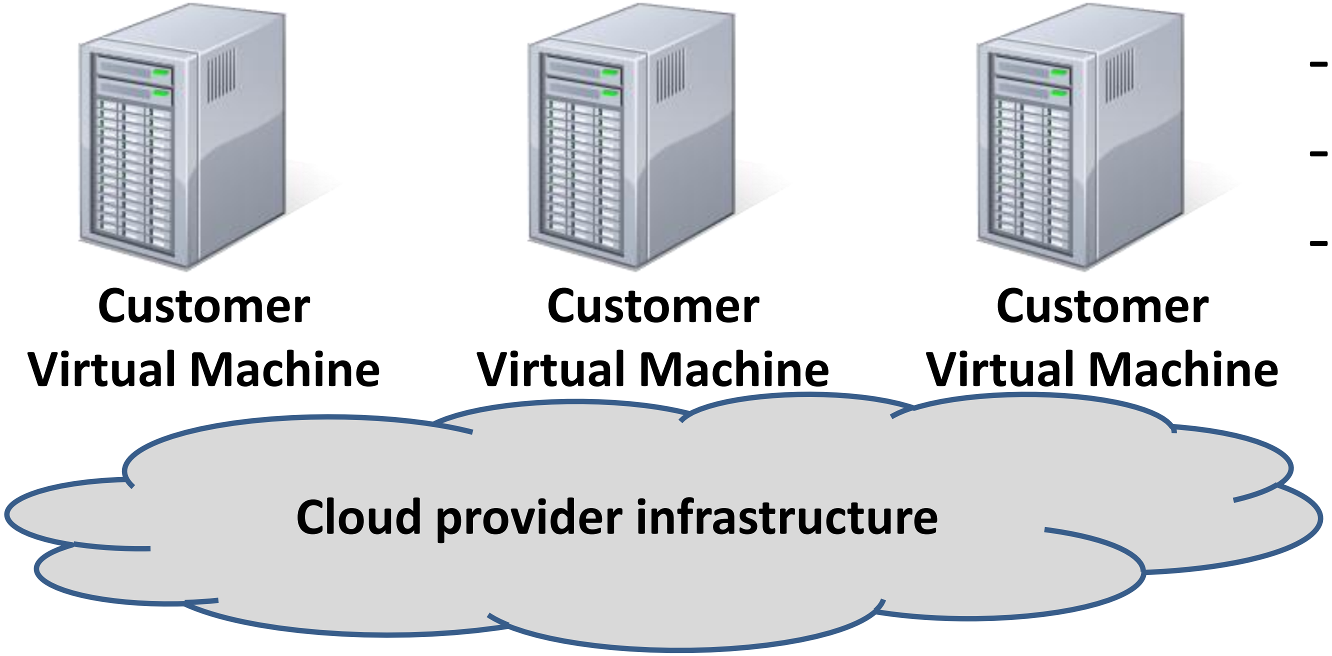
Computer Meteorology: Monitoring Compute Clouds

Lionel Litty, H. Andrés Lagar-Cavilla,
David Lie

University of Toronto

Infrastructure as a Service (IaaS)



Customer Virtual Machine

Customer Virtual Machine

Customer Virtual Machine

Cloud provider infrastructure

Examples:

