DPN

THE DIGITAL PRESERVATION NETWORK

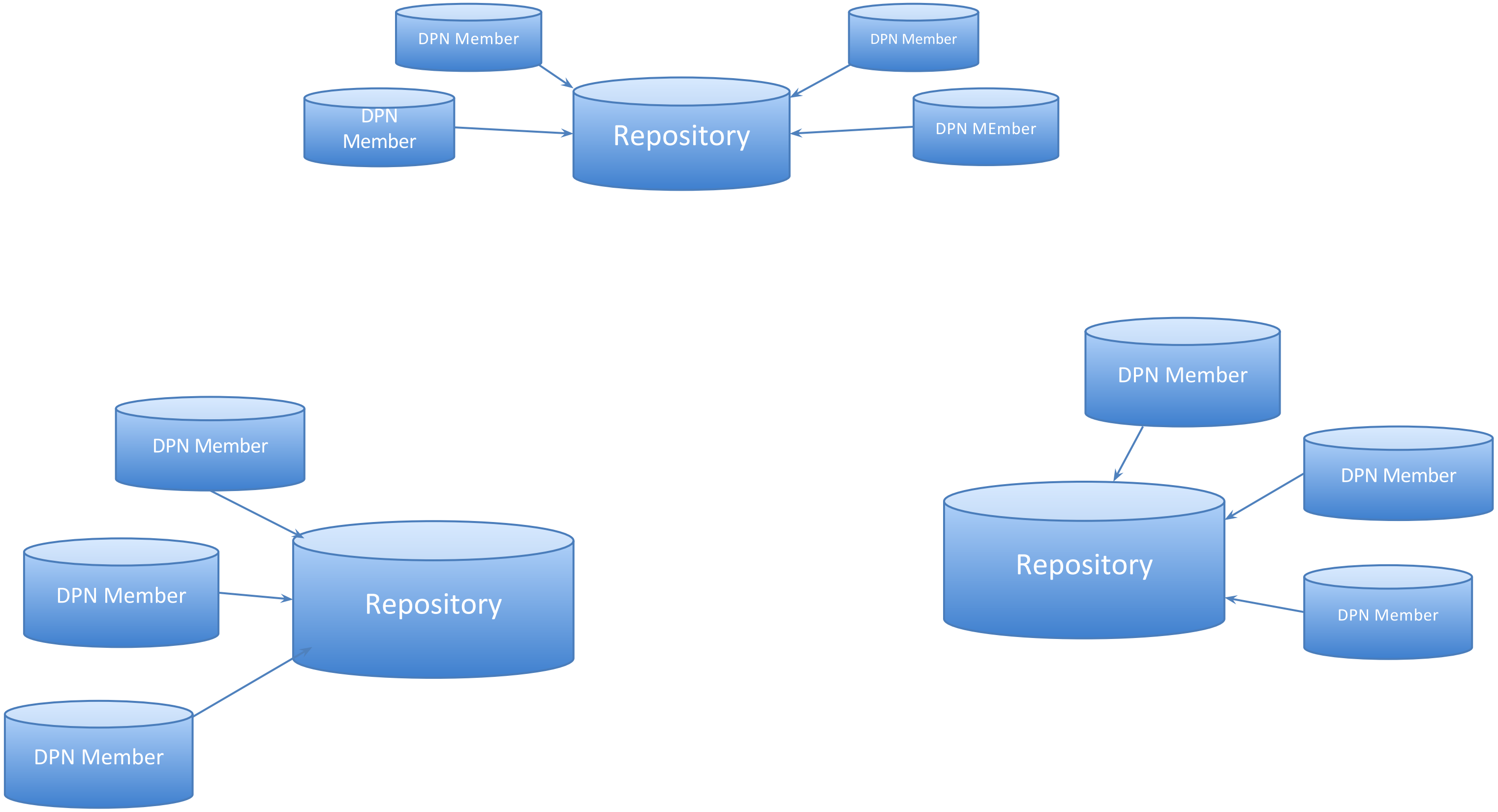
A Report On DPN’s Emerging Architecture, System Protocol, and Service model

PASIG

May 21, 2013 Washington D.C.

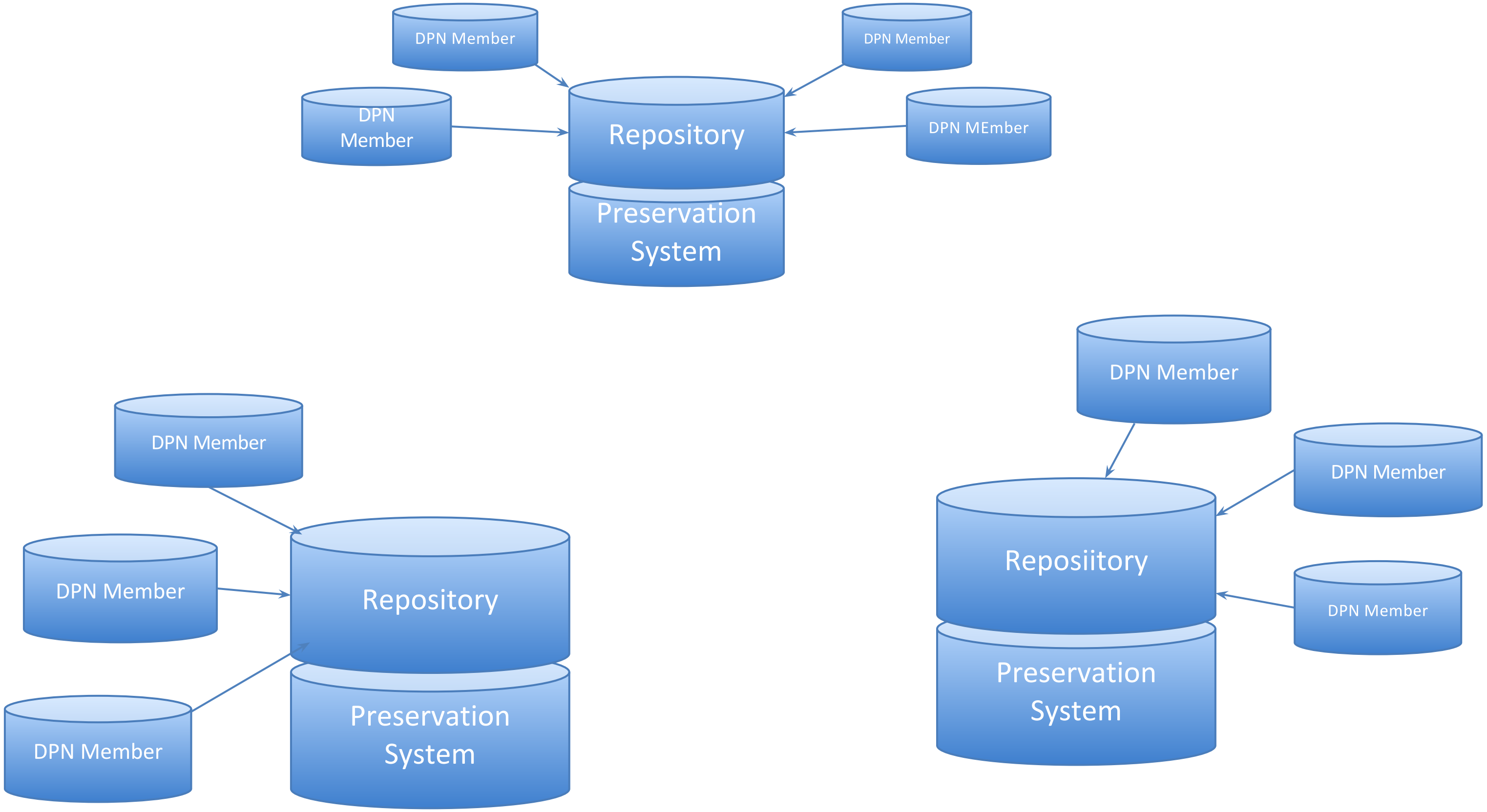
DPN Technical Team

What Is DPN?



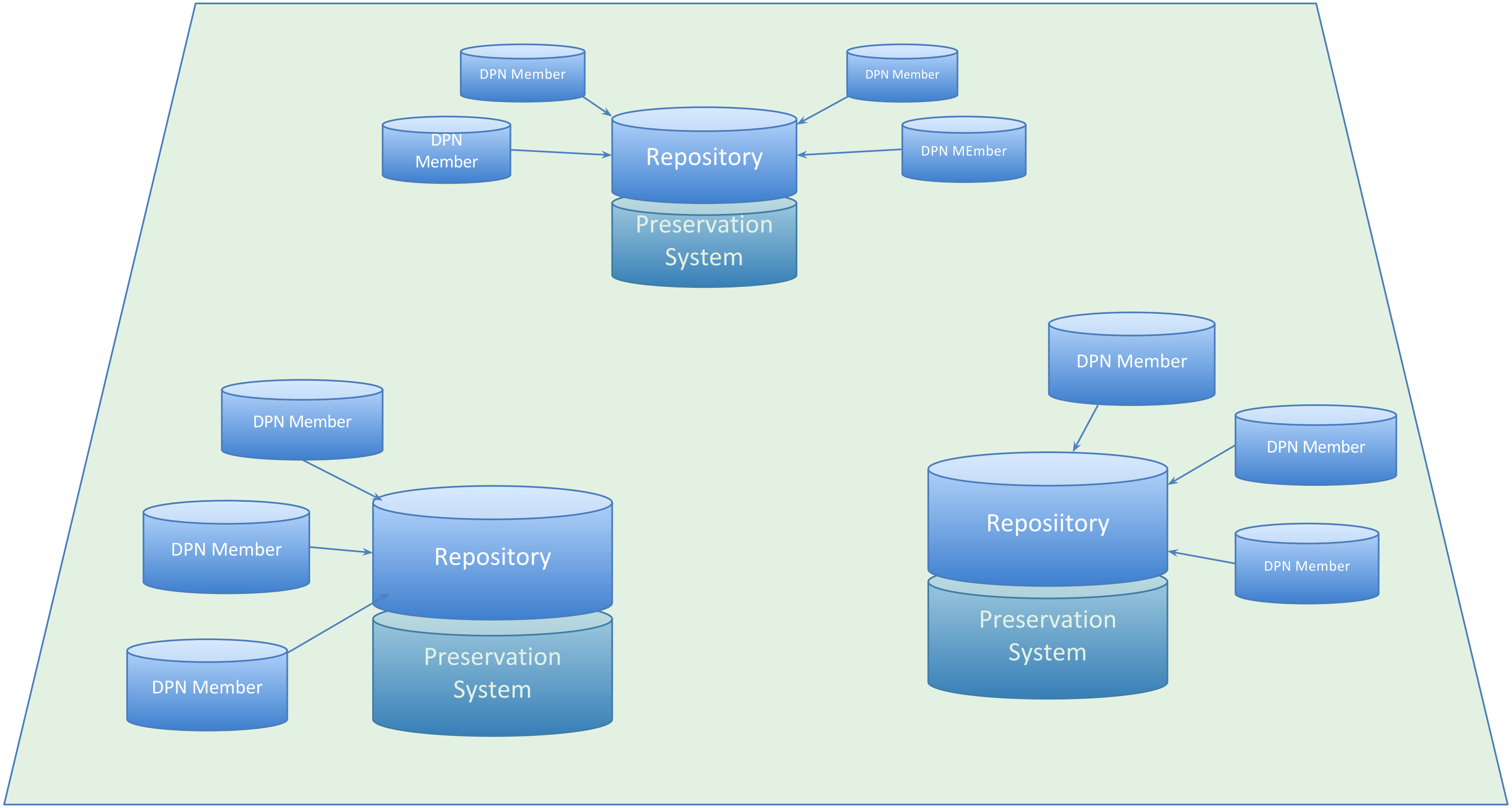
57 member organizations cooperatively investing in long-term, scalable, digital preservation

What Is DPN?



