In-Guest Mechanisms to Strengthen Guest Separation

XenSummit 2013

Philip Tricca
<philip.tricca@citrix.com>

Background

- Xen does security well
  - Strong isolation using hardware
  - Doesn't try to do too much
- Offload complexity to VMs / userspace
  - Drivers / QEMU / inter-VM communication (IVC)
  - Keeps Xen small
  - Separation now requires guest cooperation
- Value is added when VMs communicate / cooperate provided that …
  - Separation is preserved & mechanisms are strong
  - Sensible policy semantics are preserved
  - By "policy" I mean FLASK / XSM / SELinux

User-space mechanisms

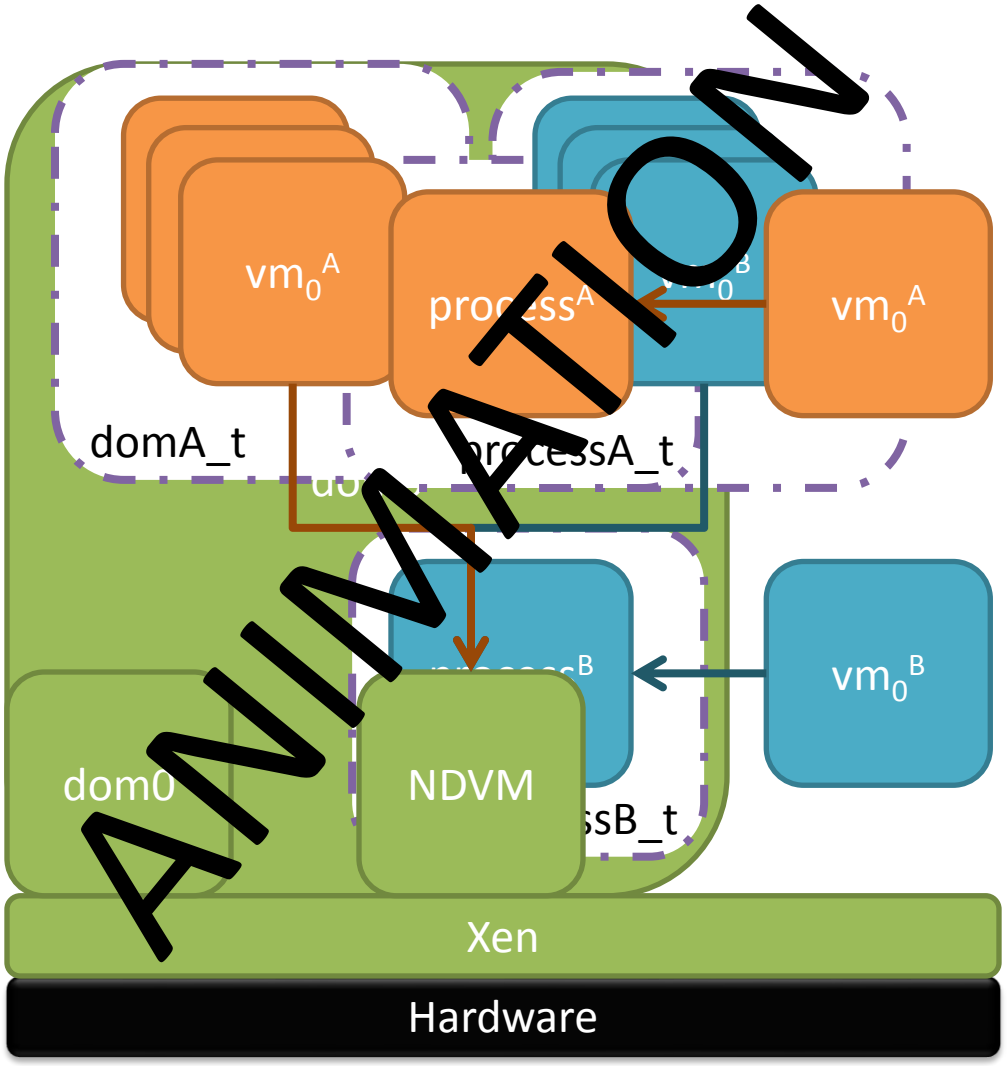
- Some work we've done
  - Strengthen separation between guest specific resources in dom0
  - sVirt implementation on XenClient XT
  - Separate QEMU instances
- Some work we're doing
  - Strengthen separation between guest VMs
  - Multi-tenant orchestration of mutually distrusting orgs
  - sVirt for management
- Apply these architectures to new work
  - Inter-VM communication (IVC)
  - Policy semantics in V4V in XenClient XT
  - IVC using front/back driver model

