

# Encompass

## *Functionality-Centric Image Management System*

Oleg Goldshmidt

`olegg@il.ibm.com`

Haifa Research Lab



# Agenda

- Managing functionality
- Encompass basics
- Encompass operation
- Image customization
- Storage architecture options
- Status



# Managing Functionality



# Customer Requirements

- I need ...
  - ... a web server
  - ... or a database server
  - ... or  $N$  WebSphere nodes
  - ... or a cluster of  $M$  Linux machines
- focus on **FUNCTIONALITY**
  - possibly with some additional requirements, e.g., performance



# Head of IT's Reaction

- OK, we have
  - $N$  racks full of BladeCenters with 14 blades in each
  - $M$  storage controllers with  $S$  GB space in each
  - somewhere among these resources there should be enough storage space and computing power to satisfy the requirements
- do we have enough ...
  - ... floor space?
  - ... rack space?
  - ... BladeCenters?
  - ... electrical power? AC? sockets? bandwidth?
- focus on **RESOURCES**



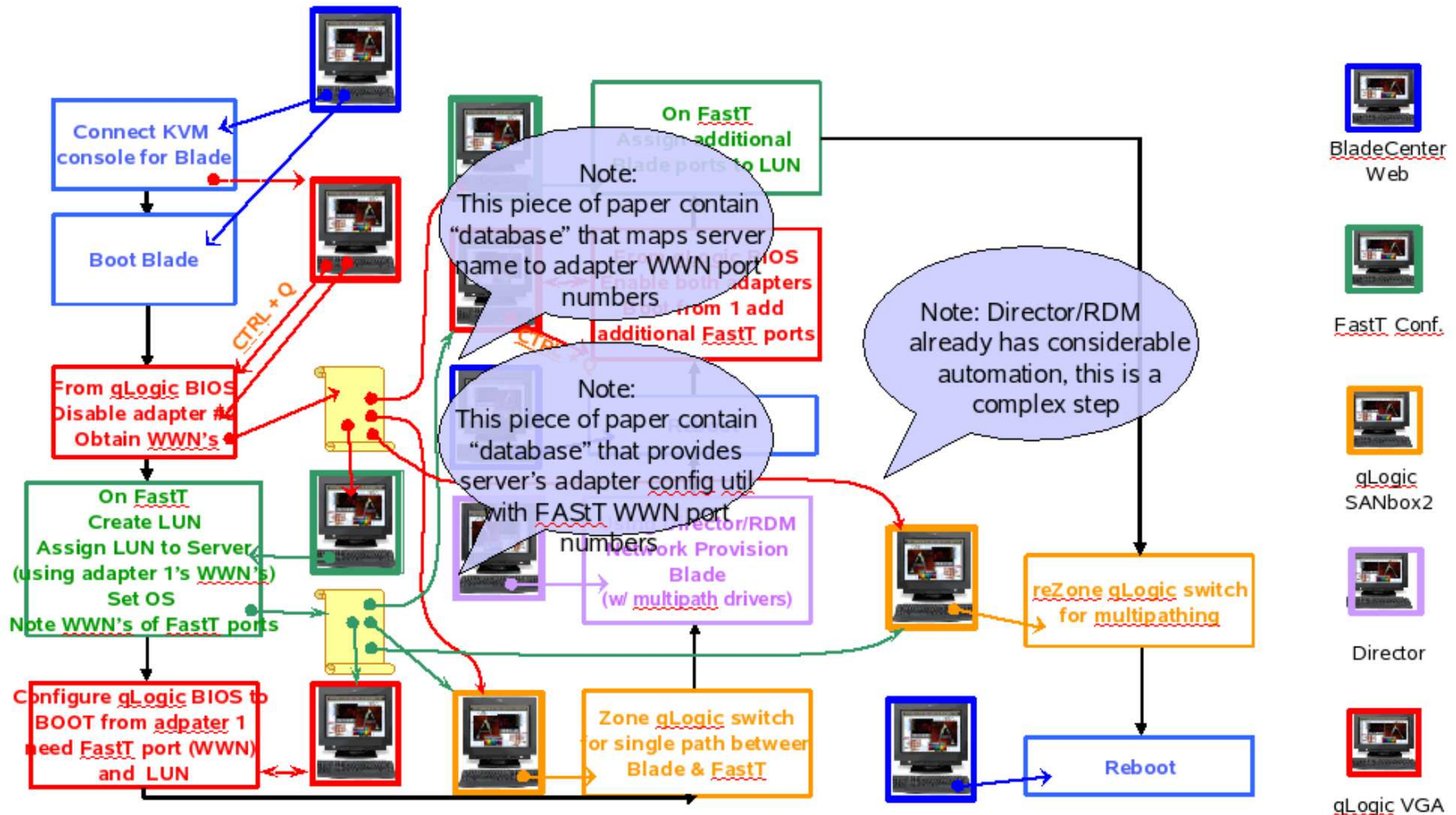
# SysAdmin's Headache

- I need to install (configure, customize, test ...) OS  $A$ , with application packages  $B$ ,  $C$ ,  $D$ , and dependencies
  - Oh, there is a suitable image on the install server
  - Now I need to allocate  $N$  volumes of  $T$  GB each
  - ... and find  $N$  spare machines
  - ... and configure the fabric
  - ... and attach the volumes to the machines
  - ... and make sure that the network is configured
  - ... and write down what is installed on these volumes and which machines are attached to them
  - ... and there seem to be 5 unused instances of this web server, why didn't I reuse them?
- focus on **HEADACHE**



# Just How Much Headache?

## Deploy image on remote storage and boot



# SSPT et al.

- The previous slide was stolen from a presentation about SSPT — Server Storage Provisioning Tool
- SSPT provides a level of automation of the mess in the previous slide
  - talks to the storage controller, HBAs, FC switches, and configures everything needed to connect a storage volume to an HBA port
  - eliminates the need for the little yellow pieces of paper
  - limited HW support for now
