Justifying Integrity using a VM Verifier

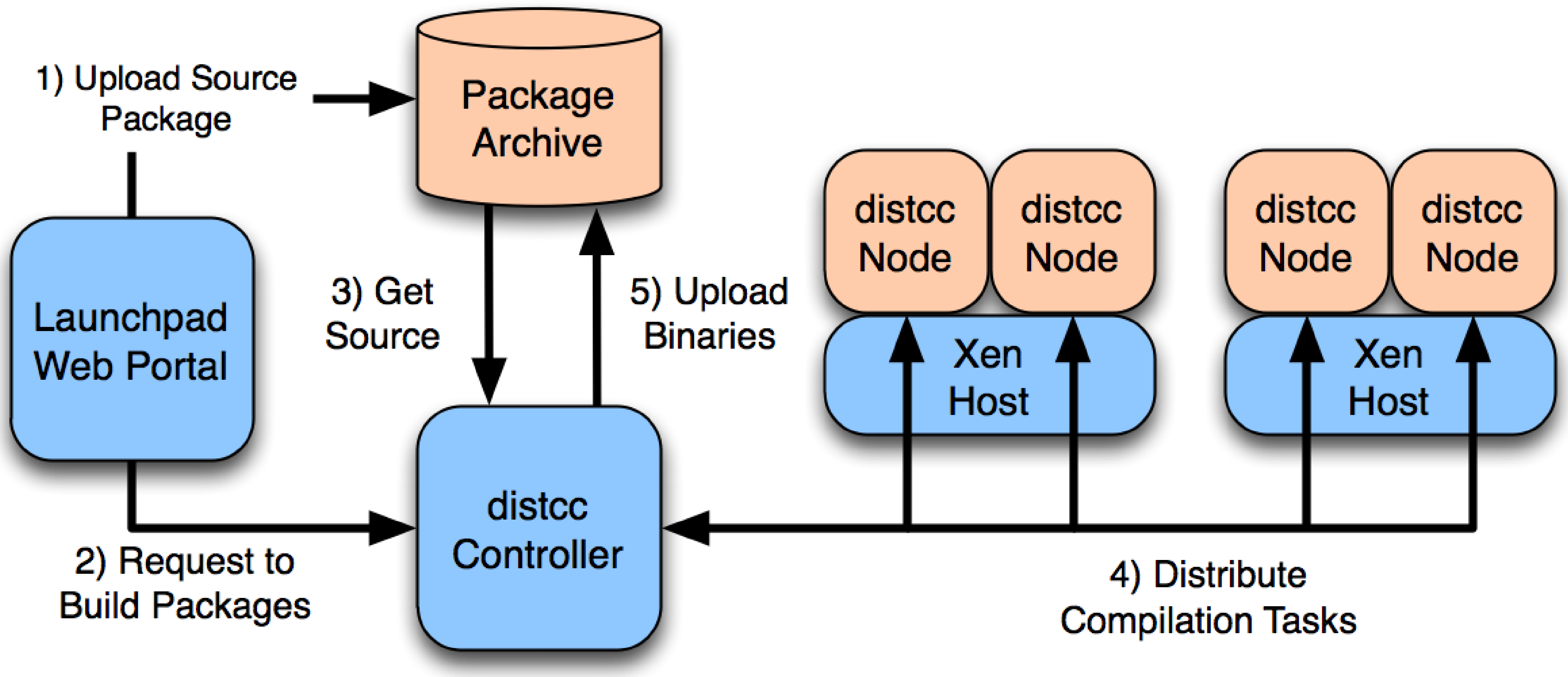
Joshua Schiffman, Thomas Moyer,
Trent Jaeger and Patrick McDaniel

- Cloud computing offers businesses and customers on-demand computing, storage, and virtual resources for their distributed applications.
- Cloud application integrity depends on the integrity of all components and inputs.
- However, the underlying infrastructure is opaque to consumers.

