Achilles' heels in LTDP or how do we address risks

G Ganis, J Blomer

DPHEP @ WLCG/HSF workshop
Naples, 27 March 2018

Preface

- Investigative talk on a methodology developed for other fields
- Triggered by a problem experienced by CMS Opendata

The CMS Opendata issue

CMS Opendata Approach

- VM: SL5 compatible
  - From /cvmfs/cernvm-slc5.cern.ch
  - Full analysis environment: compile + run
- Software: frozen CMSSW version (2010)
  - From /cvmfs/cms.cern.ch
- Data: frozen set
  - Access via XRootD, DPHEP portal, EOS
  - Conditions data on /cvmfs/cms-opendata-conddb.cern.ch
- Deployed as OVF/OVA bundle
  - Easy auto-installation in VirtualBox

The recent CMS issue (1)

The http clients stopped working on the SL5 VMs; e.g.

```
wget –O foo2.root https://eospublichttp.cern.ch/eos/opendata/cms/…
...
Unable to establish SSL connection
```

- Reason: eospublichttp.cern.ch server requires TLS v1.2
  - The RHEL 5 based operating systems can only speak the older versions SSL v2 or v3 or TLS v1.0

- This problem was solved using the XRootD EOS door:

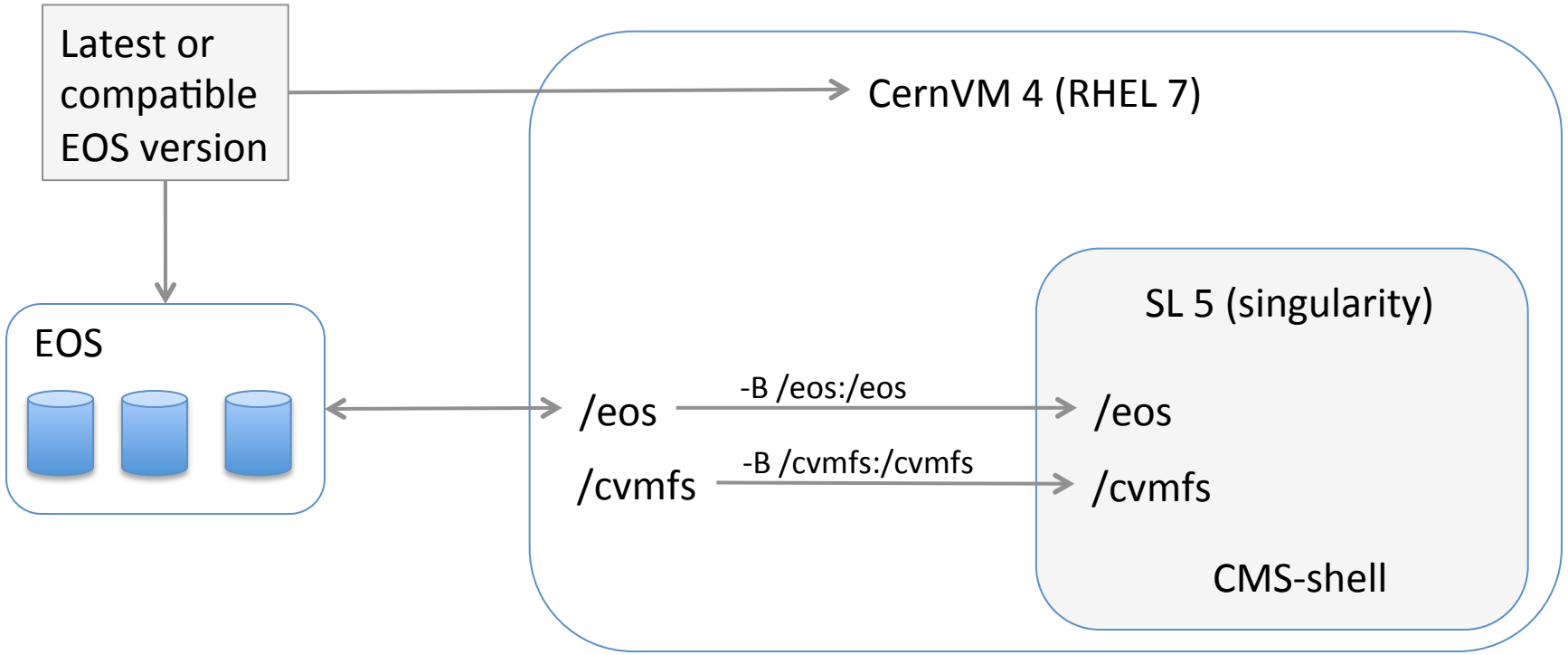
```
xrdcp root://eospublic.cern.ch/eos/opendata/cms/… foo2.root
```

