*Reflections on*

# Self-service Cloud Computing

**Vinod Ganapathy**

vinodg@cs.rutgers.edu

Associate Professor of Computer Science

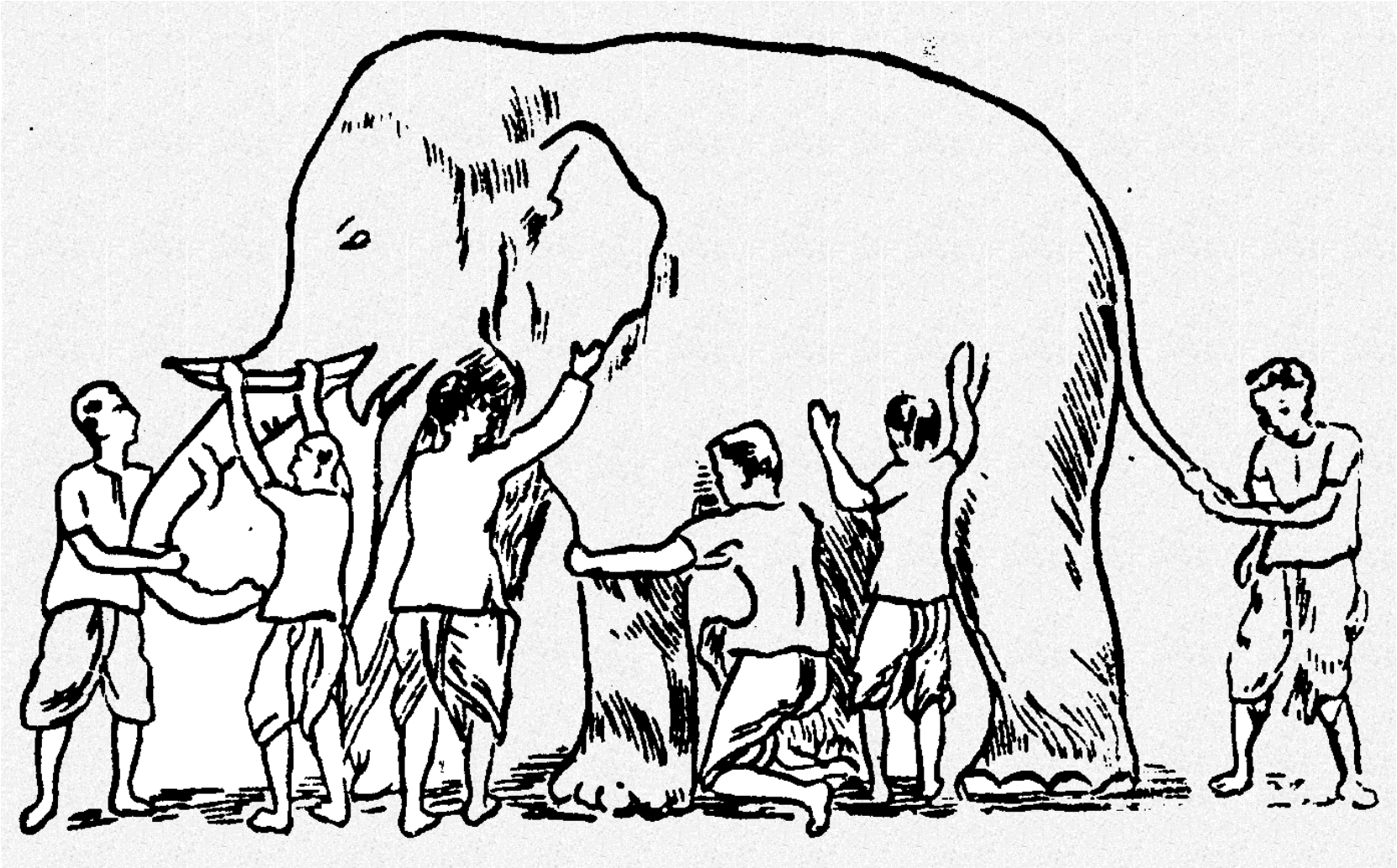
Rutgers, The State University of New Jersey

# My group's research

**Computer Security and Software Engineering**

- **Cloud platform security:**
  - "How can I entrust the cloud with my code and data?"
  - "Is the cloud provider is billing me correctly?"
- **Web browsers and apps:**
  - "How do I ensure the privacy of my browsing activity?"
  - "Can I trust the new browser app I just downloaded?"
- **Smart devices and apps:**
  - "How do I know that my phone is secure?"
  - "How do I create apps that work across diverse platforms like the iPhone, Android, Windows, etc.?"

# The Cloud



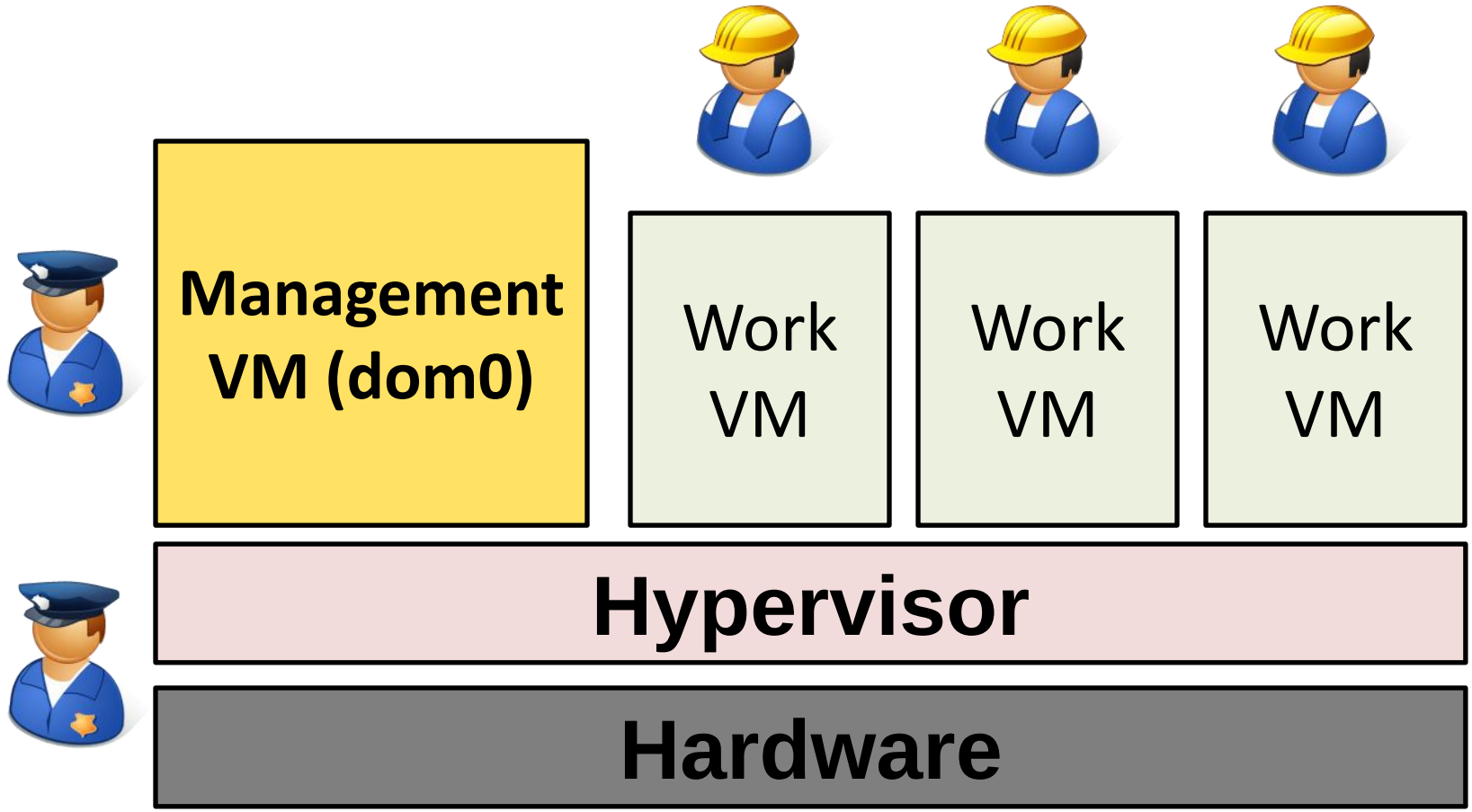
# The Cloud

**A distributed computing infrastructure, managed by third parties, with which we entrust our code and data**

- Comes in many flavours: ***-aaS**
  - Infrastructure|Platform|Database|Storage|…
- Many economic benefits
  - By 2015, 90% of government agencies and large companies will use the cloud **[Gartner'12]**
  - Many new services rely exclusively on the cloud, *e.g.,* Instagram, MIT/Harvard EdX
- **Public** versus **private** cloud infrastructures


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

# Virtualized cloud platforms




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

# Problem #1

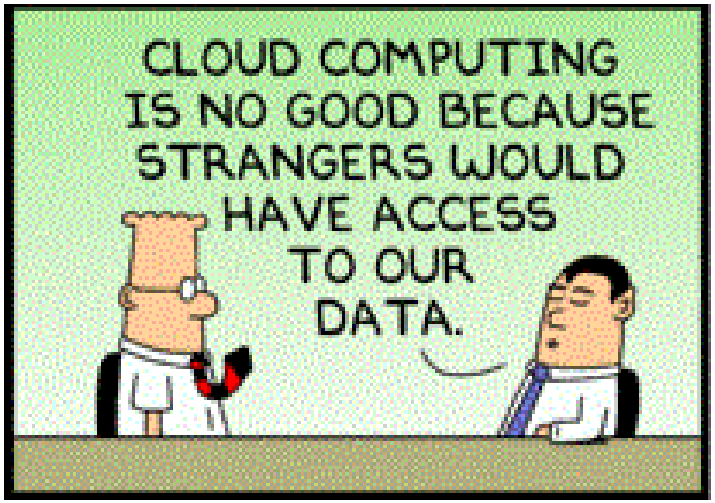
**Client code & data secrecy and integrity vulnerable to attack**

**Trust me with your code & data**

**Client** **Provider**

**You have to trust us as well**

**Cloud Administrators**




- Data breaches on the cloud a common occurrence:
  - Microsoft: Spying on employee's Hotmail account
  - Google employee: Spying on children's data
  - NSA Snowden data leaks
- Enterprises like banks and finance companies prefer to use in-house cloud offerings rather than opting for public cloud platforms

# Problem #2

**Clients must rely on provider to deploy customized services**

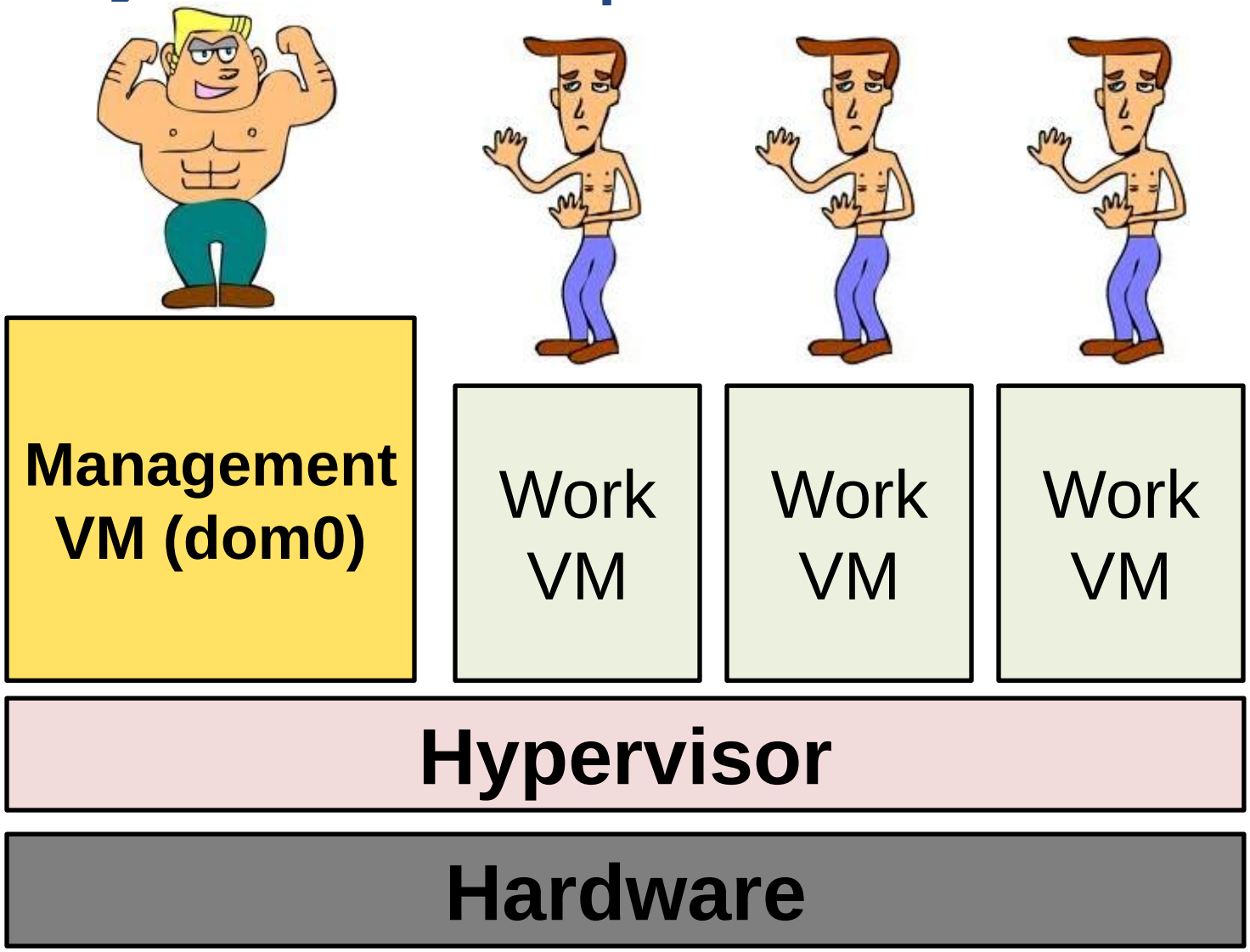
**I need customized malware detection and VM rollback**

