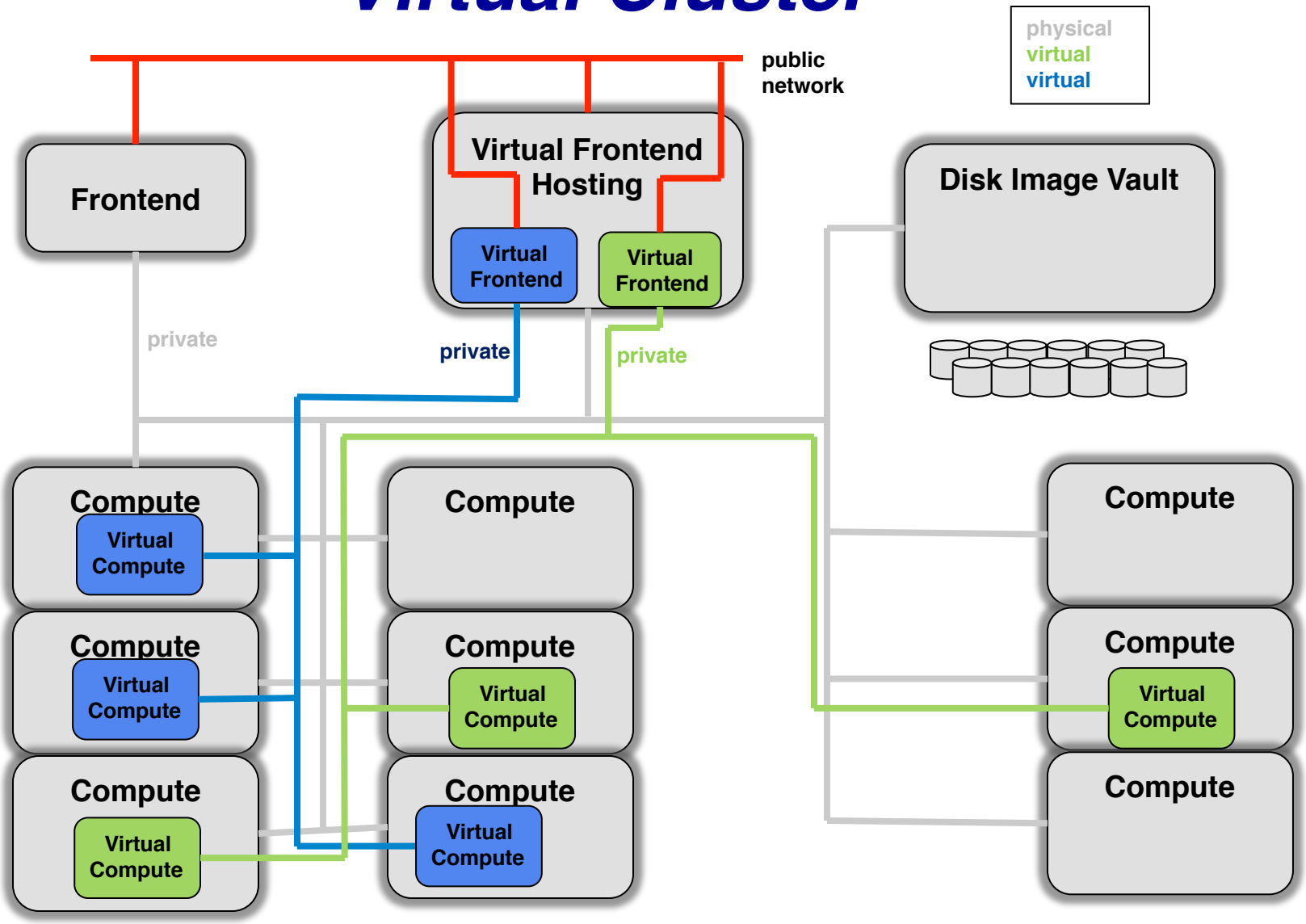
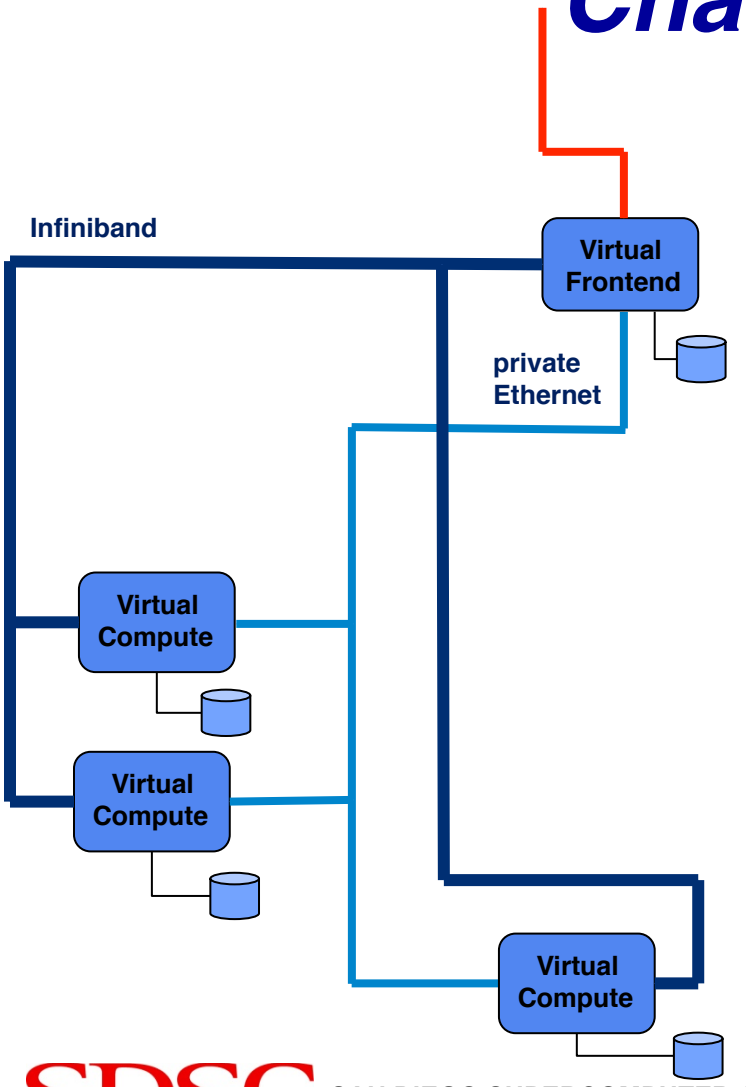
# Virtual Cluster

