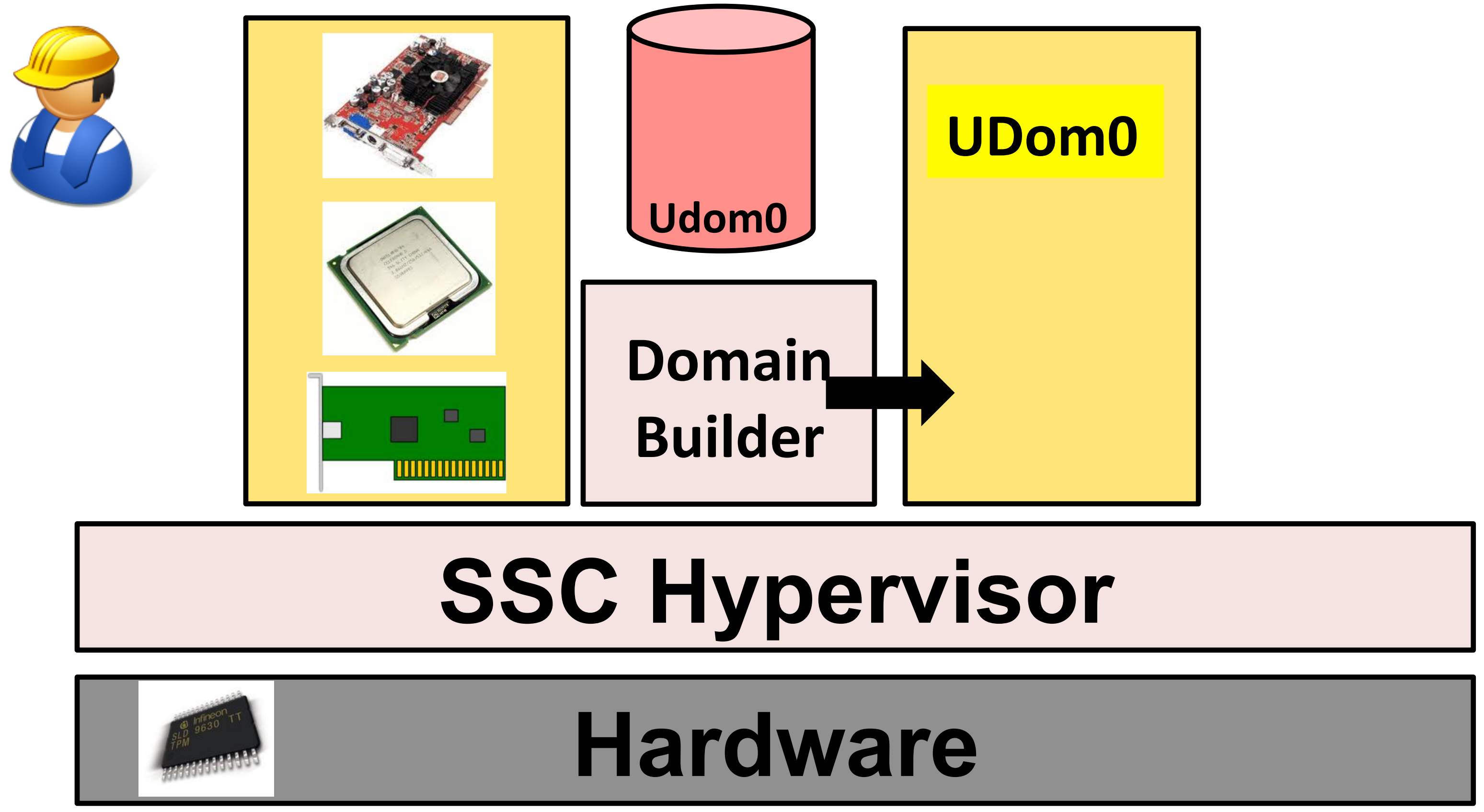


1 Udom0 image, Enc (,)



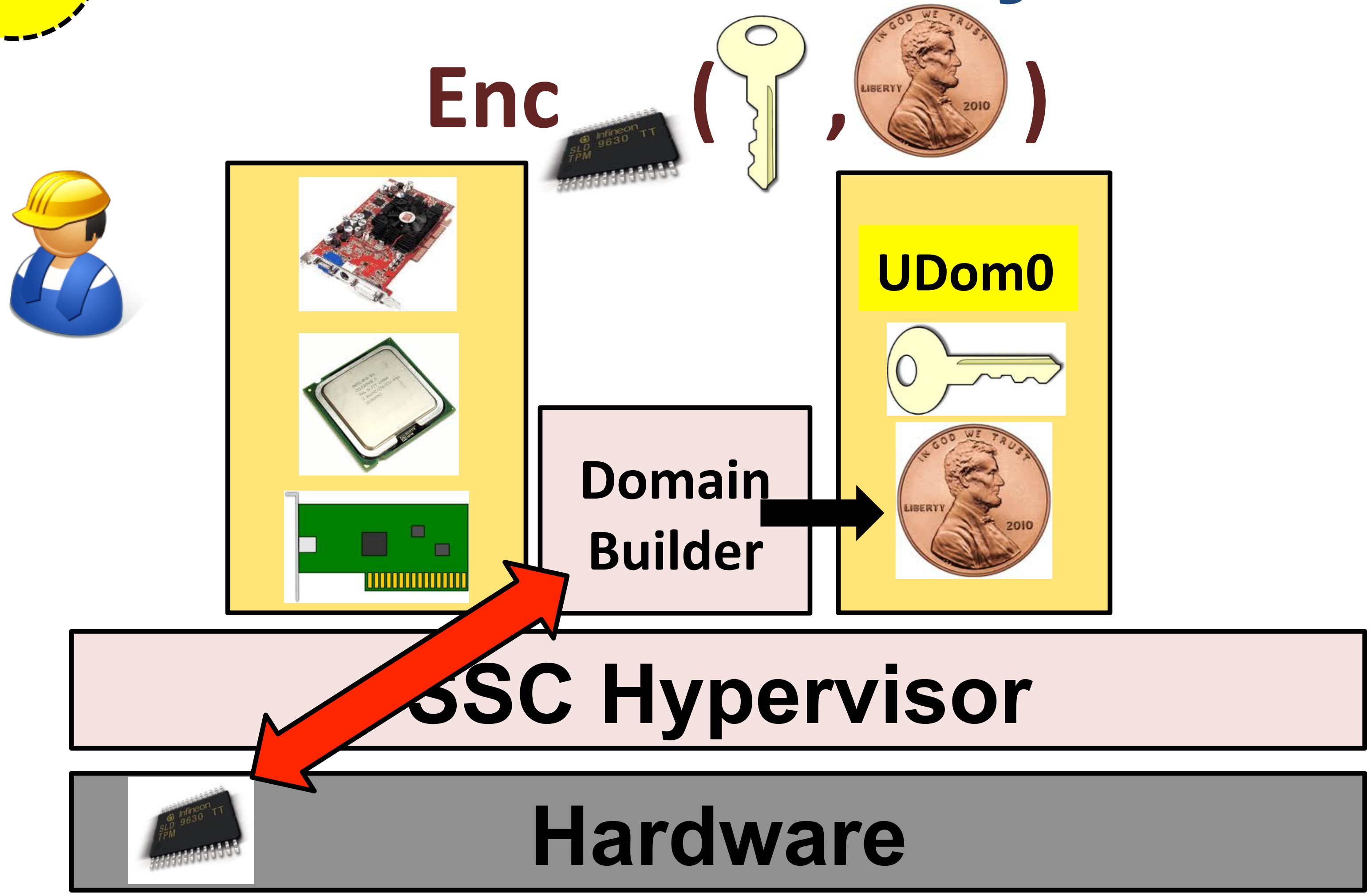
