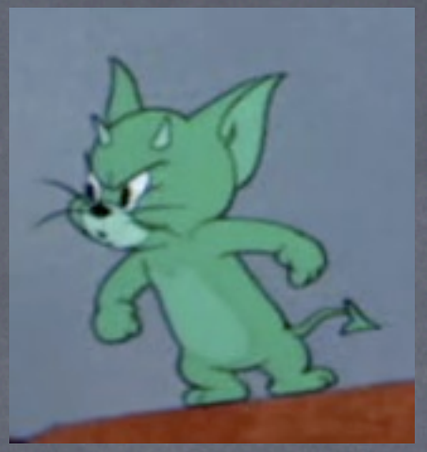
Storage Control Protocols: Introduction à la Graeme

Graeme Stewart

University of Glasgow | Experimental Particle Physics

My Role Here



- To be...
  - A disinterested observer?
  - A referee?
  - An arbiter?
  - Your fairy godmother?
  - A dumb user
    - (I will try to be as stupid as possible)

A User's Perspective on Storage

- It must be easy because all I want to do is...
  - Put
  - Get
  - List
  - Delete
- Is this hard?
- A user's view of storage is conceptually simple

But of course I also want...

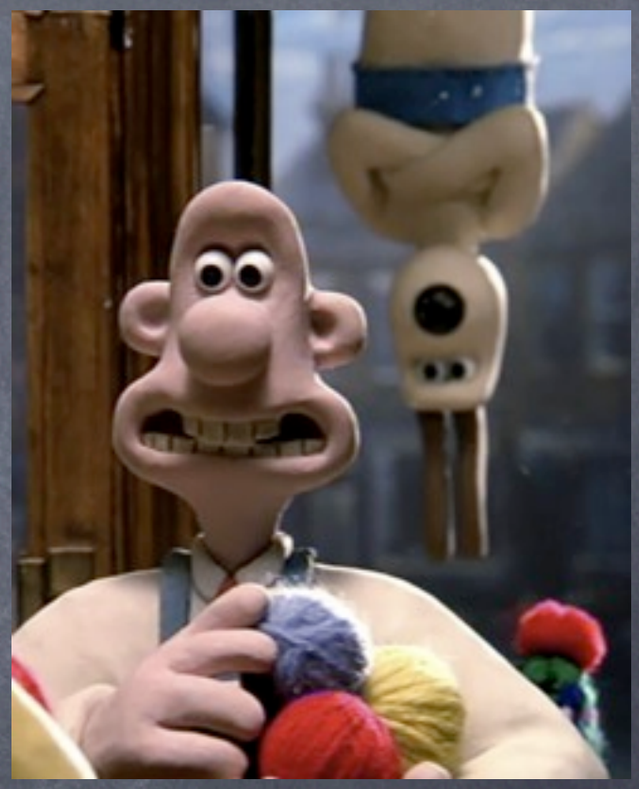
- Fast, reliable and scaleable too.
  - And it may help if you are clairvoyant...
- Mapping up a user's storage requirements into the language the storage understands can be hard - this is the role of the storage interface
  - But this is a very multilayered problem as there are so may interface levels
    - Native SE, SRM, Lcg-Utils, VO Tools

Layers on layers...

- When users are given a interface to storage it puts a layer between the users' intentions and the storage system (which is a true of SRM as gFAL)
  - This can be useful to simplify life for users
  - But it can make for trouble when users are quizzed by storage providers - no one is speaking the same language
- (in other words, users are from Mars and storage is from Venus)
- ATLAS can tell you how many datasets are subscribed and how many files moved to a site in a day, they cannot tell you how many srmLs calls were used to do it, because FTS and gFAL control that

Details, details, ...

- The details of these interactions curse our lives:
  - If you expose technical details to users there is probably a defect in the underlying system
  - If a user tells you they want to be exposed to the technical details there is probably a defect in the underlying system
  - If a user absolutely insists on knowing the technical details then there is probably a defect in the underlying user



Power Users

- For the LHC VOs there were some extra considerations
  - Put simply, these were:
    - Is the data on tape?
    - Is the data on disk?
    - Is it on both?
- This led to a very abstracted implementation in SRM of spaces, most of the functionality of which seems, to my possibly naïve view, to be unnecessary
  - It's certainly not being used in the way that it was conceived and that indicates it does not match user requirements well

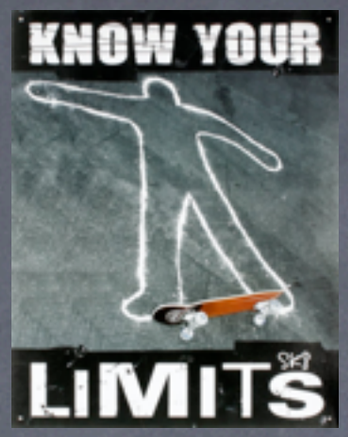
The road to storage hell is paved with ... implementation details

- Non-uniform implementations of interfaces cause real problems for users:

  if castor () elif dcache () elif ...

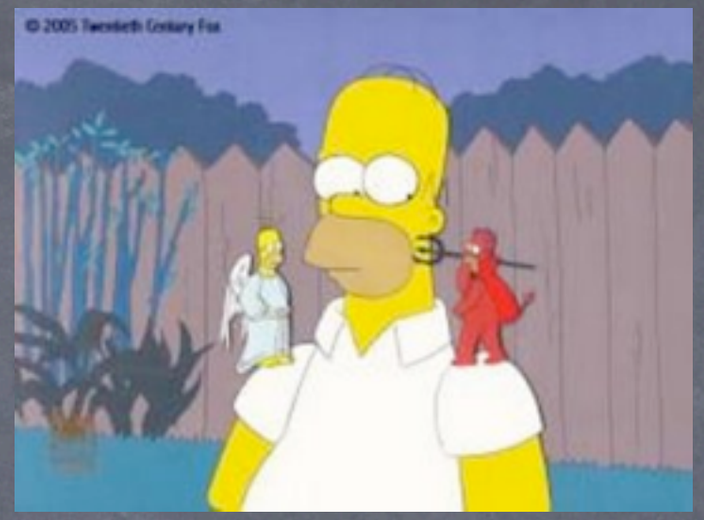
- Suffer from
  - Reduced performance of mediating layer
  - But still having to care about the underlying system
- Abstraction without generalisation
  - In this case the interface causes more trouble than it's worth

Know your limits!



- Storage systems have to be able to withstand load
- Their interfaces should be able to absorb excess load and dissipate it
  - At the moment our systems tip over into crisis too easily
- This is clearly something which needs to be addressed very urgently
- But likewise we cannot suffer plodding storage in the LHC era
  - So performance needs to be better than it is today

Some things to remember...



- What is urgently needed today is stability and scalability with the SRM we have now
  - Things cannot be significantly changed in advance of LHC data taking
- Users want an interface which is conceptually simple and genuinely uniform
  - Technically you can be brilliant, but the user should never know
- If this cannot be achieved then many users would probably prefer to talk directly to the backend system
  - However, if key elements in storage interaction can be identified and simple interfaces built to provide this service to users then there is real value to gain from this