BRITISH LIBRARY

The large-scale archival storage of digital objects

Digital Preservation Coalition Meeting

York Science Park

22nd April 2005

Speakers

The work described has been carried out as part of the British Library's Digital Object Management (DOM) Programme

- Richard Masters, DOM Programme Manager
- Sean Martin, Head of Architecture & Development
- Jim Linden, Head of Infrastructure Strategy & Development
- Roderic Parker, DOM Communications Officer

Overview of the British Library's needs and approach

DOM Programme Mission and Vision

Our mission is to enable the United Kingdom to preserve and use its digital intellectual property forever

Our vision is to create a management system for digital objects that will:

- store and preserve any type of digital material in perpetuity
- provide access to this material to users with appropriate permissions
- ensure that the material is easy to find
- ensure that users can view the material with contemporary applications
- ensure that users can, where possible, experience material with the original look-and-feel

DOM Programme Scope

We need a generic and cost-effective approach

- to take in material coming from many sources
- to take in material of any and all types
- to store it securely for the long term
- to allow controlled access
- to be enduring

DOM Programme Scope - life cycle of objects

DOM is concerned at present with the familiar processes of (in conventional library terms):

Collection

- Selection
- Acquisition
- Accession
- Description

Retention

- Storage
- Preservation

Access

- Resource discovery
- Delivery
- Rendering

A complete life cycle would also include

- creation
- deletion

DOM Programme Content Drivers

- Legal deposit legislation for non-print material: royal assent in October 2003 but still needs secondary legislation to bring it into force
- Existing voluntary deposit scheme operational since 2000
- Digitised versions of BL material from early ’90s onwards
- New digitisation initiatives: newspapers, sound, etc
- Electronic journals
- Sound Archive’s 15TB of material per year (with 50 year collection)
- Web archiving
- Cartography and datasets
- &c &c

DOM Programme Principles

- Our approach is to be incremental, not ‘Big Bang’
- Prototype so as to learn, understand, reduce risk and uncertainty, and demonstrate the basis of a good solution
- Use standard industry tools (e.g. Microsoft Message Queue and BizTalk)
- Aim for 3 releases per year
- A principal goal is to define an overall long term “logical architecture”
  - Within this, there will be successive generations of physical architectures
- We will use our knowledge of the storage marketplace to manage storage procurement
  - We are certain that we will need very large amounts of storage, but we are uncertain when – so we need flexible and scalable procurement

DOM System Design Principles

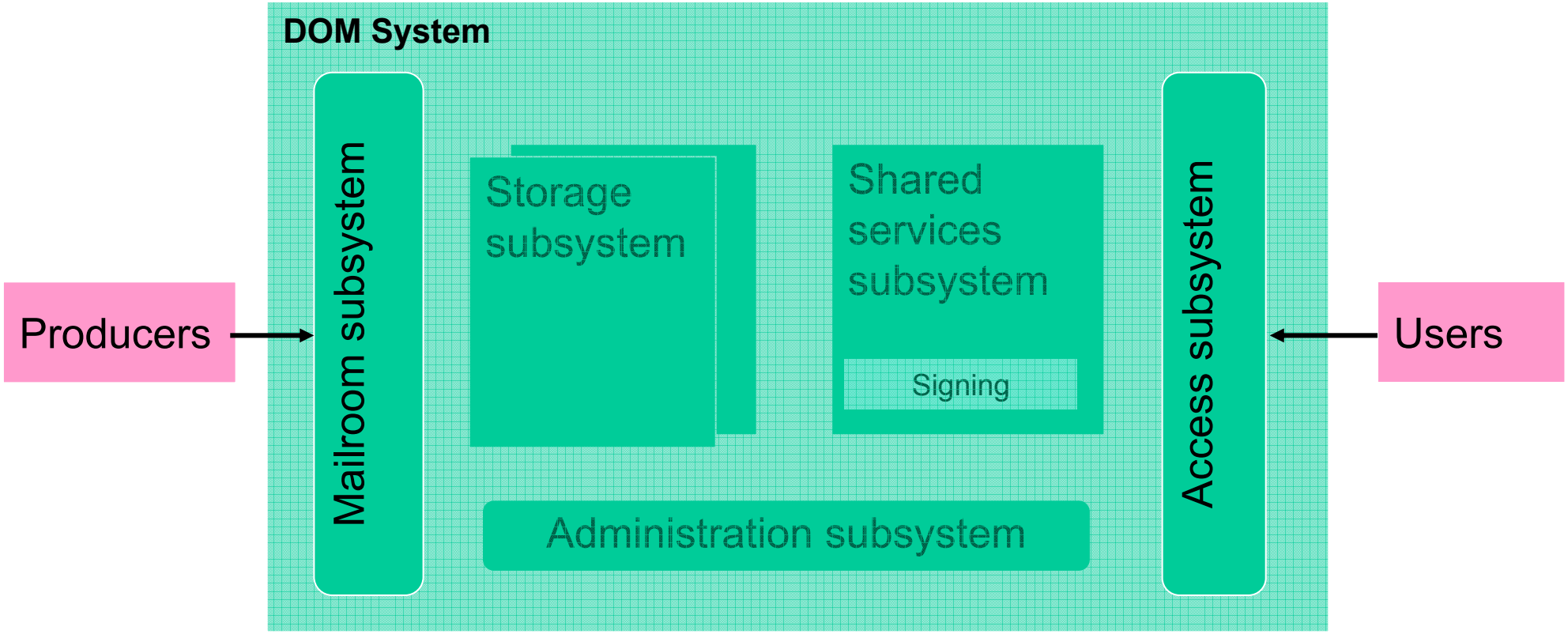
- Significant number of objects will be stored in perpetuity
- Objects can be considered to be invariant and some will be large
- Objects will typically be accessed infrequently

