HEPIX VWG Image transfer.

Owen Synge for the HEPIX virtulisation working group

Owen Synge
HEPIX VWG Image transfer.
HEPIX spring 2011

HELMHOLTZ ASSOCIATION

DESY

HEPIX VWG Assumptions

- For new customers Cloud may be all we need.
  - But HEP is not a new customer.
- HEP Experiment software is currently partially trusted.
  - Sites allow NFS 3, sites are not ready for untrusted images.
  - HEP experiments are not ready to abandon rshell data access.
- Virtualising Worker node can be transparent for grid users.
  - We need to trust images more than with a cloud infrastructure.
- Accountancy.
  - Cloud model of billing does not fit with current systems.
- We should find a way to use as much of grid as possible.
  - We should demonstrate our ideas work.

Image transfer Objective

- How to transfer images securely.
  - We know who made the image (Endorser)
  - We know the image is unmodified after endorsement.
  - We know the endorser cant repudiate their image list.
- Privileged images on sites must be authorized by administrator.
  - Can subscribe to an image from an image list.
  - Have minimal work for a site admin.
- Site must be able to revoke.
  - An image, an endorser or an image list subscription.
- We have Implementations for image transfer.
  - Present to HEPIX and hope for site approval
  - Plan to present to experimental communities if approved by HEPIX.

