



cloudTM
SOFTWARE GROUP

Storage Management API v3 (SMAPIv3)

Mark Syms

Storage Engineering Lead, XenServer

Cloud Software Group

First - Why?

- Previous XenServer Storage Management API (SMAPIv1) is tightly integrated with Xapi
- Layering violations with API calls not being given all the context they require
- Reliance on Xapi for locking and reference counting
- Untestable/Unusable without running Xapi

Enter SMAPIv3

- Simplified API for interacting with Storage for Virtual Machines
- Key principles
 - No reliance on external entities (such as Xapi)
 - Responsible for its own locking and reference counting
 - Separation into conceptual entities, Storage Repositories (SR), Volumes, Datapath
- Independently testable in isolation

What is it not?

- Strictly just an API
- No concrete implementations aside from simplistic examples

Can it be used in non-xapi environments?

- Yes, it can
 - requires a small number of Python scripts
- But, the code for generating the Python API bindings lives in the Xen-API github repository.
- Documentation still describes this as being a means for connecting Xapi to storage
- Currently not updated to Python 3, but this is planned

What is required to implement?

- Dependent on what the intended storage can provide
- Offload as much as possible to the storage, e.g use of reflinks or built-in snapshots
- Only implement what's needed

Example

<https://github.com/xapi-project/xen-api/tree/master/ocaml/xapi-storage/python/examples>

- Simple, unstructured, file based volumes (aka raw)
- losetup + blkback for IO datapath to Xen guest
- Assumes a pre-mounted filesystem location
 - Adding a probe and mount (optionally mkfs) operation
- No snapshot or clone support
 - Use a filesystem with reflinks and use “cp –reflink=always” to implement

Future

- Move API definitions and generator to OPAM?
- Possible future implementations
 - Ceph?
 - GlusterFS
 - dmthin/btrfs for host local storage
 - ?



cloudTM
SOFTWARE GROUP