Verifiable Cloud Outsourcing for Network Functions

(+ Verifiable Resource Accounting for Cloud Services)

Vyas Sekar

Carnegie Mellon

vNFO joint with
Seyed Fayazbakhsh, Mike Reiter
VRA joint with
Chen Chen, Petros Maniatis, Adrian Perrig, Amit Vasudevan

"Middleboxes" are valuable, but have many pain points!

Based on survey responses + discussions

Type of appliance	*Number*
Firewalls	166
NIDS	127
Media gateways	110
Load balancers	67
Proxies	66
VPN gateways	45
WAN Optimizers	44
Voice gateways	11

?

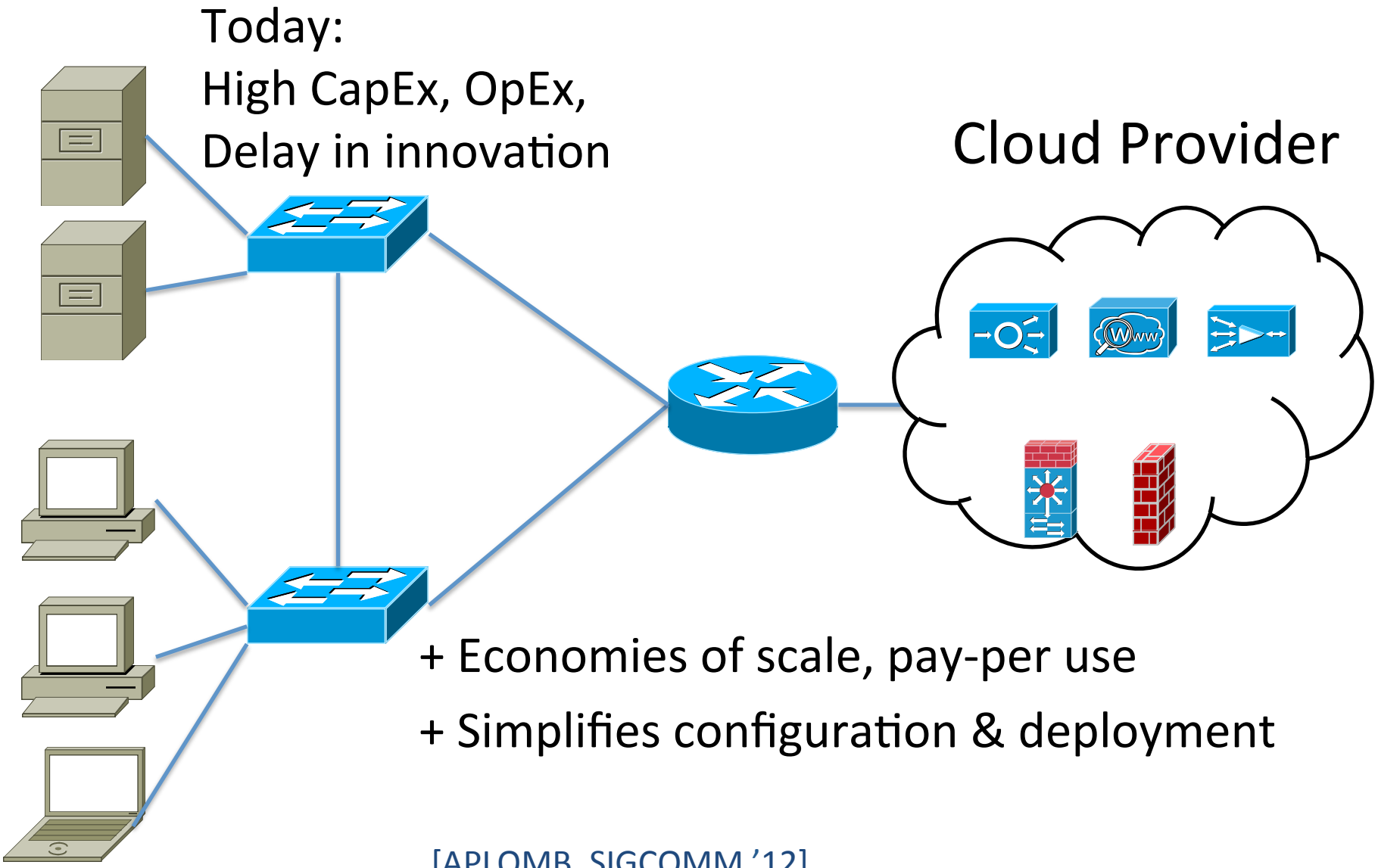
High Capital Expenses
Device Sprawl

High Operating Expenses
e.g., separate management teams need manual tuning

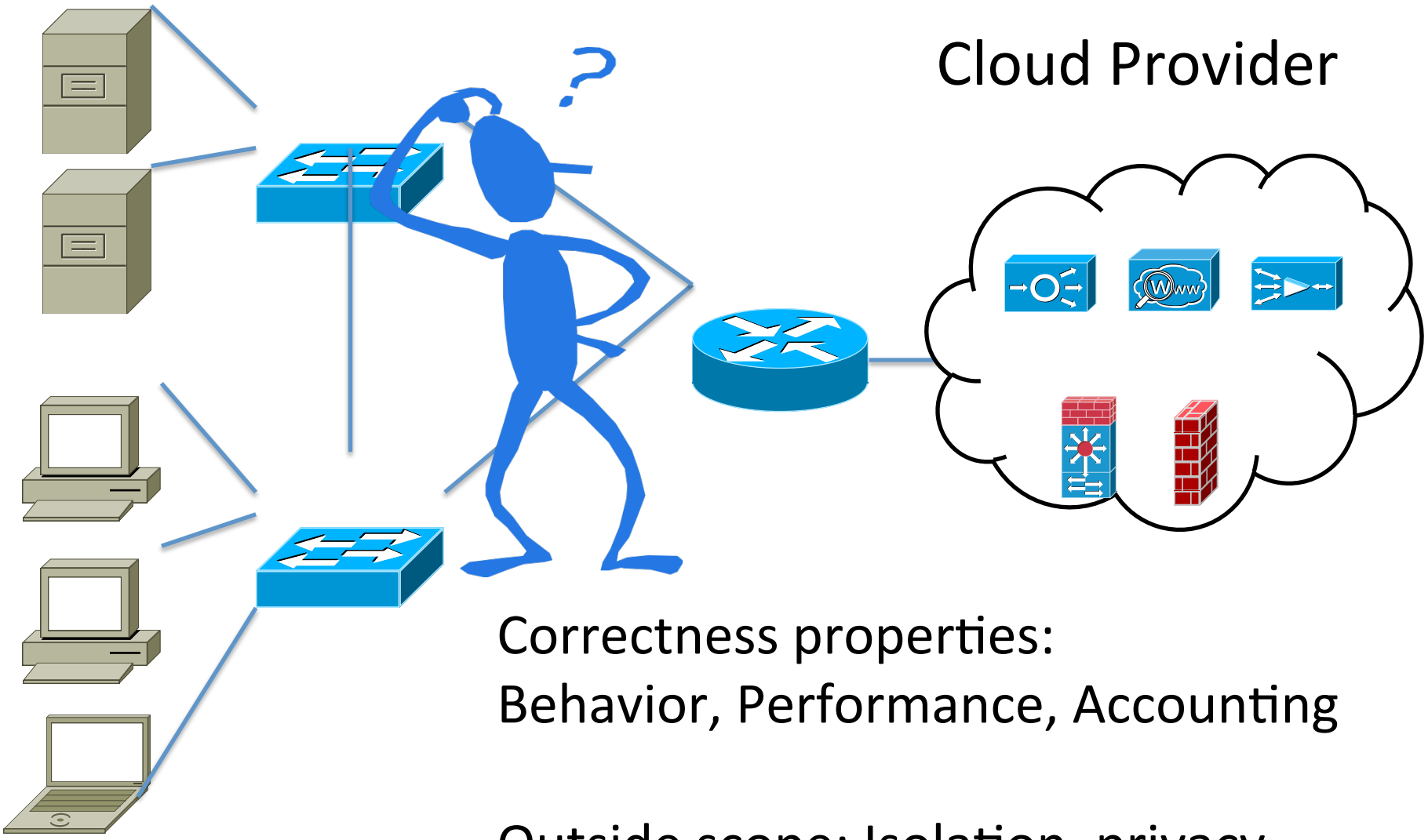
Inflexible, difficult to extend
→ need for new boxes!