- SMI-S, CIM, iSNS will all hopefully help
- but all that is still about volumes and switches and storage controllers and HBAs



# Managing Machines

- a machine is a familiar physical embodiment of
  - resources (CPU, memory) that determine performance
  - functionality (programs and data on disks)
- technologies that can potentially cause a paradigm shift
  - remote (shared) storage separates functionality from computing resources
  - virtualization creates a single resource pool
- traditional management paradigm is still prevalent
  - pretend that everything works as before: remote disks are just like local, virtual machines are just like physical



# Where Are The "Smarts"?

- dealing with physical devices is essential, but not sufficient for effective IT management
- none of the existing tools deal with “content” or “functionality”
  - what is stored on this volume?
  - are the contents compatible with the machine the volume is connected to?
  - does this setup provide the functionality that the user wants?
  - are the resources utilized optimally?
- the low-level technologies are enablers
- the “smarts” are left to the sysadmins



# Encompass: Management By Function

- some “smarts” — keep track of what is where
- maintain a library of system images tagged by functionality
  - “Apache 2.0.54 on RHEL 4 with kernel 2.6.13-ELsmp”
  - “web server for client X”
- provide a means to clone and customize an image automatically (or with minimal intervention)
- leverage existing technologies (such as SSPT) for basic tasks, e.g., discovery, provisioning
- OS and application agnostic (modulo customization)
- mechanism, not policy (an “enabler”)



# Encompass Basics



# Encompass Terminology

- **image**: a bootable system image, comprising one or more volumes, with the MBR and all of the software; provides a certain functionality (e.g., web server) when **deployed**; may be stored on block storage devices (e.g., SAN), files, etc..
- **machine**: a set of computing resources (CPU, memory, etc.) that together form a deployment platform for an image; can be **physical** or **virtual**.
- **server**: the collective functionality of the software stack of an image.