- Each object will have a unique persistent invariant identifier (DOMID)
- All systems external to the Store interact only using a DOMID

- The design of the system must be inherently scaleable in terms of capacity, the number of objects, and the ability to deliver objects
- Need inherent resilience so that object loss is extremely unlikely
- Short interruptions to, or degradation in, service can be tolerated, but extended loss of complete service cannot be tolerated

- Integrate off-the-shelf components
- Be cost conscious

DOM System


DOM System
Producers
Mailroom subsystem
Storage subsystem
Shared services subsystem
Signing
Administration subsystem
Access subsystem
Users

Integrity and authenticity of objects within DOM

- Integrity:
  - System has capability to monitor continuously the object store to detect object corruption
  - Based on using secure hash algorithm (SHA-1)
  - It would then initiate object recovery

- Authenticity:
  - Provide long-term assurance that an object that is re-presented is as it was when it was ingested
  - Based on the use of cryptographic signing techniques
  - Each object is signed when it is ingested
  - The signature is verified when required
  - The signing mechanism is “tightly” controlled

- Integrity and Authenticity can be determined locally within the architecture

Disaster tolerance rationale for a multi-site design

- One can obtain commercial disaster recovery (DR) solutions for common equipment configurations
- However one cannot obtain such solutions for systems comprising multi-100 Tb systems
- So we must build in the need for DR into the design of the system
- A single site solution, subject to a common-mode disaster, would suffer considerable loss of availability after a disaster, and so is not acceptable
- This implies that we need a multi-site solution
- Conventionally based on a master-standby where 50% of kit delivers service
- Our design is based on the use of multiple autonomous independent peer sites that cross-synchronise so 100% of the kit delivers normal service
- Service continuity: full service, albeit slower, is deliverable by only one site

DOM in the context of the storage market: resilience and performance

- The dominant segment of the market focuses on delivering high performance within a highly resilient single site

- However:
  - Many of our objects will be rarely accessed
    - So we do not want to pay for “maximised” performance we do not need
  - We have resilience by using multiple sites, hence we have a reduced need for resilience within a site
    - so we do not want to pay for “maximised” resilience we do not need
- These observations helped us in designing a cost-effective large scale resilient solution

DOM in the context of the storage market: procurement and rolling programmes

- A major cost is in physical storage
- The market for storage systems is changing rapidly, and this implies that supplier “lock-in” is not sensible
- We thus need flexibility to change supplier over time
- Cost of storage is reducing by 30-40% per year
- So we procure on rolling basis just ahead of demand
- We also will replace storage on a rolling basis on expiry of warranty

