Table of Contents

Virtualization image catalog API

Description

This document is meant as a design document for the image catalog API. It will be based on the Virtual Machine Image Catalog (VMIC) design document.

User interface design ideas

Definitions

- VMIC: meta data bit of the catalogue also known as inventory
- Repositories: contains the actual images

Required information

General information:

- a list of trusted endorsers
- a list of images in the repository

The following informations are required for each trusted endorser:

- endorser DN
- location of the signed list of endorsed images of this endorser

The following informations are required for each image in the site repository

- A globally unique identifier (UUID)
- A VO tag
- Image registration time
- suitable checksum
- OS version and 64/32bit status
- Hypervisor requirements
- DN of endorser
- (local?) image download information

For the definition of these fields check the VMIC design document

Optional information

- detailed description of the image
- additional information for publishing system

Repositories

The repository of the actual image files listed in the catalog. The repository plus the catalog is what is called VMIC.

Roles and ACLs

Endorser

The endorsers of an image can:

- update and publish the list of images endorsed by him
- notify sites about urgent updates (?)

The endorser cannot:

- remove images from repositories

Site admin

The site admin can

- manipulate the list of trusted endorsers on the site VMIC
  - ♦ add new endorsers plus their DNs
  - ♦ remove endorsers from the list
- invalidated individual images in the local VMIC
- manipulate the VMIC contents
  - ♦ archive unused images in the repository
  - ♦ remove images from the catalog

The site admin cannot:

- modify images themselves (contextualization is allowed)

Site image distribution system

The local image distribution mechanism is up to the site. It must be compliant with the security model.

The site image distribution system must be able to

- request the stage-in of a specific image by UUID or VO-tag

End user

- query the site catalog to see what is supported by the site

Use cases.

The use cases have been broken down by actors and then actions after this we have a quick look at parameters.

Actors

- Endorser list
- Enstatniation
- Job
- VO
- Endorsor

The following Actors are then broken down my actions they can perform.

Endorser list Actions.

- Update image list.
- Revoke image from list.
- Error handling.

Image lists are gathered regularly by the VMIC. When images are no longer in the list they are considered no longer endorsed. Endorser lists are unavailable they should be regarded as they are still available, as a fall over they should have an expiry date.

Endorser list Metadata.

The endorser list must contain.

- DN of endorser.
- Date of last update of list.
- List of images to endorse.
    - Image UUID
    - Image Creator
    - VO of image (multple VO's ? so should it be a list)
    - Date Image created.
    - Endorser image Location URI
    - Checksum of image.

Enstantiation Actions.

- Image approved.
- Get UUID by Tag.

Image approved.

Image approved request must contain.

- UUID
- VO (the Vo the image is to be used by).

Get UUID by Tag.

Getting the UUID from a tag

- Tag
- VO

Job Actions.

The VO needs to know what images/Tags are approved at a site

- List available images.
- List available tags.

List Available Images/Tags.

- VO

VO Actions.

- Publish VO Endorser.
- Revoke endorser.
- List UUID's by tag.
- List Tags.

Publish VO endorsers.

A list of approved Endorsers for the VO. This allows the Endorser to add images to a VO tag.

- VO
- List
    - Endorser DN.

Publish VO endorsers Questions:

Should endorsers be required to be registered by VO.

Should images be allowed to be used by other VO's. (this would break the ability of VO's to stop all endorsers endorsements?)

Revoke endorser.

Assumed to be just removing DN from list of endorsers.

List UUID's by tag.

Same as Job Actions.

List Tags.

Same as Job Actions.

Unstructured

- Site A wants to share images with the world
- Endorser E1 has a new image which he wants to push to sites
- Endorser E1 wants to replace/update an existing image with a new one
- Site A wants to add an image which is published by site B
- A security instance found an issue with an image
- A user demands to run a specific image on the site

Work flows

Some procedures/work flows are already defined in the VMIC design document.

Implementation

Basics guide lines

- All methods should have one parameter in and out.
- This will be the simplest type possible. Boolean, list or dictionary
- Minimum dictionary content should be clearly marked up to the end user.
- Deprecated dictionary content shall be ignored if newer dictionary content is added, to allow parameter changes.

Functions and methods

In this model we may need:

- functions/methods to manage the global bit of the inventory of the catalog
  - ♦ add/remove/modify endorser list
  - ♦ add/remove/modify image lists
- functions/methods to query the inventory
- functions/methods to manage individual images
  - ♦ synchronize catalog with endorser lists (global/individual)
- functions/methods to manage the repository
  - ♦ add/archive/remove images to the repository

Sample implementation

This is a straw man implementation of Django and soon XML rpc interfaces.

Source code

```
svn co http://svnweb.cern.ch/guest/vmimagecat
```

This topic: HEPIX > Virtualization
Topic revision: r4 - 2010-04-30 - UlrichSchwickerath