# Encompass Terminology (Cont.)

- **masters and clones**: there is a library of master images; each master corresponds to a specific type of **server** (i.e., functionality); masters are never used directly, but are **cloned**; clones may be customized and booted on **machines**.
- **customization**: modification of the image contents and **metadata** to prepare it to run under particular circumstances
  - parameter configuration (hostname? static IP?)
  - set of services to start on boot
  - choose suitable OS kernel, set of drivers (initrd)
  - possibly p2v/v2p/v2v
  - application-specific



# Image Metadata

- storage volumes
- HW profile — machine requirements
- SW profile — server functionality, system utilities
- state (discussed in detail below)
- related images
  - reference to master (for clones)
  - list of clones (for masters)
  - version control???
- machine the image is assigned to
- customizations (discussed in detail below)



# Potentially Useful Scenarios

- populate the Encompass catalog by adding new images
- capture a new master image from a donor system
- create one or more clones of a specified master image (includes customization)
- deploy a specified clone image on a chosen machine (includes customization)
- deploy a server of a particular type on a machine (as opposed to a specific clone) — may need to find or create a suitable machine
- migrate a server from one machine to another
  - for failover, for resource optimization, etc.



# Potentially Useful Scenarios (Cont.)

- machine reassignment
  - serve customers during the day, compute payroll at night
  - run electronic trading system from 9 to 5, price mortgage pools nightly
- server substitution (change propagation)
  - deploy a critical update on all the servers of type X
  - all IIS to be substituted by Apache...
  - brings servers to a common baseline — often useful
  - easy rollback