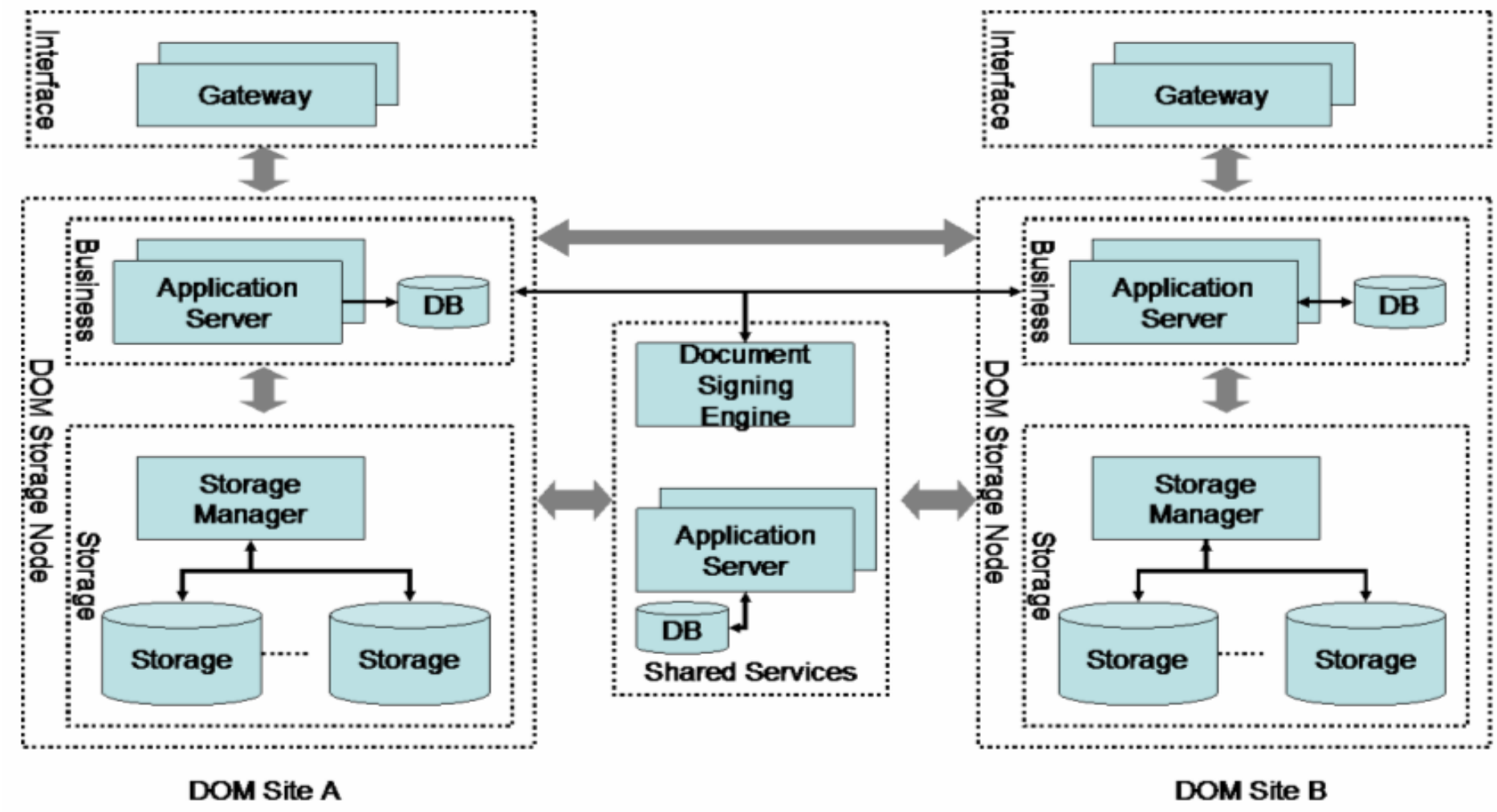
- The rolling procurement and replacement programmes imply the need to be able to support a heterogeneous hardware product solution
- The design of the logical architecture thus supports storage sourced from multiple storage vendors