2

DomB builds domain



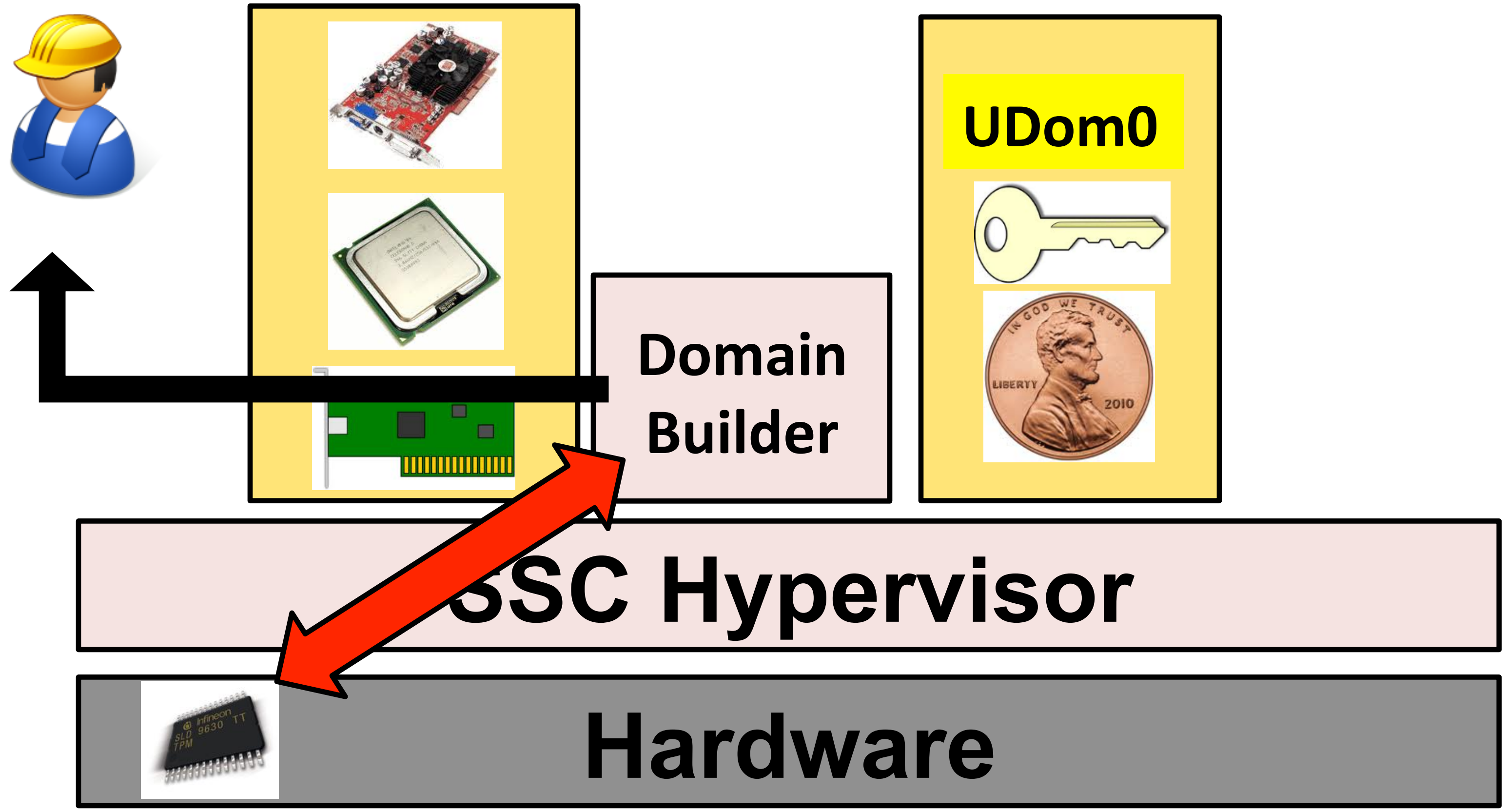
3

DomB installs key, nonce

