Technology Matters: A personal view

Robert Sharpe
23rd January 2014
Agenda

• Quick introduction to Tessella
• What systems do I need for digital preservation?
• On-site & off-site systems
• How automatic can it be?:
  – Ingest example
• Scalability
• Conclusions
SDB: Digital Archiving Systems in 11 Countries across 4 Continents

Wellcome Library
UK National Archives

Corporate Archives
Financial Archives

Finnish National Archives
Estonian National Archives
Latvian National Archives

Gemeente Rotterdam
Rotterdam City Archive

National Archives of Hungary
Budapest City Archives

Dutch National Archives

Malaysian Archives

UK Parliament

Swiss Federal Archives

Austrian Archives

Science & Technology Facilities Council
FamilySearch

Tessella
Preservica: Digital Preservation as a Service
What systems do I need?

External (source) systems:
- RMS
- Web
- Email

Digital Preservation System:
- Transfer System
- Ingest System
- Preservation System
- Management
- Technical Registry
- Access System
- Storage System

Archival metadata:

Descriptive catalogue:
On-site Systems

- **Pros:**
  - Security concerns reduced
  - Have control
  - Easy (easier) to migrate to new system
  - Can customise (can considerably reduce manual overheads)
  - Bandwidth issues reduced

- **Cons:**
  - Need own hardware
  - Need own support
  - Costs more!

- **Conclusion:**
  - Good if have budget & bespoke needs
  - Good for large volumes
Off-site Systems (e.g., cloud)

- **Pros:**
  - Cheaper
  - Low / zero up-front cost:
    - Don’t buy hardware
  - Lower operational costs:
    - Shared support
  - Pay for what you need

- **Cons:**
  - Harder to customise
  - Possible security concerns
  - Bandwidth for large volumes
  - Harder to migrate

- **Conclusion:**
  - Good for low budgets
  - Limited ability to be bespoke
How automatic can it be?

- **Golden rule:**
  - Humans make judgements
  - Let software implement your judgements:
    - Will make less mistakes
    - Can be driven by machine-readable policy

- **Sometimes lack of trust:**
  - Good to test software
  - Once passed test, use it!
  - If issue occurs in production:
    - Fix it
    - Get your supplier to fix it
How automatic can it be? Ingest

- Capture human judgement as policy up front:
  - Decide what to keep?
  - Decide what to structure / catalogue?
  - Decide storage policy (how many copies to store)?
  - Decide which steps are necessary?
Real system example: Ingest

- In operation, let the software do its job:
Real system example: Ingest

- Pick workflow to start:

- In fact even this is often automated:
  - Watch for arrival of complete SIPs
Real system example: Ingest

- Watch (if you want to):
Real system example: Ingest

- Deal only with issues that the system can’t:
How automatic can it be? Ingest

- Example issue: metadata impedance:
  - Source metadata:
    - Info in ERMS, e-mail system
    - Very little (e.g., web crawling)
  - Traditionally:
    - Translate to archival schemas (EAD etc.)
- Could manually map metadata:
  - As part of manual cataloguing
- Can automate:
  - Set up transform
- OR can bypass:
  - Embed original metadata
  - Use technology to view/edit/index/search without transform
Scalability

- Ingest: Series sequential steps
- Tool like DROID (format identification) typical time:
  - Small files: ~20s per 1000 files
  - Large files: ~ 8s per GB (c. 10TB per day)
- Large volumes:
  - Throughout more important than individual run speed
  - Need ability to run in parallel (multiple threads)
  - Automation important
  - Resilience important
- It can be done:
  - SDB ingests FamilySearch ingests at 50TB every day
  - Note doesn’t need very expensive processing power:
    - 6 Application Servers @ c. $5k each = $30k
    - Ingest disk arrays and network  Higher
    - Storage costs  Dominant
Conclusions

- Try to minimise number of systems:
  - Will cost more in interfaces if you don’t
- Choose system:
  - On-site / Off-site
- Archivists / Librarians / Curators are in charge:
  - Do what you are good at
  - Buy software / services to do the rest
- Automate everything that you can:
  - Use software that already does this
- Scale by system engineering:
  - Don’t judge by speed of 1 thread on your desktop
- Lots of interesting issues to resolve:
  - But don’t reinvent the wheel!