HEPIX VWG Image transfer.

HEPIX virtulisation working group : 6 Month update

A short summary of the Documentation produced.
HTML PDF A4 PDF Letter

Owen Synge
HEPIX VWG Image transfer.
HEPIX Spring 2012

HELMHOLTZ ASSOCIATION

DESY

Hepix Virtualisation Working Group introduction.

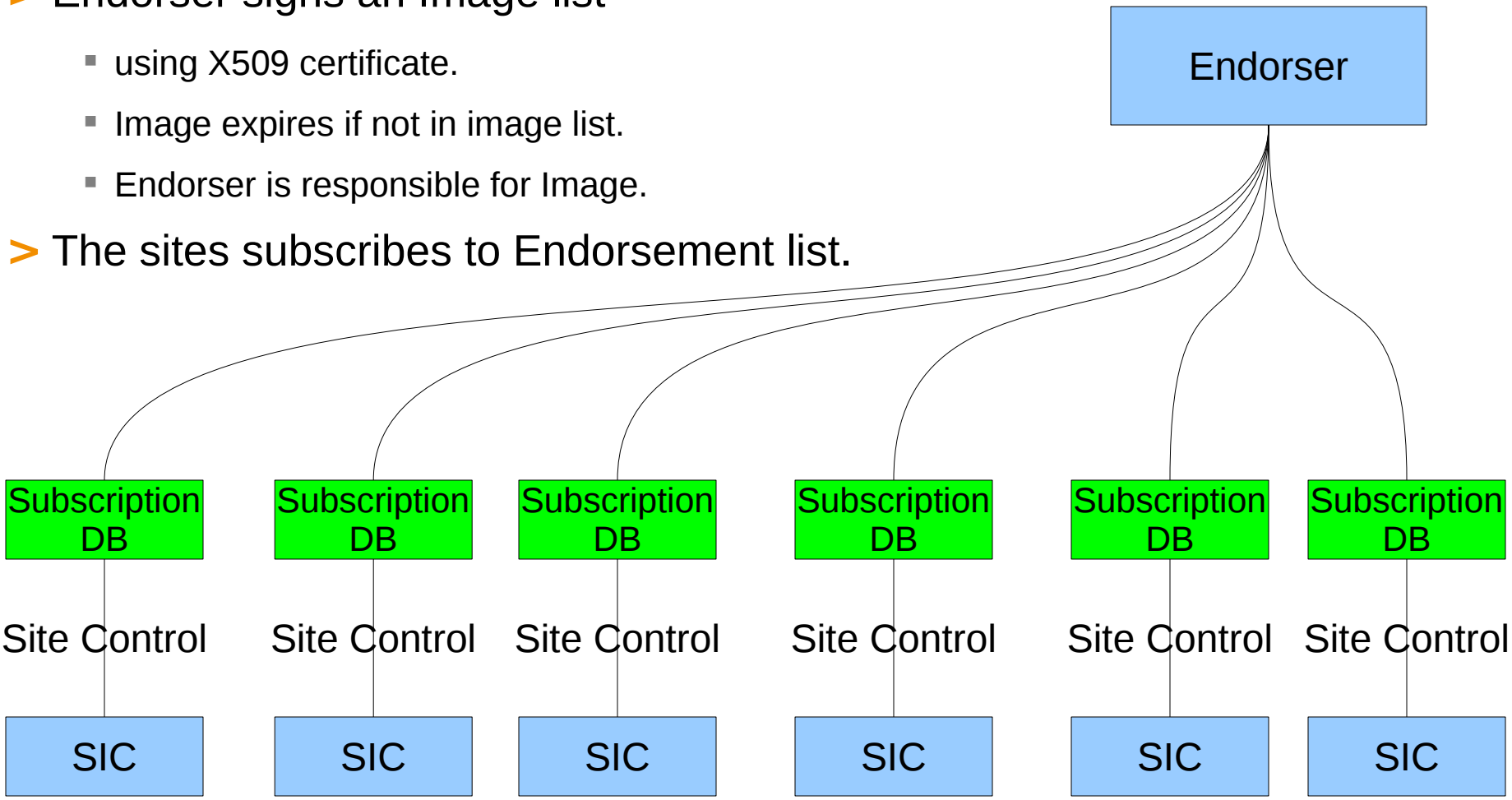
- Aim to support images running on multiple sites.
- To make it easy for users/scientists/sites to manage images.
  - Simple way to publish images.
    - Webserver/ storage element can be used to share images.
    - Easy to use CLI for generating secure image list.
  - We are all subscribing to CernVM project.
  - Easy for sites to automate upgrades.
- Some sites will not need to trust the VM Image.
  - Sites with Virtual Network per user.
    - Giving users safe network isolation from each other.
  - User communities that are using secure protocols.
  - Still need to get the right VM images so security is still important.