[COMB, NSDI '12]

Case for Network Function Outsourcing (NFO)



Concerns with ceding control



Correctness properties:
Behavior, Performance, Accounting

Outside scope: Isolation, privacy, ..

[vNFO, HotMiddlebox '13]

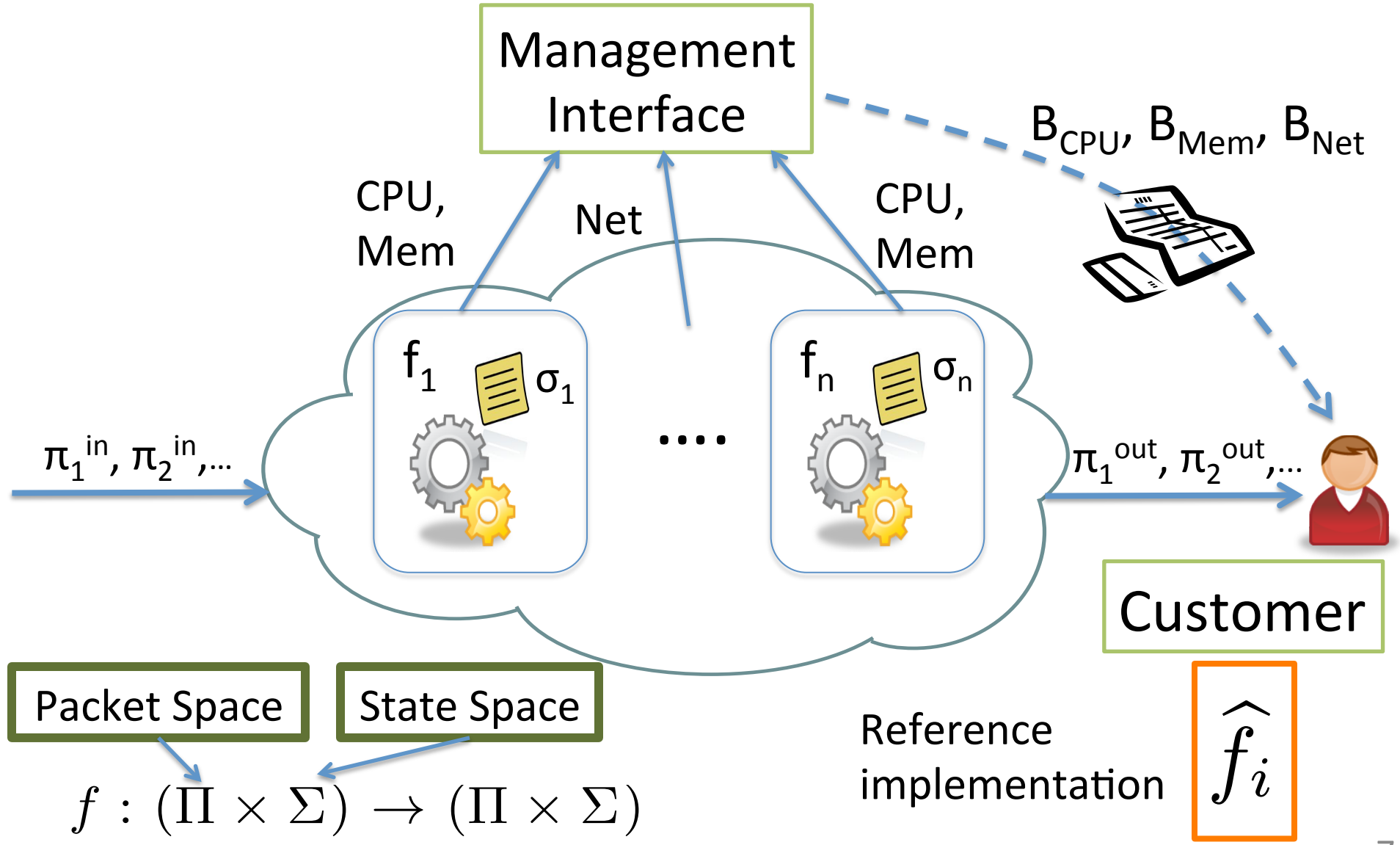
What makes this challenging?