Publish Subscribe

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

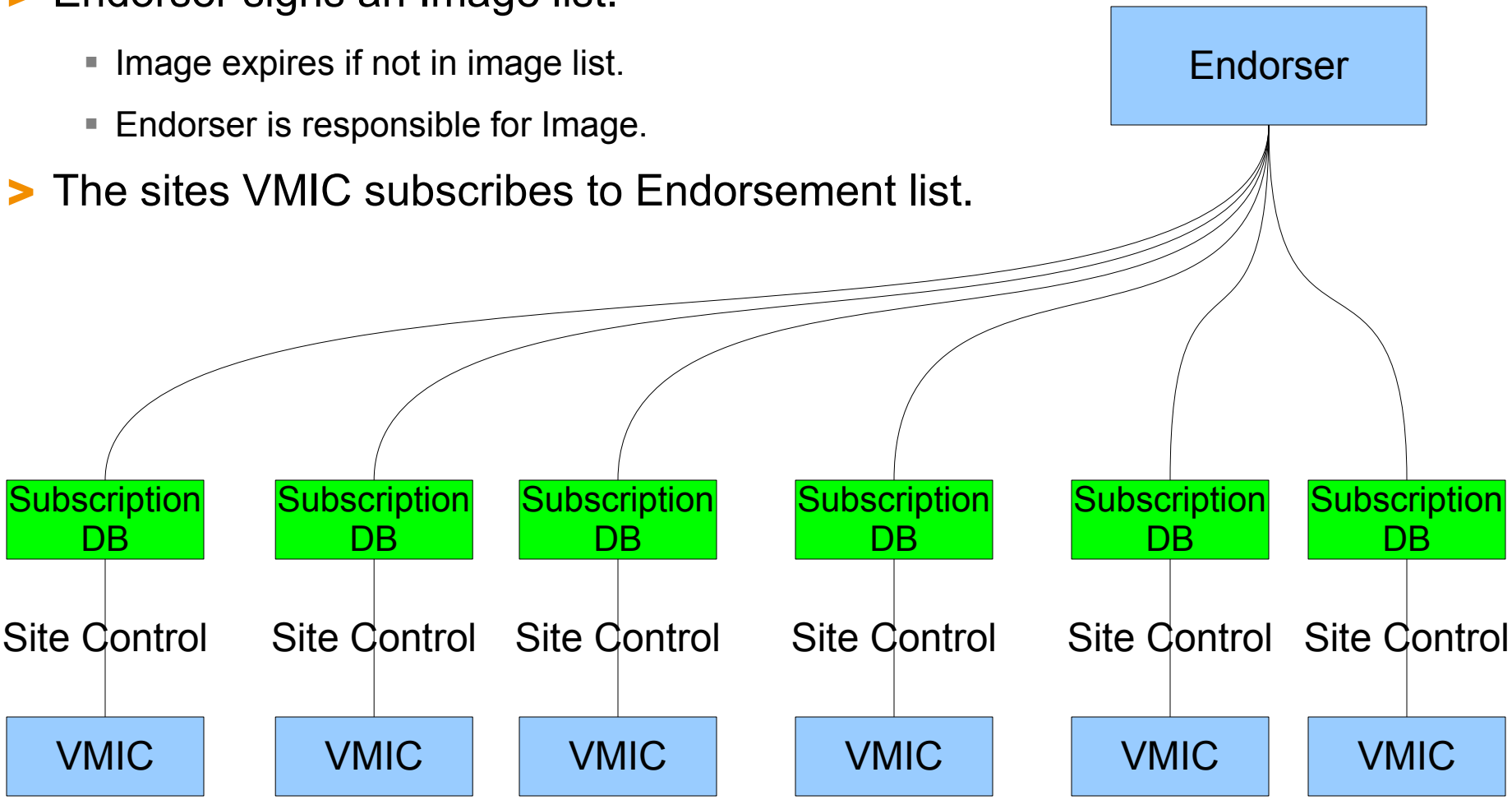


Image security.

- 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.

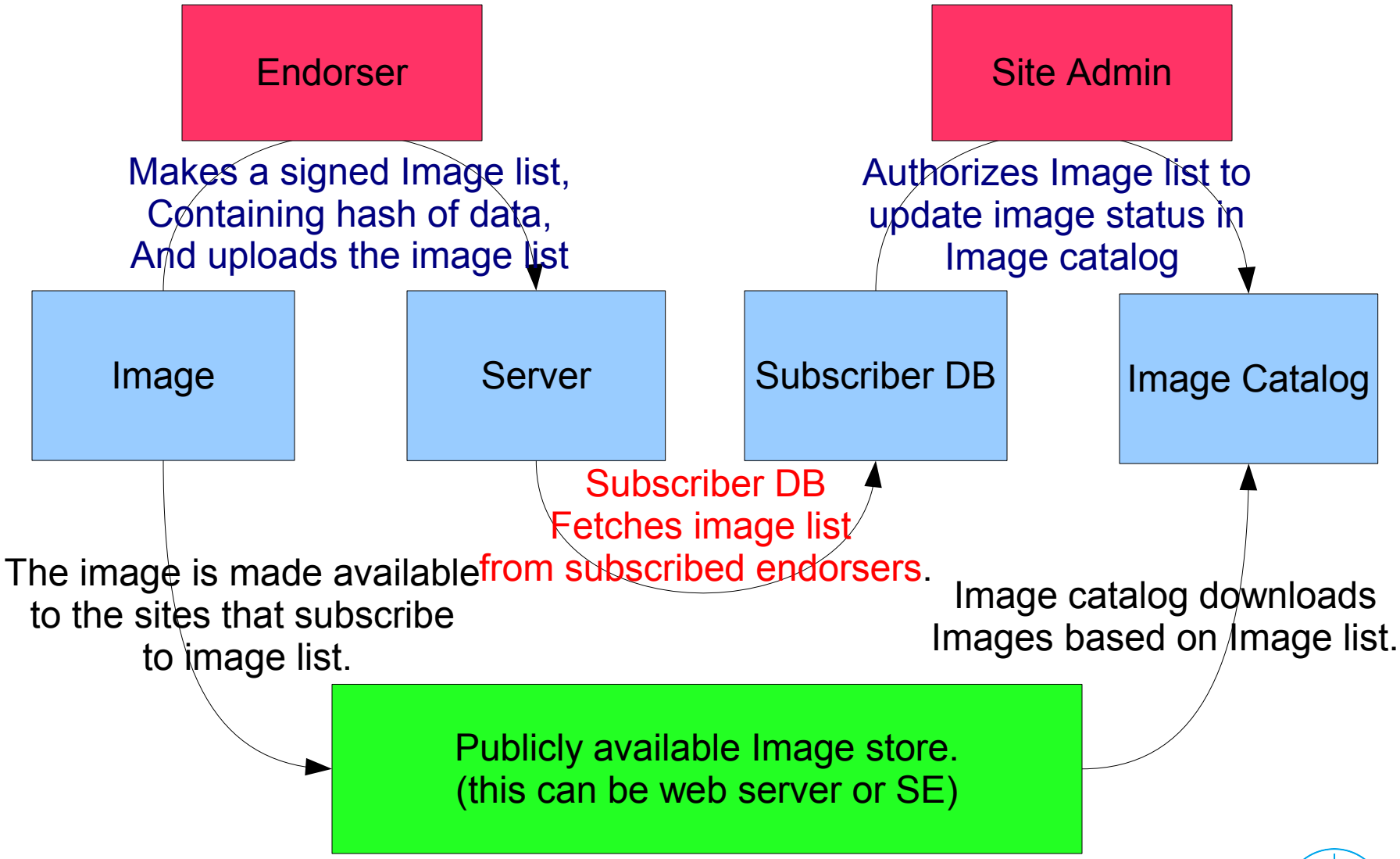
Meta-data Security.