HELMHOLTZ ASSOCIATION

DESY

HVWG model is Publish and Subscribe like a podcast!

- Endorser signs an Image list
  - using X509 certificate.
  - Image expires if not in image list.
  - Endorser is responsible for Image.
- The sites subscribes to Endorsement list.



Meta-data Security.

> Meta-data authenticity.

- X509 + signatures. (SMIME or XML signatures)
  - Gives non repudiation, and confidence in who endorsed.
  - Give tamper proof message.
  - Signature can be checked by all clients,
  - Allows checking of historic meta-data changes.
- Version number.
  - Prevents man in middle attacks replaying older images.
  - Man In Middle attempts to return an old list blocked by this.
- UUID on Image and Image list
  - Allows messages to be identified.
  - So messages cannot effect each other.
  - So images can be expired and updated.

Image security.

> Image to Meta data binding.

- Cryptographic hashes.
  - It is easy to compute the hash value for any given data.
  - It is infeasible to generate a message that has a given hash.
  - It is infeasible to modify a message without hash being changed.
  - It is infeasible to find two different messages with the same hash.
- Chose to use sha512 and file size to validate data.
  - Following Stratuslabs recommendation.
- Other hashes can be added.
  - If sha512 and size are later found to be too week.
- URI to retrieve image.
  - Can be cached locally.
- Each image has a UUID
  - So we know which image is expired and which is upgraded.