- Lack of visibility into the workload
- Dynamic, traffic-dependent, and proprietary actions of the network functions
- Stochastic effects introduced by the network

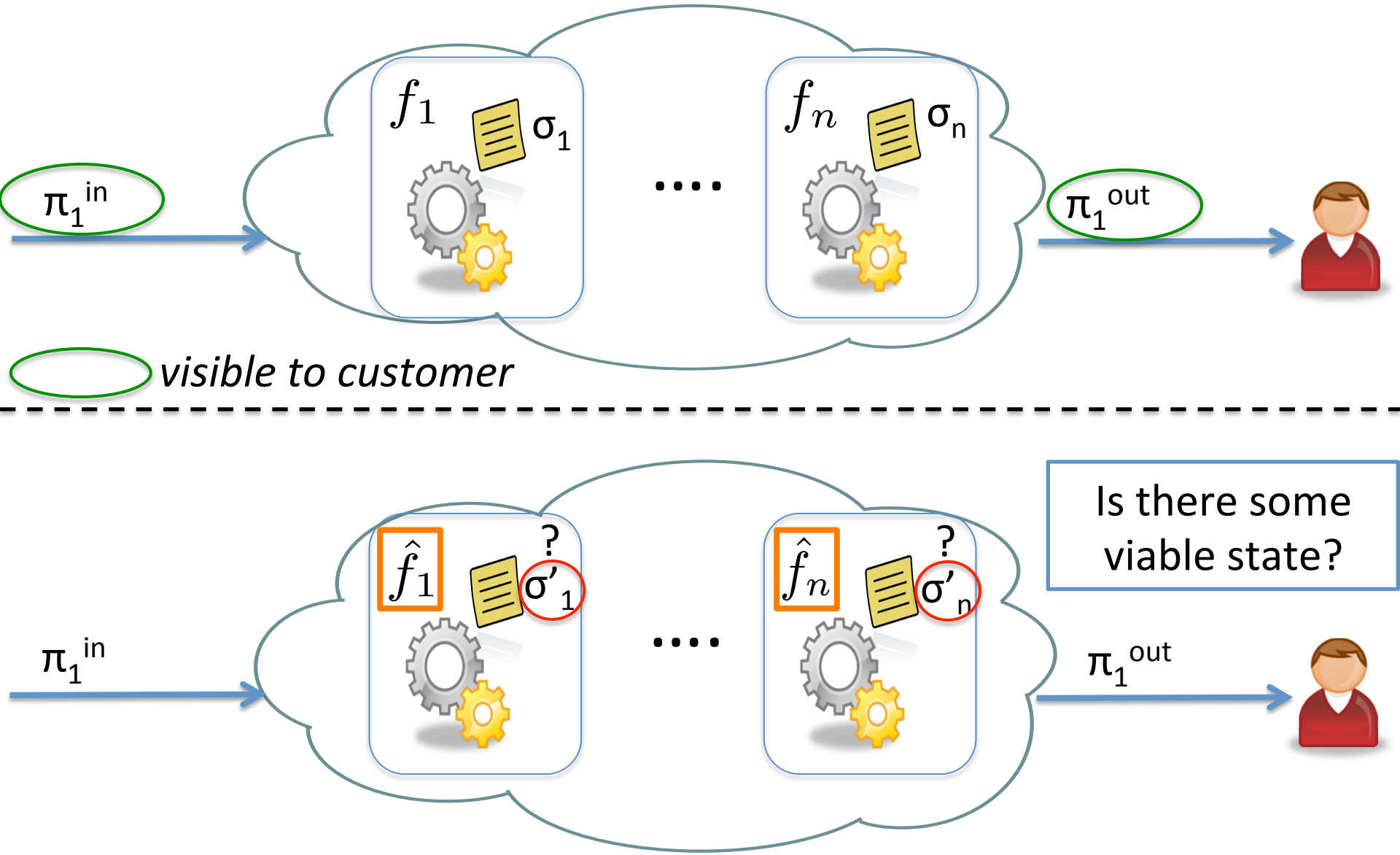
Outline

- Motivation for verifiable NFO
- Formalizing properties
- A roadmap for vNFO
- Discussion

