Systems and Internet Infrastructure Security

Network and Security Research Center
Department of Computer Science and Engineering
Pennsylvania State University, University Park PA

How Much Control Should Customers Demand over Cloud-Based Applications?

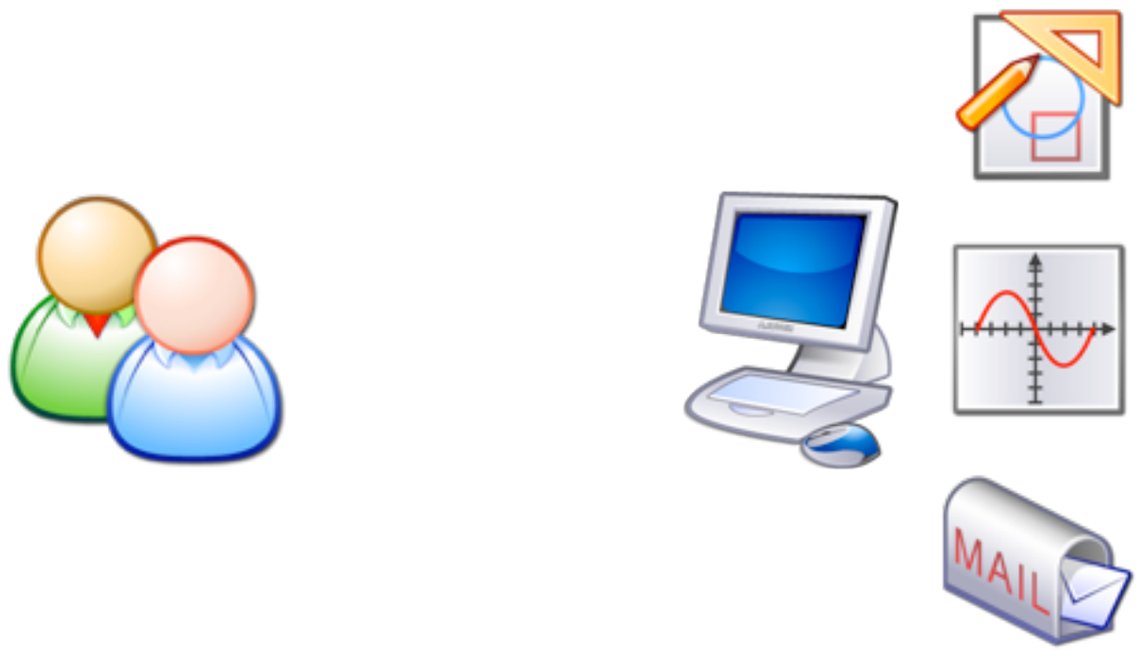
Trent Jaeger
Penn State University
Workshop on Trustworthy Clouds 2013

Cloudy Foundations

Cloudy Foundations

Cloudy Foundations

Cloudy Foundations

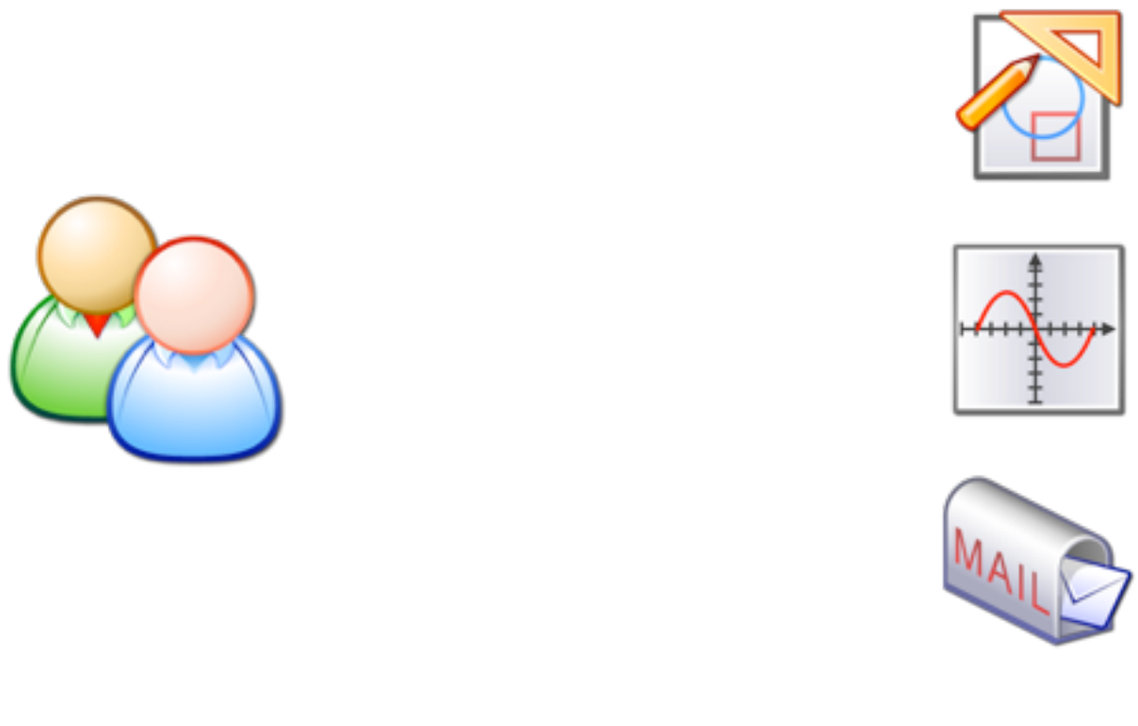


Cloudy Foundations

Cloudy Foundations

PENNSTATE 1855

rackspace HOSTING



Cloudy Foundations

PENNSTATE 1855

rackspace HOSTING

amazon web services™

Cloudy Foundations

PENNSTATE 1855

Cloudy Foundations

Cloudy Foundations

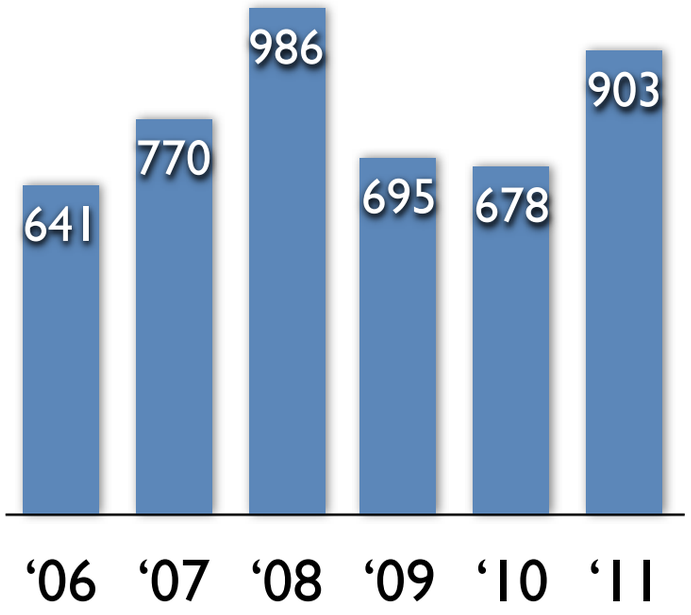
Cloudy Foundations

Can customers move their services and validate that they still protect data security?

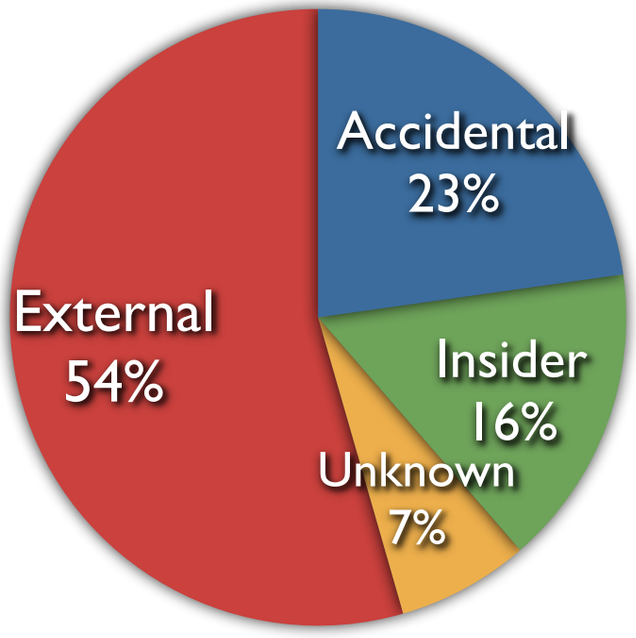
Reasons to Doubt

- History has shown they are vulnerable to attack
  - SLAs, audits, and armed guards offer few guarantees
  - Insiders can subvert even hardened systems

Data Loss Incidents



Incident Attack Vector



Credit: The Open Security Foundation datalossdb.org

Cloudy Future

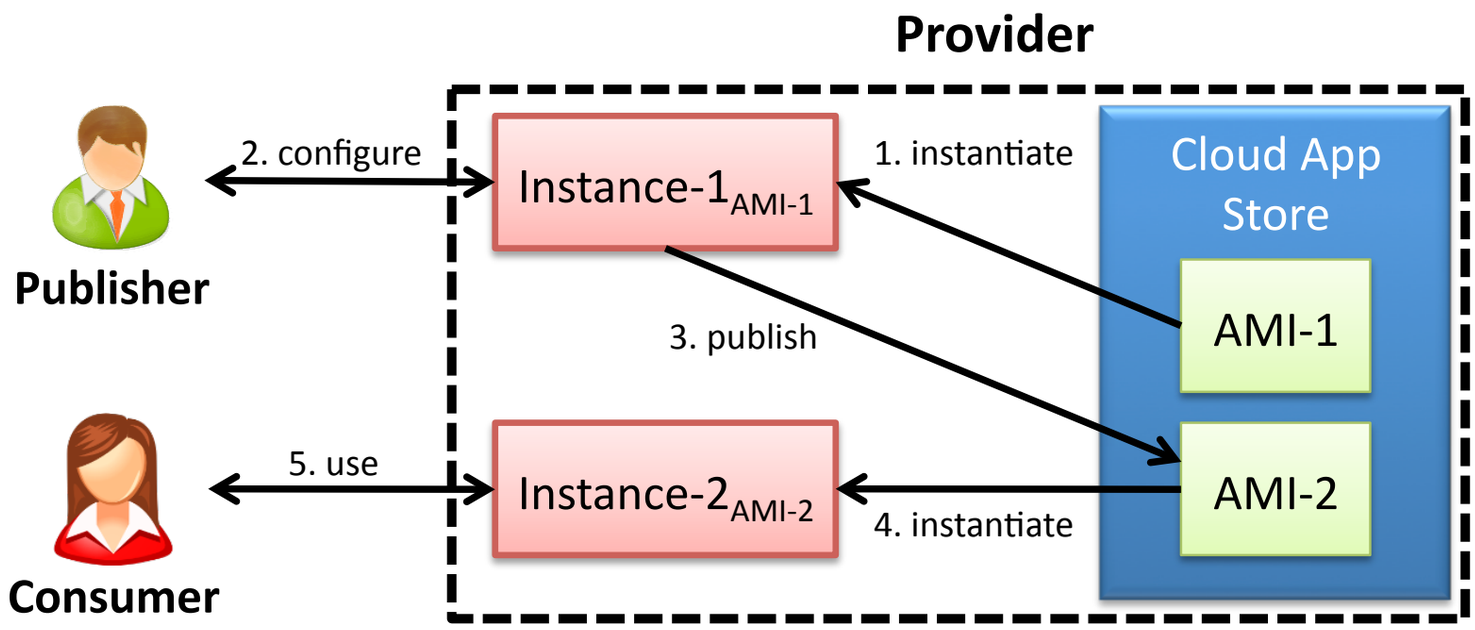
- New problem or new solution?
  - New challenges brought on by the cloud (plus old ones)
  - Utility could provide a foundation for solving such challenges

Cloudy Future

- Improve on data centers? On home computing?
  - Seems like a low bar

AmazonIA

Consumers use published instances *[CCS 2011]*



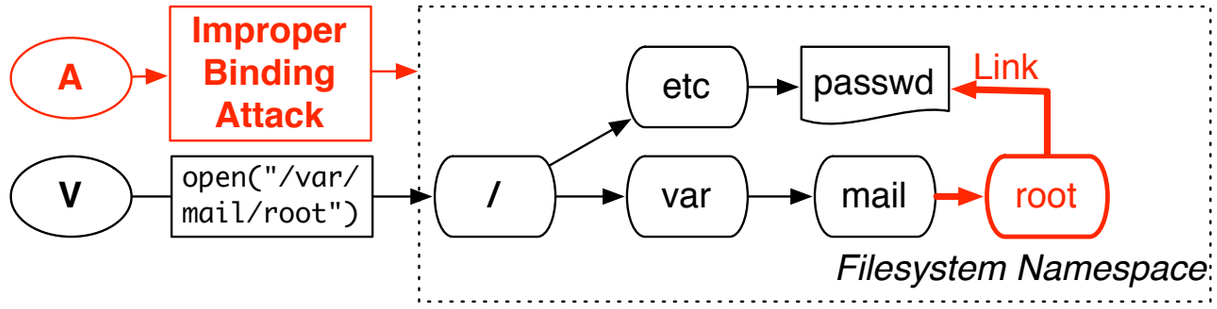
Instances may be flawed - have adversary-controlled public and private keys

Security Configuration

- Zillions of security-relevant configurations for instances
  - Firewalls
  - Mandatory access control
    - SELinux, AppArmor, TrustedBSD, Trusted Solaris, MIC
  - Discretionary access control
  - Application policies (e.g., Database, Apache)
  - Pluggable Authentication Modules (PAM)
  - Application configuration files
  - Application code enforces security
- Plus new configuration tasks for the cloud - e.g., storage

DAC Permission Diffs

Study of link traversal attacks *[USENIX Sec 2012a]*

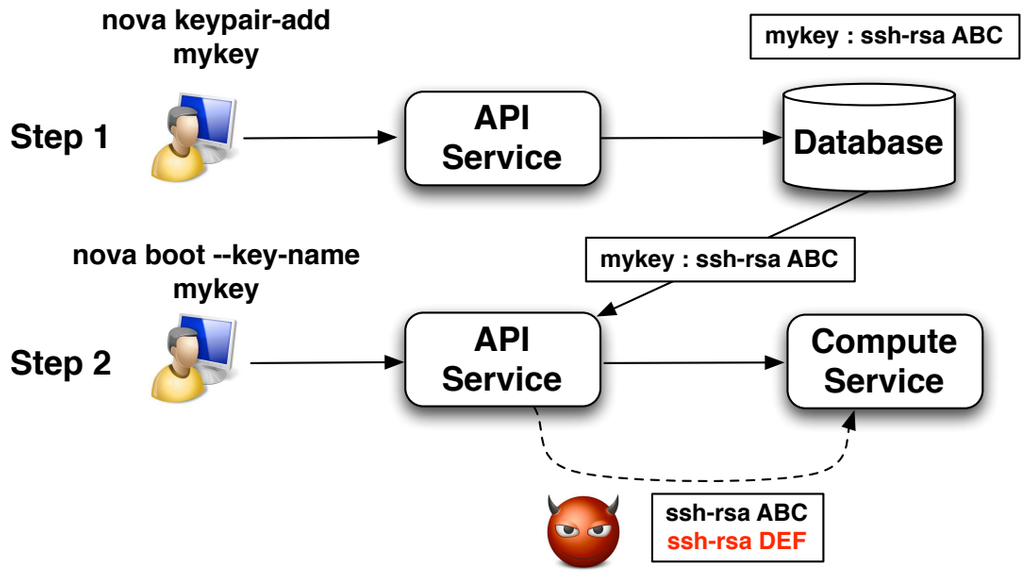


Permissions and vulnerabilities varied between two Linux distros

Program	Vuln. Entry	Priv. Escalation DAC: uid->uid	Distribution	Previously known
dbus-daemon	2	messagebus->root	Ubuntu	Unknown
landscape	4	landscape->root	Ubuntu	Unknown
Startup scripts (3)	4	various->root	Ubuntu	Unknown
mysql	2	mysql->root	Ubuntu	1 Known
mysql_upgrade	1	mysql->root	Ubuntu	Unknown
tomcat script	2	tomcat6->root	Ubuntu	Known
lightdm	1	*->root	Ubuntu	Unknown
bluetooth-applet	1	*->user	Ubuntu	Unknown
java (openjdk)	1	*->user	Both	Known
zeitgeist-daemon	1	*->user	Both	Unknown
mountall	1	*->root	Ubuntu	Unknown
mailutils	1	mail->root	Ubuntu	Unknown
bsd-mailx	1	mail->root	Fedora	Unknown
cupsd	1	cups->root	Fedora	Known
abrt-server	1	abrt->root	Fedora	Unknown
yum	1	sync->root	Fedora	Unknown
x2gostartagent	1	*->user	Extra	Unknown
19 Programs	**26**			**21 Unknown**

Cloud Service Vulnerabilities

- Vulnerabilities have been found in cloud services
  - E.g., OpenStack identity service, web interface, and API service
- Adversaries who compromise such services may launch a variety of attacks
  - E.g., Key Injection Attack



Insiders

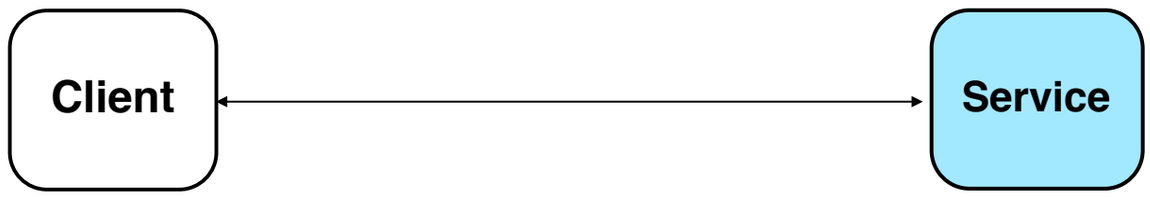
- Although the vendor may have a good reputation, not every employee may

Side Channels

- Shared infrastructure leads to visibility for others
  - You can't monitor, but others can
- Get Off My Cloud - Ristenpart et al. *[CCS 2009]*
  - Caches (Memory)
  - Devices (I/O)
  - CPU
  - Scheduling
- Ari Juels -- "Many of the security implications of the cloud stem from tenants entrusting computing resources to a third party that they controlled in the past."
- Not really going to discuss this further