Example of what signed meta-data can look like


```
------EAE3006C97F670EE450F46AC8DF4C070
{
   "dc:date:created": "2011-03-10T17:09:12Z",
   "dc:date:expires": "2011-04-07T17:09:12Z",
   "dc:description": "a README example of an image list",
   "dc:identifier": "4e186b44-2c64-40ea-97d5-e9e5c0bce059",
   "dc:source": "example.org",
   "dc:title": "README example",
   "hv:endorser": {
      "hv:x509": {
         "dc:creator": "Owen Synge",
         "hv:ca": "/C=DE/O=GermanGrid/CN=GridKa-CA",
         "hv:dn": "/C=DE/O=GermanGrid/OU=DESY/CN=Owen Synge",
         "hv:email": "owen.synge@desy.de"
      }
   },
   "hv:images": [
      {
         "hv:image": {
            "dc:description": "This is an README example VM",
            "dc:identifier": "488dcdc4-9ab1-4fc8-a7ba-b7a5428ecb3d",
            "dc:title": "README example VM",
            "hv:hypervisor": "kvm",
            "hv:size": 2147483648,
            "hv:uri": "http://example.org/example-image.img",
            "hv:version": "1",
            "sl:arch": "x86_64",
            "sl:checksum:sha512":
"8b4c269a60da1061b434b696c4a89293bea847b66bd8ba486a914d4209df651193ee8d454f8231840b7500fab6740620c7111d9a
17d08b743133dc393ba2c0d4",
            "sl:comments": "Vanila install with contextulization scripts",
            "sl:os": "Linux",
            "sl:osversion": "SL 5.5"
         }
      }
   ],
   "hv:uri": "http://example.org/example-image-list.image_list",
   "hv:version": "1"
}
------EAE3006C97F670EE450F46AC8DF4C070
Content-Type: application/pkcs7-signature; name="smime.p7s"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7s"

MIIHdAYJKoZIhvcNAQcCoIIHZTCCB2ECAQExCzAJBgUrDgMCGgUAMAsGCSqGSIb3
DQEHAaCCBSUwggUhMIIECaADAgECAgIz7DANBgkqhkiG9w0BAQUFADA2MQswCQYD
VQQGEwJERTETMBEGA1UEChMKR2VybWFuR3JpZDESMBAGA1UEAxMJR3JpZEthLUNB
MB4XDTExMDExMDE1MDMxN1oXDTEyMDIwOTE1MDMxN1owRjELMAkGA1UEBhMCREUx
EzARBgNVBAoTCkdlcm1hbkdyaWQxDTALBgNVBAsTBERFU1kxEzARBgNVBAMTCk93
ZW4gU3luZ2UwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCkgbPFZrVL
pmwf7GKBBFkwTK5V7RmlupsU3Z3FqdfMnJGn2NrrnHIhthUTCTq4WbLIZTbOEh0n
JqZgZBvYcwJV4V9pais4YlsEug+JLMbB9hZ6e2XgdjXWgLqz6vBSIf6KXi4KhCxe
a4FylvIk7OtY+bg0mg5IFHib6uP7fXhFKdBEapoi+B05wpluBMA+2DBdSt+rjzA8
SwiHUuan60VIyJAxammyOe3IKSpwyBXkQ10XjIhIpoSavqYXJboFOVzUcqxawdbX
Con2W8QfiwFKYupohG/VTusDXFT2MP4k+KxG3/rTTPWUDJme7VUPv3+CTcEO+z4v
X8/XhI44oAXlAgMBAAGjggInMIICIzAMBgNVHRMBAf8EAjAAMA4GA1UdDwEB/wQE
AwIE8DAdBgNVHQ4EFgQUGakUy66kgvuINBIf18WBXjGolqYwXgYDVR0jBFcwVYAU
xnXJKKzRC/w8/7m1HtNfO4BiEjShOqQ4MDYxCzAJBgNVBAYTAkRFMRMwEQYDVQQK
EwpHZXJtYW5HcmlkMRIwEAYDVQQDEwlHcmlkS2EtQ0GCAQAwHQYDVR0RBBYwFIES
b3dlbi5TeW5nZUBkZXN5LmRlMB8GA1UdEgQYMBaBFGdyaWRrYS1jYUBpd3luZnpr
LmRlMDUGA1UdHwQuMCwwKqAooCaGJGh0dHA6Ly9ncmlkLmZ6ay5kZS9jYS9ncmlk
a2EtY3JsLmRlcjAaBgNVHSAEEzARMA8GDSsGAQQBlDarLAEBAQUwEQYJYIZIAYb4
QgEBBAQDAgWgME4GCWCGSAGG+EIBDQRBFj9DZXJ0aWZpY2F0ZSBpc3N1ZWQgdW5k
ZXIgQ1AvQ1BTIHYuIDEuNSBhdCBodHRwOi8vZ3JpZC5memsuZGUvY2EwJAYJYIZI
AYb4QgECBBcWFWh0dHA6Ly9ncmlkLmZ6ay5kZS9jYTAzBglghkgBhvhCAQgEJhYk
aHR0cDovL2dyaWQuZnprLmRlL2NhL2dyaWRrYS1jcHMucGRmMDMGCWCGSAGG+EIB
AwQmFiRodHRwOi8vZ3JpZC5memsuZGUvY2EvZ3JpZGthLWNybC5kZXIwDQYJKoZI
hvcNAQEFBQADggEBAMbn91TOQ6r4D/aKwgIFXiXe40B7iccz/P5pCFSi1R6IC3KH
Ui4s/f9iAGl9rA21h8QAaRaJ/h1OQNlgMZbc9jDCWcqxr8wQTYAQDiBkspLT68ZO
5xVFRiq3HjkkhwnFfFzsNSiLFYZTRjChPluclYG3TEvSg8dz9Lv/IEJxE5C5lZ2d
e3CSu0vcD0DESiu/sVqPOOHi8NL/59U2ine3z23Y+piCabQCxjT0inT2MmR8UNDF
ij2JJYxlt56U/SQCEe0304w3x1jIg8vcpm4dfh+L2IjJ9hVfEeLaCyhv9Wjbmu5O
vk0yLjcEZ7b4RKeo7djVYh+5kCWJYCr/W6uGW44xgglXMIICEwIBATA8MDYxCzAJ
BgNVBAYTAkRFMRMwEQYDVQQKEwpHZXJtYW5HcmlkMRIwEAYDVQQDEwlHcmlkS2Et
Q0ECAjPsMAkGBSsOAwIaBQCggbEwGAYJKoZIhvcNAQkDMQsGCSqGSIb3DQEHATAc
BgkqhkiG9w0BCQUxDxcNMTEwMzEwMTczMzU1WjAjBgkqhkiG9w0BCQQxFgQUd43y
VT05Zk+7acFF+EeqExNI57cwUgYJKoZIhvcNAQkPMUUwQzAKBggqhkiG9w0DBzAO
BggqhkiG9w0DAgICAIAwDQYIKoZIhvcNAwICAUAwBwYFKw4DAgcwDQYIKoZIhvcN
AwICASgwDQYJKoZIhvcNAQEBBQAEggEAkA0RgB5AkGIYvFsFETzx7QHKWu9qas5k
vIHn2a+EpRE9K1p+qrFNzS53E2BGqubyRcePfgG/WyGqYOK2h20d6GZH+ENUFkvM
EAthbvQaHye6WEvF/0GUrr0QUBT1gQswkkryPHcqTVmJANQORakkNvCwynEBmfSC
vb2TEppRuOCmxx3zqrzMr7zPNPY4w2+YaXQ1fHfmEmOrlf0ImP20TyTKIoQWqzbq
WXwlRhZBUoD9zfiEM/jFvOvkuxLkQeiEcSzlLAGHXsHJ3anPMX9sobJFbJl0wYdN
sUOInHRhksokh2ow68KZK4vXLI173v5yZE7FZZ1Gl9T+YpkmOIW4iQ==

------EAE3006C97F670EE450F46AC8DF4C070--
```