DOM in the context of the storage market: conclusions

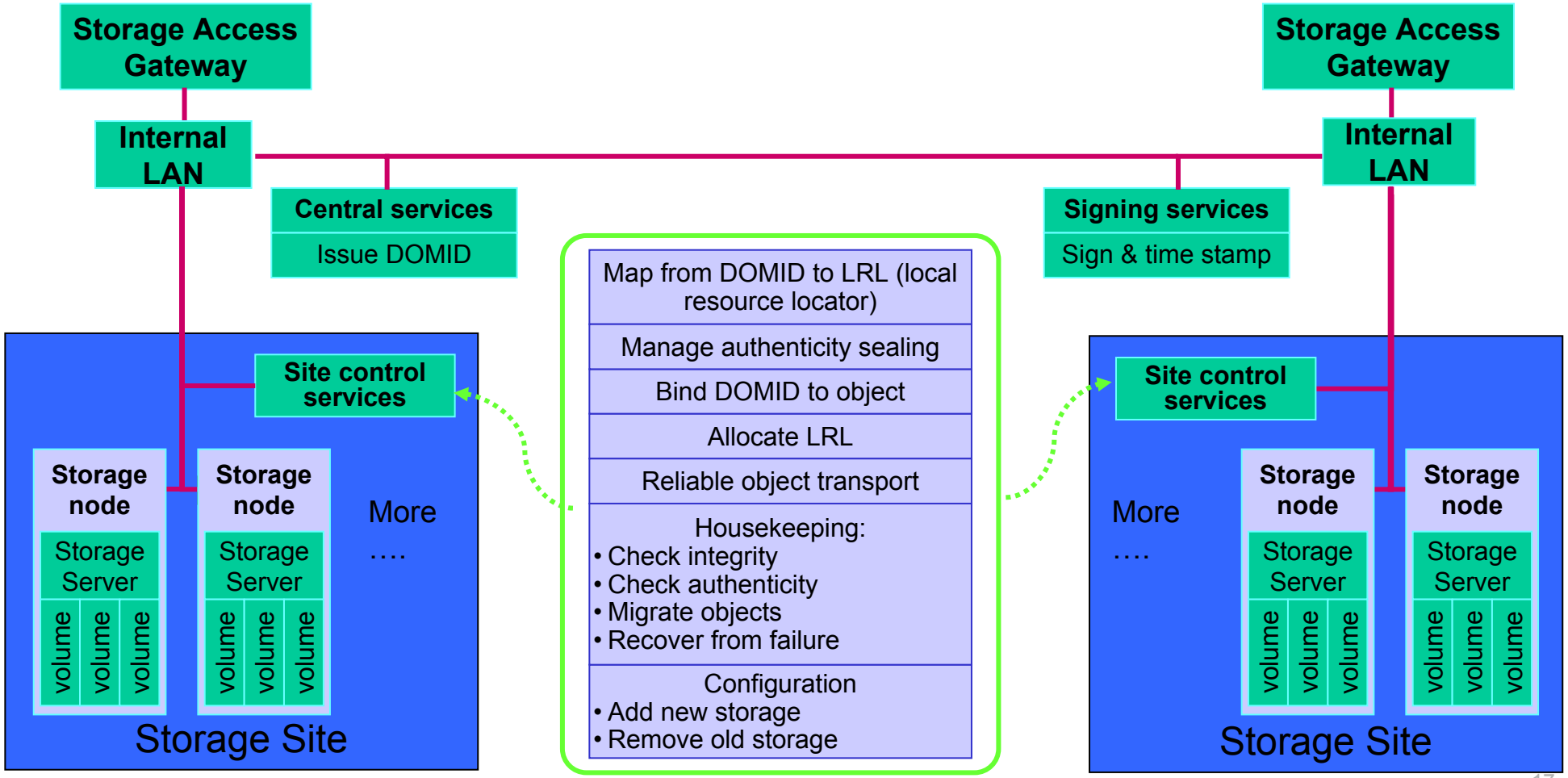
- We do not want to pay for “maximised” performance that we do not need
- We do not want to pay for “maximised” resilience within a site that we do not need
- We will procure as we need storage, and we do need not be tied to a single vendor

- These all imply that we can seek to obtain commodity storage hardware solutions from the marketplace, so in that sense:
  - We manage the market
  - It does not manage us