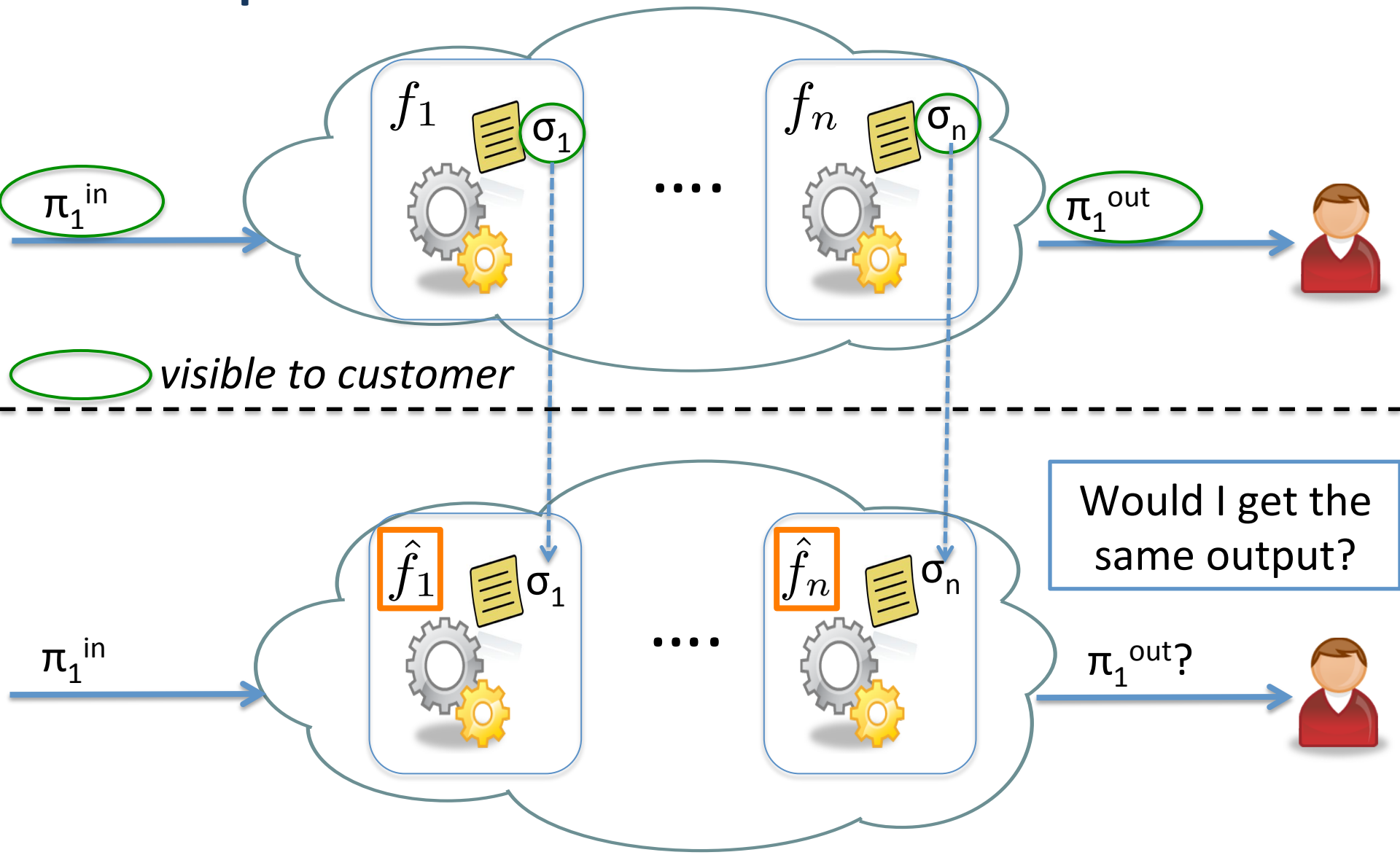
Formal Framework



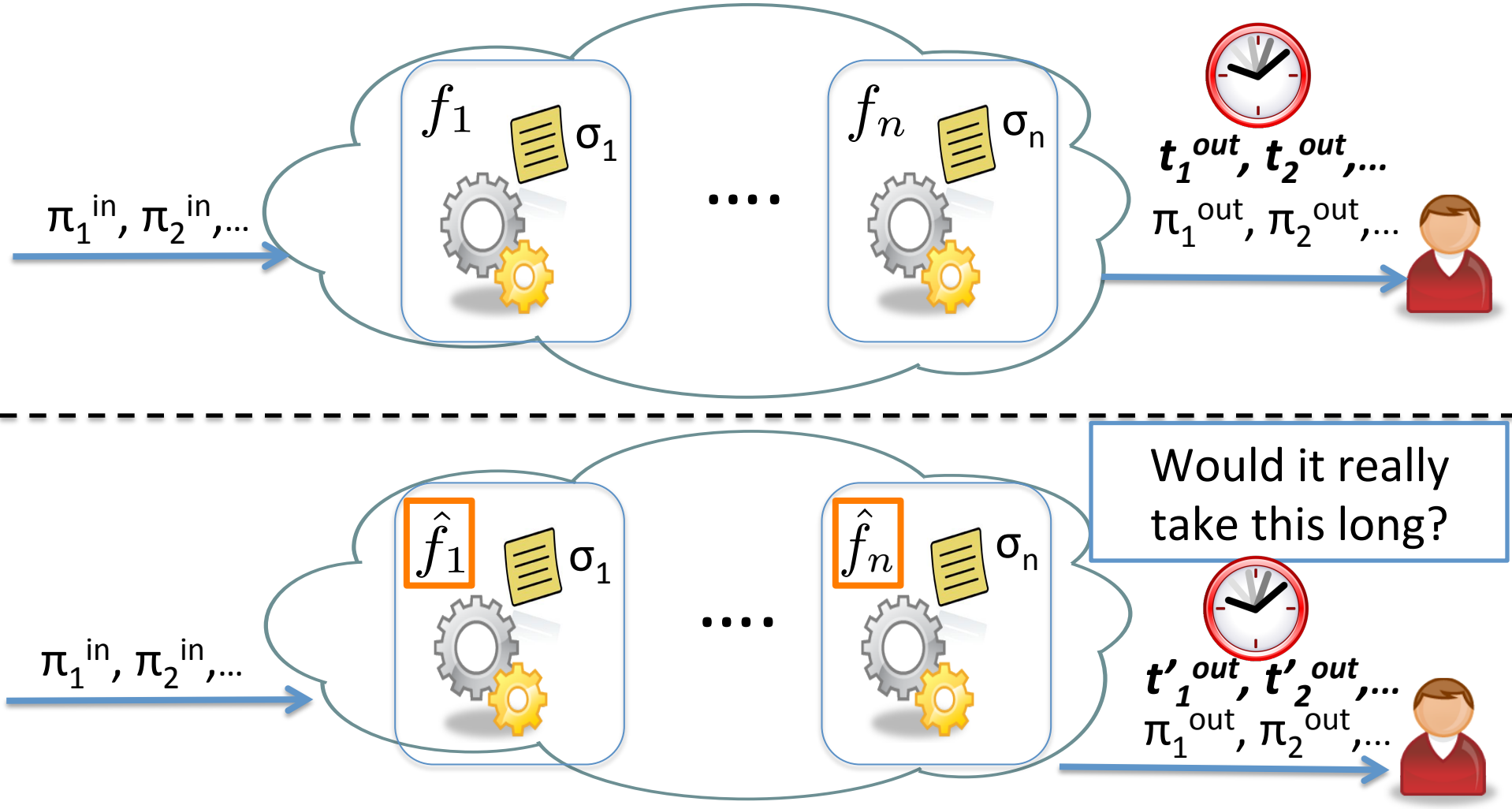
Blackbox Behavioral Correctness



Snapshot Behavioral Correctness

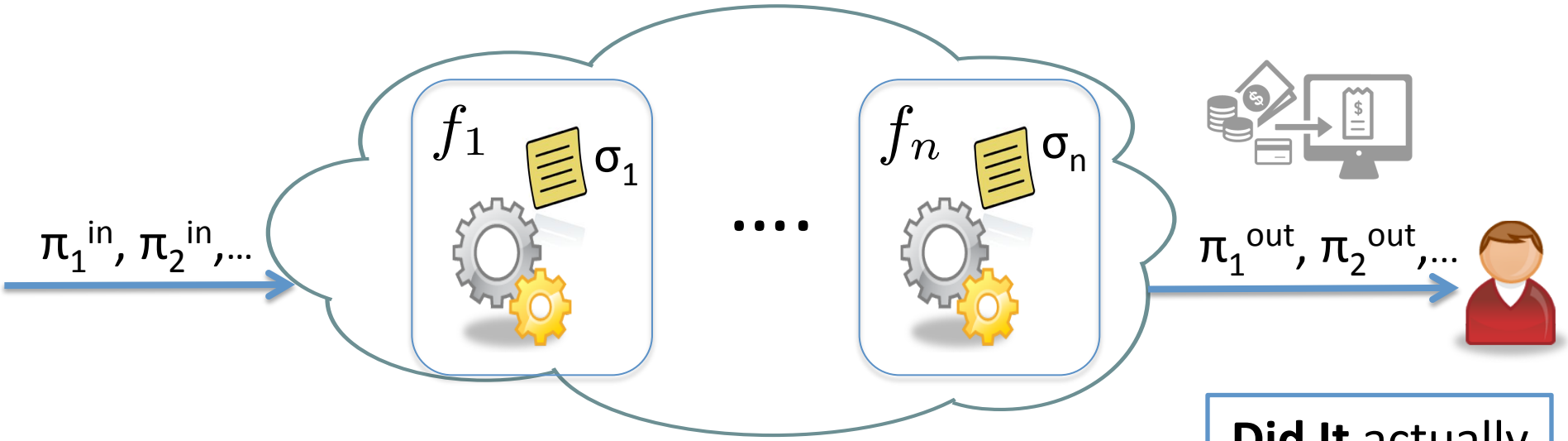


Performance Correctness



Observed provider performance ≈ Reference performance

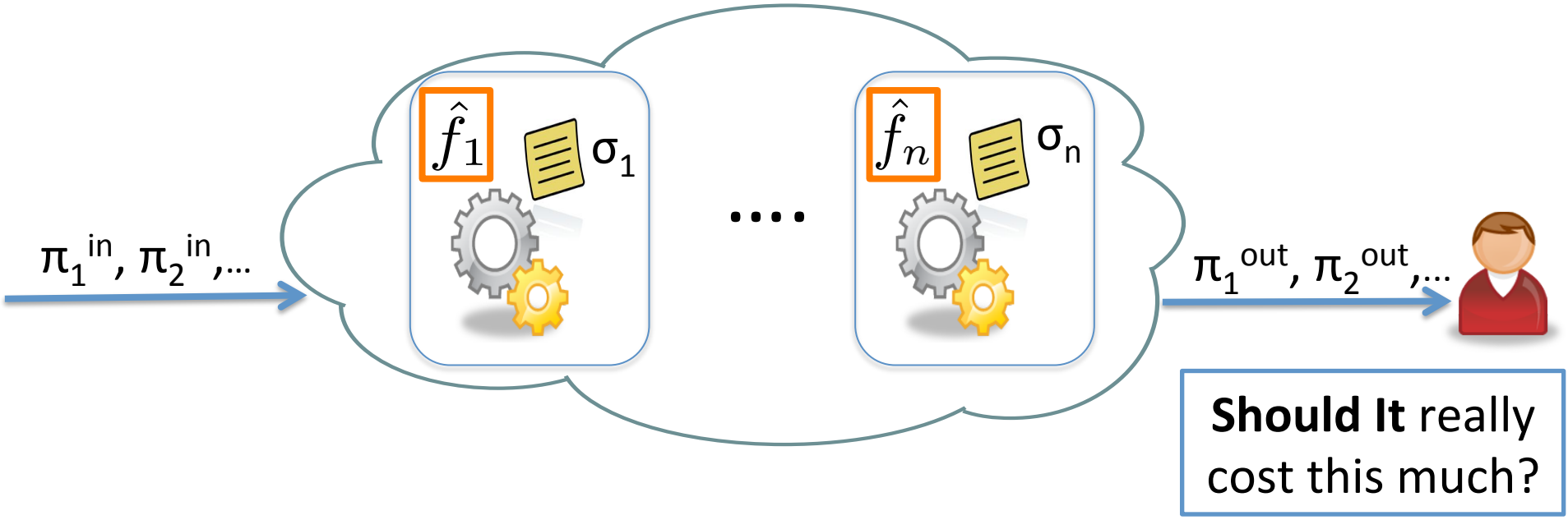