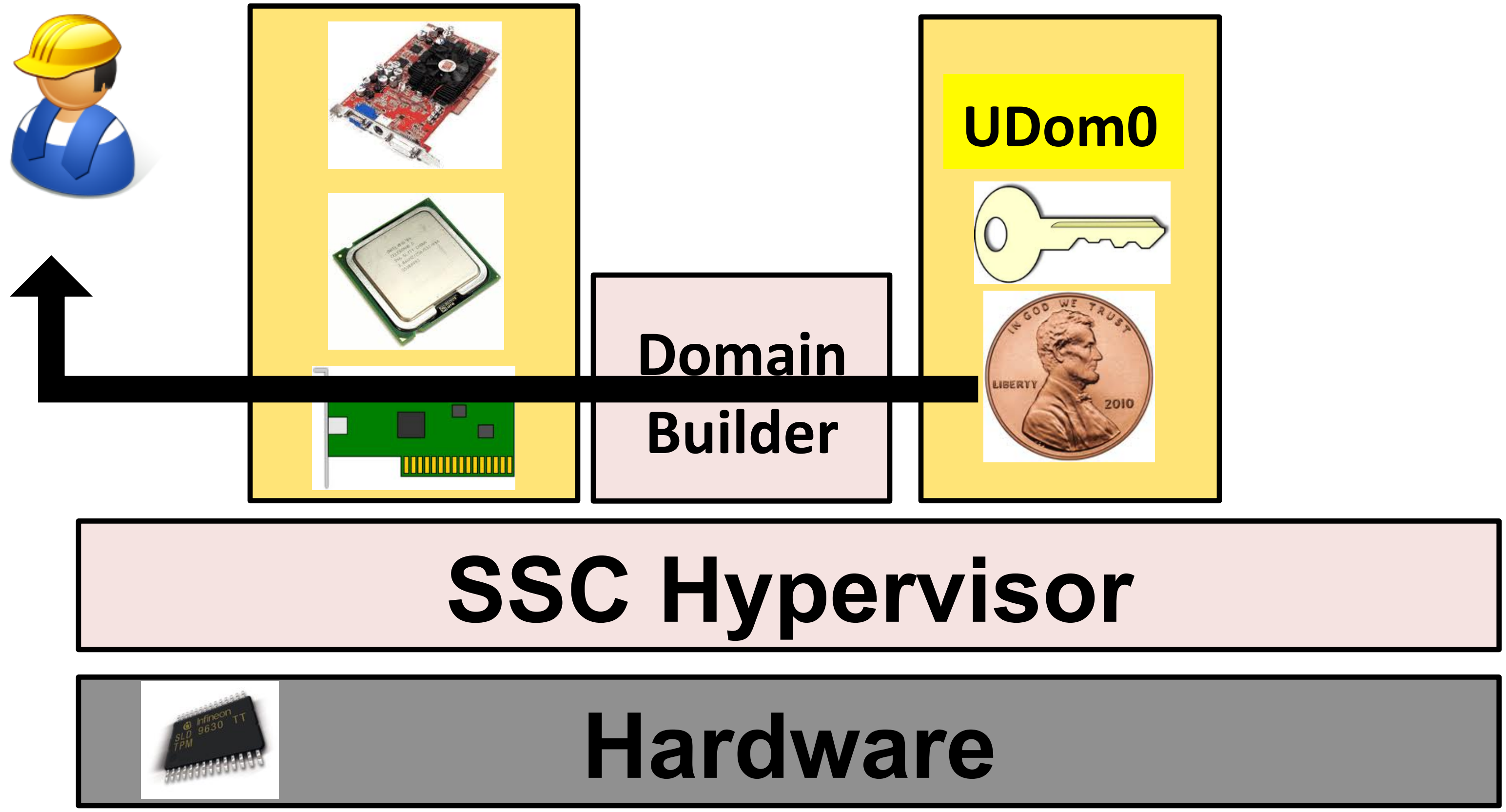


4

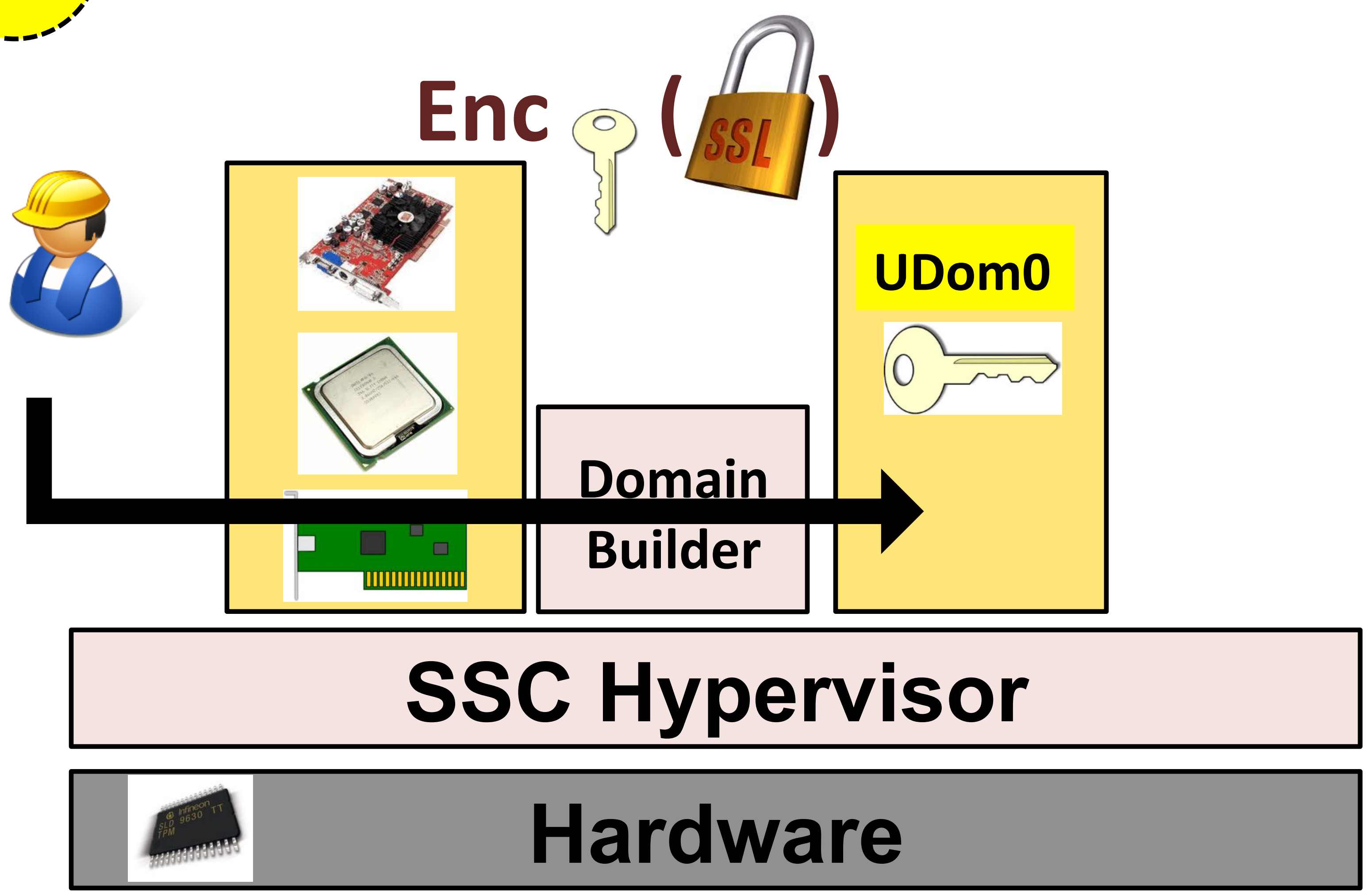
Client gets TPM hashes



5 Udom0 sends to client