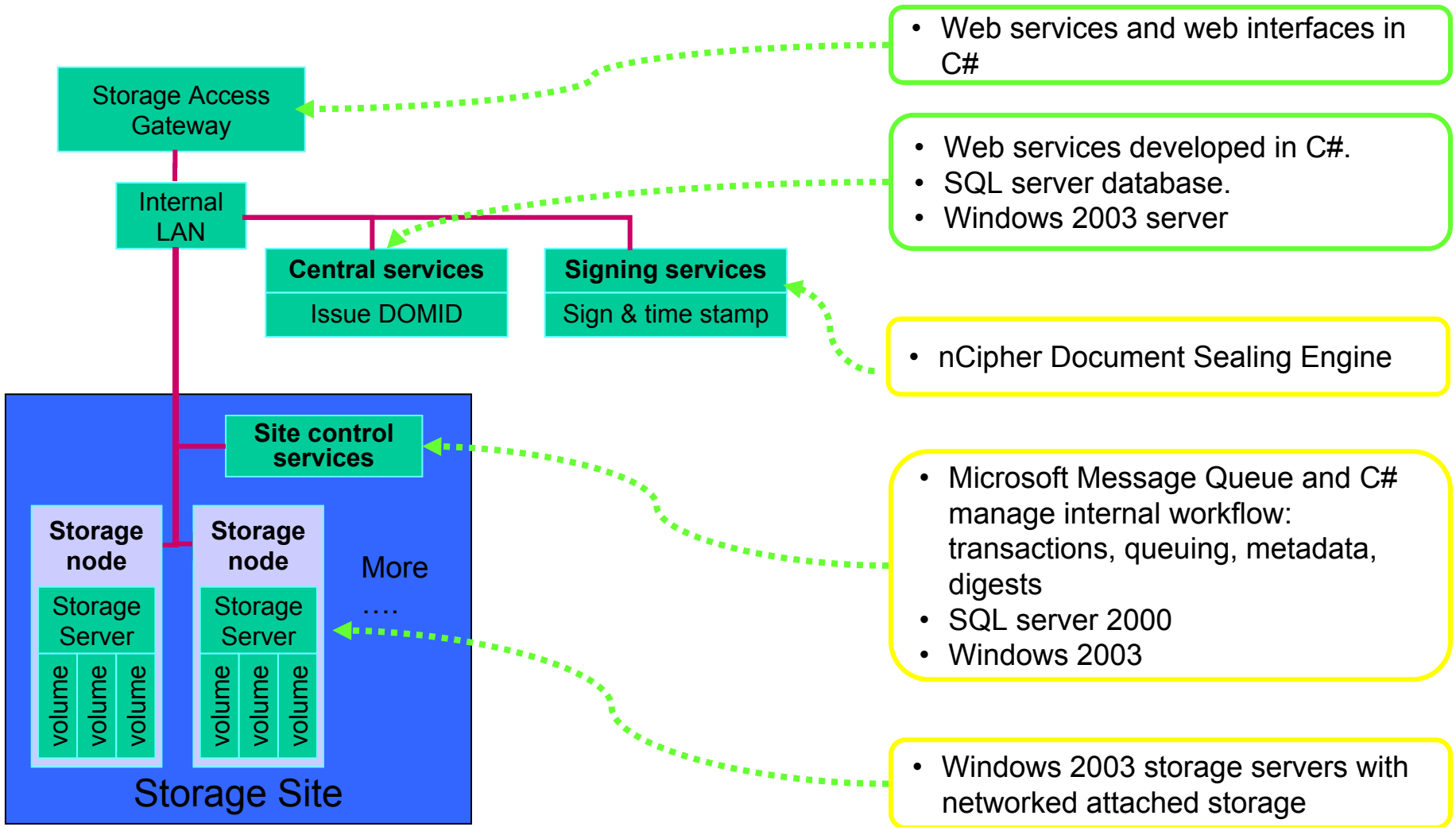
DOM Long-term Storage



Storage Service design



Storage Service design (detail)



Overview of principal actions

Primary Site

- **Copy content file from source to Temp Store**
- **Assign DOMID and notify client**
- **Produce digest and timestamp object**
- **Create signature file and store with content file**
- **Verify signature**
- **Assign local resource locator**
- **Copy content & signature files from Temp Store to Preservation Store**
- **Verify signature**
- **Notify remote site(s) and central services that object is stored**