- 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.
    - 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.

Why an Image list?

- We believe that image lists are better than just image meta data.
  - Any Image not on the list is not endorsed.
    - Prevents lost endorsements.
  - Endorsers only have one item to manage.
  - Any later image list published overrides the old image list.
    - Provides a simple way to deprecate images.

Image and image list transfer overview



Making the Meta-data

- CERN VMIC automates this with new patch.
- Process for signing Meta-data.
  - 1) Create a template for the image list.
    - vmilisttool --json image_list_template.json
  - 2) Create a template for an image reference.
    - vmilisttool --image /home/jdoe/rawdiskimage.img --generate Vmmetadata.json
  - 3) Add your newly updated image meta-data to the image list
    - vmilisttool --template image_list_template.json -add VMmetadata.json --json merged_image_list.json
  - 4) Sign the now assembled meta-data list.
    - vmilisttool --template merged_image_list.json -s signed_image_list
- Currently JSON, but XML will also be used.
  - Compatibility with new Clemson VMIC messages.
- Can edit the file easily before signing.
  - After signing the edits will make the list invalid.
- Extra fields can be added.
  - These are for endorsers customers use and will have no effect on the HEPIX infrastructure.

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+JLMbB9hZ6e2XgdjXWgLqz6vBSlf6KXi4KhCxe
a4FylvIk7OtY+bg0mg5IFHib6uP7fXhFKdBEapoi+B05wpluBMA+2DBdSt+rjzA8
SwiHUuan60VIyJAxammyOe3IKSpwyBXkQ10XjIhIpoSavqYXJboFOVzUcqxawdbX
Con2W8QfiwFKYupohG/VTusDXFT2MP4k+KxG3/rTTPWUDJme7VUPv3+CTcEO+z4v
X8/XhI44oAXIAgMBAAGjggInMIICIzAMBgNVHRMBAf8EAjAAMA4GA1UdDwEB/wQE
AwIE8DAdBgNVHQ4EFgQUGakUy66kgvuINBIf18WBXjGolqYwXgYDVR0jBFcwVYAU
xnXJKKzRC/w8/7m1HtNfO4BiEjShOqQ4MDYxCzAJBgNVBAYTAkRFMRMwEQYDVQQK
EwpHZXJtYW5HcmlkMRIwEAYDVQQDEwlHcmlkS2EtQ0GCAQAwHQYDVR0RBBYwFIES
b3dlbi5TeW5nZUBkZXN5LmRlMB8GA1UdEgQYMBaBFGdyaWRrYS1jYUBpd3luZnpr
LmRlMDUGA1UdHwQuMCwwKqAooCaGJGh0dHA6Ly9ncmlkLmZ6ay5kZS9jYS9ncmlk
a2EtY3JsLmRlcjAaBgNVHSAEEzARMA8GDSsGAQQBIDarLAEBAQUwEQYJYIZIAYb4
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+YaXQ1fHfmEmOrIf0ImP20TyTKloQWqzbq
WXwlRhZBUoD9zfiEM/jFvOvkuxLkQeiEcSzlLAGHXsHJ3anPMX9sobJFbJI0wYdN
sUOInHRhksokh2ow68KZK4vXLI173v5yZE7FZZ1Gl9T+YpkmOlW4iQ==