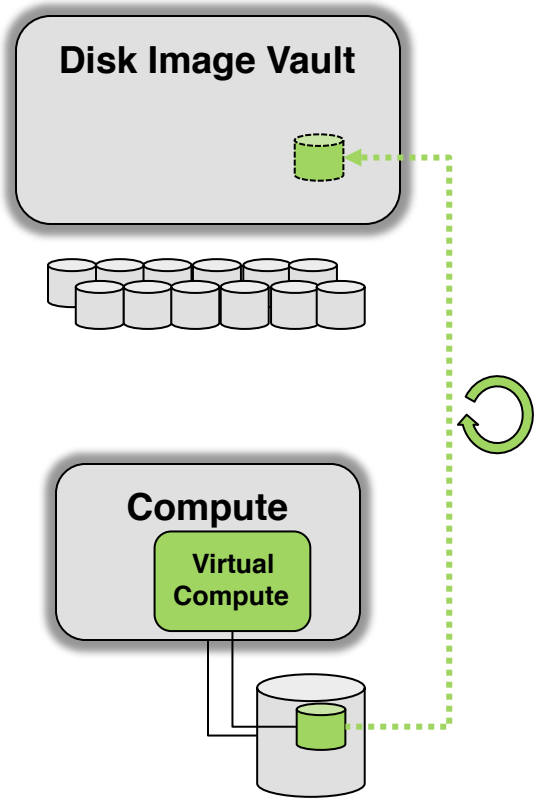


# High Performance Virtual Cluster Characteristics




- All nodes have
  - Private Ethernet
  - Infiniband
  - Local Disk Storage
- Virtual Compute Nodes can Network boot (PXE) from its virtual frontend
- All Disks retain state
  - *keep user configuration between boots*