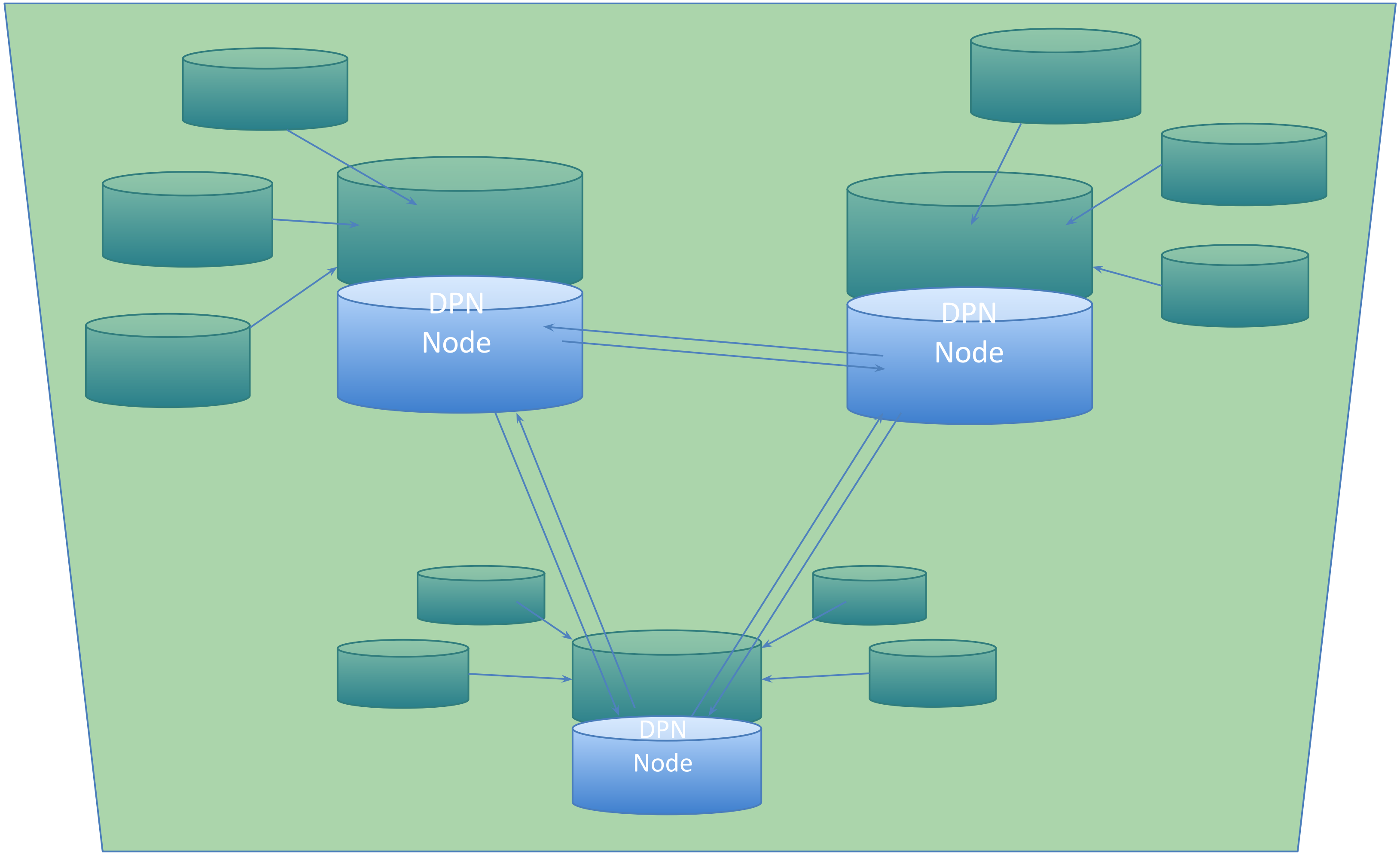
technical staff and systems from 5
large scale preservation repositories

What Is DPN?



…working groups of experts in succession rights, business services, communications and research data…

What is DPN?



All building a digital preservation backbone for the academy

Initial DPN technical partners

Initial DPN launch will feature five nodes:

- Academic Preservation Trust (APTrust)
- Chronopolis
- HathiTrust
- Stanford Digital Repository (SDR)
- University of Texas Data Repository (UTDR)

And a participating partner:

- DuraSpace

The DPN Technical Team

APTrust

- Scott Turnbull
- Tim Sigmon
- Adam Soroka

Chronopolis

- David Minor
- Mike Smorul
- Don Sutton
- Mike Ritter

DuraSpace

- Andrew Woods

HathiTrust

- Sebastien Korner
- Bryan Hockey

Stanford

- Tom Cramer
- James Simon

Texas Data Repository

- Ladd Hanson
- Christopher Jordan
- Dan Galewsky

What Does DPN Do?

1. Establishes a network of heterogeneous, interoperable, trustworthy, preservation repositories (Nodes)
2. Replicates content across the network, to multiple nodes
3. Enables restoration of preserved content to any node in the event of data loss, corruption or disaster
4. Ensures the ongoing preservation of digital information from depositors in the event of dissolution or divestment of depositors or an individual repository
5. Provides the option to (technically and legally) "brighten content" preserved in the network over time

DPN Benefits

1. Resilience
2. Succession
3. Economies of scale
4. Efficiency
5. Extensibility
6. Security

Critical Assumptions & Definitions

- All content enters DPN by deposit into one of the DPN Nodes.
- The “First Node” is the point of entry for a given piece of content; Nodes with copies of this content are “Replicating Nodes”.
- DPN Members will work directly with an individual DPN Node to negotiate contracts and determine service levels
- Service levels and contracts will reflect “standard” DPN services; they may also reflect the First Node’s unique offerings in terms of access, hosting or other services.
- Content in Replicating Nodes will be held “dark”, and inaccessible except for preservation actions.

Critical Assumptions & Definitions

- DPN shall redistribute preserved content as Nodes enter and leave the Network, ensuring continuity of preservation services over time.
- DPN will provide a large-scale network of dark archives that enable the opportunity to brighten content in the future, but does not mandate how this is done.
- Depositors, First Nodes and their designated communities will collaborate to ensure that the information contents of DPN deposits are accessible for reuse in the future, using the appropriate (and evolving) community standards for any given set of content.

Specifications (sample of…)

1. *DPN will make multiple copies of content from a First Node to Replicating Nodes.*

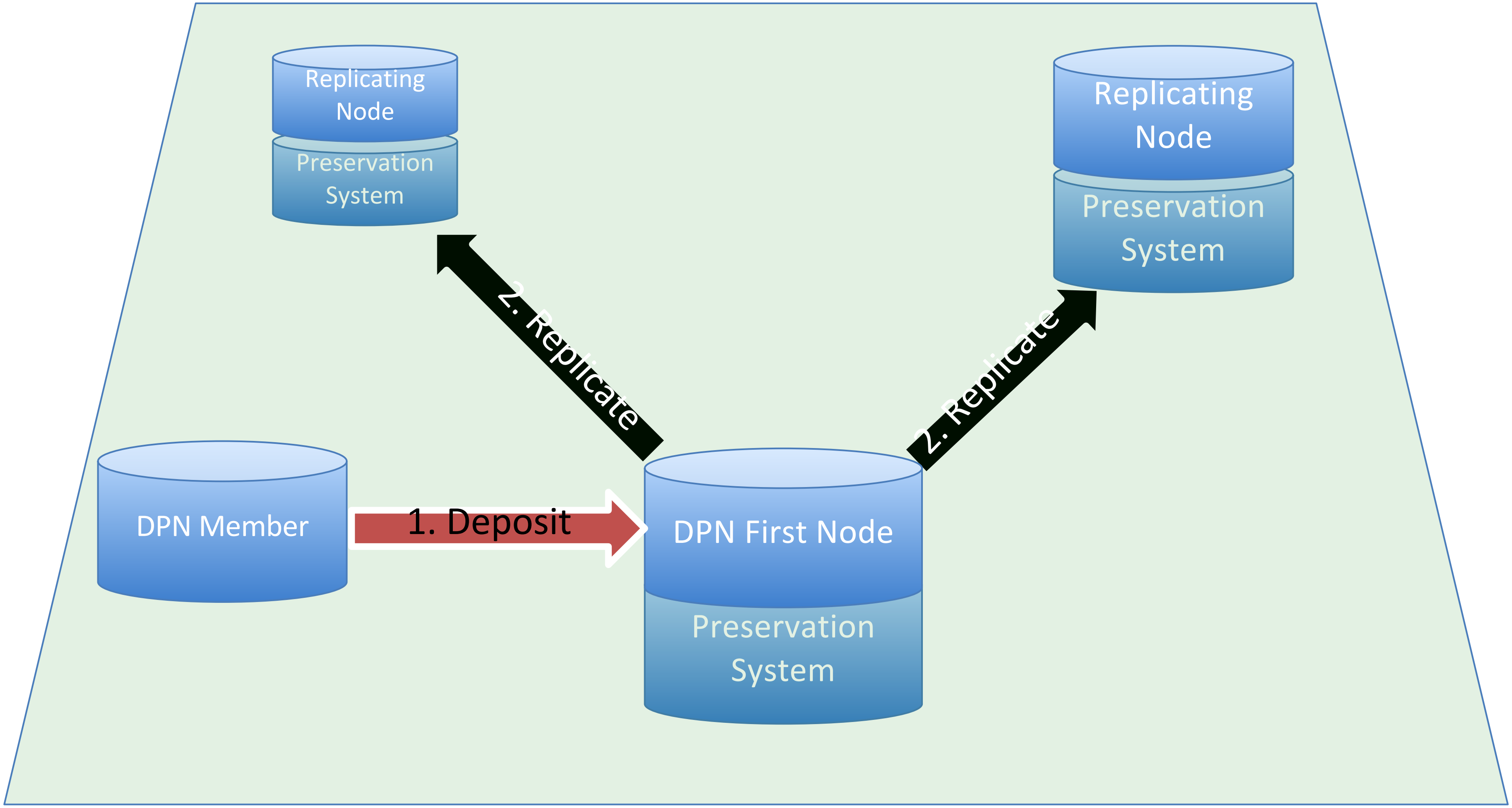
5. *DPN will repair or replace any replicated content at any node when corruption is detected.*

7. DPN will assure the security of replicated content during transmission so that no content is lost, corrupted, or exposed.