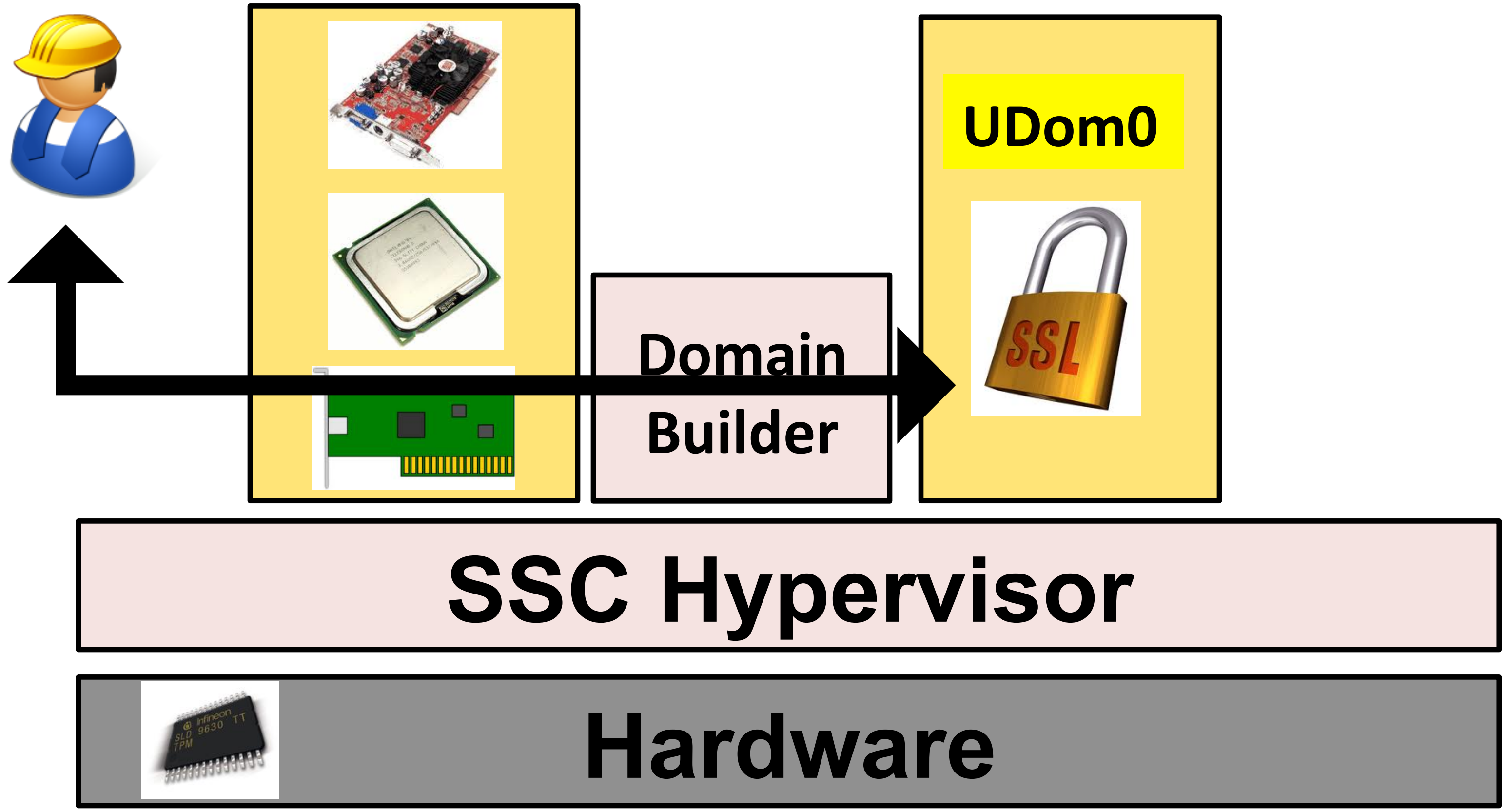


6 Client sends Udom0 SSL key

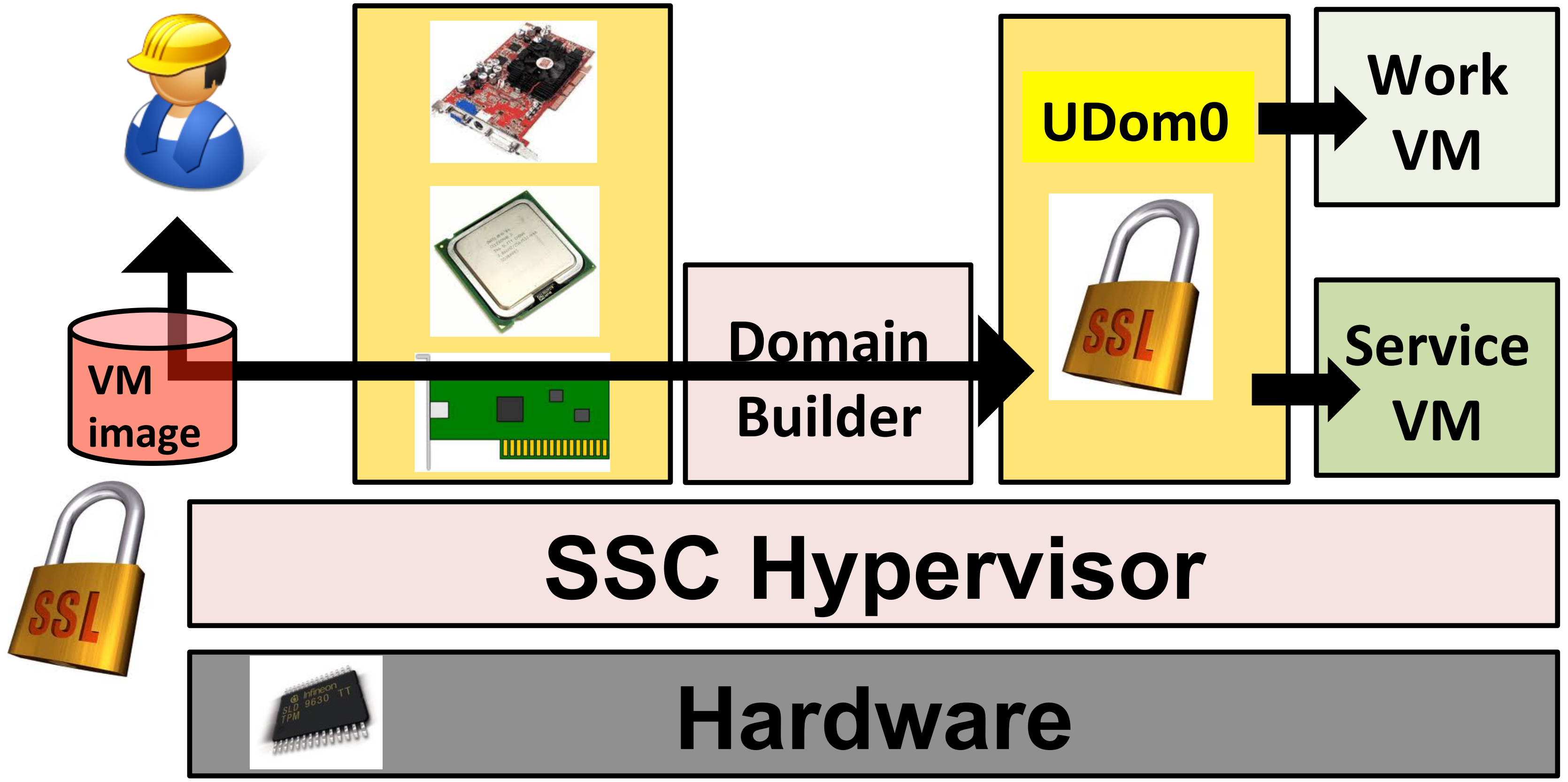


