



UNIVERSITY OF
OXFORD

Persistent IT Architectures

Building Digital Archives That Last

Neil Jefferies

R&D Project Manager

Systems & eResearch Services (SERS)

Oxford University Library Services (OULS)

A Digital Asset Management System (DAMS)?

- Bookshelves for the 21st Century
- Legal Deposit
- Voluntary Deposit
- Digitisation
- Extended Remit - Research Data
- What Next?

The Library Experience

- People change
 - Academic/library sectors have relatively low turnover but trending upwards
- Processes/use patterns change
 - The way content is used changes
- Buildings/institutions change
- Only the content persists over the long term

Paradigm 1: User Focus

- “The coolest thing to do with your data will be thought of by someone else”
- Most of our users aren't born yet
- Most of those alive we don't know
- Many constituencies
 - Academics, Librarians
 - The University
 - The Nation...

Paradigm 2: Project Focus

- A system is developed, put into production and ultimately replaced - really?
- There is NO exit for a library
- Big bang implementations always experience a degree of failure
- Web-scale interoperability requirements
- In practice, development doesn't stop
 - Recognise this and deal with it

Paradigm 3: Modelling/Specification

- We don't know what we will be storing
- We don't know how it will need to be described
- We don't know what users are going to want to do with it
- “The coolest thing...”
 - The biggest risk is that premature proscription prevents the coolest thing...”

Paradigm 4: Stable Platforms

- Technology changes rapidly
 - Academics tend to take advantage of developments sooner
- Standards emerge and develop
 - Interoperability outside our domain
- Software/suppliers change
 - Long term dependencies ultimately cost

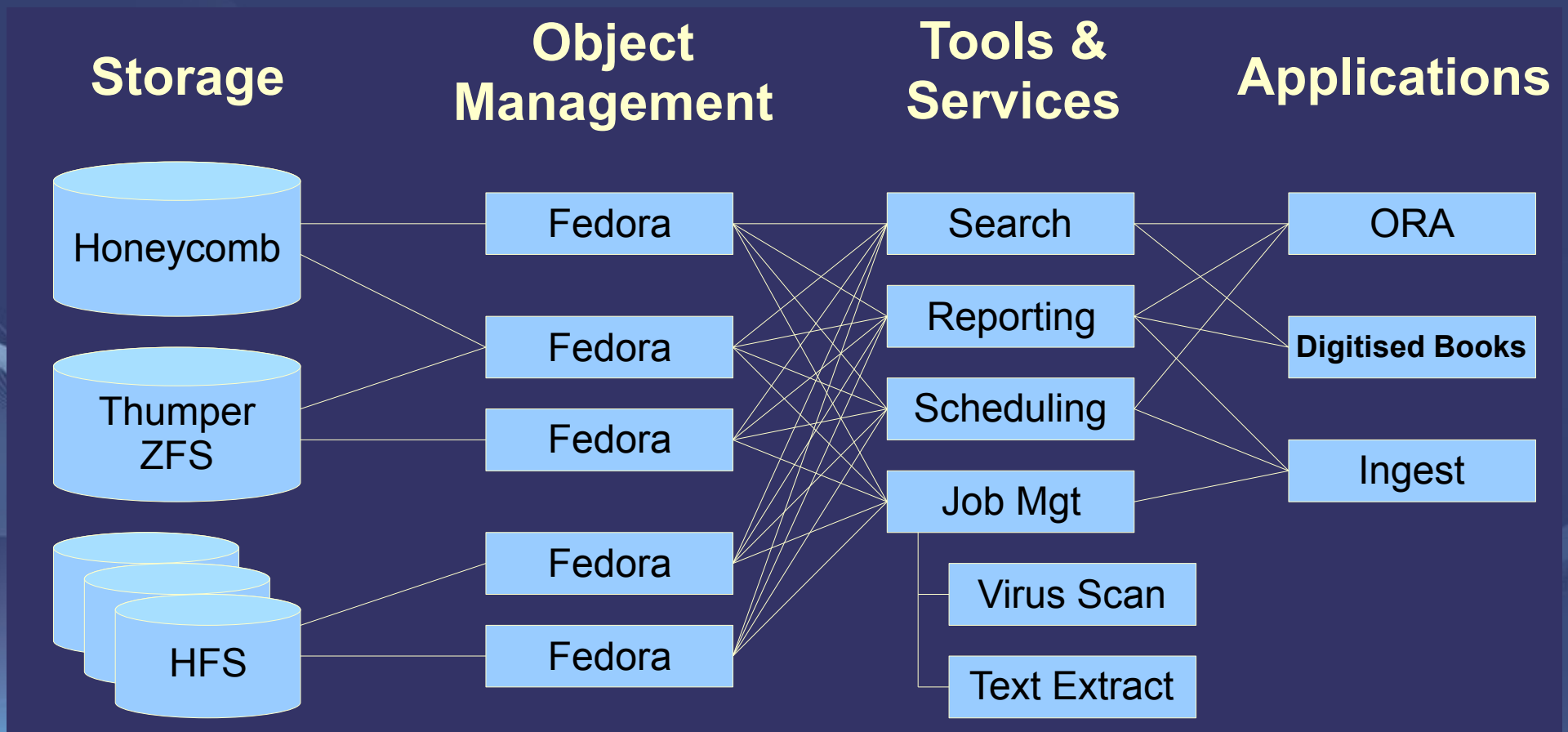
Is this a problem?

- Only if the system is considered as a whole
- Separation of concerns
 - Decompose system into *independent* components that *are* tractable
- The “system” is defined by a set of principles/guidelines for how components should be incorporated

Separation of Concerns

- Bitstream Preservation – handled by the Storage Layer
- Location and Retrieval – handled by the Object Management Layer
- Discovery and Dissemination – handled by the Tools and Services Layer
 - Also Active Preservation and Access Control
- Presentation and Workflow – Application Layer

Separation of Concerns



Key Principles 1

- Every component has a well-defined function
 - Not all capabilities may be used
- Every component is replaceable
 - Use abstract API's (ideally minimally complex)
- Above storage, every component is expendable
- Every component is scalable
 - Use load-balancing/resolvers/job management

Key Principles 2

- Avoid premature design decisions
- Support Heterogeneity
 - Multiple implementations of similar functionality co-exist
- Capability focus
 - Review decisions in terms of what they might prevent
- Avoid re-invention wherever possible

In Practice

- FEDORA
 - Does not force file or metadata format decision
 - Flexible arbitrary object model
 - We use a limited subset of functions
- Honeycomb/Akubra decisions
 - Fedora resolver means minimal impact
- EPrints ingest service

Remaining Questions

- How do we run the system operationally?
 - Since there is no formal transition from development
- Managing change is still a challenge
 - Although we can minimise its impact we still need control/documentation
- Non-technical challenge of sustainability



UNIVERSITY OF
OXFORD

Questions

Neil Jefferies

neil.jefferies@sers.ox.ac.uk

www.sers.ox.ac.uk

Oxford Research Archive

ora.ouls.ox.ac.uk

Developer's Blog

oxfordrepo.wordpress.com

Google Code

look for: `python fedora-commons`