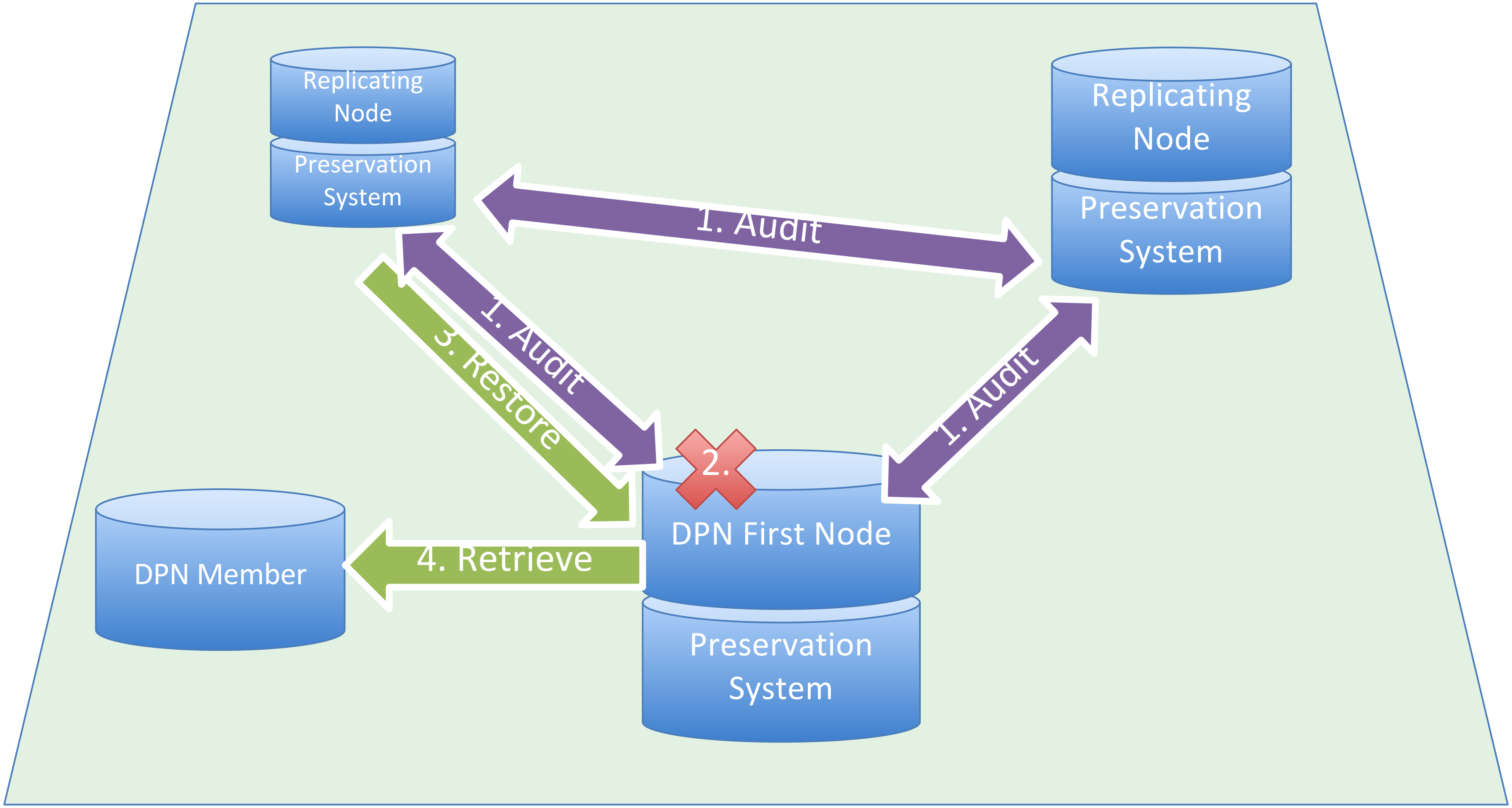
16. DPN will be able to support the introduction or exit / cessation of DPN Nodes by redistributing content among new/continuing nodes to ensure sufficient copies are kept according to policies.