# Steady State for a Virtual Compute Node Disk

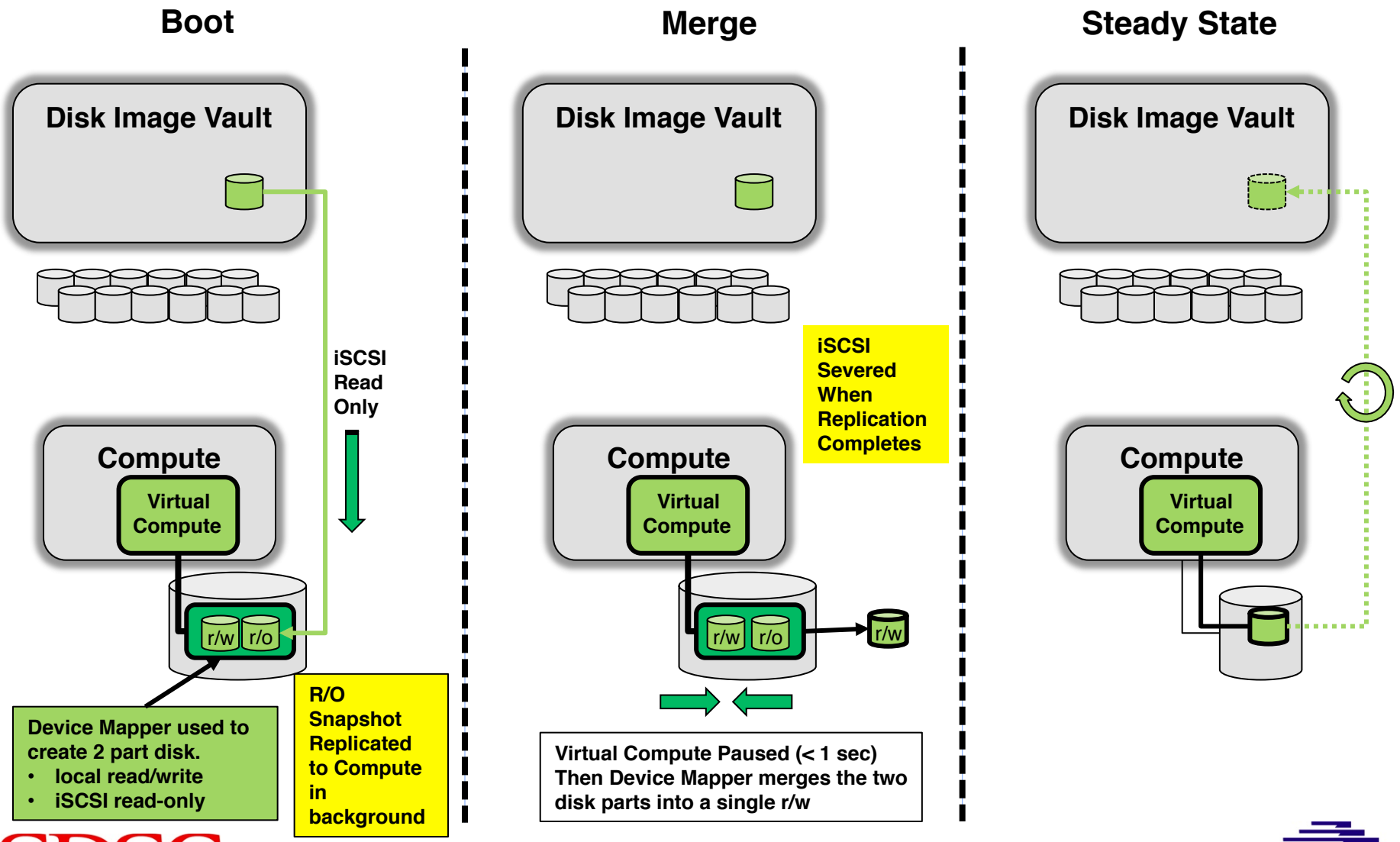



- Virtual Node Disk Image is on local physical disk of Physical Compute node
- Virtual Disk Image is periodically Synched to Disk Image Vault

- At Virtual Node Shutdown: Virtual disk image is synched to disk vault and then removed from Physical compute node

**How do we get to steady state?**

# Getting to Steady State



# *Advantages*

- **Virtual Machines boot very quickly**
  - Disk Image on Storage Vault is replicated in background
- **On Shutdown, final disk state is returned to vault**
- **All Disks are lazily replicated**
  - If physical compute node fails, state of virtual disk is close to “up to date”
- **Utilizes the parallel I/O busses of all the compute nodes**
  - Disk vault can be built using commodity components.