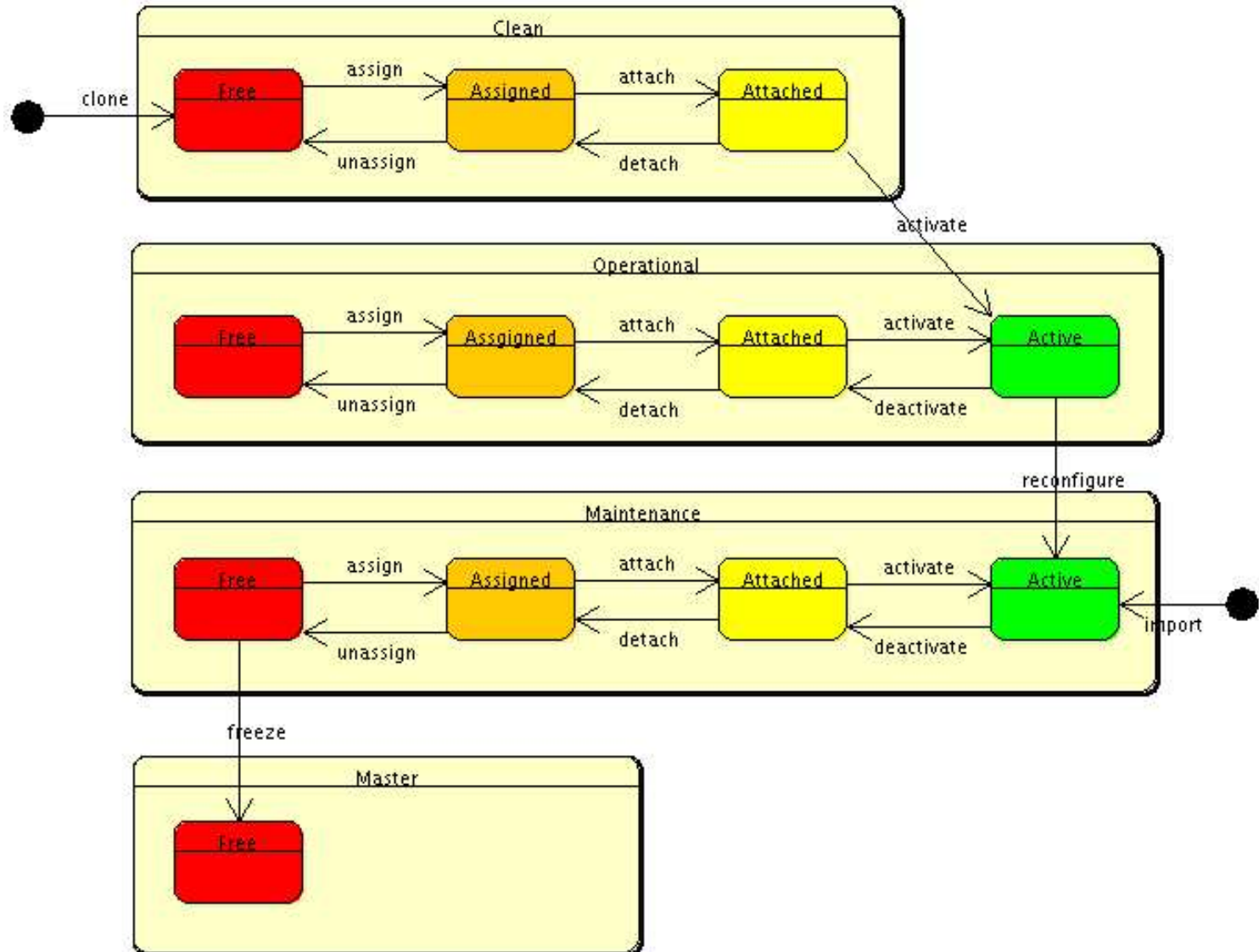


# More Scenarios: Export and Import

- **export**: take the set of master images, add a snapshot of the Encompass metadata describing the current deployment solution (what server runs on what machine, customizations, configurations, etc.), dump on removable media.
- **import**: read the recorded data at a different site, clone the master images as needed, deploy the servers, configure as necessary...
- useful for disaster recovery, opening a new branch, providing a turnkey solution to a client, etc..
- not always necessary (or possible) to recreate the exact configuration