See https://wiki.duraspace.org/display/DPNC/Specifications

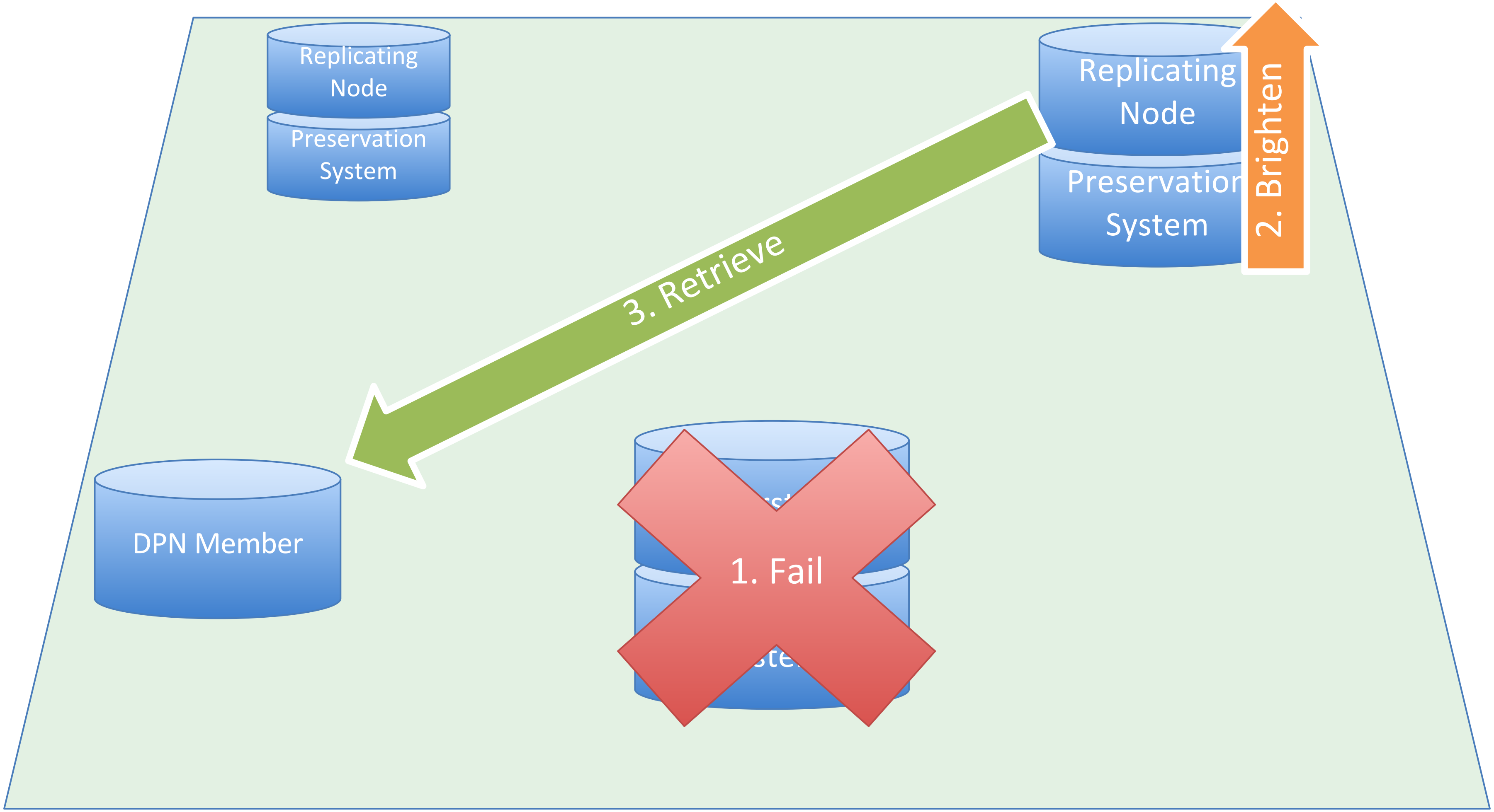
Scenario 1: Ingest & Replication



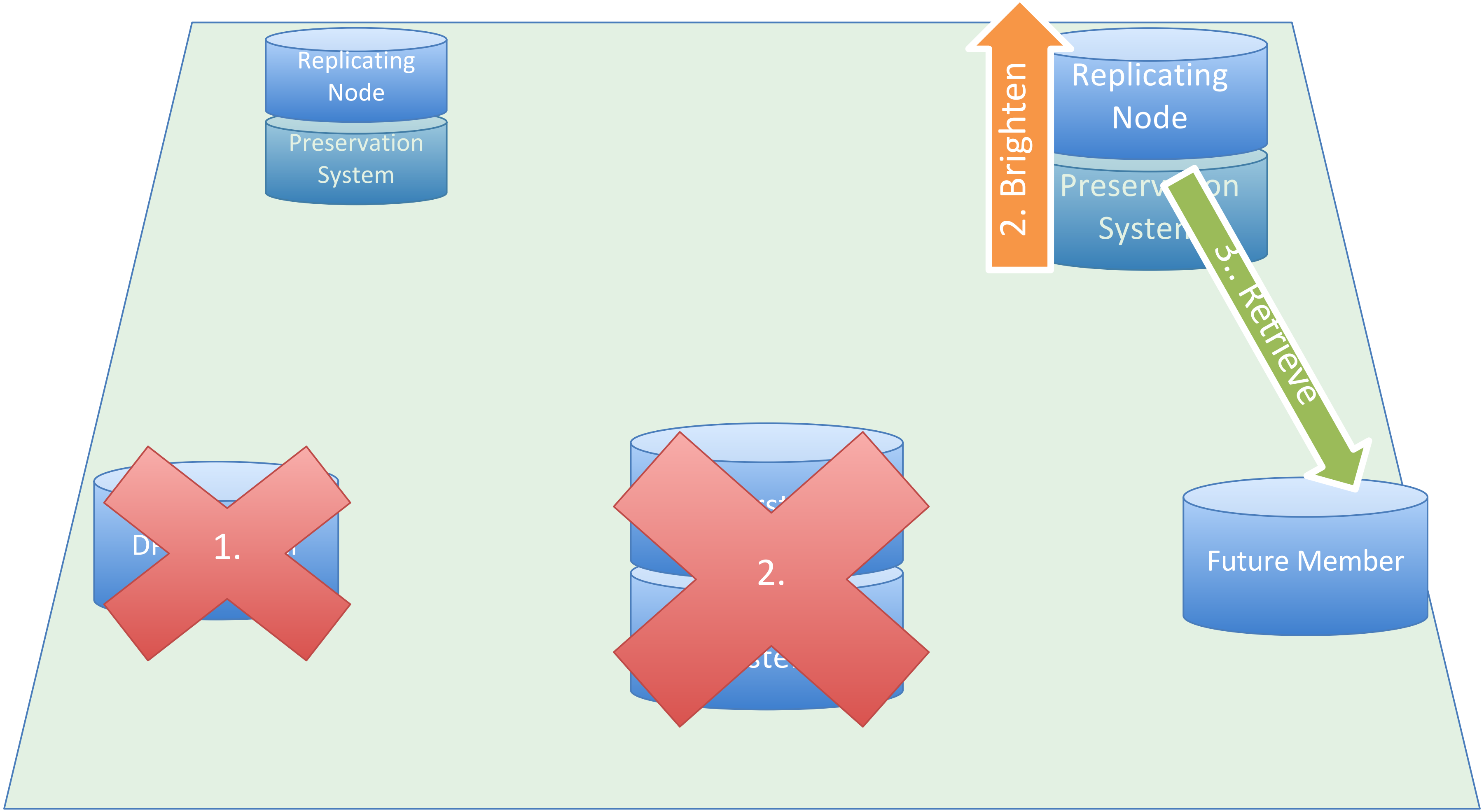
Scenario 2: Restoration of Content



Scenario 3: First Node Cessation



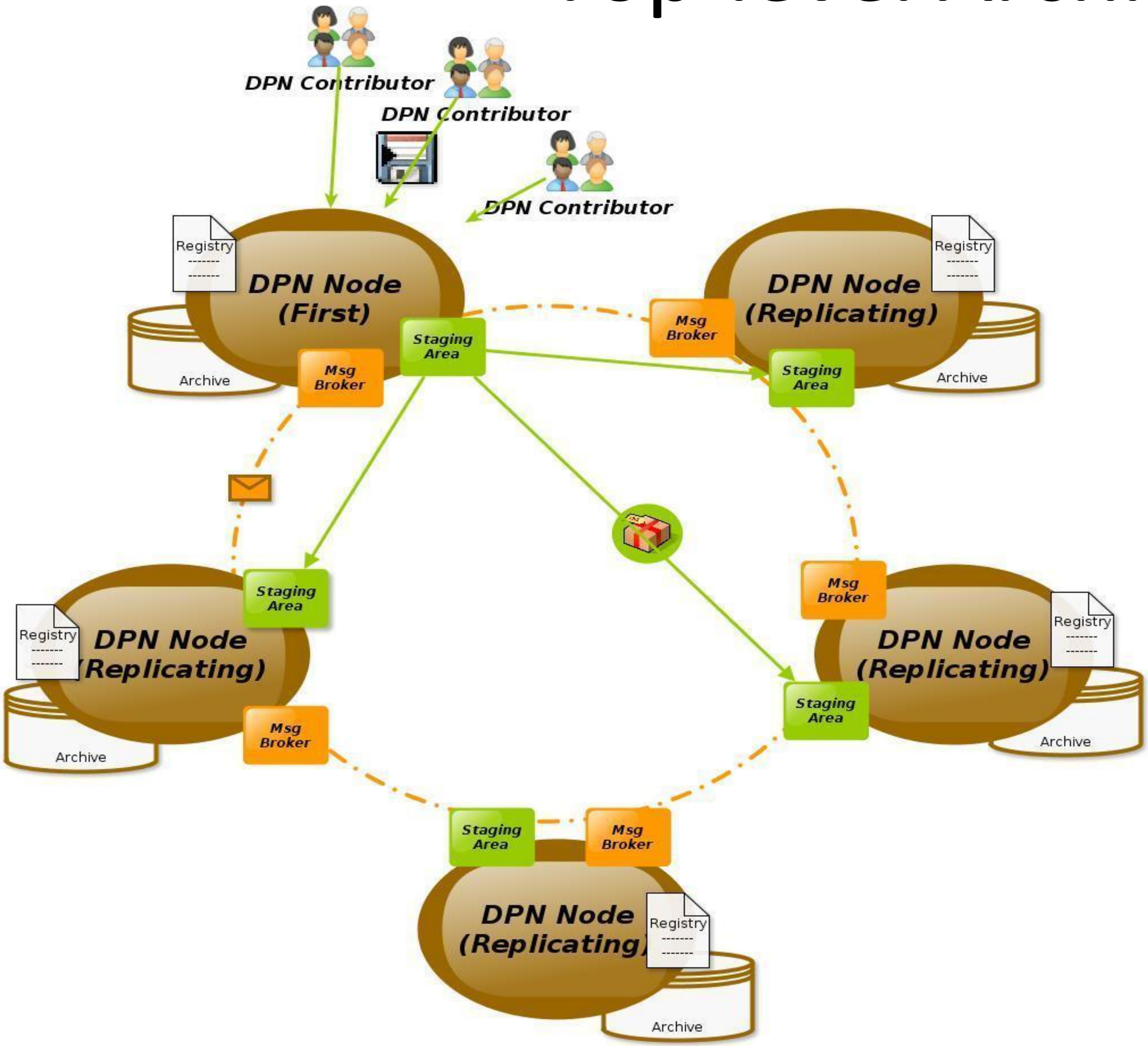
Scenario 4: Successioning



Architectural Overview

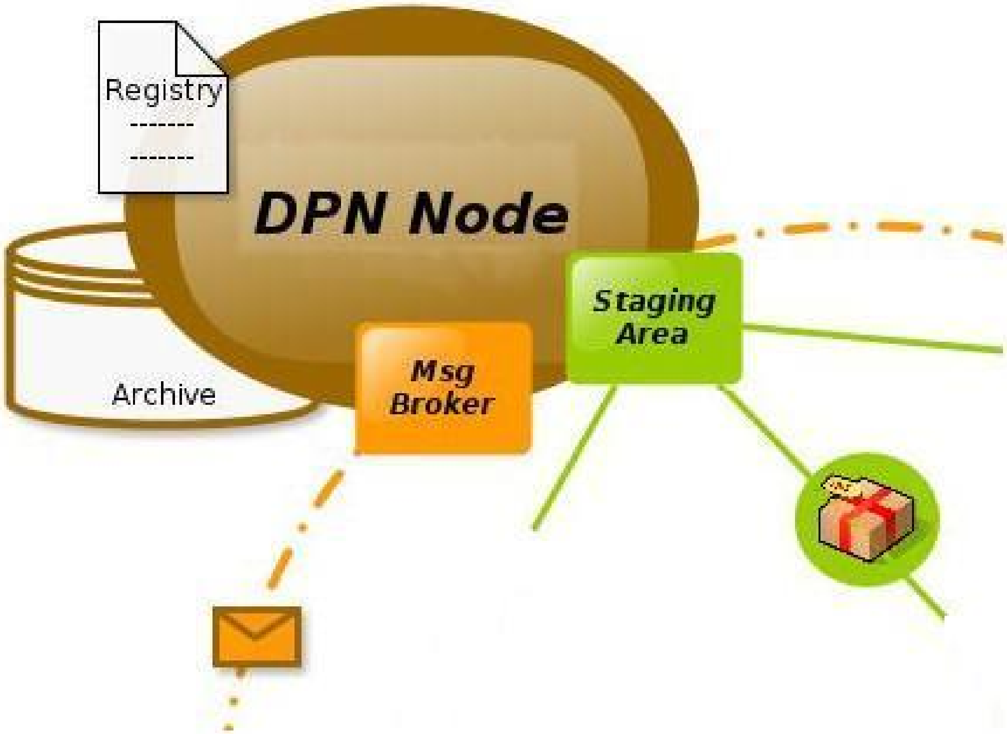
- Architectural Premise
  - *Core capabilities founded on proven institutions and repositories*
- Design Considerations
  - *Distributed Nodes, loosely coupled*
  - *Standards and protocol-based integrations*
  - *Separate implementations*
  - *Distributed infrastructure*

Top-level Architecture



Infrastructure Components

- *Institutional Archive/Repository*
- *Federated Messaging*
- *Distributed Registry*
- *Transfer Mechanisms*
- *Content Packaging*
- *Security and Encryption*