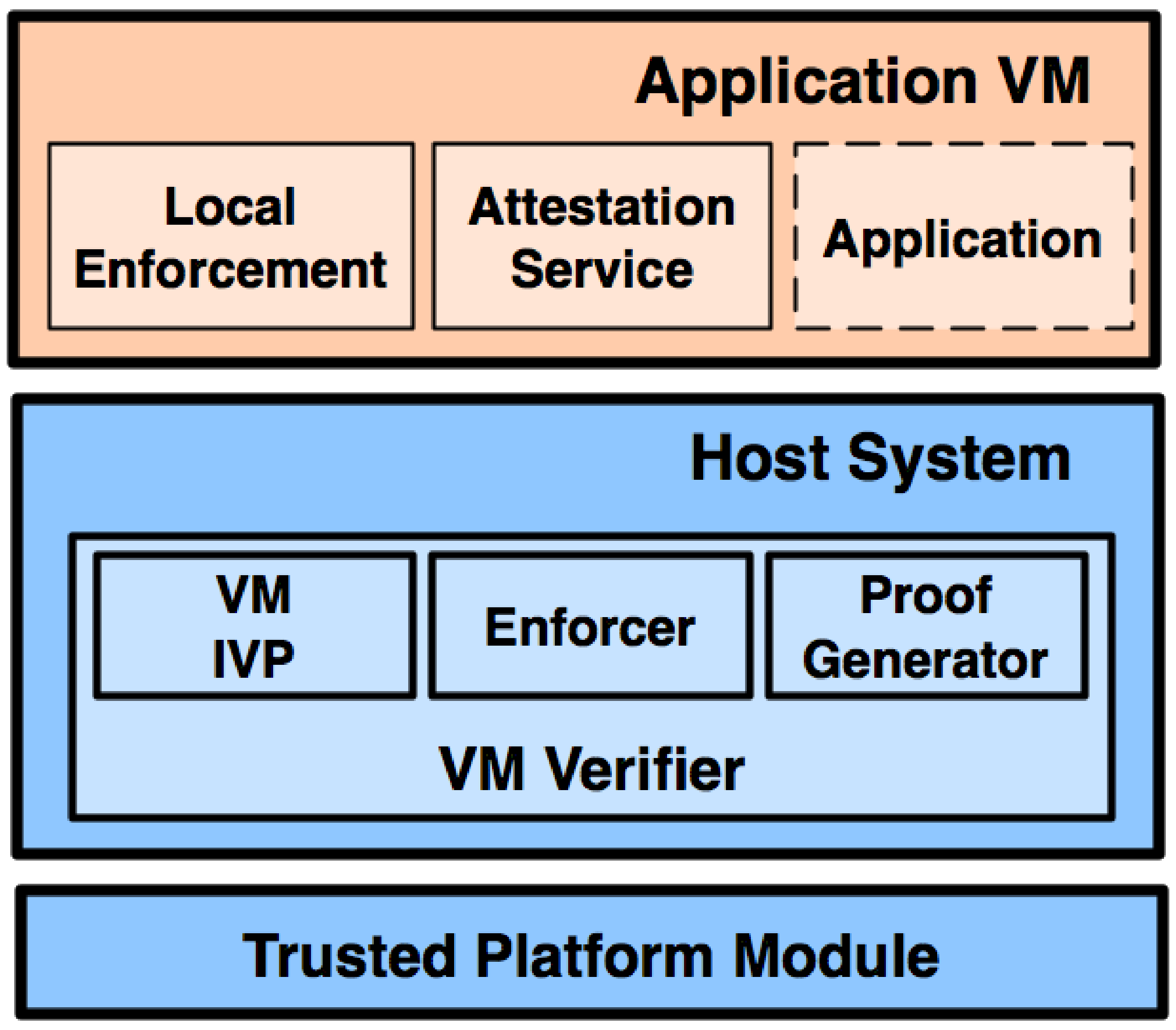
- Users and developers desire that cloud application satisfy specific risk guarantees to ensure:
  - VMM host integrity
  - VM data and code integrity
  - Untrusted inputs are discarded or upgraded
  - Data storage integrity

Example Scenario: Distributed Compilation

- Software distributions often involve compiling many source files for multiple target platforms
- Distributed compilation services like Canonical's Personal Package Archive compile source packages on a distributed compilation cluster
- Subscribers to a PPA depend on the service to produce safe packages
- Need to ensure only high integrity processes and inputs affect the computation



Solution: Virtual Machine Verifier



- VM Verifier (VMV) justifies that a VM meets a classical integrity model like Clark Wilson.

.Verifies the VM's initial integrity, installs integrity enforcement components in the VM, and provides a proof of the base's integrity.

- Input from remote systems are integrity verified against an integrity criteria
- Overhead introduced by the VMV on a proof of concept PPA was less than 4% increase in compilation time, with the majority due to IPsec

Integrity Criteria

- Integrity criteria define the specific requirements a cloud application must meet for protecting its integrity. Our proof of concept system enforces an approximation of CW integrity called CW-Lite

- The VMV uses this criteria to verify VM integrity. The VMV also generates proofs of the VM's integrity to remote parties

- Remote systems are verified using a VMV component called the *Port Authority*. If an input to application comes from a low integrity source, the PA must either discard or upgrade the input.

- A challenge is determining how to handle untrusted inputs to the system in general.

Publications

Joshua Schiffman, Thomas Moyer, Christopher Shal, Trent Jaeger, and Patrick McDaniel, **Justifying Integrity Using a Virtual Machine Verifier**. *25th Annual Computer Security Applications Conference (ACSAC)*, December 2009. (Acceptance rate 19%)

Sponsored by NSF (CNS-0627551) and IBM Research