**Client** **Provider**

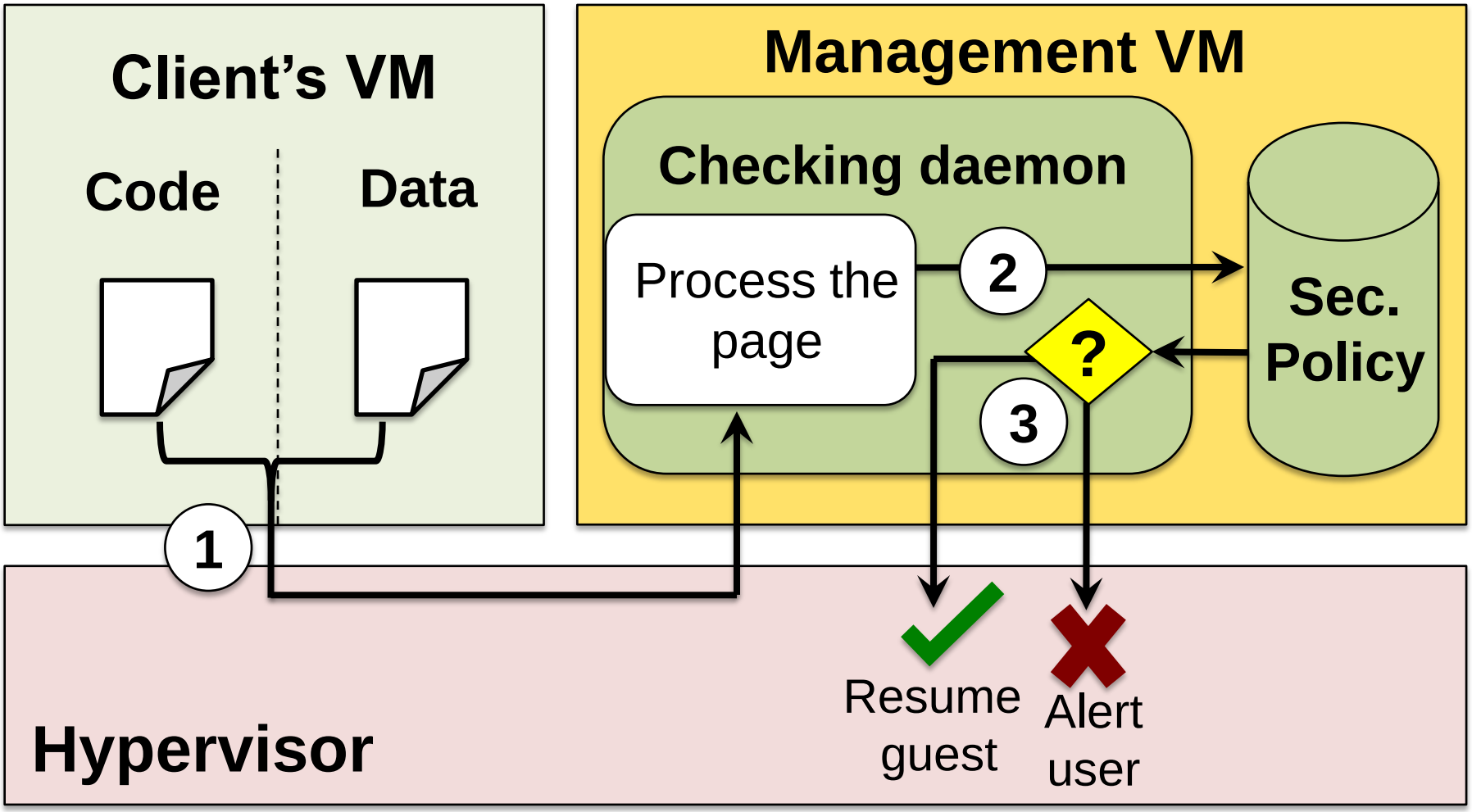
**For now just have checkpointing …**

**Client** **Provider**

# Why do these problems arise?



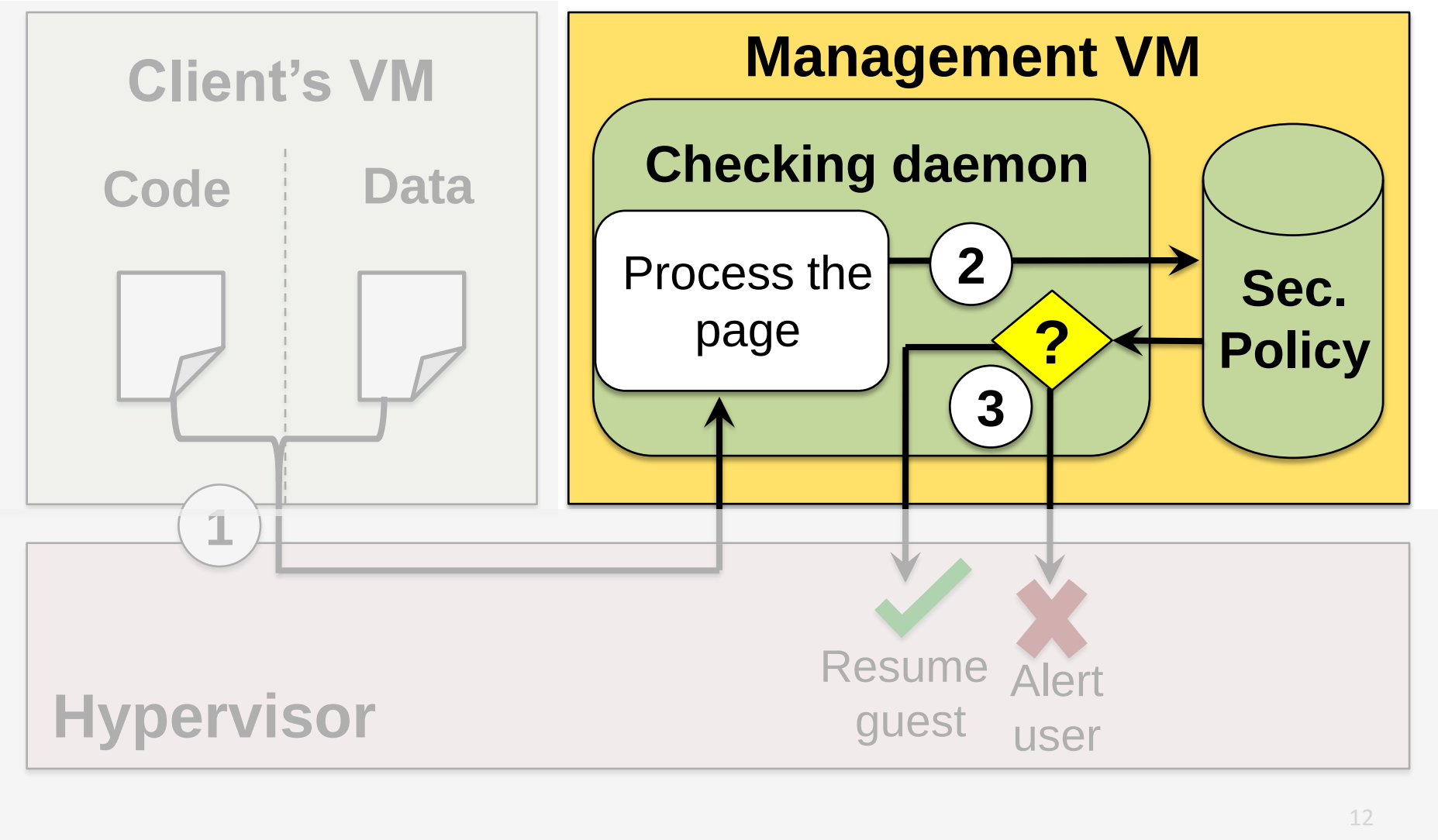
# Example: Malware detection




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

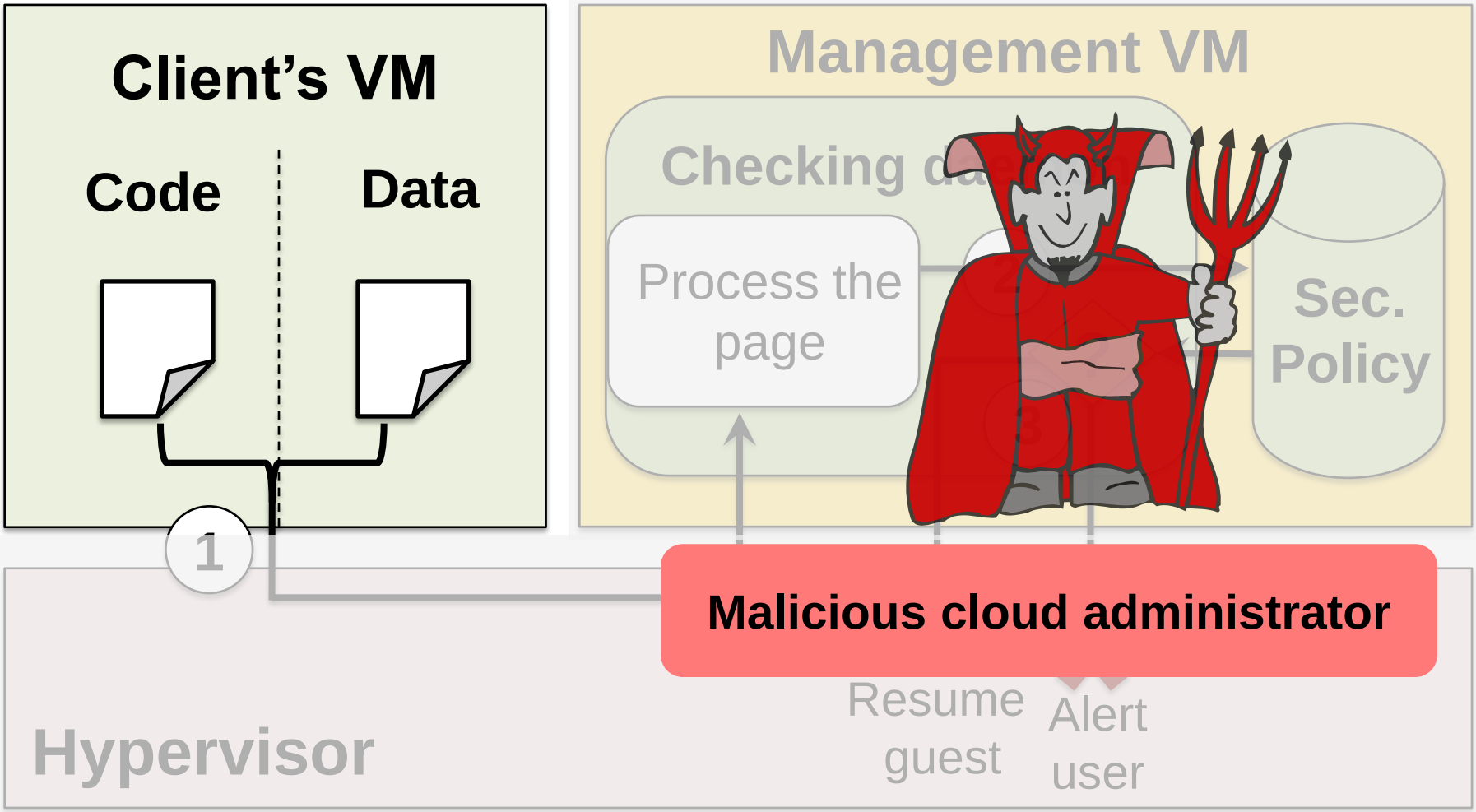
# Problem

**Clients must rely on provider to deploy customized services**



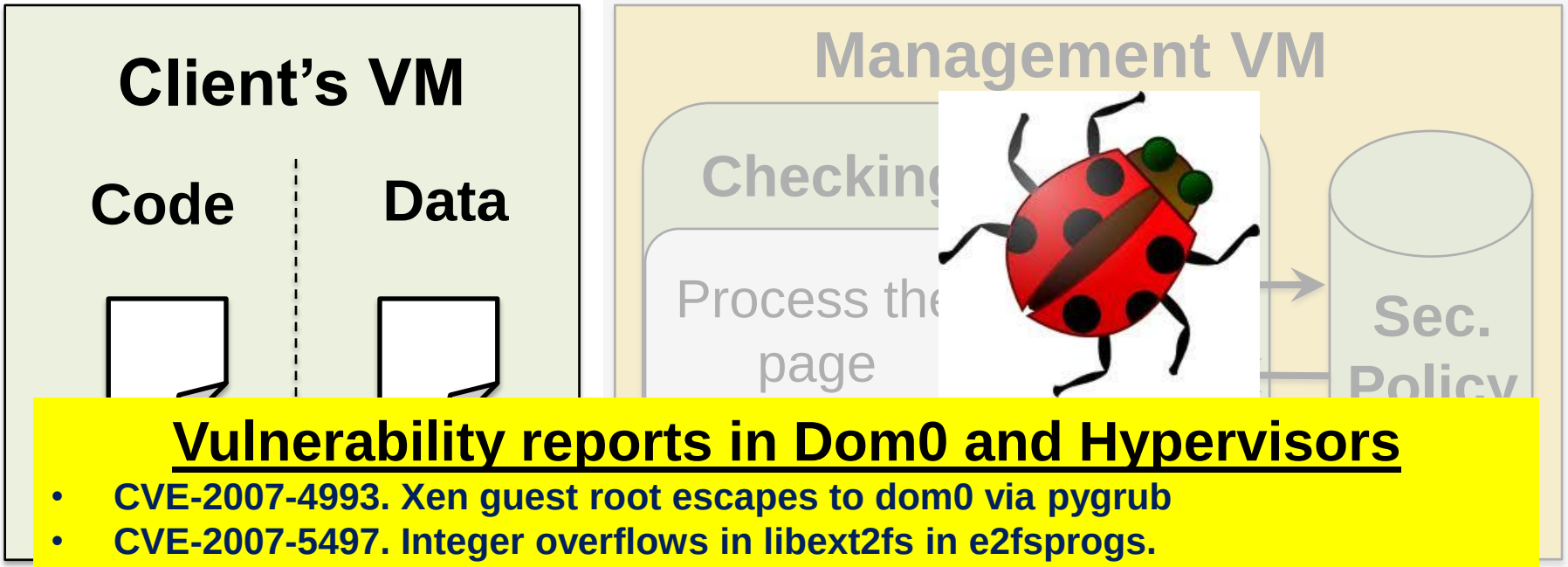
# Problem

**Client code & data secrecy and integrity vulnerable to attack**



# Problem

**Client code & data secrecy and integrity vulnerable to attack**

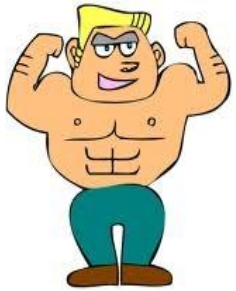



## Vulnerability reports in Dom0 and Hypervisors

- 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]

# Self-service cloud computing



**Management VM**

**Client's VMs**

**Hypervisor**

**Hardware**

# The threat model

- On the cloud, we have **providers** and **administrators:** Who to trust?

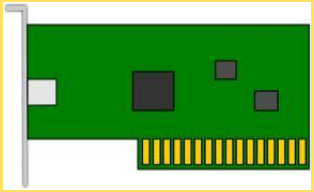
| Solution | Provider | Administrator |
|---|---|---|
| Contemporary cloud platforms |  |  |
| Cryptographic solutions/Intel SGX |  |  |
| **Self-service Cloud Computing** |  |  |

# Remainder of this talk

- Disaggregation and new privilege model
- Technical challenges:
  - Balancing provider's and client's goals
  - Secure bootstrap of client's VMs
- Experimental evaluation
- SSC versus the Intel SGX