Infrastructure Components

- ***Institutional Archive/Repository***

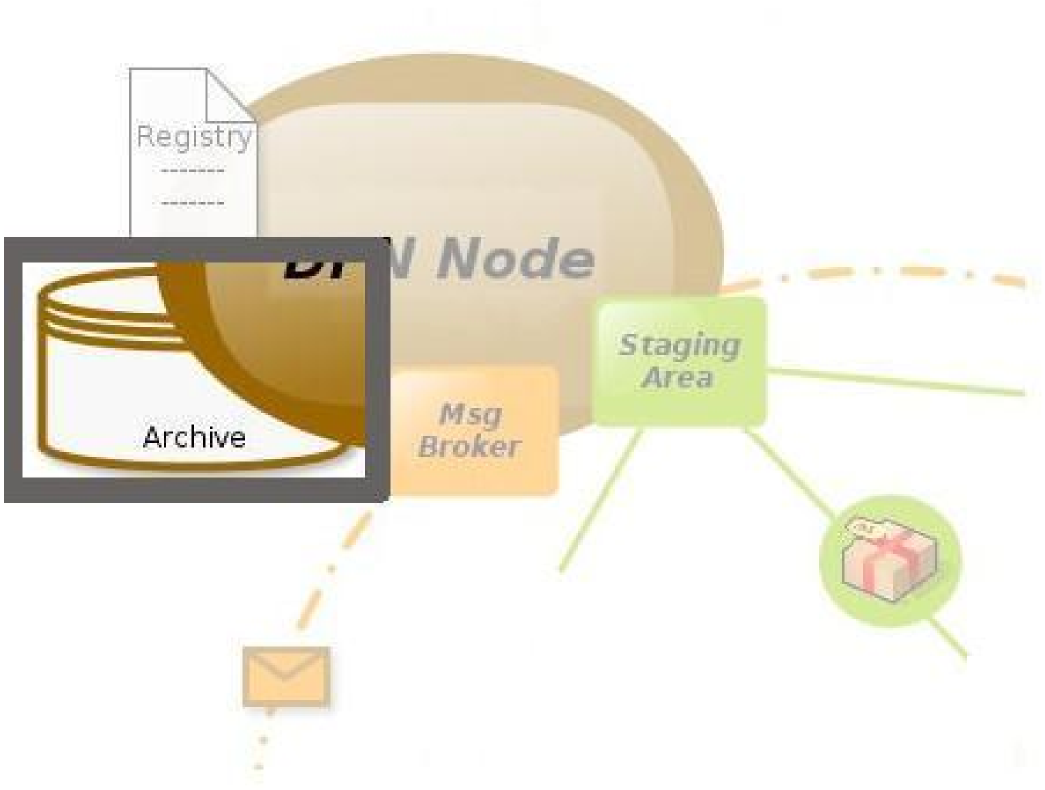
- *Federated Messaging*

- *Distributed Registry*

- *Transfer Mechanisms*

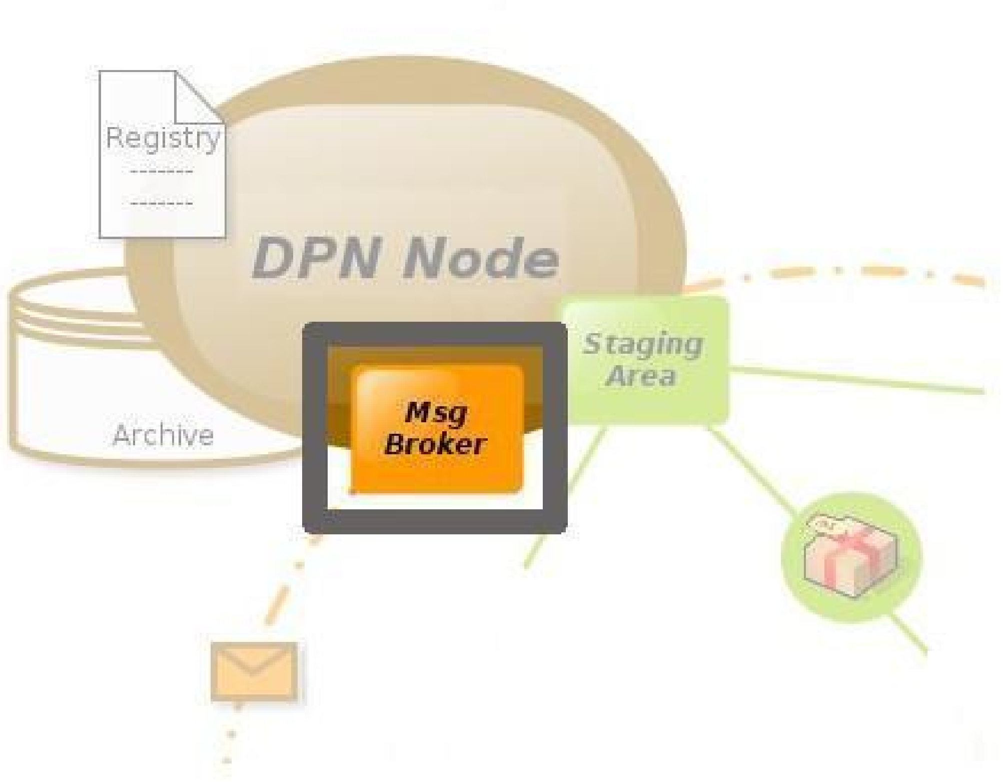
- *Content Packaging*

- *Security and Encryption*



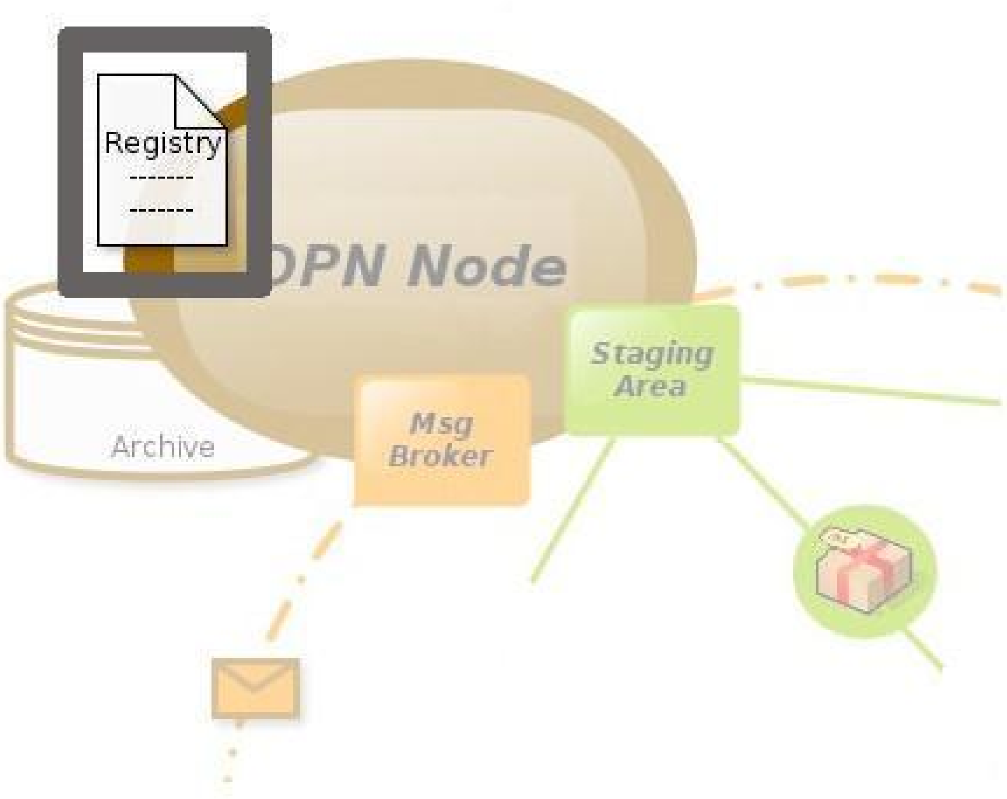
Infrastructure Components

- *Institutional Archive/Repository*
- ***Federated Messaging***
- *Distributed Registry*
- *Transfer Mechanisms*
- *Content Packaging*
- *Security and Encryption*



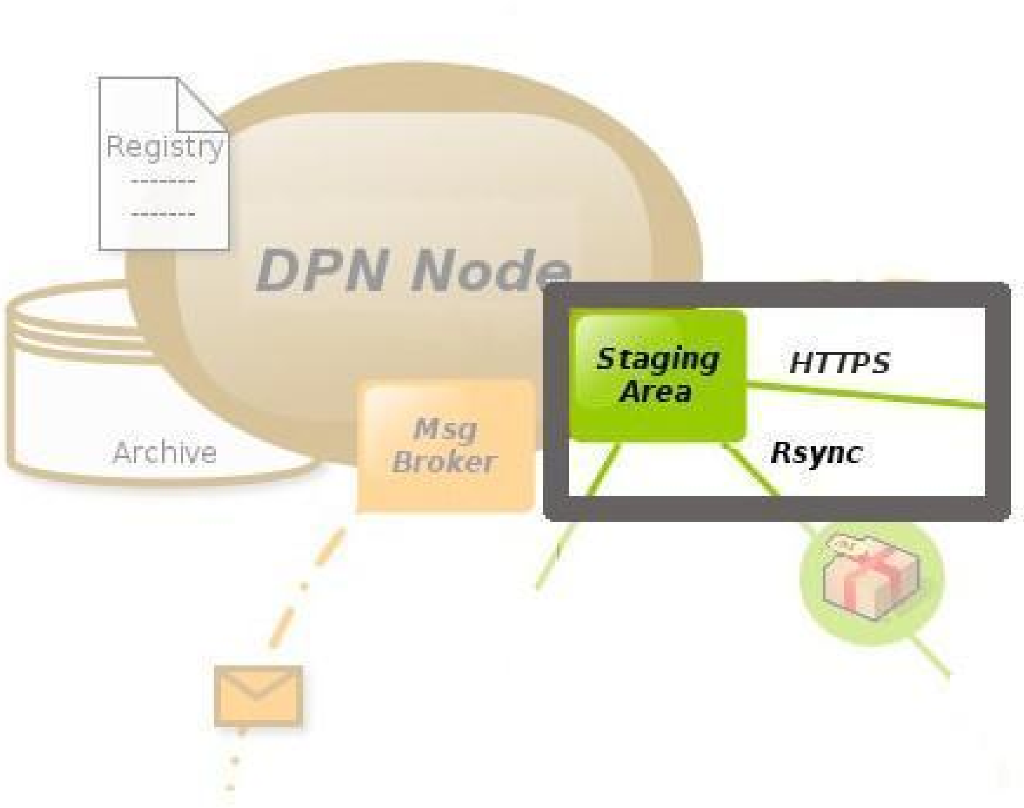
Infrastructure Components

- *Institutional Archive/Repository*
- *Federated Messaging*
- ***Distributed Registry***
- *Transfer Mechanisms*
- *Content Packaging*
- *Security and Encryption*



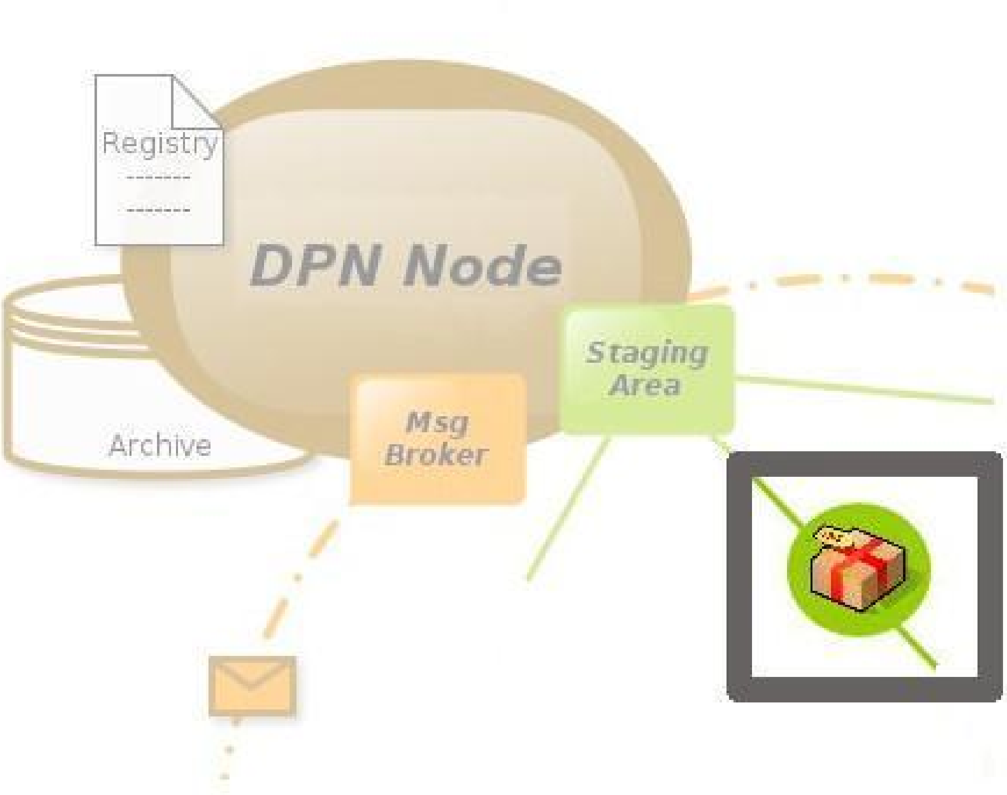
Infrastructure Components

- *Institutional Archive/Repository*
- *Federated Messaging*
- *Distributed Registry*
- ***Transfer Mechanisms***
- *Content Packaging*
- *Security and Encryption*



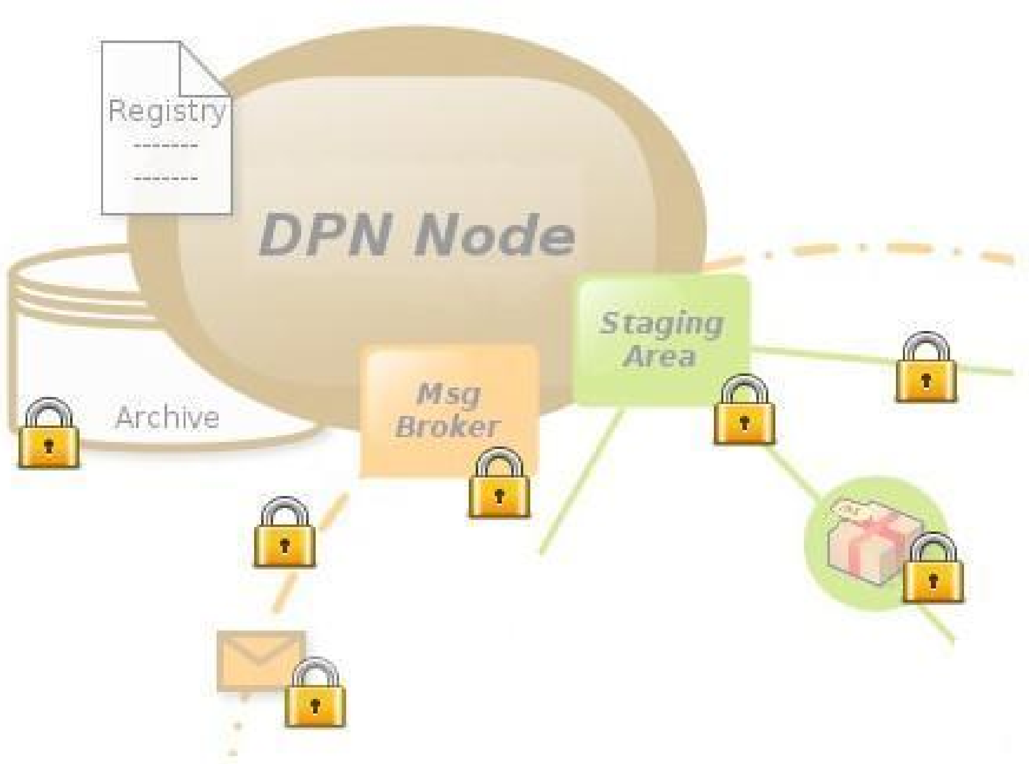
Infrastructure Components

- *Institutional Archive/Repository*
- *Federated Messaging*
- *Distributed Registry*
- *Transfer Mechanisms*
- ***Content Packaging***
- *Security and Encryption*



