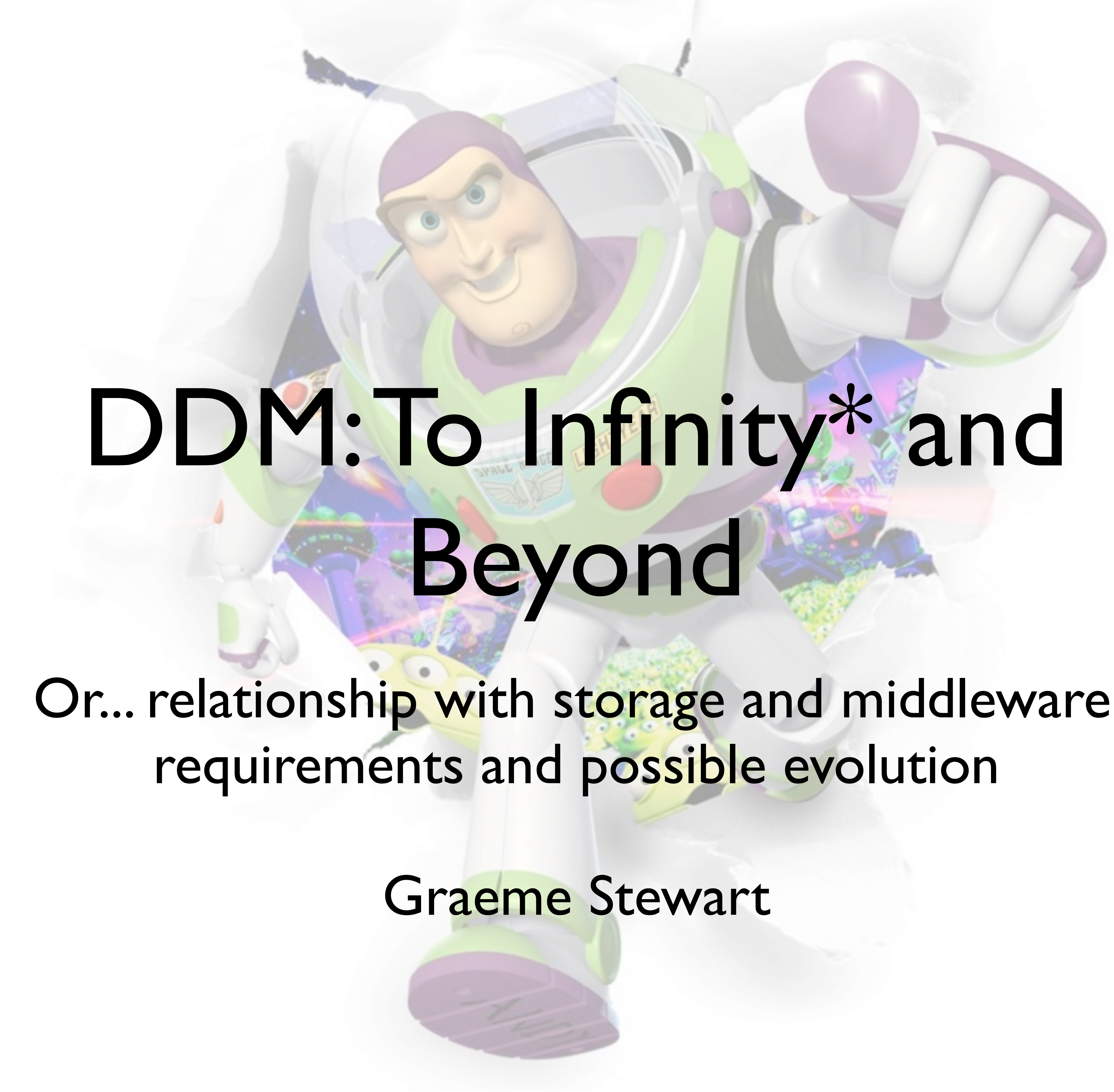

DDM: To Infinity* and Beyond

Or... relationship with storage and middleware, requirements and possible evolution

Graeme Stewart

University of Glasgow | Department of Physics & Astronomy

*Here Infinity = 2013

Global and Local

- DDM is global
    - DDM does not store data, but organises data - metadata layer on top of local storage
        - With fragmented local storage we always need this (right!?)
- Site storage is a local abstraction layer, hides many internal details
    - Functionality
        - More features help us (ACLs, quotas, partitioning are useful)
            - But do these really need to be global features (quotas, partitioning)
            - Do they close the door on interesting non-HEP alternatives?
                - HDFS, Cloud Storage,
            - e.g., GUID lookup (local LFC)
                - This is the 'primary key' for DDM and would mean that the storage namespace acts as its local catalog
                    - Attractive but outlandish?

Interfaces

- Site Storage (continued)
    - WAN Interface layer
        - Today mainly SRM: WLCG specific, complex, fragile
        - Better alternatives:
            - gridftp, http, S3 protocol
            - Web services: S3 and cloud storage systems
        - No big bang - evolve, but need support from middleware
    - Why FTS at all?
        - We do need a *transfer service*
        - Can storage protect itself from meltdown?
    - LAN interface layer
        - SRM too slow for LAN access: use xrootd, ~~rfio~~, dcap, etc.
            - Mainly implemented by root
        - Standard is very clear: **file://**
            - FUSE provides transition - stable, scalable?
        - Data access with WAN fallback - grid is my backup
            - xrootd (need checksums)

Cache vs. Archive

- Now managing T2 disk much more like a cache
    - Technology is archive based: catalogs, dataset registrations
    - Caches manage themselves much more dynamically
        - LRU deletion
        - Bloom filter publication of contents (used by squid)
    - Attractive to use this model instead
        - ARC Demonstrator
- But do we also need archive at T2s?
    - User outputs
    - Larger T2s take load off T1s
        - MC repositories, Backup primary copies
    - What model makes T2 storage reliable?
        - Distributed failover systems (T2s as JBOS: Just a Bunch Of Sites)
            - Extend xrootd to smaller T2s?
            - ttree reads with cache minimise performance hit - test me now
        - What about outputs?

LFC and DDM

- LFC maps GUIDS/LFNs→SURLs
- LFC consolidation happening this year
  - Beyond that?
  - DDM CC could manage GUID→SURL mappings internally
    - Simplification for many clients
    - Helps enforce consistency
  - Probably cannot be planned now, but this seems the right way to go
    - Evaluation as LFC consolidation proceeds