-Amazon EC2

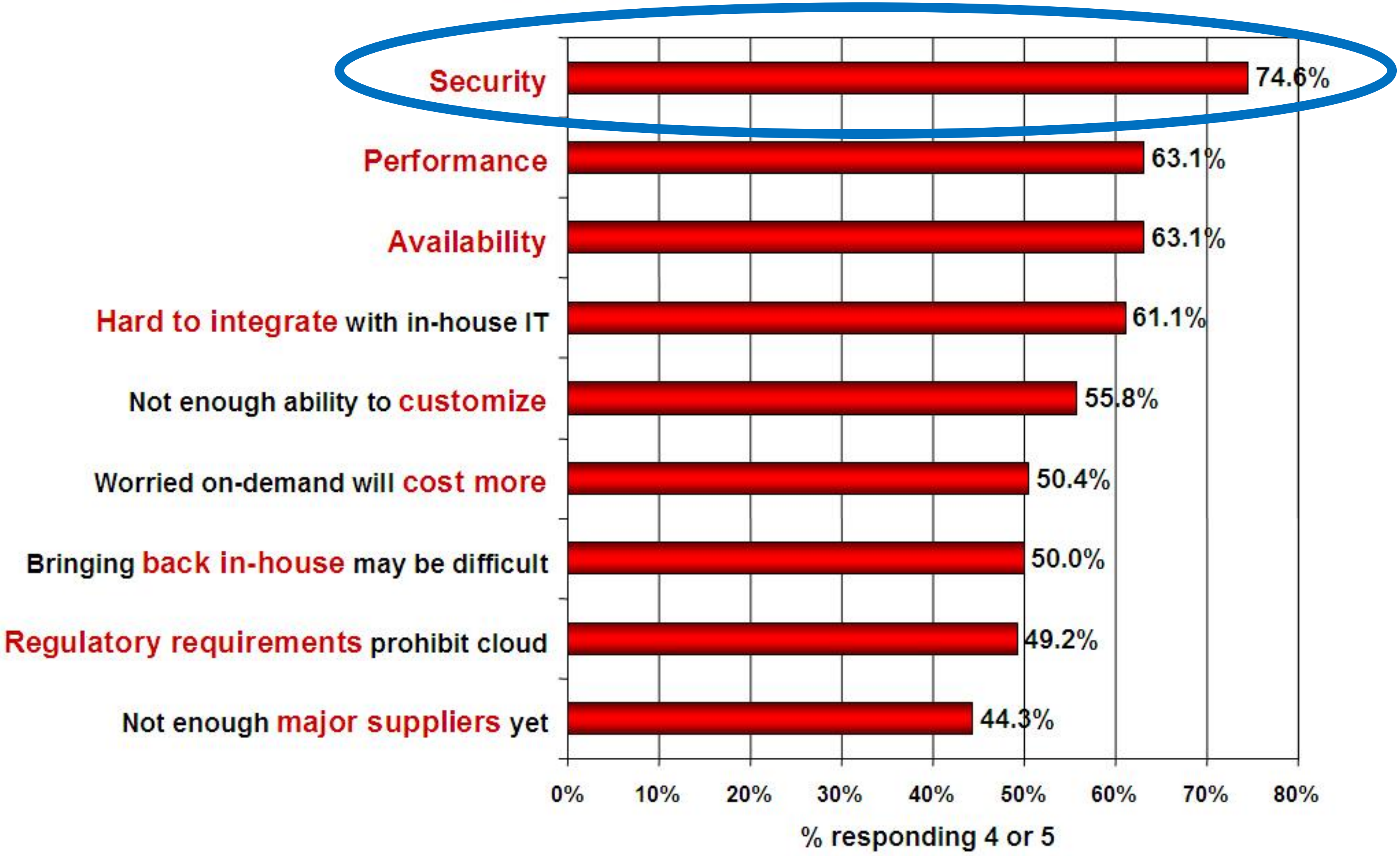
-GoGrid

-Mosso

-...

Q: Rate the challenges/issues ascribed to the 'cloud'/on-demand model

(1=not significant, 5=very significant)



Source: IDC Enterprise Panel, August 2008 n=244

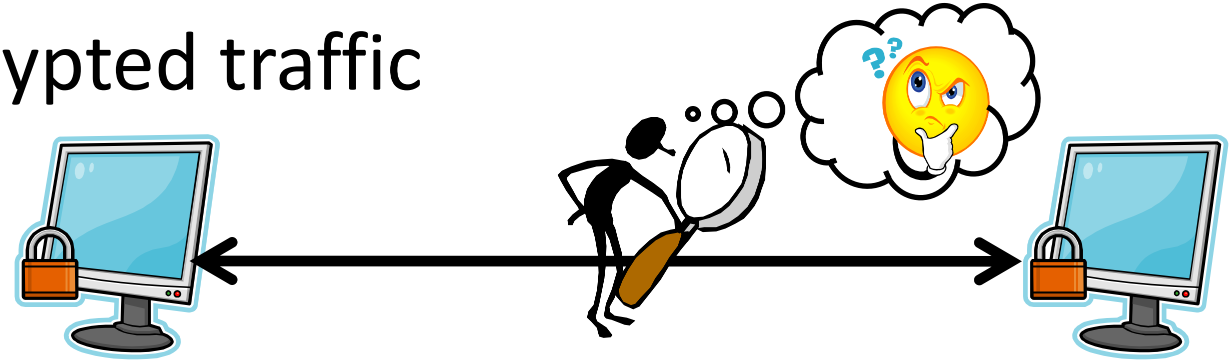
Security

- Miscreants can abuse the cloud provider’s resources:
  - Spam.
  - Use infrastructure to attack other computers.
  - Hosting illegal content.
- This has consequences for the cloud provider:
  - Damage to reputation.
  - Technical consequences: Shared IPs blacklisted.
  - Potential legal concerns.

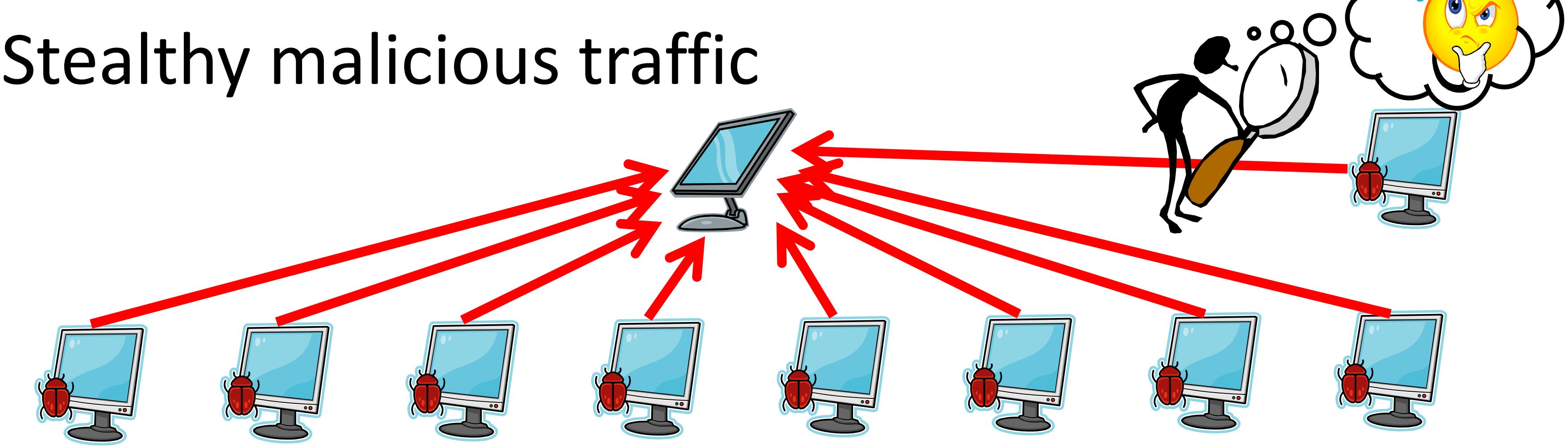
Solutions?