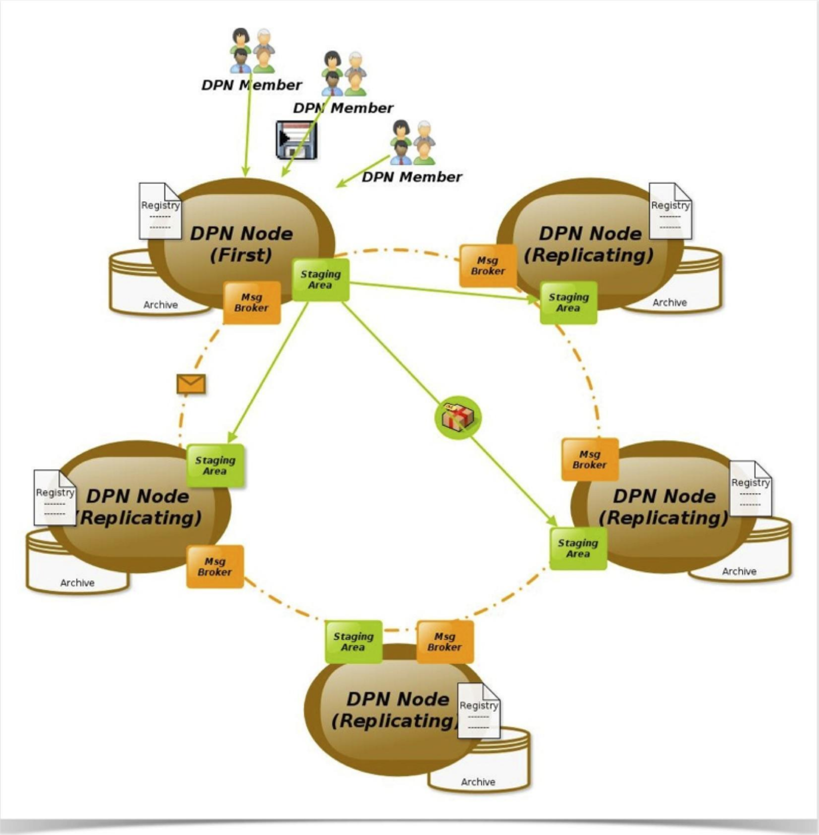
Infrastructure Components

- *Institutional Archive/Repository*
- *Federated Messaging*
- *Distributed Registry*
- *Transfer Mechanisms*
- *Content Packaging*
- ***Security and Encryption***



The DPN Federation

- Each node in the federation acts as both a First Node, and a Replicating Node
- The Nodes will put content into DPN from their own repository, or from a DPN member
- There is a new hierarchy for content, with DPN First Nodes having responsibly for preservation activities within the DPN federation
- Part of those activities include keeping agreements up to date, transitioning fixity to new algorithms, fixity audit across the federation, logging, etc.



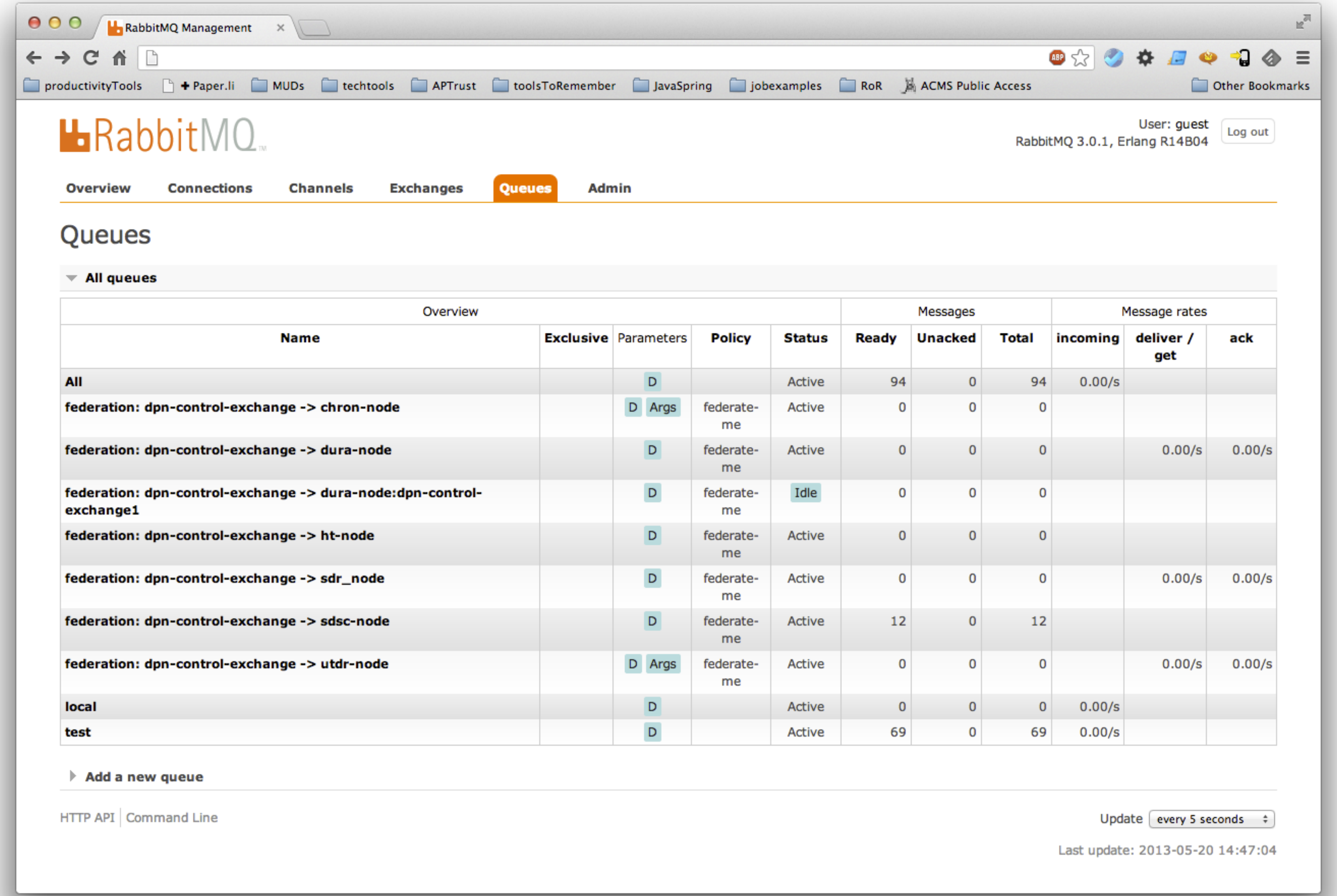
Components in Technical Architecture

- Messaging infrastructure to support federated services
- Registry to track objects within the federation, including copies, version, rights, brightening information
- Transfer mechanisms (rsync, https, gridFTP, etc.)
- Private PKI for securing transport layers
- Logging and reporting
- Other components we implement separately, but may be common, for example a secure transfer area.
- DPN objects that hold administrative content such as DPN framework agreements, DPN bagit profiles, versioned Brightening information for a collection/repository

Federated Messaging

- DPN uses messaging for in band communication and control and replication and services are handled out of band
- Using RabbitMQ message brokers, which support AMQP (Advanced Messaging Queueing Protocol)
- RabbitMQ also supports federated messaging easily via a plugin
- DPN messaging model uses Topic Queues for broadcast messages and direct queues for one-to-one communication between nodes

Federated Broker View



User: guest
RabbitMQ 3.0.1, Erlang R14B04

Queues

All queues

Overview					Messages			Message rates		
Name	**Exclusive**	Parameters	**Policy**	**Status**	**Ready**	**Unacked**	**Total**	**incoming**	**deliver / get**	**ack**
All		D		Active	94	0	94	0.00/s		
federation: dpn-control-exchange -> chron-node		D Args	federate-me	Active	0	0	0			
federation: dpn-control-exchange -> dura-node		D	federate-me	Active	0	0	0		0.00/s	0.00/s
federation: dpn-control-exchange -> dura-node:dpn-control-exchange1		D	federate-me	Idle	0	0	0			
federation: dpn-control-exchange -> ht-node		D	federate-me	Active	0	0	0			
federation: dpn-control-exchange -> sdr_node		D	federate-me	Active	0	0	0		0.00/s	0.00/s
federation: dpn-control-exchange -> sdsc-node		D	federate-me	Active	12	0	12			
federation: dpn-control-exchange -> utdr-node		D Args	federate-me	Active	0	0	0		0.00/s	0.00/s
local		D		Active	0	0	0	0.00/s		
test		D		Active	69	0	69	0.00/s		

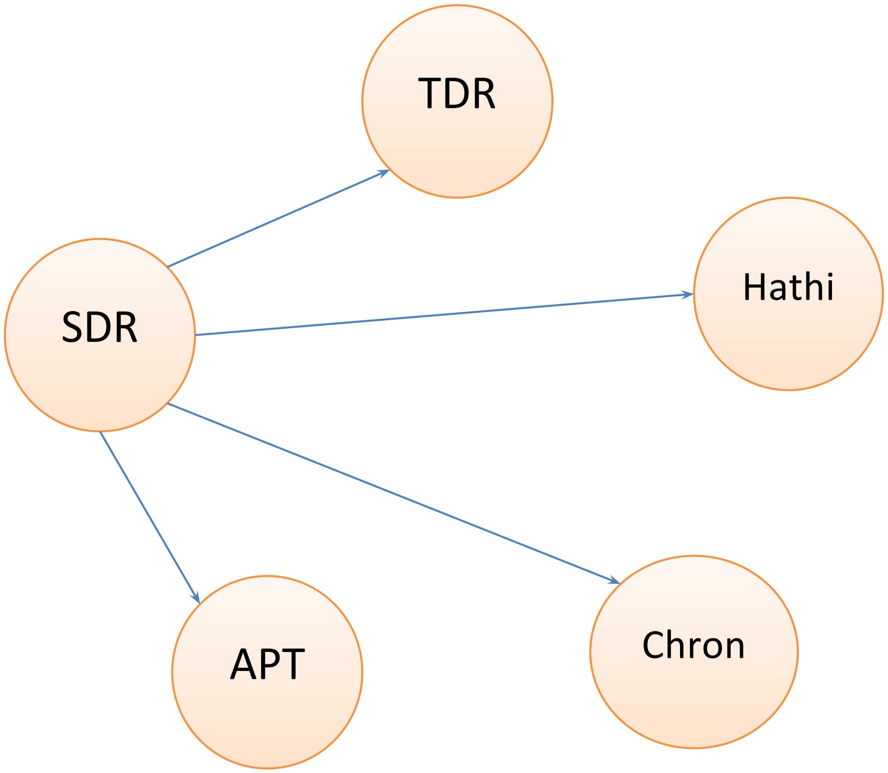
Add a new queue

Update every 5 seconds

Last update: 2013-05-20 14:47:04

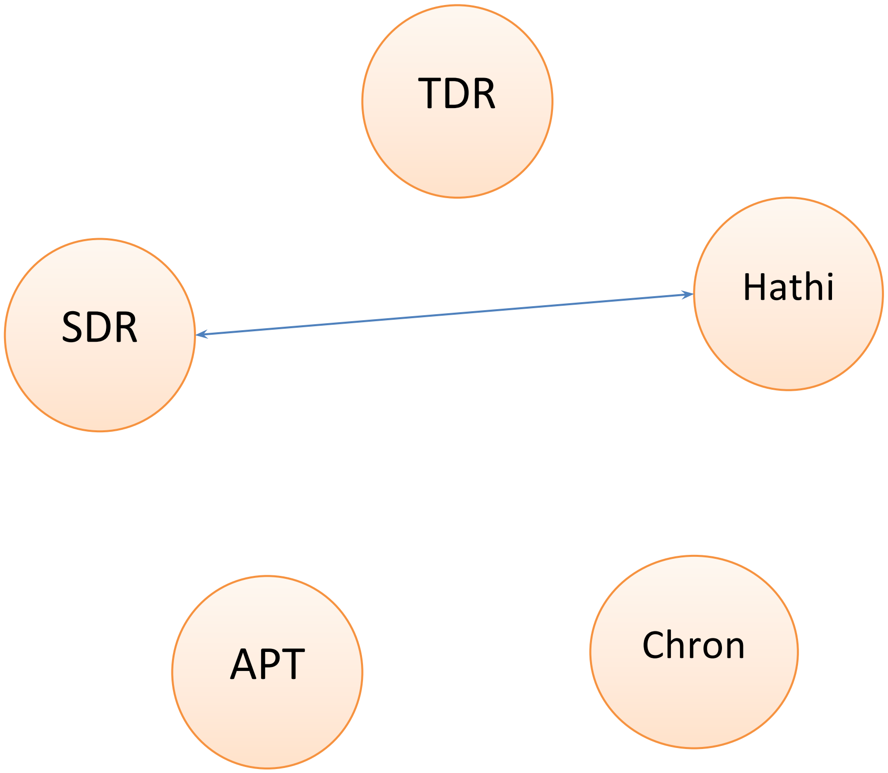
Messaging Model

- Broadcast messages are sent to all node brokers
- Node brokers federate all messages, so if one broker is down it is still possible to communicate