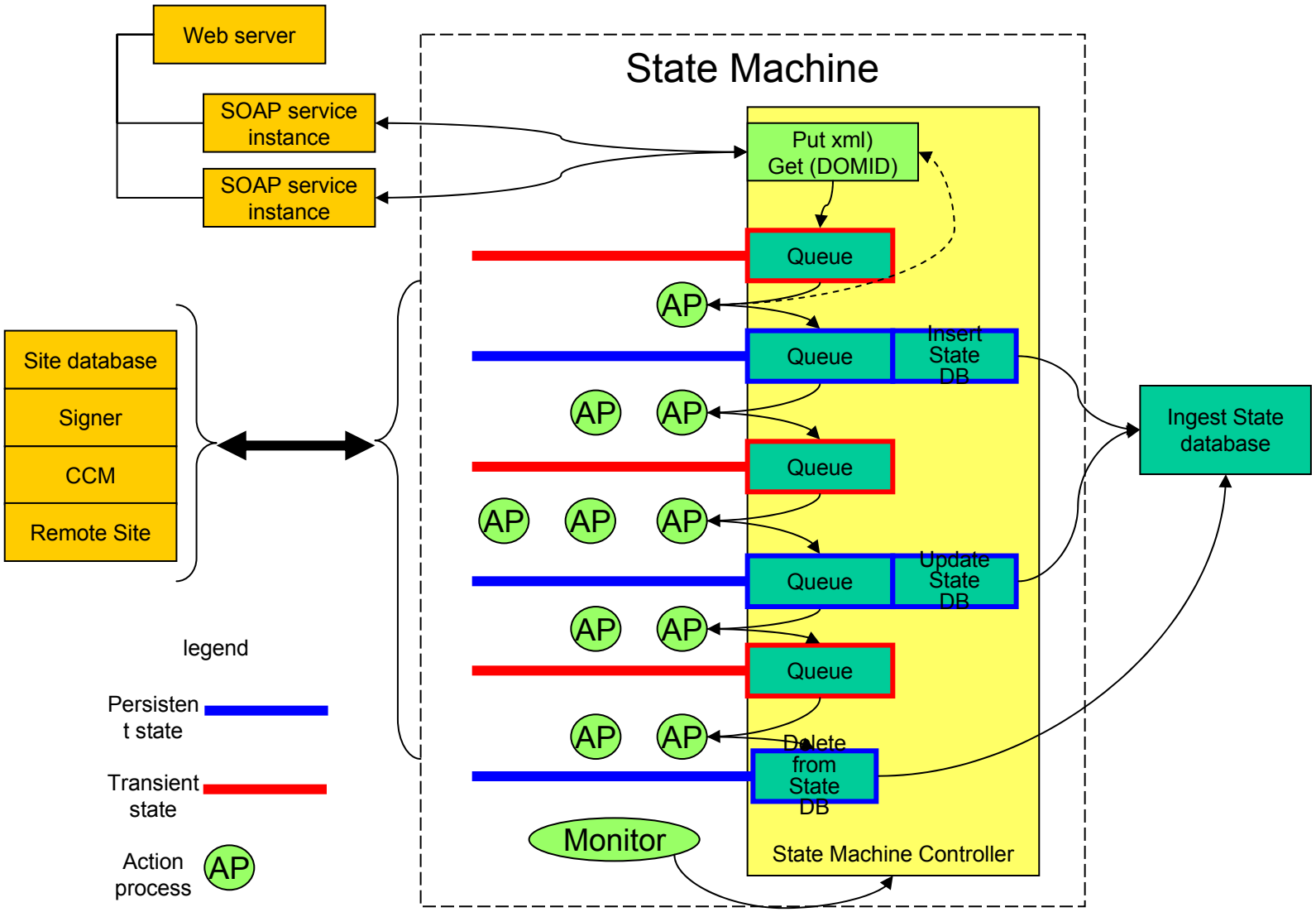
Remote Site (when notified)

- **Copy content & signature files from primary site to Temp Store**
- **Resume at first verify signature action above**

Both

- **Tidy up Temp Store when object has been stored in more than one Preservation Store**

Site controller design



Total Cost of Ownership

Generic drivers in architectural design

- Manage Total Cost of Ownership covering a complete life cycle:
  - Initial purchase
  - Operations support
  - Data Centre Costs
  - Application support and enhancement
  - Replacement cost (hardware and application)
- Disaster recovery
  - Minimise impact on service through common mode environmental incident / disaster
  - Minimise risk of failure
  - Maximise continuity of service
  - Minimise time to recover
- Adaptability of architecture for anticipated requirements
- Performance (though commodity performance seems adequate)

Goals and working assumptions: a slowly evolving story - 1

- Seek to apply a uniform consistent approach as widely as possible
- Avoid constraints or assumptions that a 2nd cluster uses the same kit, or is internally organised in a similar way to the 1st cluster
- Identify and utilise Off The Shelf (OTS) products where applicable
- Seek to provide low cost of ownership by designing the architecture:
  - To minimise staff tasks
  - To take advantage of competitively priced commodity storage

How do we design the architecture to minimise staff tasks?