“Did-I” Accounting Correctness



Did It actually consume?

Charged value of resource *r* ≈
Consumption of resource *r* by provider

"Should-I" Accounting Correctness

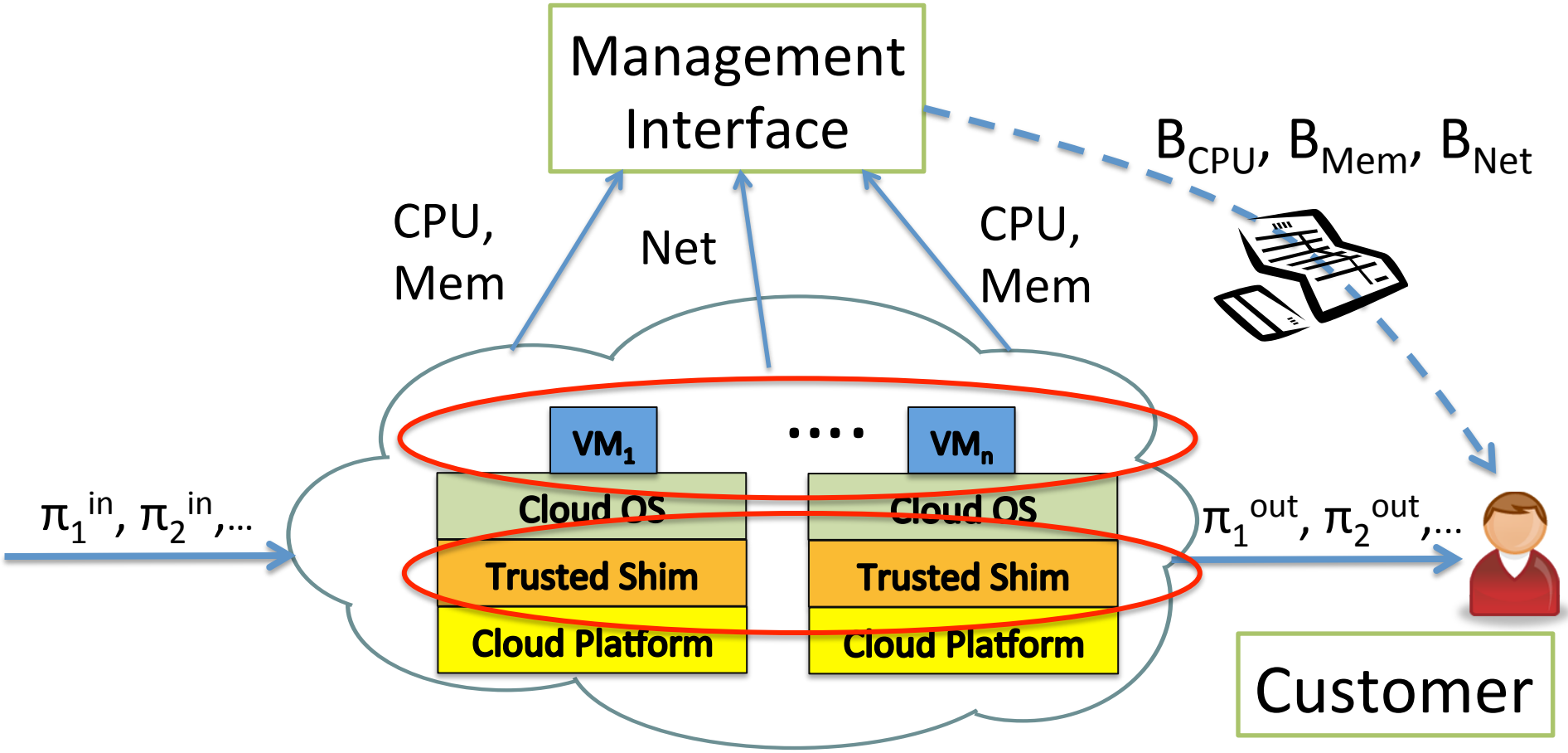


Consumption of resource *r* by provider ≈
Consumption of resource *r* by reference implementation