Messaging Model

- Direct messages are between two nodes, used for replies in a message sequence
- Broker federation still applies, so communication channels are redundant



Messaging Control Flows

- Message control flows are transactional, and asynchronous

Examples of control flows

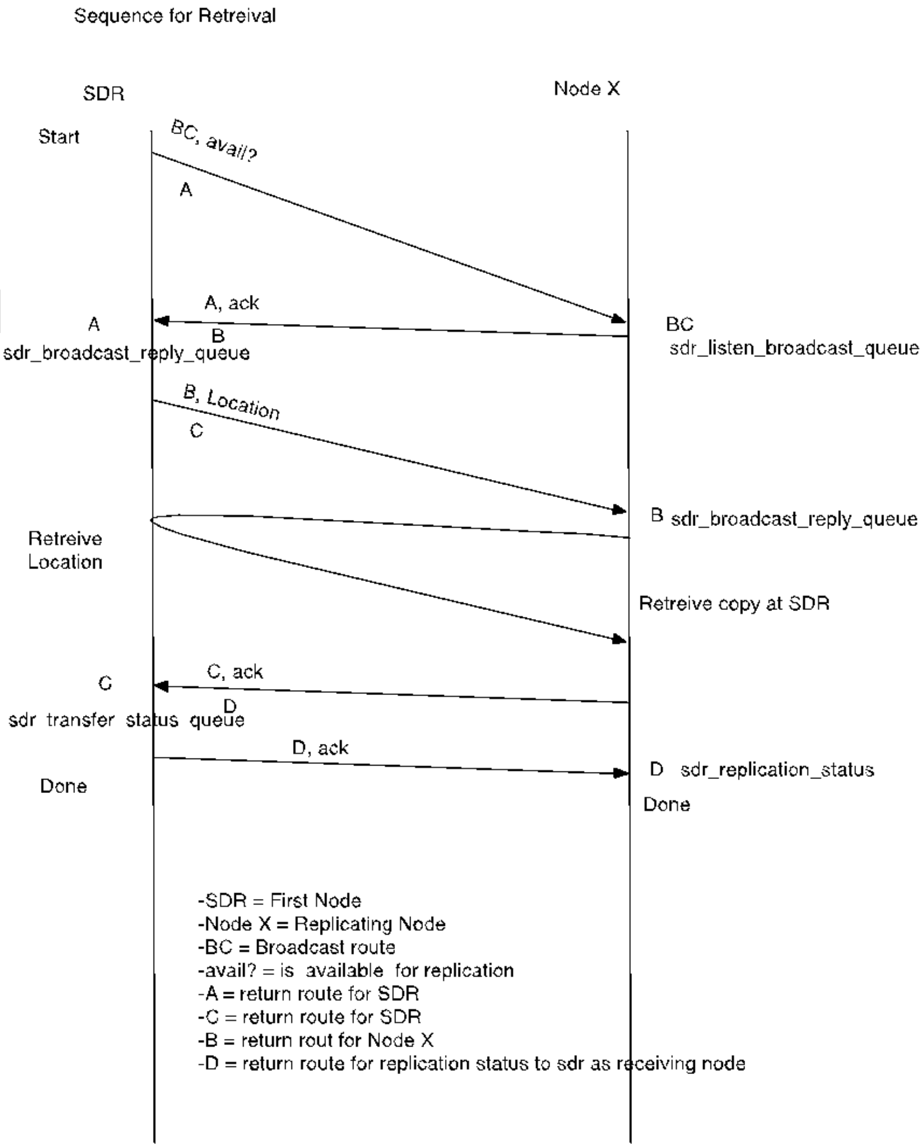
- Replication Request
- Registry Item Create
- Registry Synchronize
- Recovery (digital object, registry entry, registry, etc.)
- Fixity Audit flows
- At any given time each node may be handling multiple message control flows/ sequences at once

Messaging / Not Workflow

- Because DPN is federated and each node is independent,
- DPN does not have a “workflow” system
- So, each of the message sequences provides the mechanism to control the flow of work that needs to be done across the DPN federation
- In this environment, messages are not enough, as there may be failures at any point in a message sequence
- Each node will have to keep track of state of the messages and will have a mechanism to time out and/or recover

Replication Control Flow

- Retrievals happen out of band over secure channel
- Each step can be canceled and the transaction stopped
- Content is checked for fixity (sha256) at the replication node and also at the First node (SDR in the example) after copy
- State of transaction managed by each node separately



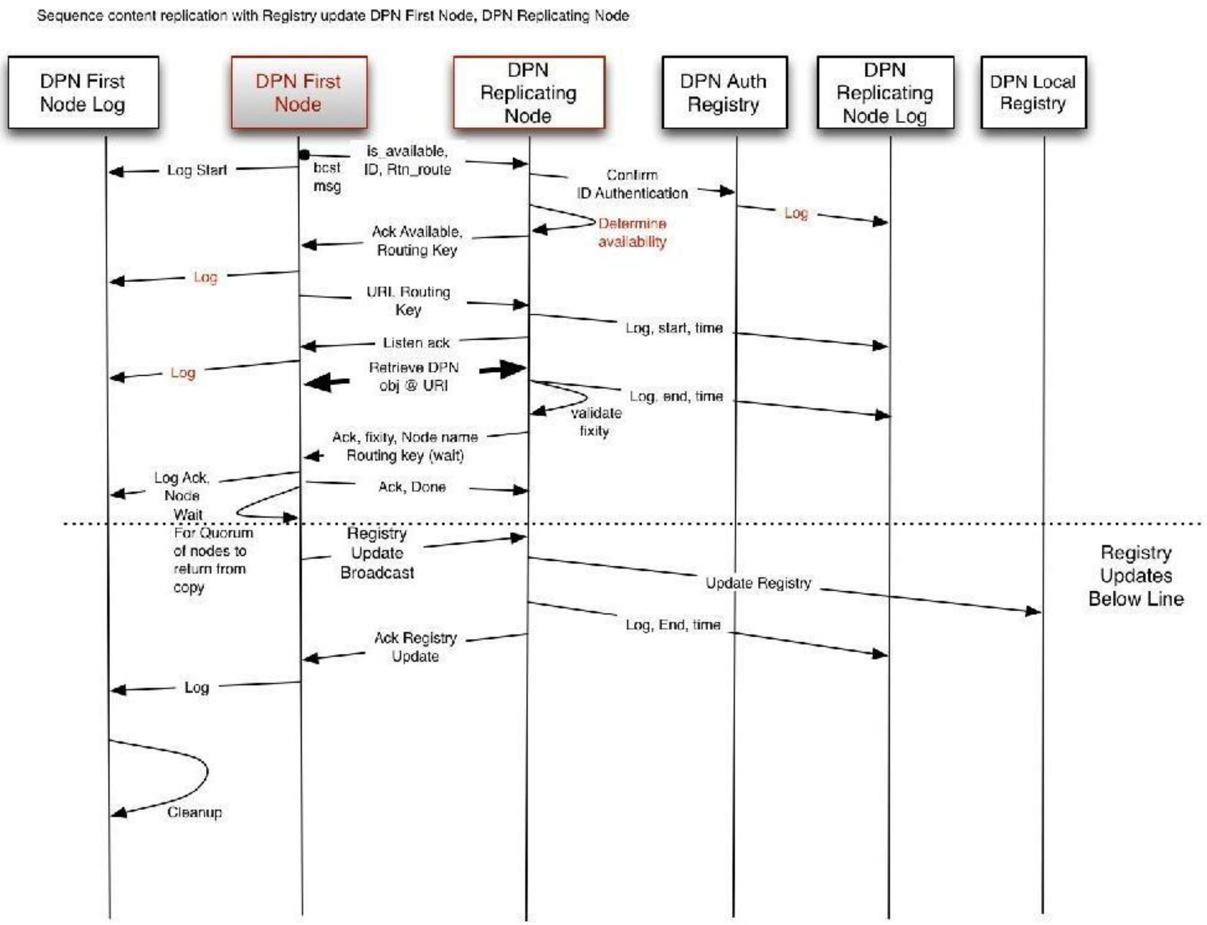
-SDR = First Node
-Node X = Replicating Node
-BC = Broadcast route
-avail? = is available for replication
-A = return route for SDR
-C = return route for SDR
-B = return rout for Node X
-D = return route for replication status to sdr as receiving node

Replication & Registry

- The previous slide showed the first part of a Replication, once a node has copied the content, it must now wait for a message indicating that it must update its registry
- The First Node must also track the successful replications and when enough have completed, issue a Registry Create Entry message to ALL the nodes
- Once the Registry is updated DPN has a copy of the content
- As part of the DPN processes, the DPN messaging protocol must handle partial replication scenarios (e.g. two out of three complete) and incomplete registry updates

Replication With Registry Update and Logging

- A more complete replication entails a few more house keeping steps
- Along with a DPN wide registry update



Sequence content replication with Registry update DPN First Node, DPN Replicating Node

DPN First Node Log

DPN First Node

DPN Replicating Node

DPN Auth Registry

DPN Replicating Node Log

DPN Local Registry

bcst msg

is_available, ID, Rtn_route

Log Start

Confirm ID Authentication

Log

Determine availability

Ack Available, Routing Key

Log

URI, Routing Key

Log, start, time

Listen ack

Log

Retrieve DPN obj @ URI

Log, end, time

validate fixity

Ack, fixity, Node name Routing key (wait)

Log Ack, Node

Ack, Done

Wait For Quorum of nodes to return from copy

Registry Update Broadcast

Update Registry

Registry Updates Below Line

Log, End, time

Ack Registry Update

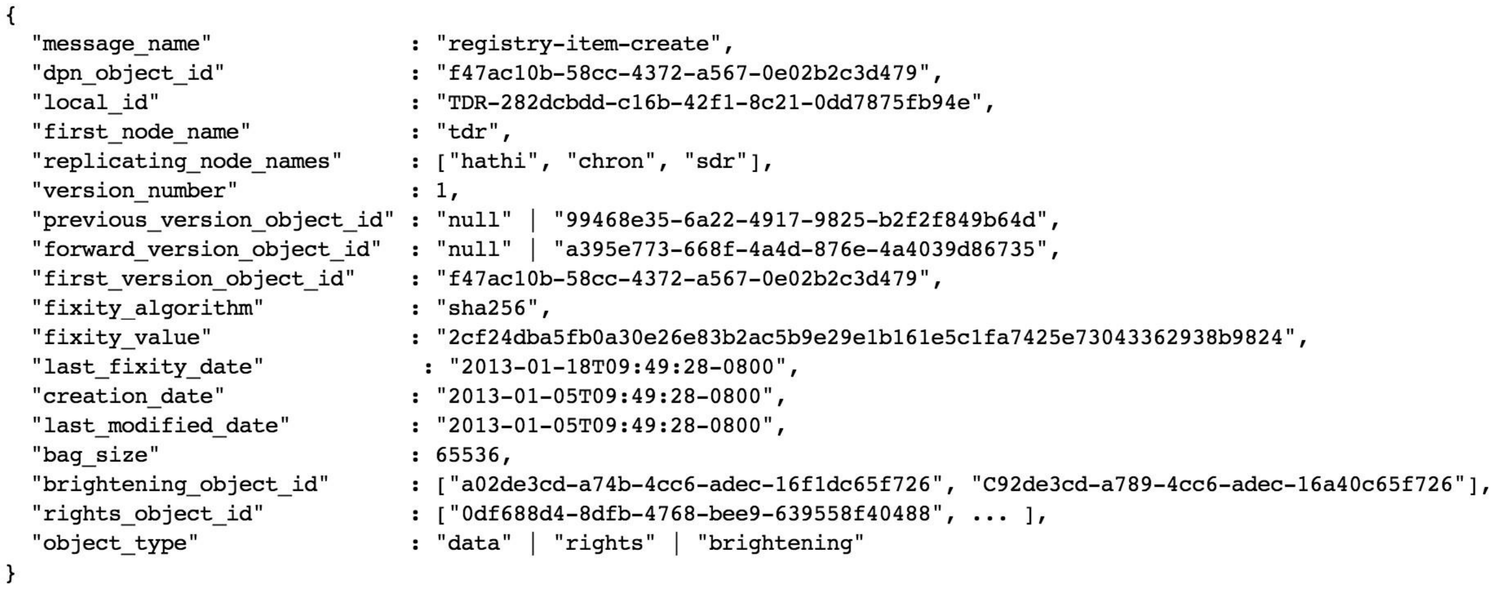
Log

Cleanup

Registry (messages)