Topic	Enterprise Systems	DOM	Comments
Decommission old kit	✓	✓	Need to migrate content to new store
Commission new kit	✓	✓	unchanged
Resolve defects	✓	✓	Need reliable kit
Deploy upgrades & patches	✓	✓	unchanged
Take & manage backups	✓	✗	Resilience already provided by having multiple copies
Data recovery	✓	✗	Automatic detection & recovery for “small” failures
Routine monitoring	✓	✗	Single DOM user does not overfill disks

The IT Storage Market

and Emerging Technologies

Goals and working assumptions: a slowly evolving story - 2

- **Proportions of capacity**
  - >80% of capacity will be for large invariant objects that are rarely accessed
  - ~10% of capacity will be for more frequently accessed objects, most of which are small
  - <~1% of capacity will be for conventional variant data
- **This discussion:**
  - focuses predominantly on the “80%” though many of the issues are relevant for the “10%”
  - Assume that the “1%” will be dealt with separately and conventionally
- **Speed of response / delivery assumptions**
  - ~200-300 msec for access objects
  - ~2 sec for preservation objects (though could explore 20-30 secs if significantly cheaper)

Goals and working assumptions: a slowly evolving story - 3

- **There is one uniform view of storage, as seen by the logical architecture comprising nodes, volumes, etc**
- **The uniform view of storage allows for different nodes/volumes to be designated as fast/slow etc**

- **There are three classes of storage, assumptions:**
  - Small access objects are preferentially stored in faster storage (and could be deleted …)
  - Other objects , incl. preservation objects, are preferentially stored in slow storage (not deleted)
  - There is a separate role for cache, probably held near the gateway
    - but this is likely to hold only “unrestricted” objects that anyone may view

Features that do not add value – 1

- **Mirroring of disk volumes:**
  - Each object is already “mirrored” at the site level and is held independently on a 2nd system
  - Disk mirroring is likely to contravene the desire to avoid the constraint to use the same kit/volume structure on a 2nd system
- **Smart techniques for backup management:**
  - The principal basis for resilience is not based on the need to take backups of invariant objects
- **Snapshots:**
  - As invariant objects are added, remain unchanged, and are rarely deleted, snapshots do not seem relevant

Features that do not add value – 2

- **A storage vendor's notion of content indexing and resource discovery will not compare with a Library's view of same**
- **Dynamic resizing of disk volumes:**
  - New volume would be declared to system
  - System would fill it (up to prescribed %)
  - Objects not, or only rarely, deleted
  - Dynamic resizing, when new capacity is added, does not help

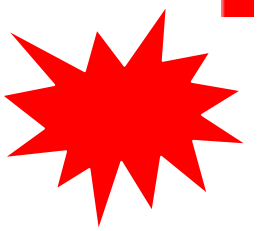
Features that add less value than in conventional systems – 1

- **There are products with additional resilience measures, in addition to RAID:**
  - e.g. multiple paths, controllers etc
  - These are typically deployed where there is no 2nd on-line peer system
  - Thus provide less additional value to DOM than in conventional systems
  - Hence much less likely to be cost effective than in those systems
- **Storage resource management (SRM) typically migrates infrequently accessed objects to slower cheaper storage**
  - We can use a simple storage designation algorithm, based on object type
  - We may store some small objects in unnecessarily fast storage, but are unlikely to store many big infrequent ones in fast storage – "hence a simple allocation algorithm will not do a bad job"

Features that add less value than in conventional systems – 2

- **Virtualisation software – transforms many small volumes into a big one**
  - DOM copes with any arbitrary size of volumes (e.g. 1T is treated in same way as 100T) though we may end up with more unused space
  - Concern over extent of data that may be effected by failure of resource – this could make this a very “bad” idea
- **Internal integrity management:**
  - Given goal to support heterogeneous storage kit with minimum eligibility criteria, we provide our own system wide assurance of integrity and authenticity
  - Implies little additional value added by internal integrity management within a vendor’s offering