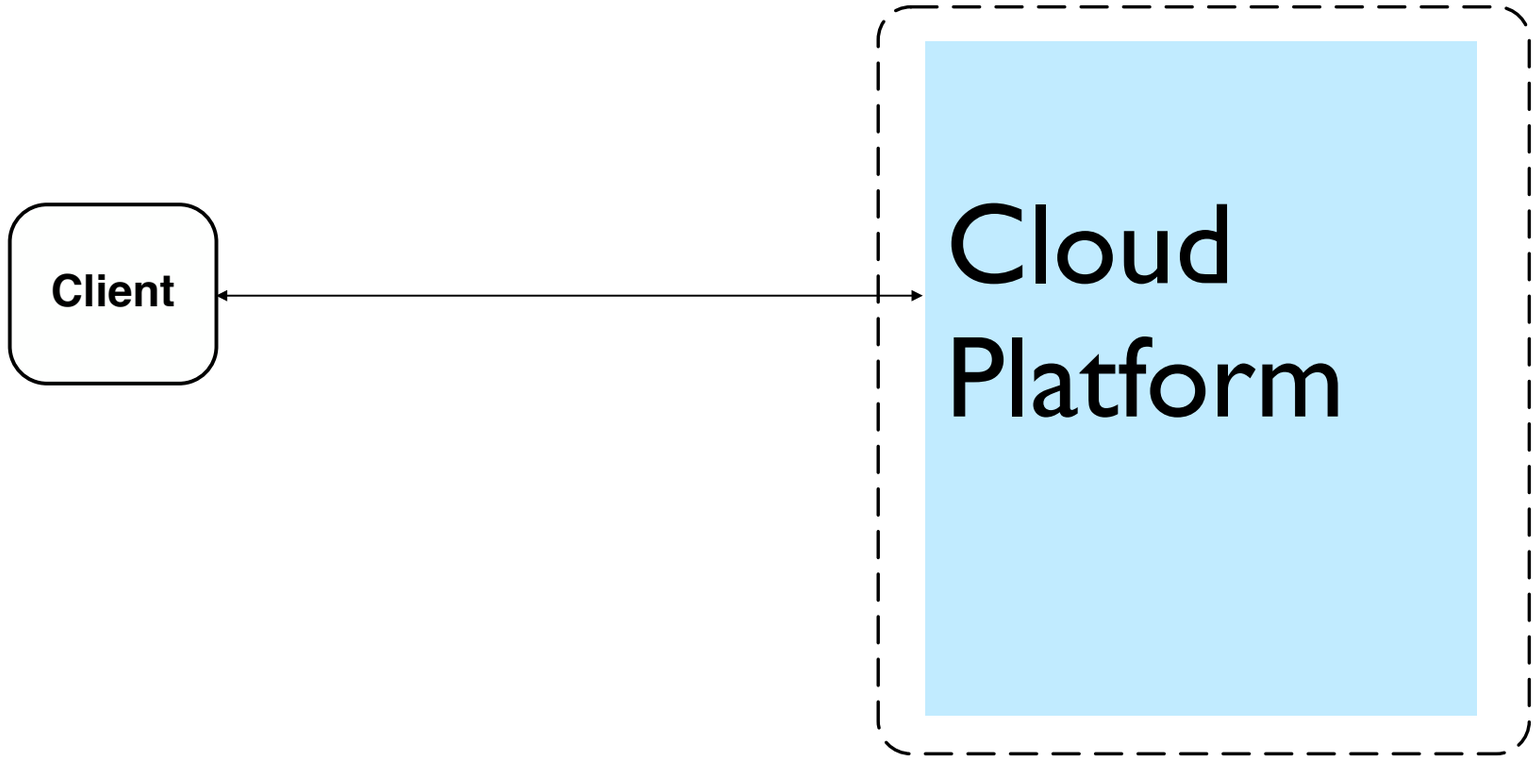
Cloud Complexity

- Cloud environments add further challenges
  - Opaque, Complex, Dynamic



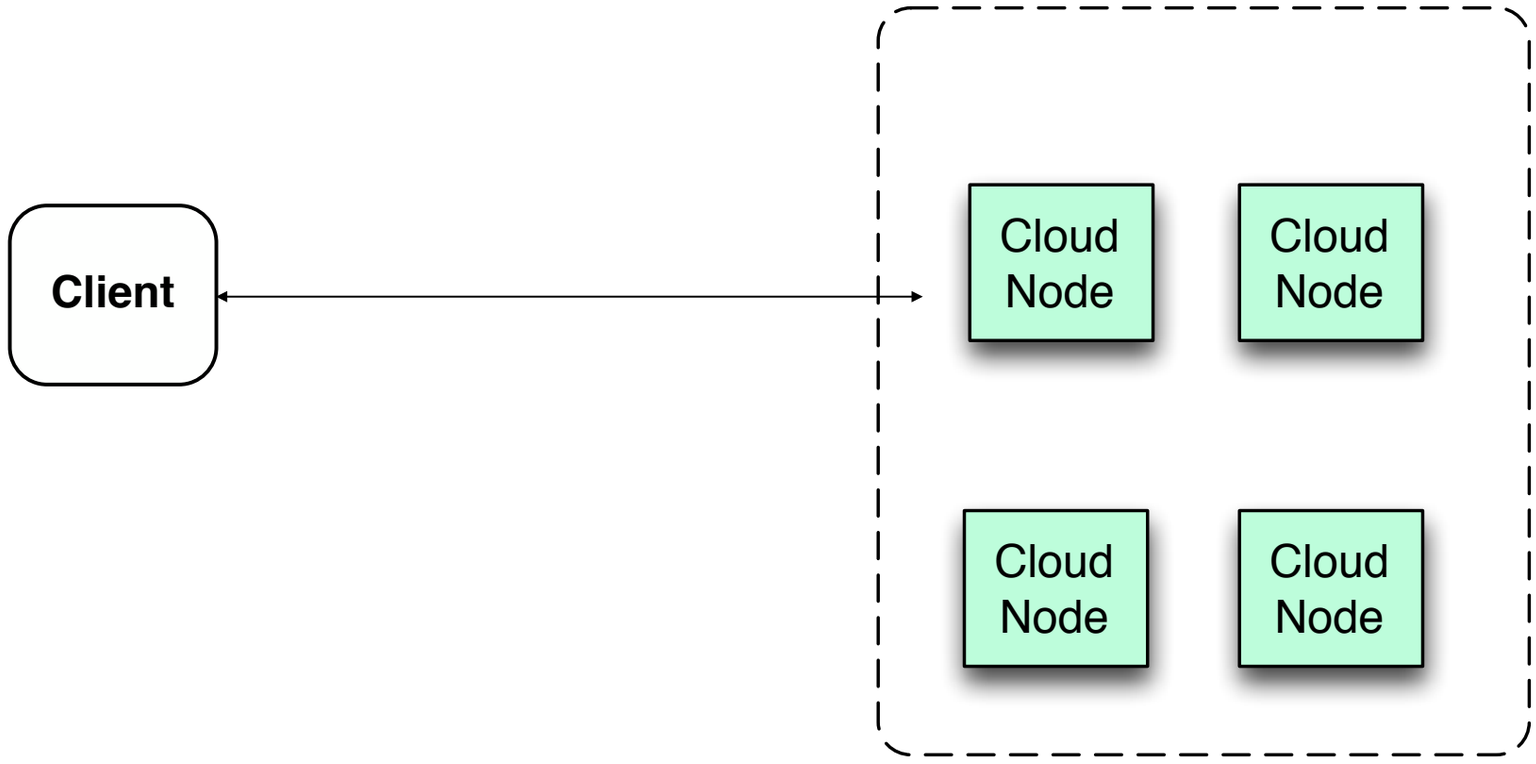
Cloud Complexity

- Cloud environments add further challenges
  - Opaque, Complex, Dynamic



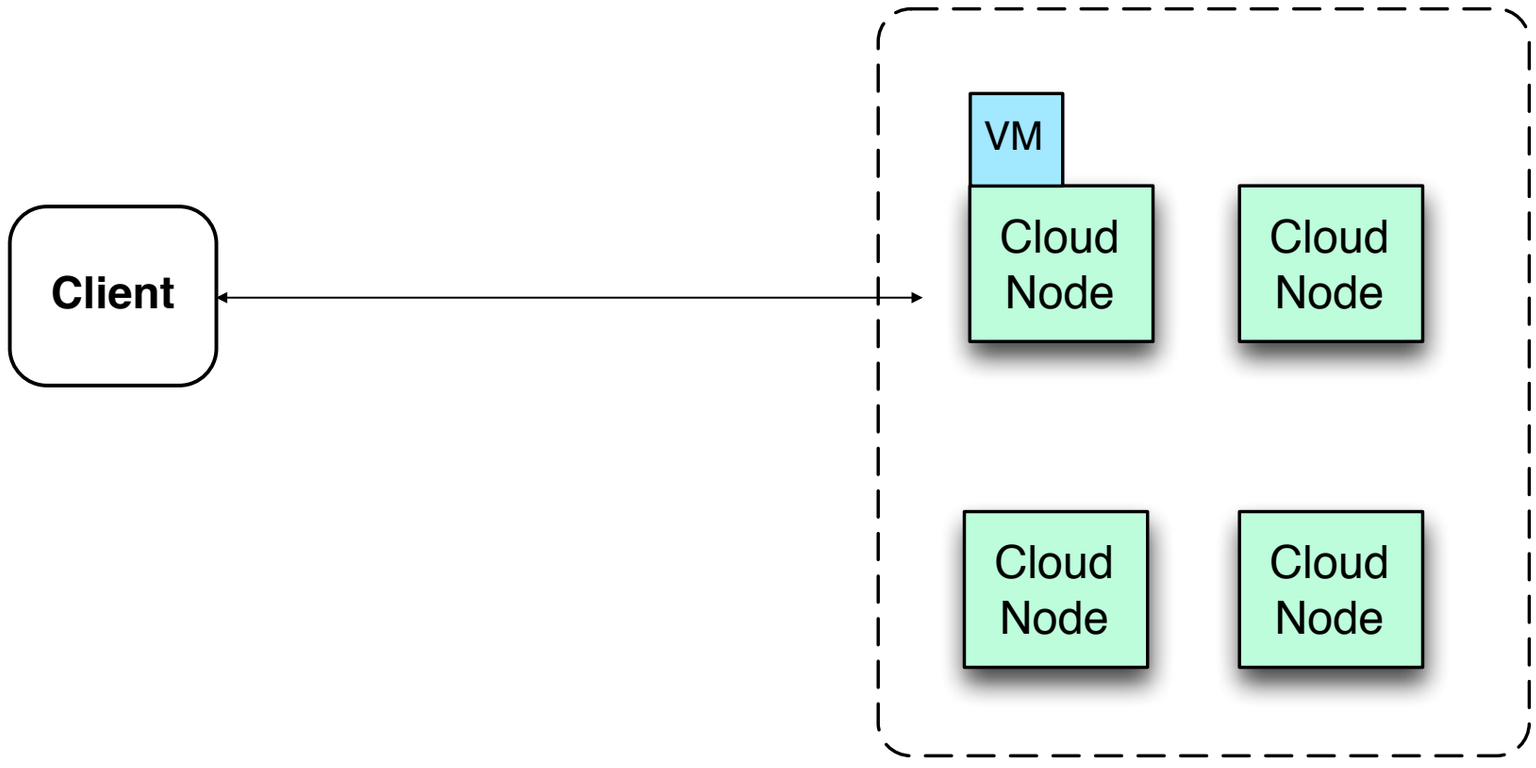
Cloud Complexity

- Cloud environments add further challenges
  - Opaque, Complex, Dynamic



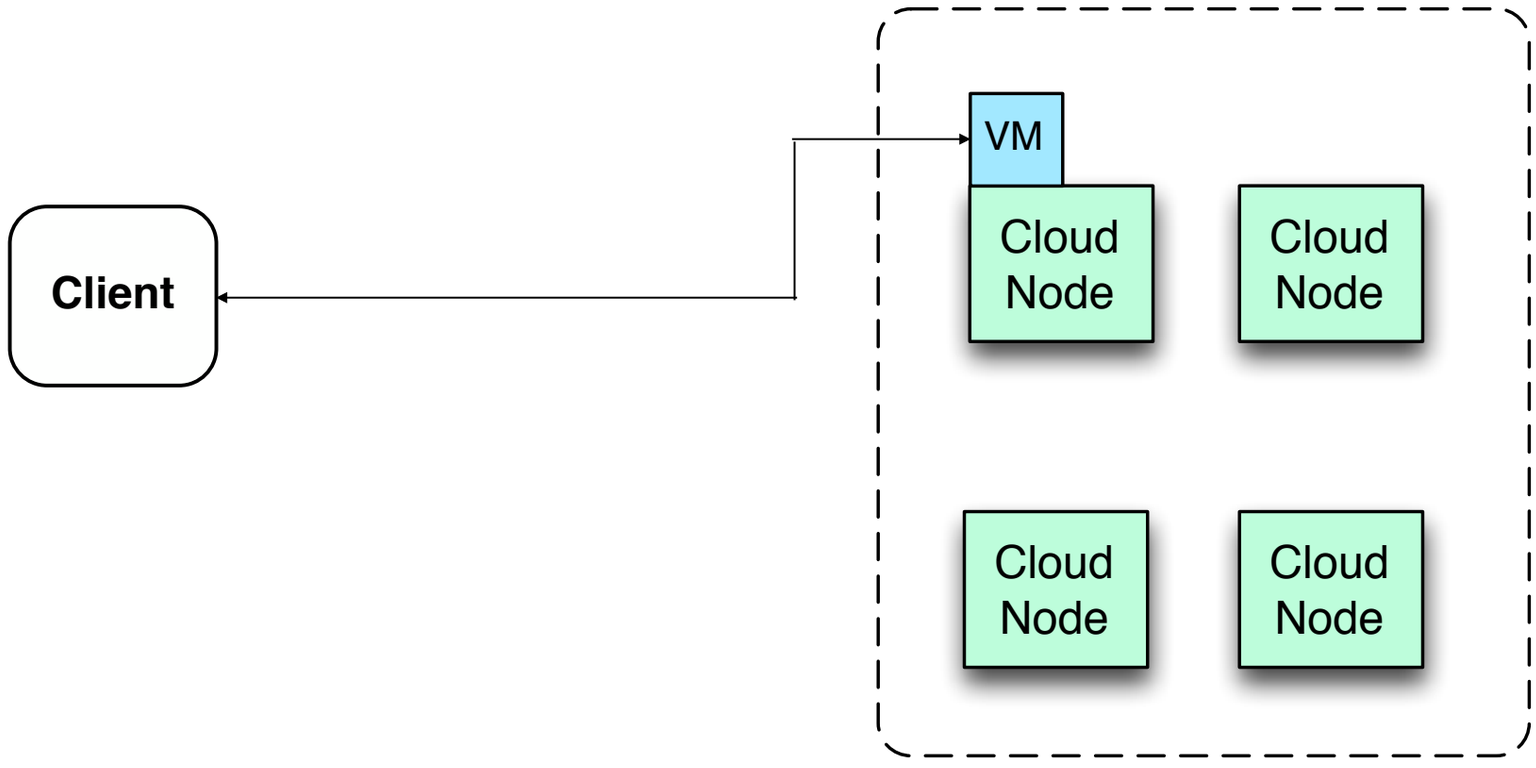
Cloud Complexity

- Cloud environments add further challenges
  - Opaque, Complex, Dynamic



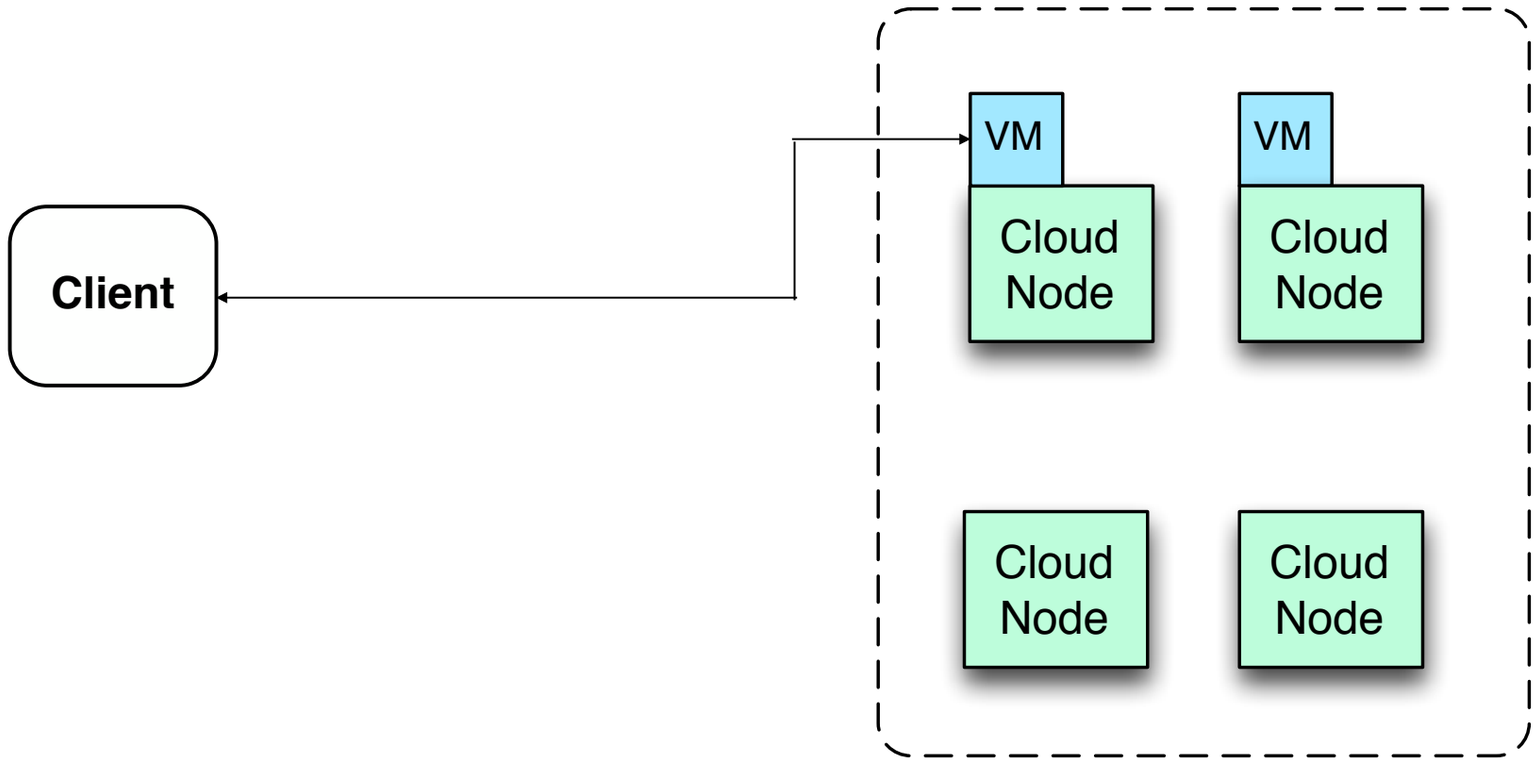
Cloud Complexity

- Cloud environments add further challenges
  - Opaque, Complex, Dynamic



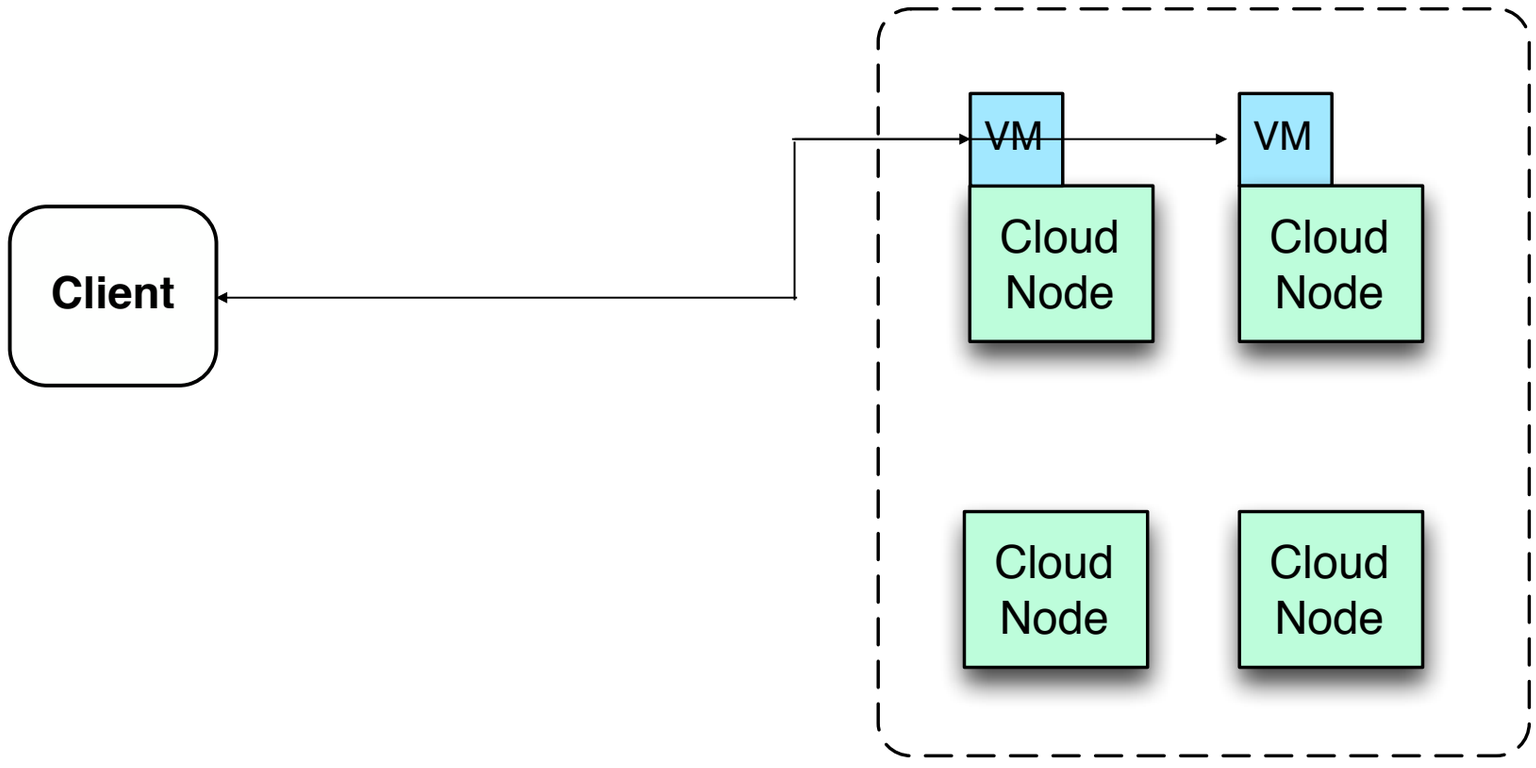
Cloud Complexity

- Cloud environments add further challenges
  - Opaque, Complex, Dynamic



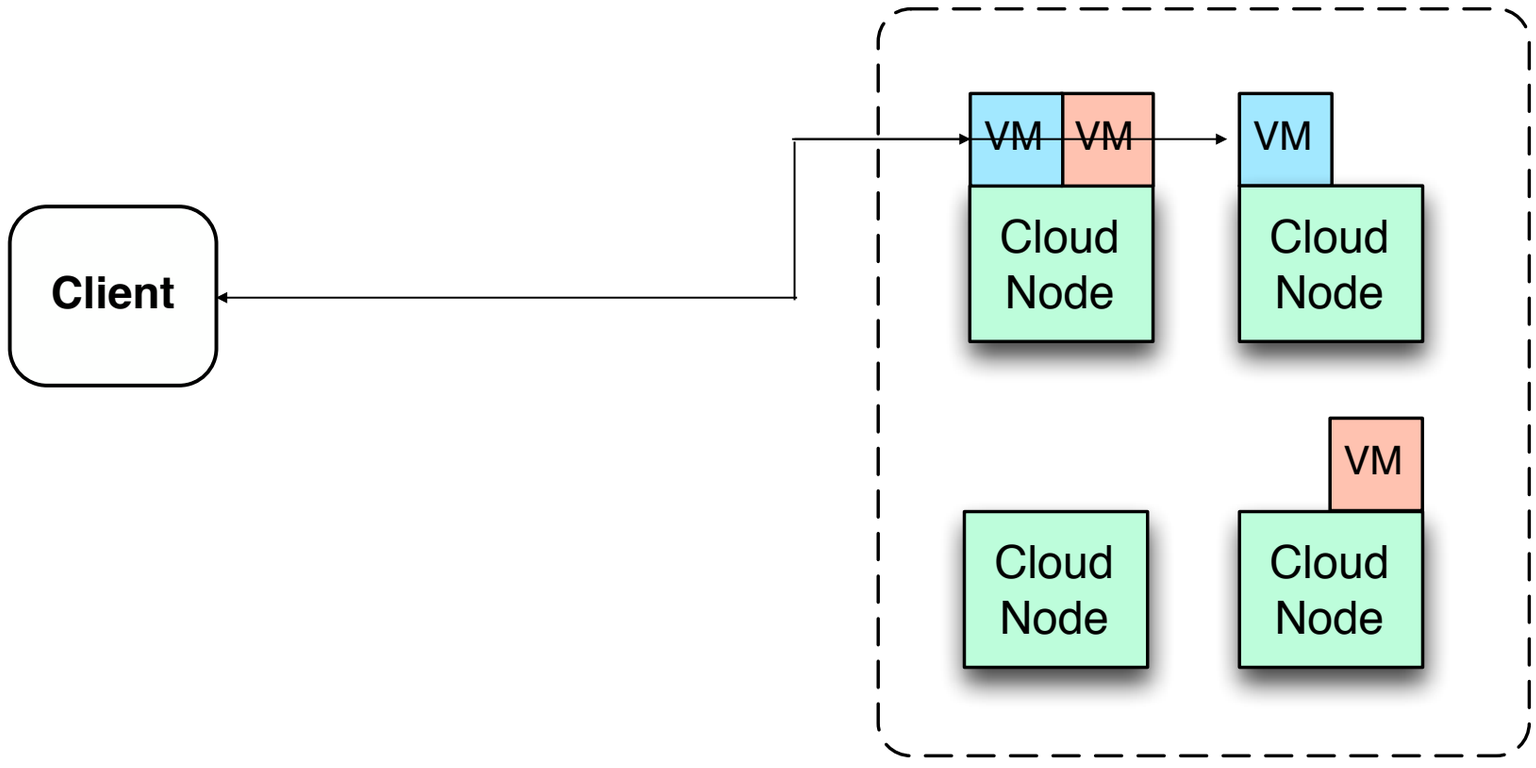
Cloud Complexity

- Cloud environments add further challenges
  - Opaque, Complex, Dynamic



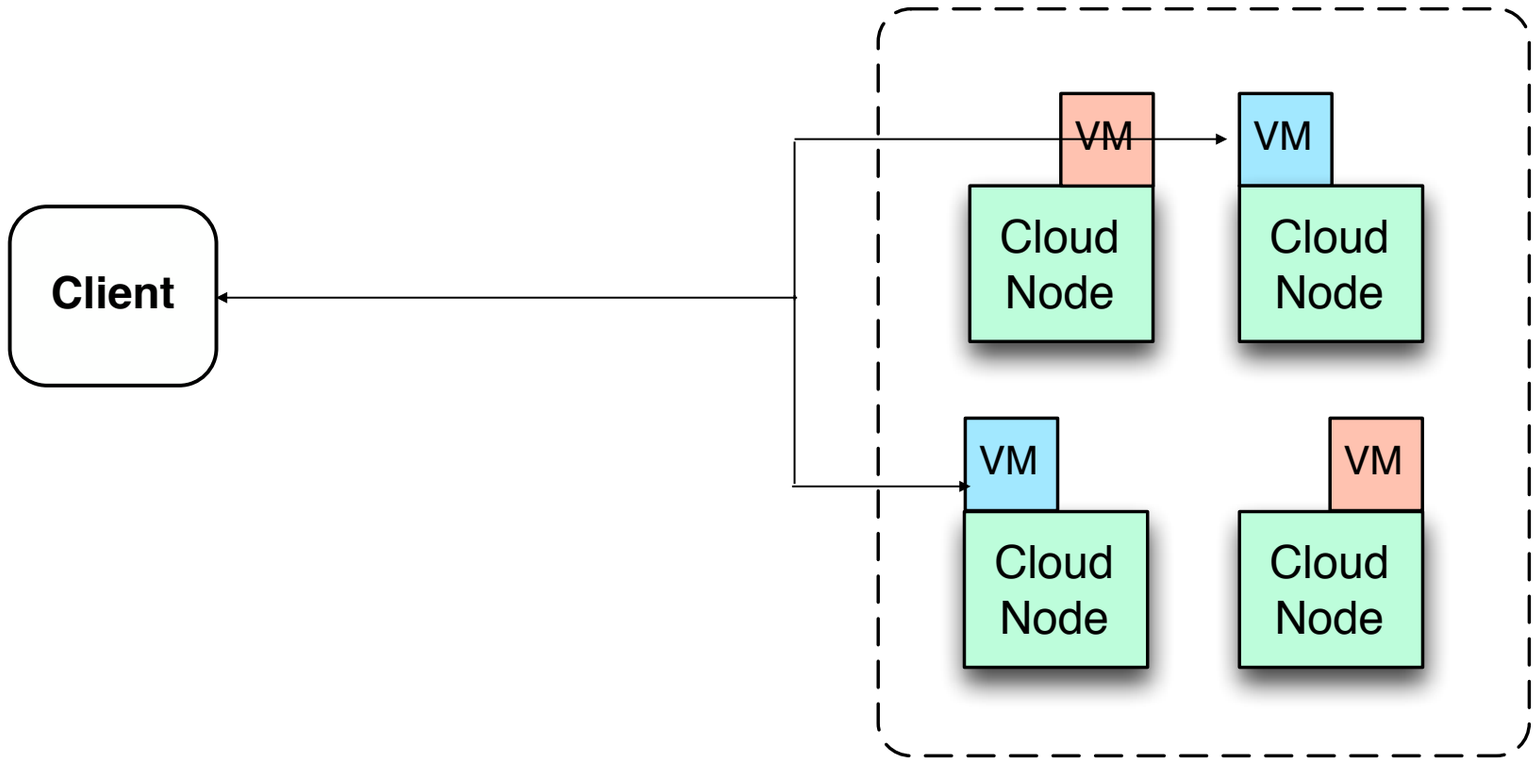
Cloud Complexity

- Cloud environments add further challenges
  - Opaque, Complex, Dynamic

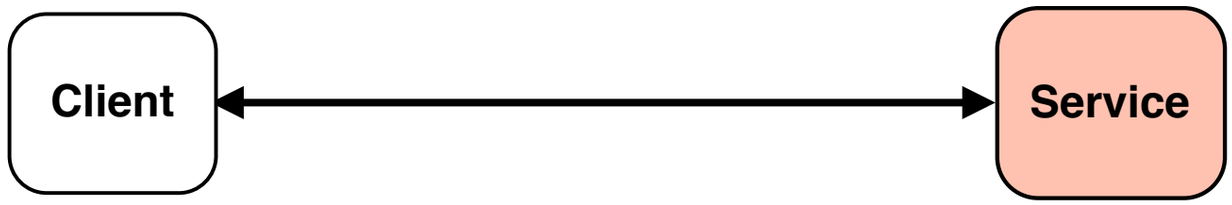


Cloud Complexity

- Cloud environments add further challenges
  - Opaque, Complex, Dynamic

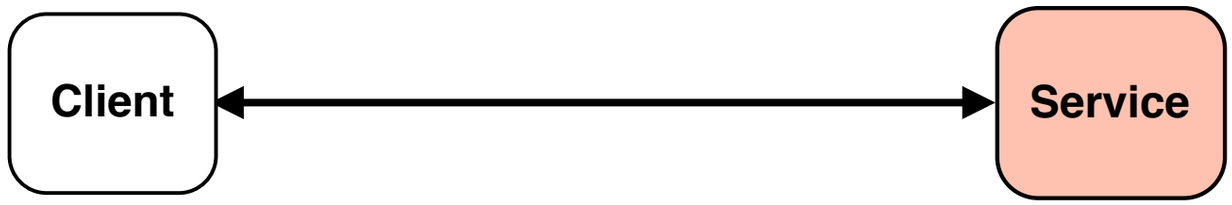


Blind Trust



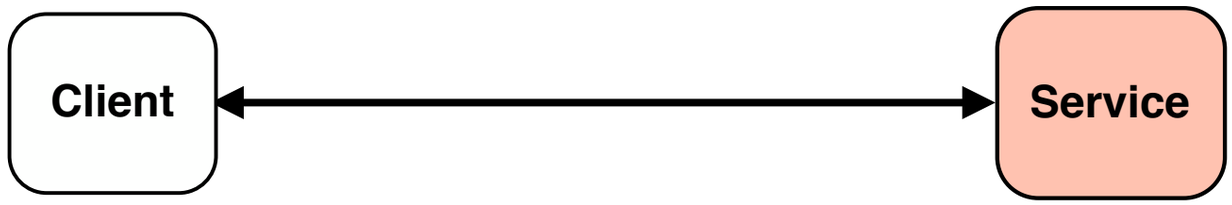
Blind Trust

- Cannot see what your services are doing



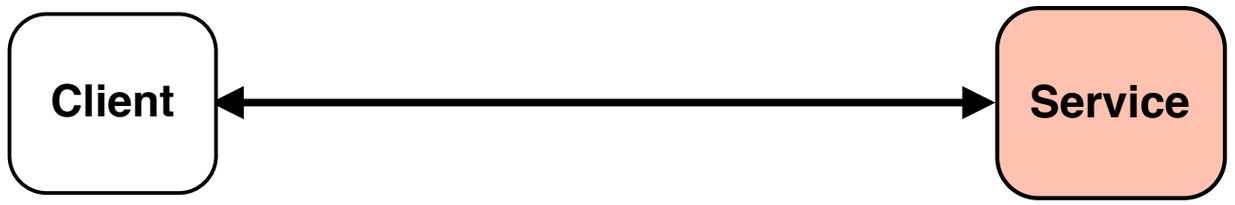
Blind Trust

- Cannot see what your services are doing



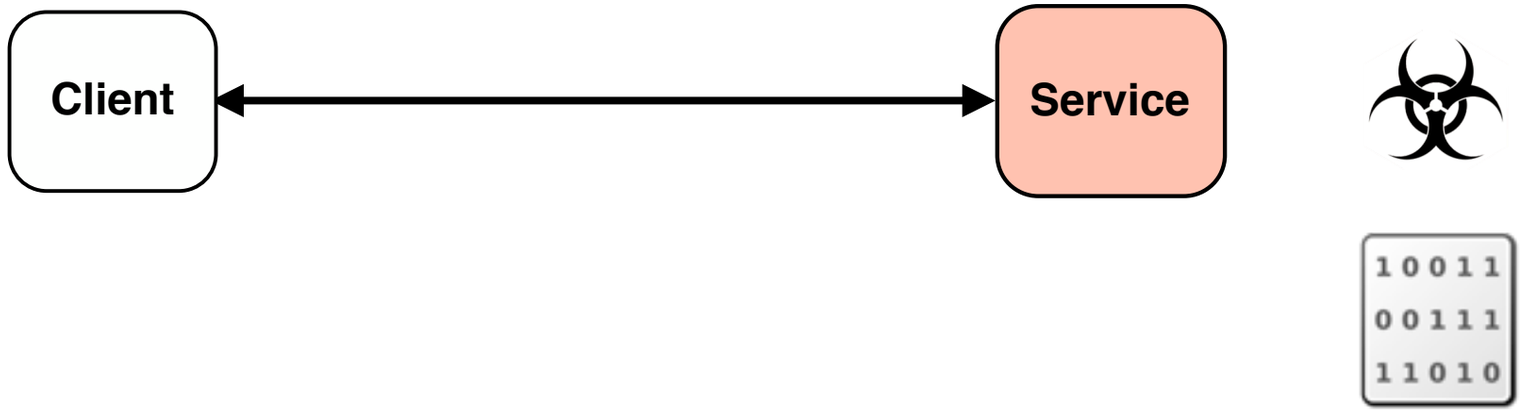
Blind Trust

- Cannot see what your services are doing



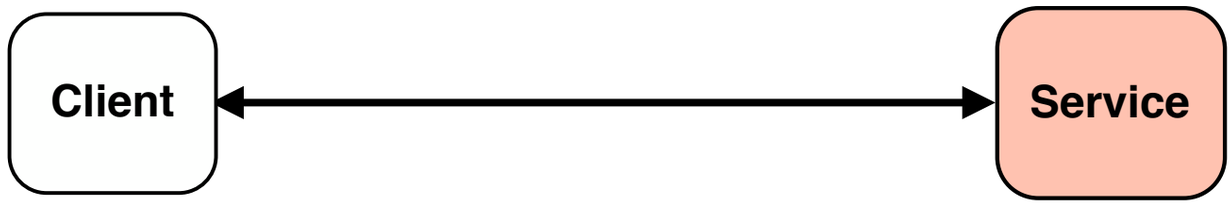
Blind Trust

- Cannot see what your services are doing



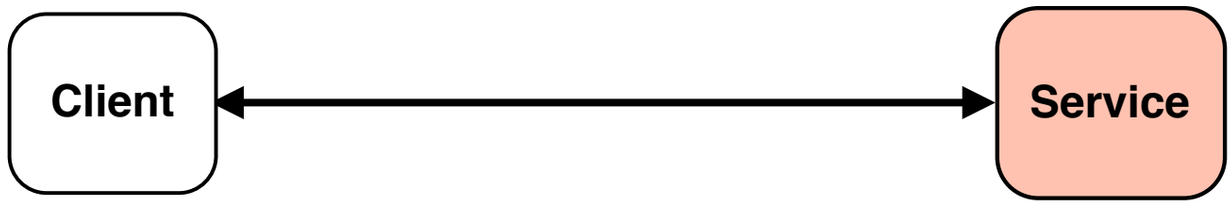
Blind Trust

- Cannot see what your services are doing



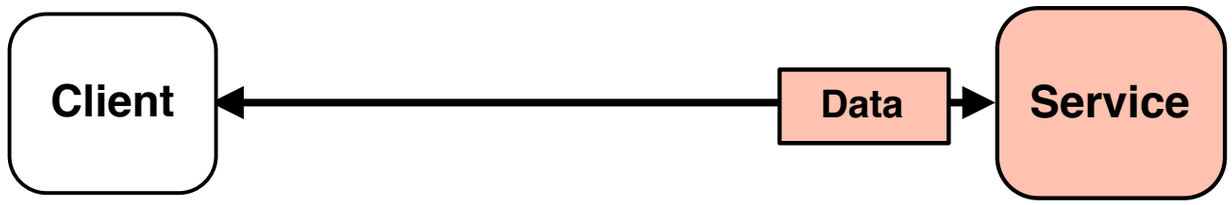
Blind Trust

- Cannot see what your services are doing
  - Are the data produced from a trustworthy system?



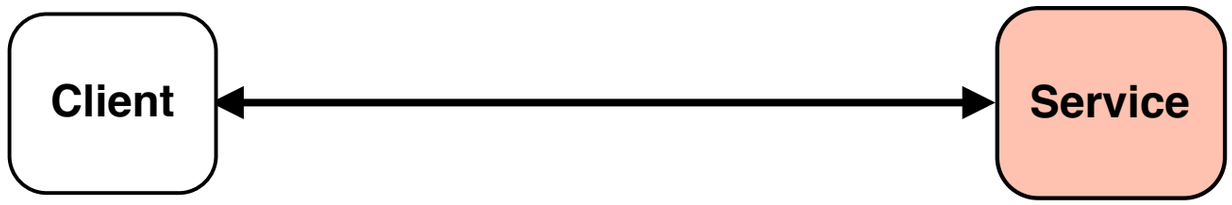
Blind Trust

- Cannot see what your services are doing
  - Are the data produced from a trustworthy system?



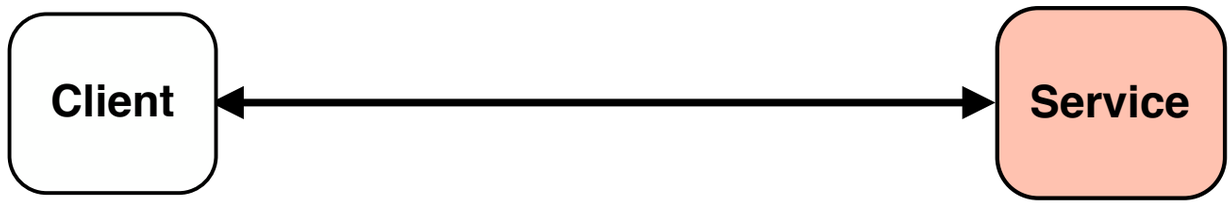
Blind Trust

- Cannot see what your services are doing
  - Are the data produced from a trustworthy system?



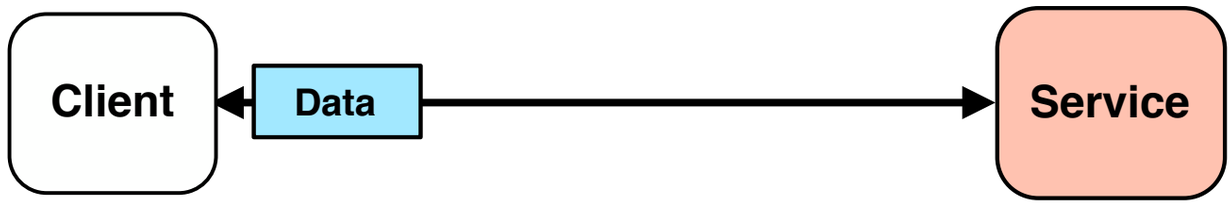
Blind Trust

- Cannot see what your services are doing
  - Are the data produced from a trustworthy system?
  - Will the service protect client data?



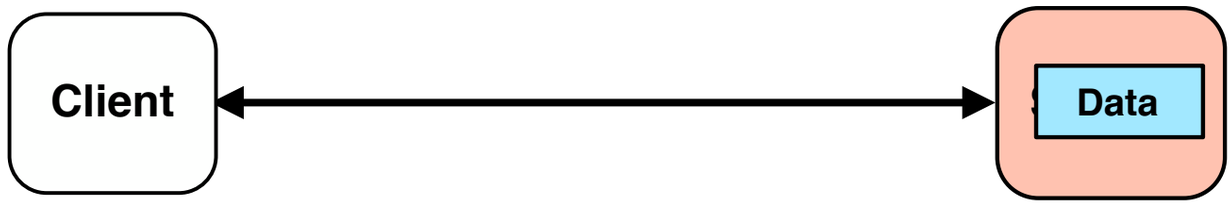
Blind Trust

- Cannot see what your services are doing
  - Are the data produced from a trustworthy system?
  - Will the service protect client data?



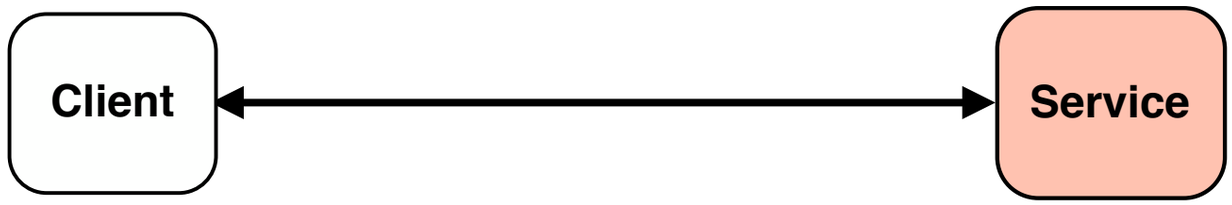
Blind Trust

- Cannot see what your services are doing
  - Are the data produced from a trustworthy system?
  - Will the service protect client data?



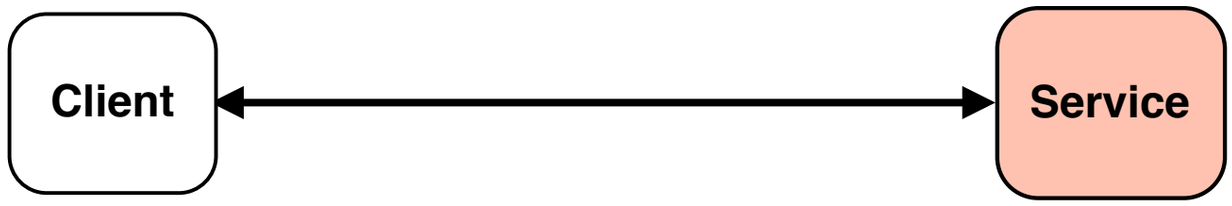
Blind Trust

- Cannot see what your services are doing
  - Are the data produced from a trustworthy system?
  - Will the service protect client data?



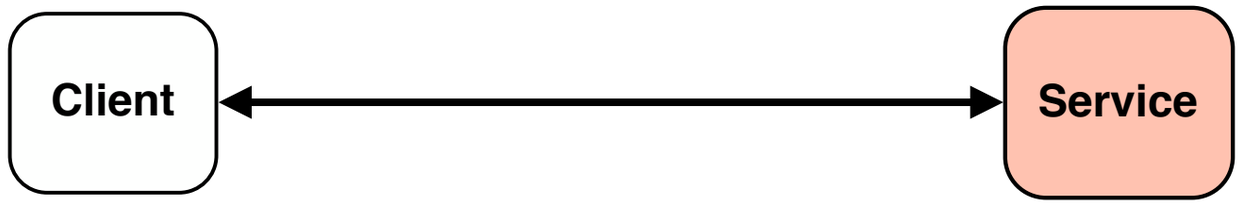
Blind Trust

- Cannot see what your services are doing
  - Are the data produced from a trustworthy system?
  - Will the service protect client data?