Caution

- Will move around a lot
  - Architectures that touch all aspects of virtualization
  - High-level concept to low-level implementation
  - Ask questions
- References provided
  - Time constraints
  - Please engage ‘off-line’

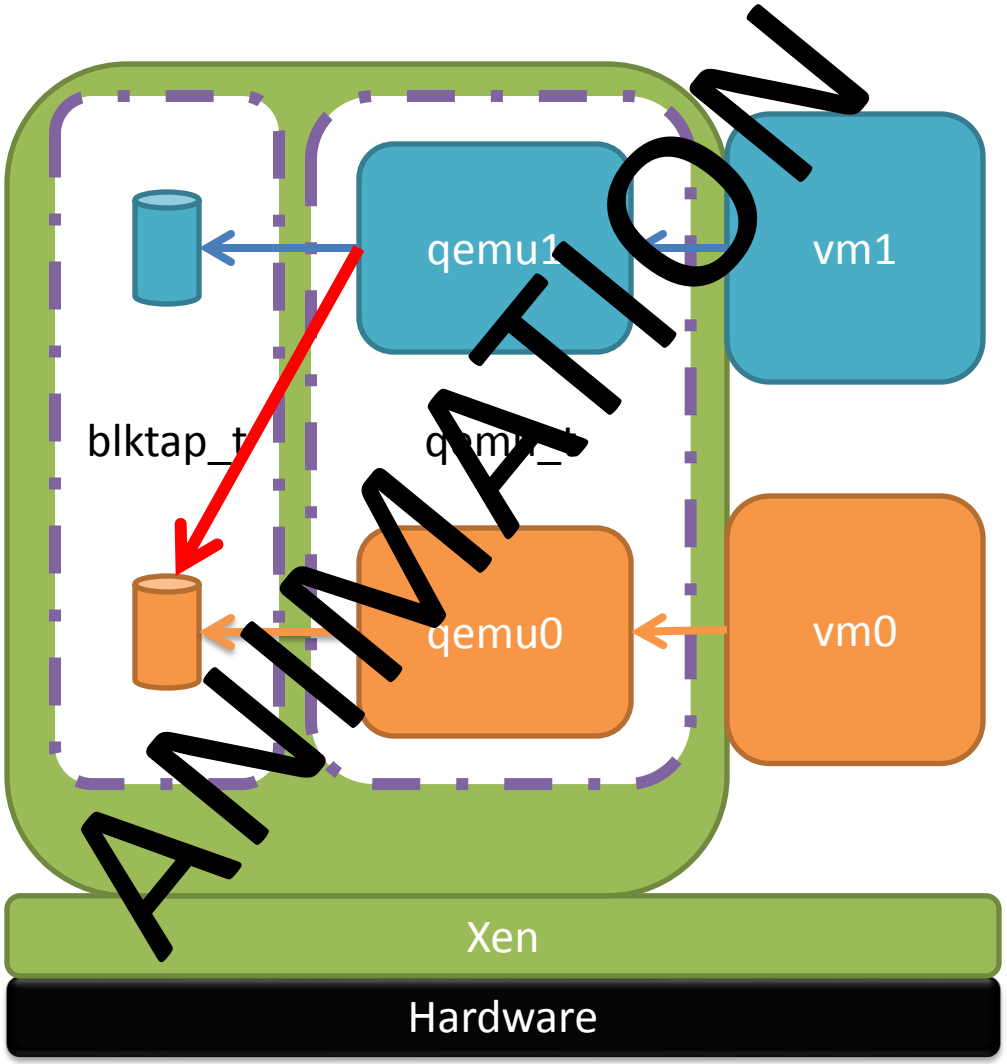
Diagrams & Conventions

- Colors
  - Green for ‘platform’ stuff (Xen + VMs)
  - NDVM = Network Driver VM
  - Blue & orange for ‘guest’ VMs
- Boxes
  - VMs are above ‘Xen’
  - Processes are inside VMs
  - Not drawn to scale, may shrink / grow
- Arrows denote binding or interaction
- Security boundaries: something_t is a FLASK type (SELinux / XSM)