Outline

- Motivation for NFO + vNFO
- Formalizing vNFO properties
- A roadmap for vNFO
- Discussion

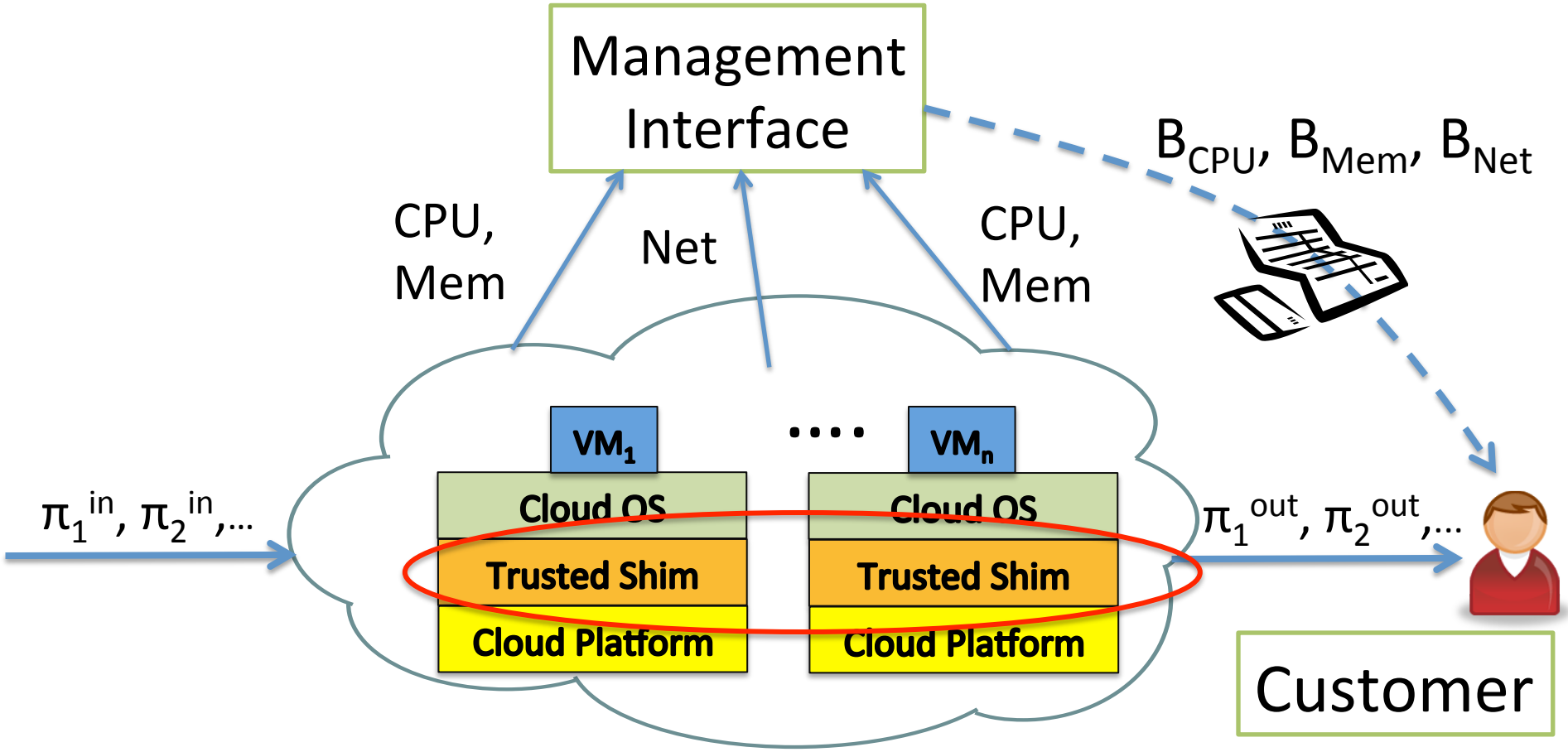
Verifiable NFO (vNFO) Overview



Each function is implemented as a *virtual appliance.*

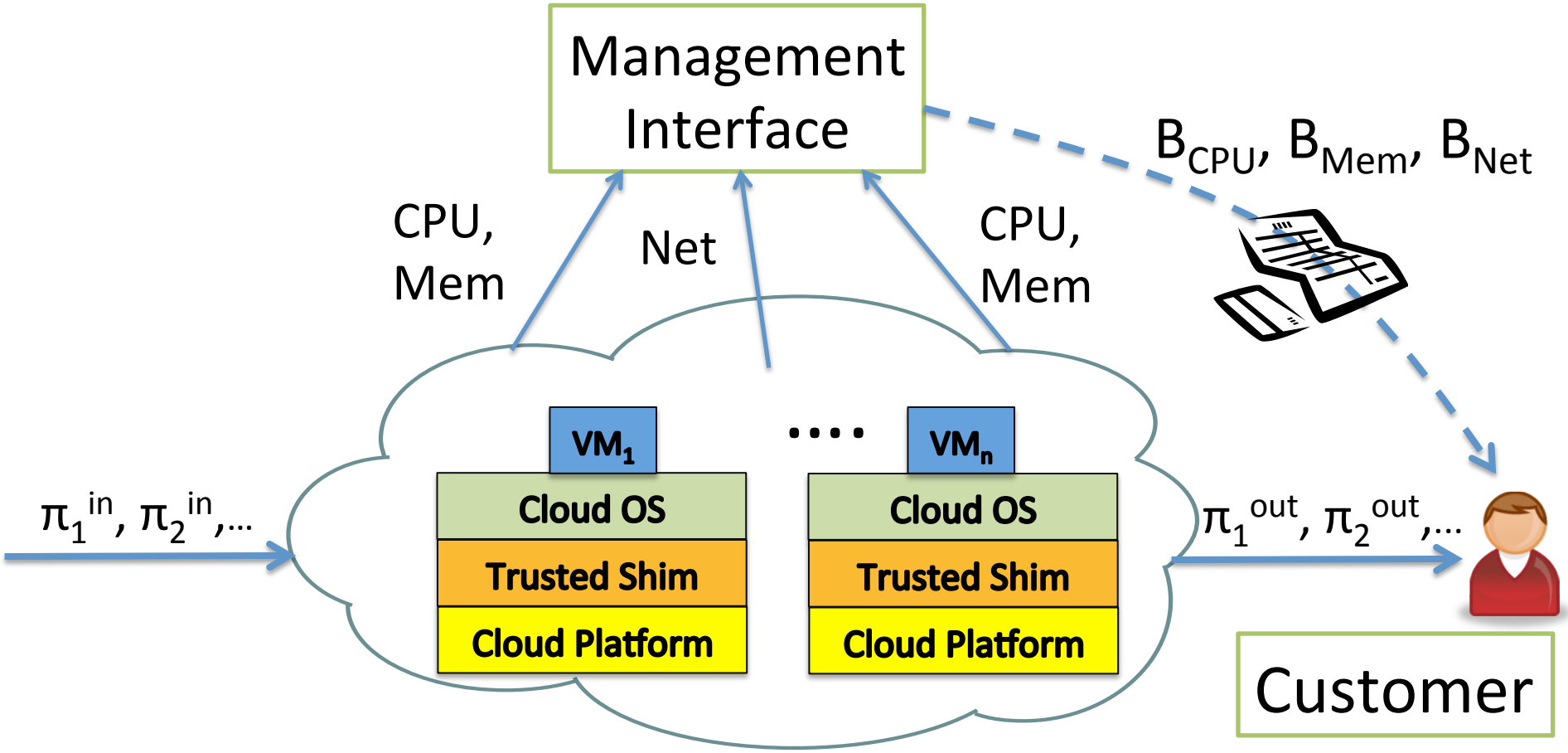
NFO provider deploys a trusted shim for logging.

Behavioral + Performance Correctness



Shim logs: every packet, VM state, timestamps per packet

Challenges!



1. Middlebox actions make it difficult to correlate logs
2. Scalability and performance impact due to logging