Network monitoring (NM) has limitations:

- Encrypted traffic



- Stealthy malicious traffic

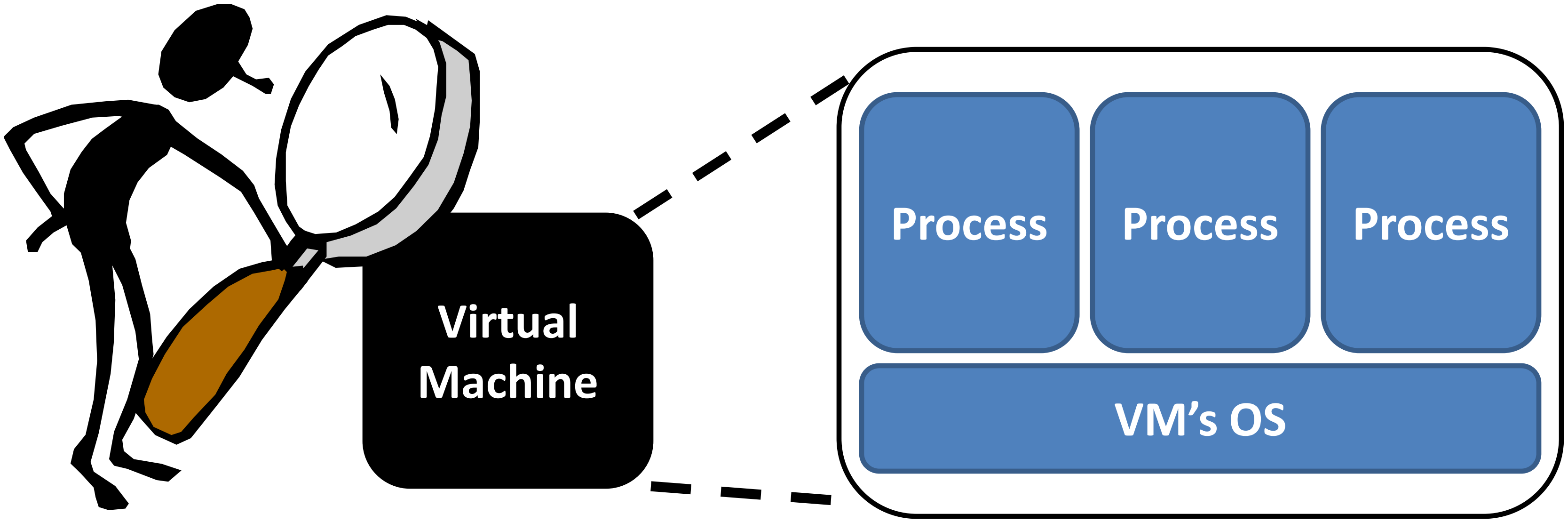


Distributed attack using botnet.

ISPs use NM and have done poorly.

Unlike ISPs, cloud providers control the execution platform:
Can they use this to their advantage?

Introspection



Virtual Machine

Process

Process

Process

VM's OS

Reductionist approach: understand a complex system by understanding its parts.

- Identify processes.
- Analyze the behavior of each process.

Non-malicious and Malicious VMs

- Non-malicious: may be vulnerable, not yet compromised.
- Malicious: under miscreant control.
  - Attacker can blur boundaries between processes.
- Tamper-evident monitor:
  - Either report accurate information
  - Or report that it cannot obtain accurate information.

Introspection properties

- Power

  Can it see everything?