------EAE3006C97F670EE450F46AC8DF4C070--

Please Note:

XML implementation would look different but will be functional identical.

XML signatures store signature as XML.

Publishing endorsers Image list.

- To publish endorsers image list.
  - Must be available to the subscribers.
  - Subscription URL in image list must match your publishing location.
  - Must accept UUID constraints.
    - Image list UUID is unique
    - Each Image UUID is unique to your list.
  - Man in middle attacks must be blocked.
    - Suggest x509 based web server.
    - Could use ordinary https web server.
- To publish endorsers image
  - Must be available to subscribers.
  - All data integrity and authenticity in the image list.
- To expire images.
  - Endorsers do not reference image in the image lists latest version.
- Suggest endorser sets up a subscriber to endorsers own image list.
  - So endorser knows before subscribers that they have an issue!

DESY

Meta data subscription validation.

- Must validate the image lists.
  - Using x509 Signatures. (handling CA, CRL's, and CA namespaces)
    - SMIME is supported XML signatures intended.
  - Manage a list of endorsers for an image list.
    - So that more than one person can provide and image list (eg for Atlas.)
    - So that only authorized people can update an image list.
  - Must enforce UUID constraints.
    - UUID is same as other subscriptions
    - UUID of each image is exclusive to subscription.
  - Must query for signed image list using the image hash.
    - So you can find the endorser for a given image and their signature
      - Non repudiation feature from image
    - So you can expire images from an image cache.
  - Should inform image producer if an image list breaks subscriptions constraints.
    - Unsure how this should be done.

Meta-data subscription DB

- Mostly no admin interaction!
  - All subscriptions updated from a cron script.
  - All data is derived from subscriptions to image lists.
    - So just need to store signed image lists which you should anyway.
  - Migration is simply install a second in parallel.
- Simple RDBMS
  - No critical data to back up.
- Adding an image list all that is required to subscribe.
  - Since Image list contains where to get update to image list.
- Image cache as a client of the subscription data base.
  - Not yet implemented. (2011-04-29)
    - Will be writing this during HEPIX and shortly after.
  - Very simple directory containing image's.
  - Expired images will be deleted.
  - Current images will be validated.

VMIC deployed at CERN and 3 other sites.

- The admins image authorization interface.
- Allows multiple admins to cover for each other.
  - Uses service cert.
- Allows sites to securely deploy images around a site.
  - Produces a site image list. (similar to an external image list.)
  - Action to commit database data to a signed message.
    - Allows site to atomically update image lists.
- Provides a history of images deployed at a site.
  - Stores the sites signed image list.
- Patch came in from Clemson University to provide Image list import and export directly.
  - This talk does not cover this system as I am yet to investigate this new functionality.

Summary (Concepts matter not implementation)

- Signed image lists define images.
  - First version of meta data is defined.
  - RDF(XML) and JSON are functionally equivalent.
- Non repudiation of image lists through signatures.
- Only Images on current Image list are endorsed.
  - This means images expire when not in current image list.
- Principle is generic to Clouds, virtualised worker node.
- Two implementations of message generation/consumption exist.
  - My fault for not being aware of Clemson University patch.
  - We need to compare features and maybe make messages inoperable.
- We recommend the concept of Signed image lists to HEPIX.

DESY