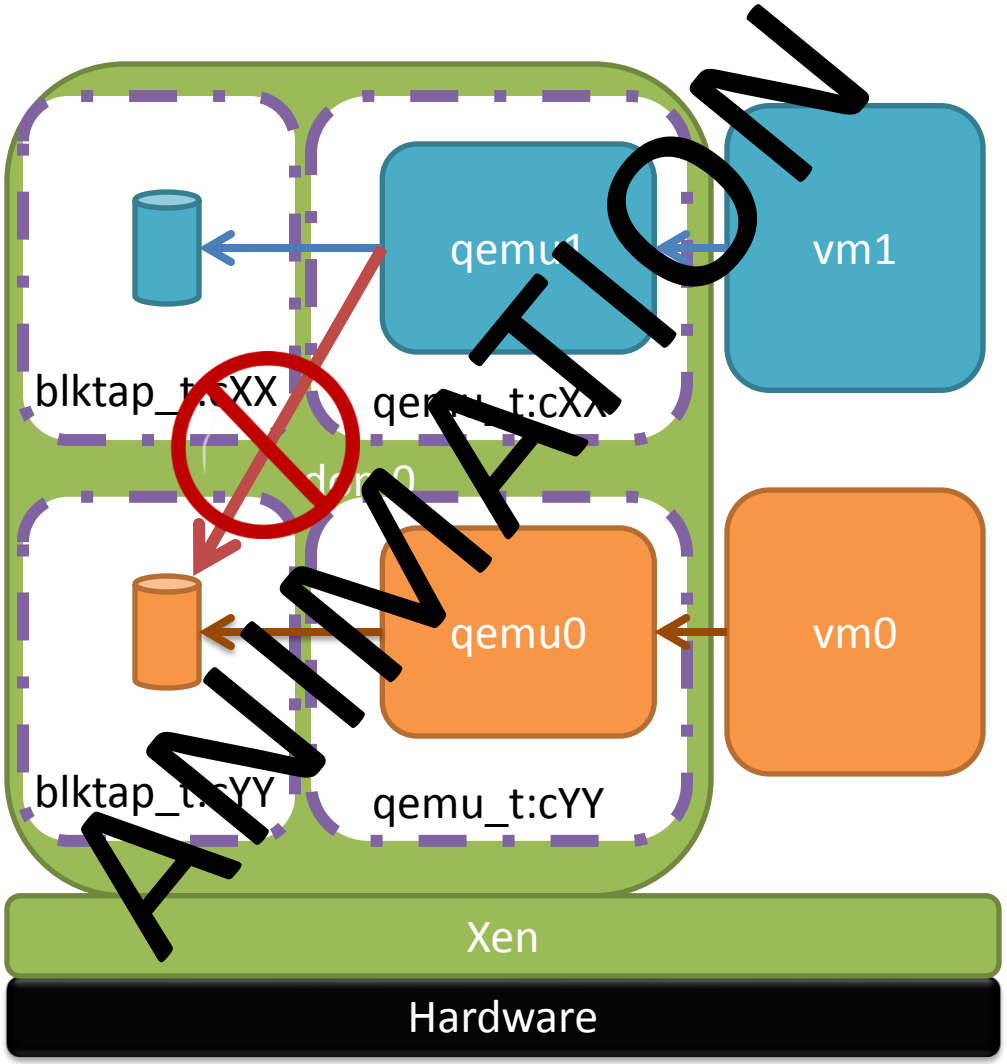
sVirt

- Separation between guests relies on mechanisms in dom0 / Linux
  - QEMU instances backing guests
  - Run with ‘root’ perms
  - Add SELinux but policy granularity is still wrong
  - qemu_t r/w blktap_t



sVirt

- Goals
  - Separate QEMU instances
  - Limit access to appropriate resources
- Implementation
  - Assign random MCS category to QEMU & relevant resources
  - Ensure category uniqueness