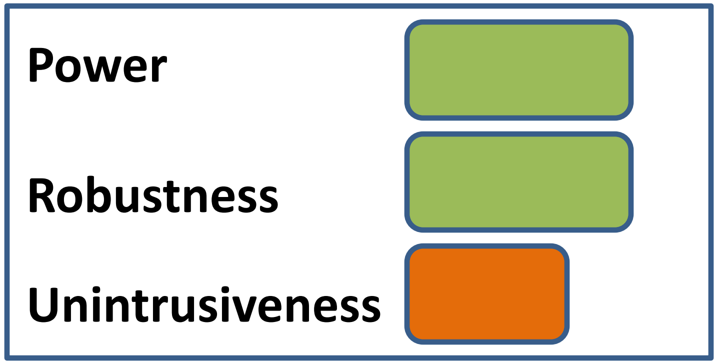
- Robustness

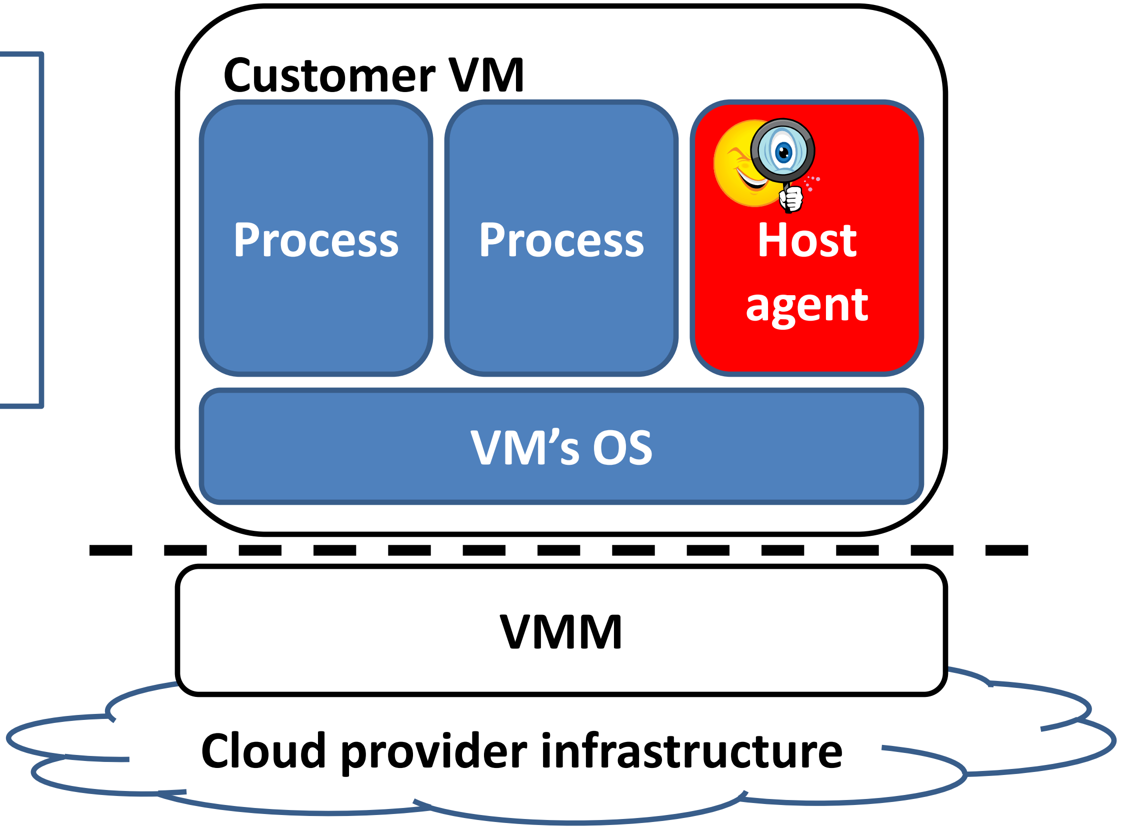
  Is it resilient to changes in the monitored system?

- Unintrusiveness

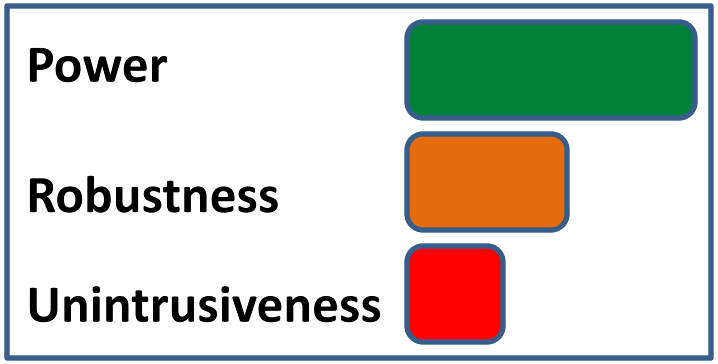
  Can it negatively impact the monitored system?