# Duties of the management VM





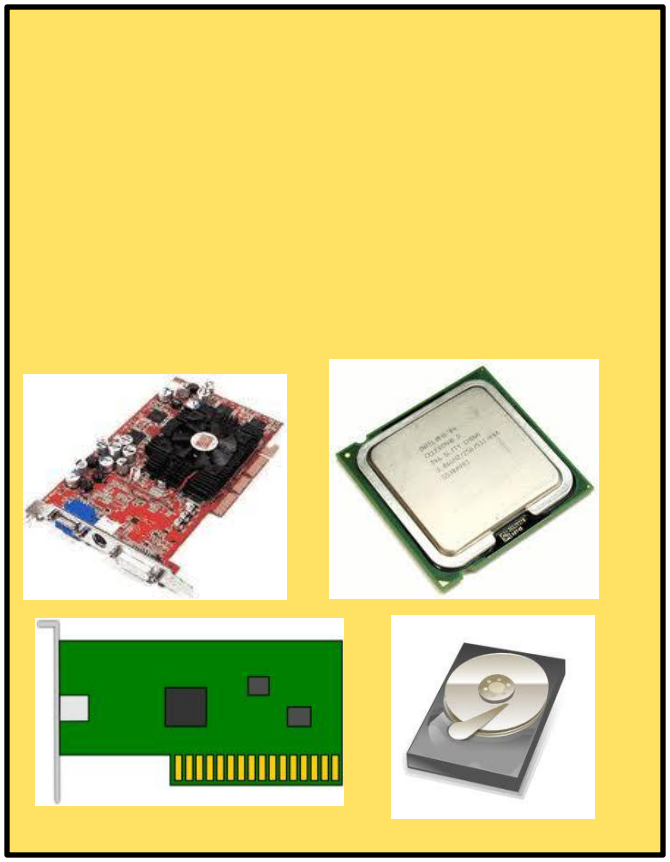

Manages and multiplexes hardware resources

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**

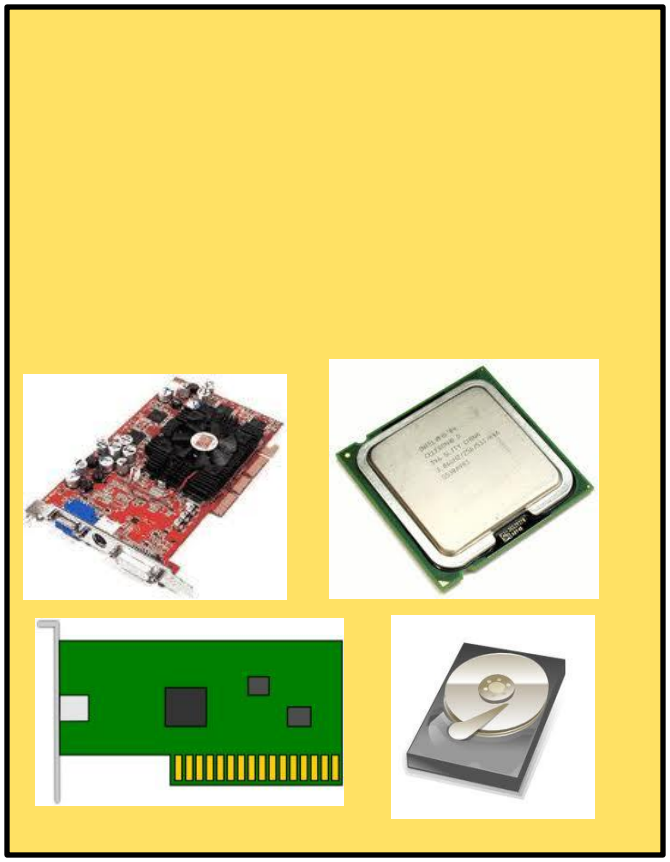
**Solves problem #2**

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

**Solves problem #1**

# Embracing first principles

**Disaggregate the management VM**



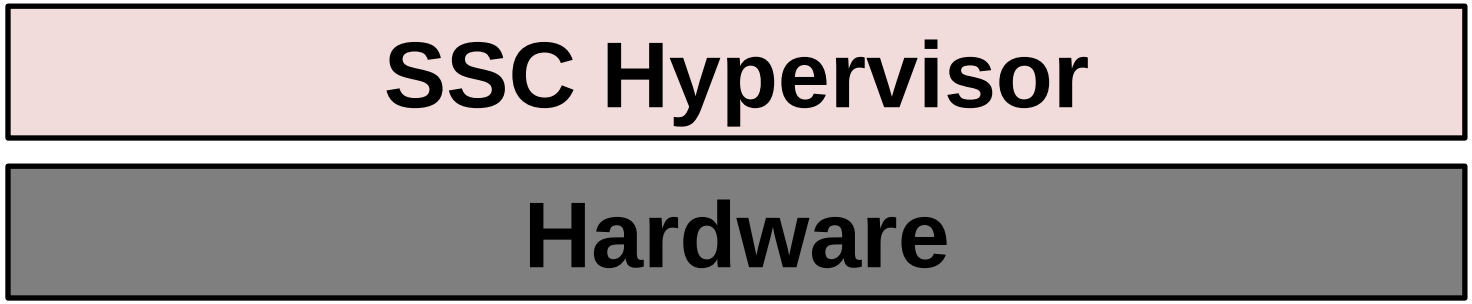
**System-wide Mgmt. VM (SDom0)**

**Per-Client Mgmt. VM (UDom0)**

**Principle of separation of privilege**

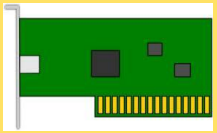
**Principle of least privilege**

# An SSC platform



# An SSC platform

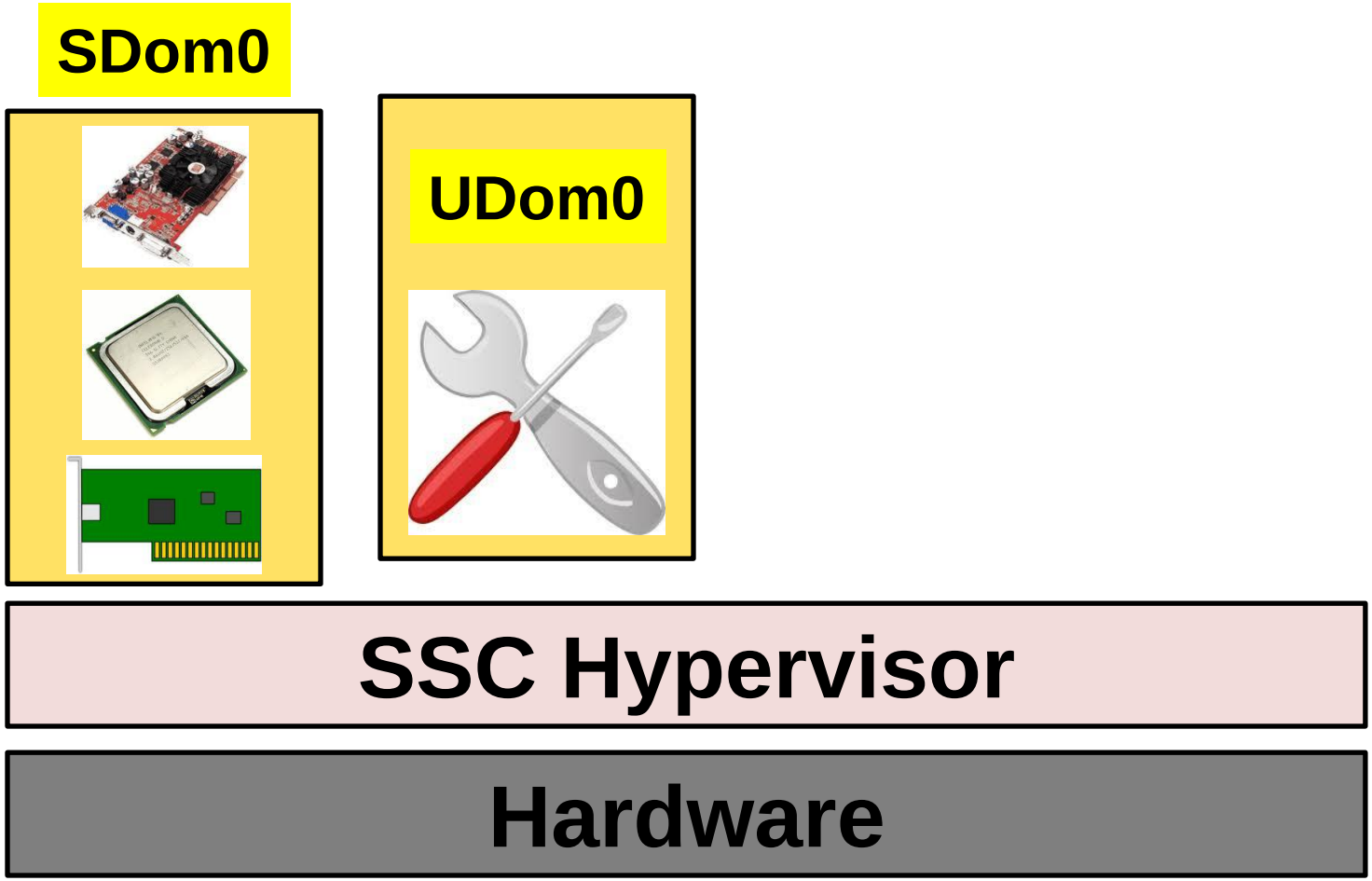
**SDom0**



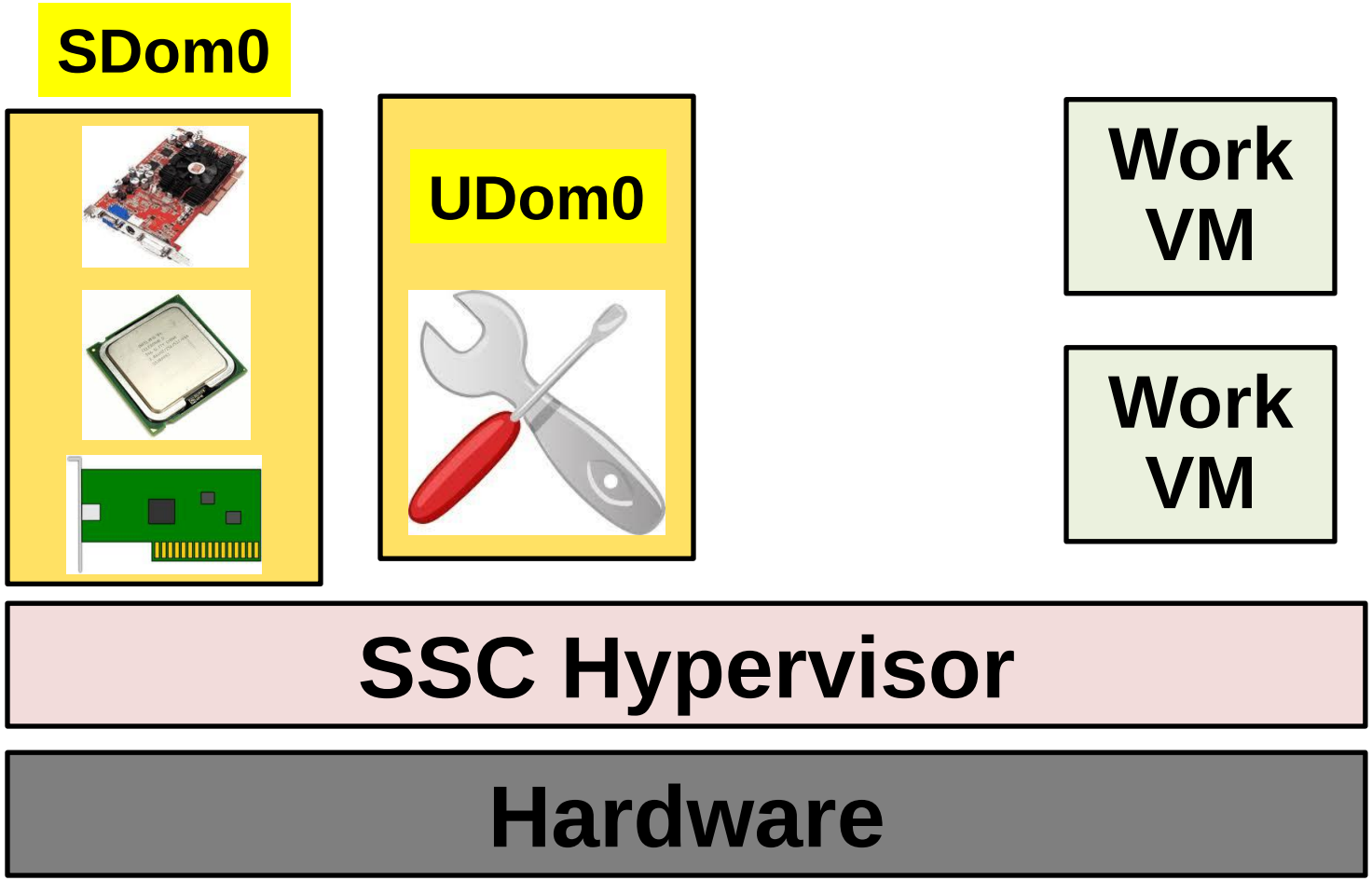
**SSC Hypervisor**

**Hardware**

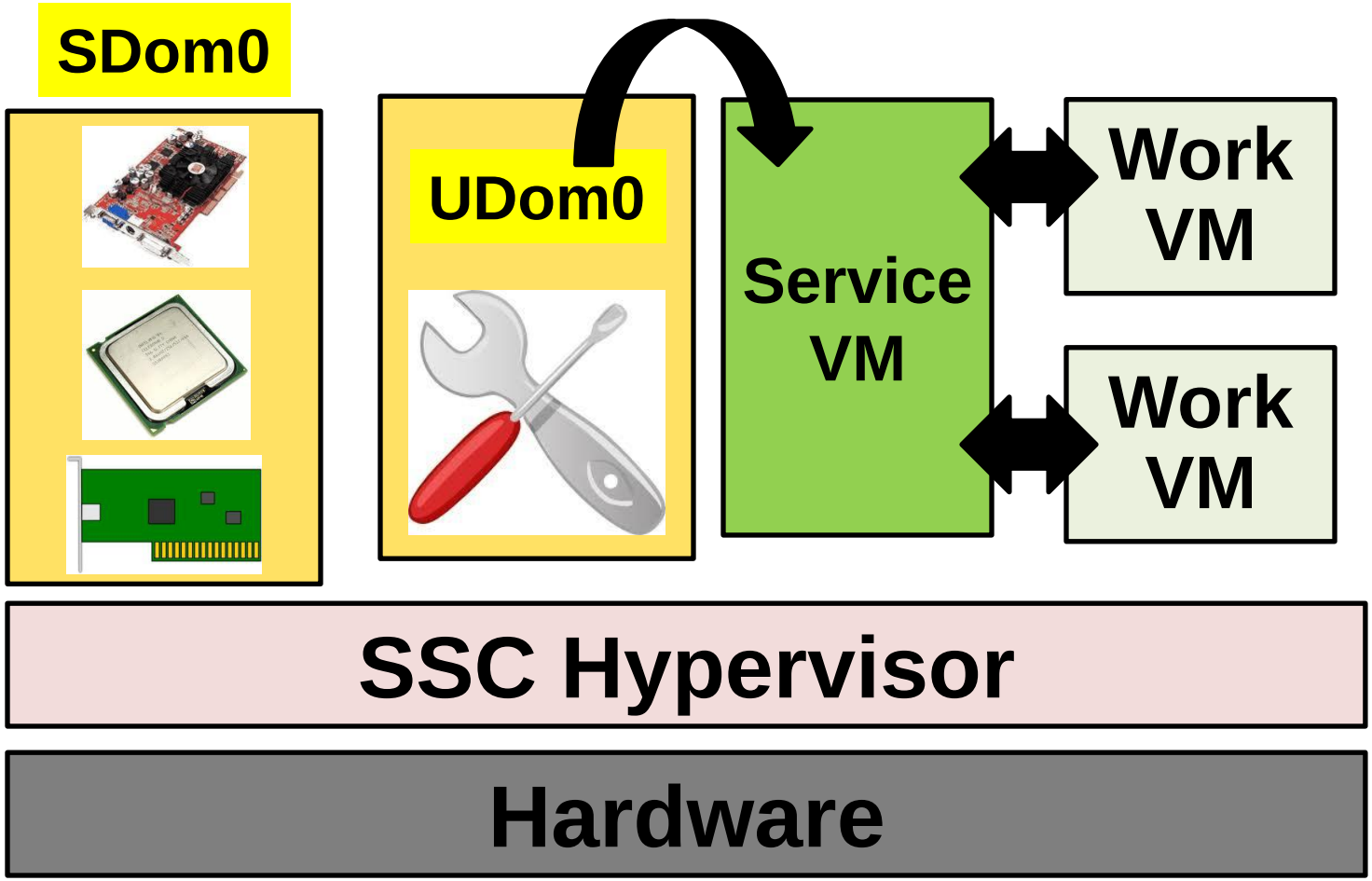
# An SSC platform



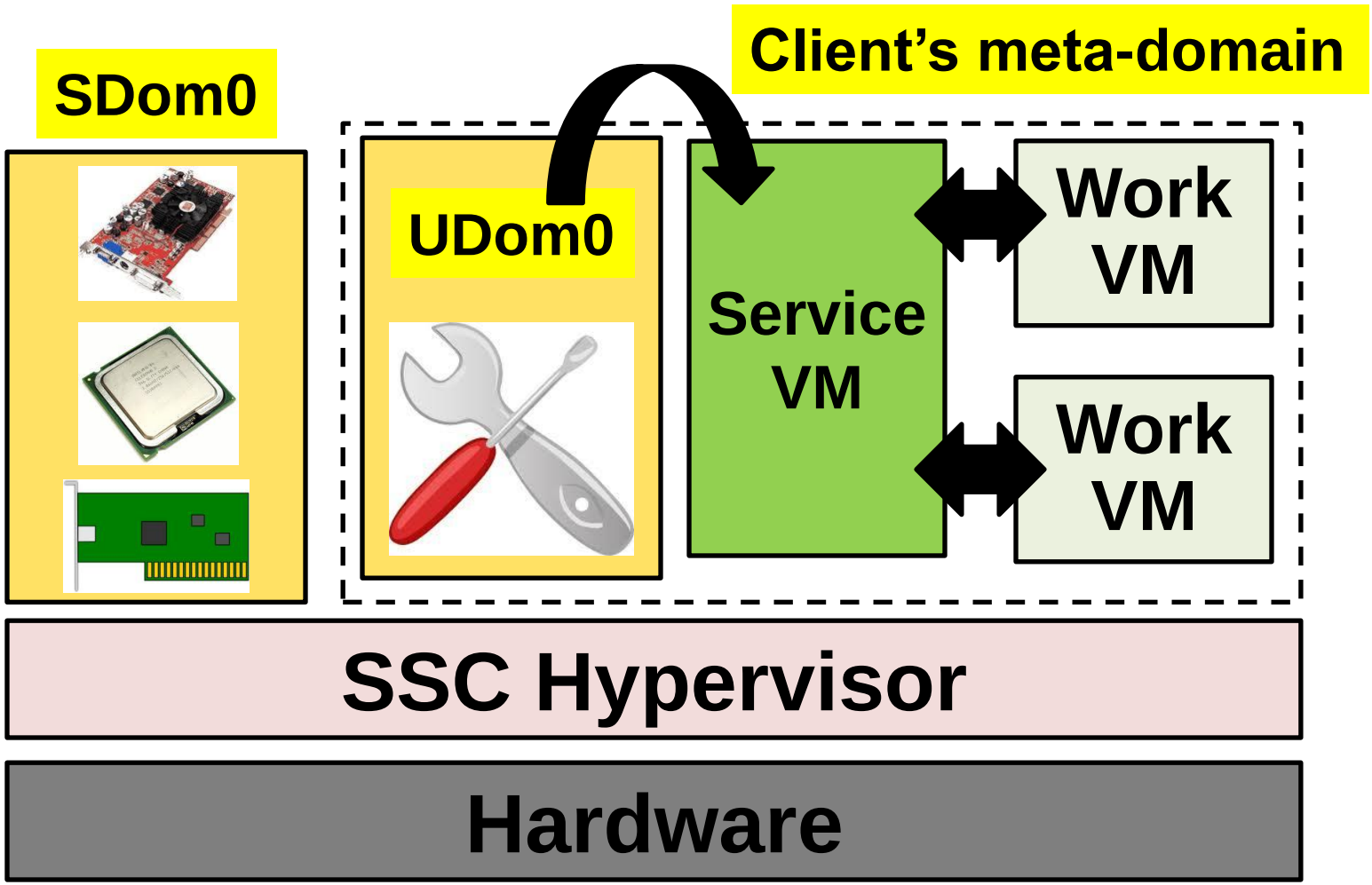
# An SSC platform



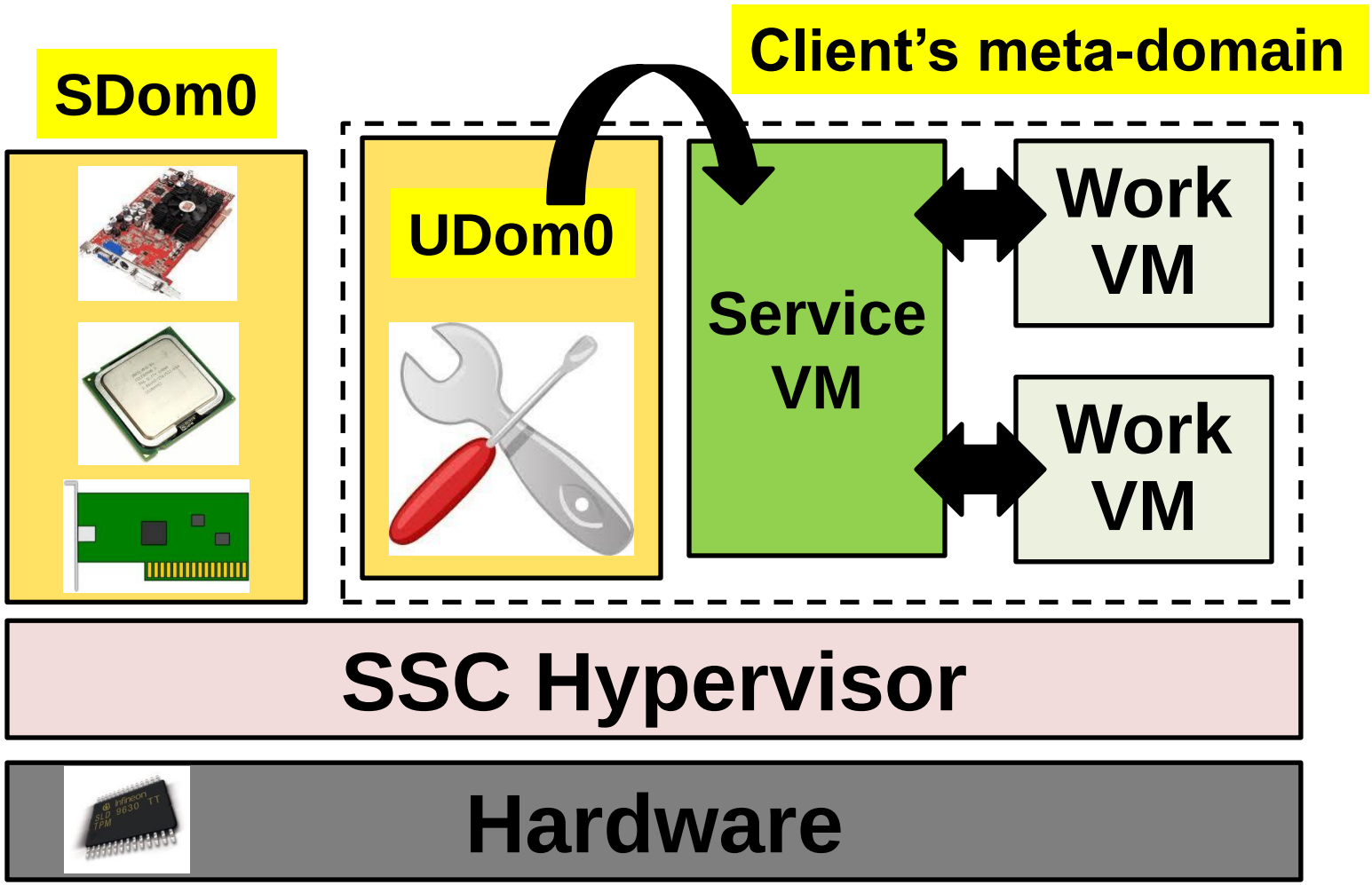
# An SSC platform



# An SSC platform

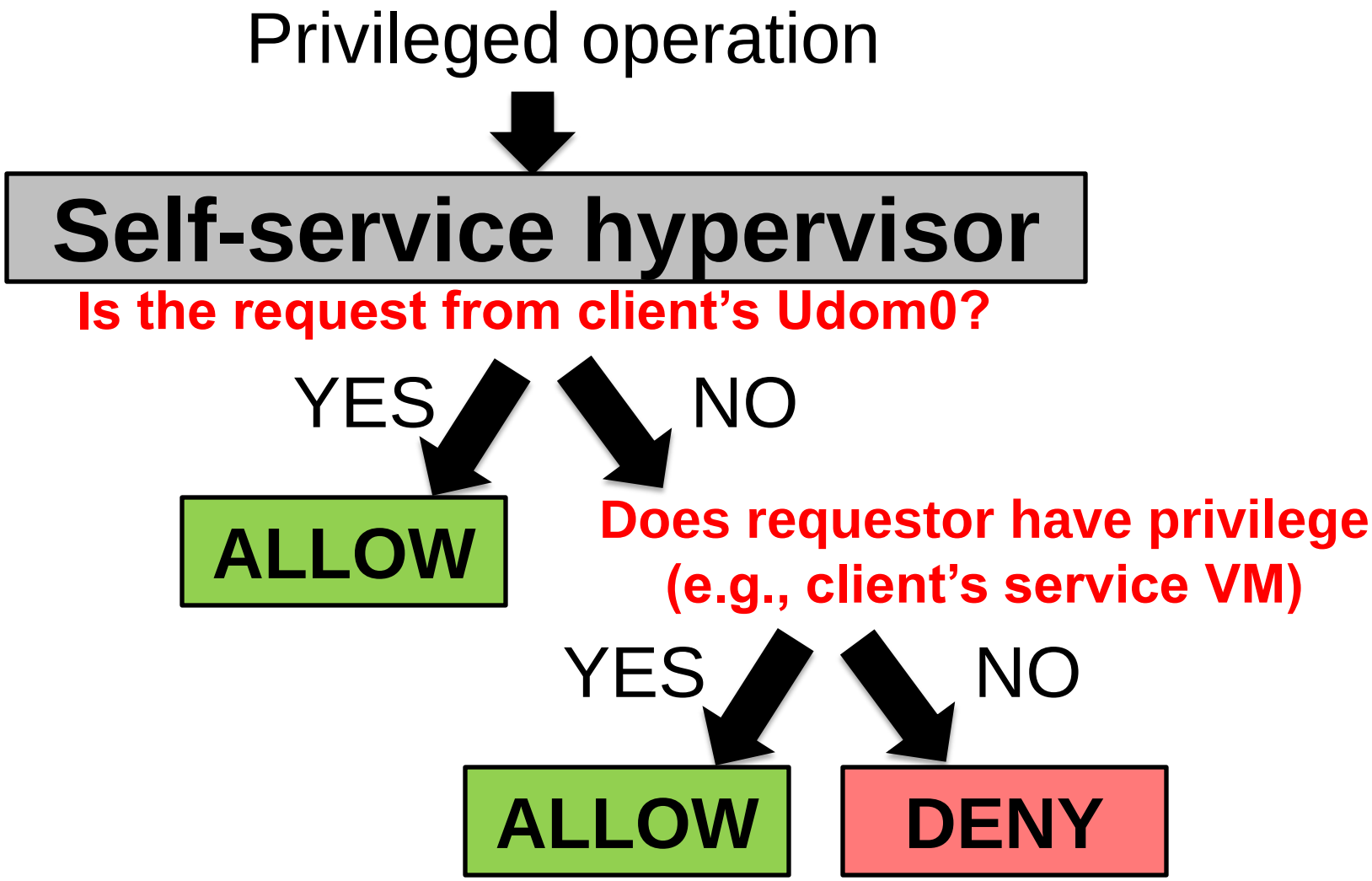