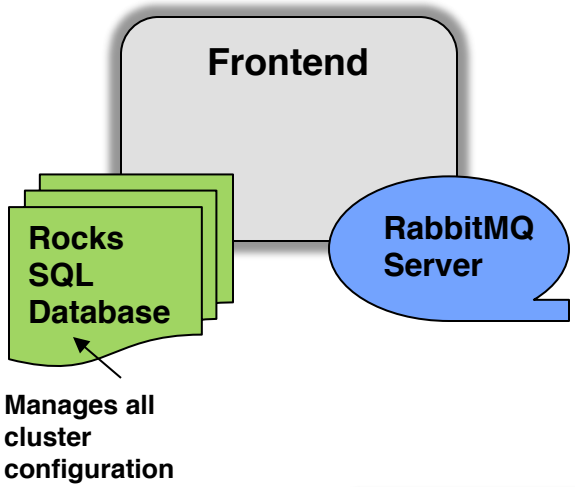
# *Disadvantages*

- **More failure modes to manage, for example**
  - Disk vault fails during the boot cycle of a virtual compute node
  - Physical compute node loses local disk
  - iSCSI issues
  - Data replication failures (forward or reverse)
- **Need to determine how NOT to overload disk vault when a large virtual cluster is booting**

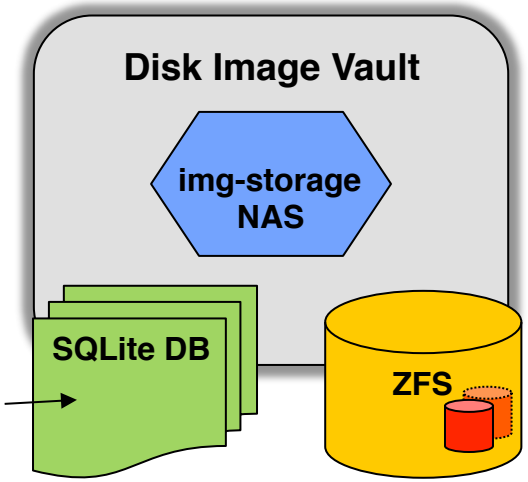
# *Implementation*

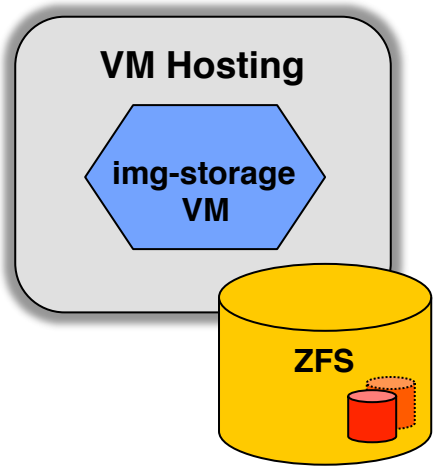
- **Software Systems**
  - ZFS file system
    - Highly Reliable
    - Snapshot and Snapshot replication. Supports Incremental Replication
    - Virtual Volume (disk) Support
    - No specific hardware requirements
  - RabbitMQ – Active Message Queue Management
  - SERF – Cluster membership via gossip Protocol
  - Rocks Cluster Toolkit
  - SQLite – Manage/record state of disks
- **Currently Alpha Quality**