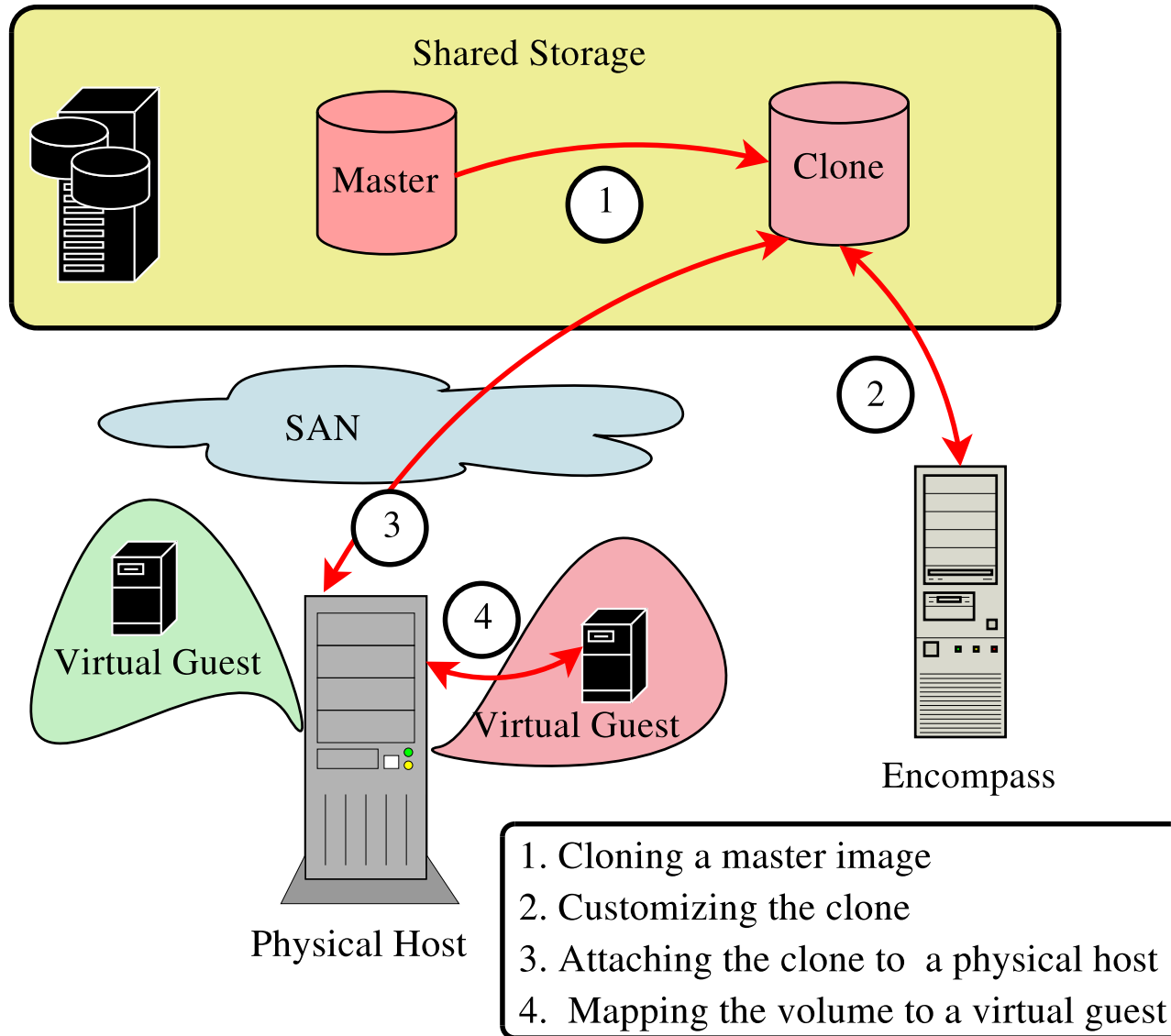
# Image State Diagram



# Encompass Operation



# Encompass Operation



# Mapping To Virtual Guests (in Xen)

- attach a volume to the physical host: `dom0` should see it as, e.g., `/dev/sd<something>`
  - do SCSI scan in `dom0` as a part of the `attach` operation, e.g.,

```
echo '- - -' > \
/sys/class/scsi_host/host0/scan
```
  - map the LUN to device, e.g.,

```
ls -l /sys/block/sdm/device
```
- map the device that `dom0` sees to the proper device in `domU`, e.g.,

```
disk = [ 'phy:sdm,sda,w' ]
```
- if the volume is a file on NAS,

```
disk = [ 'file:<path>,sda,w' ]
```



# Image Customization



# File-Based Customization

- UNIX philosophy: everything is a file
- procedure: mount a volume, add, replace, or modify the specified, files, unmount
  - target machine needs not be involved, can be done on a “utility server”, parallelized for many clones
  - e.g., for clones to be deployed in virtual machines the host OS, VMM, or a privileged domain may perform customization



# File-Based Customization (Cont.)

- what if **not** everything is a file?
  - example from Microsoft world: `sysprep`
  - we do require unattended customization, so any proprietary tool should be able to read input non-interactively
    - `sysprep` can be given an input file
    - boils down to file replacement anyway
- special case: adding an executable/script to boot/init
- **strive to achieve zero or minimal customization!**