Features that add, or could add, value



- **Storage management software**
  - **Very significant benefit:**
    - **Provides good view of storage resources**
    - **Performance monitoring**
    - **Fault detection and fault notification: staff and applications**
  - **Helps reduce staff effort hence costs**
- **Data protection – WORM etc**
  - **Has potential though not obvious how to apply it on its own**
  - **However, without it:**
    - **IT Operations staff could always delete files - though they should not on DOM**
    - **Object deletion ought to be a "proper use case" rather than ad hoc**
    - **Could enforce deletion of non-access objects - is explicitly repeated on 2nd cluster**

Implications for cluster architecture

- **The principal feature that adds value to the architecture is Storage Management Software**
- **There are no other hardware/system features that add any significant value**
- **Conclude:**
  - Build a cluster using basic, very cost effective but intrinsically reliable storage products

Storage Technology Choices (1)

- **Disk media – Serial ATA (SATA) – Preservation storage, SCSI and/or Fibre Channel – Temporary Store**

- **Storage controllers – concern over maturity of SATA, however a number of vendors initially married SATA storage to existing FC/SCSI controller technology**

- **Storage frames – "commodity" rather than monolithic, minimum CIFS, NFS, HTTP support, initial preference W2k03 Storage Server**

- **Current status – each site has 12TB SATA arrays (Preservation storage) and 2TB SCSI arrays (Temporary Store) – HP MSA1500 plus HP DL380SS NAS. In addition we have a Testing & Staging installation where each site has 5TB SATA arrays (Preservation storage) and 1TB SCSI arrays (Temporary Store) – ACNC JetStor plus HP DL380SS NAS.**

-

Storage Technology Choices (2)

- **Storage Networking – given our architecture (two geographically remote store "instances" plus a separate Dark Archive) – iSCSI is attractive but technology possibly immature – but SMS software mandates a networked storage environment.**

- **Current Status – investigate during 2005.**

- **Storage Management Software – Microsoft say we don't need it but can use the native utilities included in Storage Server 2003 and future versions – but most of these are command line and not especially user friendly.**

- **In order to maximise storage admin staff resource utilisation we need to invest in third-party SMS which supports heterogeneous storage hardware and which mandates a networked storage environment. Two challengers in this area which are interesting are AppIQ and Creekpath.**

- **Current Status – investigate during 2005.**

Dark Archive

- Need to specify – requirements largely undefined.
- "Last chance saloon" in the event of disaster or the loss of one or more objects from both instances of the preservation storage
- Storage media technology therefore needs to be archive level approved and immutable for a known period.
- Should be located geographically separately to either storage instance – ideally translates as "offsite" – outsourced permanently - or initially

Current Status

- **Competitive tender awarded to Intechnology in March, 2005**
- **Objects will be replicated via the Library's St Pancras internet link to Intechnology's secure network, and stored at their secure repository.**

Storage Market – Emerging Technologies

- Still a significant amount of consolidation as monolithic vendors buy the technologies they need to in order to offer “Information LifeCycle Management” solutions
- Monolithic vendor profit margins lie in the sale of software, maintenance and support services – which largely we do not need.
- MAID (Massive Array of Inactive Disks) – power saving – marketed as virtual tape – Copan Systems
- SATA disk capacity – 500GB (2005 Q2), 1TB (2007)
- Perpendicular Recording (2007/8)
- Replacement Technologies (2010 ->)
- Reductions in disk form factor, power requirements and heat generation

Further information

Contact details

- **On the Library Web pages**
  under ‘About us / Policies & Programmes’

  http://www.bl.uk/about/policies/dom/homepage.html

- **Contact us**

richard.masters@bl.uk	**01937 546888**
sean.martin@bl.uk	**01937 546716**
jim.linden@bl.uk	**01937 546868**
roderic.parker@bl.uk	**01937 546090**

Open Forum

- **Persistent identifiers**
  - What to identify?
  - Which scheme? DOI, NBN, …
  - Resolution, services
- **Integrity and authenticity**
  - Extend the chain of custody
  - Long term implications
- **Metadata**
  - Standards: METS / MPEG21?
  - Preservation metadata – PREMIS?
  - Long term storage of metadata…