# An SSC platform




**Equipped with a Trusted Platform Module (TPM) chip**

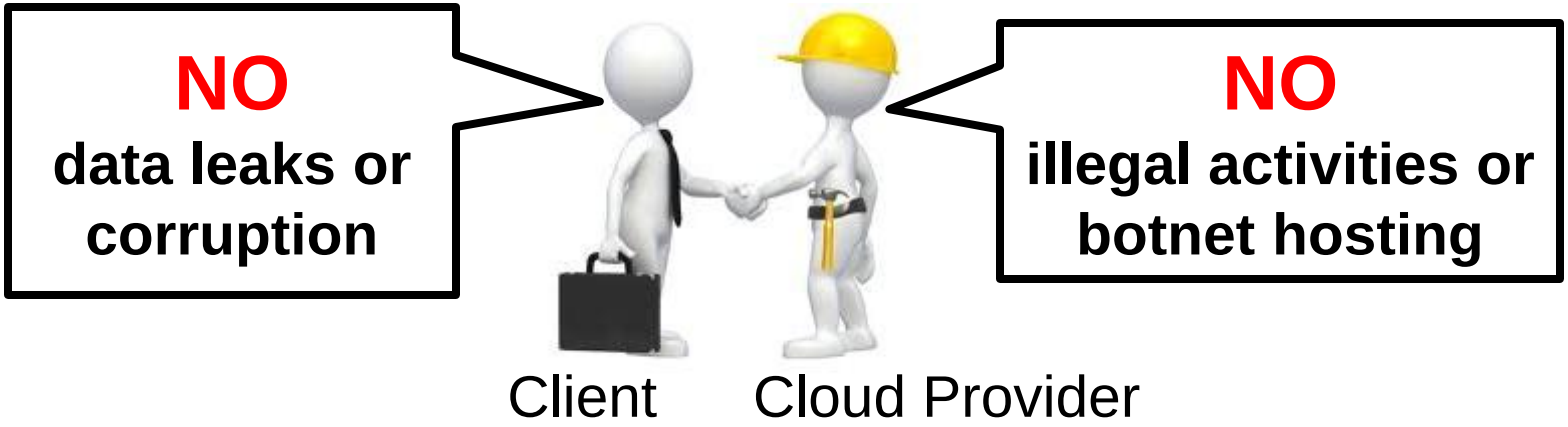
# SSC's privilege model



# Key technical challenges

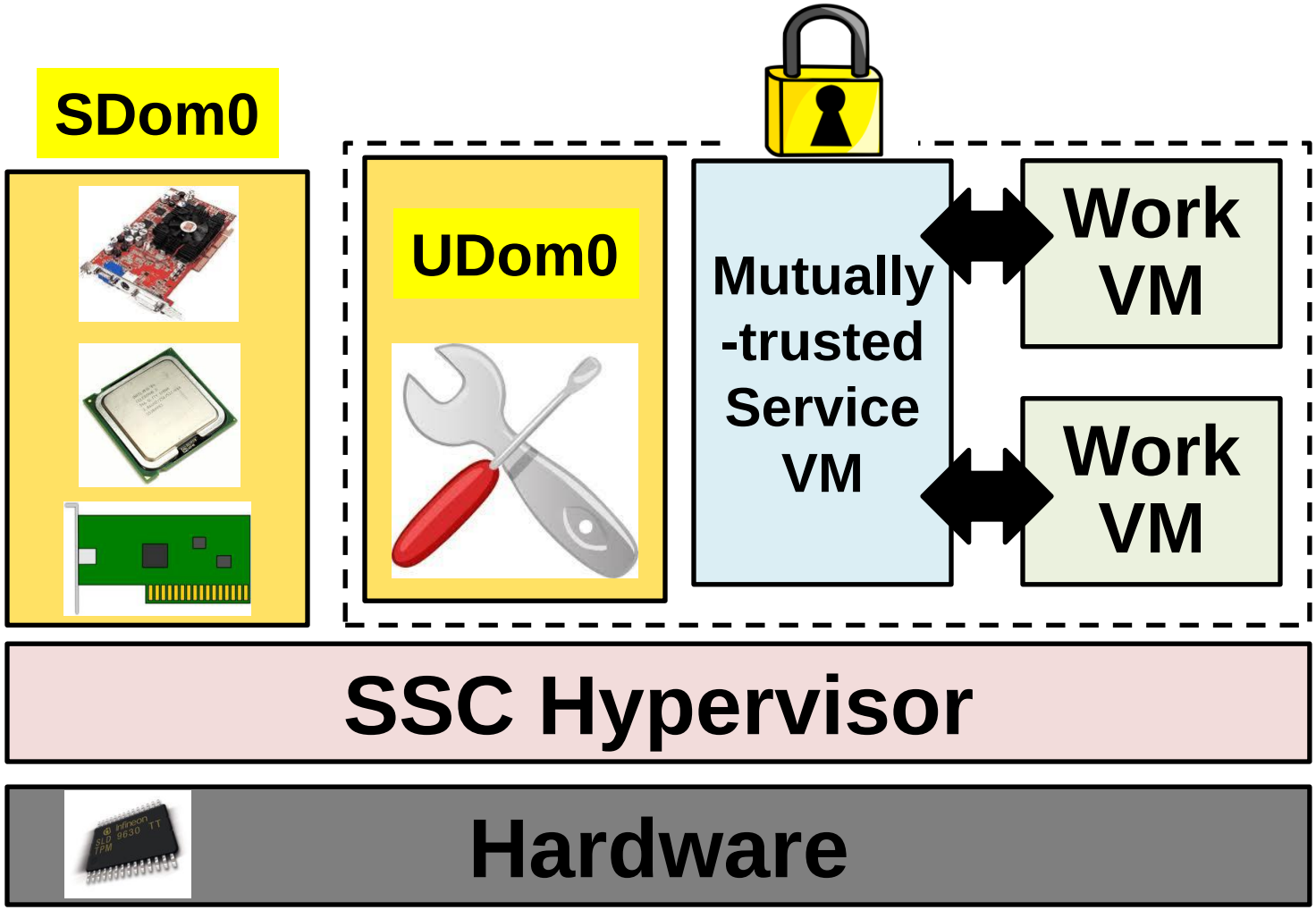
1. **Providers want *some* control**
   - To enforce regulatory compliance (SLAs, etc.)
   - **Solution**: Mutually-trusted service VMs
2. **Building domains in a trustworthy fashion**
   - Sdom0 is not trusted
   - **Solution**: the Domain Builder
3. **Establishing secure channel with client**
   -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

# #1: 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**

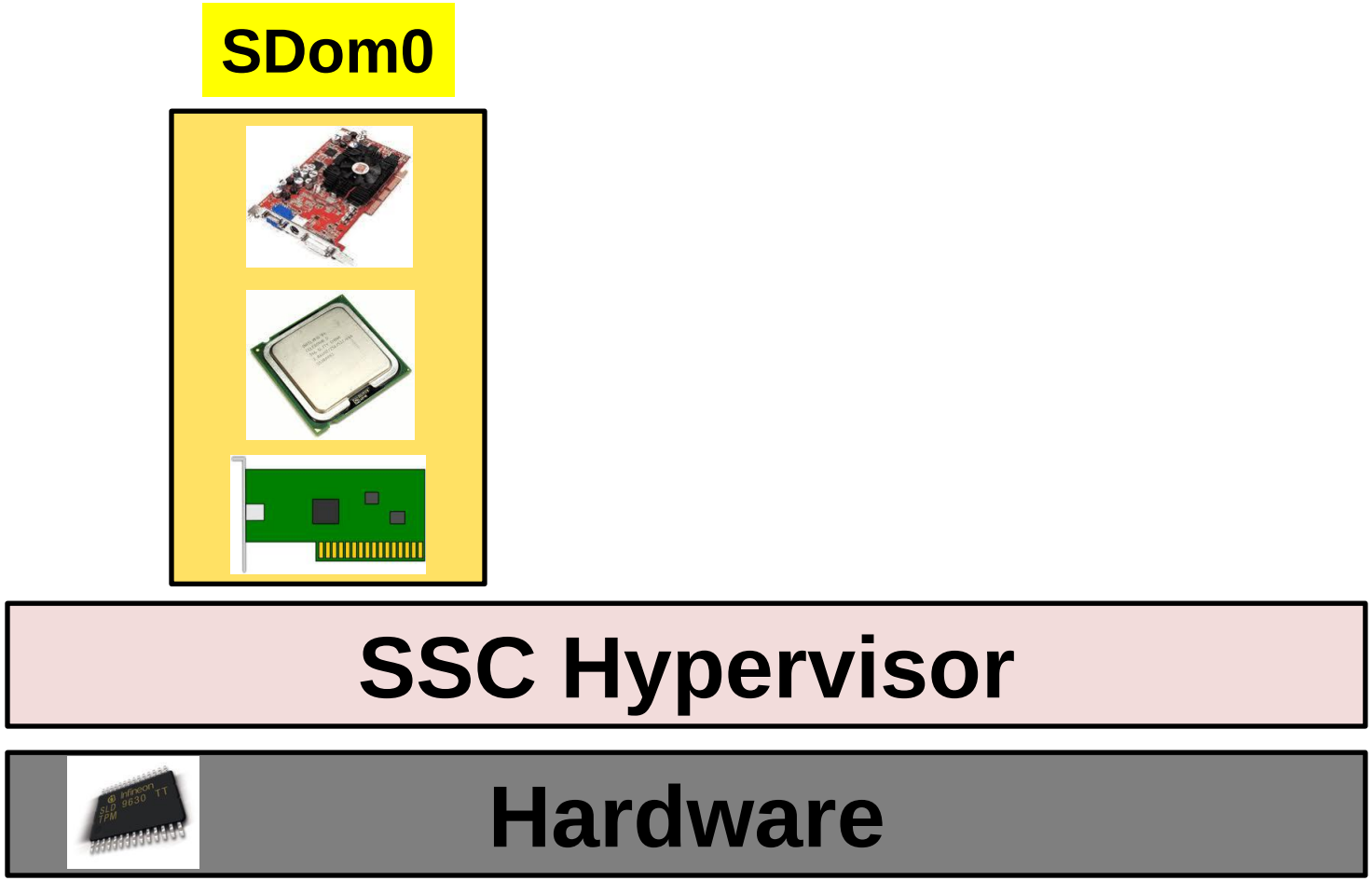
# Trustworthy regulatory compliance



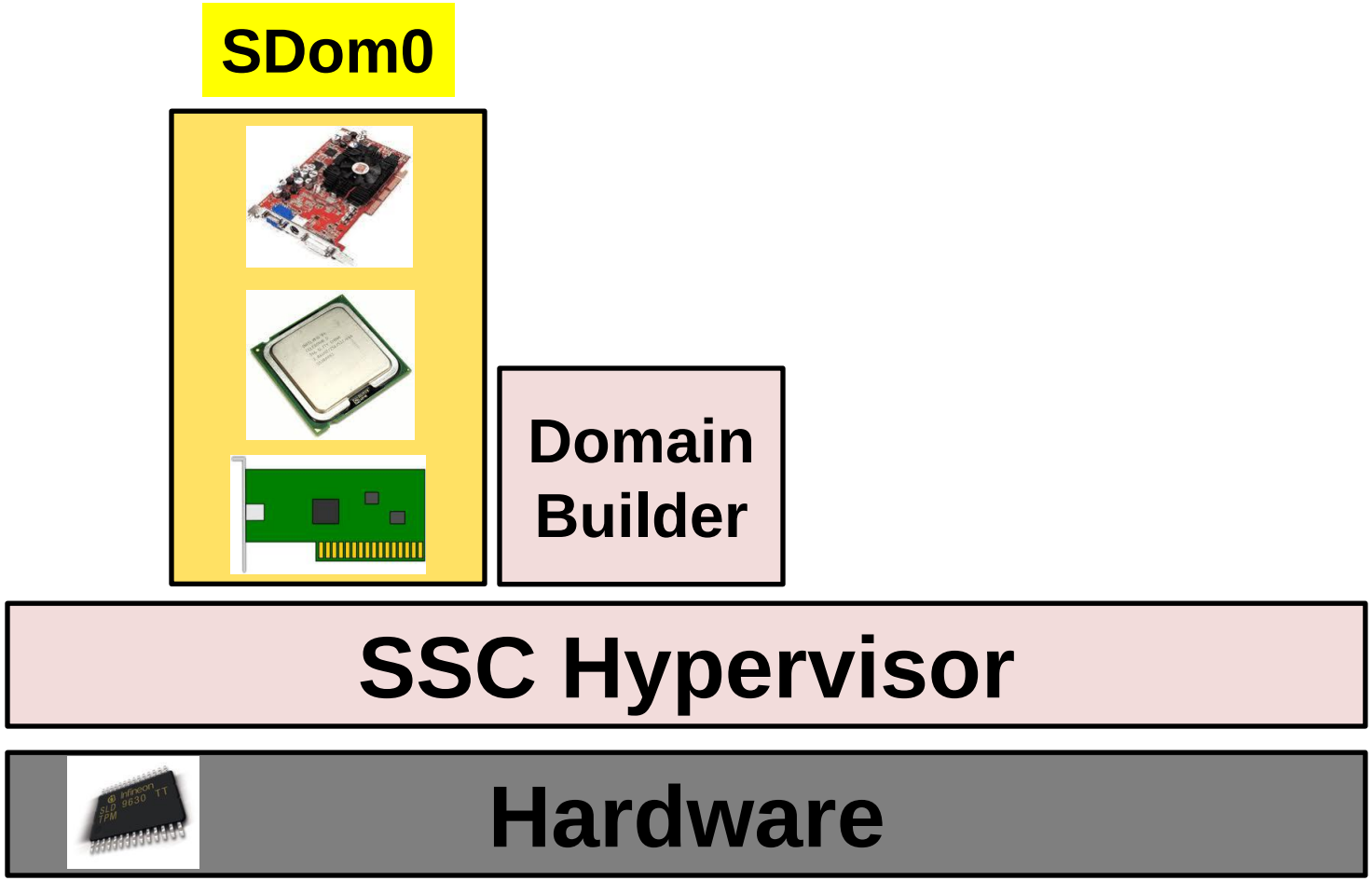
# Other applications of mutual trust

- **Mutually-trusted resource accounting**
  - Metering network usage, CPU consumption
- Today, resource accounting is done by the cloud provider
  - Clients can cross-check cloud provider
  - If results are inconsistent, who is correct?
- With mutually-trusted service VMs
  - Client and provider can agree on resource-accounting software