Host agent


Power
Robustness
Unintrusiveness


Customer VM
Process
Process
Host agent
VM's OS
VMM
Cloud provider infrastructure

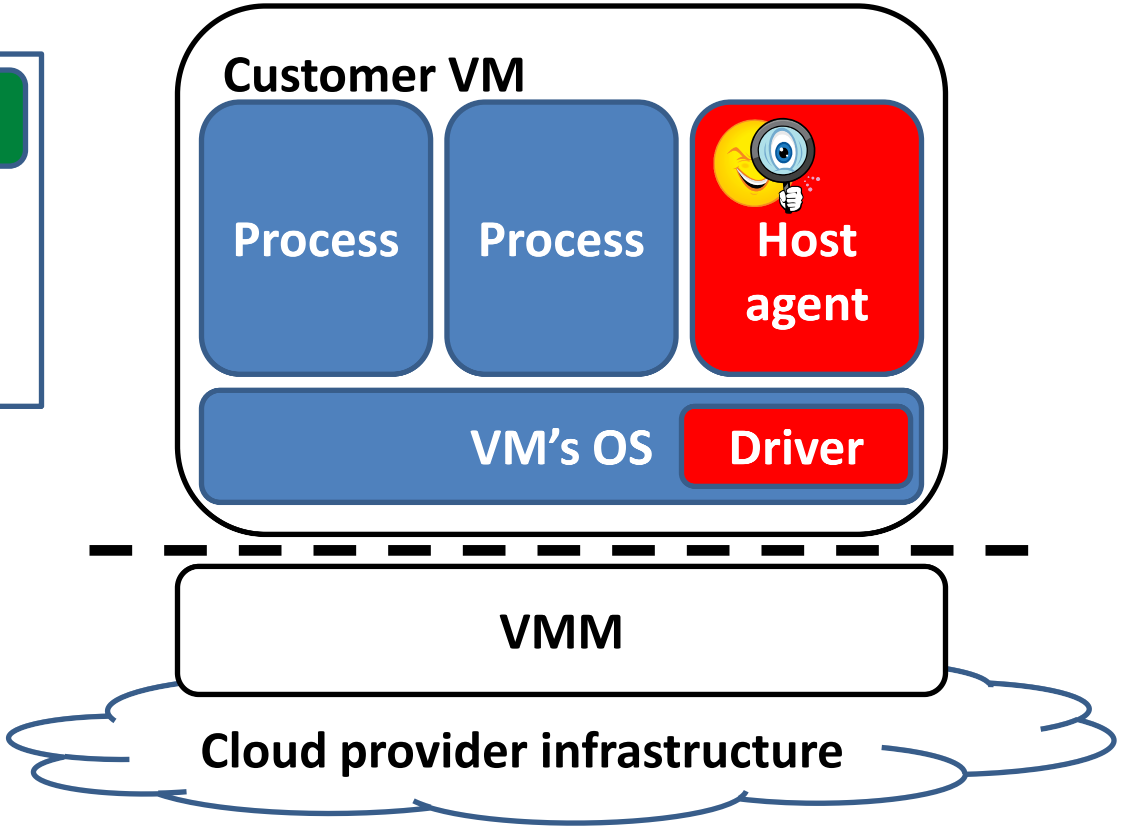
Host agent w/ driver



Power

Robustness

Unintrusiveness



Customer VM

Process

Process

Host agent

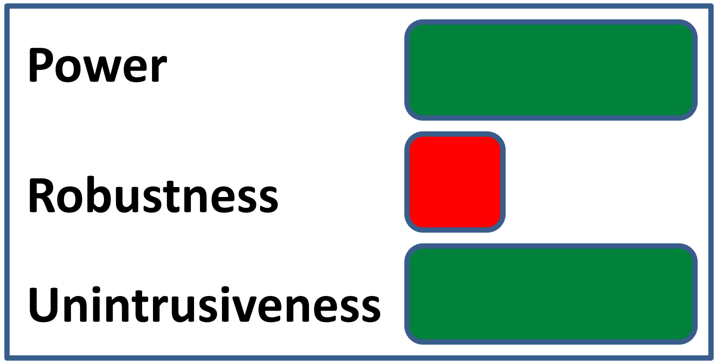