7

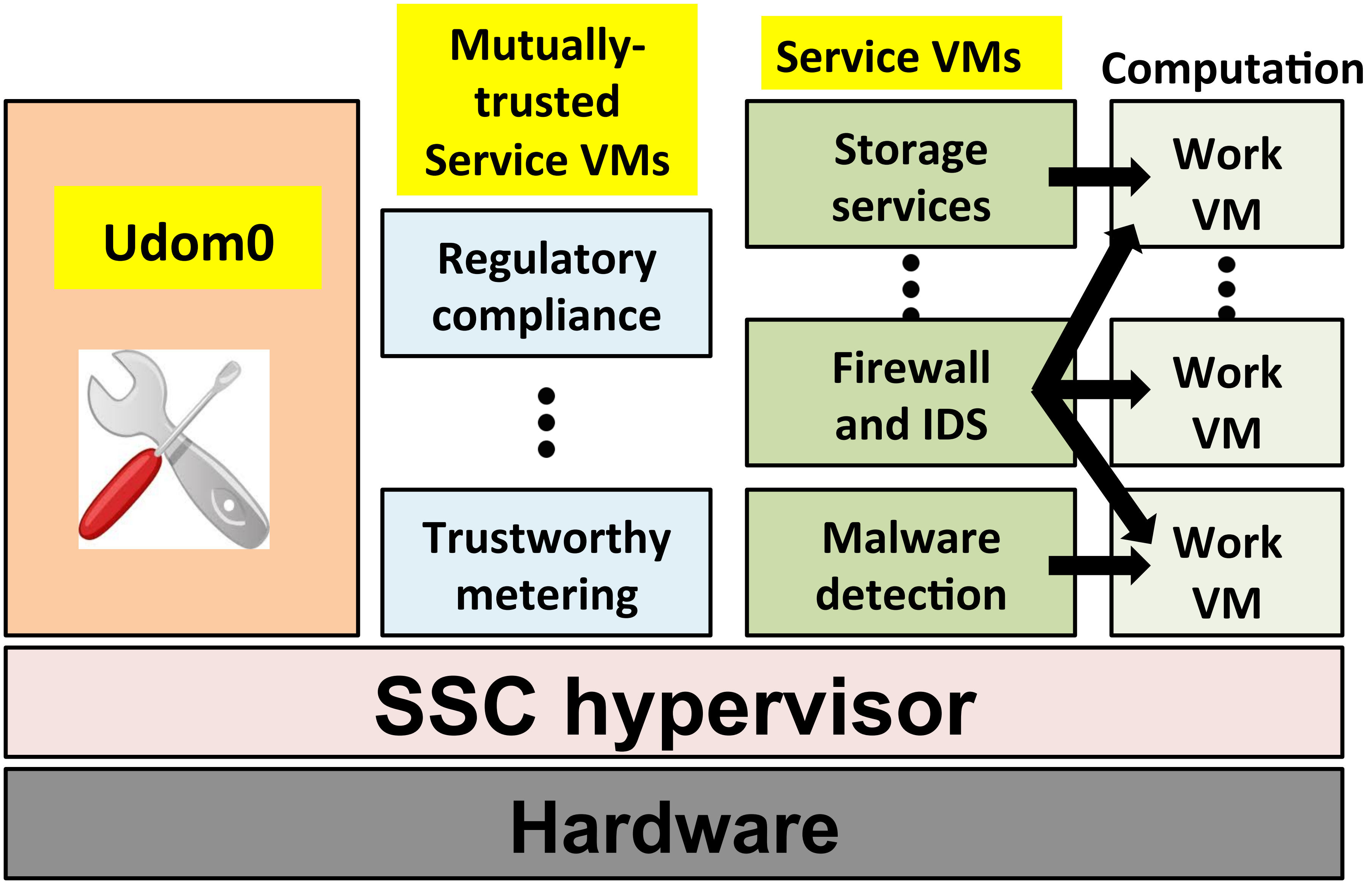
SSL handshake and secure channel establishment