This example has a single image. Multiple images can be published with a single image list.

Updates to Metadata

- New optional fields added to the meta-data for an image.
  - Required no software changes.
- Field hv:core_minimum
  - Minimum number of cores to run VM.
  - Integer.
- Field hv:ram_minium
  - Minimum amount of ram to run VM
  - Integer in bytes.
- Field hv:format
  - More than one image format available.
    - HV has upgraded its default format.
  - Must still be documented as a enumerated list with definition of each format.
    - Important for CERN VM as at least 10 different formats supported.

Image Formats and the HVWG

- Old image format was a simple virtual disk compressed with gzip.
  - Lowest common denominator
    - Every one has tools to work with Raw disks. (Kpartx and LVM in all Linux distributions)
    - All clouds can be bmade to work with raw images.
  - Multicore VM's not accounted for.
    - Each core needs 2G RAM for HEP.
    - Swap space should be related to RAM.
    - Each core needs 10-20 Gb Disk for scratch
- Changed image format.
  - Root and boot file system in raw disk.
  - Swap and scratch space bound at boot time.
- VM image transfer software is data format neutral
  - So no code changes where required in image transfer software.

Desy Image SHaring: DISH

- Based on dCache as a http(s) server. (We could have used Apache or NginX)
  - Stable service for the past 6 months – providing images to DESY Cloud.
  - Easy to check host key. Preventing man in middle attacks.
- Size of images was dependent on cores.
  - Large files slow and difficult to handle.
  - 10Gb Spool + 2Gb Swap per core.
  - Images with size related to number of Core's could not continue.
    - Explosion in number of images.
- Creating images with Quattor and kickstart + puppet.
  - Quattor for internal DESY images.
    - Makes images identical to Batch WN on Grid.
  - Kickstart + puppet for HEPIX format images.
    - Not being coupled to main production system allows more exploration.