Potential solutions to challenges

1. Lack of visibility into middlebox actions:
   – Packets may be modified by middleboxes.

FlowTags: NSDI '14

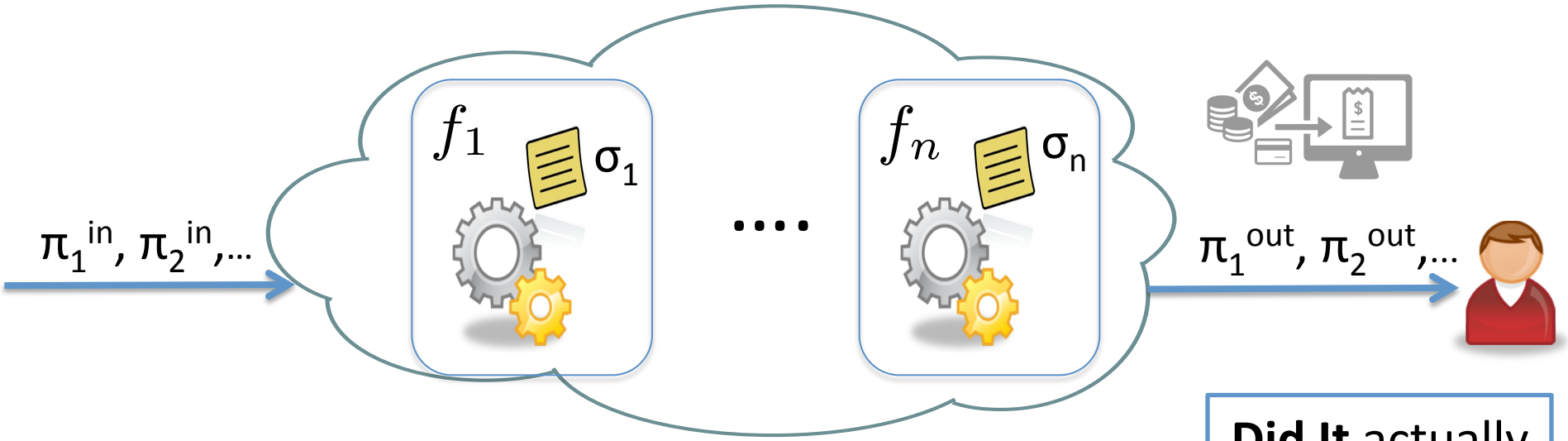
2. Scalability
   – Infeasible to log all packets and processing stats.

Trajectory Sampling

Outline

- Motivation for NFO + vNFO
- Formalizing vNFO properties
- A roadmap for vNFO
  - Verifiable accounting for Did-I correctness
- Discussion

“Did-I” Accounting Correctness



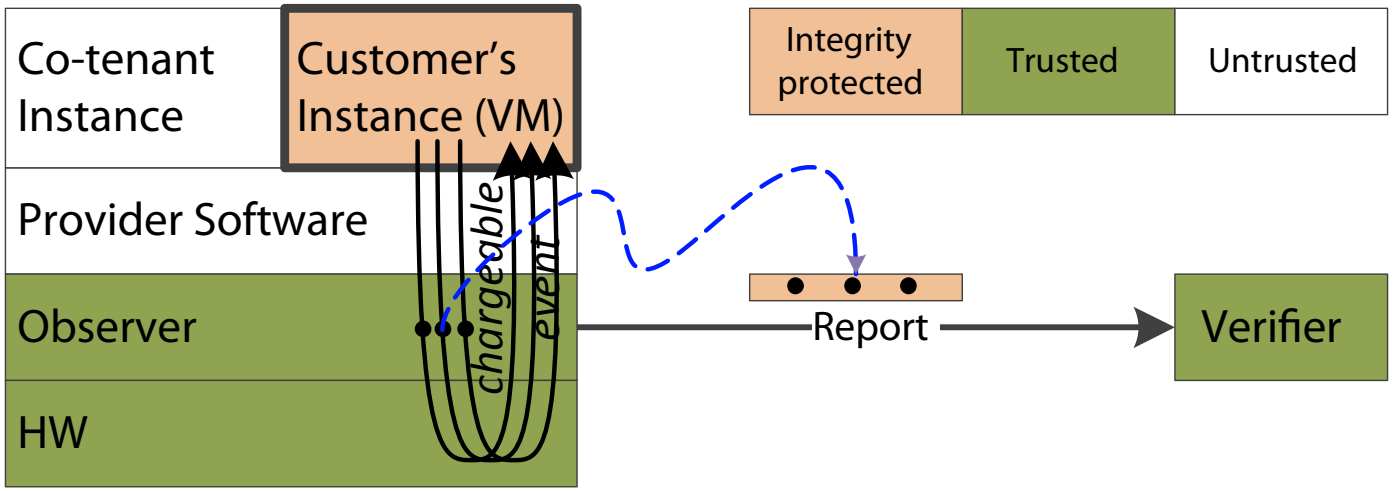
Did It actually consume?