# Customization In Image Metadata

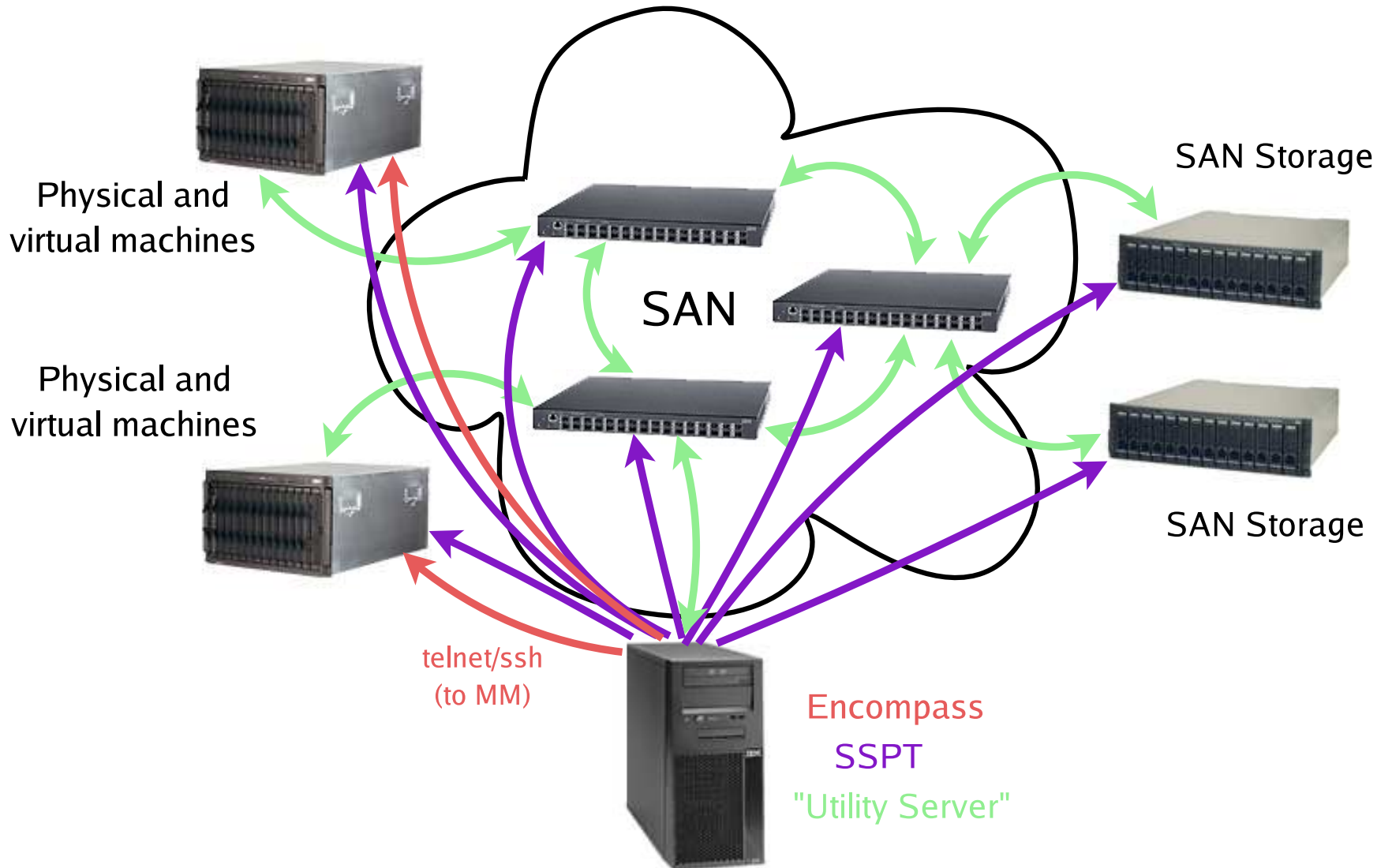
- what to customize — in master metadata
  - ordered list of “customization items”
  - item: volume, partition, path, customization type (e.g., file replacement, regexp replacement), description, prompt
  - possibly defaults (to uncustomize)
- customization input — in clone metadata
  - input for each customization item (may be given interactively)
  - customization state (what has been customized already?)