VM's OS

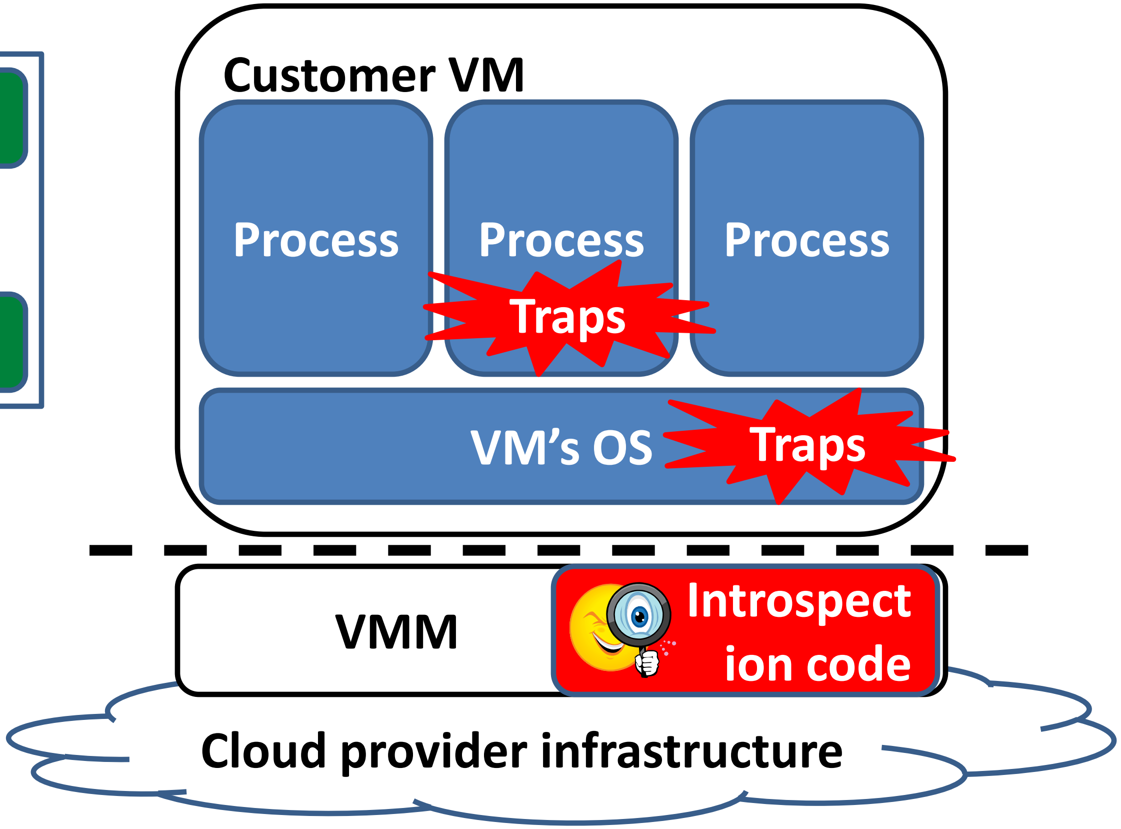
Driver

VMM

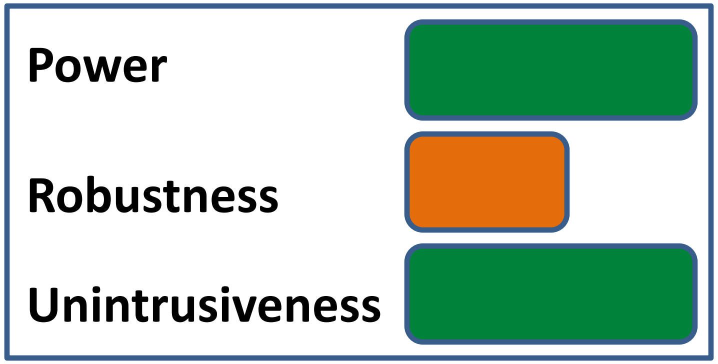
Cloud provider infrastructure

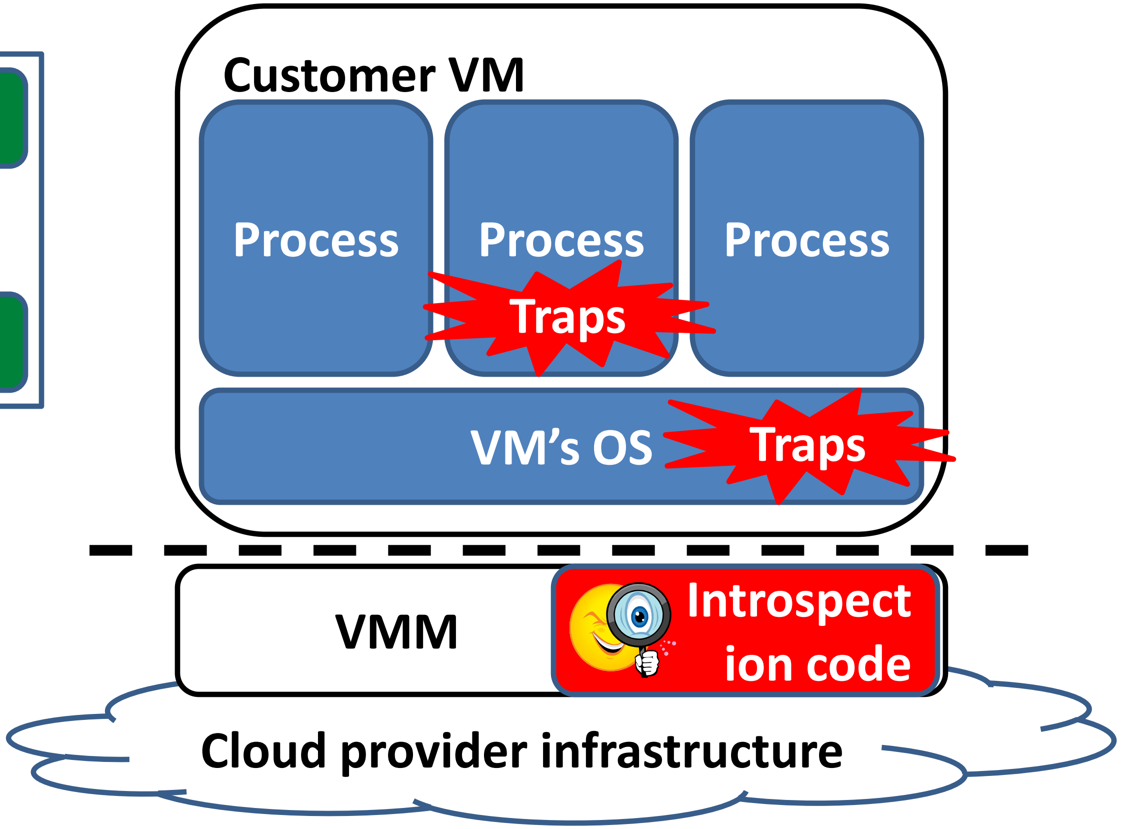
Trap & Inspect



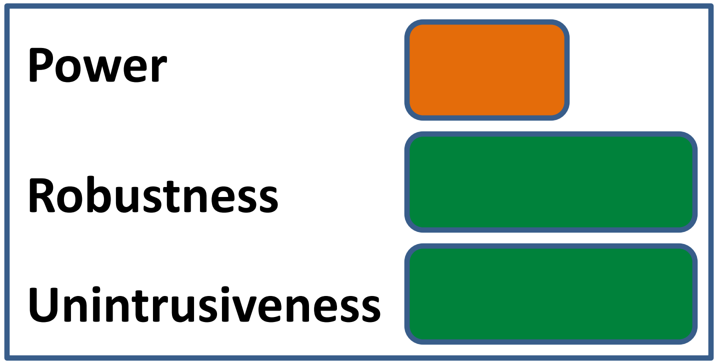


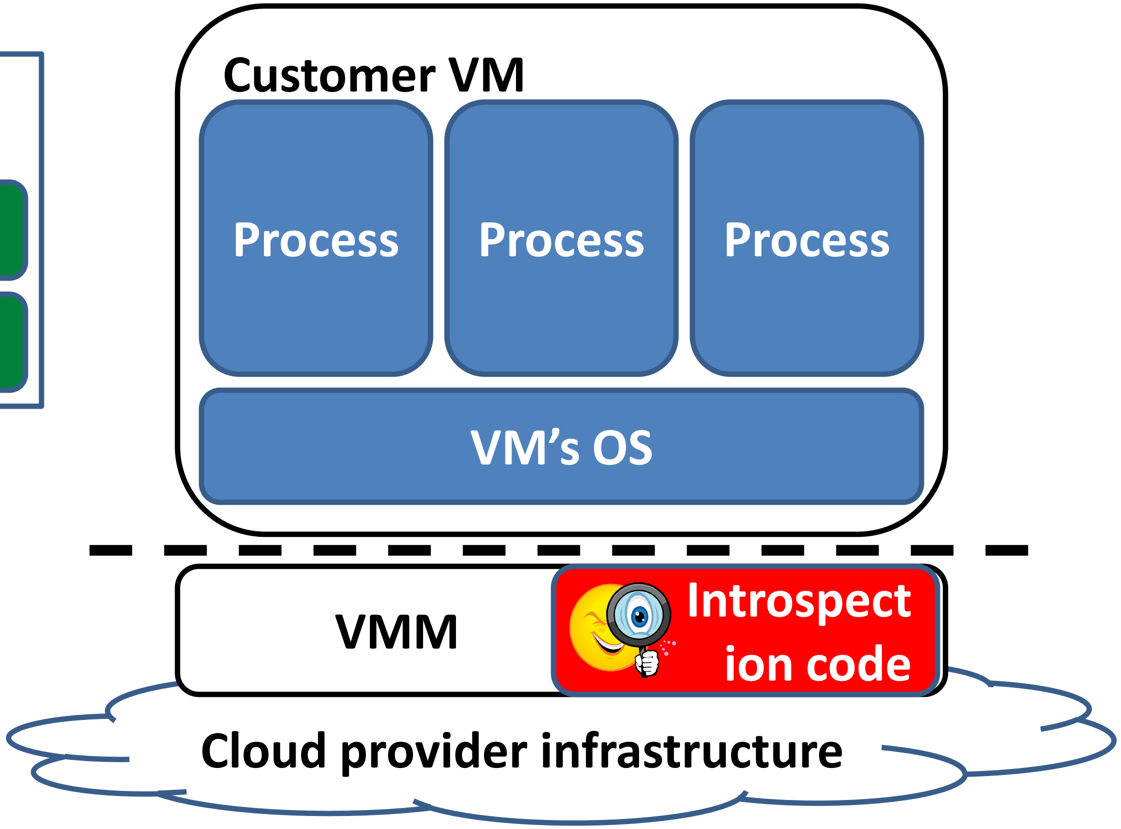
Checkpoint & Rollback





Architectural Introspection





Summary of introspection approaches

	Power	Unintrusiveness	Robustness
Host agent	Good	Poor	Good
Host agent w/ driver	Best	Worst	Poor
Trap & Inspect	Best	Best	Worst
Checkpoint & Rollback	Best	Best	Poor
Architectural monitoring	Poor(?)	Best	Best