Charged value of resource *r* ≈
Consumption of resource *r* by provider

Desired Properties

- Image Integrity
  - What is running
- Execution Integrity
  - How it is running
- Accounting Integrity
  - Only chargeable events are accounted

ALIBI Design Overview



- Image Integrity via Attested Instance Launch
- Execution Integrity via Guest-Platform Isolation
- Accounting Integrity via Bracketing

ALIBI architecture

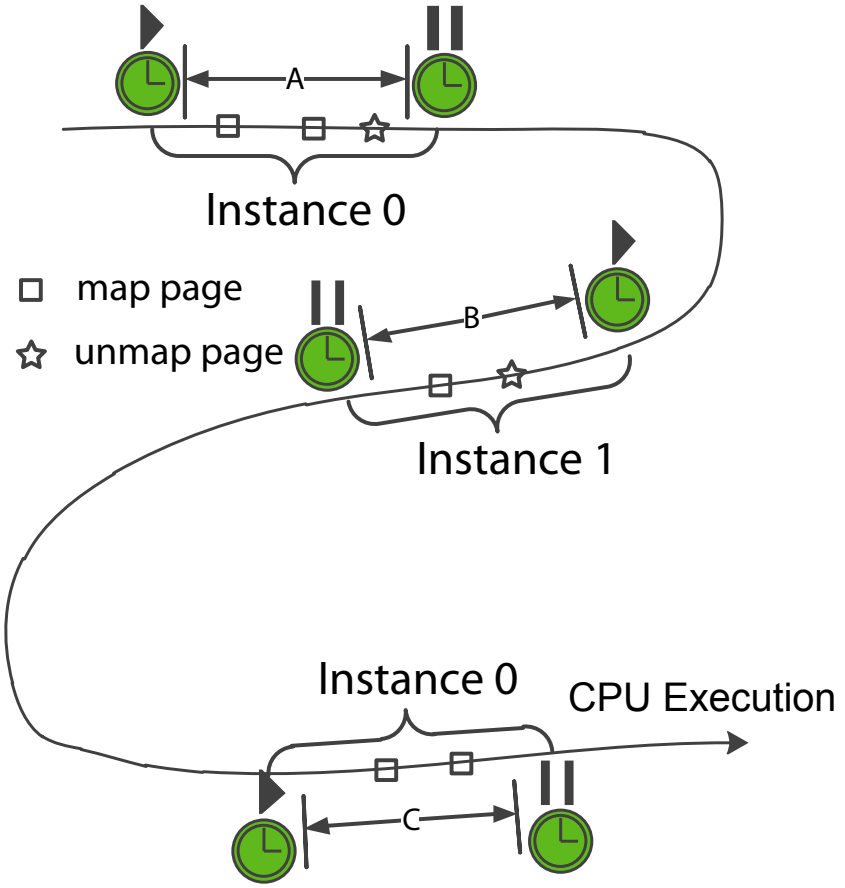
Enhance KVM nested virtualization with resource accounting and protection

- Advantage
  - Intercept critical events
  - No modification to L1 hypervisor
- Current Implementation
  - CPU accounting
  - Memory accounting

Guest-Platform Isolation (Execution Integrity)

- Memory Integrity
  - Isolate memory pages M by instances
  - M_i is writeable only when instance i is running
- Control Flow Integrity
  - Protect program stack by memory protection
  - Monitor and validate guest-CPU state changes
- Storage Integrity
  - Integrity protected file system

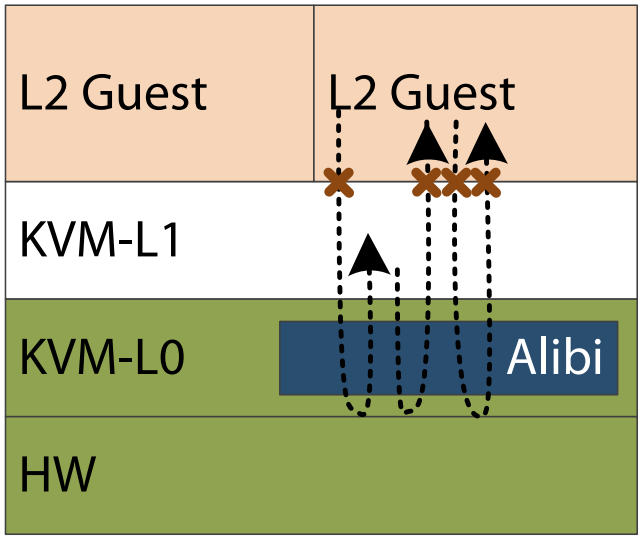
Bracketing (Accounting integrity)



- Event Detection
  - Control transfer
  - Memory mapping and unmapping
- Event Attribution
  - Associate resource usage with CPU ownership
- Event Reporting
  - Collect event measurements
  - Store and protect event measurements

CPU Accounting Case Study

- Account CPU cycles directly used by L2 guest
- Protect Time Stamp Counter (TSC) register



- Get CPU cycles, e.g., RDTSC
  - Entry into L2 guest
  - Exit from L2 guest
- Virtualize TSC register

Overhead of ALIBI

- HW: Intel Xeon E3-1220 (3.10Ghz) with 8GB RAM
- L2/L1: Ubuntu 9.04 (kernel version 2.6.18-10)
  L0: Ubuntu 12.04 (kernel version 3.5.0) and ALIBI

- Single-level virt. vs. native (no virt.) : ~9.5% slowdown
- Nested virt. vs. Single-level virt. : ~6.3% slowdown
- ALIBI additional: ~0.5% slowdown

Outline

- Motivation for verifiable NFO
- Formalizing properties
- A roadmap for vNFO
- Discussion

Discussion

- Is the NFO provider willing to deploy a shim?
- What are the market implications for customers?
- What is the role of SLAs?
- Should-I accounting? I/O accounting?
- Interesting anecdotes of correctness or accounting problems?
- Minimal TCB? without nested?
- Crowdsourcing correctness?
- …