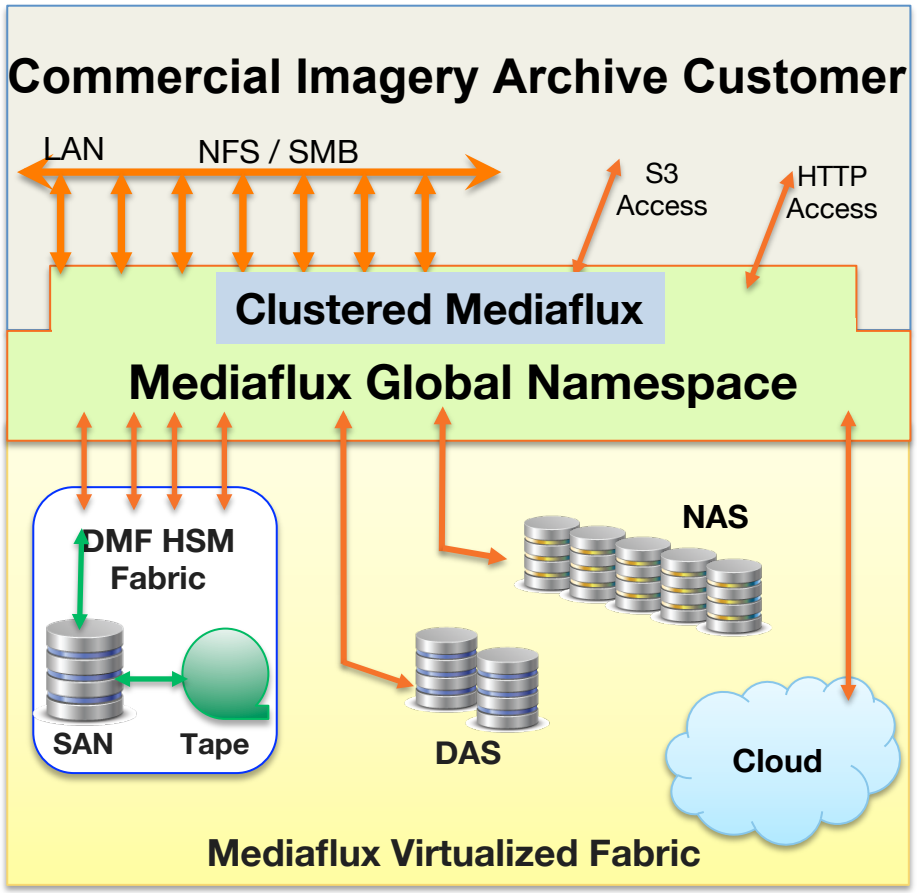

Achieving Zero Data Loss DR
with DMF Integrated Backups
and MediaFlux
Zsolt Ferenczy/SE
zsolt@sgi.com
sgi

Large DMF/MediaFlux System

- Flexibly managing data growth and reducing costs
- Managing >16PB of unique data and growing
- Ingesting >10 TB per hour new data via 12 ClusterFlux nodes on CXFS
- 50 Billion assets growing to greater than 500 Billion in 24 months
- Mediaflux & DMF Integrated Solution
  - Virtualize multiple silos of NAS and DAS storage and use the power of Mediaflux metadata to manage all assets.
  - Enable hybrid of multiple storage types, to accommodate legacy and new storage.
  - Implement metadata-driven Mediaflux Flexpools for policy-based global storage management.