(1) Summary of a mail exchange between K Lassila-Perini, T Simko, A Geiser, J Blomer, G Ganis

Is Xroot client more resilient?

- The short answer is no:
  - The problem comes from version divergence
- Nothing can protect against that
  - At a certain point the EOS server will require an authentication protocol or a version that the (old) client will not understand

Alternative solution: removing version incompatibility



Lesson

- Access protocols evolve and there is no guarantee of backward compatibility
- Having alternatives mitigates the problem
  - Another protocol
- Exploiting system features transfers the problem
  - Mount points

Lesson, revisited

- Backward incompatibility is a *risk* for LTDP
- We have to do something to *mitigate* and/or *transfer* the risk
- I.e, some sort of *risk analysis*
  - *Risk assessment and management*

Risk Management and LTDP

Risk Management

- Methodology described by ISO 31000
  - Guidelines on principles and implementation of risk management
- Risk: the effect of uncertainty on objectives
  - Mathematically: Likelihood x Impact

- Typically applied in other fields: finance, enterprise, medical devices, (mega-)projects, …

Risk Management in IT

- Security
  - Incident handling / mitigation action plan
    - Preparation, Identification, Containment, Eradication, Recovery, Lessons Learned
- Software development process
  - Delivering software incrementally mitigates the risk of late finding of problems and anticipate action

How RM would translate in LTDP

- Objective
  - Enable long term sustainable use of collected data

- Risk
  - To lose bits (i.e. the storage medium)
    - Addressed with bit preservation, ISO 16363
  - To lose knowledge (i.e. the SW env, Docs, …)
    - Addressed with migration, emulation, doc portals

Risk Management Principles (excerpts)

- Should create a value
  - Resources expended to mitigate the risk should be less than the consequence of inaction
  - LTDP: the value is at most the cost of recreating the lost data
- Be integral part of the organizational process
  - Should not be a stand-alone activity or be separate from the main activities and processes
- Be part of the decision making process
- Be a systematic and structured process

Risk Management Principles (excerpts) (2)

- Explicitly address uncertainty and assumptions

- Be dynamic, iterative and responsive to change
- Be capable of continual improvement / enhancement
- Be continually or periodically re-assessed

- Take human factor into account

Risk Treatment Categories

- Avoidance
  - Do not adopt something that carries a risk of loss
    - Ideal, but not always possible
- Reduction or Mitigation
  - Reduce the impact of a risky event
    - E.g. outsourcing, diversifying
- Sharing or Transfer
  - Share with another party the burden of loss
    - E.g. contracting insurances
- Retention
  - Accept the loss when it occurs
    - E.g. in case of (rare) catastrophic events

Risk Treatment in LTDP

- Bit preservation: reduction
  - Backups and regular copies on new storage
- Migration: reduction, retention
  - Port code to new OS, run quality tests
    - Trying, eventually, to fix possible issues
- Emulation: transfer, reduction/retention
  - Host responsible for emulation quality
    - Mitigate or accept residual issues

Best practices (for today's exps)?

- Transfer would be the ideal solution
  - Relying on components which are mainstream for everybody
- Not always possible
  - We need components not available mainstream
- Focus on mitigating or reducing the impact of this
  - Ideally promote our solutions mainstream
  - Or *consolidate* system protocols to decrease the likelihood that something breaks
    - Keeping multiple options decreases the impact if something in the preserved software stack breaks
- Make it part of the experiment ecosystem during the experiment lifetime

Recap

- LTDP is about reducing the risk of losing the data
- LT sustainibility requires risk transfer or sustainable mitigation
- Techniques of Risk Management may help in
  - Assessing, prioritizing, treating
- Perhaps a useful framework to rationalize the problem