8 Can boot other VMs securely



Client meta-domains



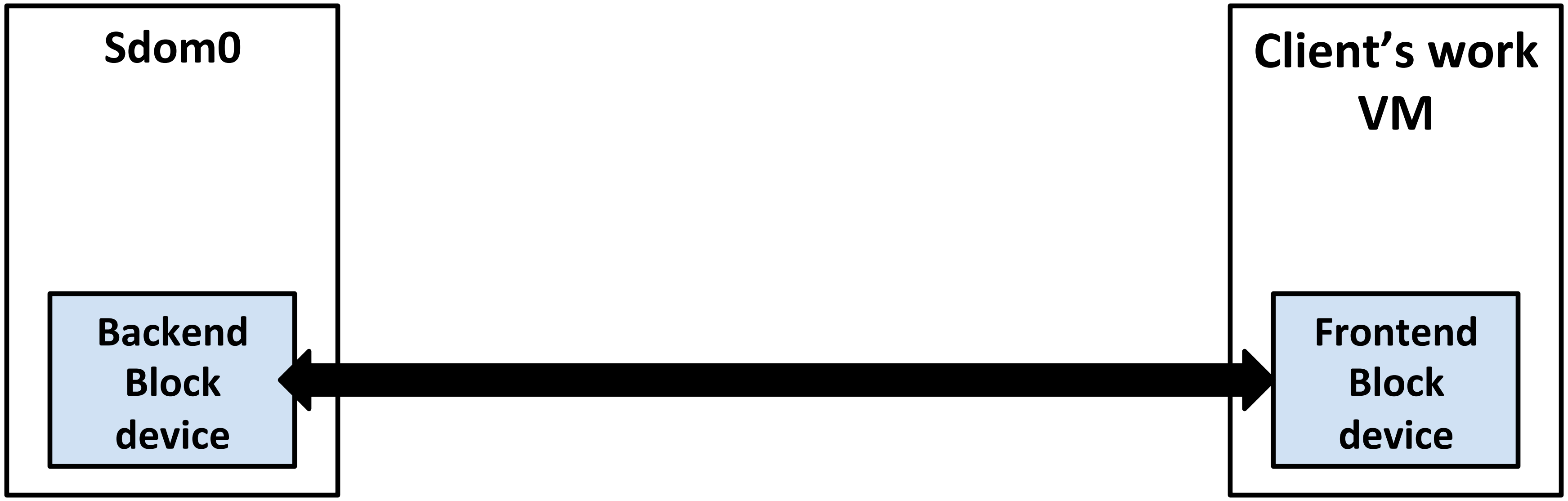
Case studies: Service VMs

- Storage services: Encryption, Intrusion detection
- Security services:
  - Kernel-level rootkit detection
  - System-call-based intrusion detection
- Data anonymization service
- Checkpointing service
- Memory deduplication
- **And compositions of these!**

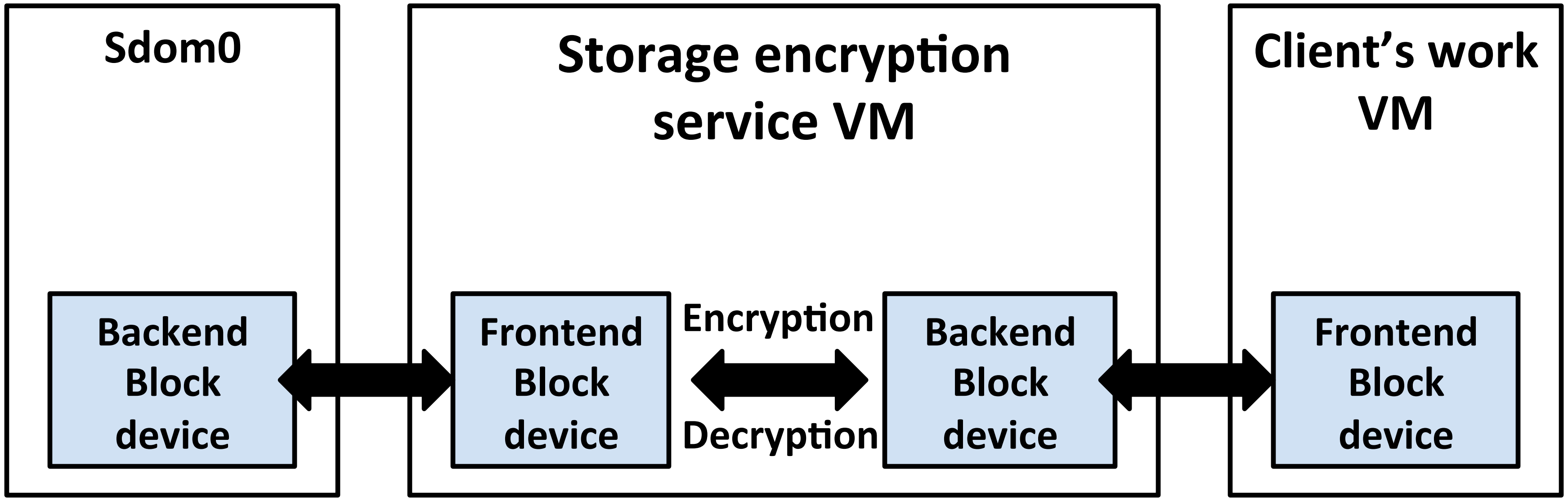
Evaluation

- Goals
  - Measure overhead of SSC
- Dell PowerEdge R610
  - 24 GB RAM
  - 8 XEON cores with dual threads (2.3 GHz)
  - Each VM has 2 vCPUs and 2 GB RAM
- Results shown only for 2 service VMs
  - Our ACM CCS’12 paper presents many more