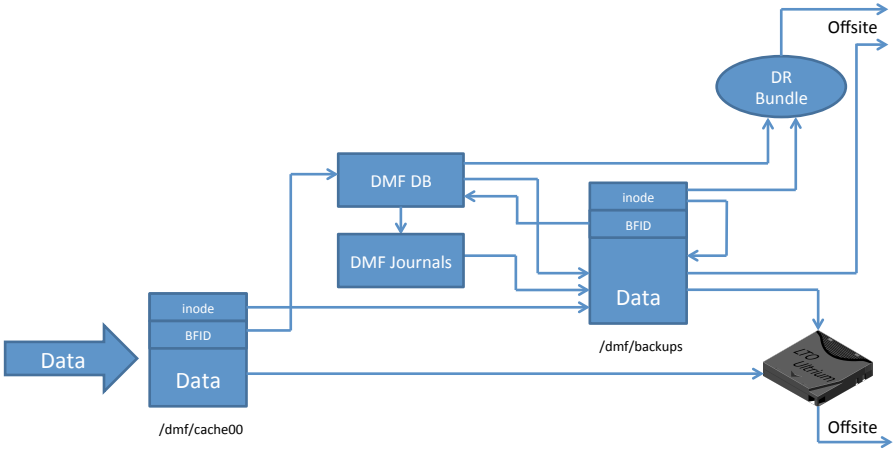
Customer data spread across heterogeneous storage, including DAS, NAS, SAN, Tape, and S3-based Cloud storage

New DMF6 & MediaFlux4 Integration Points

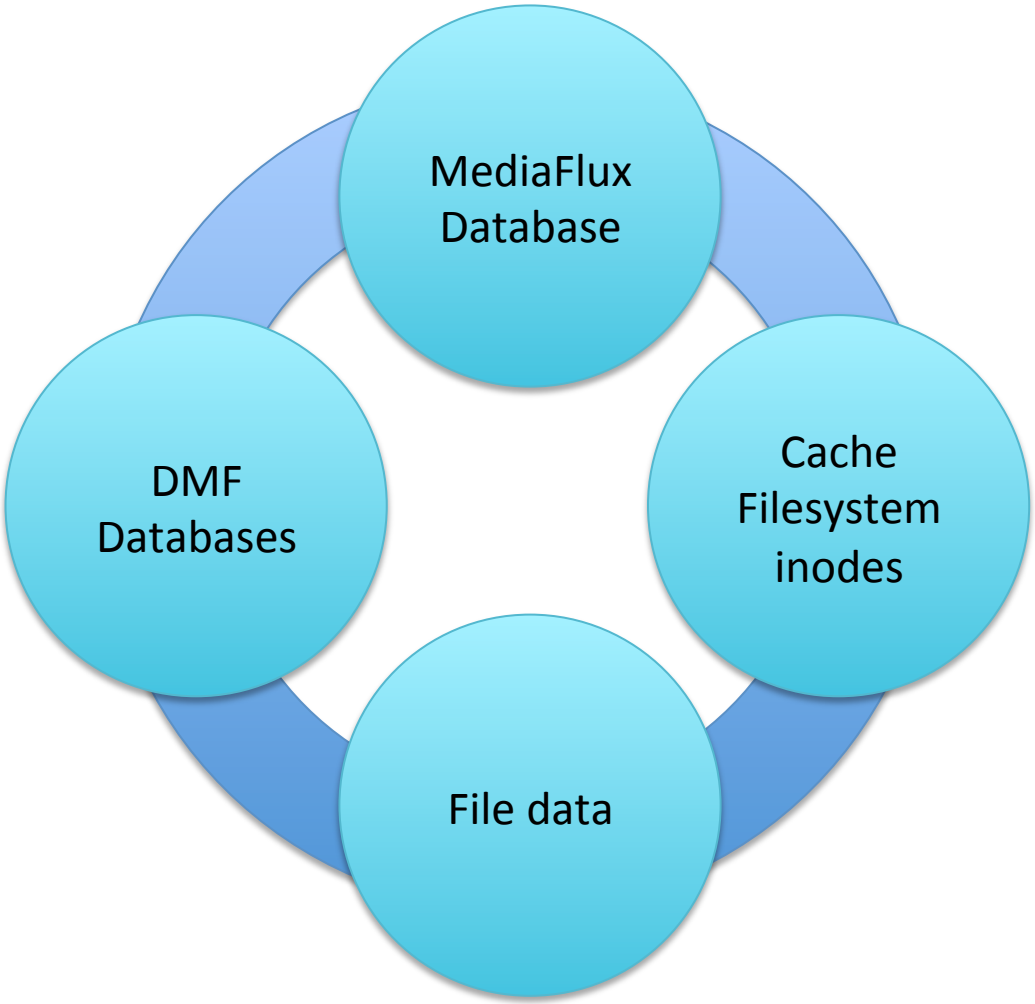
- ClusterFlux scales IO linearly on CXFS
- MF drives DMF directly via dmusrcmd API in a very efficient way
- MF can select a DMF policy by data store
  - dmf.tag.bits.commit
- Optimized recall from DMF
  - Asset prepare from a background queue with large window
  - Uses asynchronous dmget API with callbacks
  - Manage DMF priority level
  - Retry failed recalls
- MF has an interface to dmqview
  - asset.store.dmf.queue.describe
- MF can store DMF BFIDs in XODB
  - Supported by IMMUTABLE_BFIDS
  - Can query an asset by DMF BFID giving giving instant access to the path in the DMF cache (OID path).
  - MF can rebuild the DMF cache filesystem from data solely stored in XODB.
    - asset.content.store.recovery.mapping.describe
- MF can be plugged into DMF Integrated Backups