# Key Components




**Commands to img-storage Daemons are queued through RabbitMQ**



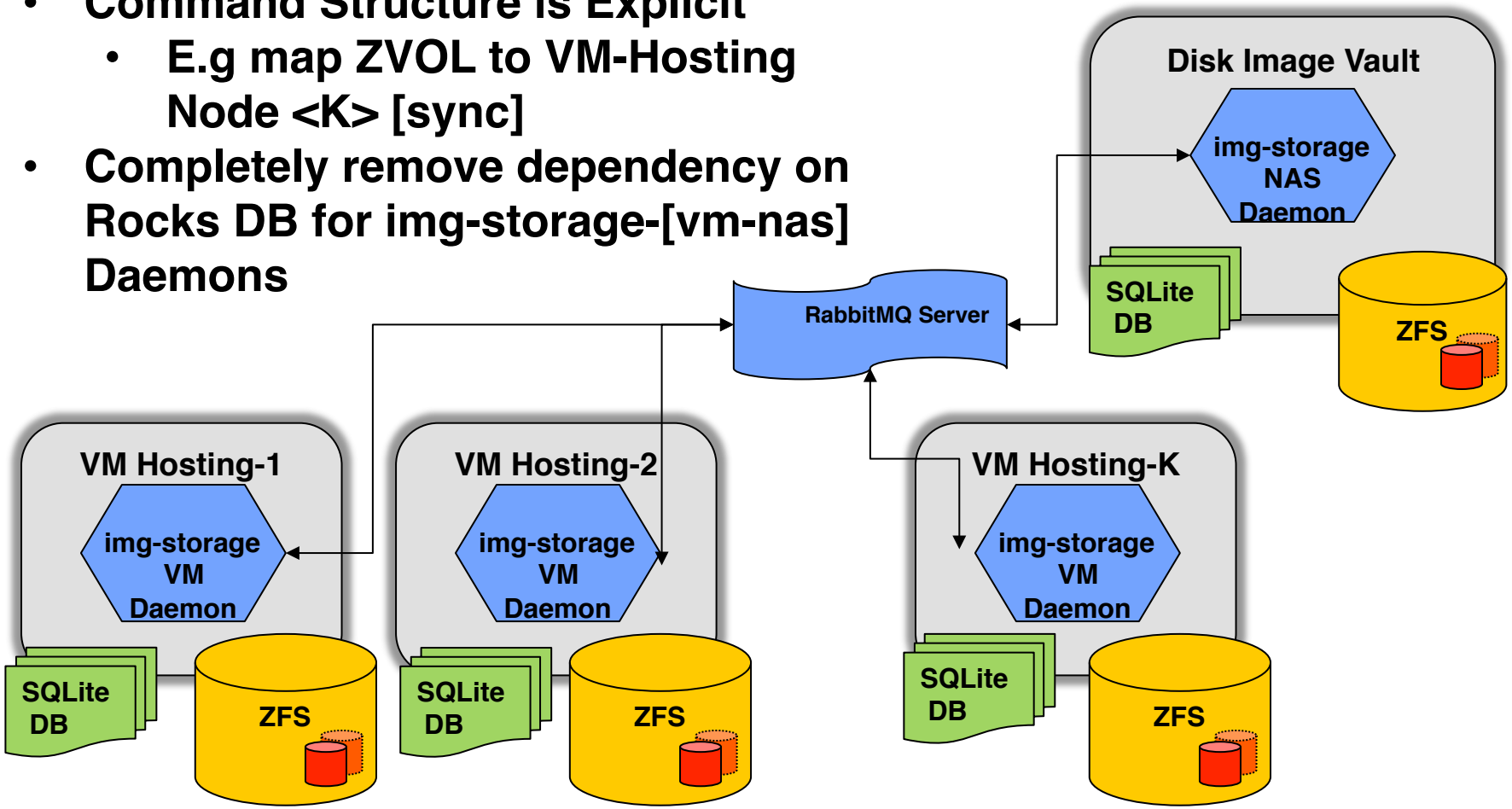




**img-storage-vm**
- Create local device mapper
- Merge
- Replicate
- Shutdown

**img-storage-nas**
- Create virtual volume
- Periodically replicate
- Status
- Manage Snapshots

# Taking Rocks Dependencies out of Key Components

- Command Structure is Explicit
  - E.g map ZVOL to VM-Hosting Node <K> [sync]
- Completely remove dependency on Rocks DB for img-storage-[vm-nas] Daemons



# *Improvements from PRAGMA29*

- **Adding per volume synching parameters**
  - Frequency of sync
  - Explicit next sync time
  - Setting upload/download speed (throttling)