- Currently we are investigating messages that will support Registry services
    - Create, read, update, delete
- Delete is a special case, with special handling
- Creation of new Registry entry will be at the request of a First Node.
    - It will only happen after a quorum of correct copies have been made to Replicating Nodes
    - The Registry entry will be updated at ALL nodes
    - Note that this is a distributed environment, so we expect that the registries will be eventually consistent following Brewers theorem

Registry Item Creation Message



```
{
  "message_name"               : "registry-item-create",
  "dpn_object_id"              : "f47ac10b-58cc-4372-a567-0e02b2c3d479",
  "local_id"                   : "TDR-282dcbdd-c16b-42f1-8c21-0dd7875fb94e",
  "first_node_name"            : "tdr",
  "replicating_node_names"     : ["hathi", "chron", "sdr"],
  "version_number"             : 1,
  "previous_version_object_id" : "null" | "99468e35-6a22-4917-9825-b2f2f849b64d",
  "forward_version_object_id"  : "null" | "a395e773-668f-4a4d-876e-4a4039d86735",
  "first_version_object_id"    : "f47ac10b-58cc-4372-a567-0e02b2c3d479",
  "fixity_algorithm"           : "sha256",
  "fixity_value"               : "2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824",
  "last_fixity_date"            : "2013-01-18T09:49:28-0800",
  "creation_date"              : "2013-01-05T09:49:28-0800",
  "last_modified_date"         : "2013-01-05T09:49:28-0800",
  "bag_size"                   : 65536,
  "brightening_object_id"      : ["a02de3cd-a74b-4cc6-adec-16f1dc65f726", "C92de3cd-a789-4cc6-adec-16a40c65f726"],
  "rights_object_id"           : ["0df688d4-8dfb-4768-bee9-639558f40488", ... ],
  "object_type"                : "data" | "rights" | "brightening"
}
```

- Body of the message needs enough information so that all of the Nodes can track a DPN objects origin and replicating nodes
- Not all nodes that have this entry in their Registry will have copies of the content

Federated Registry Synchronization

- At any given time there is a possibility that a node is down, and may not receive Registry messages to create entries, or update entries
- The First Node that issues a create/update can wait and retry, but eventually may give up, i.e. the time to live for the message expires
- To accommodate synchronization, each node will keep a list of registry entries that the node has updated within its own registry
- During a synchronization, each node will exchange synchronization lists and compare against its own list, items missing will show up on other nodes lists and can be recovered

Registry Suspects

- In addition to synchronization lists, nodes will keep track of possible problem registry entries by keeping a Suspect list
- DPN nodes will send Suspect lists entries to the First Nodes responsible for the content/registry validity
- The First Nodes will validate the entries and either issue an ok, delete, or an updated entry for use by the Federation

DPN Packaging - BagIt

- Standard packaging method shared by all nodes
- Minimal, standard bag metadata to enable tracking identity, source and fixity of content bags
- No DPN-wide requirements on descriptive metadata or content structure below the top level bag

DPN Packaging - BagIt

- DPN packages will conform to the BagIt packaging format

- DPN packages may either be
  - serialized (e.g. a single tar)
  - un-serialized (e.g. exploded directory structure)

- DPN packages will conform to a TBD BagIt profile, still under discussion

Proposed DPN Bag Structure

```
<DPN-Object-ID>/
        |  bagit.txt
        |  manifest-sha256.txt
        |  bag-info.txt
        |  tagmanifest-sha256.txt
        \------ data/
                |  [payload files]
        \------ dpn-tags/
                |  dpn-info.txt
                |  dpn-registry.txt
        \------ [optional node tag directories]/
                |  [optional node tag files]
```

DPN Versioning

- Many of the objects held by DPN will benefit from versioning.
  - First, the archival objects will have a way of creating differential versions, as well as a whole copy version (this may be repository dependent)
  - Second, a number of DPN administrative objects will also benefit from versioning. These objects include Brightening information, Rights Information, Agreements

DPN Versioning

- DPN defines versioned objects in a Bag that contains the object and in the Registry entries at each node
- An object is versioned simply by stating that it is a new version and linking to the previous version in the Bag, and by linking to the previous version and putting a forward reference in the previous version in the Registry (a doubly linked list)
- Versioning does not change previous content in any way

DPN Content Replacement

work in progress

Question: Should First Nodes have sole discretion to make content replacement decisions for content they have submitted into DPN or should DPN explicitly state what replacement scenarios are allowed?

Considerations: DPN mission, data integrity, economic viability, existing First Node policies and operations

Example Replacement Scenarios

work in progress

Currently acceptable to everyone:

- replacement as part of routine repository maintenance
- replacement of bogus content submitted into DPN and discovered by the First Node who submits the replacement

Currently only acceptable to some:

- a new OCR pass is made using a new algorithm
- format migration

Replacement Safeguards

work in progress

Potential safeguards for acceptable replacement scenarios include:

- replaceable flag in the registry
- additional control flow validation measures
- delayed deletion of replaced content
    - auto time delay (e.g. 6 months)
    - DPN always retains at least 2 versions
    - require human validation of new content

DPN Threat Model 1

- Goals of the Threat Model
    - Describe common threats addressed by the architecture
    - Identify threats that need to be addressed
    - Provide an ongoing mechanism for risk assessment and design revision
- Threat Model Status
    - Detailed initial document created by sub-team
    - Reviewed by entire tech team
    - Iteration on document will continue

DPN Threat Model 2

- Basic categories
    - Operational and Security threats
- Assume that things will break and nodes will be compromised
- If current architecture addresses the threat, document process
- If threat is not addressed or it is unclear, highlight the threat for consideration

DPN Data Transport

- Used only for copying bags within DPN:
  - Initial Replication
  - Restoring replicas after a failure
- Use widely-supported, easy-to-script transport mechanisms
- Also plan to support high-performance mechanisms where possible
- Confirmation of fixity done outside transport

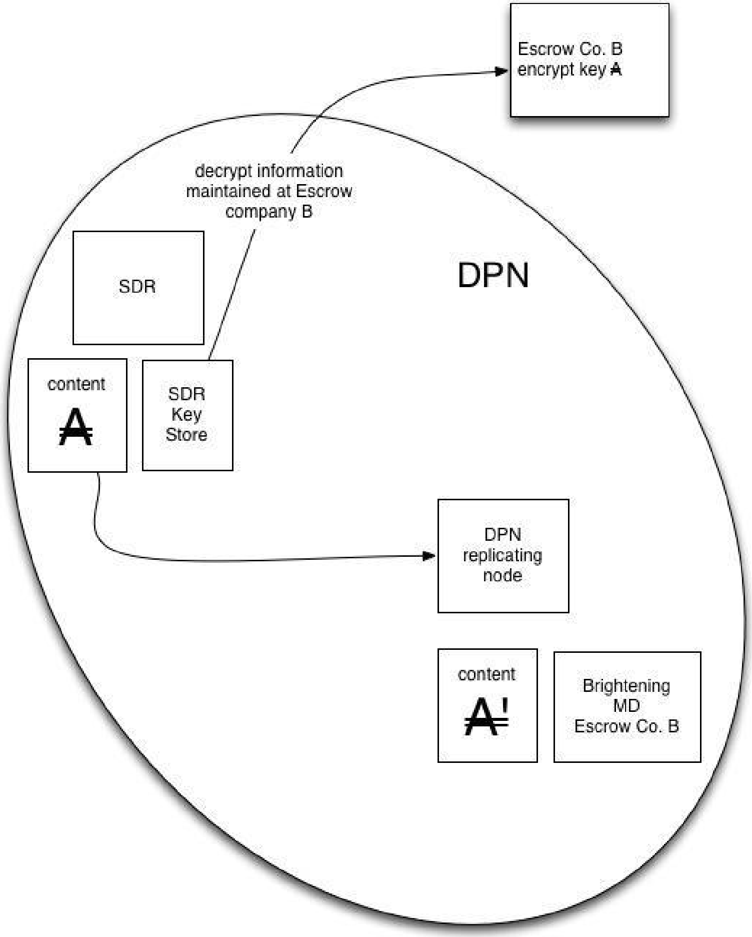
DPN Transport Mechanisms

- Work in progress - not a final list
- HTTPS
  - Simple to use, widely supported
- rsync-over-ssh
  - Ubiquitous on Unix hosts
- GridFTP - More technical complexity, but excellent for long, fat pipes

DPN Encryption

work in progress

- Some content may be encrypted at rest
- Depositors / First Nodes must have confidence that content is secure
- Key escrow to allow content to survive any succession events



Unique Development Paradigm

- Federated environment rather than a single application.
- Heterogeneity as a design principle in DPN means a different implementation at each Node.
- Open Standards vital for interaction between Federated Nodes.
- Heavy dependency on policy agreements shapes the conversation on standards.

Implementation Diversity

- APTrust - Python
- Chronopolis - Java
- HathiTrust - JRuby
- Stanford Digital Repository - Ruby
- University of Texas Data Repository - PHP
- Transfer protocols may vary per Node:
  - HTTPS
  - Rsync
  - Others perhaps

Concurrent Development

- Strong specifications are critical given diversity of implementations.
- GitHub for more social coding, code review tools, and tracking of changes over time.
- Consensus-based decision making by implementation team.
- Healthy debate over details of specifications have had very good results.

Development Challenges

- Diversity of architecture complicates growth of services and refactoring by # nodes.
- Diversity of missions between nodes in the federations make some implementation decisions more difficult to reach.
- Challenge coordinating a geographically diverse team with varying responsibilities and availability.

Advantages

- Diversity of nodes in the federation means we are able to draw on a pool of highly talented people.
- Right people in the right place with the right skills.
- Good interactions on the implementation team. Proactive, productive, healthy conflict can uncover bad assumptions.
- Flexibility of Federation also avoids implementation confiicts that might otherwise occur.

For more information ...

DPN Website:

http://www.dpn.org

DPN Public Wiki:

https://wiki.duraspace.org/display/DPNC/Digital+Preservation+Network

Contacts:

General: inquiry@dpn.org

Steven Morales: steven.morales@dpn.org

DPN

THE DIGITAL PRESERVATION NETWORK