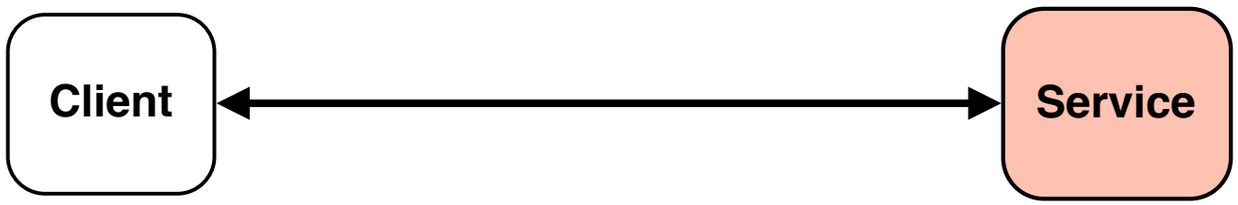
Need to verify the service's integrity

Integrity Monitoring



Integrity Monitoring

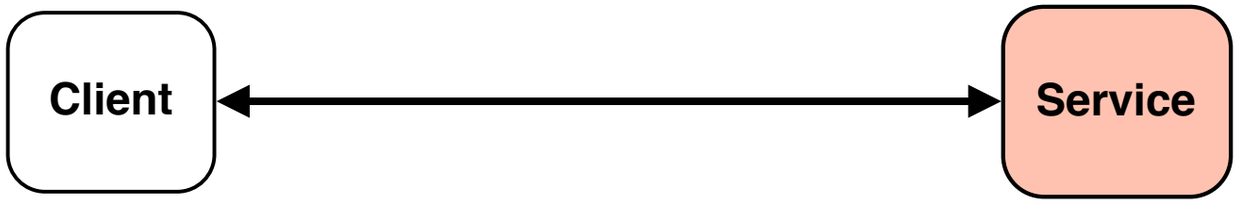
- We need to monitor the system's integrity
  - Define criteria for a trustworthy system
  - Verify the system meets those criteria



Integrity Monitoring

- We need to monitor the system's integrity
  - Define criteria for a trustworthy system
  - Verify the system meets those criteria

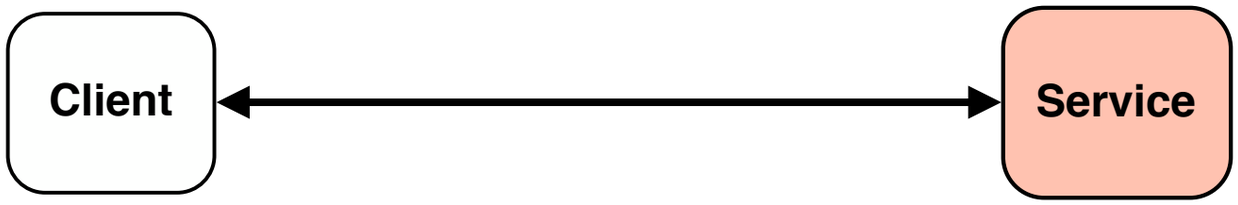
Integrity Criteria



Integrity Monitoring

- We need to monitor the system's integrity
  - Define criteria for a trustworthy system
  - Verify the system meets those criteria

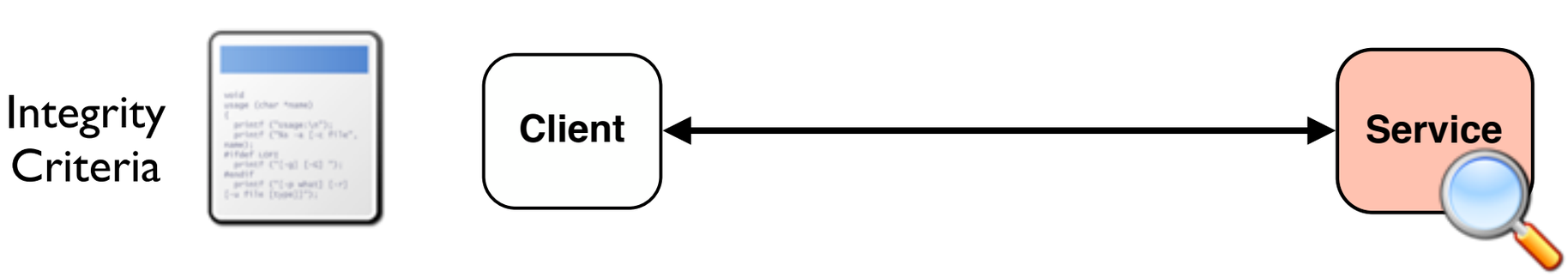
Integrity Criteria



- How do we measure the system's configuration?

Integrity Monitoring

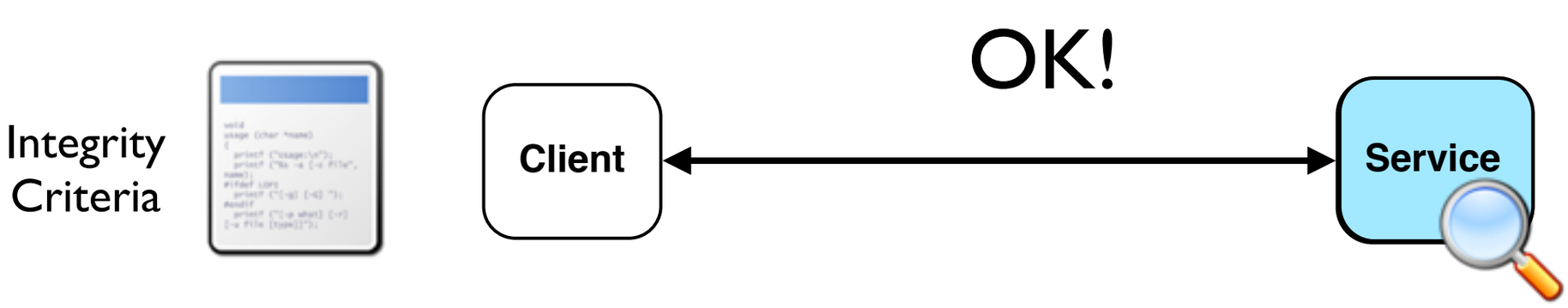
- We need to monitor the system's integrity
  - Define criteria for a trustworthy system
  - Verify the system meets those criteria



- How do we measure the system's configuration?

Integrity Monitoring

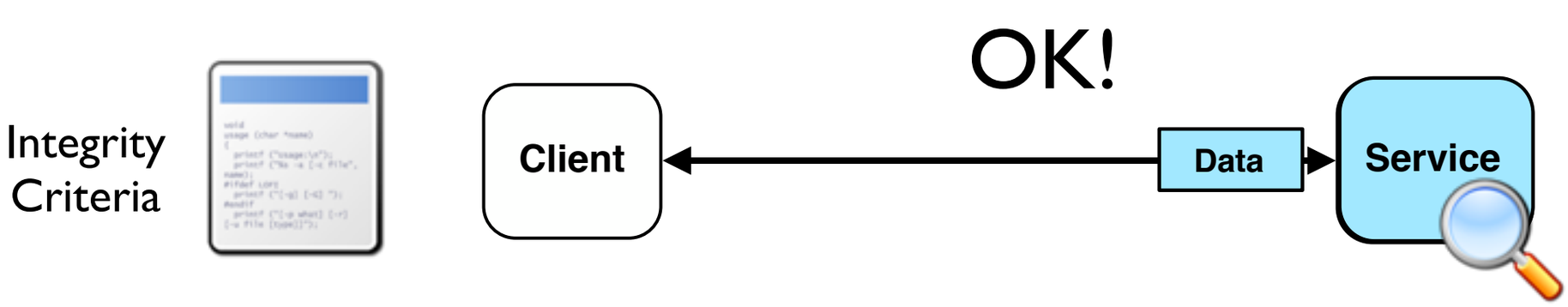
- We need to monitor the system's integrity
  - Define criteria for a trustworthy system
  - Verify the system meets those criteria



- How do we measure the system's configuration?

Integrity Monitoring

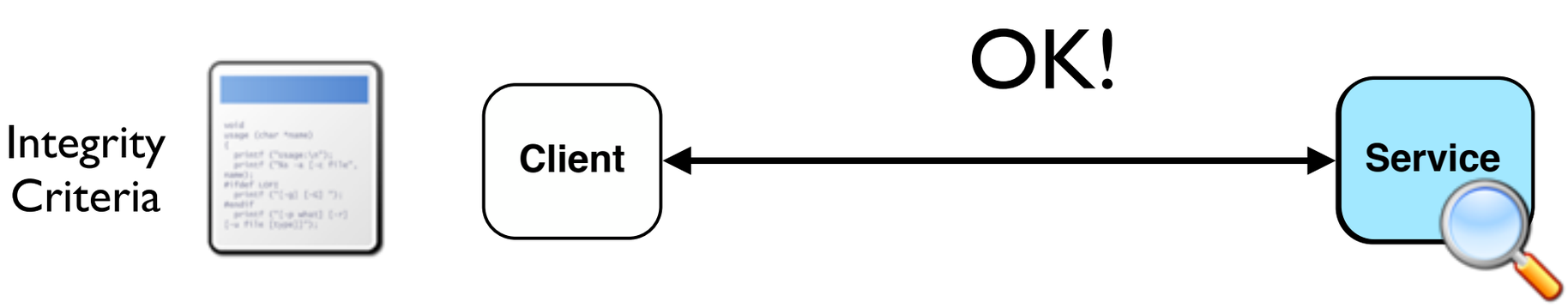
- We need to monitor the system's integrity
  - Define criteria for a trustworthy system
  - Verify the system meets those criteria



- How do we measure the system's configuration?

Integrity Monitoring

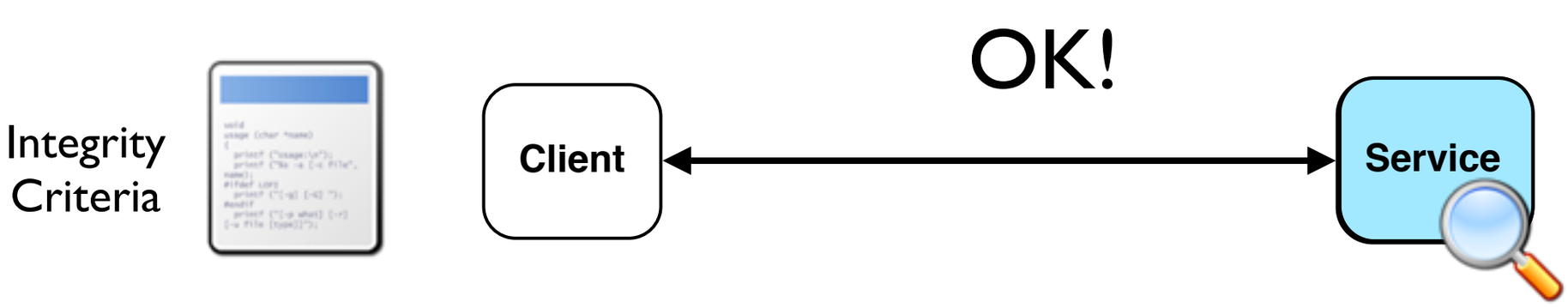
- We need to monitor the system's integrity
  - Define criteria for a trustworthy system
  - Verify the system meets those criteria



- How do we measure the system's configuration?

Integrity Monitoring

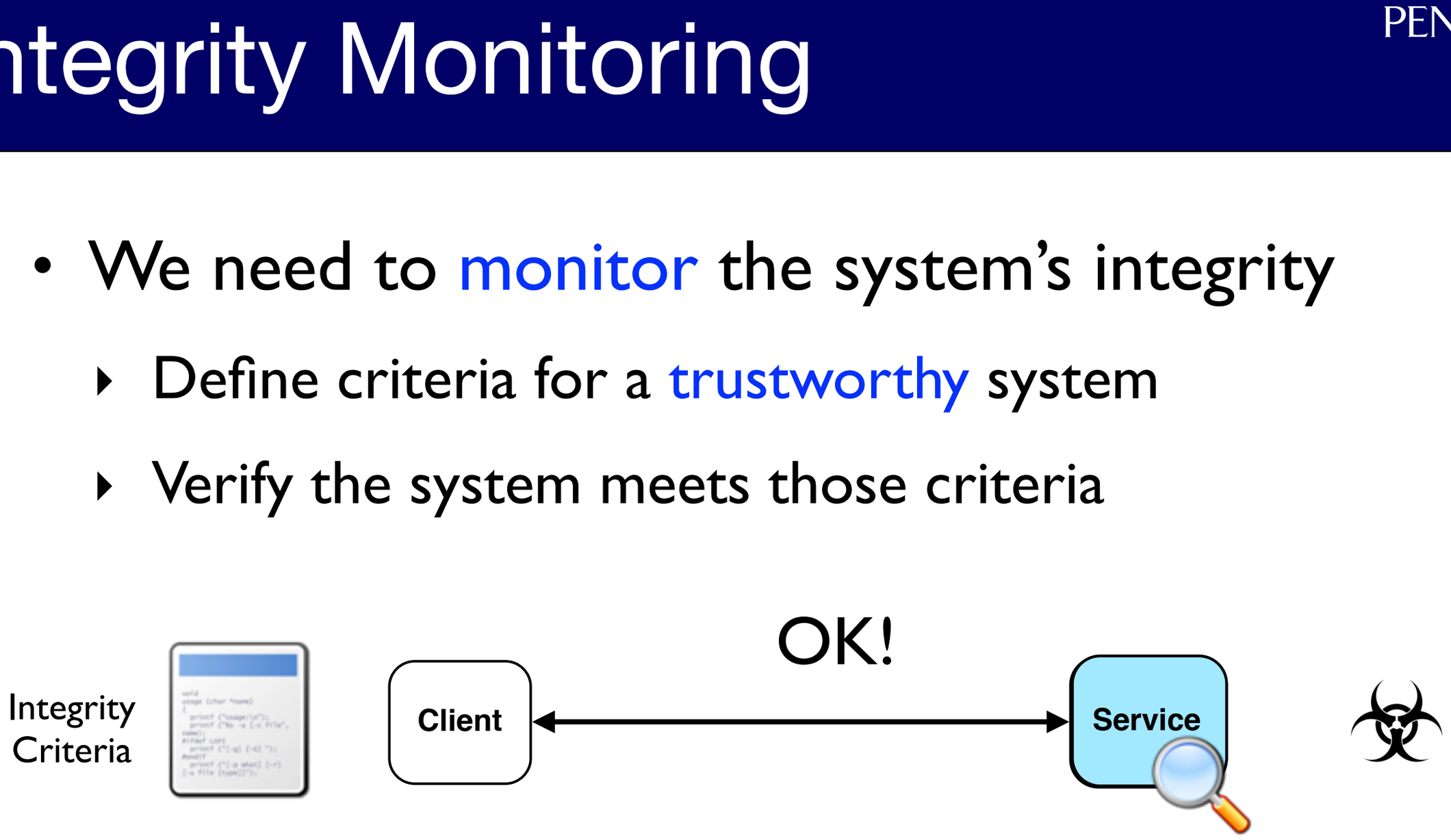
- We need to monitor the system's integrity
  - Define criteria for a trustworthy system
  - Verify the system meets those criteria



- How do we measure the system's configuration?
- Will the service remain trustworthy?

Integrity Monitoring

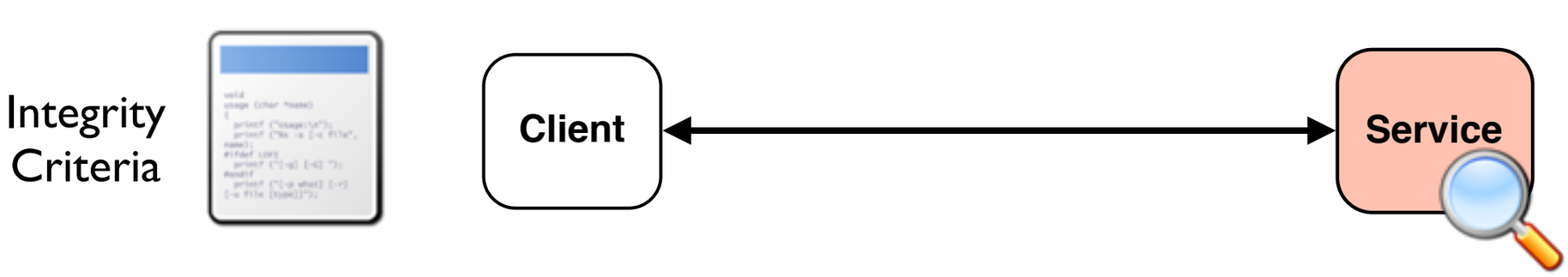
- We need to monitor the system's integrity
  - Define criteria for a trustworthy system
  - Verify the system meets those criteria



- How do we measure the system's configuration?
- Will the service remain trustworthy?

Integrity Monitoring

- We need to monitor the system's integrity
  - Define criteria for a trustworthy system
  - Verify the system meets those criteria



- How do we measure the system's configuration?
- Will the service remain trustworthy?

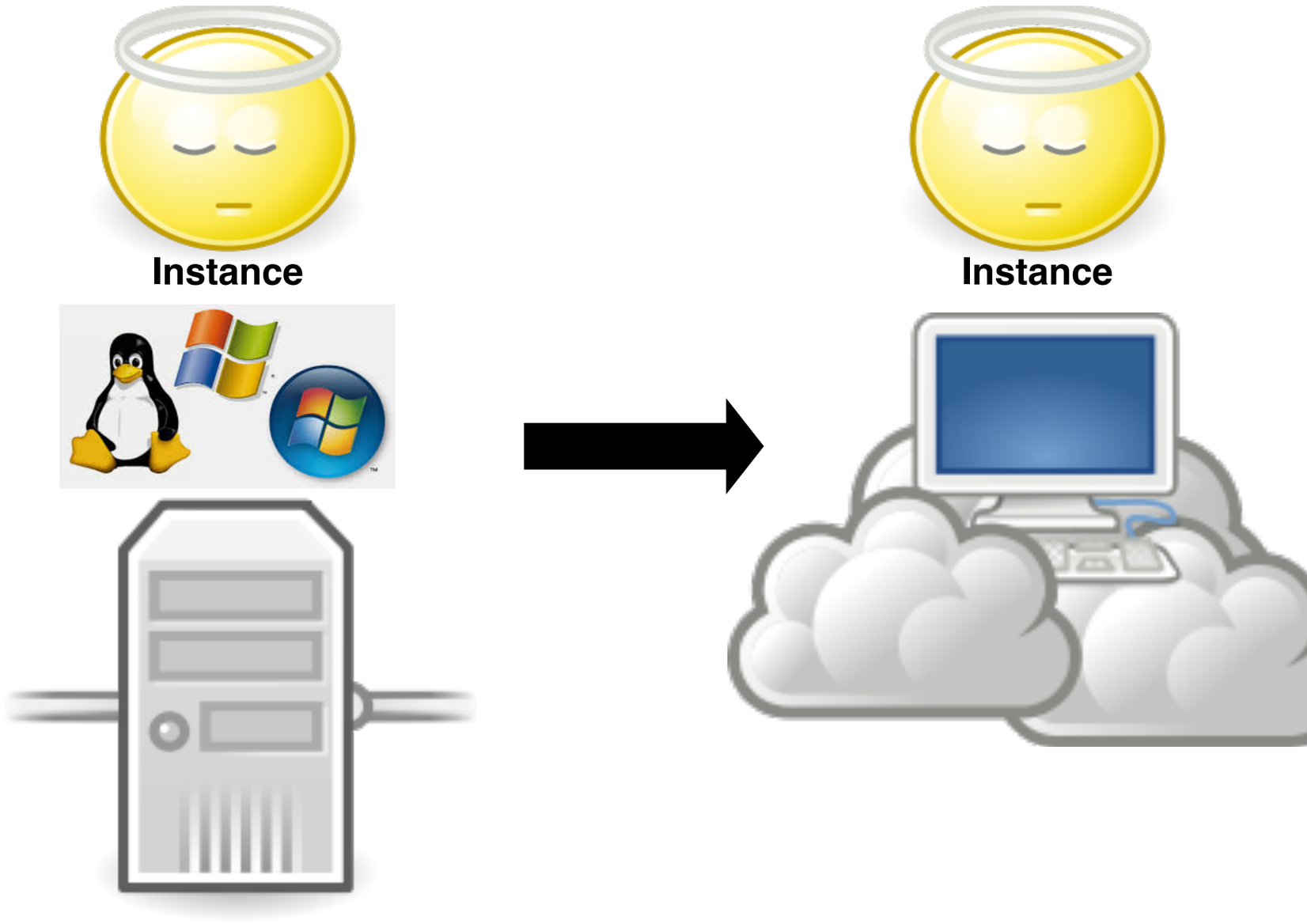
Research Goal

Construct a mechanism to monitor cloud-hosted services to ensure they satisfy a broad range of customer-specified requirements with minimal verification overhead

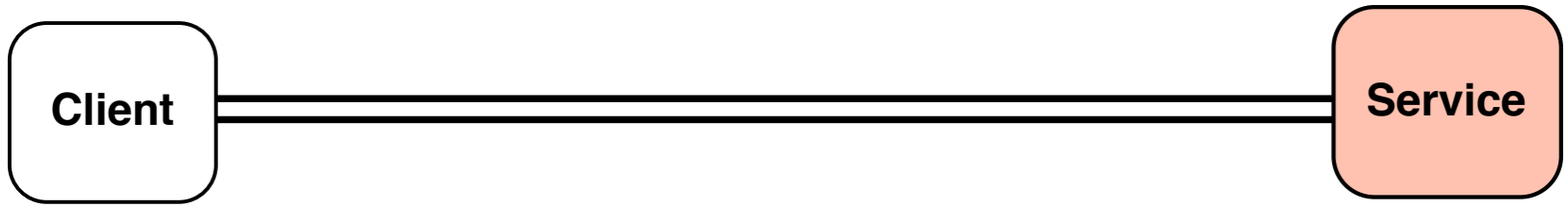
From Data Center to Cloud

PENNSTATE 1855

Goal: achieve/improve data center monitoring

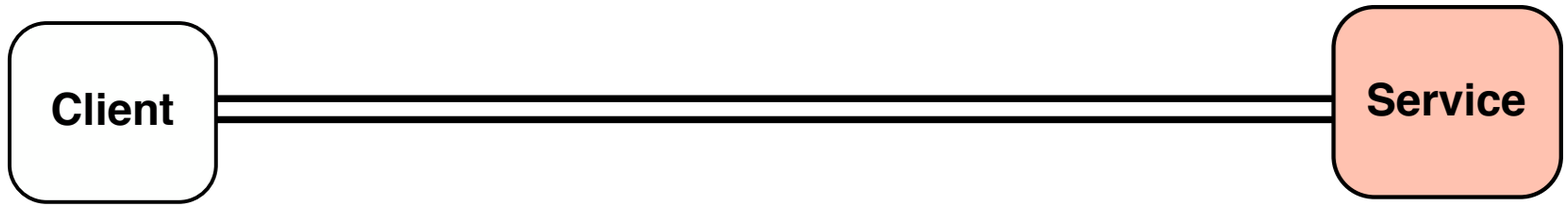


Integrity Monitor Concept



Integrity Monitor Concept

- Integrity monitor similar to a reference monitor
  - Mediate access to service based on integrity criteria



Integrity Monitor Concept

- Integrity monitor similar to a reference monitor
  - Mediate access to service based on integrity criteria

Integrity Monitor Concept

- Integrity monitor similar to a reference monitor
  - Mediate access to service based on integrity criteria

Integrity Monitor Concept

- Integrity monitor similar to a reference monitor
  - Mediate access to service based on integrity criteria

Integrity Monitor Concept

- Integrity monitor similar to a reference monitor
  - Mediate access to service based on integrity criteria

Integrity Monitor Concept

- Integrity monitor similar to a reference monitor
  - Mediate access to service based on integrity criteria

Integrity Monitor Concept

- Integrity monitor similar to a reference monitor
  - Mediate access to service based on integrity criteria

Integrity Monitor Concept

- Integrity monitor similar to a reference monitor
  - Mediate access to service based on integrity criteria

Integrity Monitor Concept

- Integrity monitor similar to a reference monitor
  - Mediate access to service based on integrity criteria

Integrity Monitor Concept

- Integrity monitor similar to a reference monitor
  - Mediate access to service based on integrity criteria

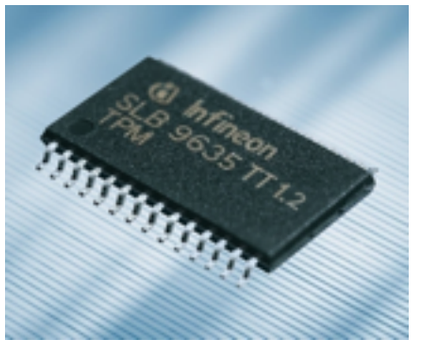
- Challenges
  - Where do we measure integrity-relevant events?
  - How do we verify ongoing integrity?
  - How can we deploy this in a cloud environment?

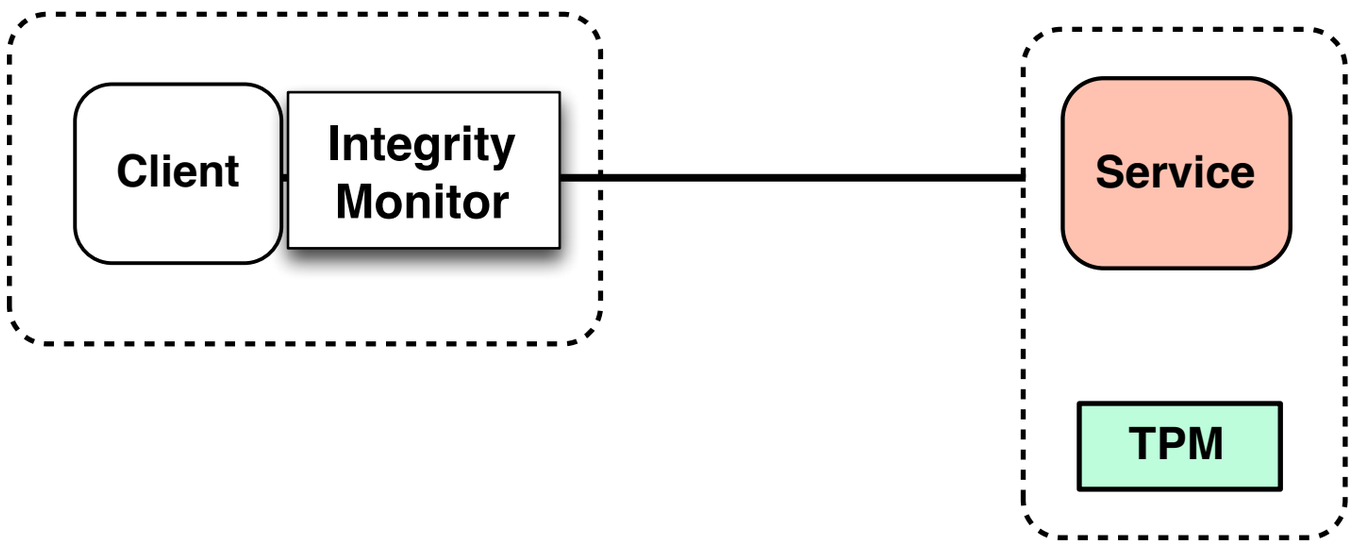
Measurement Techniques

- System management tools
  - Nessus, Nagios, Ganglia
- Challenge-based verification
  - Genuinity, Pioneer, Viper
- Hardware-based attestation
  - IMA, PRIMA, LIM, TNC, BIND, Flicker, Terra, Trustvisor, ...

TCG Remote Attestation

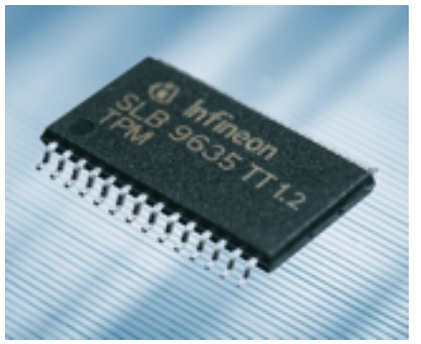
- Trusted Platform Module (TPM)
  - PCRs store event measurements
  - Protected key pair uniquely identifies platform
  - Enables remote attestation of recorded events

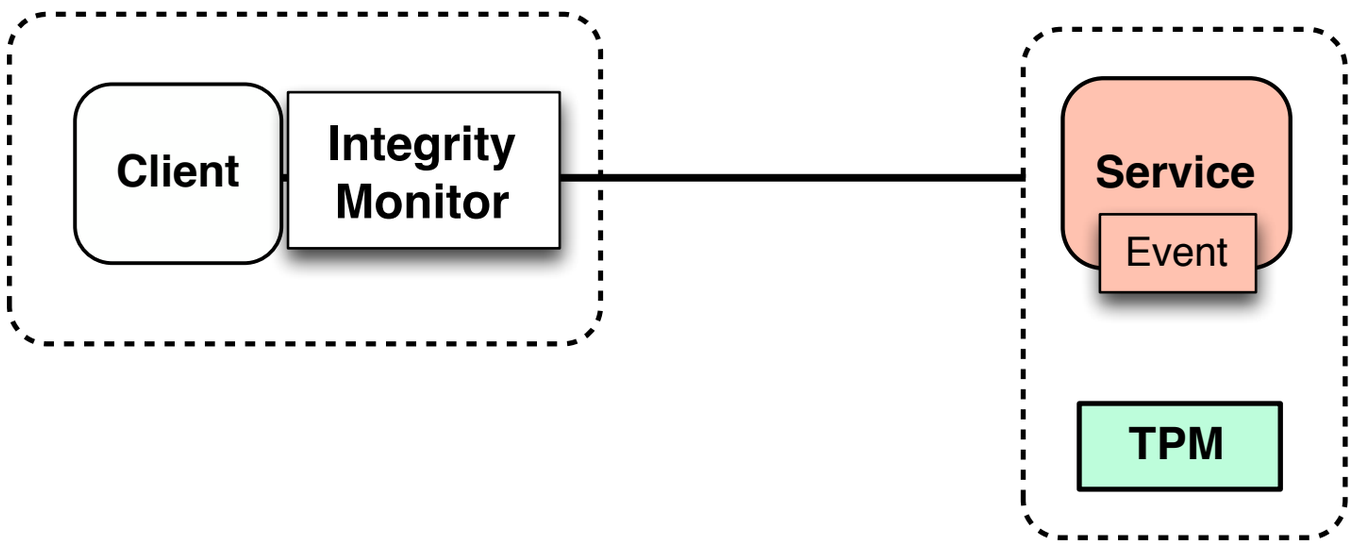




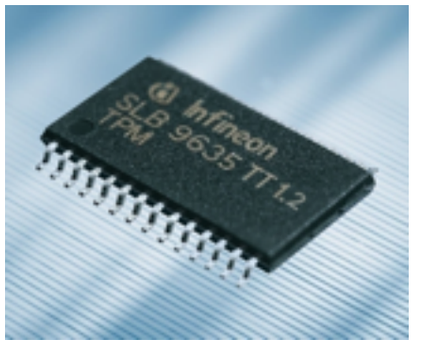
TCG Remote Attestation

- Trusted Platform Module (TPM)
  - PCRs store event measurements
  - Protected key pair uniquely identifies platform
  - Enables remote attestation of recorded events