DMF6 Integrated Backups

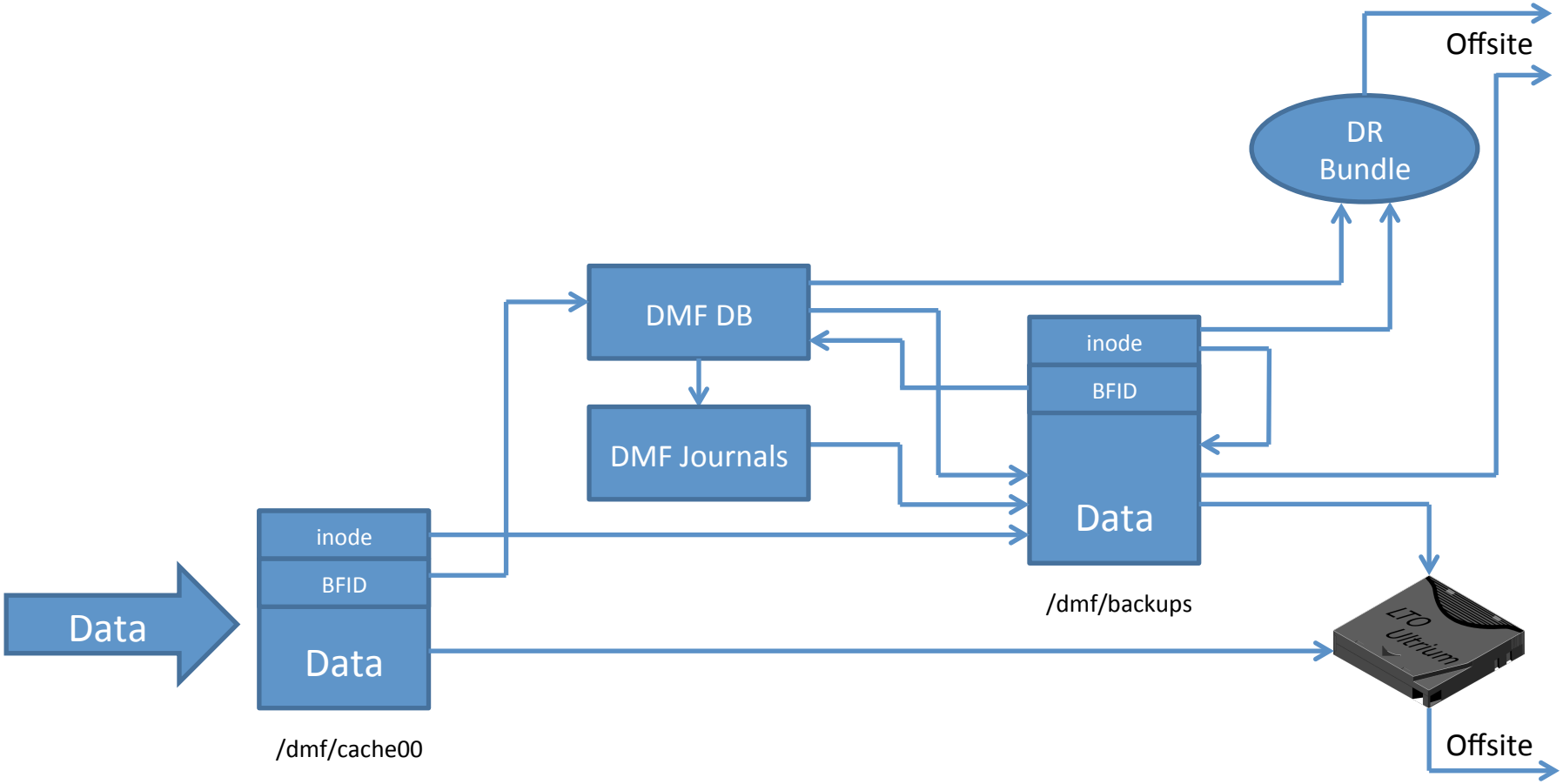
- Extends the current system in use for many years
- Disk to Disk to Tape
- Leverages DMF tape backend
- Can use any DMF MSP
- Saves system configuration
- Dump Groups
- Enables offsite DR
- Site specific customization via scripts
- Can plug-in MediaFlux
- Available starting with DMF 6.4