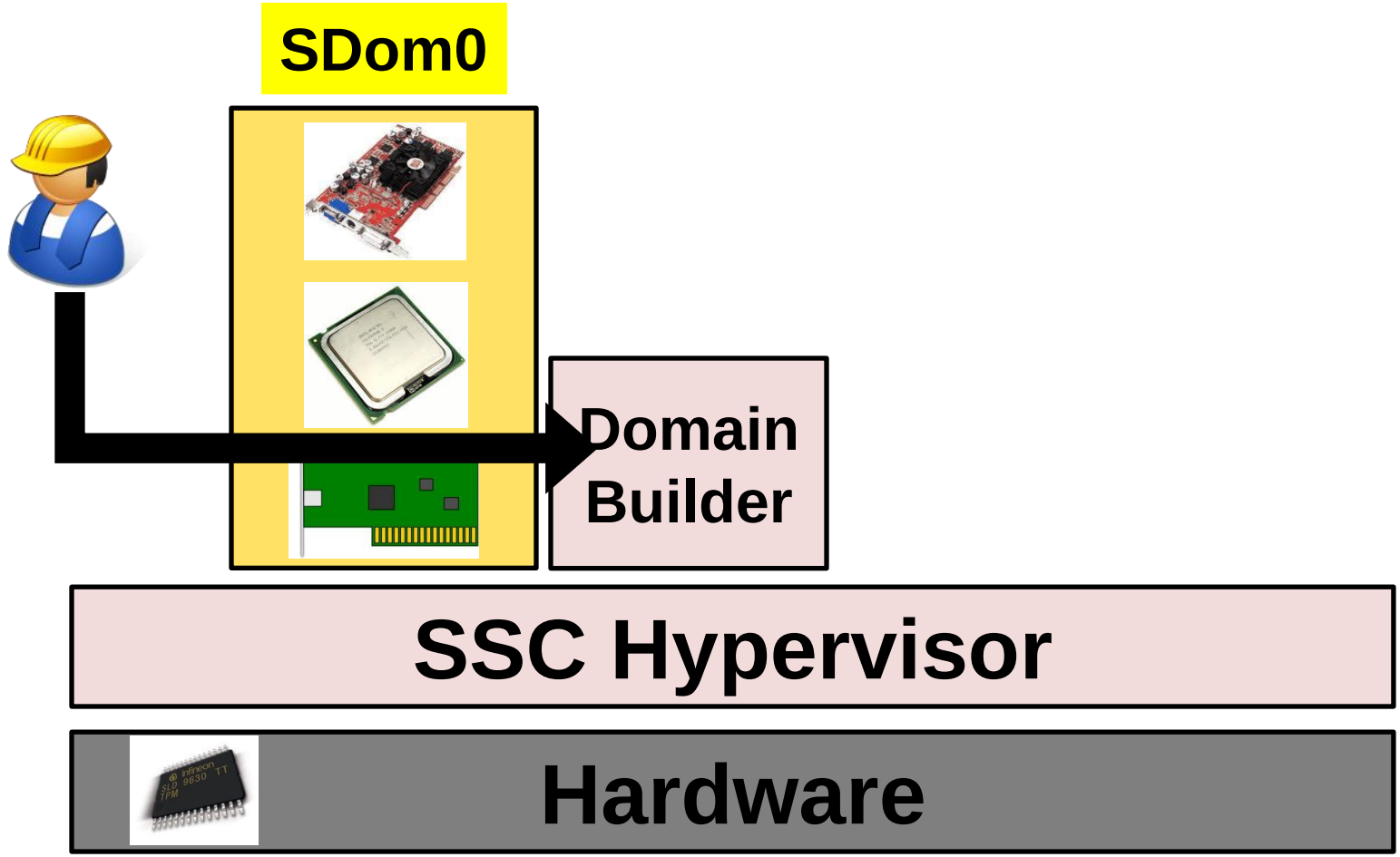
# #2 Bootstrap: the **Domain Builder**



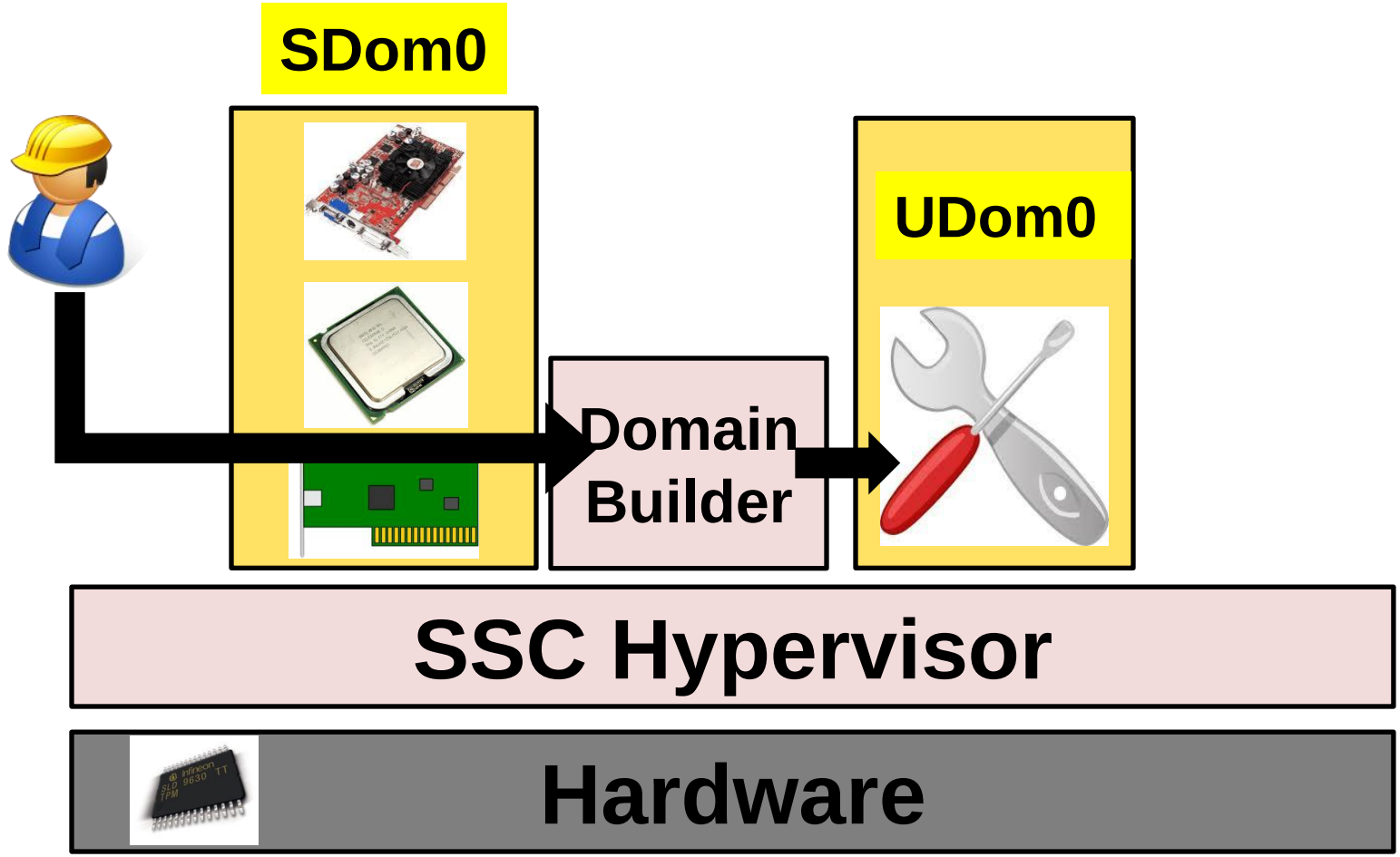
# #2 Bootstrap: the **Domain Builder**



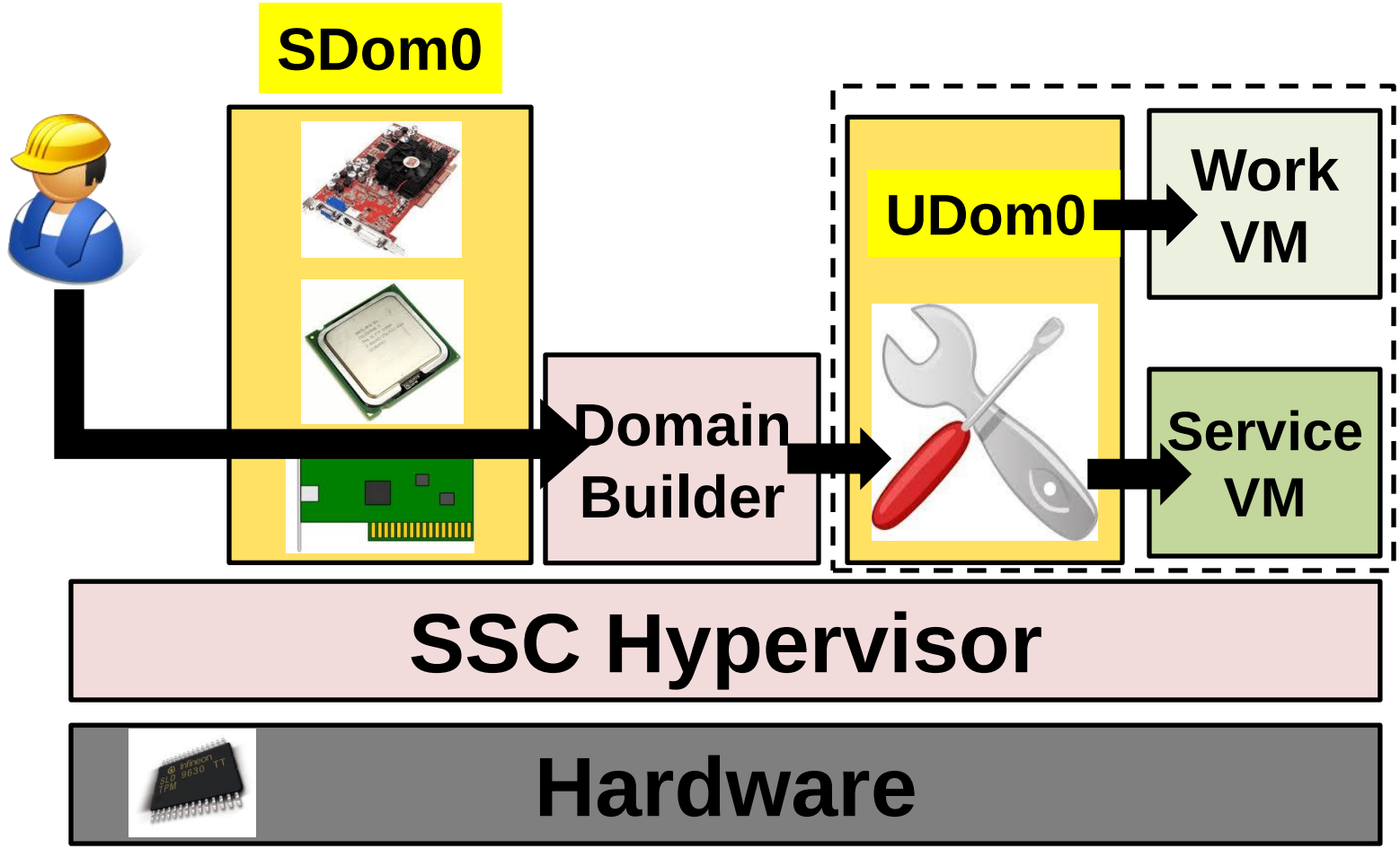
# #2 Bootstrap: the **Domain Builder**



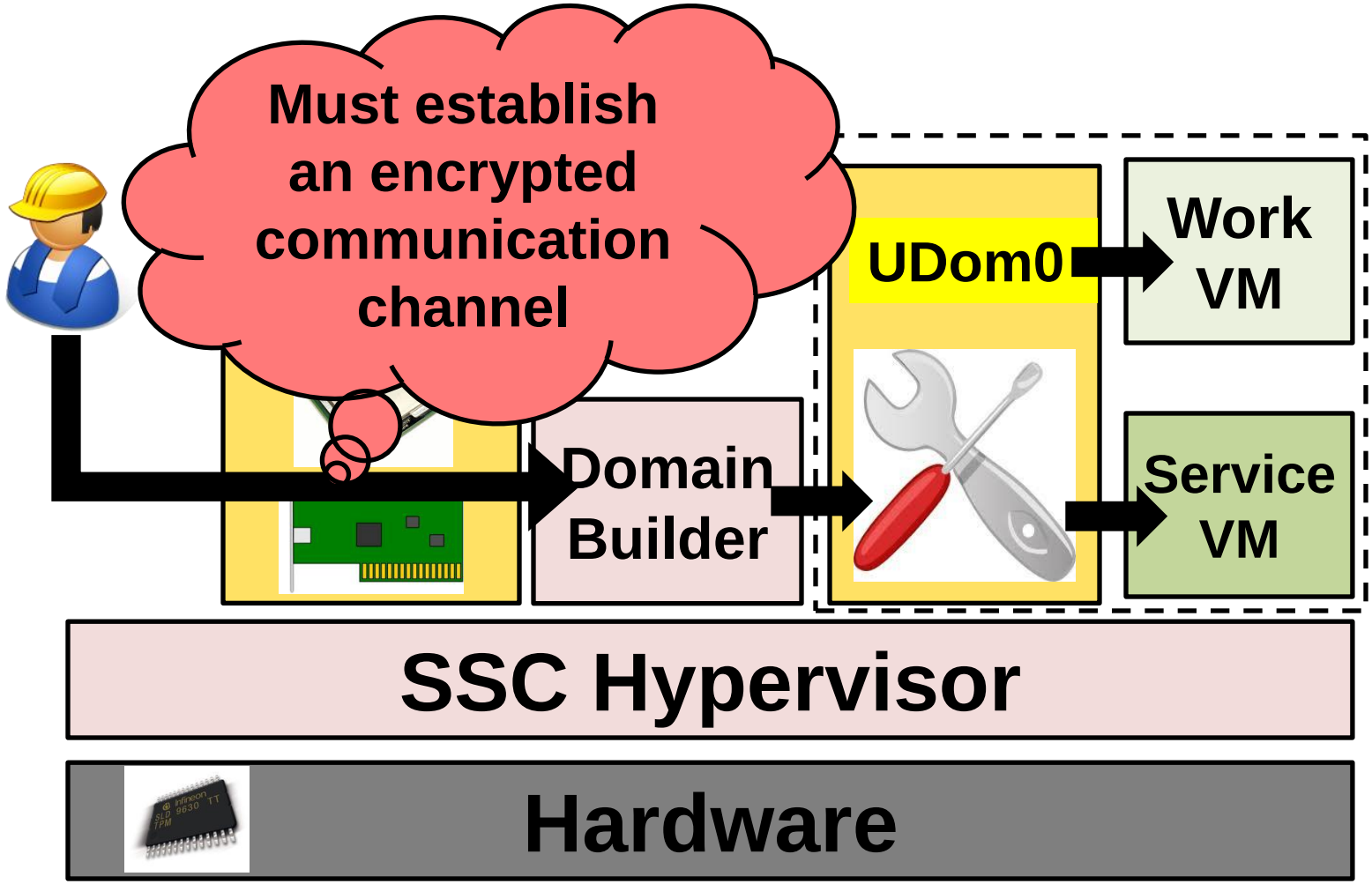
# #2 Bootstrap: the **Domain Builder**



# #2 Bootstrap: the **Domain Builder**



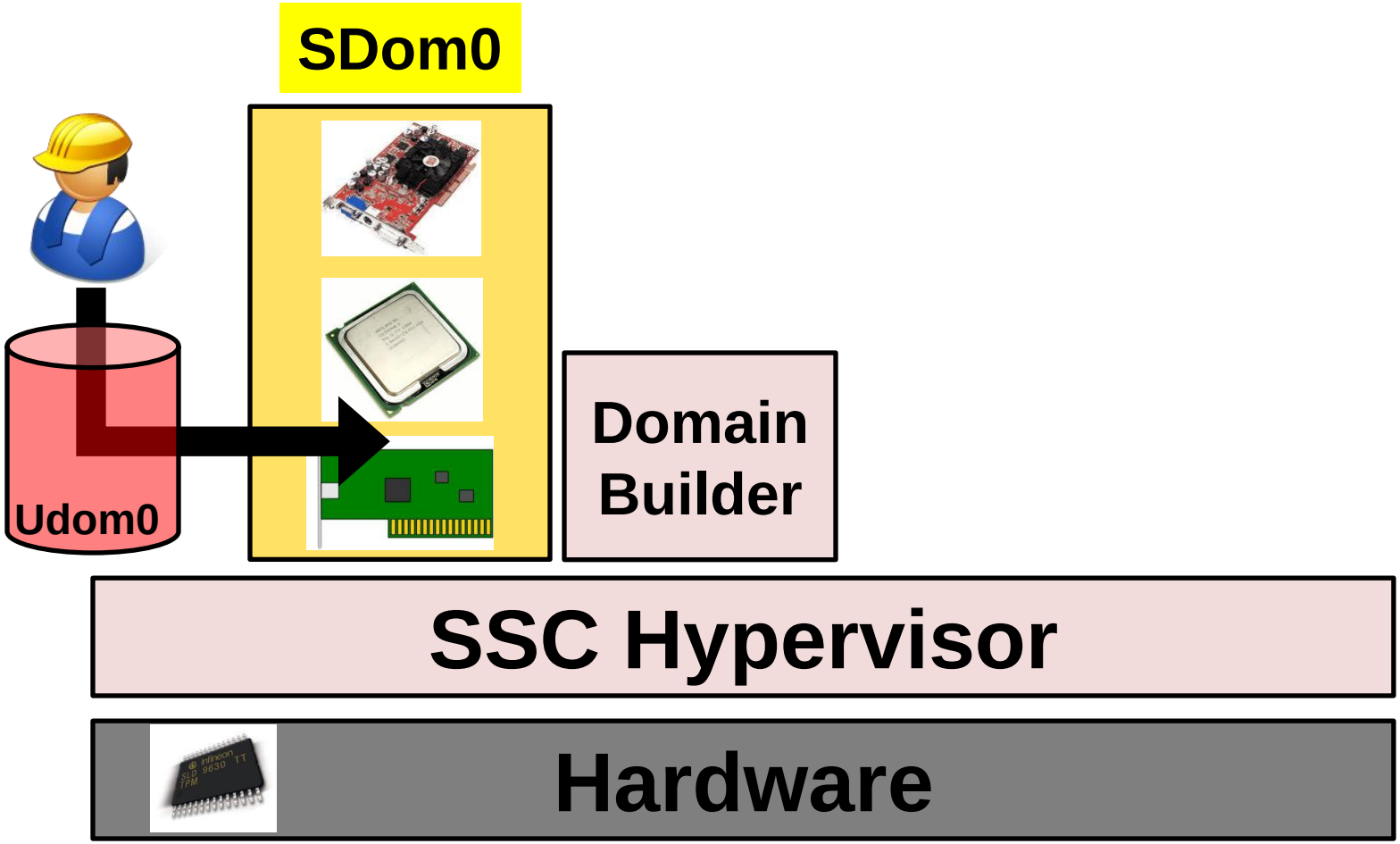
# #3 Secure bootstrap: SSL setup



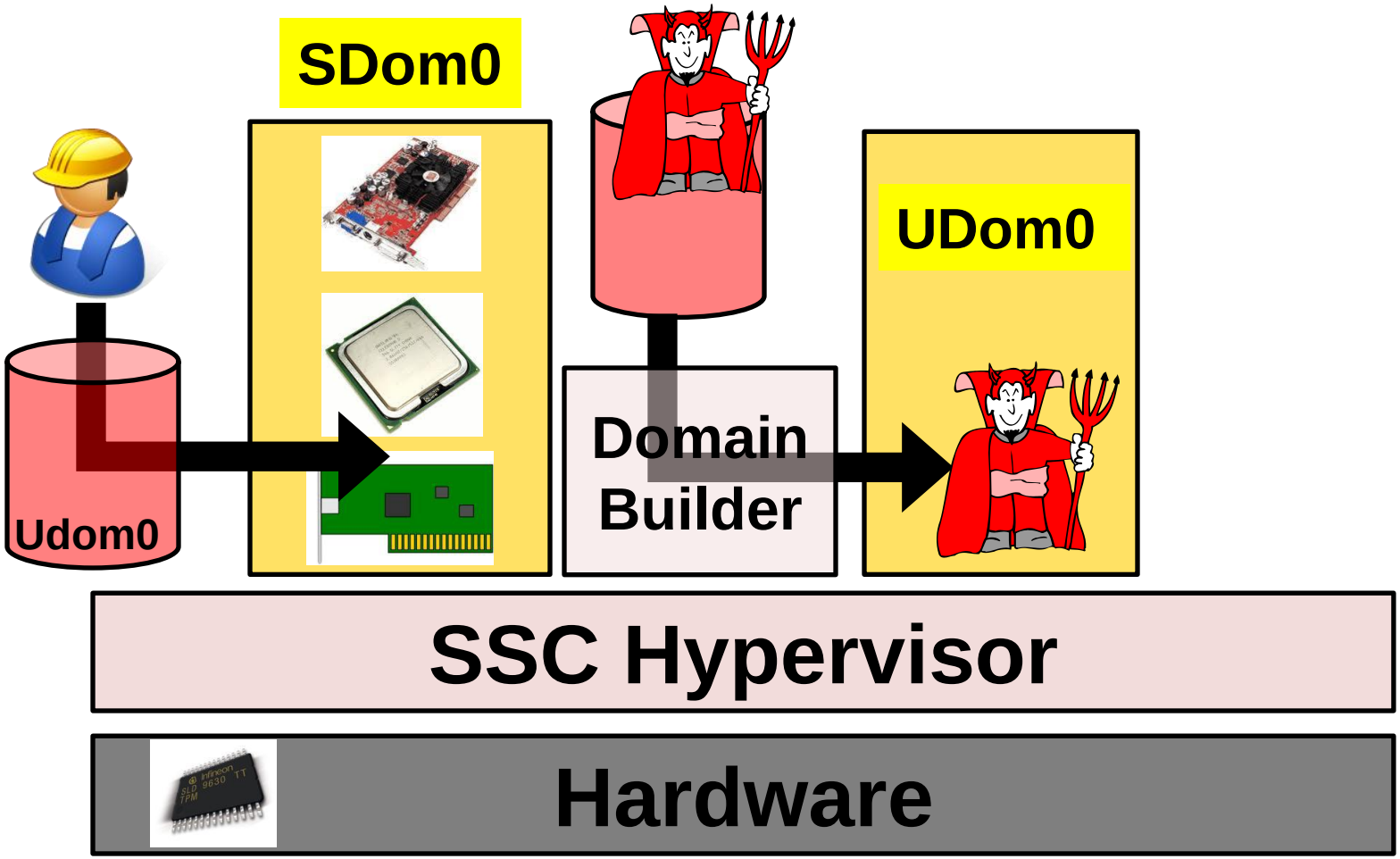
# Secure bootstrap protocol

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