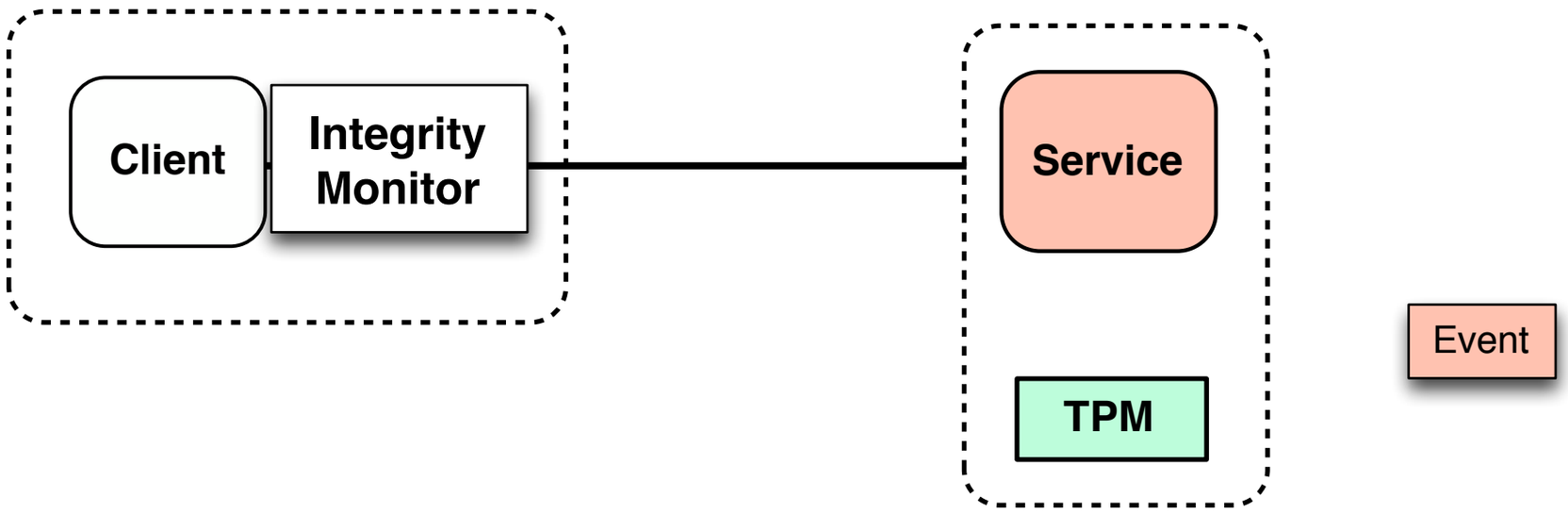




TCG Remote Attestation

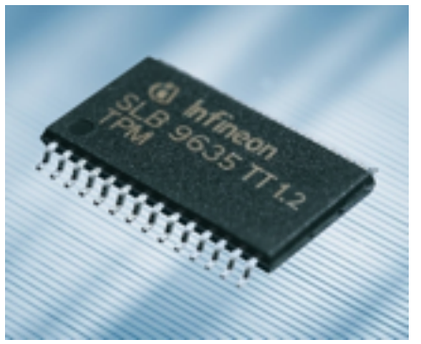


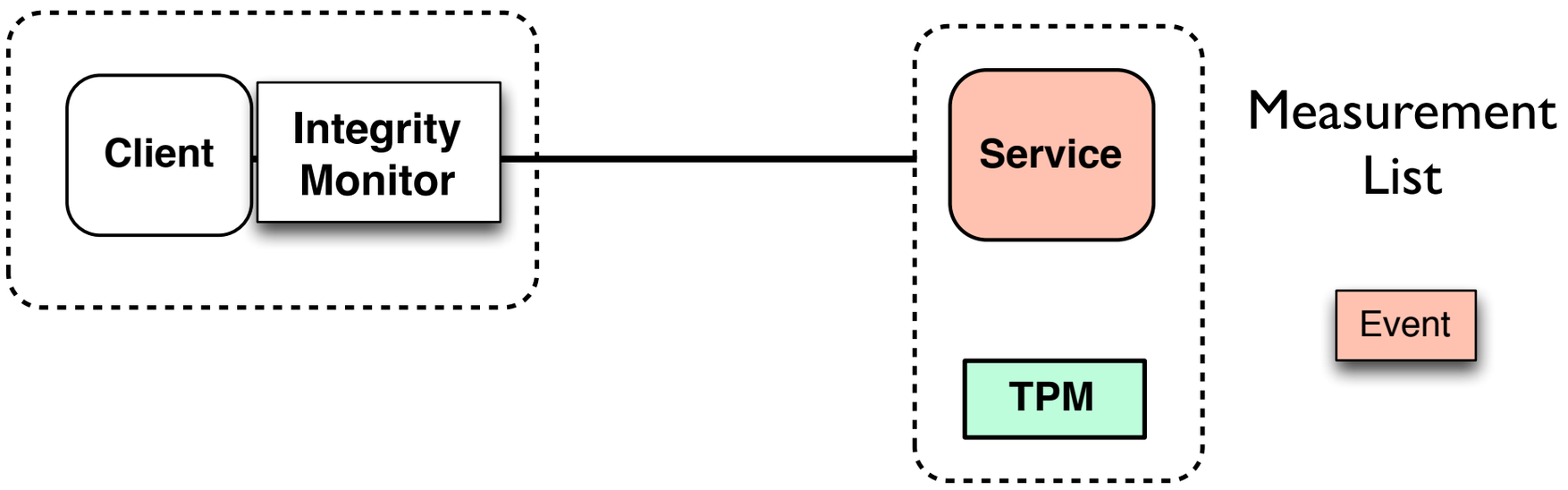
- Trusted Platform Module (TPM)
  - PCRs store event measurements
  - Protected key pair uniquely identifies platform
  - Enables remote attestation of recorded events



TCG Remote Attestation

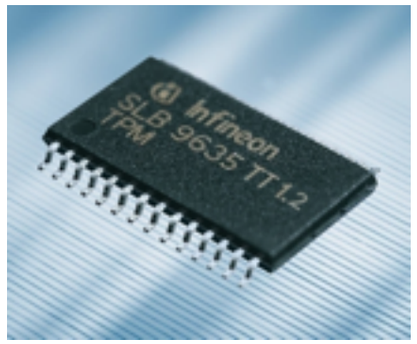
- Trusted Platform Module (TPM)
  - PCRs store event measurements
  - Protected key pair uniquely identifies platform
  - Enables remote attestation of recorded events

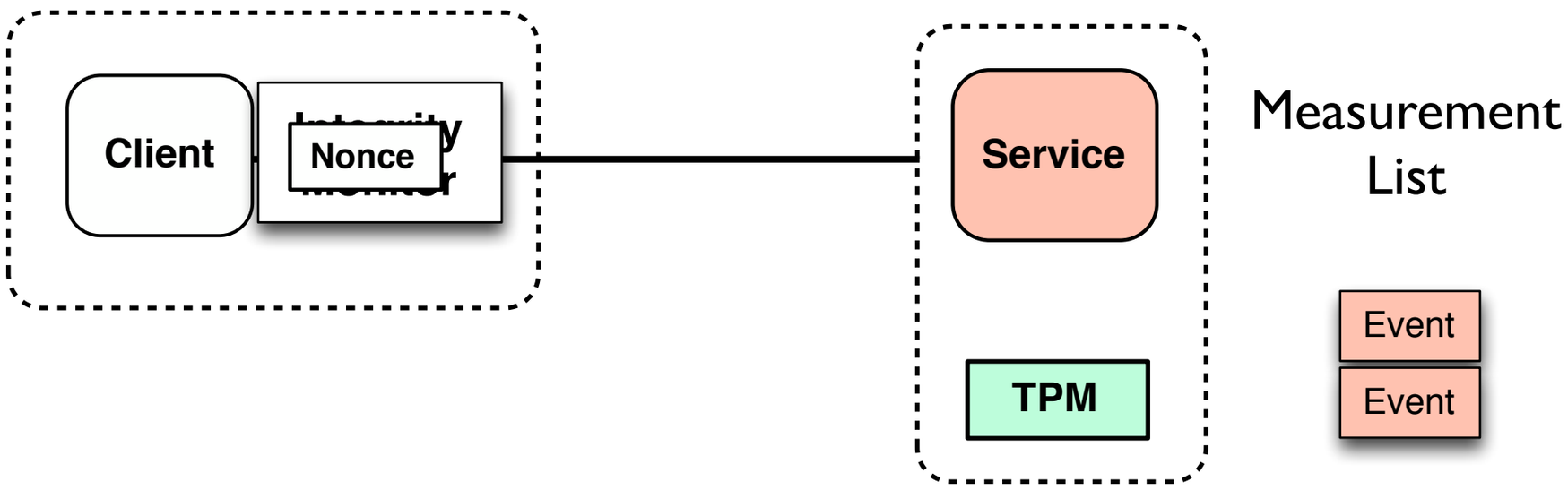




TCG Remote Attestation

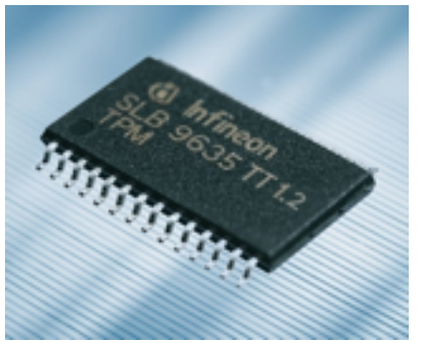
- Trusted Platform Module (TPM)
  - PCRs store event measurements
  - Protected key pair uniquely identifies platform
  - Enables remote attestation of recorded events

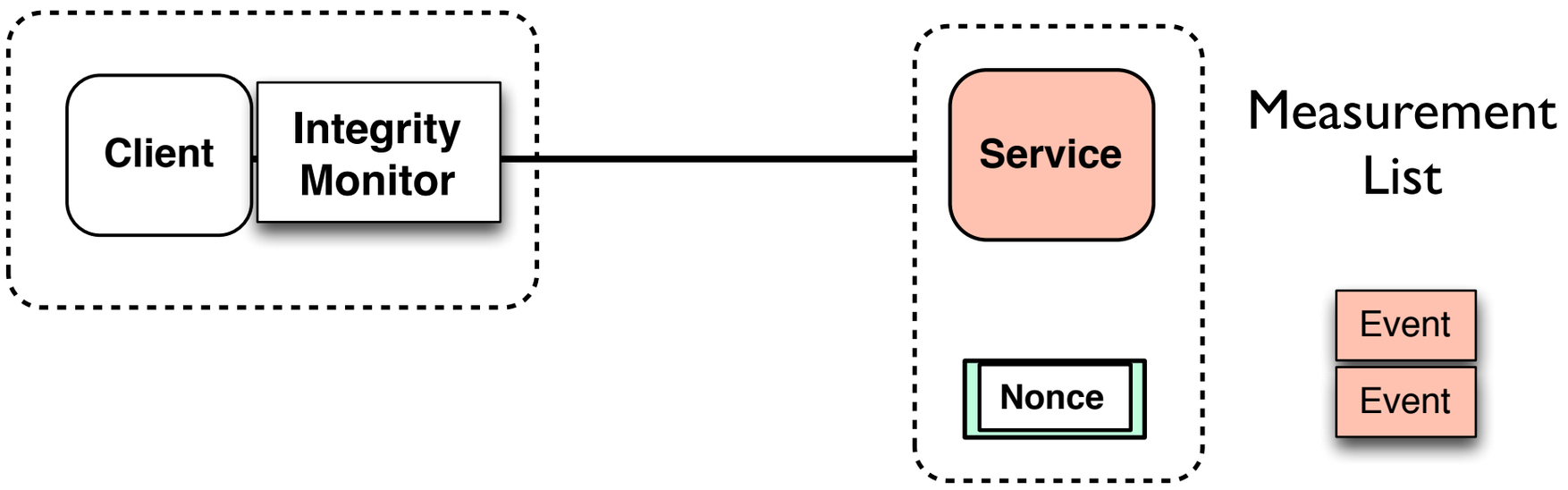




TCG Remote Attestation

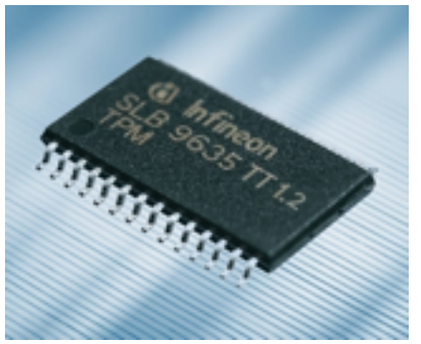
- Trusted Platform Module (TPM)
  - PCRs store event measurements
  - Protected key pair uniquely identifies platform
  - Enables remote attestation of recorded events

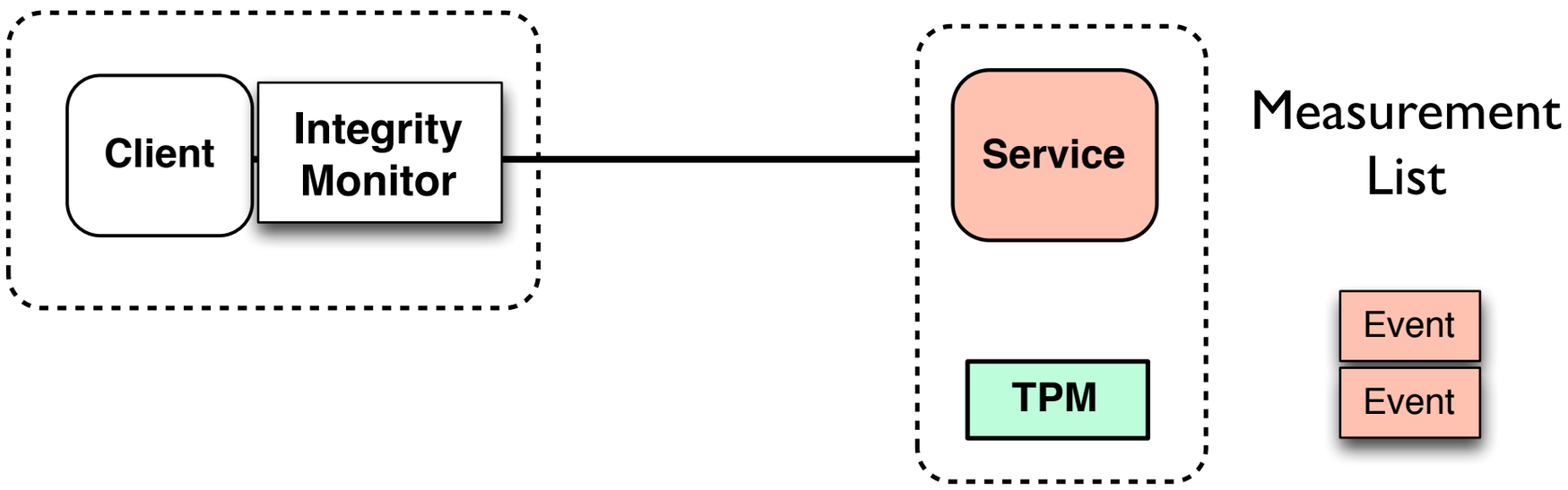




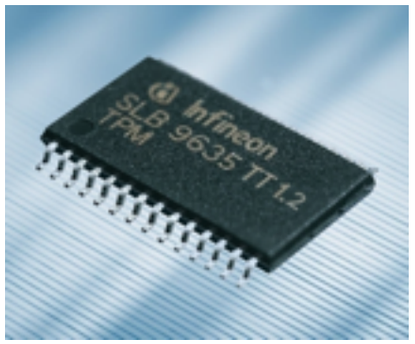
TCG Remote Attestation

- Trusted Platform Module (TPM)
  - PCRs store event measurements
  - Protected key pair uniquely identifies platform
  - Enables remote attestation of recorded events

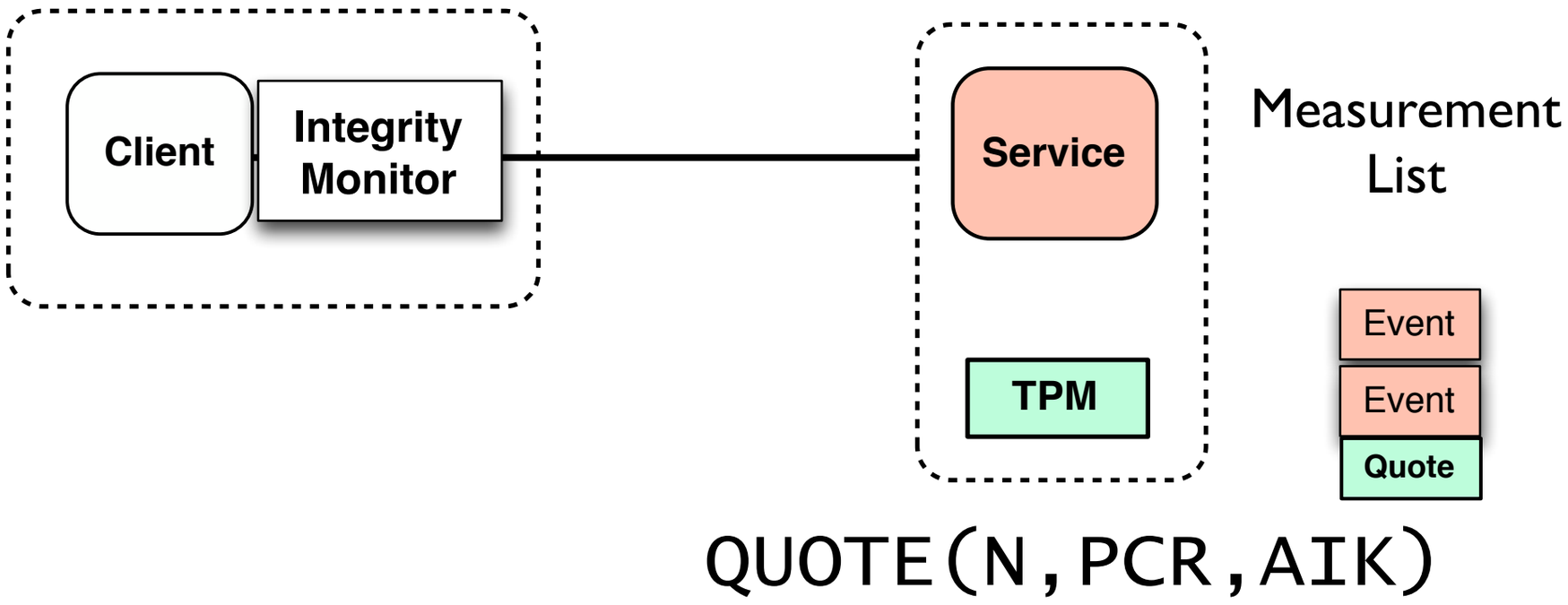




TCG Remote Attestation

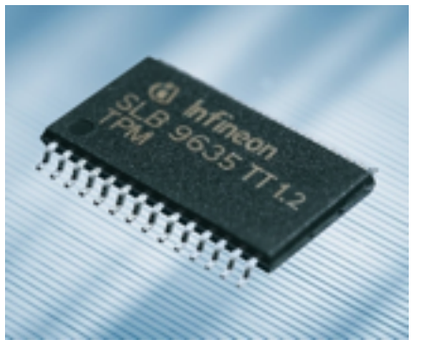


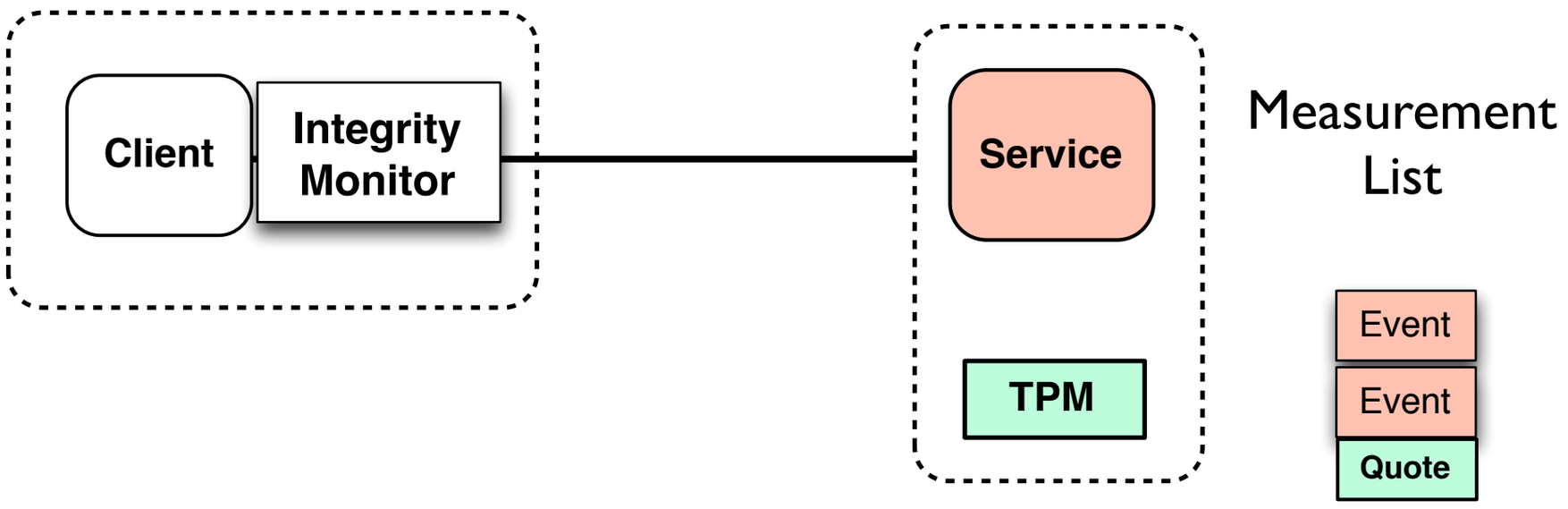
- Trusted Platform Module (TPM)
  - PCRs store event measurements
  - Protected key pair uniquely identifies platform
  - Enables remote attestation of recorded events



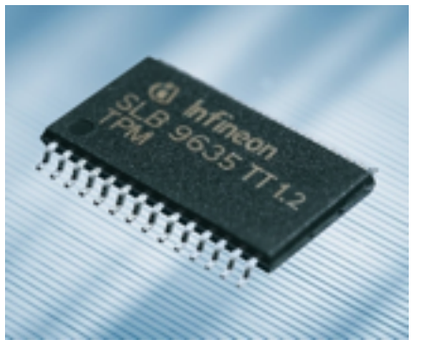
TCG Remote Attestation

- Trusted Platform Module (TPM)
  - PCRs store event measurements
  - Protected key pair uniquely identifies platform
  - Enables remote attestation of recorded events

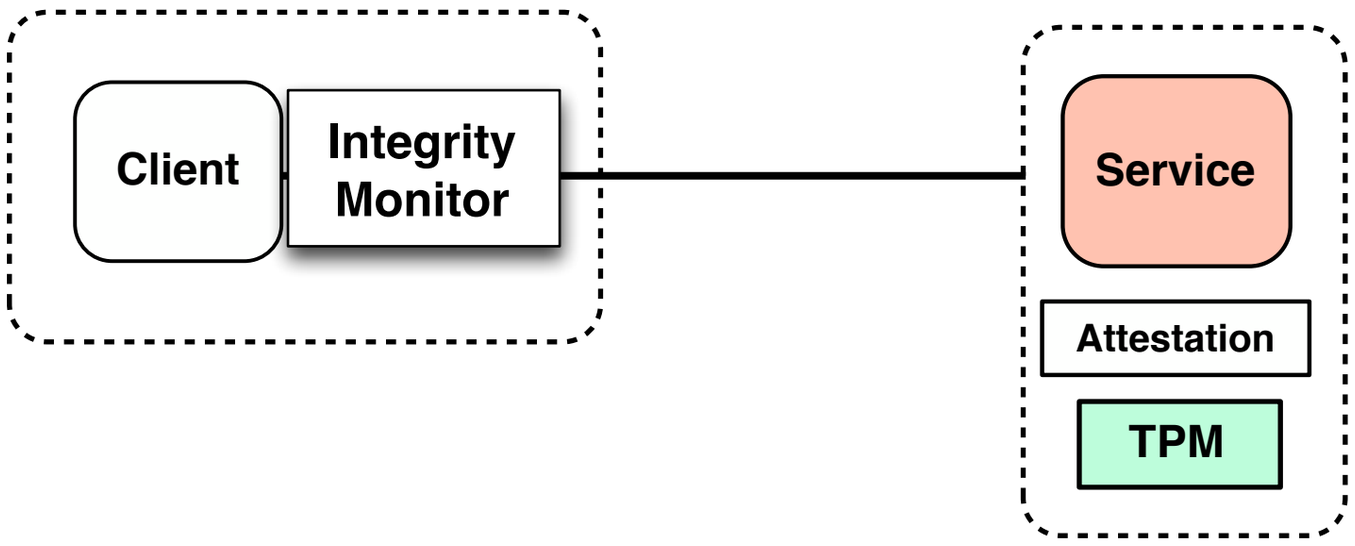




TCG Remote Attestation



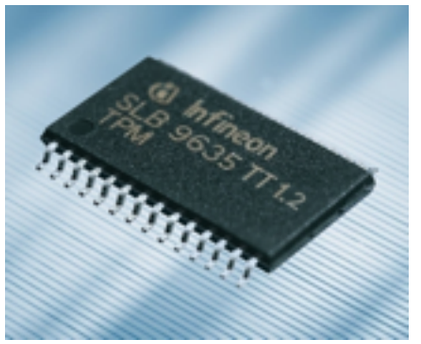
- Trusted Platform Module (TPM)
  - PCRs store event measurements
  - Protected key pair uniquely identifies platform
  - Enables remote attestation of recorded events

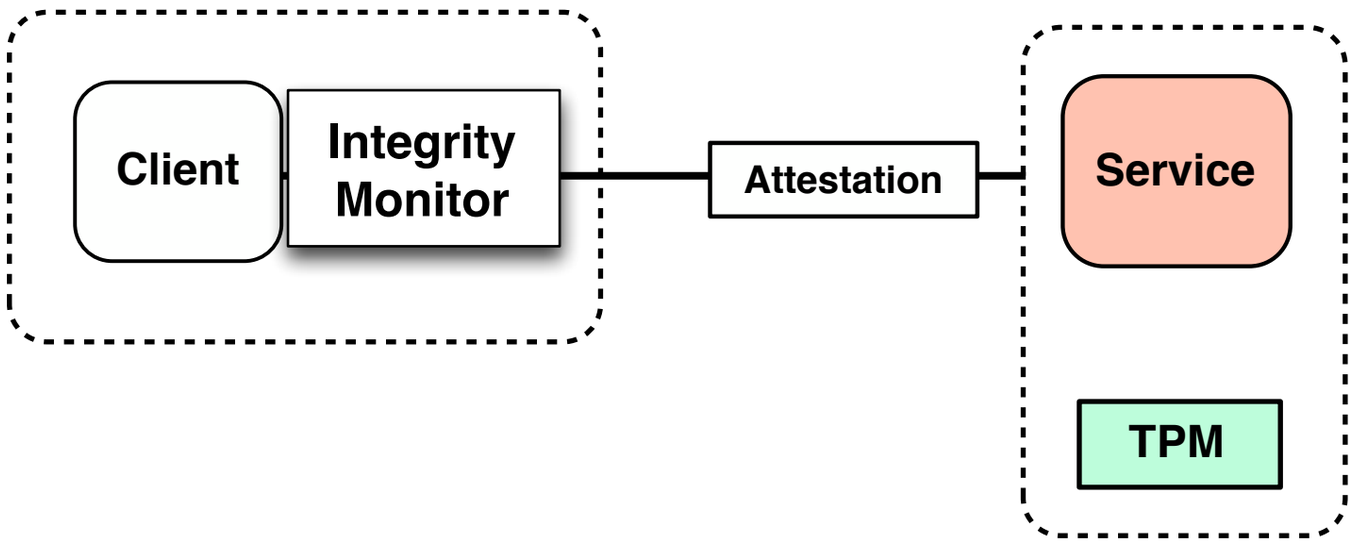


TCG Remote Attestation

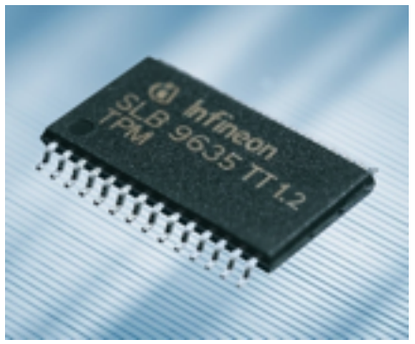
PENNSTATE 1855

- Trusted Platform Module (TPM)
  - PCRs store event measurements
  - Protected key pair uniquely identifies platform
  - Enables remote attestation of recorded events

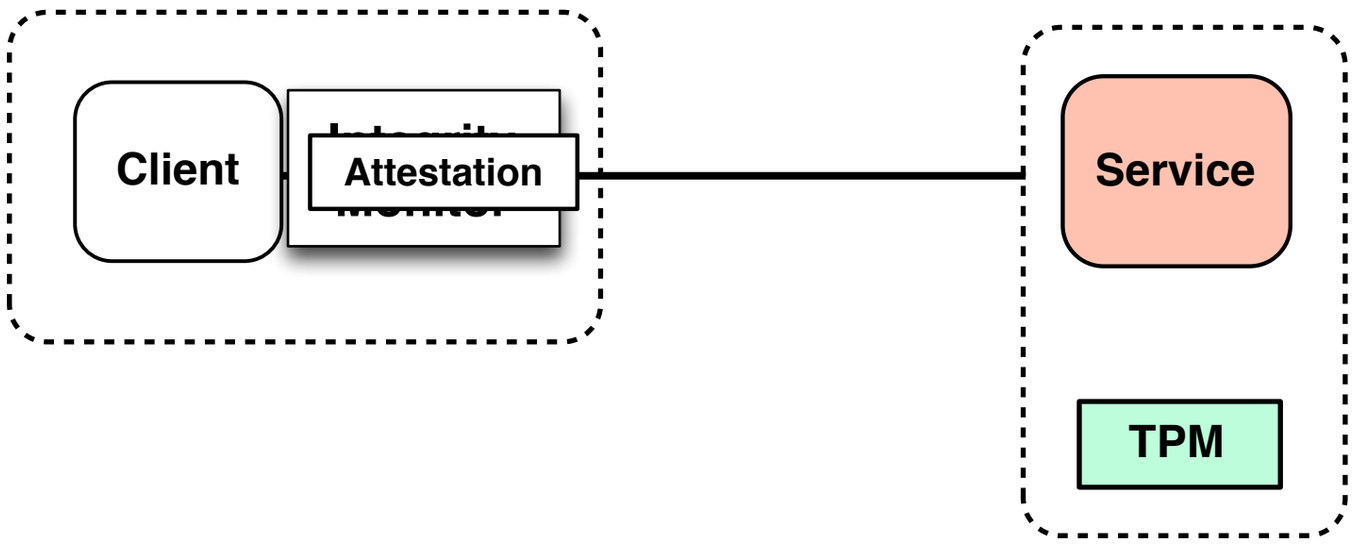




TCG Remote Attestation

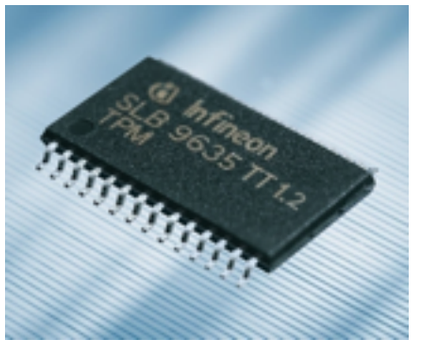


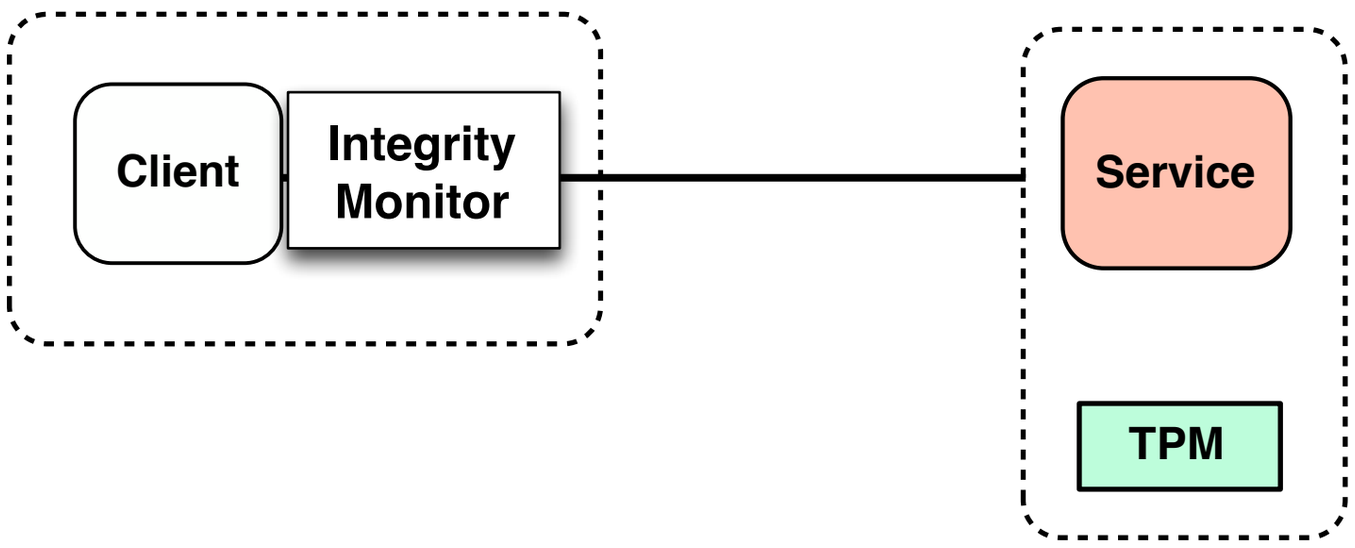
- Trusted Platform Module (TPM)
  - PCRs store event measurements
  - Protected key pair uniquely identifies platform
  - Enables remote attestation of recorded events