Four Pillars of Data



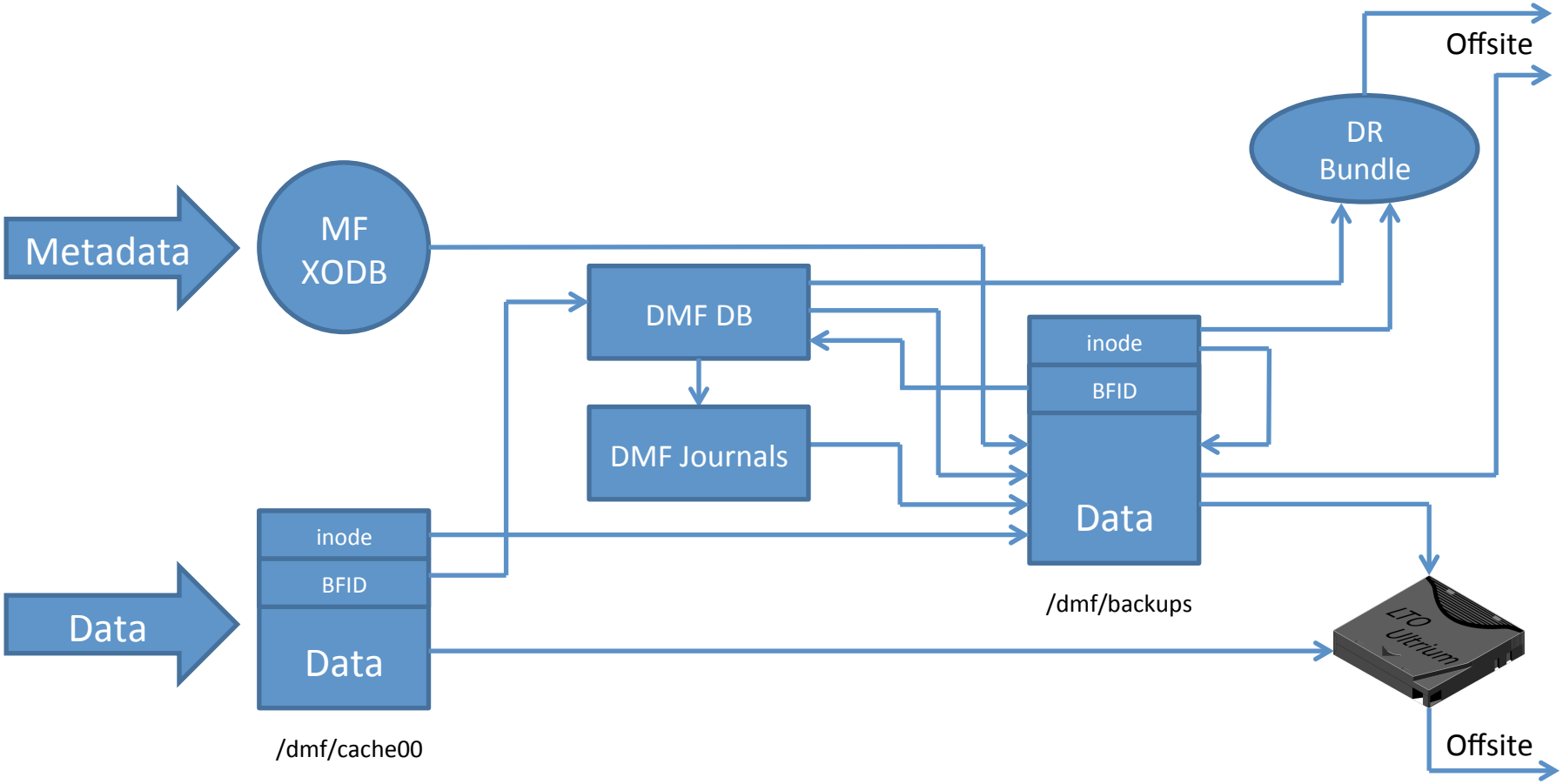
Integrated Backups Detail



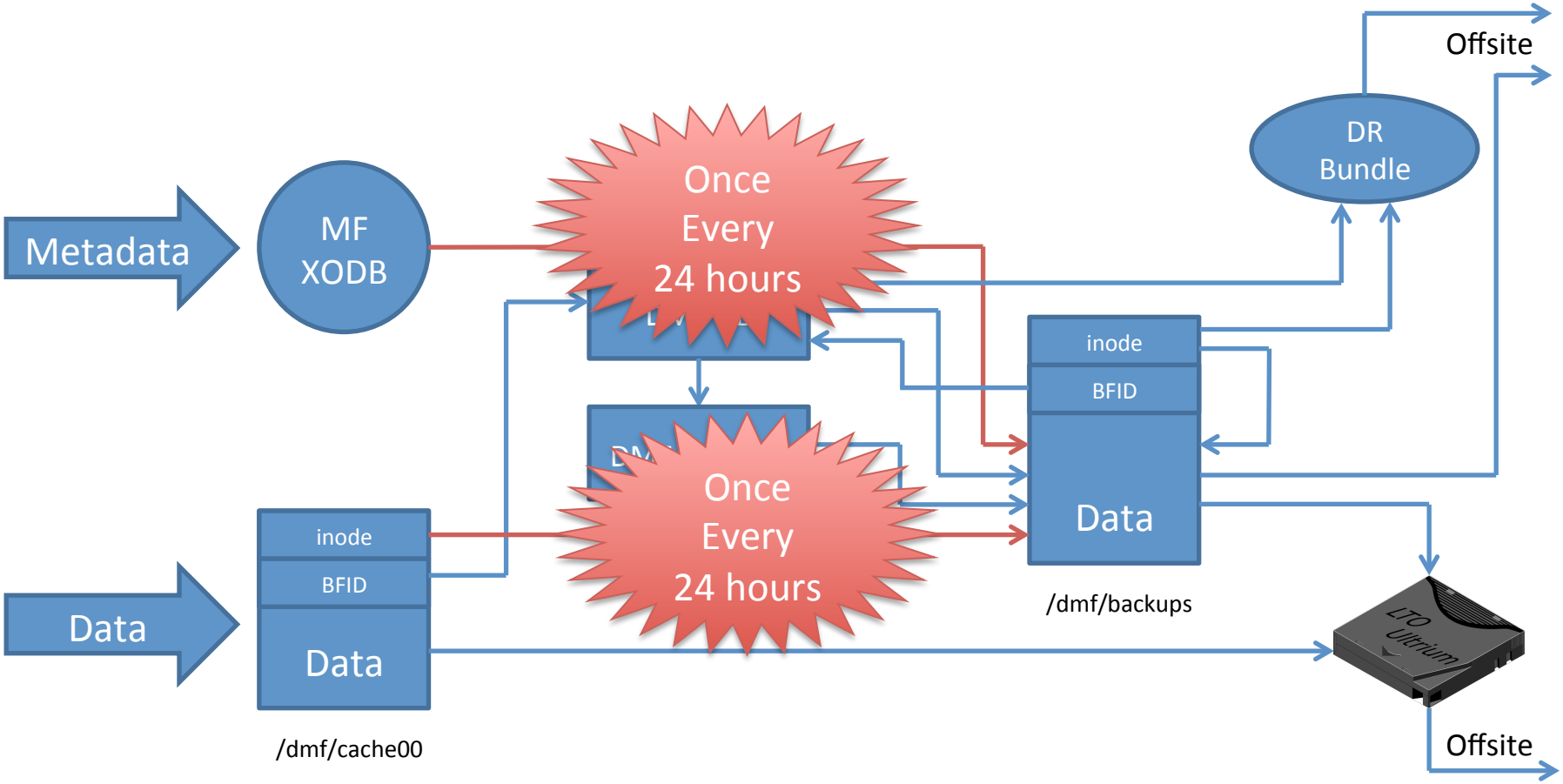
Integrated Backups Detail

- Uses a DMF Managed DUMP_DESTRINATION filesystem
  - Typically mounted as /dmf/backups
- Direct disk to disk xfsdumps
  - Can be multi-stream for reduced dump time
- Backups migrated to any MSP by policy
- Gathers system configuration
- Saves DMF database and xfsdump inventory
- Creates a DR Bundle as a bootstrap to restore the system
- The DR Bundle contains
  - xfsdump of the DUMP_DESTINATION
  - All DMF DB records for files in DUMP_DESTINATION
  - DMF configuration
  - xfsdump inventory
- Any system component can be restored if you have either:
  - The dump files
  - The DR Bundle and access to the DMF MSP media