Storage encryption service VM

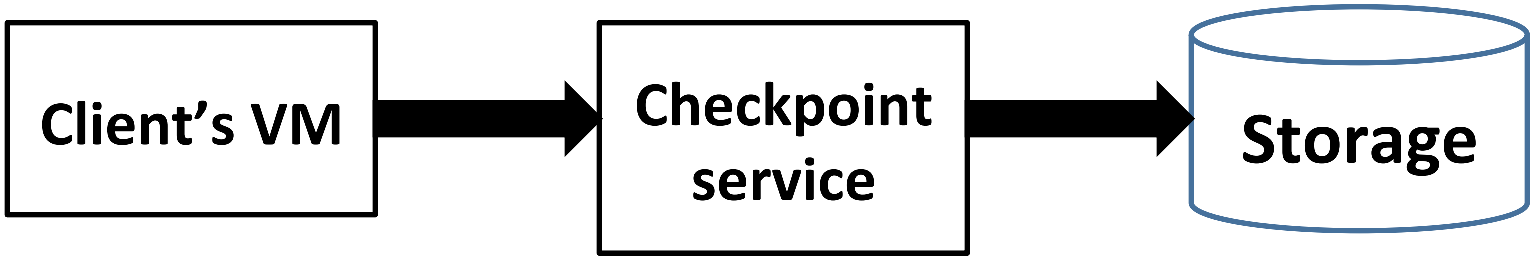


Storage encryption service VM

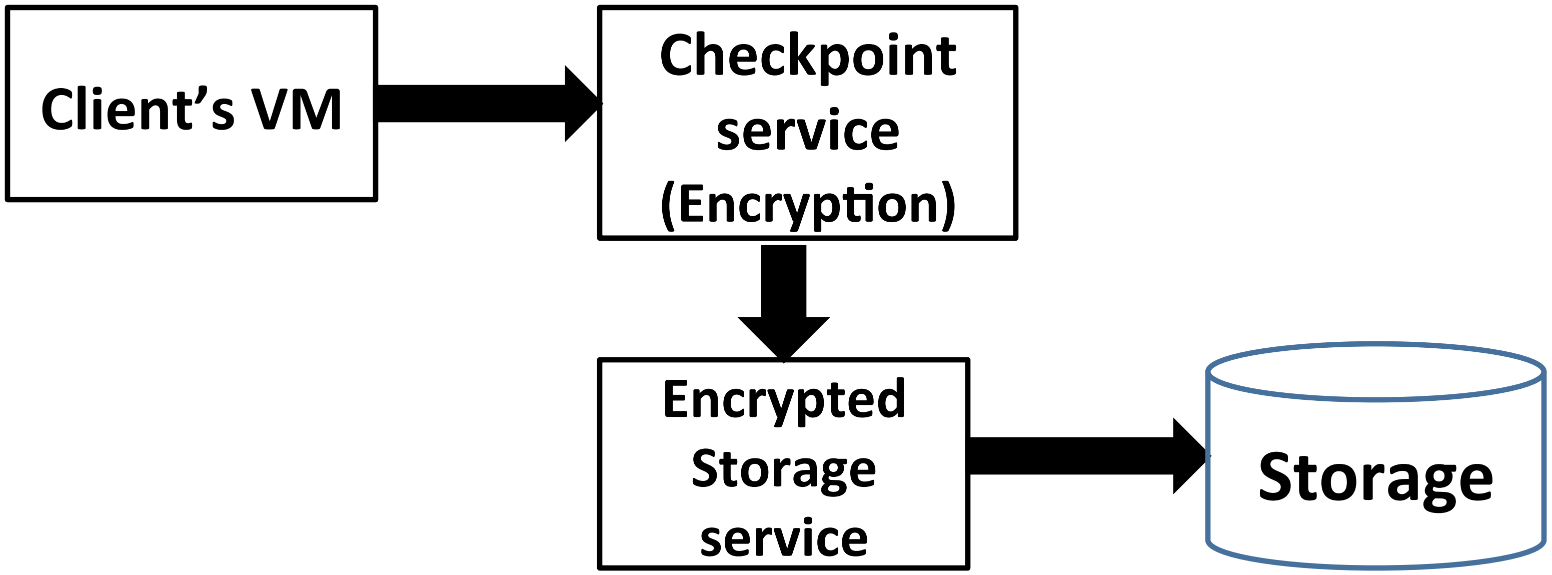


Platform	Unencrypted (MB/s)	Encrypted (MB/s)
Xen-legacy	81.72	71.90
Self-service	75.88	70.64

Checkpointing service VM



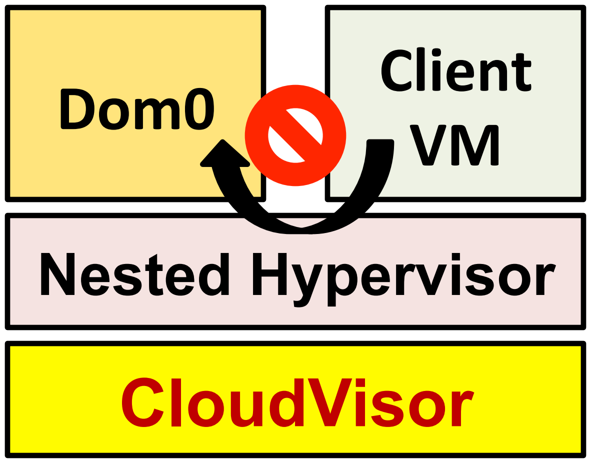
Checkpointing service VM

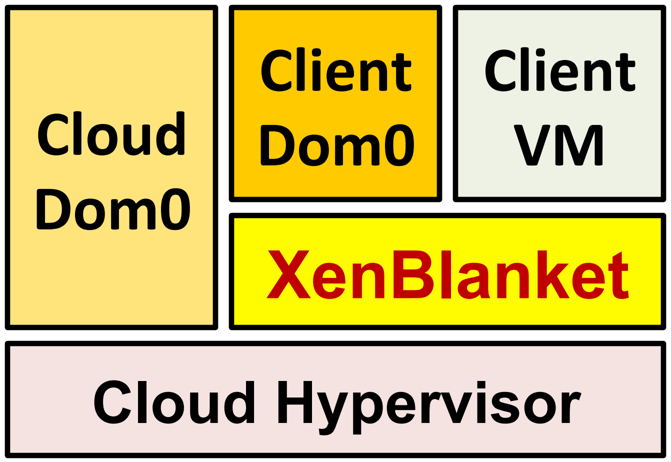


Platform	Unencrypted (sec)	Encrypted (sec)
Xen-legacy	1.840	11.419
Self-service	1.936	11.329

Related projects

CloudVisor [SOSP'11]	Xen-Blanket [EuroSys'12]
Protect client VM data from Dom0 using a thin, bare-metal hypervisor	Allow clients to have their own Dom0s on commodity clouds using a thin shim





SSC is a cloud model that …

… Improves security and privacy of client code and data

… Enhances client control over their VMs

… Imposes low runtime performance overheads

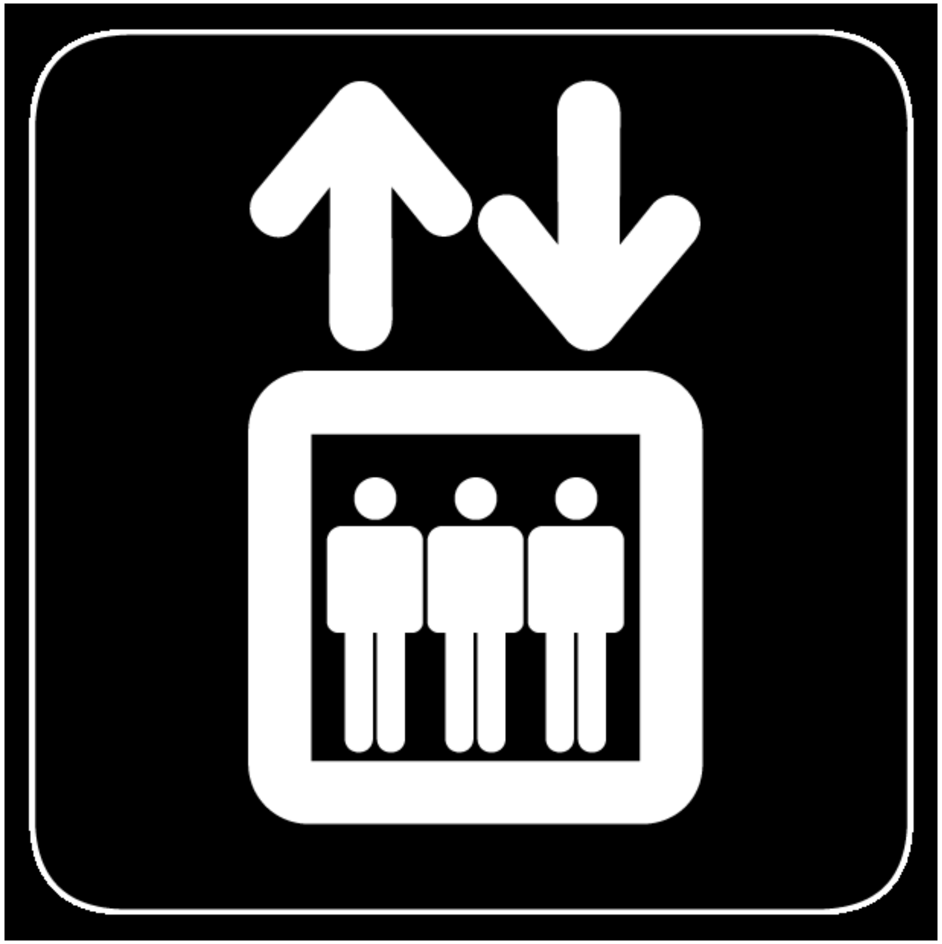
… Provides a rich source of problems for followup work ☺