sVirt Details

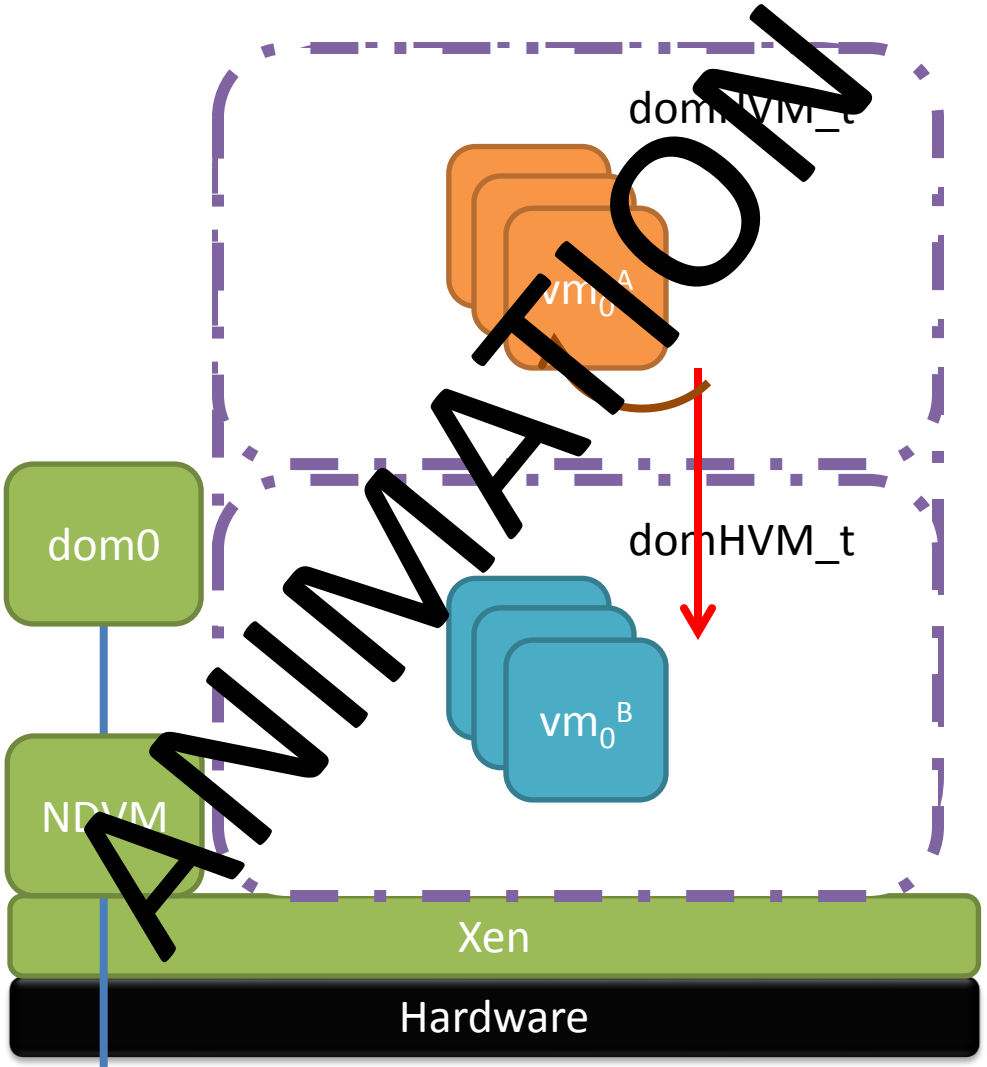
- Specification from SELinux community
  - Not a new thing: 2008
  - http://selinuxproject.org/page/Svirt_requirements_v1.0
  - libvirt security driver
- XenClient XT implementation
  - Minimally invasive
  - libvirt not an option
  - Binary interposed between toolstack and QEMU
- SELinux constraint / categories
  - Sets are cool
  - http://twobit.us/blog/2011/07/understanding-multi-level-security-part-3/
- OpenSource
  - Code: https://github.com/flihp/svirt-interpose
  - Analysis: http://twobit.us/blog/2012/02/svirt-in-xenclient/
  - Works on Xen OSS (requires some tweaks)
  - Interest from upstream?

Multi-Tennant Orchestration

- On-going work in XenClient XT
- Goals
  - Provide separate orgs management mechanisms on a single platform
  - Maintain mutual distrust between orgs
- Increasingly relevant
  - Multiple virtualization mgmt stacks
  - Security concerns in larger ‘cloud’ context
  - BYOD / multi-personality devices
- Who owns the device?
- Which interests does the device represent?