Integrated Backups with MediaFlux



Integrated Backups with MediaFlux



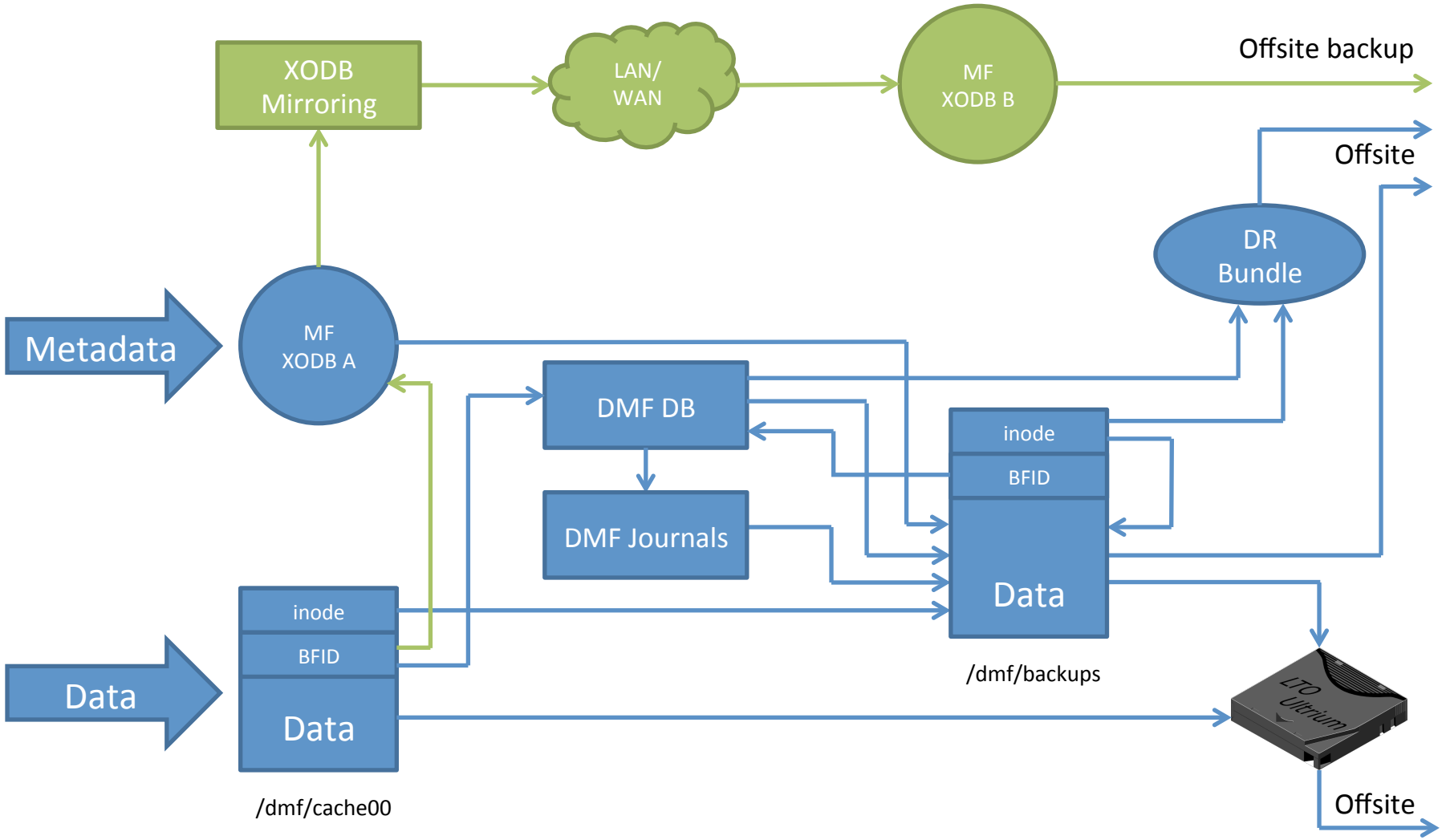
Standard RPO/RTO

Component	RPO	RTO
MediaFlux Database	Several hours to 24 hours	Several minutes to hours; depends on DB size
DMF Database	Zero. Up to 24 hours if both the DB and Journals are lost.	Less then 1 hour to several hours
Cache filesystem inodes	Several hours to 24 hours; O(n) on number of files	Several hours; O(n) on number of files
Data File Contents	Zero to several hours	Zero

Goal: Zero Data Loss

- DMF cache filesystem exposed between inode backups
- MF XODB exposed between backups
- The rest of the components are robust