# Encompass Storage Architecture Options



# SAN/SSPT



# SAN/SSPT Pros and Cons

## ● pros:

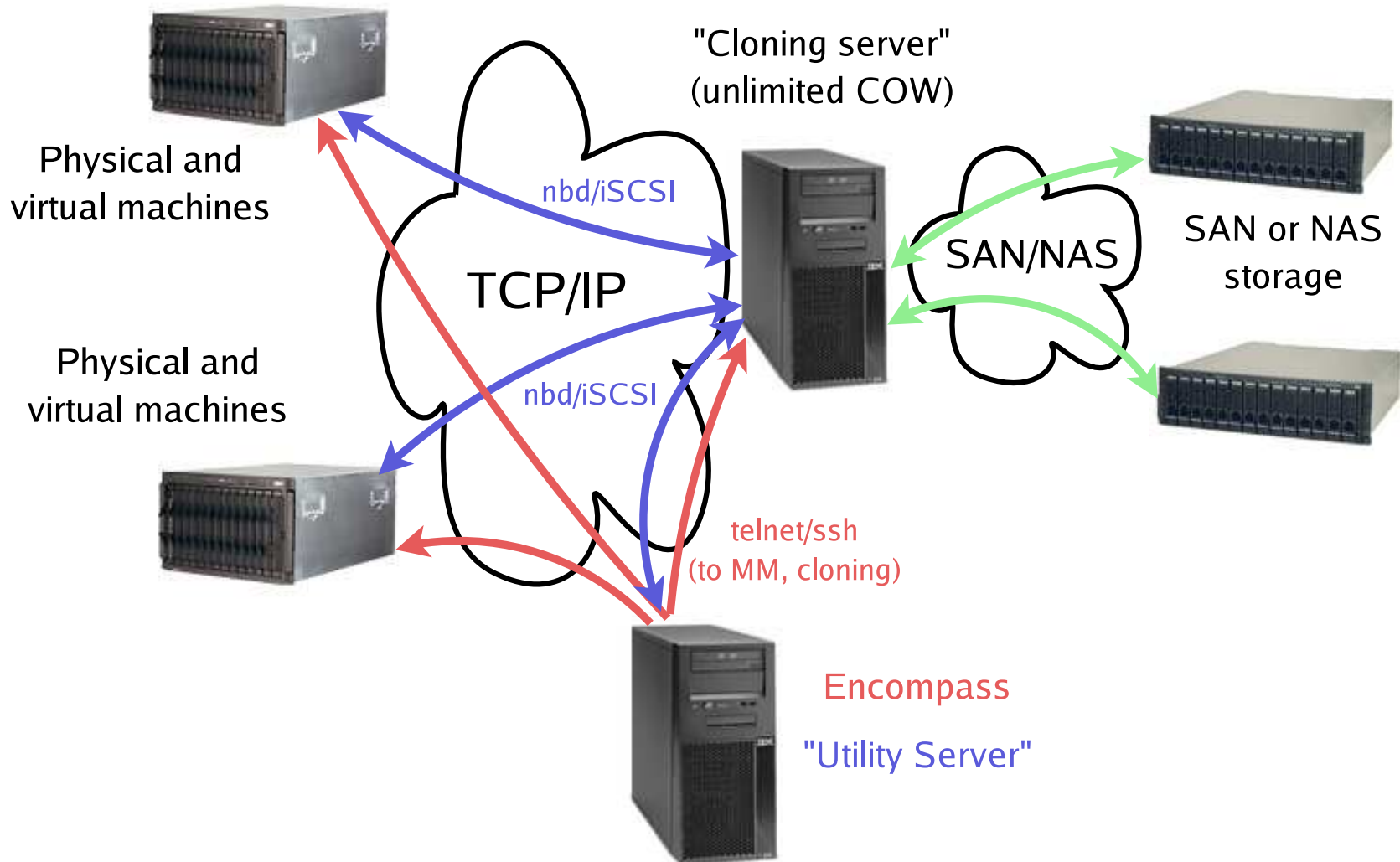
- SAN is the most common storage infrastructure
- SSPT takes care of many low-level details, talks to a wide variety of FC switches, etc., an STG product

## ● cons:

- a lot of complications due to zoning, masking, etc.
- SSPT is stateless by design, gathers state anew for each invocation, very slow
  - not an issue in its original context, becomes an issue for dynamic deployment, scalability
- efficient COW cloning (“flash copy”) is limited to at most a few copies on all the storage controllers (an implementation issue)