TCG Remote Attestation

- Trusted Platform Module (TPM)
  - PCRs store event measurements
  - Protected key pair uniquely identifies platform
  - Enables remote attestation of recorded events





Trusted Computing Pools

- TPM and OpenStack - trusted computing pools
- https://wiki.openstack.org/wiki/TrustedComputingPools
- Identify a node's trustworthiness
  - 1. Compute nodes boot with Intel TXT technology enabled
  - 2. The compute node's BIOS, hypervisor, and OS are measured
  - 3. A quote containing these measured data is sent when challenged by an attestation server
  - 4. Attestation server verifies those measurements against good/known database to determine node's trustworthiness
- Works within cloud, but not for customers

Measurement Limitations

- Administrator decides what to measure
  - Difficult to verify arbitrary criteria
  - Too little or irrelevant information

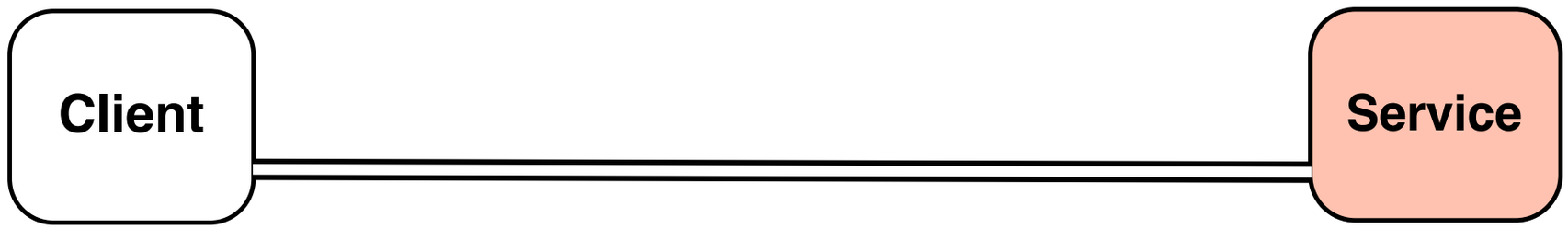
```
0 299fd283e77a12a11510b54c0be90aaa06f21cea 07 [S-CRTM Contents]
0 026a198354eb8fb843dad62b417691c533fdd69a 07 [S-CRTM Contents]
4 c1e25c3f6b0dc78d57296aa2870ca6f782ccf80f 05 [Calling INT 19h]
0 85e53271e14006f0265921d02d4d736cdc580b0b 04 [#]
1 85e53271e14006f0265921d02d4d736cdc580b0b 04 [#]
4 85e53271e14006f0265921d02d4d736cdc580b0b 04 [#]
5 85e53271e14006f0265921d02d4d736cdc580b0b 04 [#]
7 85e53271e14006f0265921d02d4d736cdc580b0b 04 [#]
4 38f30a0a967fcf2bfee1e3b2971de540115048c8 05 [Returned INT 19h]
4 b72a2b23277c4c7a1a61a22815ba892e27d23709 05 [Return via INT 18h]
4 1f7d73c9e899ca12d634a5e0af164df7877e62ed 0c [Compact Hash]
5 eec6e93d0e04af1ab094bce46ae6ca75dd46251c 0c [Compact Hash]
```

Measurement Limitations

- Administrator decides what to measure
  - Difficult to verify arbitrary criteria
  - Too little or irrelevant information

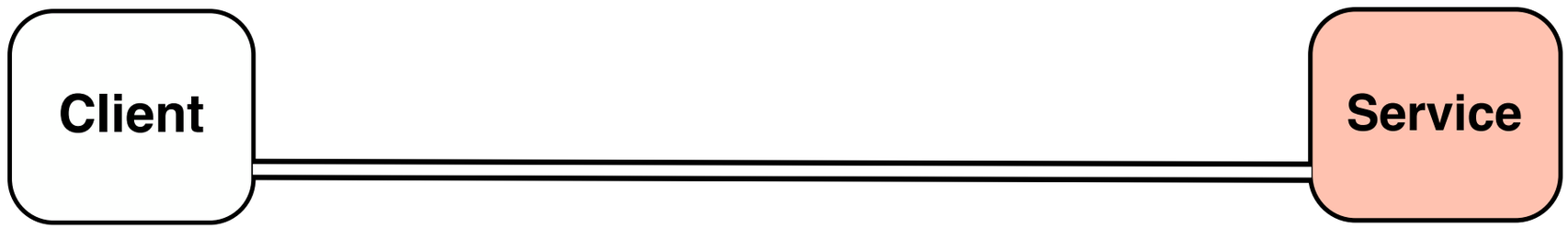
```
0 299fd283e77a12a11510b54c0be90aaa06f21cea 07 [S-CRTM Contents]
0 026a198354eb8fb843dad62b417691c533fdd69a 07 [S-CRTM Contents]
4 c1e25c3f6b0dc78d57296aa2870ca6f782ccf80f 05 [Calling INT 19h]
0 85e53271e14006f0265921d02d4d736cdc580b0b 04 [#]
1 85e53271e14006f0265921d02d4d736cdc580b0b 04 [#]
4 85e53271e14006f0265921d02d4d736cdc580b0b 04 [#]
5 85e53271e14006f0265921d02d4d736cdc580b0b 04 [#]
7 85e53271e14006f0265921d02d4d736cdc580b0b 04 [#]
4 38f30a0a967fcf2bfee1e3b2971de540115048c8 05 [Returned INT 19h]
4 b72a2b23277c4c7a1a61a22815ba892e27d23709 05 [Return via INT 18h]
4 1f7d73c9e899ca12d634a5e0af164df7877e62ed 0c [Compact Hash]
5 eec6e93d0e04af1ab094bce46ae6ca75dd46251c 0c [Compact Hash]
```

Attestation Limitations



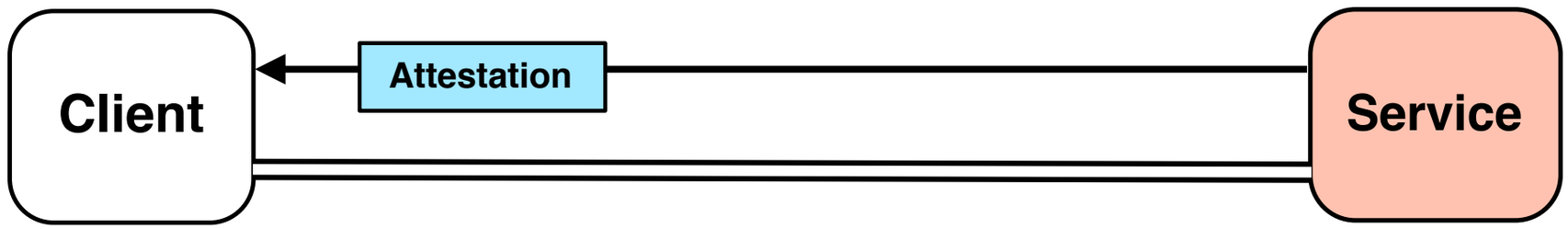
Attestation Limitations

- Attestation limits reporting timeliness



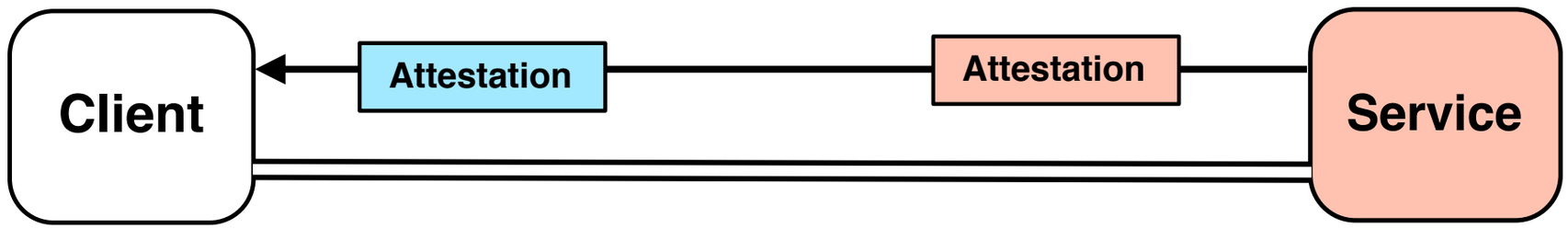
Attestation Limitations

- Attestation limits reporting timeliness



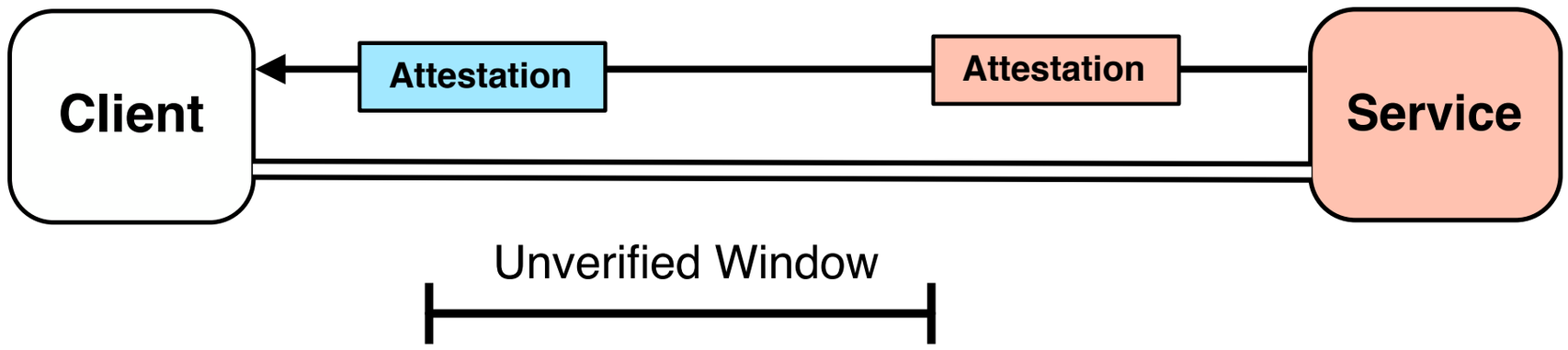
Attestation Limitations

- Attestation limits reporting timeliness



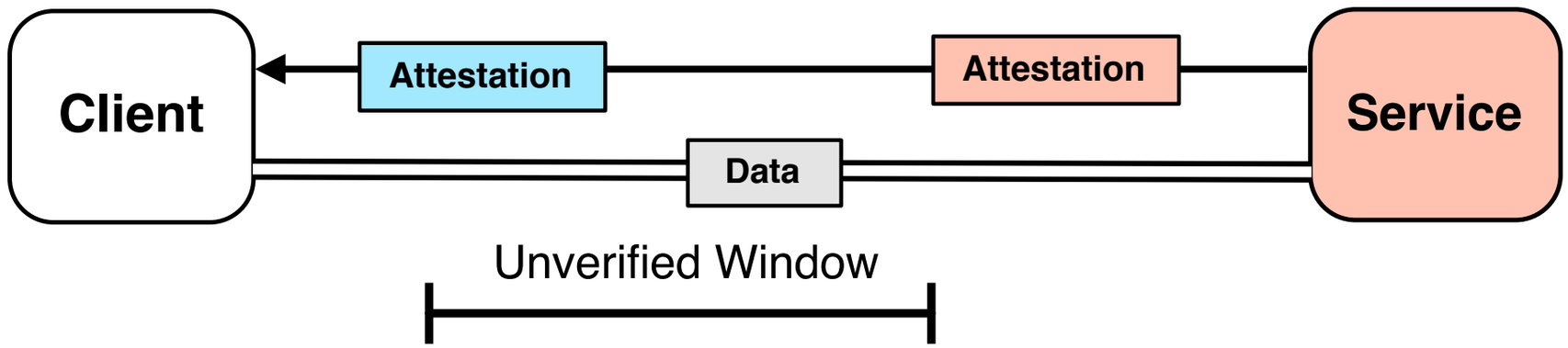
Attestation Limitations

- Attestation limits reporting timeliness



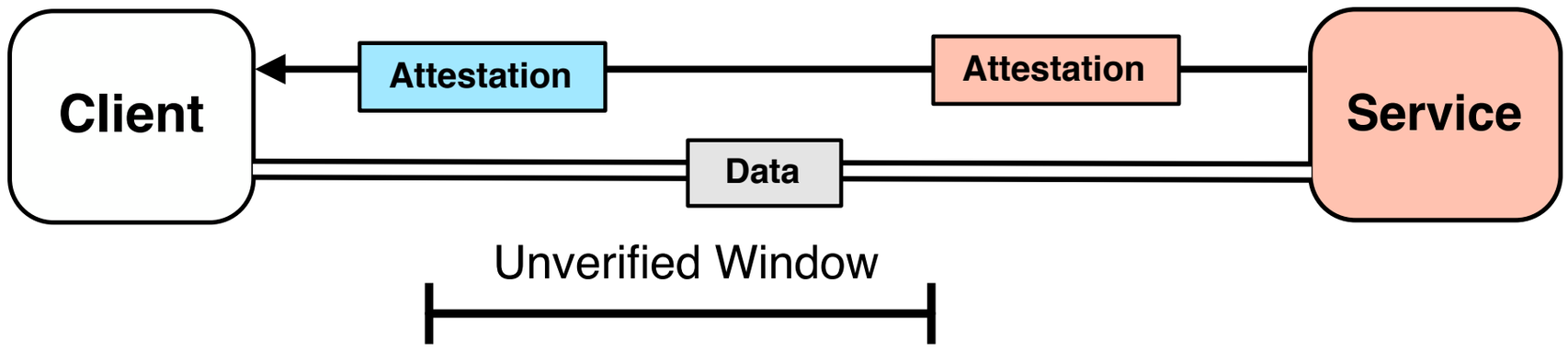
Attestation Limitations

- Attestation limits reporting timeliness



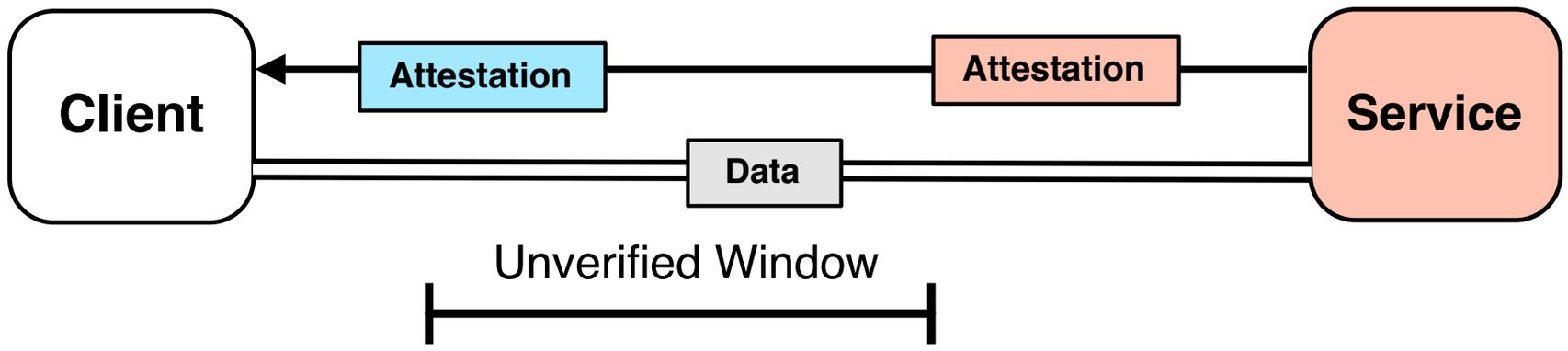
Attestation Limitations

- Attestation limits reporting timeliness
  - Made worse by slow hardware



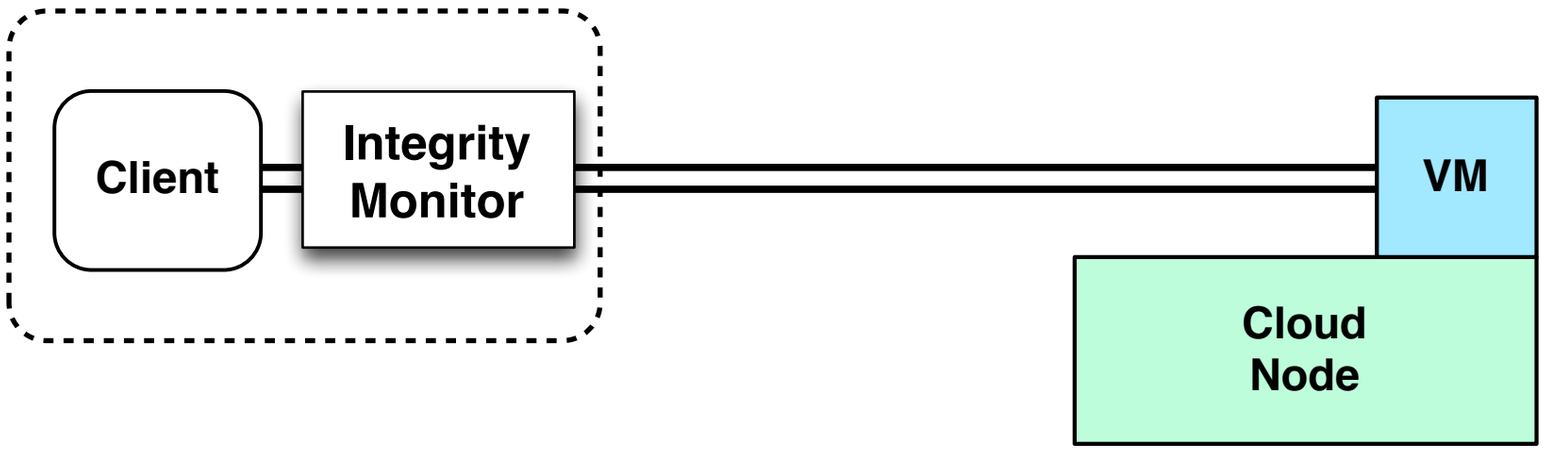
Attestation Limitations

- Attestation limits reporting timeliness
  - Made worse by slow hardware
  - High demand for attestations



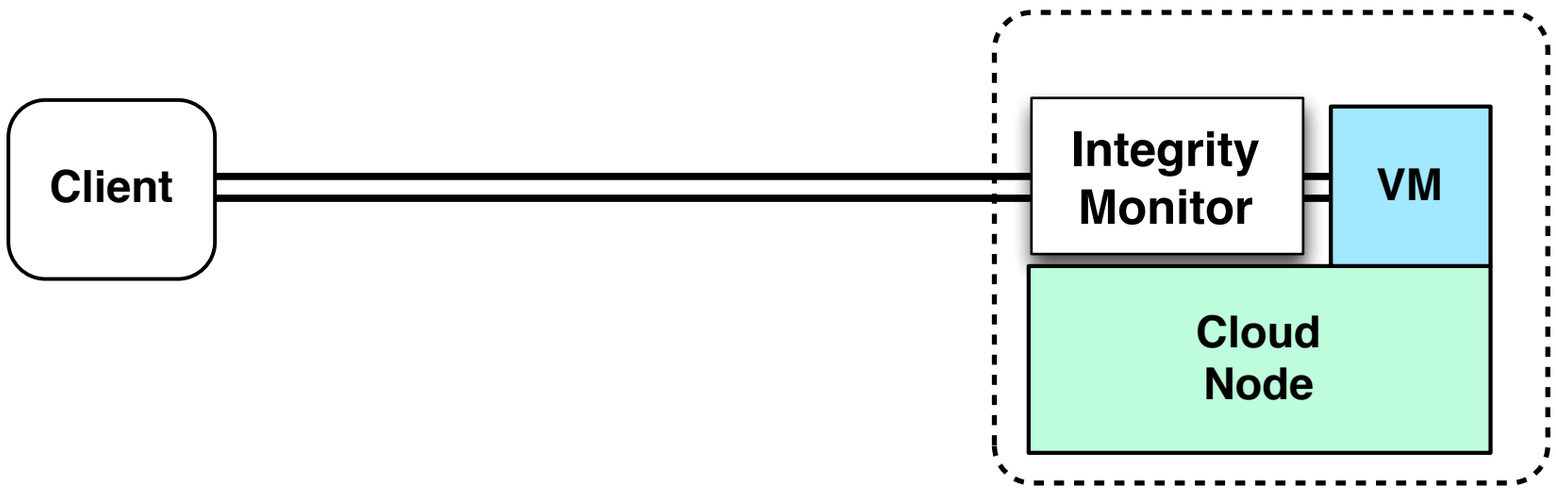
Improved Monitoring

- Insight: Move integrity monitor to the cloud node
  - Avoid having to poll for attestations
  - Measure only integrity-relevant events



Improved Monitoring

- Insight: Move integrity monitor to the cloud node
  - Avoid having to poll for attestations
  - Measure only integrity-relevant events



Challenges

- Can we trust a monitor in a cloud node?
- Can we efficiently and securely measure comprehensive integrity in a variety of conditions?
- Can we integrate monitor into a cloud platform?

Challenges

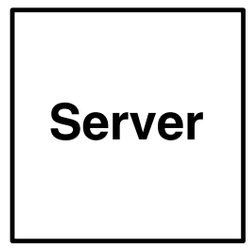
- Can we trust a monitor in a cloud node?
- Can we efficiently and securely measure comprehensive integrity in a variety of conditions?
- Can we integrate monitor into a cloud platform?

Cloud Node Integrity?

Cloud Node Integrity?

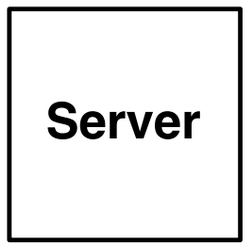
- Can we trust a monitor in a cloud node?
  - Can external clients determine the node's identity?
  - Custom distributions are hard to assess
  - Will we need to poll to check for updates and input?

Verification Benefits



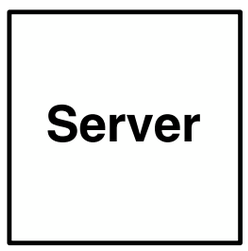
Verification Benefits

- Clouds manage node provisioning
  - Administers PKI for machine identities
  - Network installs a master disk image and customizes



Verification Benefits

- Clouds manage node provisioning
  - Administers PKI for machine identities
  - Network installs a master disk image and customizes

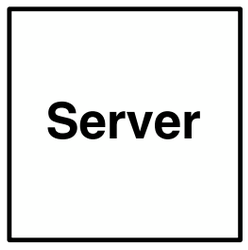


Verification Benefits

- Clouds manage node provisioning
  - Administers PKI for machine identities
  - Network installs a master disk image and customizes

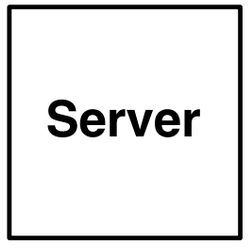
Verification Benefits

- Clouds manage node provisioning
  - Administers PKI for machine identities
  - Network installs a master disk image and customizes



Verification Benefits

- Clouds manage node provisioning
  - Administers PKI for machine identities
  - Network installs a master disk image and customizes

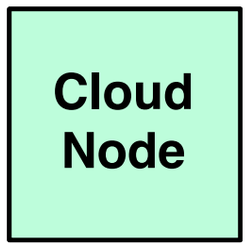


Verification Benefits

- Clouds manage node provisioning
  - Administers PKI for machine identities
  - Network installs a master disk image and customizes

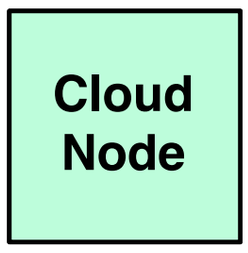
Verification Benefits

- Clouds manage node provisioning
  - Administers PKI for machine identities
  - Network installs a master disk image and customizes



Verification Benefits

- Clouds manage node provisioning
  - Administers PKI for machine identities
  - Network installs a master disk image and customizes
- Node is essentially a static hosting utility
  - Should not require persistent changes at runtime
  - Should only allow inputs to well protected interfaces



Root of Trust for Installation

- Root of Trust for Installation (ROTI) *[ACSAC 2007]*
  - Binds the filesystem to a known installer (origin)
  - Detect persistent changes across reboots
  - Reinstall to update or control admins *[ASIACCS 2012]*

Root of Trust for Installation

- Root of Trust for Installation (ROTI) *[ACSAC 2007]*
  - Binds the filesystem to a known installer (origin)
  - Detect persistent changes across reboots
  - Reinstall to update or control admins *[ASIACCS 2012]*

Root of Trust for Installation

- Root of Trust for Installation (ROTI) *[ACSAC 2007]*
  - Binds the filesystem to a known installer (origin)
  - Detect persistent changes across reboots
  - Reinstall to update or control admins *[ASIACCS 2012]*

Root of Trust for Installation

- Root of Trust for Installation (ROTI) *[ACSAC 2007]*
  - Binds the filesystem to a known installer (origin)
  - Detect persistent changes across reboots
  - Reinstall to update or control admins *[ASIACCS 2012]*

Root of Trust for Installation

- Root of Trust for Installation (ROTI) *[ACSAC 2007]*
  - Binds the filesystem to a known installer (origin)
  - Detect persistent changes across reboots
  - Reinstall to update or control admins *[ASIACCS 2012]*

`Quote(Installer,Image,FS,AIK)`

Root of Trust for Installation

- Root of Trust for Installation (ROTI) *[ACSAC 2007]*
  - Binds the filesystem to a known installer (origin)
  - Detect persistent changes across reboots
  - Reinstall to update or control admins *[ASIACCS 2012]*

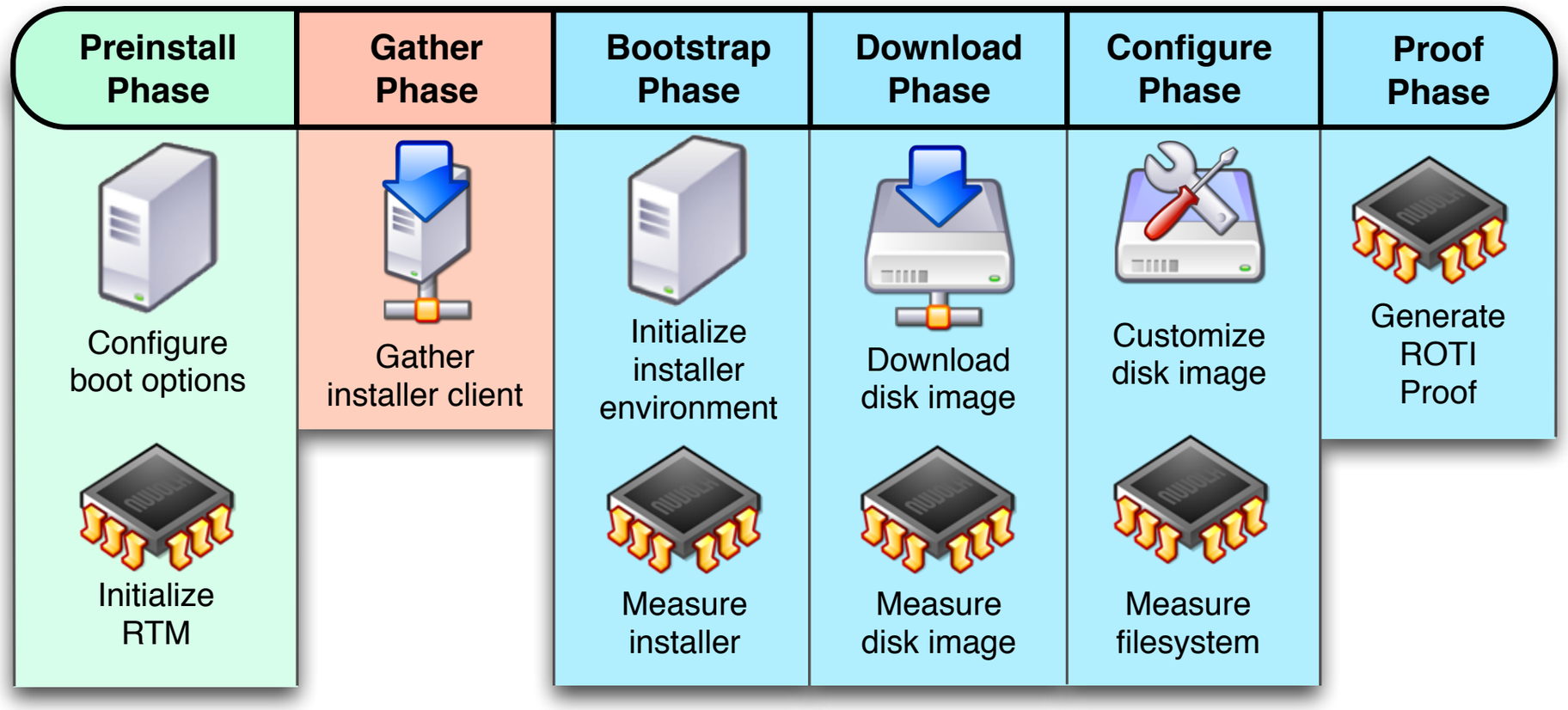
ROTI Proof

netROTI

- Measure network installation process *[IEEE S&P 2011]*
  - Network installation receives untrusted inputs
  - Bootstrap installation from a measured launch environment

netROTI

- Measure network installation process *[IEEE S&P 2011]*
  - Network installation receives untrusted inputs
  - Bootstrap installation from a measured launch environment



netROTI Proof: Sig(MLE, Installer, Image, FS, AIK)

netROTI

- Measure network installation process *[IEEE S&P 2011]*
  - Network installation receives untrusted inputs
  - Bootstrap installation from a measured launch environment

Protect Monitors on Nodes

- Can the node protect the monitors?
  - Hardware security
  - Virtualization security
  - OS security
  - Hardened services
- What happens if the VMM and/or management VM are compromised?

CloudVisor

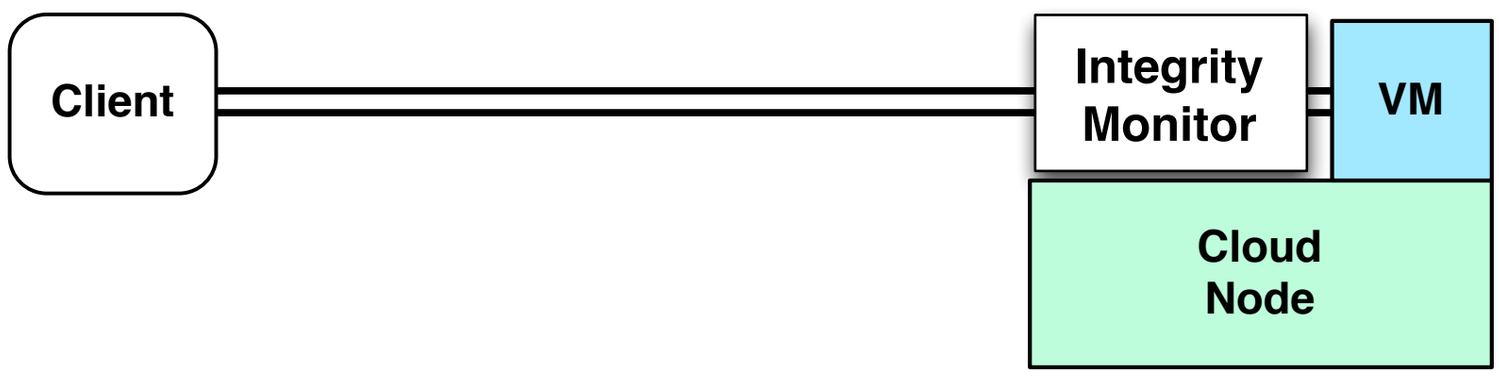
- Is a virtualization layer below VMM *[SOSP 2011]*
  - Implemented using available nested virtualization
    - Interpose on sensitive events
  - Leverage virtualization protection techniques
    - CPU - conceal CPU states from VMM
    - Memory - control VMM access to guest memory to prevent leakage
    - I/O - encrypt I/O data (e.g., disk)
  - Improves on ideas of guest VM protection from Proxos and Overshadow
  - Does not address side channels or cloud services

Challenges

- Can we trust a monitor in a cloud node?
- Can we efficiently and securely measure comprehensive integrity in a variety of conditions?
- Can we integrate monitor into a cloud platform?