DISH publishing image lists

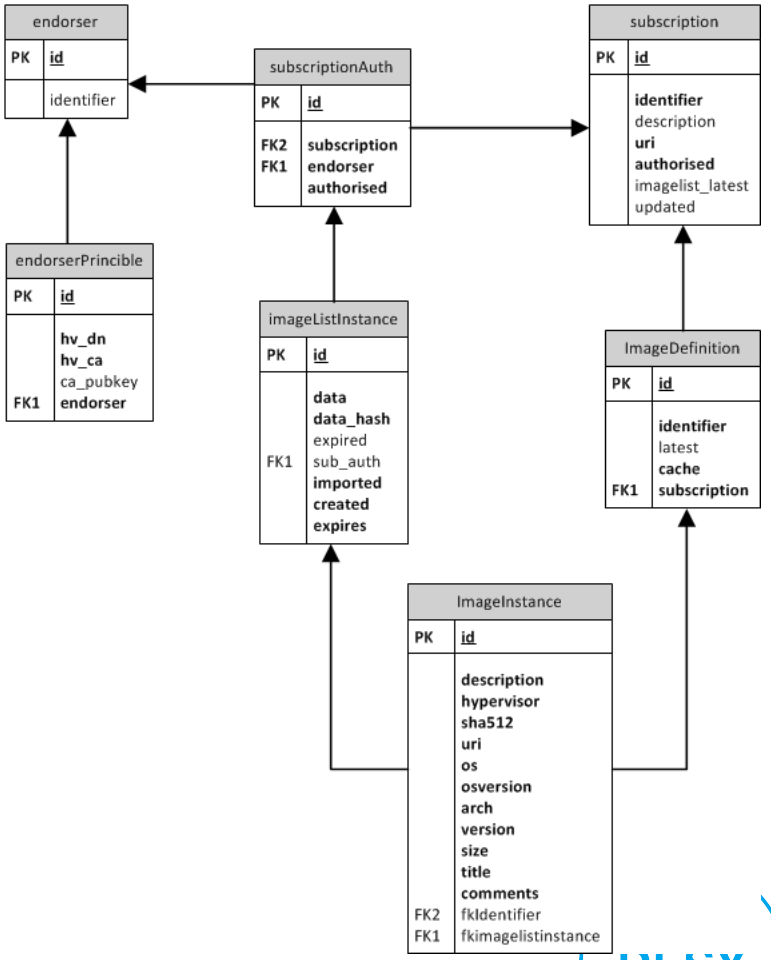
- CLI image list publisher (upload images and imagelist manually).
  - Simple to use but you have to check the meta data by hand.
  - Image updates did not update metadata automatically.
    - Frequent updates suggested too fiddly for a production service like DISH.
- Automated publisher Mk 1. (Only suitable for DESY)
  - Read public image list and updated this.
    - Massive reduction in typos and faster updates.
  - No bulk operations
    - Each image update required a image list to be updated.
  - Updating meta data outside release process impossible.
  - Multi user operation difficult.
- Automated Publisher Mk 2. (Soon ready for testing at DESY.)
  - DB based metadata store.
  - Plug able front end for gsidcap/GridFTP based uploads.
  - Decoupling image upload from image list upload.
  - Support for multiple users.

How to subscribe to an image from an image list.

- Add endorser to subscriber DB.
  - Add publisher and issuer Subjects and give endorser a name/identifier.
    - Changed due to feedback from Mischa Sallé from NIKEF. (thankyou)
- Add subscription URI to DB.
  - Run subscription application with URI as command line option.
- Select image to subscribe to from Image list.
  - Select image by UUID.
    - Image meta data can be displayed by UUID also.
- Update imagelist cache.
  - Can be automated by cron.d
- Update image cache.
  - Can be automated by cron.d

Image list subscriber Database.

- Database Structure.
  - Seems to now be stable.
  - Supports multiple endorsers.
    - With multiple DN's
  - Enforces UUID constraints.
    - Imagelists and Images UUID.
  - JSON tags not constrained by DB.
- Currently using SQLAlchamy
  - Fast to develop and change.
  - ORM models have limits.
    - Updating operations.
    - Stored procedures
    - Triggors.
  - Good enough solution.



HELMHOLTZ ASSOCIATION

DESY

Changes to subscriber in the last 6 months.

- Core Functionality has not changed. (RDBMS unchanged.)
- Event interface. (See next slide)
- Command line was getting excessively long for simple operations.
  - Environment variable equivalents to command line options.
    - All options for cron commands.
    - Most interactive command line options have equivalent
  - Command line always overrides environment variables.
- Documentation.
  - Particular focus on README accompanying code.
    - User requests and support has led to many README changes.
- Usability Testing.
  - Command line options changed slightly to avoid bugs.
- Error handling improvements.

Subscriber application Events.

- To save users to parse logfiles or database.
  - Originally made for Stephan Detrick integrating DESY cloud.
- Image caching Events implemented.
  - AvailablePrefix, AvailablePostfix, ExpirePrefix, ExpirePostfix
  - Context set in environment variables.
  - Launch a custom application which has 10 seconds to execute.
    - Prevent client blocking updates.
- Requested new events (requested from customers).
  - NewImage
    - When an image is added to an image list that was not present on subscription.
- Considered new event (Do people need/want this sort of event?).
  - NewEndorser
    - When an image-list is signed by a new endorser

Summary

- Subscriber DB has been unchanged.
  - Not even with additional optional fields.
- Book now moved with code to github.
- Most changes have been in usability/integration.
  - Events, documentation, bug fixes.
- Event interface seems to be very important to usability.
- Publishing image lists is laborious
  - Large file sizes. (time to checksum, time to transfer)
  - Passwords require interaction.