# Cloning Server



# Cloning Server Pros and Cons

## ● pros:

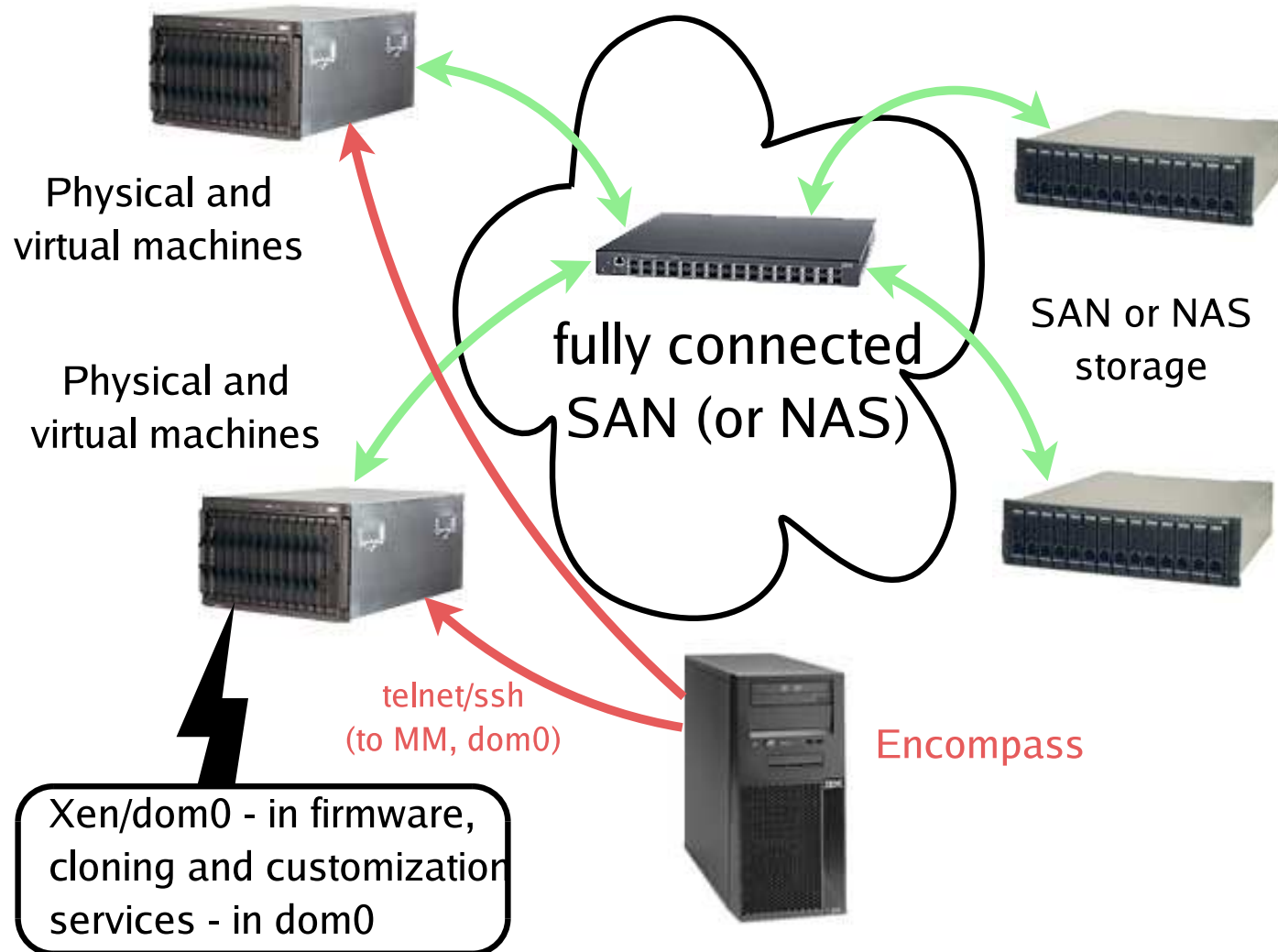
- unlimited, efficient COW cloning
- eliminates SAN complications, SSPT
- much less management communication, one (type of) entity to manage on the storage side

## ● cons:

- introduces an additional storage infrastructure to maintain
- the cloning server is on the data path
  - potential bottleneck
  - need to provide reliability, availability, etc.



# XenoBlades (CSO)



# XenoBlade Pros and Cons

## ● pros:

- cloning and customization functionality hidden in dom0, which is a part of the architecture anyway
- no additional storage infrastructure to manage: dom0 is in the data path anyway
- suitable for fully connected storage networks
- Encompass needs to communicate with dom0s only

## ● cons:

- specific to the chosen virtualization technology (but can, in principle, be implemented in different ones)
- fits the vision of a part of STG only at the moment
- suitable for fully connected storage networks only (otherwise need to manage fabric/controllers)



# Encompass Status



# Encompass Status

- prototype-level code
  - machine and storage discovery, persistent image metadata, merging discovered resources and metadata into a consistent picture, full image lifecycle implementation
  - portable C++
  - storage: SAN (DS4xxx), NAS
    - SAN implementation relies on SSPT (Java)
  - physical machines: IBM BladeCenter
    - mostly for MM-assisted discovery
  - virtual machines: Xen, adding VMware
- much interest inside IBM Research, STG, LTC
- integrating with other projects as provisioning engine



# Future

- beyond Research:
  - VSM, Director — Image Management to be integrated in 2007
    - separate development effort, Encompass serving as a research prototype, accumulator of know-how
- largely unexplored areas
  - networking, security, cloning technologies
- other ideas: a “smart” storage controller that is aware of the stored functionality and helps provision and manage it?
- separation of concerns — a technology for a “flat” world.



# More information:

<http://encompass.haifa.ibm.com>