Zero Data Loss DR



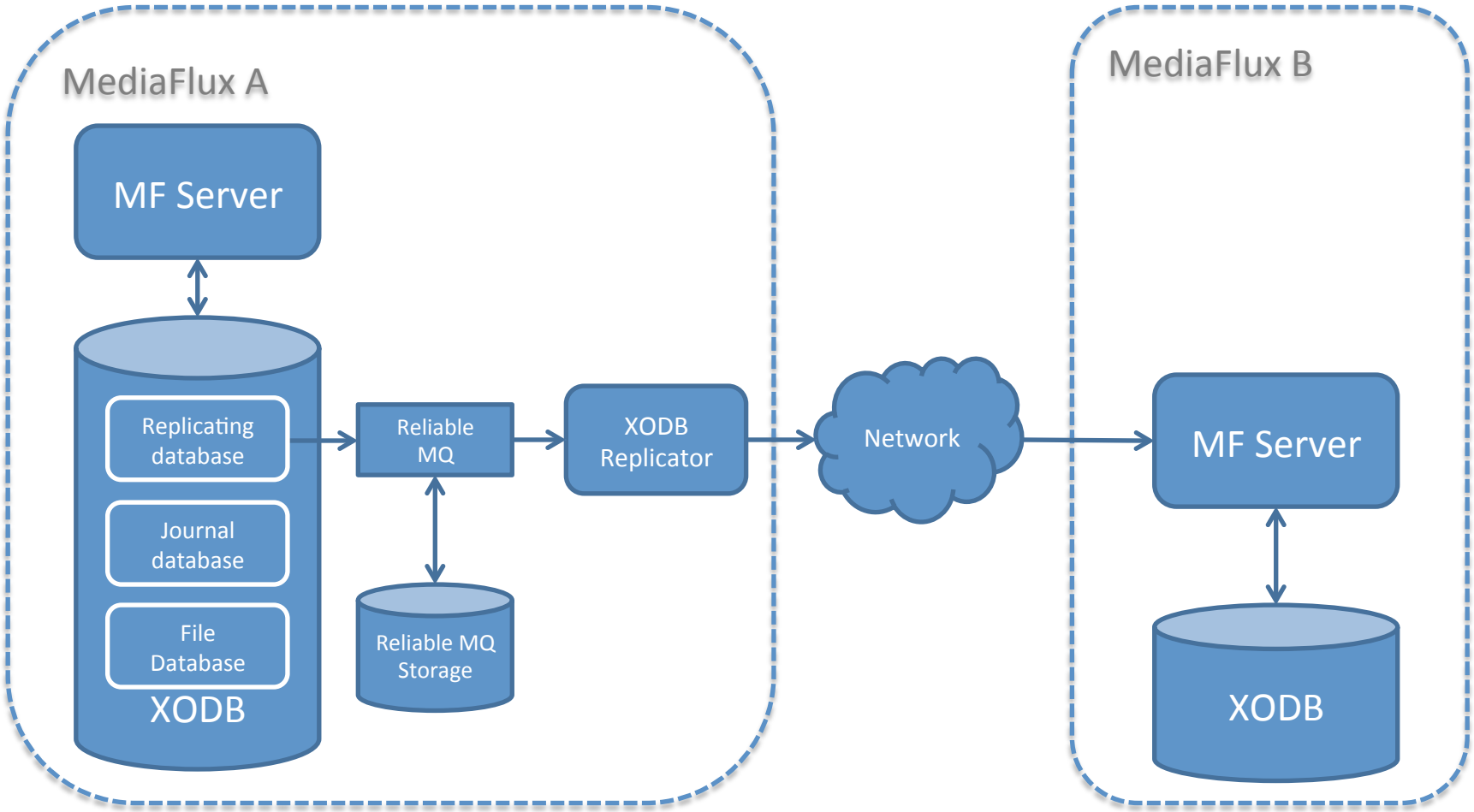
Zero Data Loss DR

- MF harvests the DMF BFID and stores it in XODB as metadata
- XODB mirroring is configured to a second MF instance
  - Second instance may be local or remote (WAN)
- Database journaling allows hot backups of the primary XODB
- XODB backups written into the DMF DUMP_DESTINATION
- DMF and MF provide services and tools to recreate inodes from XODB metadata

MediaFlux BFID Harvesting

- MF calls sync dmtag followed by async dmput API calls at transaction commit
- These are put into a Q in the event DMF is unavailable
- When MF receives the final reply from the DMF API that the file is committed to back end storage, the BFID is recorded in XODB and the “committed to tape” flag is set true
- Relies on DMF 6.3 IMMUTABLE_BFIDS feature
  - DMF will not change the BFID of a file once it has been assigned
  - Some commands, like dmunput and dmmvtree, are not available for filesystem which set IMMUTABLE_BFIDS

MediaFlux XODB Mirroring



Enhanced RPO/RTO

Component	RPO	RTO
MediaFlux Database	Zero. Some number of transactions if both the XODB and the RMQ store are lost.	Several minutes to hours; depends on DB size
DMF Database	Zero. Up to 24 hours if both the DB and Journals are lost.	Less then 1 hour to several hours
Cache filesystem inodes	Zero. (For files that have completed migration)	Zero. Several hours to fully repopulate; O(n) on number of files.
Data File Contents	Zero to several hours	Zero

Recovery Procedure

1. The DMF system is recovered from the DR Bundle in the case of a complete site failure
2. The DMF managed filesystem is restored from the inode backups
3. The MF asset.content.store.recovery.mapping.describe service is used to produce a list of asset BFIDs missing from the filesystem
4. The list is passed to the dmrestore command to regenerate inodes not contained in the backups

DEMO