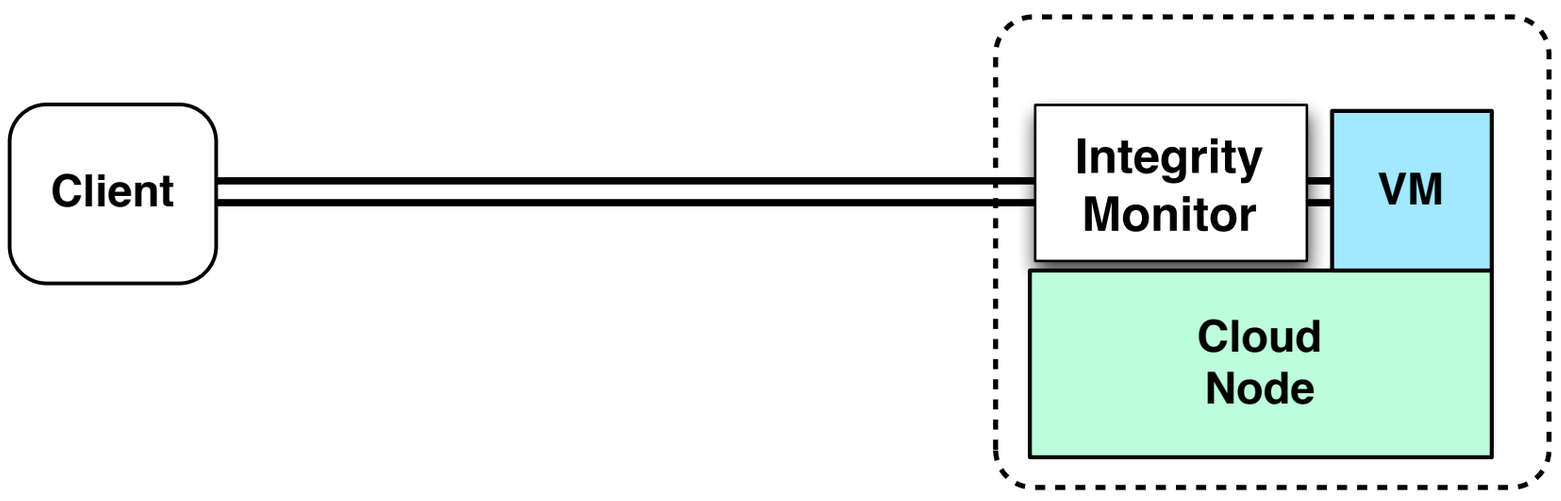
Custom Measurement

- What criteria can a client specify and monitor?
- How can the monitor measure the VM?
  - Various criteria should be supported
  - Need to protect monitor from VM
  - Avoid modifying the VM and affecting performance



Custom Measurement

- What criteria can a client specify and monitor?
- How can the monitor measure the VM?
  - Various criteria should be supported
  - Need to protect monitor from VM
  - Avoid modifying the VM and affecting performance



Measurement Approaches

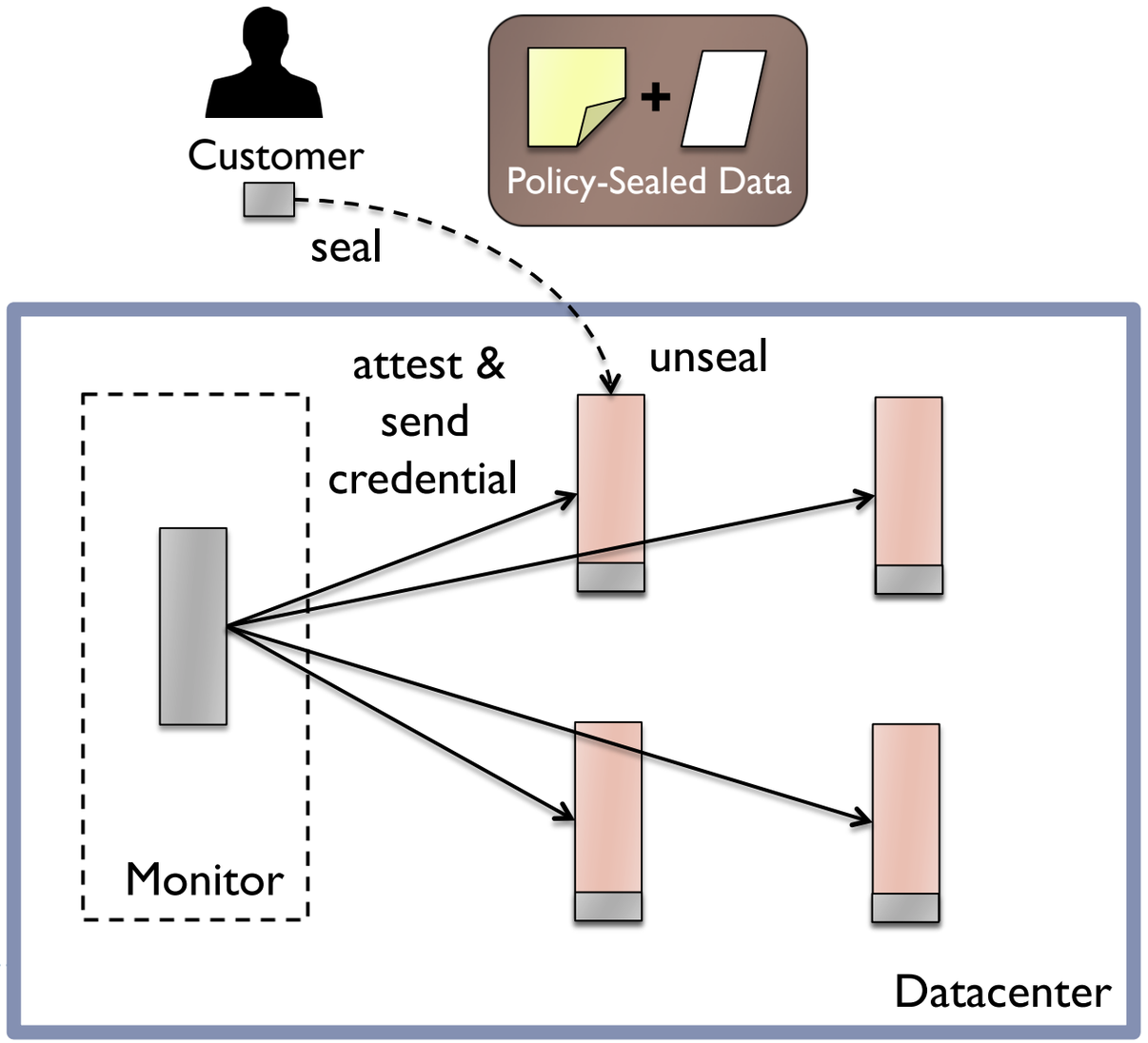
- Loadtime measurement mechanisms
  - LIM, vTPM, TSS, IMA, PRIMA, OpenTC PET
  - Requires VM support to report measurements
  - Vulnerable to compromise
- VM Introspection
  - Livewire, Xenaccess, VMSafe, Vex, SIM, VM Out-grafting
  - Performance impact due to VM suspend
  - Not aware of client requirements

Excalibur

- Policy-sealed data *[USENIX Sec 2012b]*
  - Do not release my data to the cloud until that cloud satisfies my requirements
  - Customer-chosen policy
- How to ensure that only nodes that satisfy customer-chosen policy get data?
  - Attribute-based encryption
  - Encrypt data using ABE description of load-time configuration
  - A verifiable monitor is trusted to delegate correct credentials to nodes (using TPM attestations)

Excalibur Approach

- Check node configurations
  - Monitor attests nodes in background
- Scalable policy enforcement
  - CP-ABE operations at client-side lib



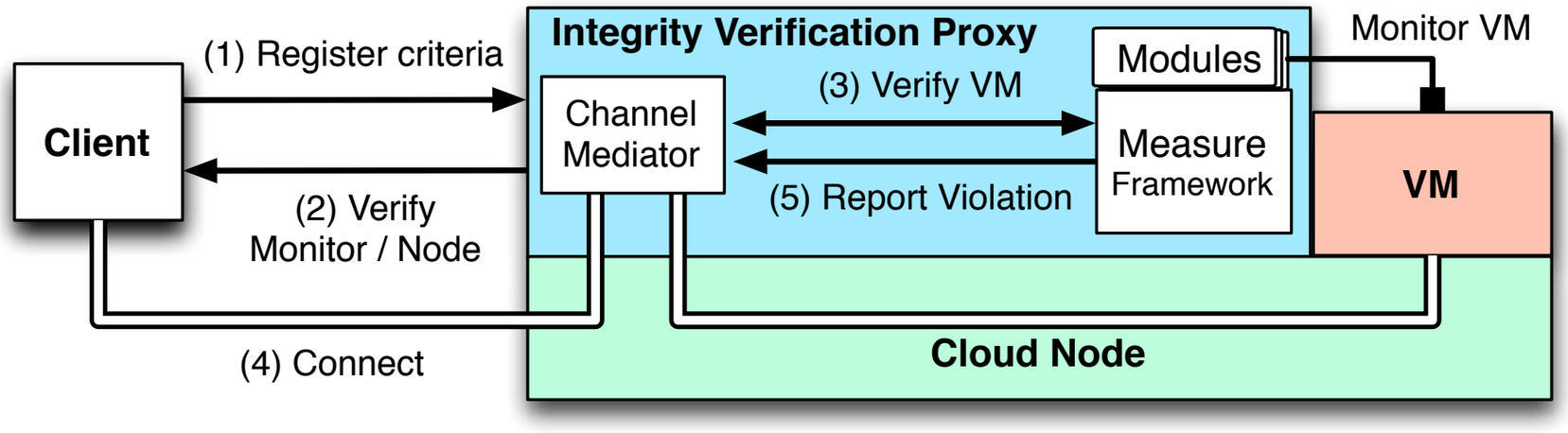
From Nuno Santos' slides

Runtime Monitoring

- Excalibur does not address runtime issues with instance
  - Customers may want to ensure that clients of their services only receive communications from satisfactory instances
  - Customer may want to take remediative actions

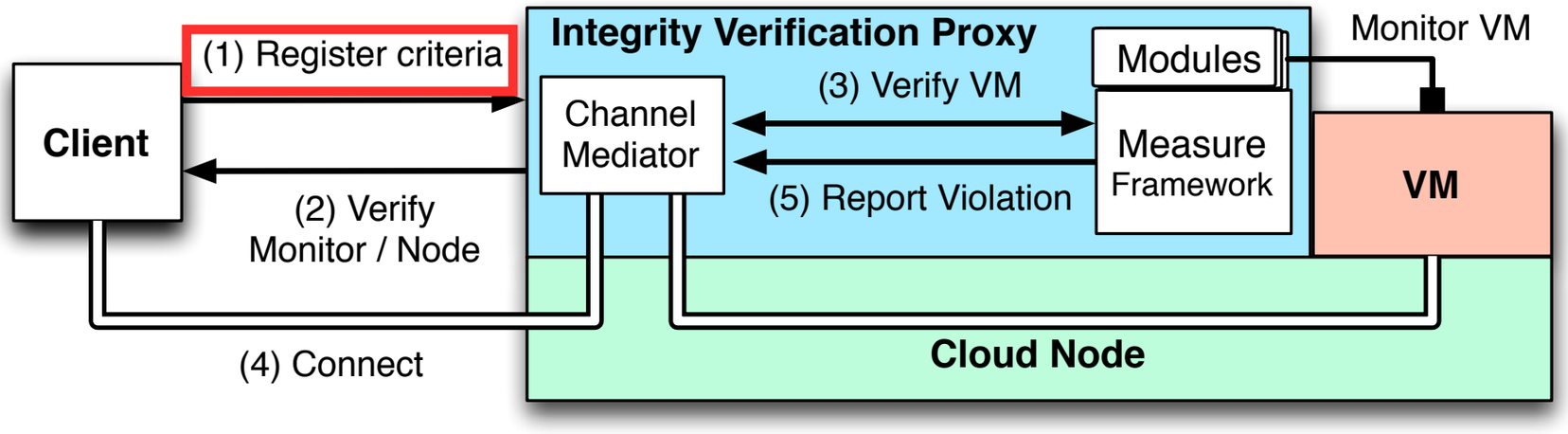
Integrity Verification Proxy

- Clients specify criteria to be enforced by a channel mediator *[TRUST 2012]*
- Set of measurement modules verifies the criteria
  - Loadtime modules measure VM components
  - VM Introspection to examine runtime criteria
    - E.g., Binaries/data loaded, enforcement disabled, policy changes, kernel data (binary handler), etc.



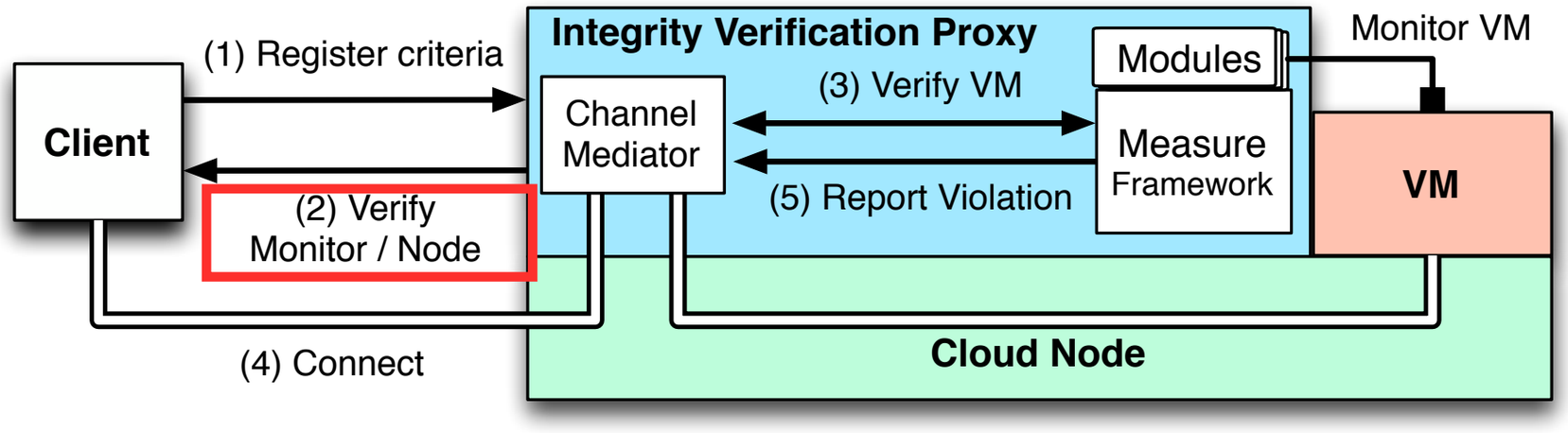
Integrity Verification Proxy

- Clients specify criteria to be enforced by a channel mediator *[TRUST 2012]*
- Set of measurement modules verifies the criteria
  - Loadtime modules measure VM components
  - VM Introspection to examine runtime criteria
    - E.g., Binaries/data loaded, enforcement disabled, policy changes, kernel data (binary handler), etc.



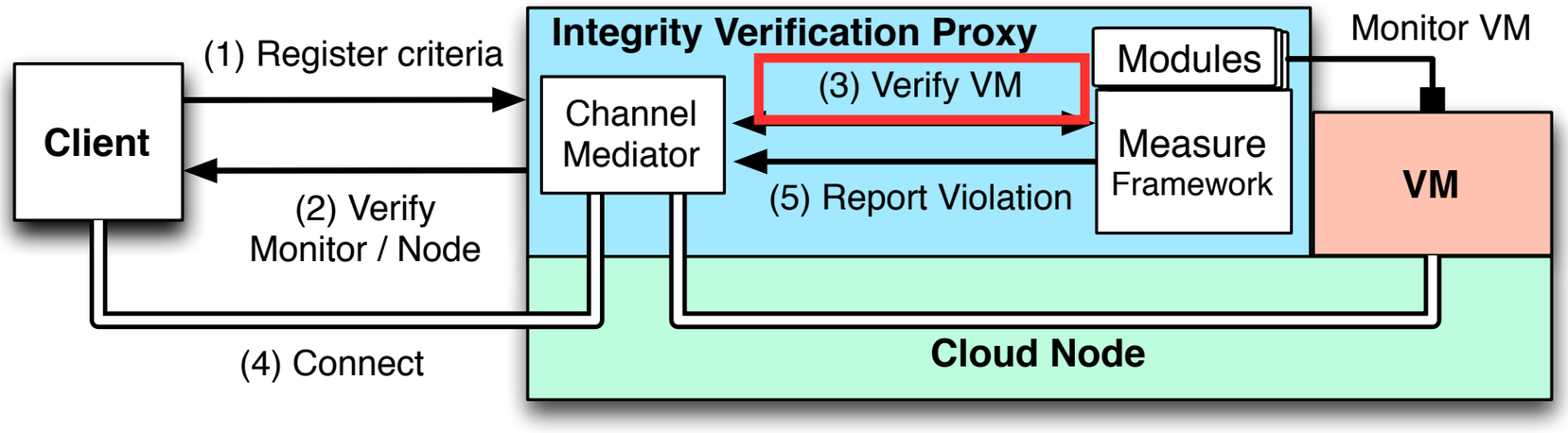
Integrity Verification Proxy

- Clients specify criteria to be enforced by a channel mediator *[TRUST 2012]*
- Set of measurement modules verifies the criteria
  - Loadtime modules measure VM components
  - VM Introspection to examine runtime criteria
    - E.g., Binaries/data loaded, enforcement disabled, policy changes, kernel data (binary handler), etc.



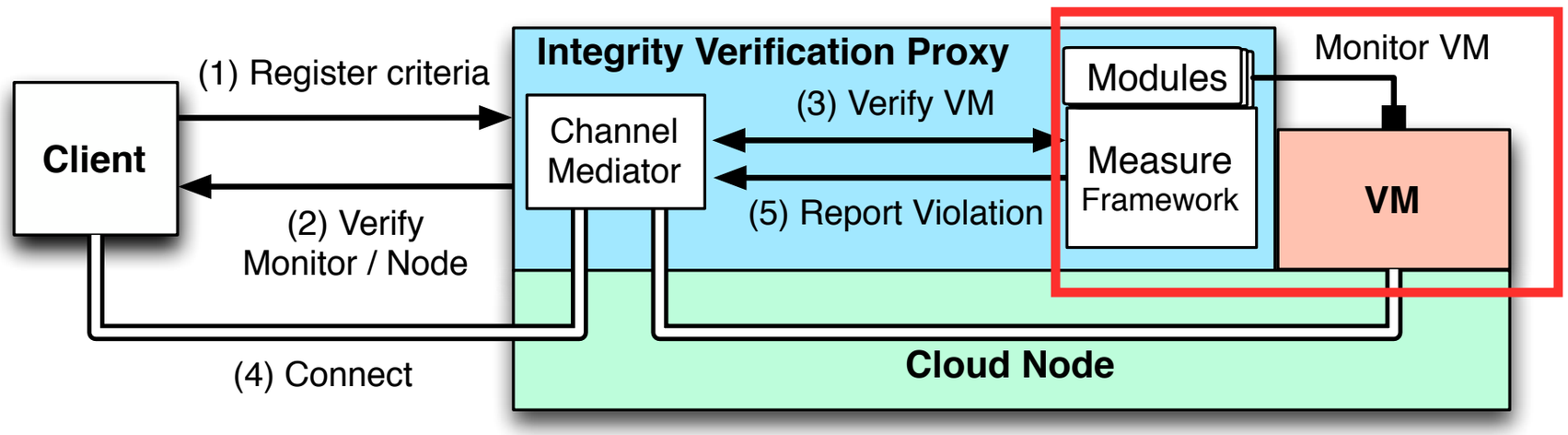
Integrity Verification Proxy

- Clients specify criteria to be enforced by a channel mediator *[TRUST 2012]*
- Set of measurement modules verifies the criteria
  - Loadtime modules measure VM components
  - VM Introspection to examine runtime criteria
    - E.g., Binaries/data loaded, enforcement disabled, policy changes, kernel data (binary handler), etc.



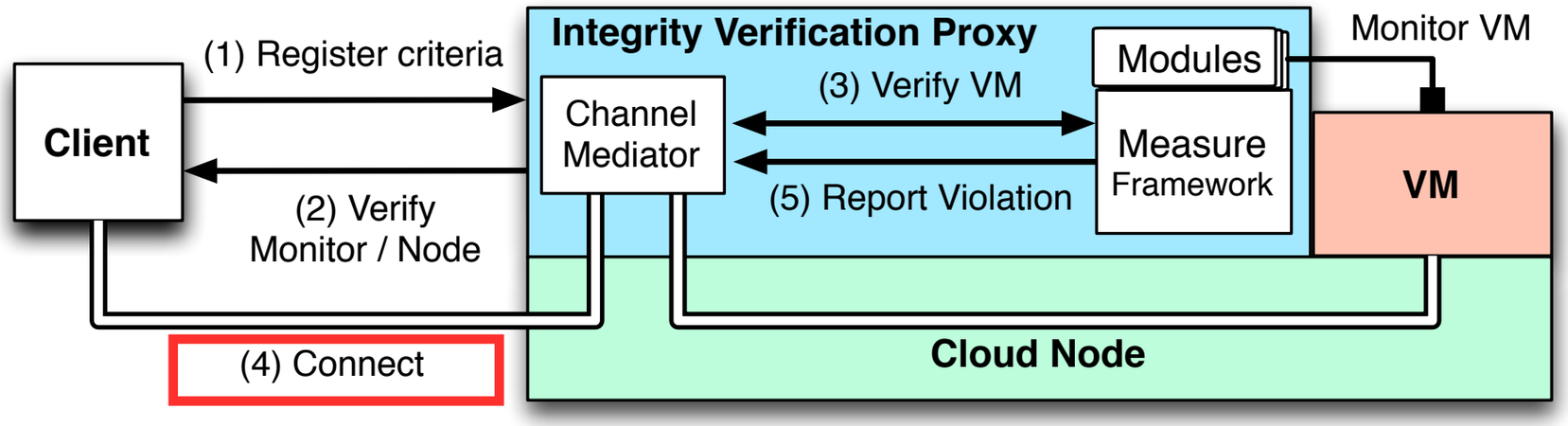
Integrity Verification Proxy

- Clients specify criteria to be enforced by a channel mediator *[TRUST 2012]*
- Set of measurement modules verifies the criteria
  - Loadtime modules measure VM components
  - VM Introspection to examine runtime criteria
    - E.g., Binaries/data loaded, enforcement disabled, policy changes, kernel data (binary handler), etc.



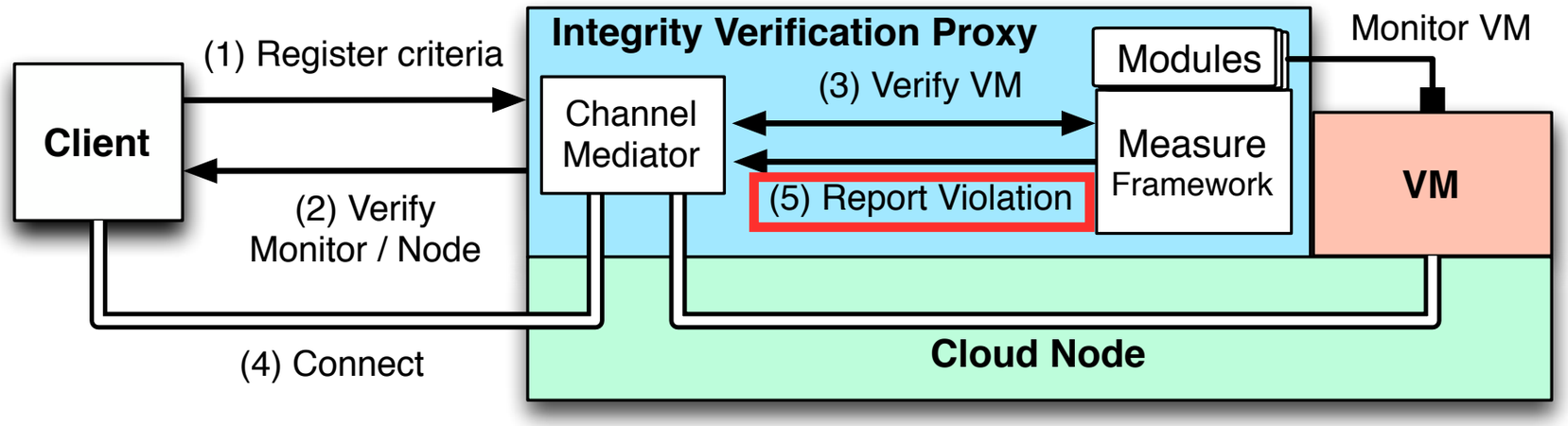
Integrity Verification Proxy

- Clients specify criteria to be enforced by a channel mediator *[TRUST 2012]*
- Set of measurement modules verifies the criteria
  - Loadtime modules measure VM components
  - VM Introspection to examine runtime criteria
    - E.g., Binaries/data loaded, enforcement disabled, policy changes, kernel data (binary handler), etc.



Integrity Verification Proxy

- Clients specify criteria to be enforced by a channel mediator *[TRUST 2012]*
- Set of measurement modules verifies the criteria
  - Loadtime modules measure VM components
  - VM Introspection to examine runtime criteria
    - E.g., Binaries/data loaded, enforcement disabled, policy changes, kernel data (binary handler), etc.



What Criteria?

- What criteria should we check for?
  - Prevent cloud attacks discussed earlier
- Some attacks are difficult to prevent in modern systems
  - Are published instances void of vulnerabilities?
  - Do configurations satisfy classical integrity and secrecy?
  - Can there be no trusted cloud services in TCB?
  - Can all insiders be fully untrusted in working cloud?

Measure enforcement

- Validiate your best effort (e.g., approx. Clark-Wilson)
  - Consistent with what you would do in data center
  - Hopefully, best effort can be improved by future research
- Built Apache VM to approximate Clark-Wilson integrity
  - Acceptable versions of published instances
  - Acceptable policies for firewall, DAC, SELinux, "Process Firewall", etc, and only acceptable code is run
    - Also, expected enforcement is on - e.g., validate kernel loading methods, SELinux enabled, etc. - initially and throughout runtime
  - Validate input parameters from cloud services comply [TR 2013]
  - Limit insider access and trust in insider

Challenges

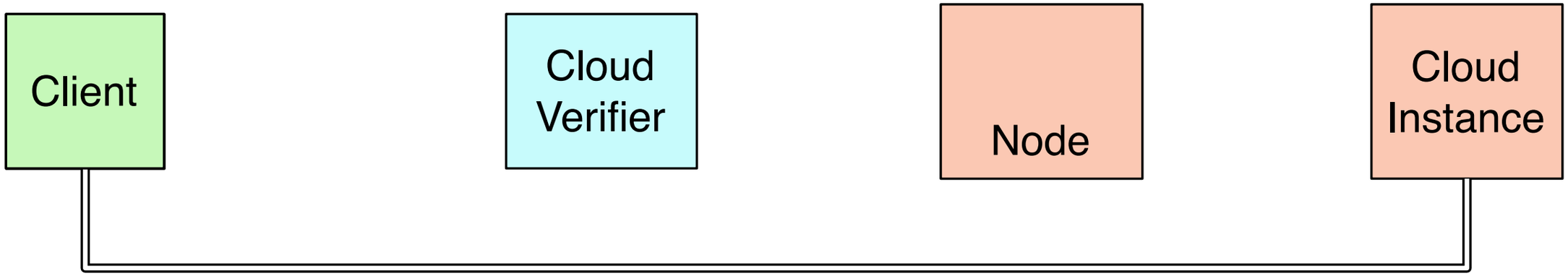
- Can we trust a monitor in a cloud node?
- Can we efficiently and securely measure comprehensive integrity in a variety of conditions?
- Can we integrate monitor into a cloud platform?