# An evil twin attack



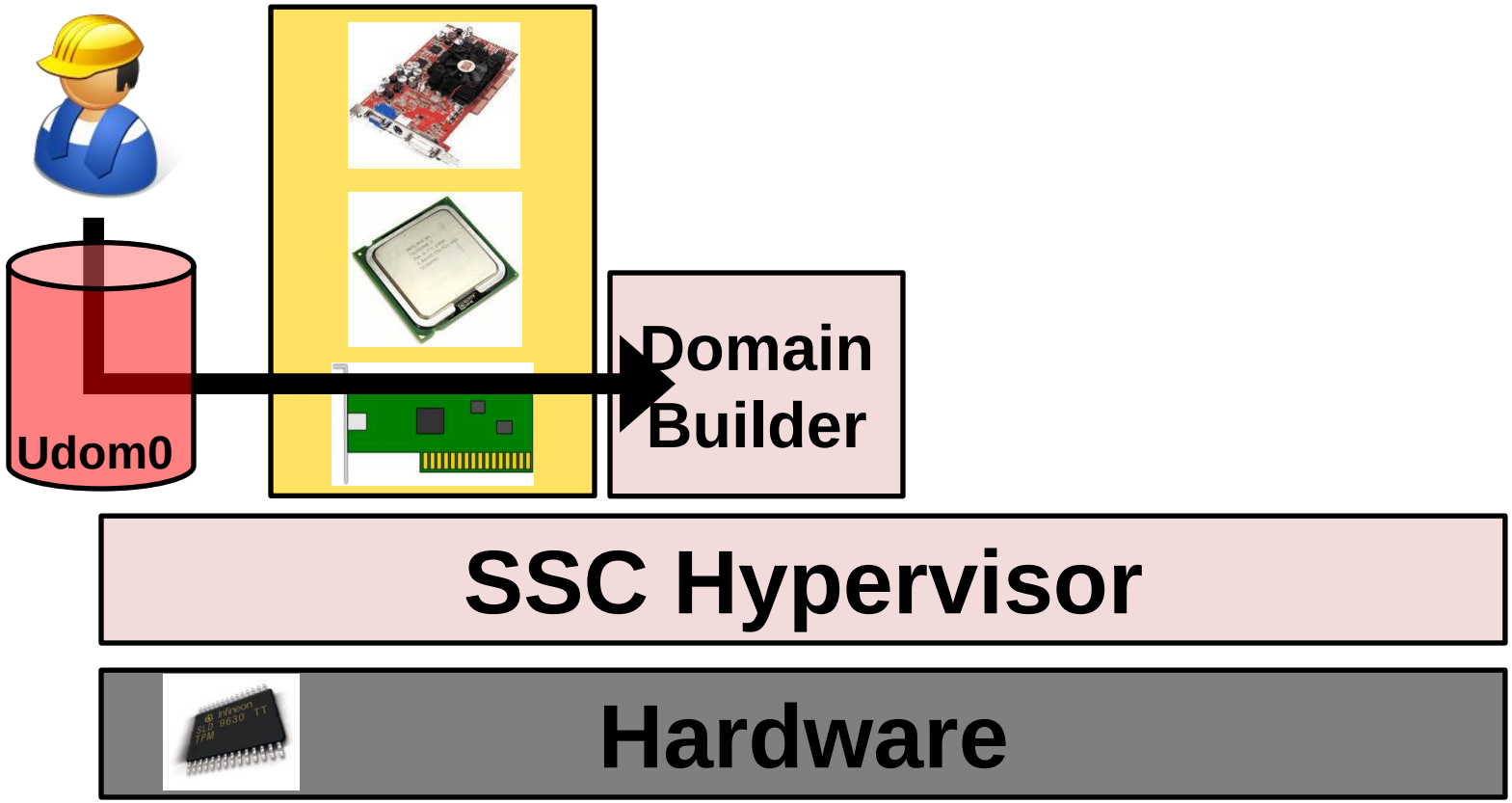
# An evil twin attack



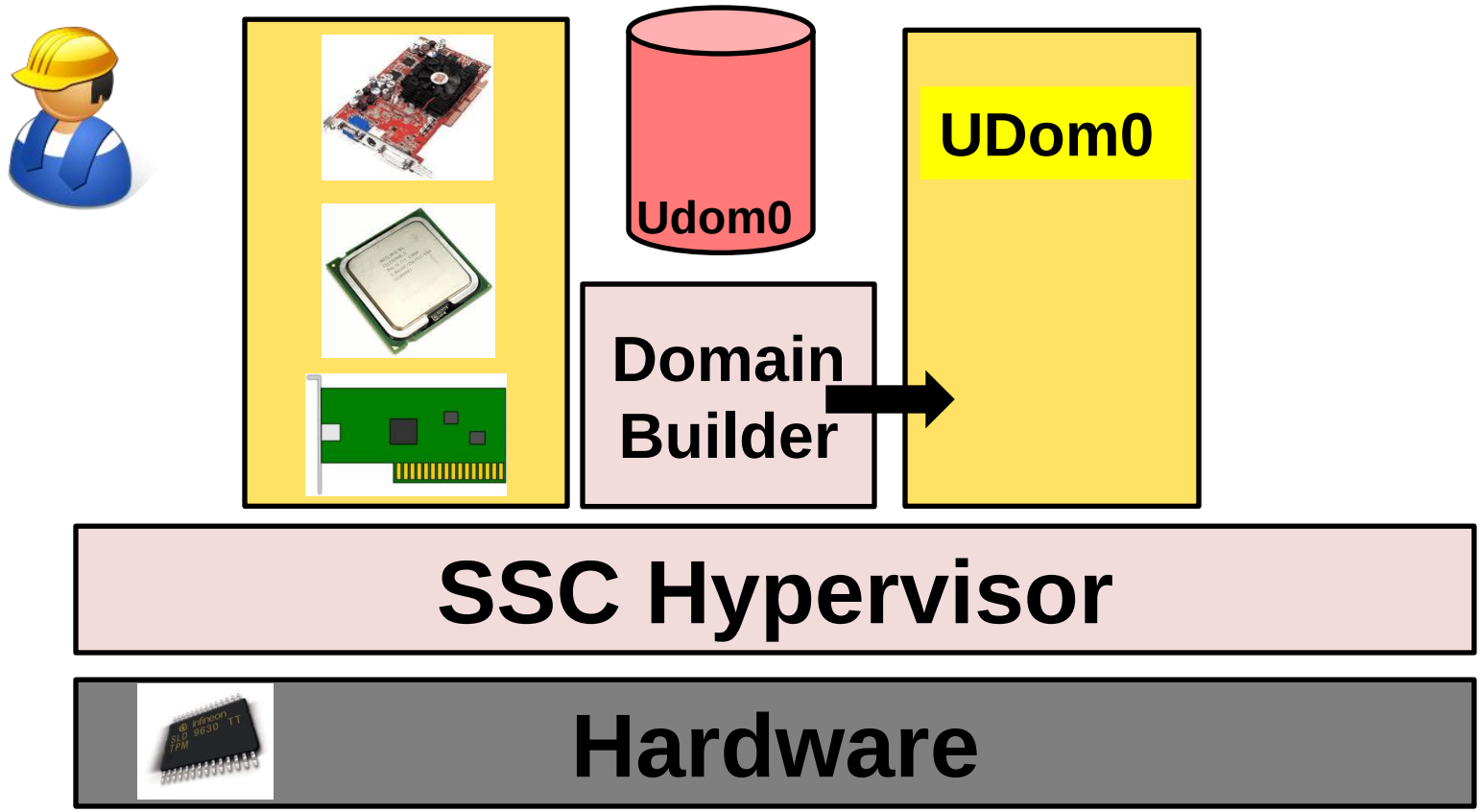
# Consequences of an evil twin

- Client must establish an SSL connection with its Udom0
  - SSL handshake requires the Udom0 to contain the client's SSL private key
  - Evil twin Udom0 can send this key to the malicious cloud administrator
- **Challenge**: Protect secrecy of client's SSL private key
- **Solution**: TPM and DomB-assisted secure bootstrap protocol

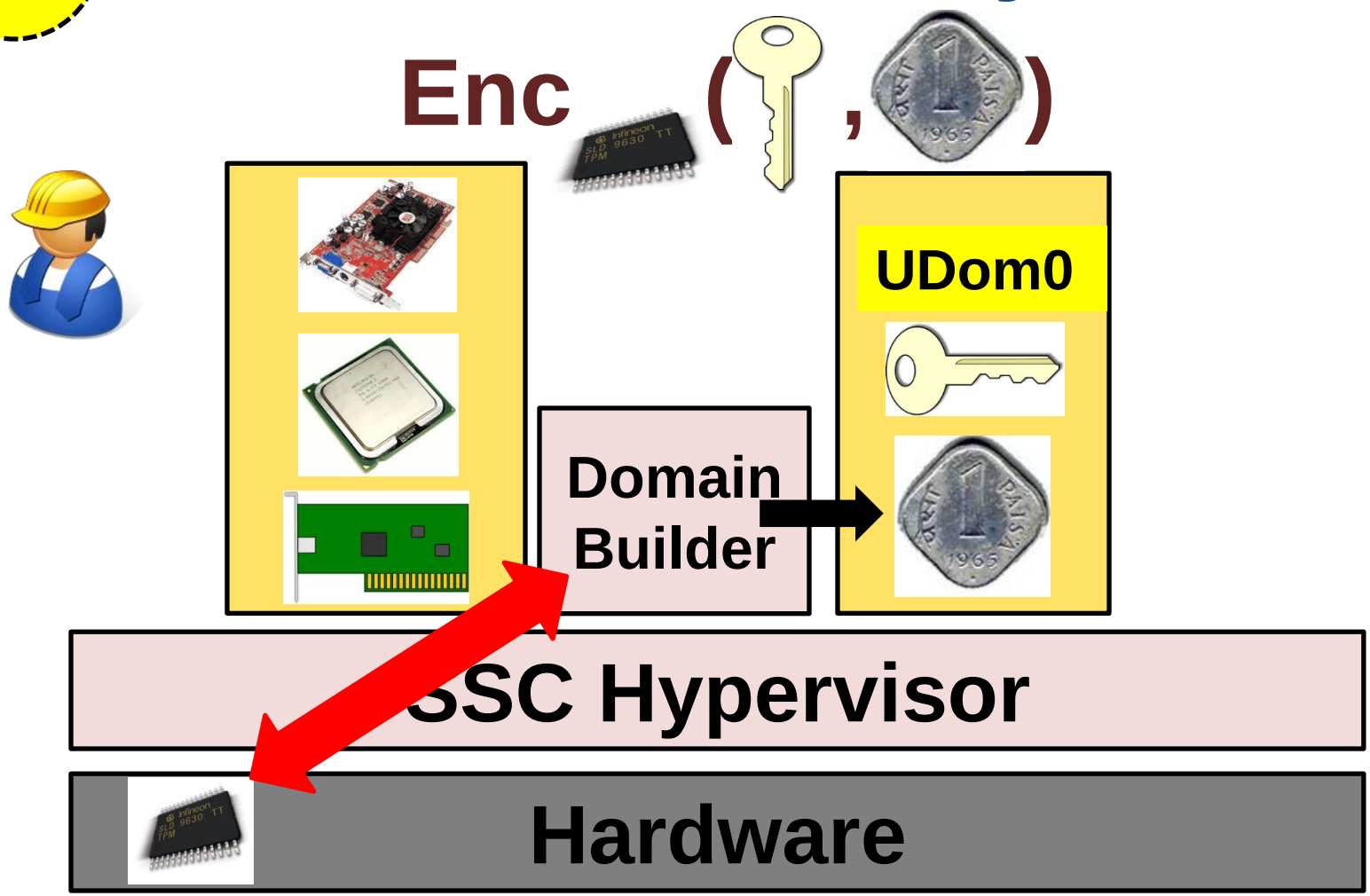
# 1 Udom0 image, Enc ( , )