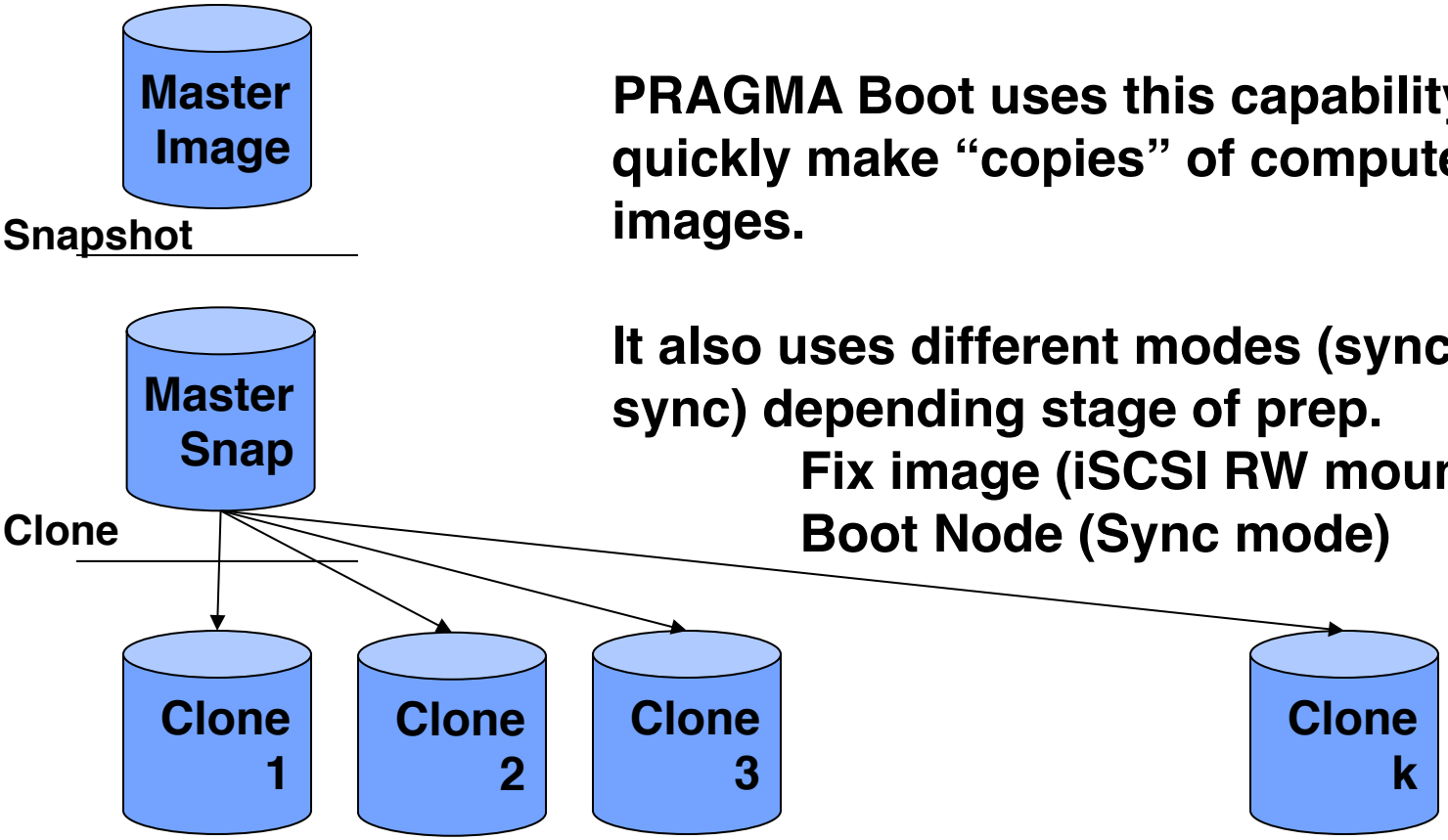
```
# rocks list host zvolattr nas-0-0 hosted-vm-0-6-0-vol
ZVOL                  FREQUENCY NEXTSYNC   UPLOADSPEED DOWNLOADSPEED
hosted-vm-0-6-0-vol   --------- 1454039927 ----------- -------------
```

# More Parameters

- **"Global" parameters per Storage Vault**
  - How many simultaneous sync
  - Default sync frequency (e.g. 5 minutes)
  - Name (RabbitMQ) and which network to use

```
# rocks list host storageattr nas-0-0
ATTR                VALUE
default_pool        tank1
img_sync_workers    8
frequency           300
name                nas-0-0
network             local
```

# Support for pre-existing disk images and clones




**PRAGMA Boot uses this capability to quickly make "copies" of compute node images.**

**It also uses different modes (sync/non-sync) depending stage of prep.**

- **Fix image (iSCSI RW mount)**
- **Boot Node (Sync mode)**