Cloud Verifier Overview

Cloud Anchor [CCSW 2010, TrustCom 2012]
+IVP in OpenStack [CSAW 2013]

Cloud Verifier Overview

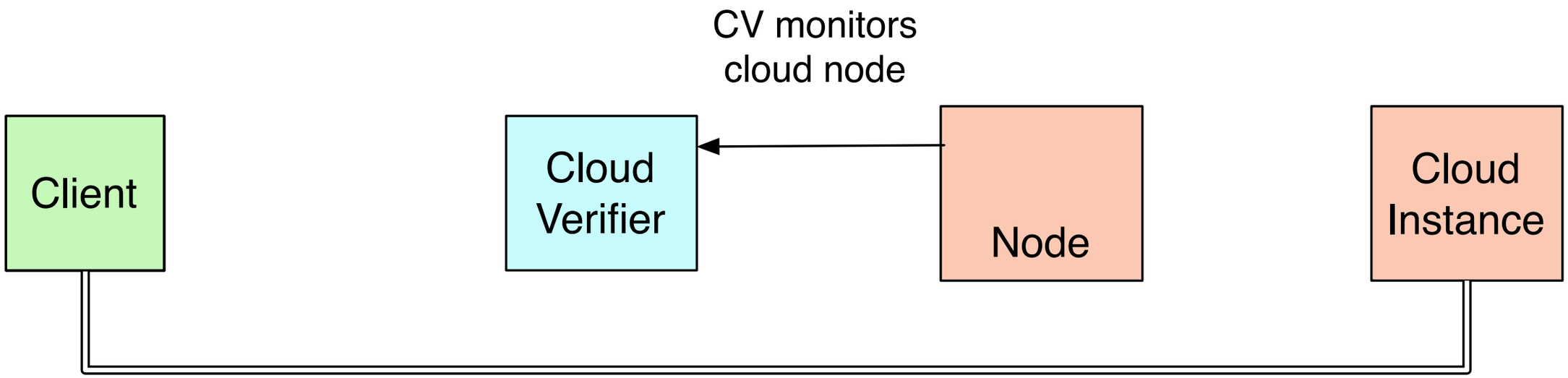
Cloud Anchor [CCSW 2010, TrustCom 2012]
+IVP in OpenStack [CSAW 2013]



Cloud Verifier Overview

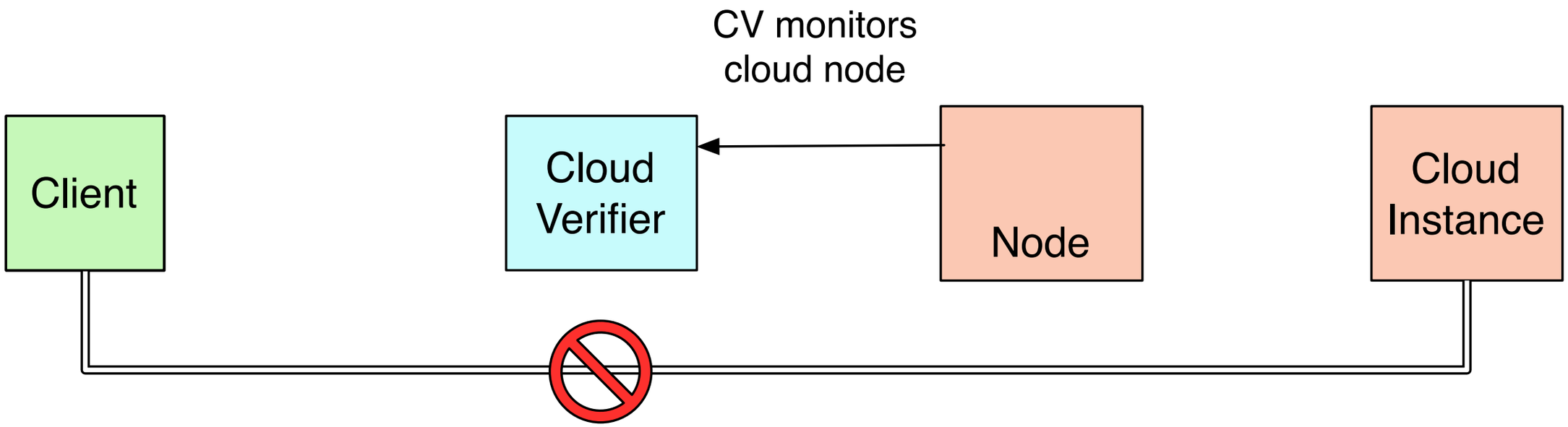
PENNSTATE 1855

Cloud Anchor [CCSW 2010, TrustCom 2012]
+IVP in OpenStack [CSAW 2013]



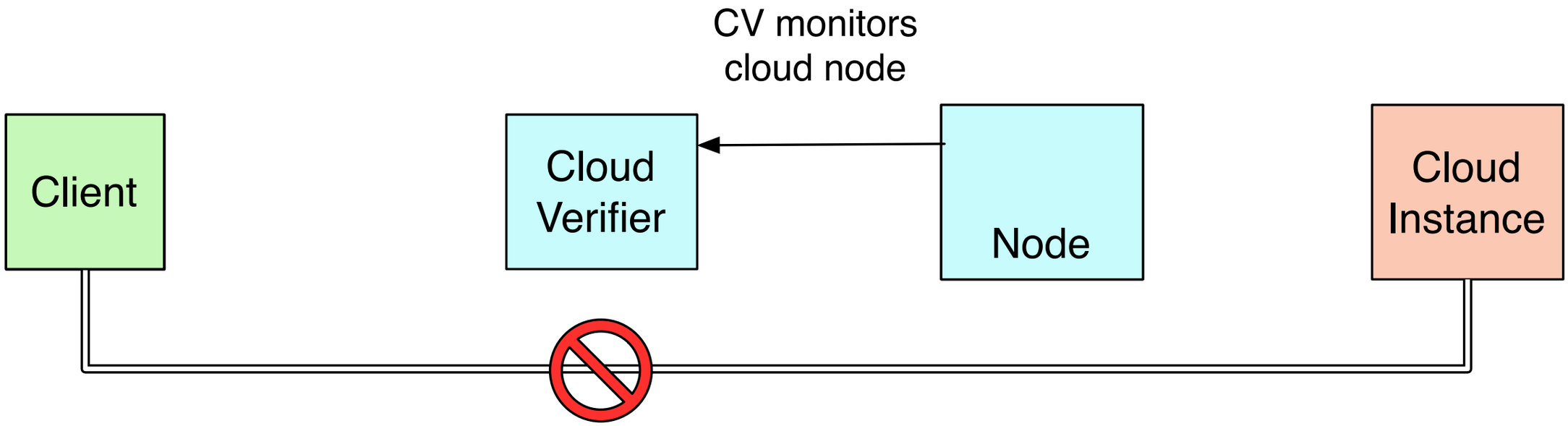
Cloud Verifier Overview

Cloud Anchor [CCSW 2010, TrustCom 2012]
+IVP in OpenStack [CSAW 2013]



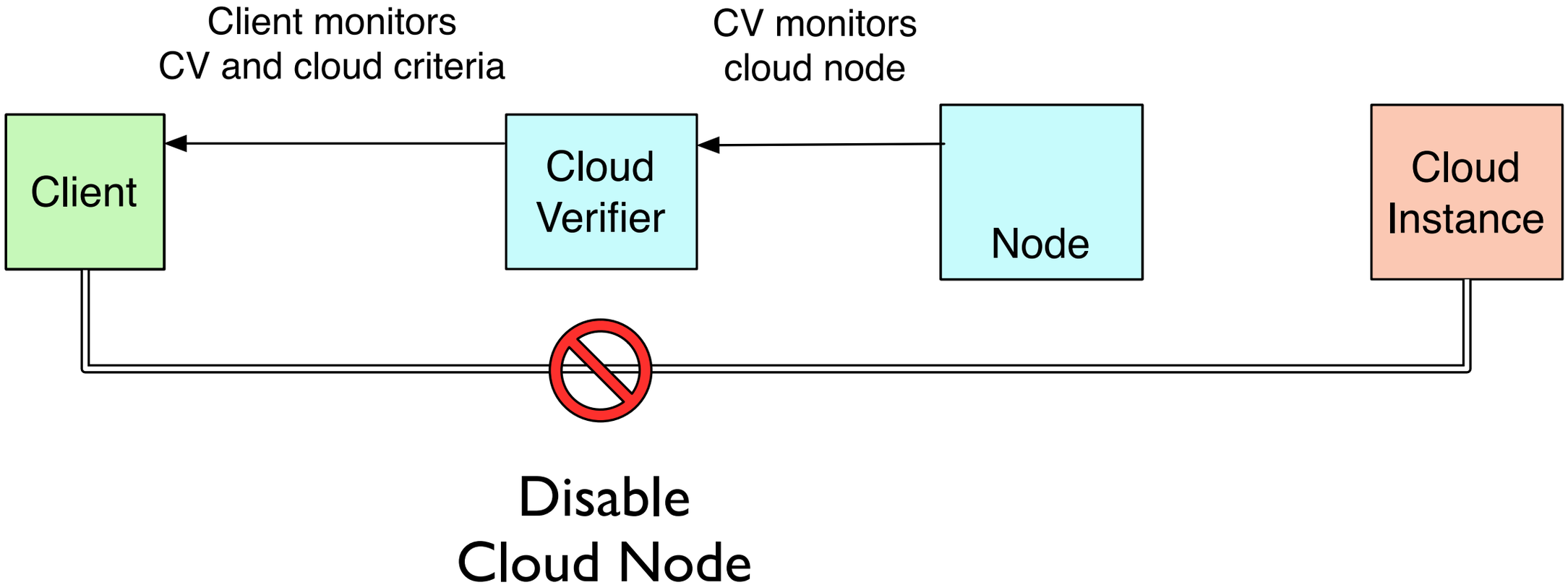
Cloud Verifier Overview

Cloud Anchor [CCSW 2010, TrustCom 2012]
+IVP in OpenStack [CSAW 2013]



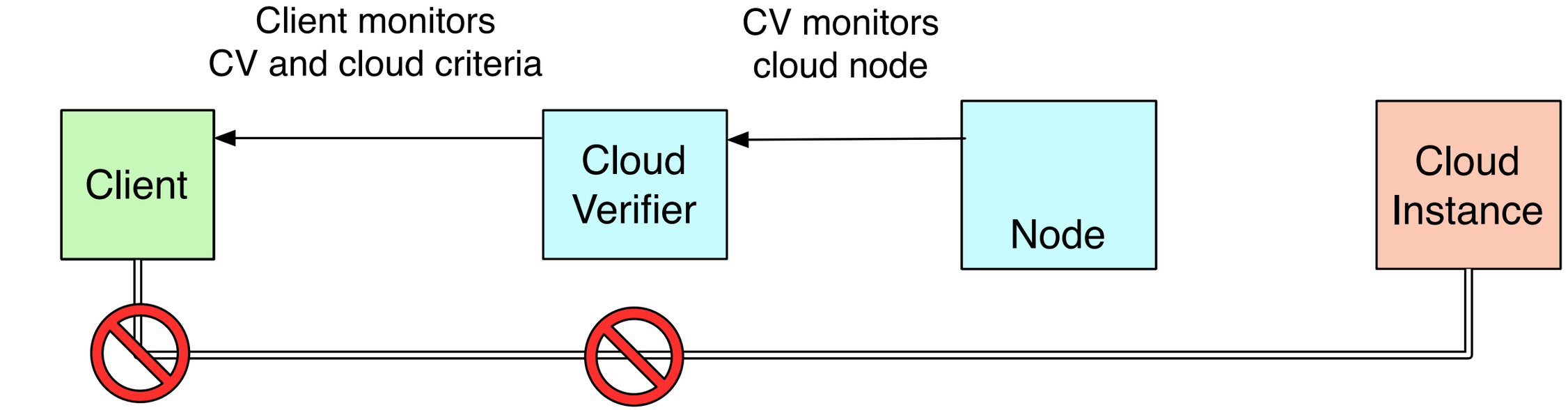
Cloud Verifier Overview

Cloud Anchor [CCSW 2010, TrustCom 2012]
+IVP in OpenStack [CSAW 2013]



Cloud Verifier Overview

Cloud Anchor [CCSW 2010, TrustCom 2012]
+IVP in OpenStack [CSAW 2013]

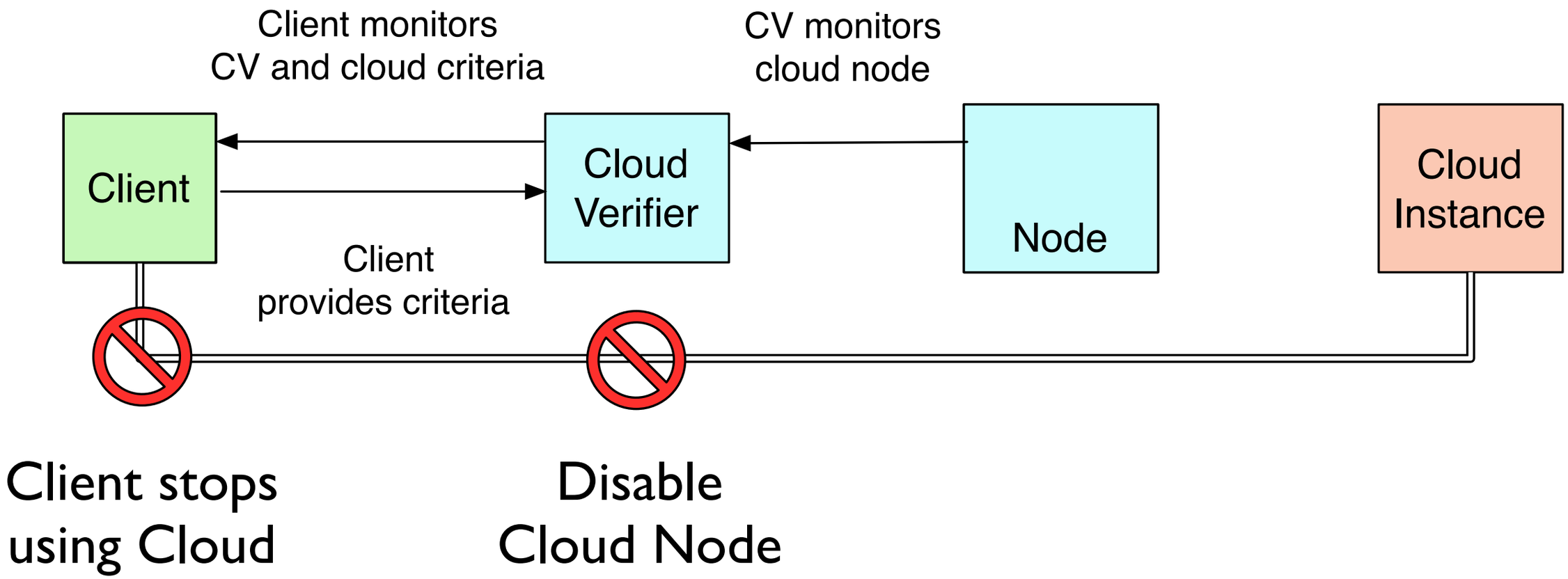


Client stops using Cloud

Disable Cloud Node

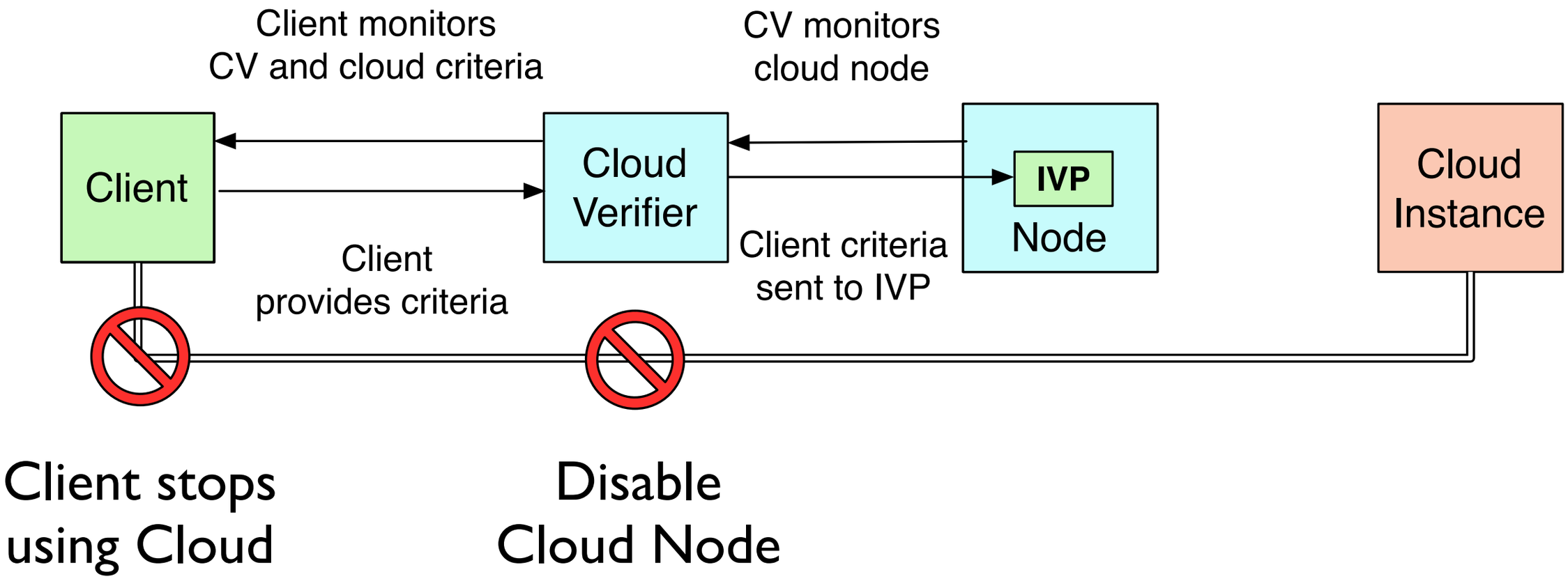
Cloud Verifier Overview

Cloud Anchor [CCSW 2010, TrustCom 2012] +IVP in OpenStack [CSAW 2013]



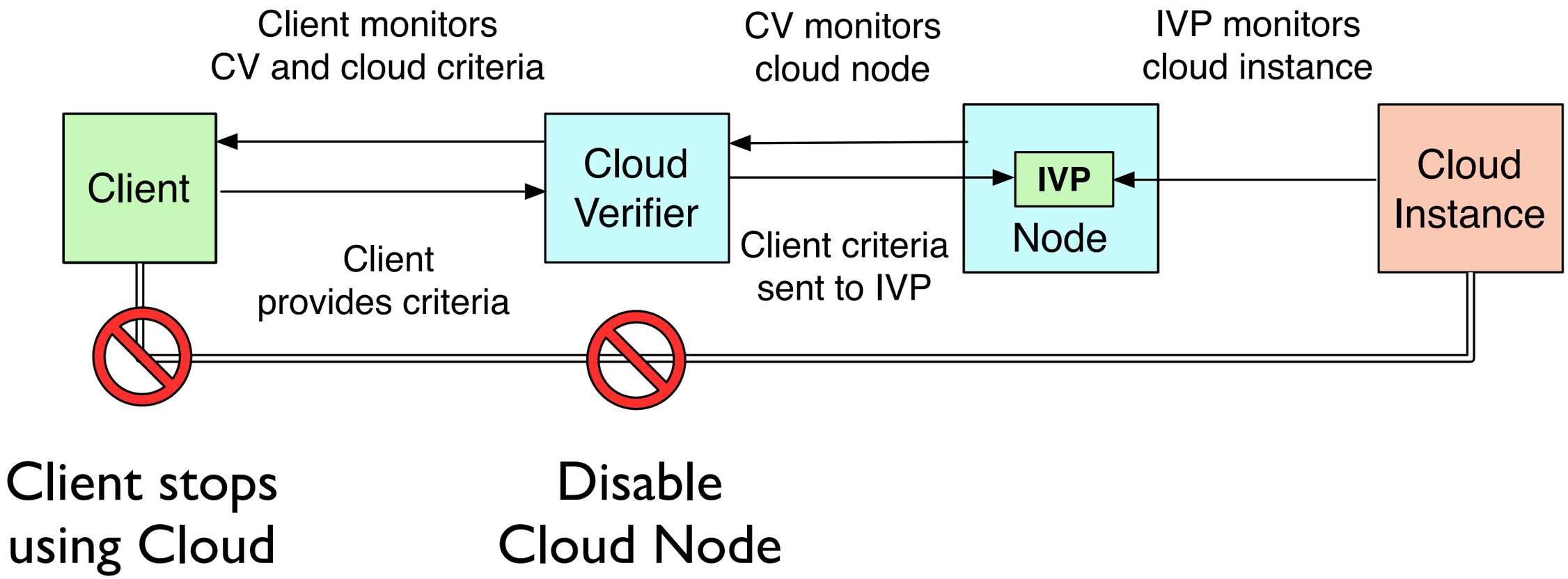
Cloud Verifier Overview

Cloud Anchor [CCSW 2010, TrustCom 2012] +IVP in OpenStack [CSAW 2013]



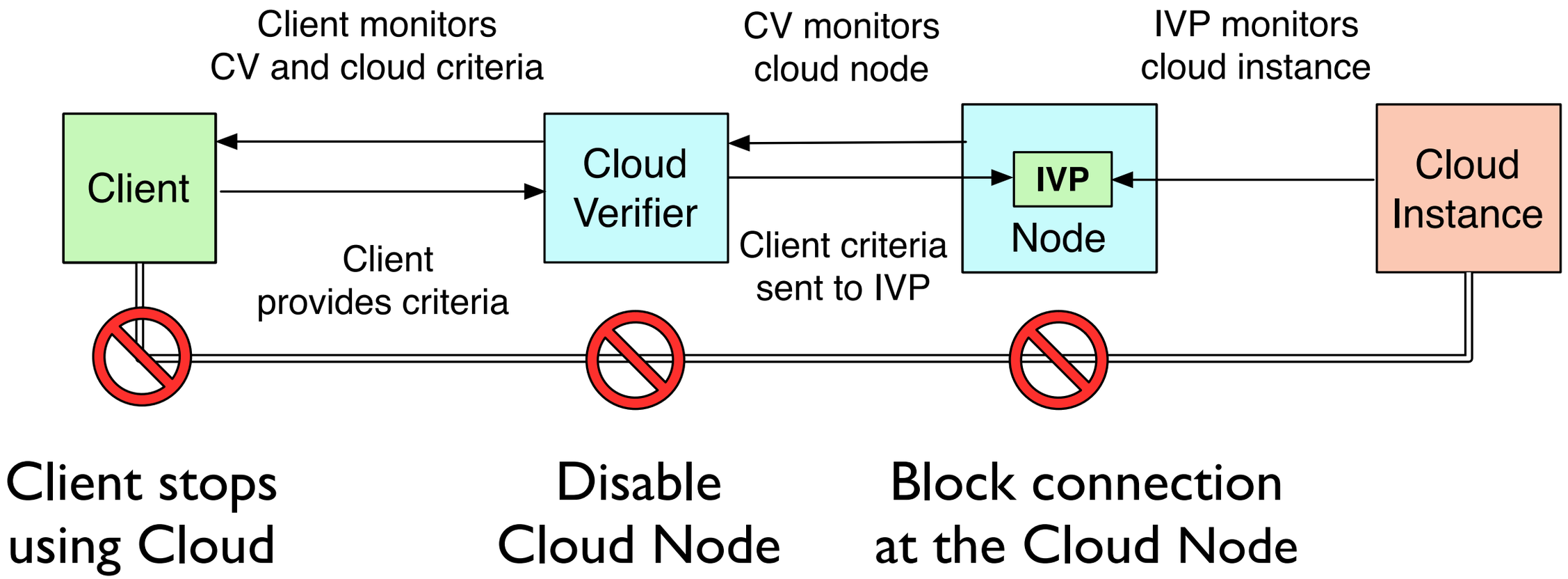
Cloud Verifier Overview

Cloud Anchor [CCSW 2010, TrustCom 2012] +IVP in OpenStack [CSAW 2013]



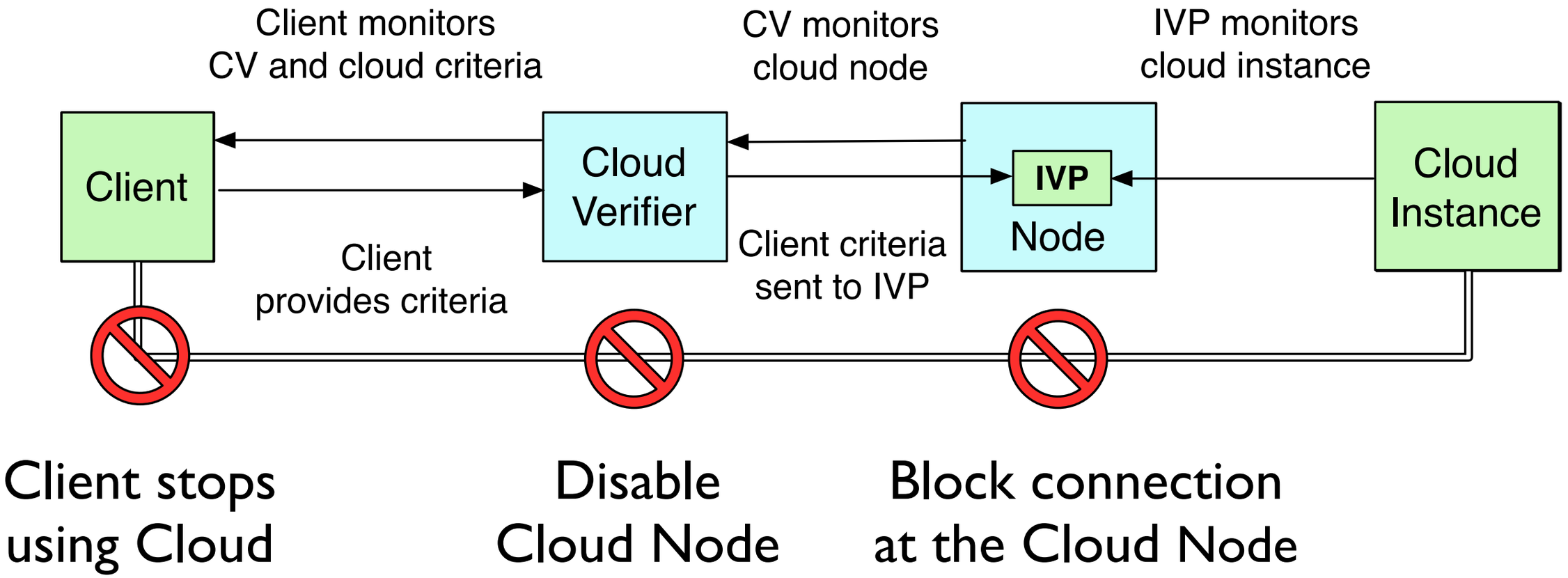
Cloud Verifier Overview

Cloud Anchor [CCSW 2010, TrustCom 2012] +IVP in OpenStack [CSAW 2013]

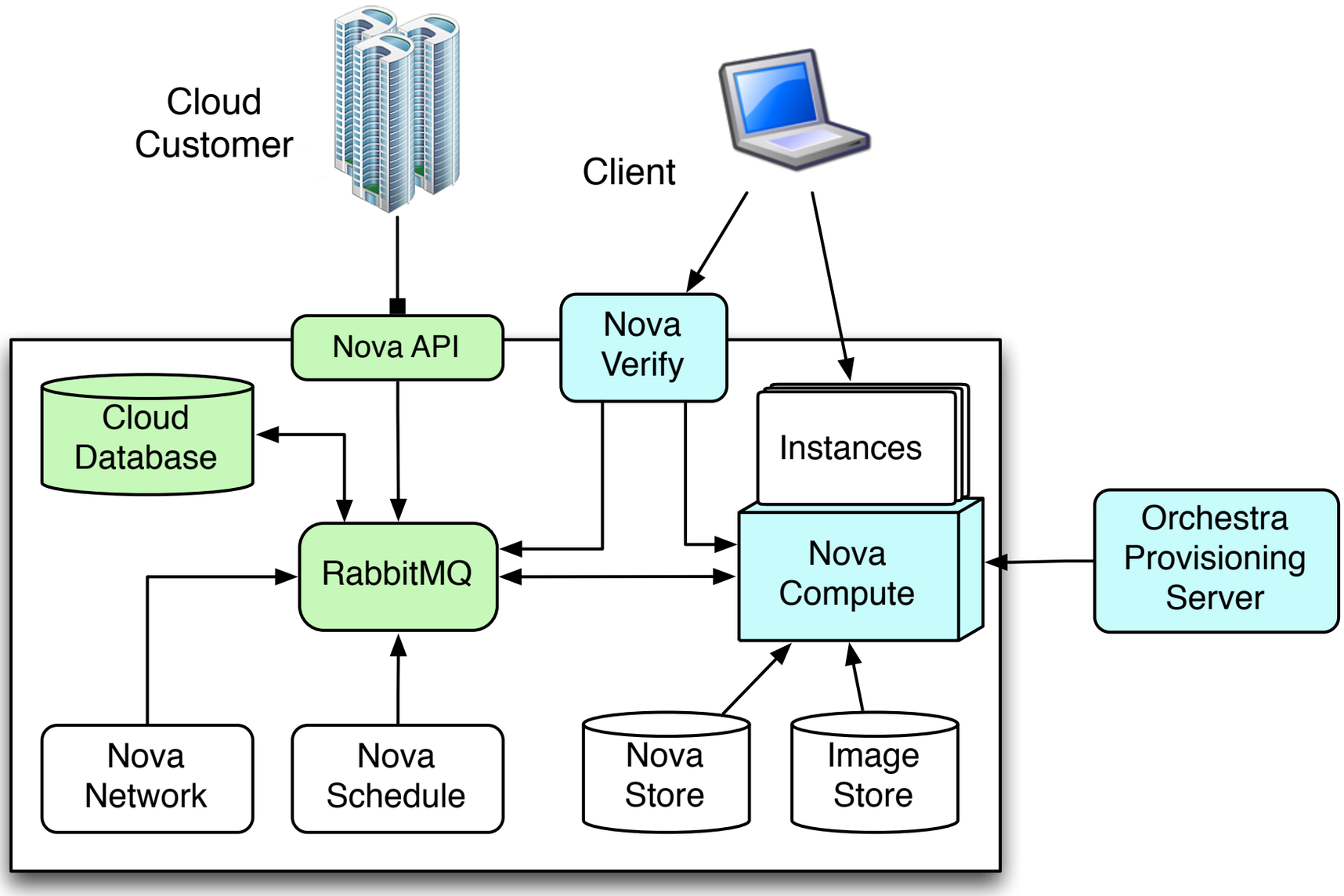


Cloud Verifier Overview

Cloud Anchor [CCSW 2010, TrustCom 2012] +IVP in OpenStack [CSAW 2013]



OpenStack Integration



Protocol Performance

- Ubuntu 12.04 OpenStack distribution testbed
  - Two Node blades + Cloud Services blade
- Join protocol introduces minor startup delay
  - Majority of overhead is TPM related
- RabbitMQ < 3% throughput slowdown

Node Join Protocol	1.68
TPM Quote	0.82
OpenSSL Node Key Generation	0.29
Node Certificate Generation	0.22
Communication	0.23
Others (read, write file etc.)	0.12

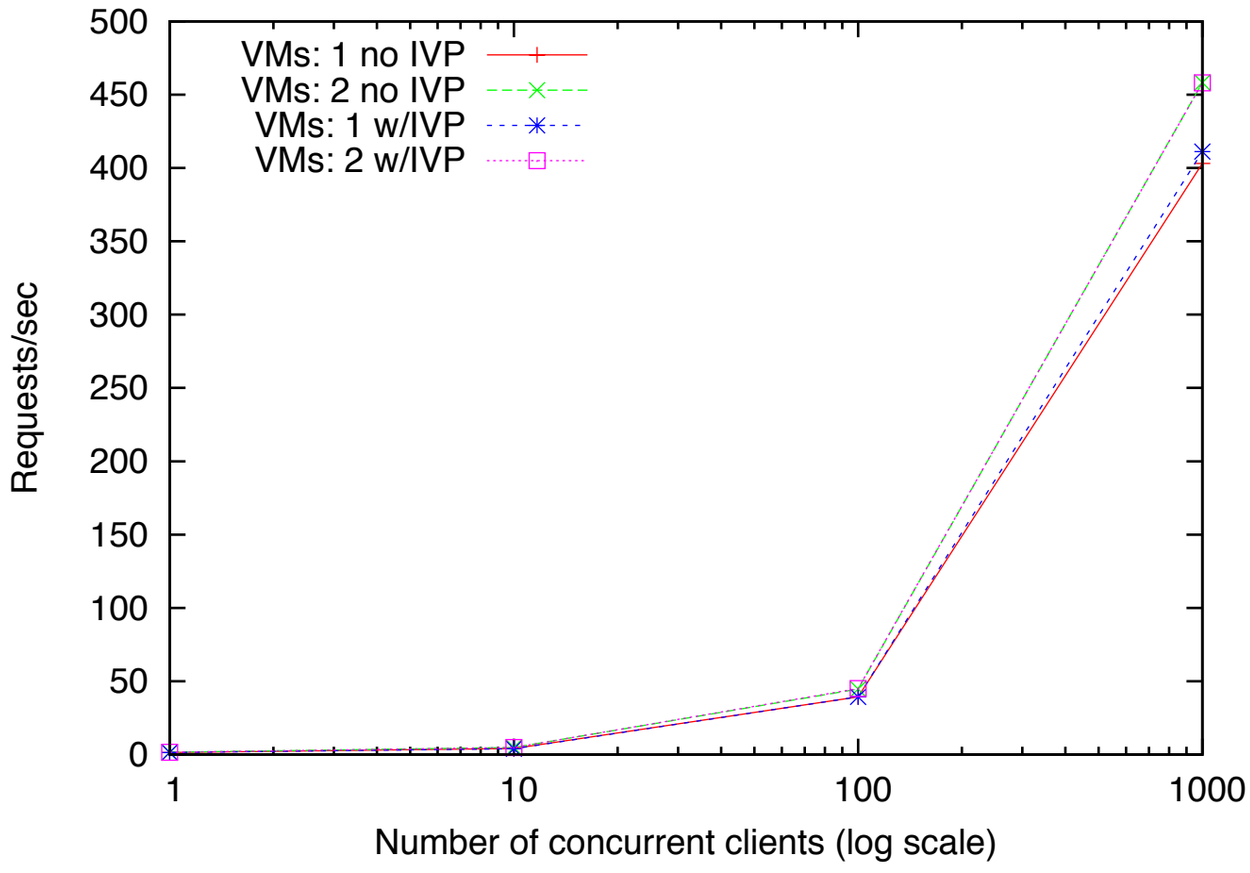
Client Criteria Registration	0.54
Openstack Processing	0.30
Instance Certificate Generation	0.22
Certificate Verification	0.04

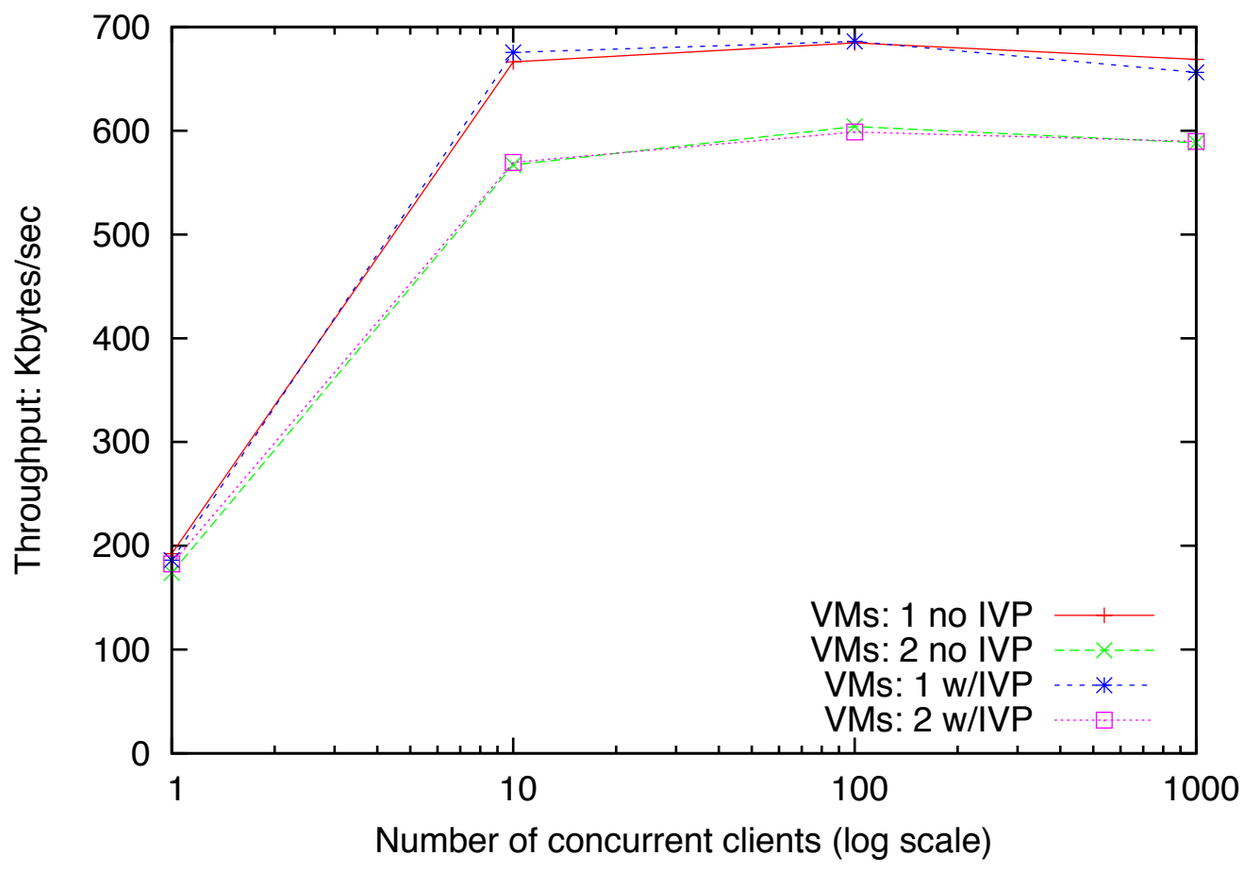
VM Startup Overhead Due to IVP	3.922
OpenSSL Key Injection	0.215
SELinux Binary Extraction	0.049
Module Initialization	3.658