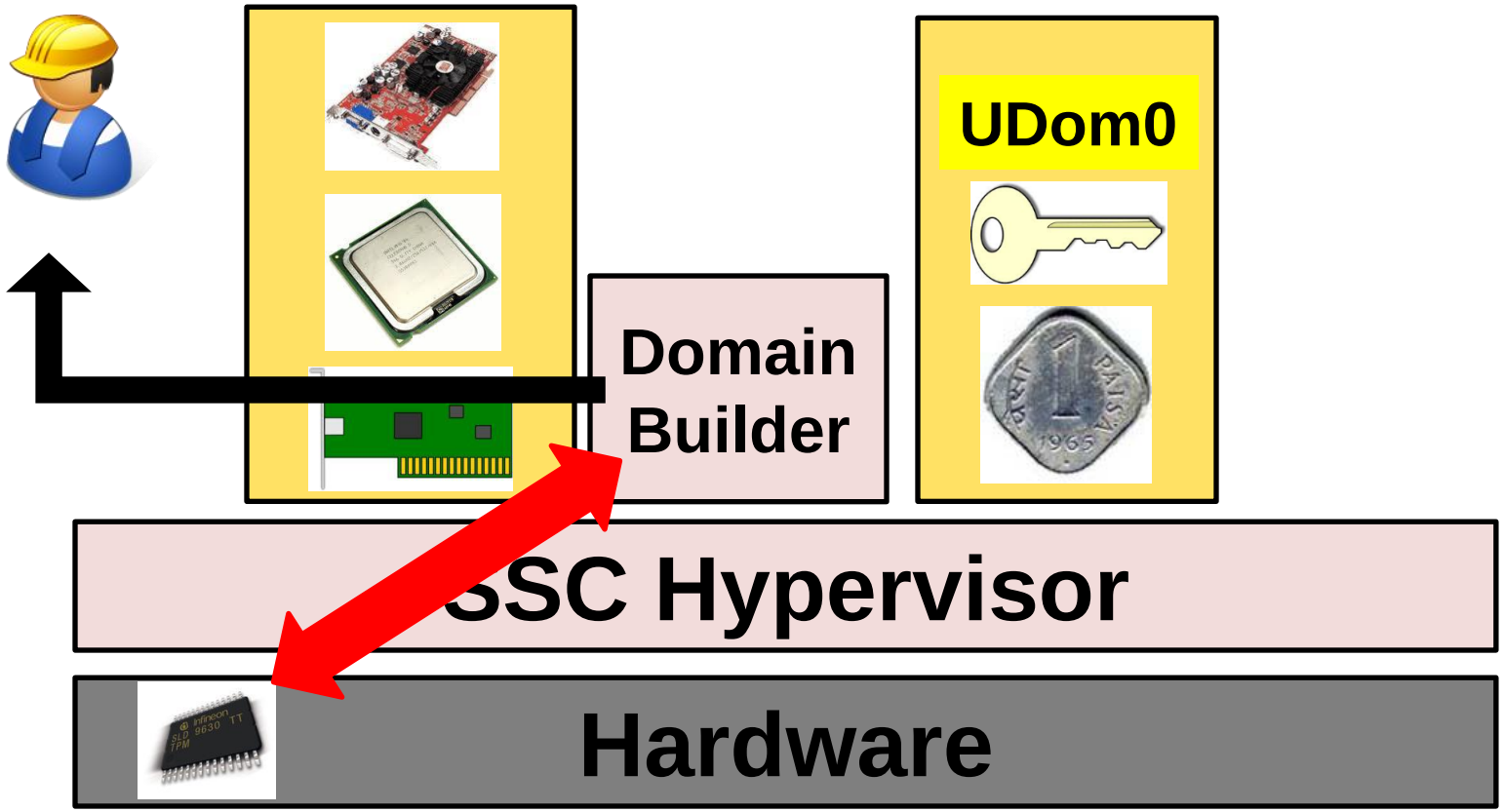
# 2 DomB builds domain



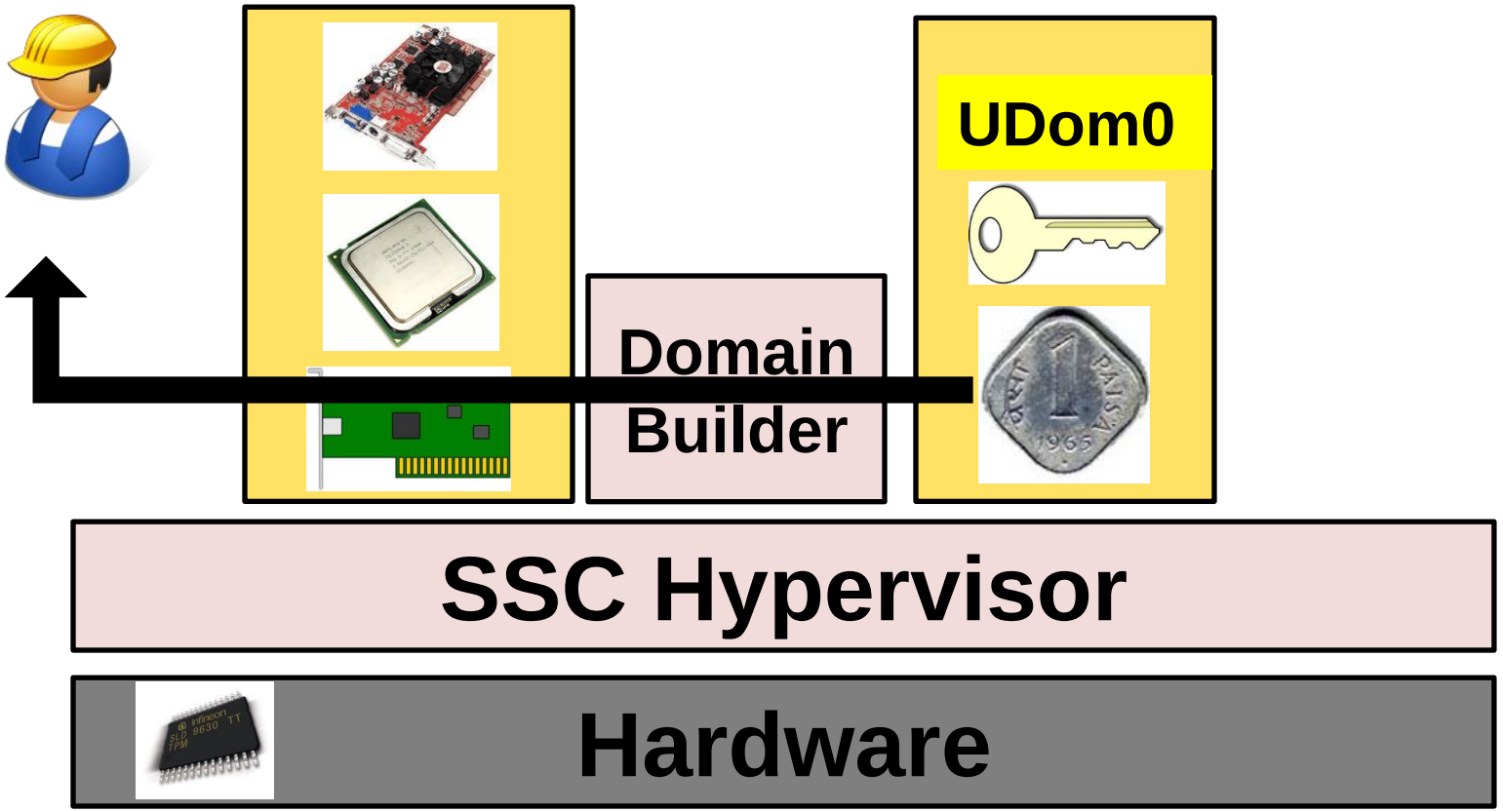
# 3 DomB installs key, nonce



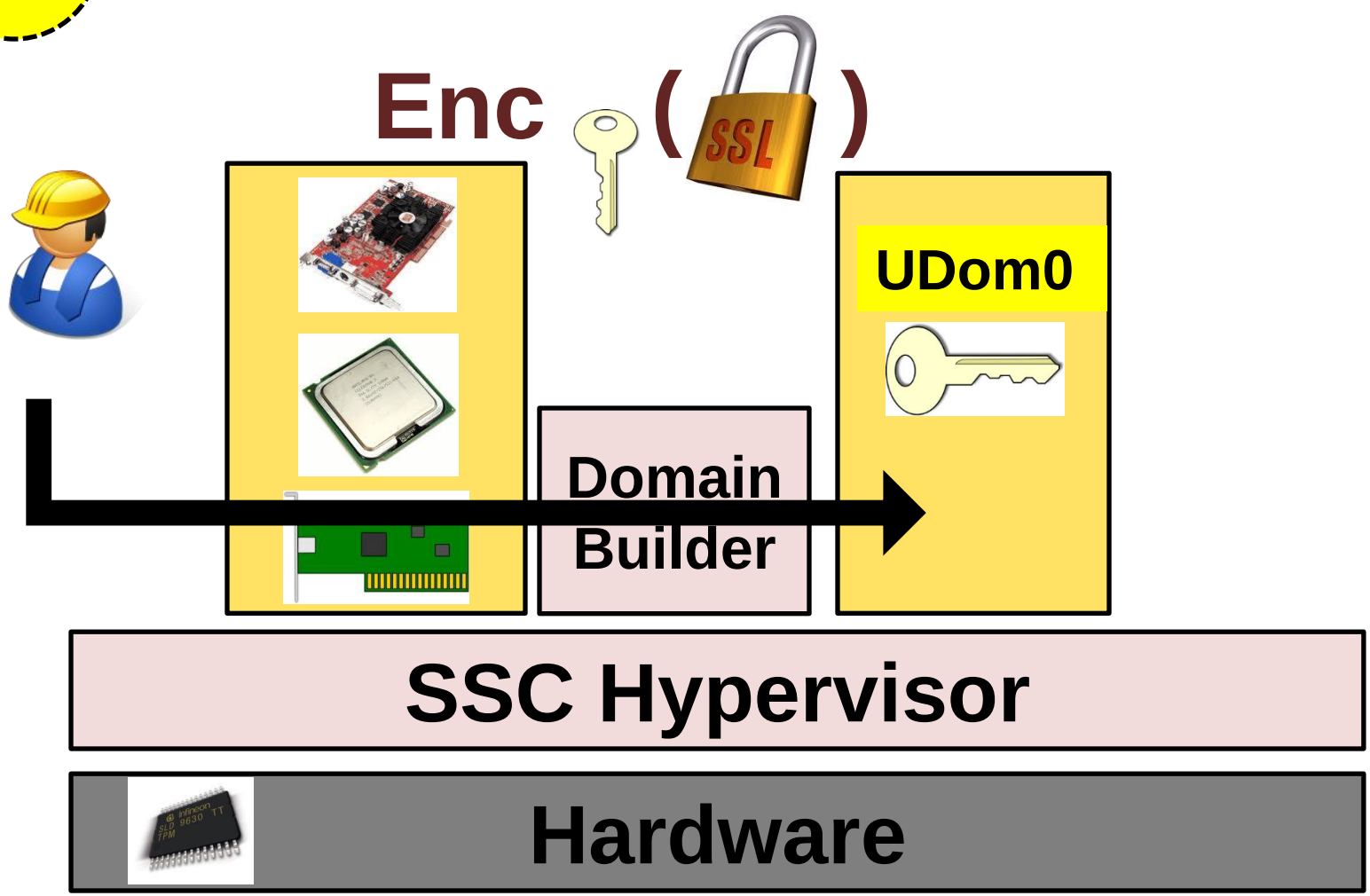
# 4 Client gets TPM hashes



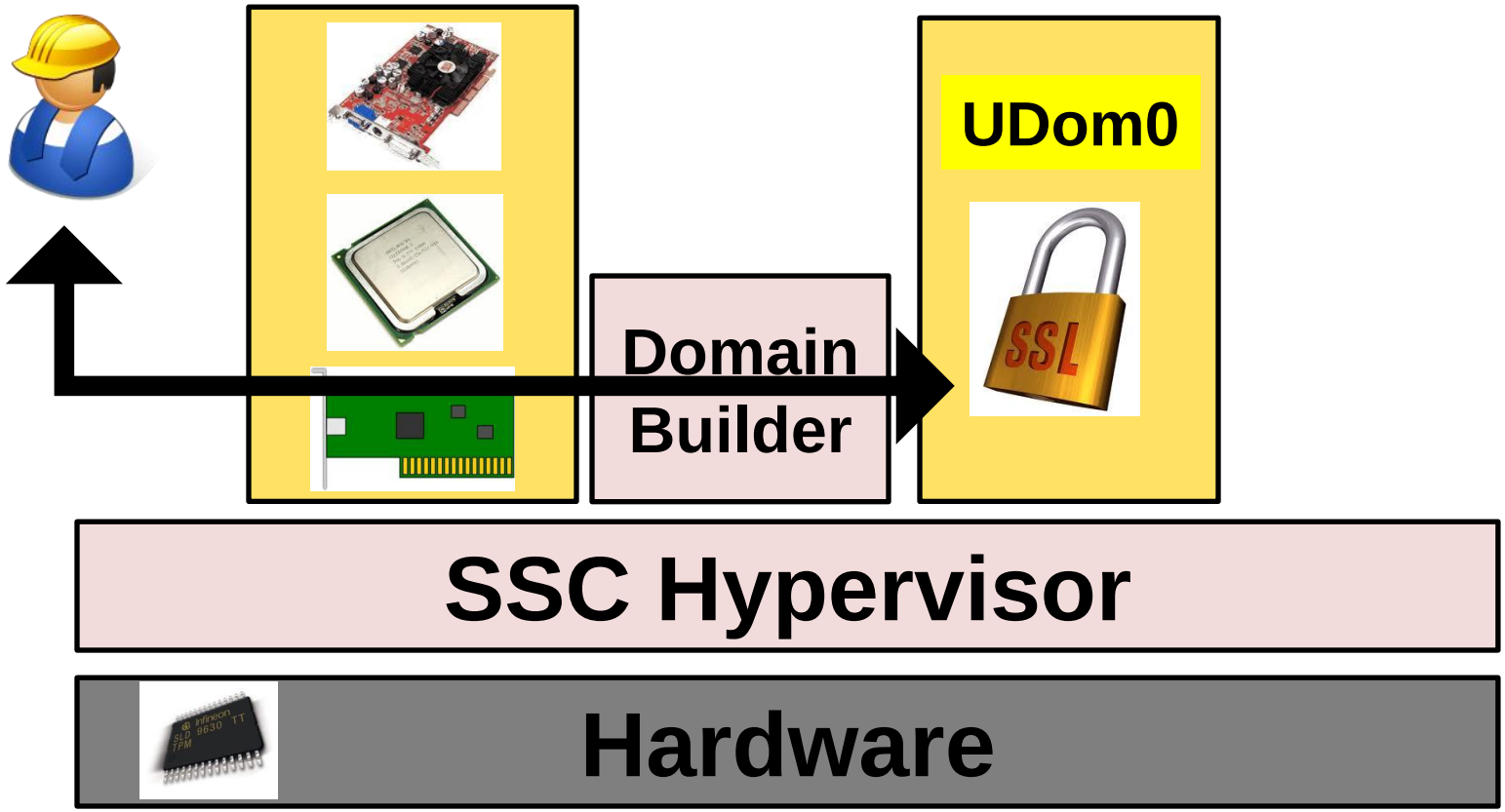
# 5 Udom0 sends  to client



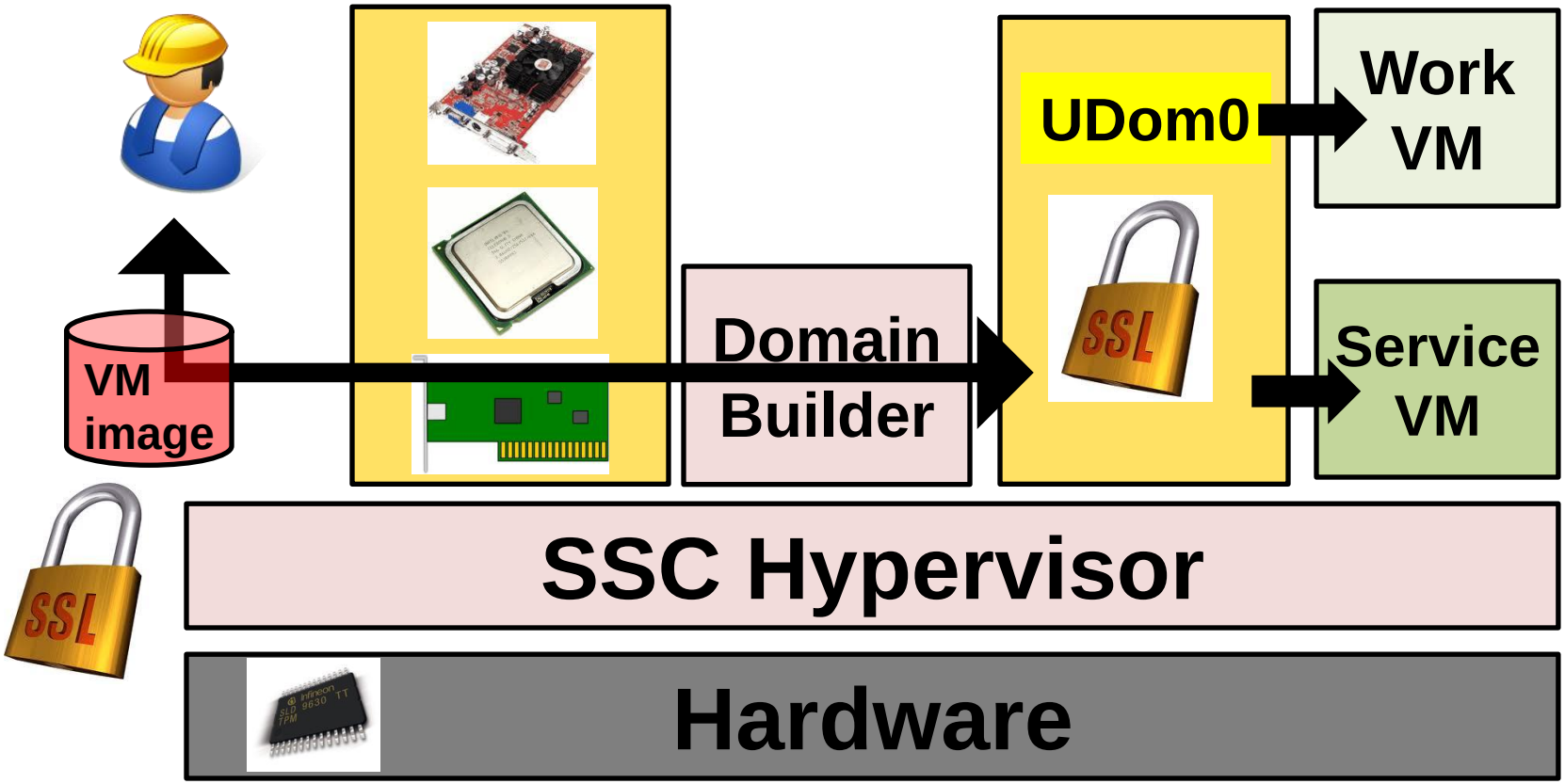
# 6 Client sends Udom0 SSL key



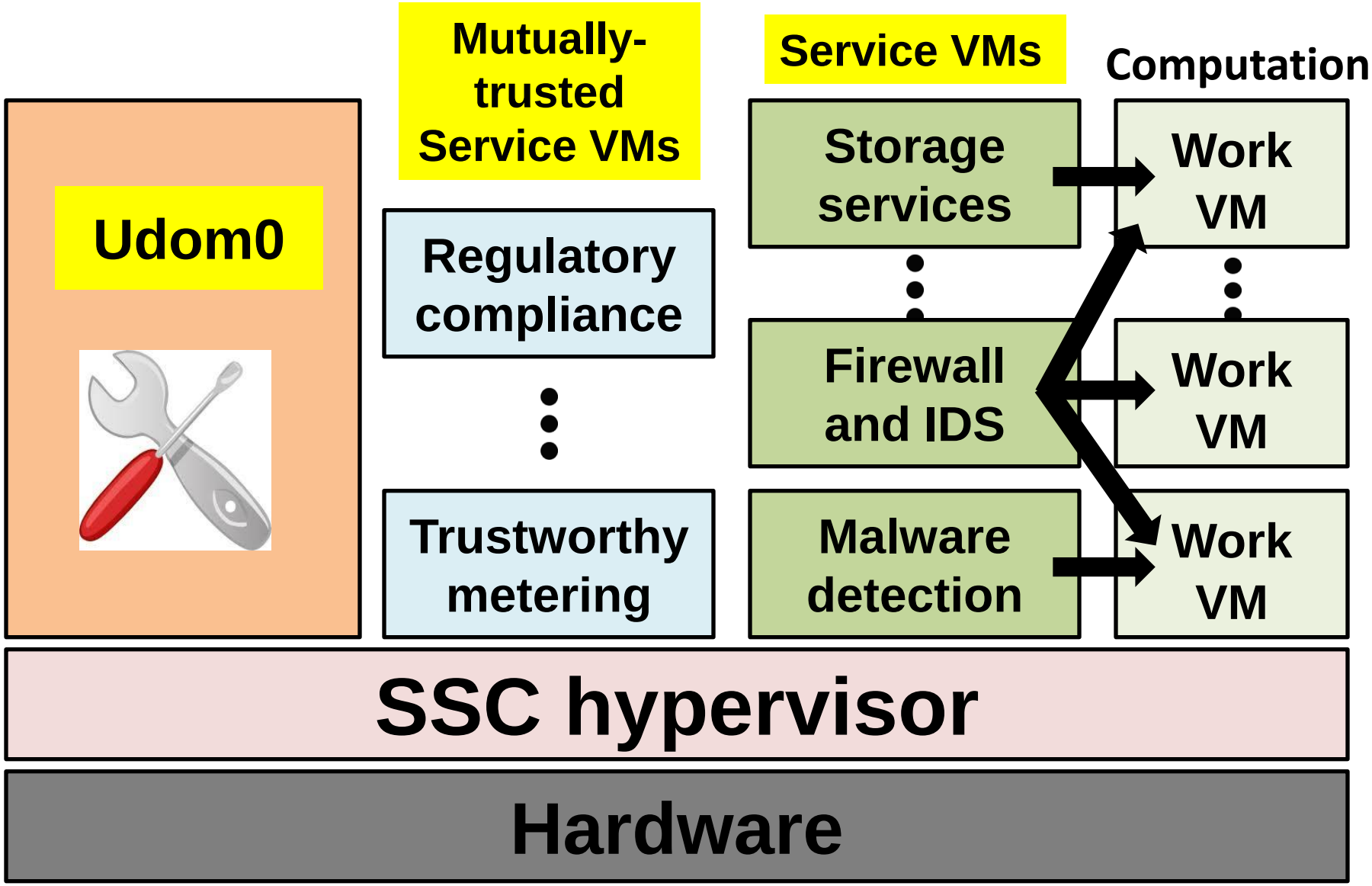
# 7 SSL handshake and secure channel establishment



# 8 Can boot other VMs securely



# Client meta-domains



# Case studies: Service VMs

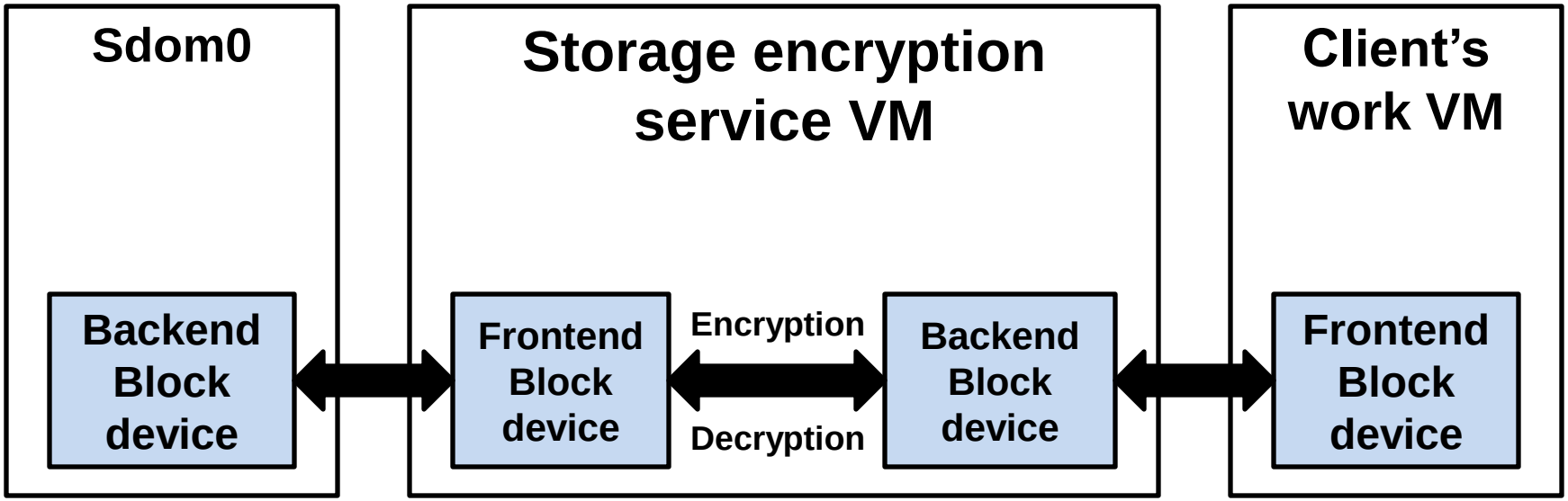
- Storage services: Encryption, Intrusion detection
- Security services:
  - Kernel-level rootkit detection
  - System-call-based intrusion detection
- Data anonymization service
- Checkpointing service
- Memory deduplication
- Network firewalls and intrusion detection systems
- Trustworthy network accounting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

# Evaluation

- Goal: Measure overhead of SSC
- Equipment: 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 two service VMs
  - Our **[ACM CCS'12]** and **[ACM SOCC'14]** papers present many more

# Storage encryption service VM

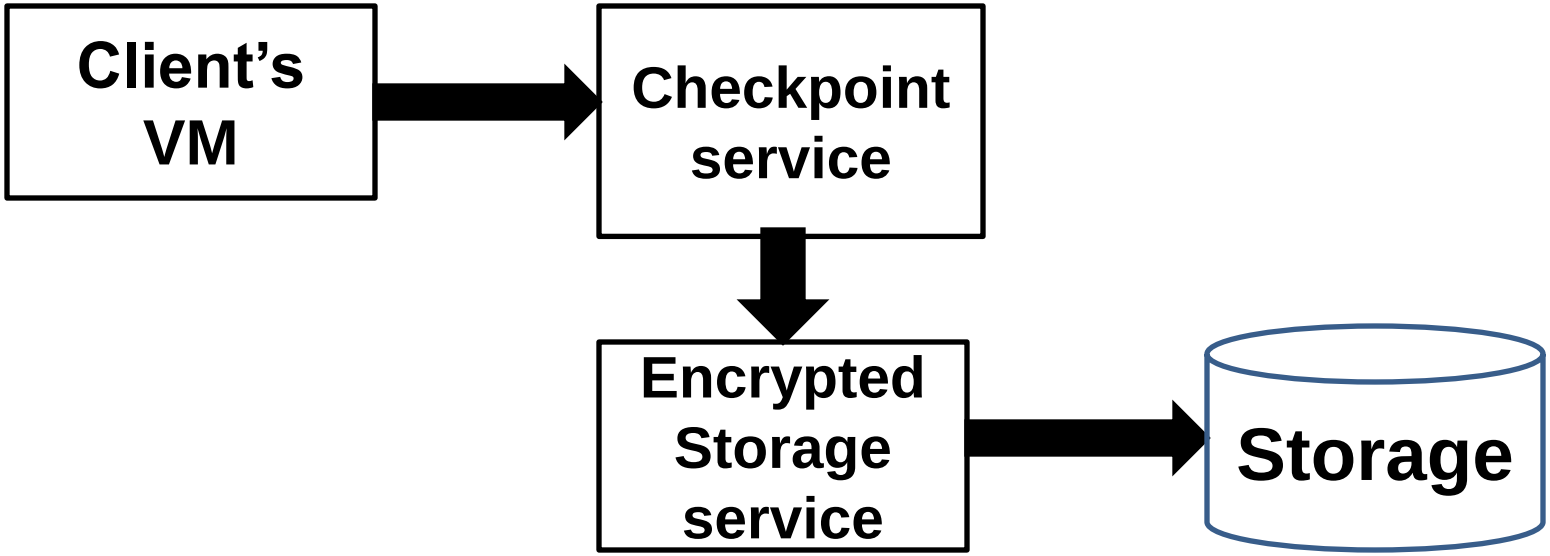
# Storage encryption service VM




| 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 |

# Recent developments: Intel SGX

- Recall SSC's threat model:

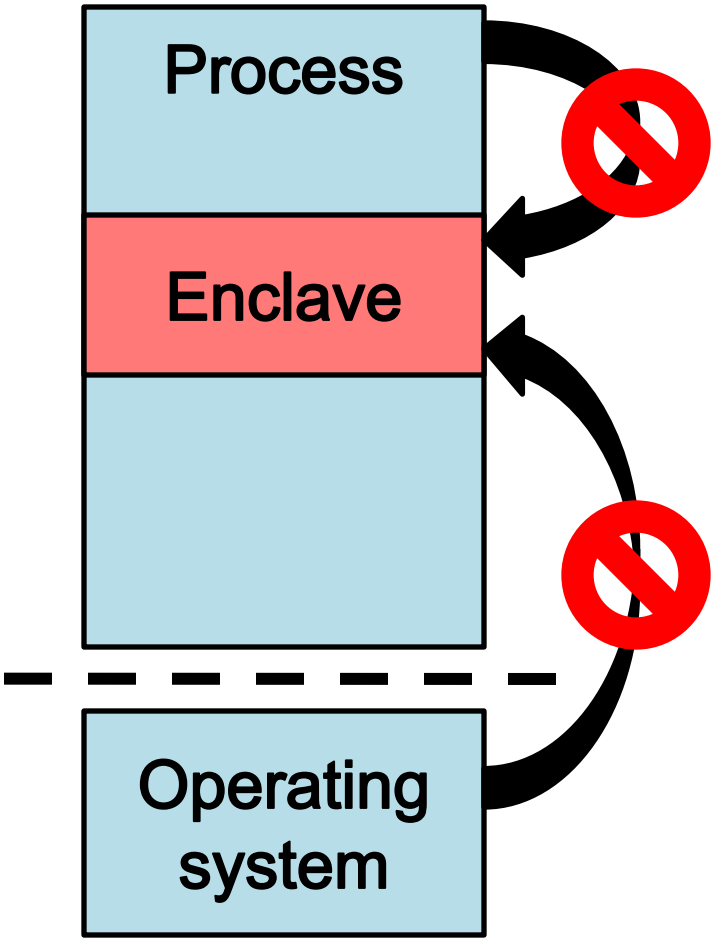
| Provider | Administrator |
|---|---|
|  |  |

- Intel SGX's threat model:

| Provider | Administrator |
|---|---|
|  |  |

# Background on the Intel SGX

- Hardware support for in-process ***enclaves***
- Processor encrypts enclave contents
  - Content accessible in the clear only from the same enclave
  - No access even from the same process or the operating system



# Implications of the Intel SGX

- Client VM contents can be protected from the cloud provider **[Haven:OSDI'14,VC3:S&P'15]**
- Cloud provider can, at worst, launch denial of service attacks, but cannot affect confidentiality or integrity of client enclaves

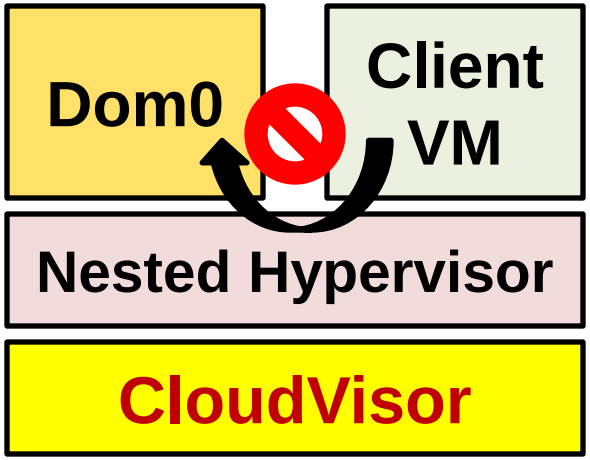
- **Question: Does Intel SGX obviate SSC?**
- **Answer: NO!**