Future vision for SSC

- **Cloud app** markets:
  - Marketplaces of service VMs.
  - Research problems: Ensuring trustworthiness of apps, enabling novel mutually-trusted apps, App permission models.

- **Migration**-awareness:
  - Policies and mechanisms for VM migration in SSC.
  - Research problems: Prevent exposure of cloud infrastructure details to competitors, TPM-based protocols that are migration-aware.

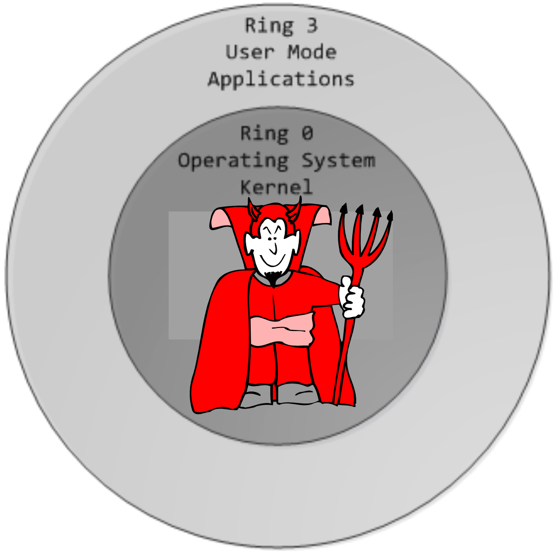
Other research projects



The Cloud The browser The smartphone

Smartphone rootkits

New techniques to detect OS kernel-level malware



- Rootkits operate by maliciously modifying kernel code and data

RESULTS:

- New techniques to detect data-oriented rootkits **[ACSAC'08]**
- Exploring the rootkit threat on smartphones **[HotMobile'10]**
- Security versus energy tradeoffs in detecting rootkits on mobile devices **[MobiSys'11]**

Securing Web browsers

Studying information leakage via 3rd party browser addons

- Addons are untrusted, privileged code
  - All major browsers support addons
  - Can leak sensitive information

RESULTS:

- Information flow tracking-enhanced browser **[ACSAC'09]**
- Static capability leak analysis for Mozilla Jetpack **[ECOOP'12]**
- New bugs found in Mozilla extensions

And many more …

- **The Cloud** (and other software systems)
  [CCS08, ACSAC08a, ACSAC09a, RAID10, TDSC11, CCS12a, CCS12b, ANCS12]
  – Security remediation using transactional programming
  – Fast, memory-efficient network intrusion detection
- **The browser** (and the Web)
  [ACSAC08b, ACSAC09b, ECOOP12a, ECOOP12b]
  – Secure mashup Web applications
  – Integrating the Web and the cloud
  – Isolation as a first-class JavaScript feature
- **The smartphone** (and other mobile devices)
  [UbiComp09, SACMAT09, HotMobile10, MobiSys11]
  – Location privacy in mobile computing
  – Secure remote access to enterprise file systems

Looking into the future…

Active ongoing projects

SSC++

Improving browser extension security

Improving mobile app security

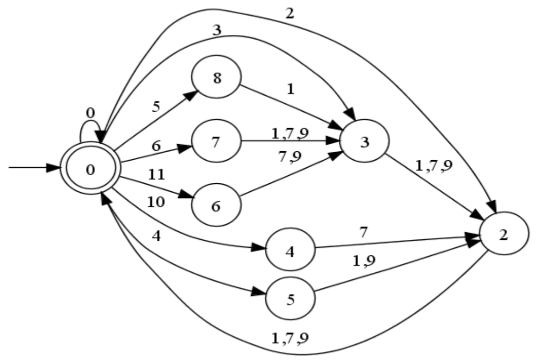
Collaborators and students

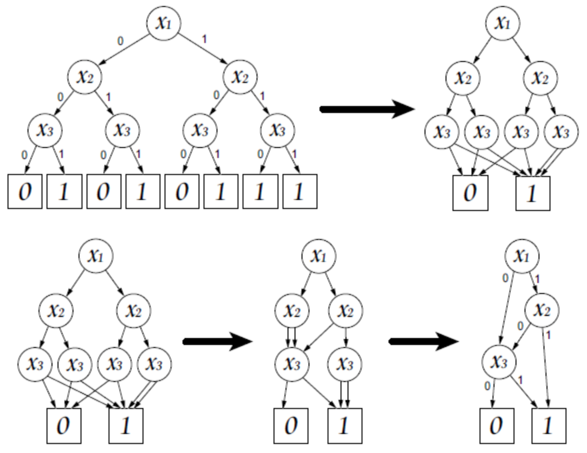
And many other camera-shy folks!

"That's all Folks!"

Fast and memory-efficient NIDS

Using ordered binary decision diagrams (OBDDs) to address time/space tradeoff in regexp matching





- Regexp matching a basic primitive in many NIDS and firewalls
- Fundamental time/space tradeoff:
  - DFAs are fast but memory intensive
  - NFAs are memory efficient but slow

<u>MAIN RESULT</u>:

- Encoding NFAs using OBDDs
- Obtains NFA-like memory consumption with DFA-like speed **[RAID'10, COMNET'11, ANCS'12]**

Transactional introspection

Security using transactional programming and machinery

```
dispatch_request ( ) {
    transaction [ principal ] {
        ...
        perform_request ( );
        ...
    }   /* Commits only if all authorization succeeds */
}
```

BENEFIT:
Security remediation for **free**

- Enforcing authorization policies with stronger guarantees **[CCS'08]**
- Detecting data structure corruptions **[RV'11]**
- Sandboxing untrusted JavaScript code using transactions **[ECOOP'12]**