Application Benchmarking

- Benchmarked two of the top Amazon AMIs
  - Ensure cloud criteria for inputs, policies, enforcement, code
  - Varied concurrency level of Apache Benchmark (ab)

Apache

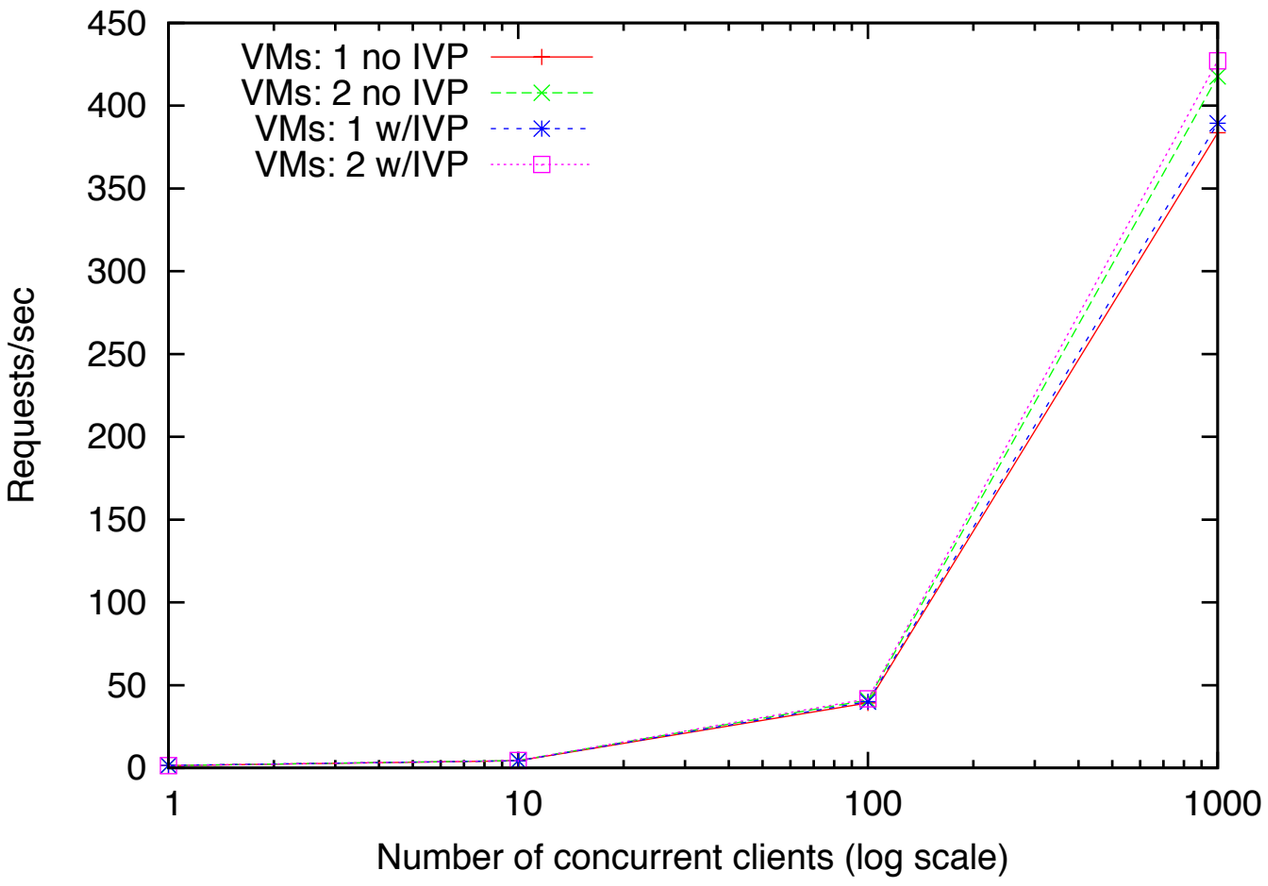


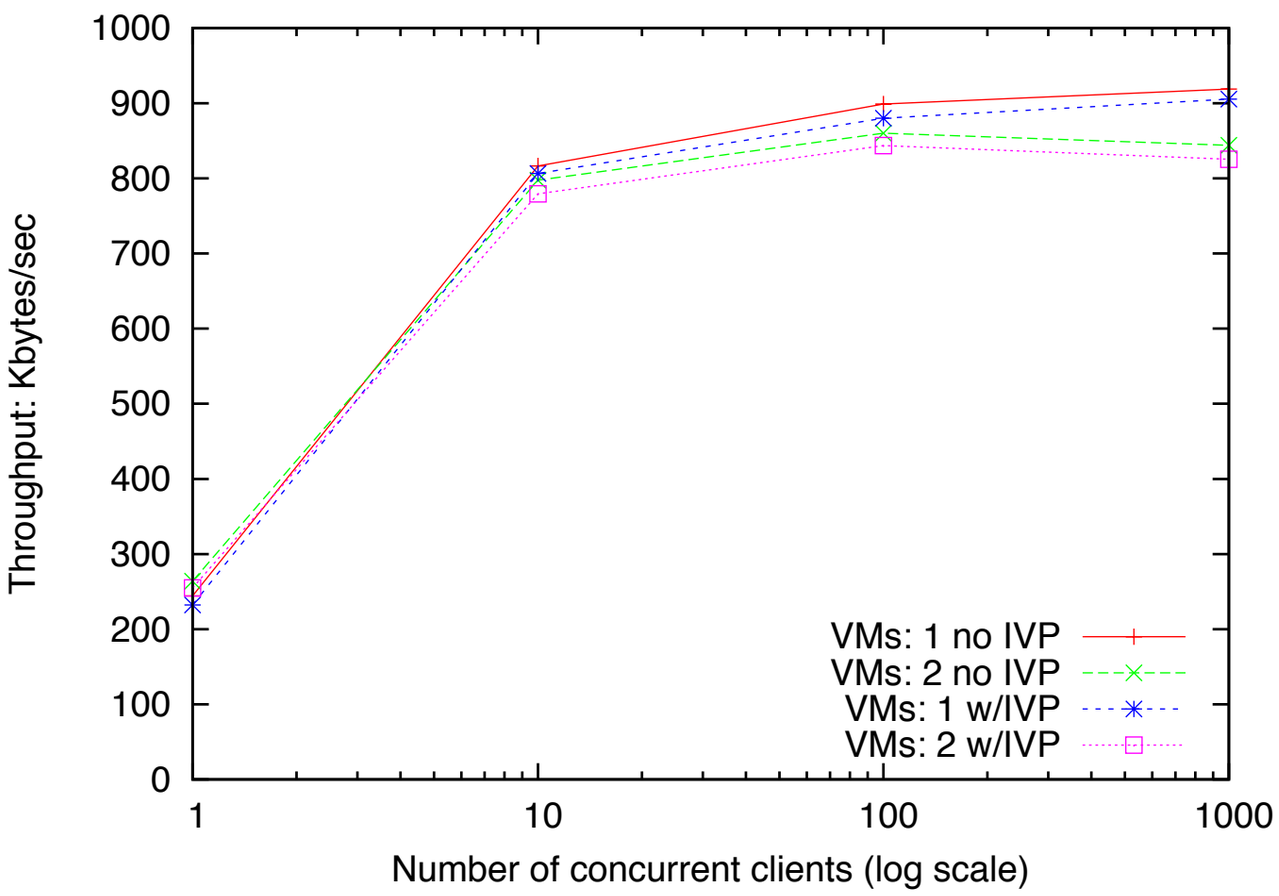


Application Benchmarking

- Benchmarked two of the top Amazon AMIs
  - Ensure cloud criteria for inputs, policies, enforcement, code
  - Varied concurrency level of Apache Benchmark (ab)

Nginx

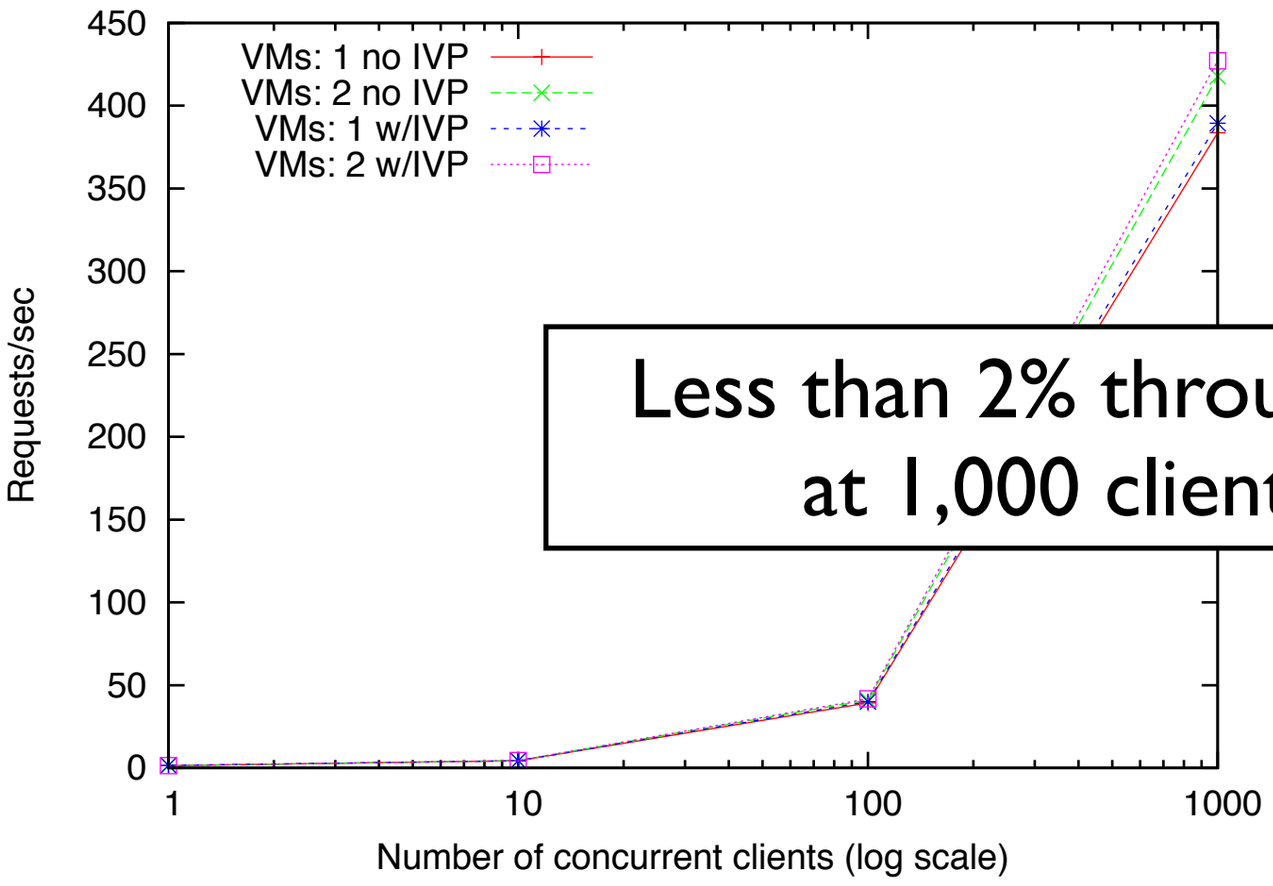


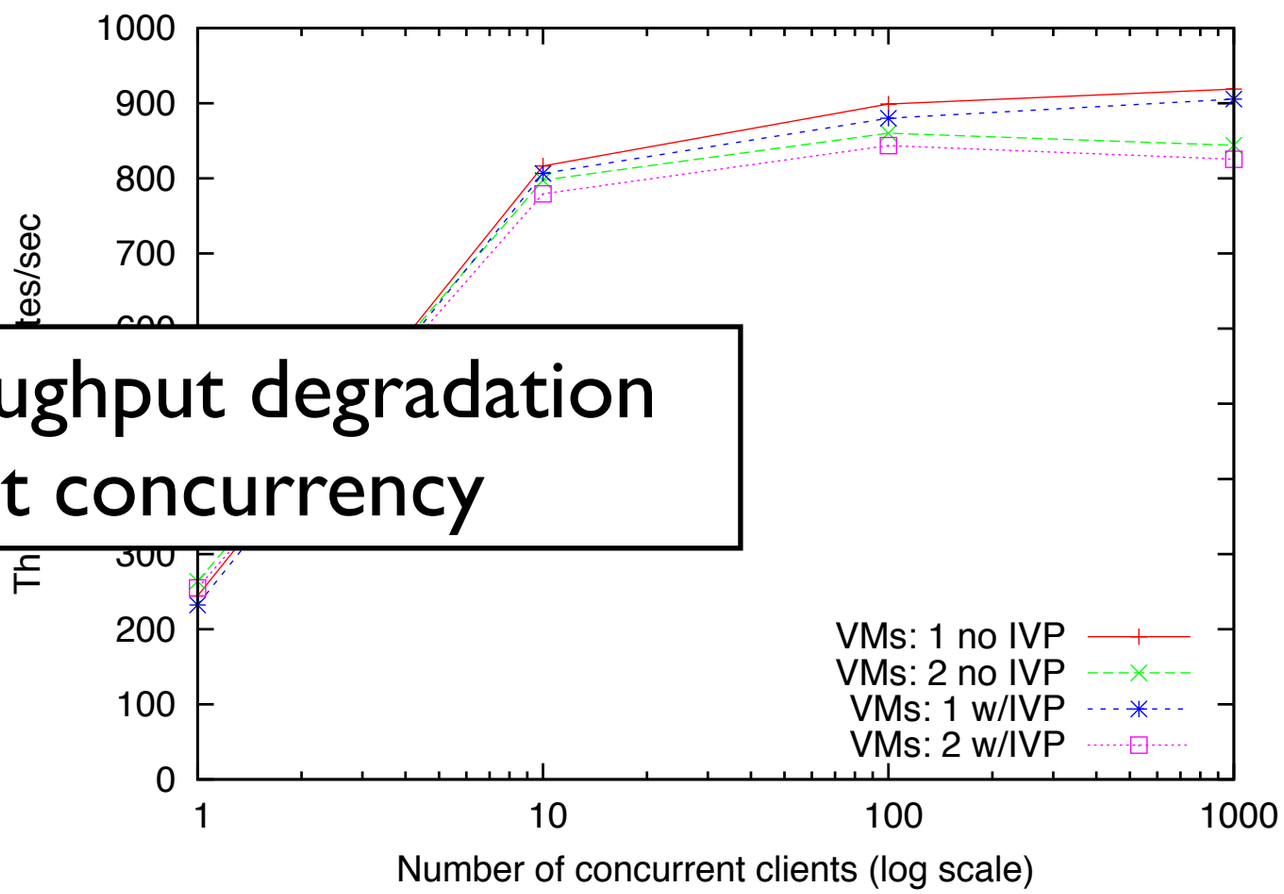


Application Benchmarking

- Benchmarked two of the top Amazon AMIs
  - Ensure cloud criteria for inputs, policies, enforcement, code
  - Varied concurrency level of Apache Benchmark (ab)

Nginx





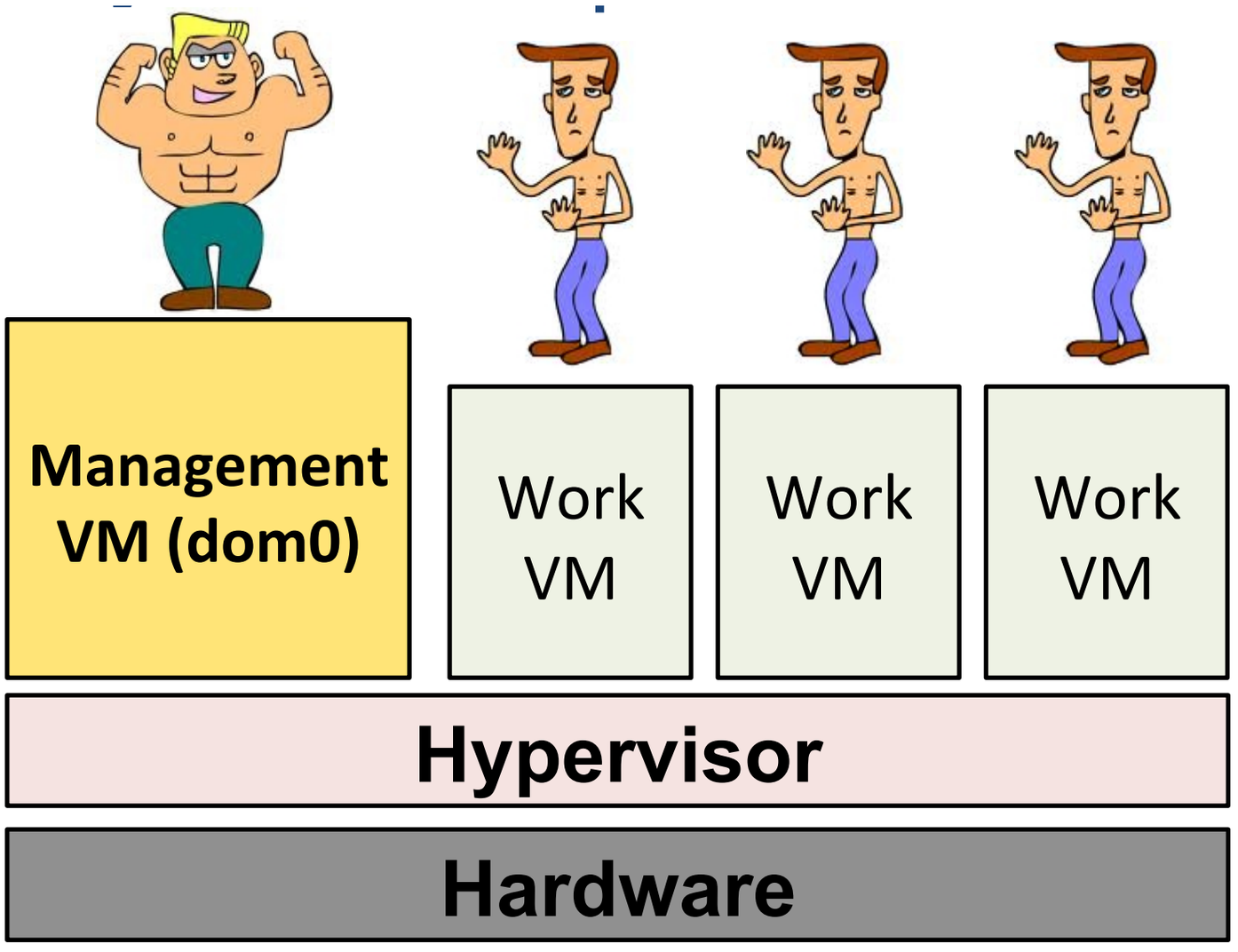
Less than 2% throughput degradation at 1,000 client concurrency

Customer-Driven Monitoring

- CV/IVP Limitation
  - IVP must be trusted by cloud vendor
  - Part of management VM
- What if you need to perform monitoring that the cloud vendors will not support?

Self-Service Clouds

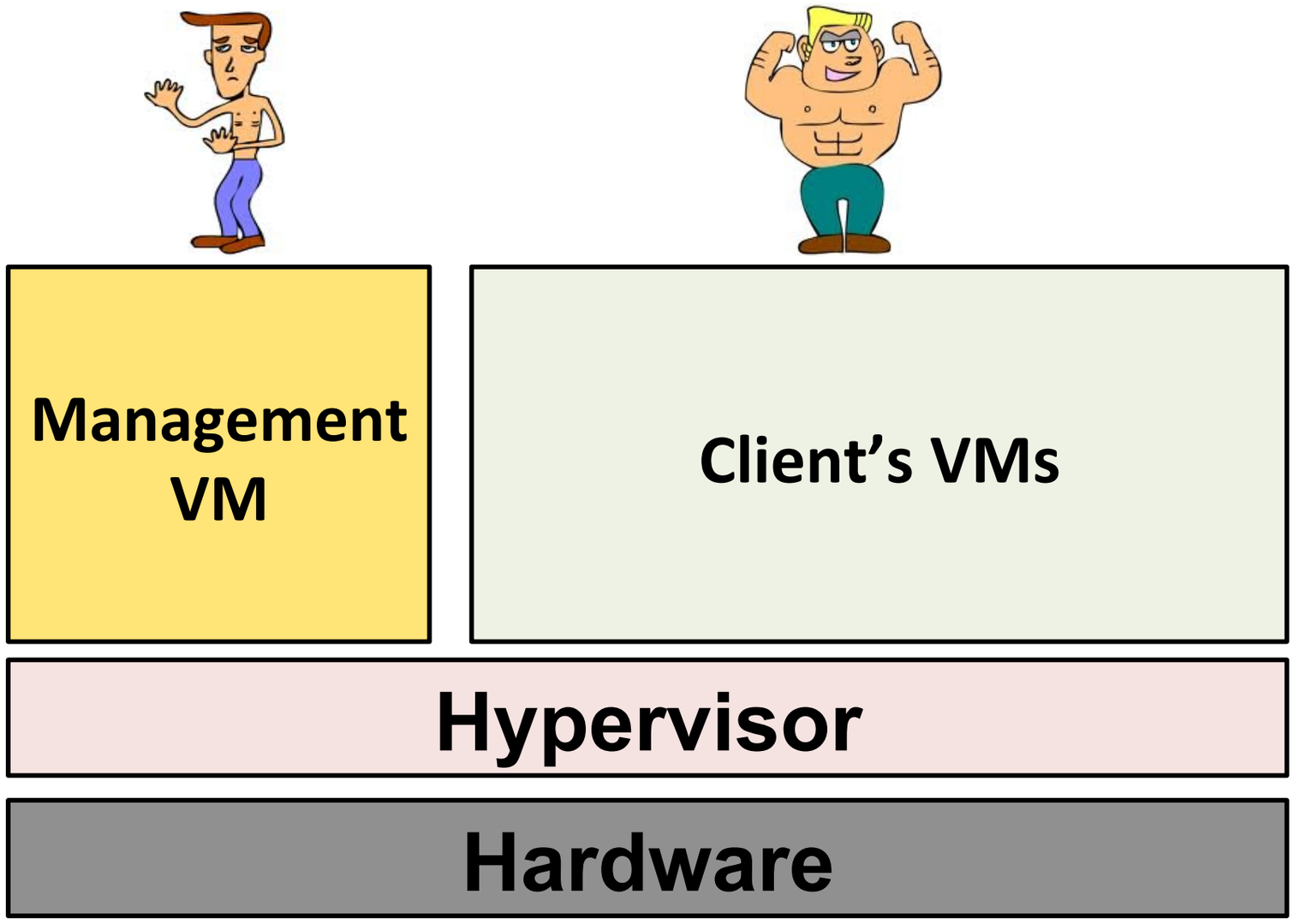
- Customizable cloud platform stack *[CCS 2012]*



Slides courtesy of Vinod Ganapathy

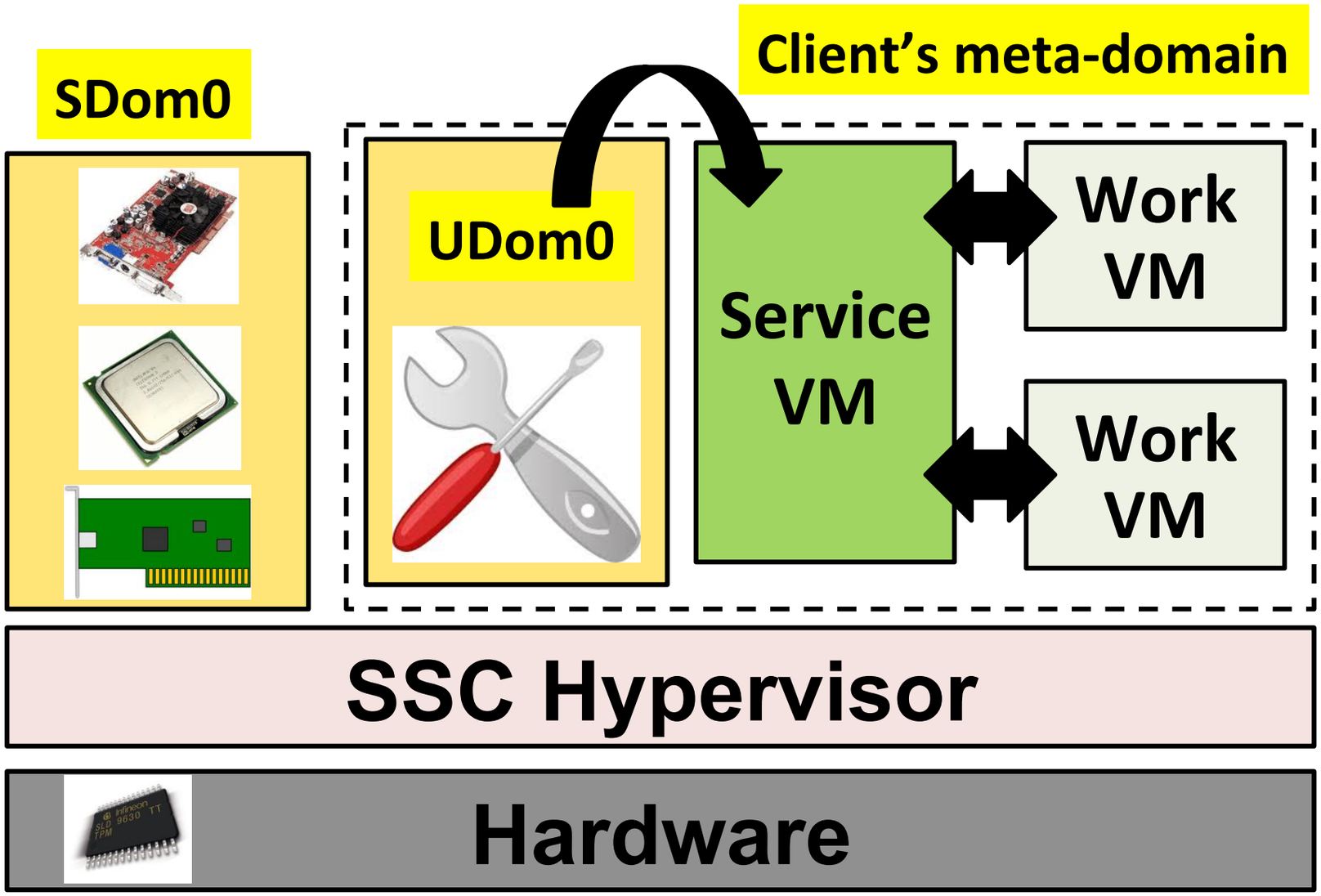
Self-Service Clouds

- Customizable cloud platform stack *[CCS 2012]*



Self-Service Clouds

- Customizable cloud platform stack *[CCS 2012]*
  - UDom0 boots customer-defined Service VMs



Equipped with a Trusted Platform Module (TPM) chip

Limitations and Next Steps

- Accuracy of monitoring depends on quality of software
  - Same security problems we have been solving
  - Minimize TCB, Harden Software, etc.
- Effectiveness of monitoring requires understanding attack paths and risks
  - Attack graphs: Analyze information flows *[ESORICS 2011] [ACSAC 12]*
  - Semantic gap: Monitor internal to guest instance as well
- We discussed monitoring a single component
  - Should expand to inter-component requirements
  - Shared reference monitor is one proposal *[ACSAC 2006]*

Open Questions

- Clouds essentially offer a hardware management utility
  - If you have "enough" hardware, then is this very valuable?
- Problem is administration of your instances
  - Cloud vendors only help minimally
    - Preconfigured instances (may have problems)
- Can utility provide world-class administration at scale?
  - Load-time: configure specialized defenses for your application
  - Runtime: detect and adapt your application to evade attacks
- Solving these problems are not specific to cloud computing as currently envisioned

Summary

- We can provide fine-grained and customizable monitoring of cloud instances
  - Even that does not disturb the cloud vendors threat model
- Currently, cloud computing makes security more difficult
  - Insiders, publishers, more services, more policies
  - But, cloud vendors lack knowledge of how to help customers in a cost-effective way
- Long-term Goal: Automated, world-class administration that leverages such monitoring
  - For now, use monitoring for one threat at a time

Thank you

Trent Jaeger (tjaeger@cse.psu.edu)

http://www.trentjaeger.com/

SIIS Laboratory (http://siis.cse.psu.edu)

References

- [CCS 2011] S. Bugiel, *et al.* AmazonIA: when elasticity snaps back, *ACM CCS*, 2011.
- [USENIX Sec 2012a] H. Vijayakumar, *et al.* STING: finding name resolution vulnerabilities in programs. *USENIX Security Symposium*, 2012.
- [USENIX Sec 2012b] N. Santos, *et al.* Policy-sealed data: a new abstraction for building trusted cloud service, *USENIX Security Symposium*, 2012.
- [CCS 2012] S. Butt, *et al.* Self-service cloud computing, *ACM CCS*, 2012.
- [TRUST 2012] J. Schiffman, H. Vijayakumar, T. Jaeger. Verifying system integrity by proxy, *TRUST 2012*.
- [CSAW 2013] J. Schiffman, *et al.* Cloud Verifier: Verifiable auditing service for IaaS clouds, Cloud Security Auditing Workshop, 2013.
- [ASIACCS 2012] S. Bleikertz, *et al.* Secure Cloud Maintenance -- Protecting workloads against insider attacks, ACM ASIACCS, 2012.
- [CCSW 2010] J. Schiffman *et al.* Seeding clouds with trust anchors, ACM Cloud Computing Security Workshop, 2010.

References

- [TR 2013] Y. Sun, *et al*. CloudArmor: protecting cloud instances by validating service operations, Penn State NSRC, 2013.
- [CCS 2009] T. Ristenpart, *et al*. Hey you, get off my cloud, *ACM CCS*, 2009.
- [ACSAC 2007] L. St. Clair, *et al*. Establishing and maintaining system integrity via root of trust installation, *ACSAC*, 2007.
- [IEEE S&P 2011] J. Schiffman, *et al*. Network-based root of trust for installation, IEEE S&P, Jan/Feb 2011.
- [SOSP 2011] F. Zhang, *et al*. Cloudvisor: retrofitting protection of virtual machines in multi-tenant cloud with nested virtualization, *SOSP*, 2011.
- [ESORICS 2011] S. Bleikertz, *et al*. Automated information flow analysis of virtualized infrastructures, ESORICS, 2011.
- [ACSAC 2012] D. Muthukumaran, *et al*. Transforming commodity security policies to enforce Clark-Wilson integrity, ACSAC, 2012.
- [ACSAC 2006] J. McCune, *et al*. Shamon: A system for distributed mandatory access control, ACSAC 2006.
- [TrustCom 2011] I. Abbadi. Cloud Trust Anchors, TrustCom, 2011.

Acknowledgements

- Past and Current Penn State Students
  - Joshua Schiffman, Yuqiong Sun, Hayawardh Vijayakumar, Thomas Moyer
- Collaborators
  - Patrick McDaniel, Vinod Ganapathy, Abhinav Srivastava
- Authors of other fine work that influenced the talk