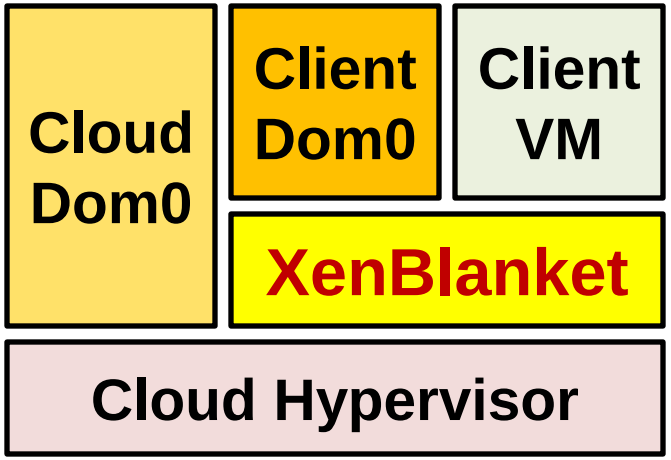
# SSC abstractions on Intel SGX

- SGX flips the threat model of traditional cloud platforms in favor of clients
  - SGX enclaves can violate regulatory compliance (SLAs)
  - And cloud provider has no way to determine if a violation has happened!
- SSC-like mutual trust abstraction may still be useful on SGX-enabled cloud platforms

# Other 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

… Enables a notion of mutual trust between cloud providers and clients

# Other research projects

**Theme: Computer Security and Software Engineering**

- Other aspects of cloud platform security
  **[ACSAC'08a, RAID'10, ANCS'11]**
- Operating system reliability and security
  **[ASPLOS'08, ACSAC'08b, ACSAC'09a, MobiSys'11, TDSC'11, TIFS'13]**
- Hardware support for software and system security
  **[CCS'08, ECOOP'12a, TIFS'13, MobiSys'16-sub]**
- Web application and Web browser security
  **[ACSAC'09b, ECOOP'12a, ECOOP'12b, ECOOP'14, FSE'14]**
- Tools for cross-platform mobile app development
  **[ICSE'13, ASE'15]**
- Retrofitting legacy software for security
  **[CCS'05, Oakland'06, ASPLOS'06, ICSE'07, CCS'08, CCS'12b]**
- Reverse-engineering x86 and ARM binary software
  **[ICSE'16]**

# Collaborators

## Senior colleagues

**Rutgers**

- Prof. Liviu Iftode
- Prof. Santosh Nagarakatte

**Penn State**

- Prof. Trent Jaeger

**Wisconsin-Madison**

- Prof. Somesh Jha
- Prof. Thomas Reps

**TU-Darmstadt**

- Prof. Ahmad Reza-Sadeghi

**Google**

- Dr. Ulfar Erlingsson
- Dr. Andres Lagar-Cavilla

**Microsoft Research India**

- Dr. Sriram Rajamani

## Students

**☺ Graduated PhDs**

1. Dr. Mohan Dhawan (IBM Research)
2. Dr. Saman Zarandioon (Amazon EC2)
3. Dr. Shakeel Butt (Nvidia → Google)
4. Dr. Liu Yang (HP Labs → Baidu)
5. Dr. Rezwana Karim (Samsung Research)
6. Dr. Amruta Gokhale (Teradata)

**☺ Former Postdocs**

1. Dr. Arati Baliga (AT&T Security Labs)

**☺ Graduated MS students**

1. Jeffrey Bickford (AT&T Research)
2. Yogesh Padmanaban (Microsoft)

**😐 Current PhD students**

- J. P. Lim, H. Nguyen, D.Kim

**Email**: **vinodg@cs.rutgers.edu**
**URL**: **http://www.cs.rutgers.edu/~vinodg**