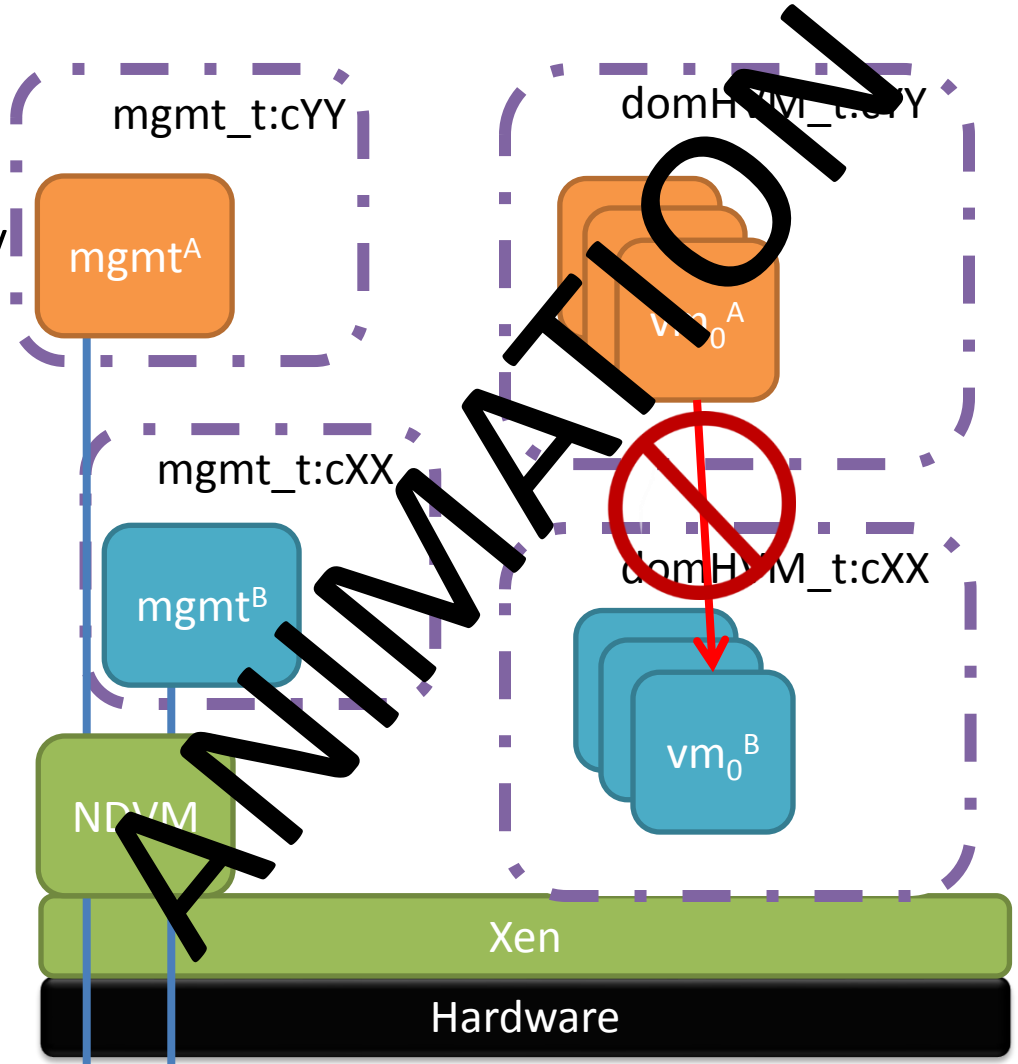
Multi-tenant Orchestration

- Similar issue with XSM & guest VMs
  - Guest VMs use a single XSM type
  - See domHVM_t ‘self’ rules
  - Implications when we consider groups of VMs belonging to different orgs



Multi-tenant Orchestration

- VM dedicated to orchestration
  - Represents the interests / actions of a single org
  - Bootstrapping non-trivial
  - Unique category to represent each mgmt realm
  - Prevent cross realm actions
- sVirt architecture applied to similar problem
  - Work on-going



Inter-VM Communication

- Short discussion on xen-devel back in June
  - http://lists.xen.org/archives/html/xen-devel/2013-06/msg01123.html
  - Alternative to V4V from XenClient is imminent
- Proposed new approach
  - Negotiate shared pages through rendezvous service
  - Possibly a 3rd party daemon
  - Possibly through xenstore
- Doing this with existing mechanisms is a ‘very good thing’

IVC policy model

- policy semantics of V4V were desirable
  - First-class xen object == clear XSM policy
  - Send / receive semantics
  - Communication channels between VMs were clear
- Policy for new IVC mechanism
  - Coms channel reflected in XSM for grant mechanism
  - Interesting possibility to extend XSM to vsock connections to create higher-level semantics
- Feasibility?
  - Caution against introducing new / competing policy mechanisms
  - Seems connection mgmt will land in XenStore anyways
  - XenStore perms sufficient?
  - User-space object manager?
  - BOF?

Policy Semantics

- Flexible, loosely-coupled, disaggregated systems
  - Good engineering practices enable good security
  - Make changes easier
  - Make dependencies obvious
  - Clear interfaces
- Hazards w/r to separation / policy goals can be subtle
- Consider attack scenarios w/r to
  - Enforcement mechanisms
  - Desired policy semantics
- What will policy look like after addition of new mechanism?