Introspection example

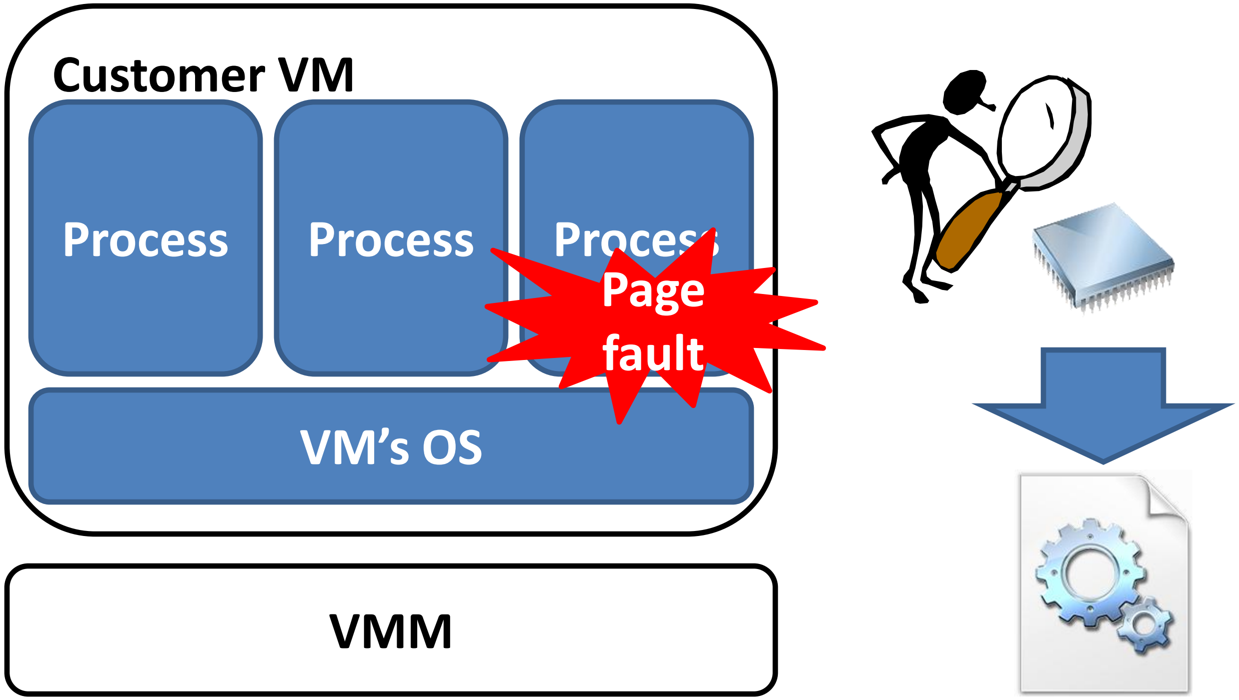
- Goal:
  - Which applications are run by a customer VM?
  - What's the version of these applications?

- Why?
  - Detect malicious code
  - Inform customer of vulnerable code
  - Deploy vulnerability-specific filters

Execution monitoring

- Goal: Identify all running binary code in a VM.
- Examples
  - Host agent: /proc, Process Explorer
  - Trap & inspect: examine OS data structures
  - Architectural monitoring: leverage MMU to identify all executing code

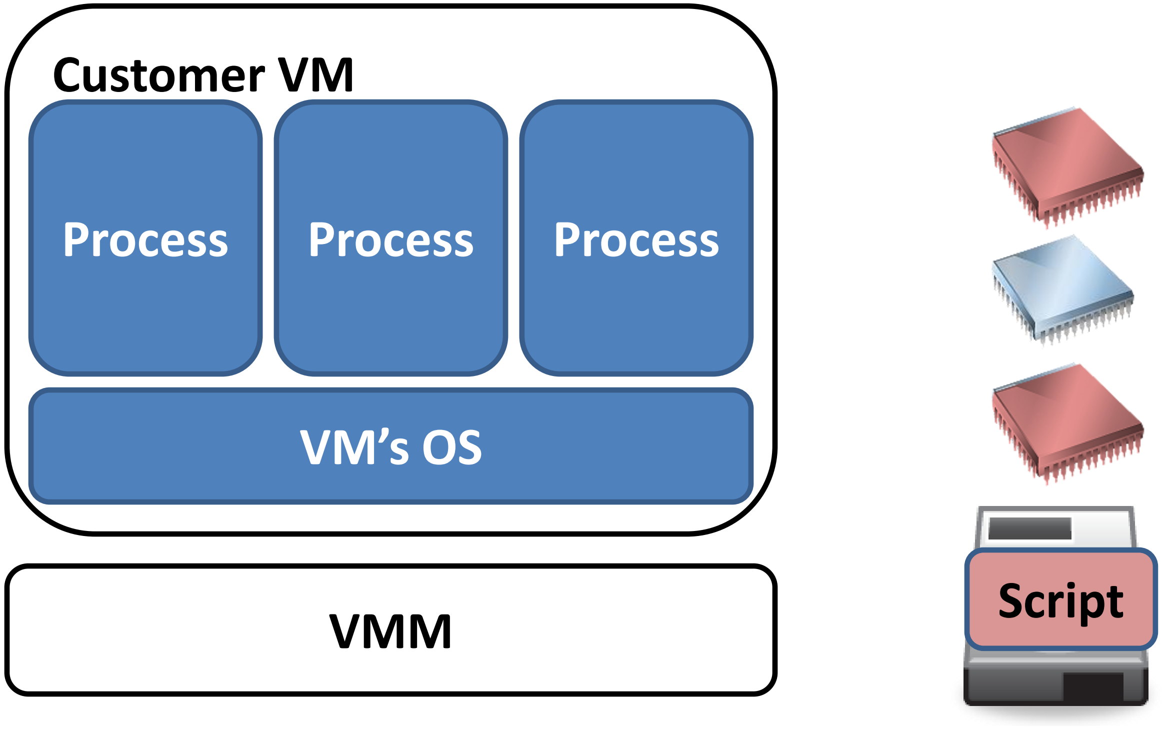
Execution monitoring



File monitoring

- Goal: What byte code is Java executing? What about the PHP interpreter?
- Examples:
  - Host-based: strace, filemon
  - Trap & inspect: examine OS data structures
  - Architectural monitoring: taint-tracking?

File Monitoring



Customer VM

Process

Process

Process

VM's OS

VMM

Script

Conclusion

- Architectural introspection should be used when possible.
- More research is needed to explore the range of events that can be monitored using Architectural introspection.
- Cloud providers should be mindful of the limitations of introspection.