Hewlett Packard Enterprise

Moving from DMF6 to DMF7

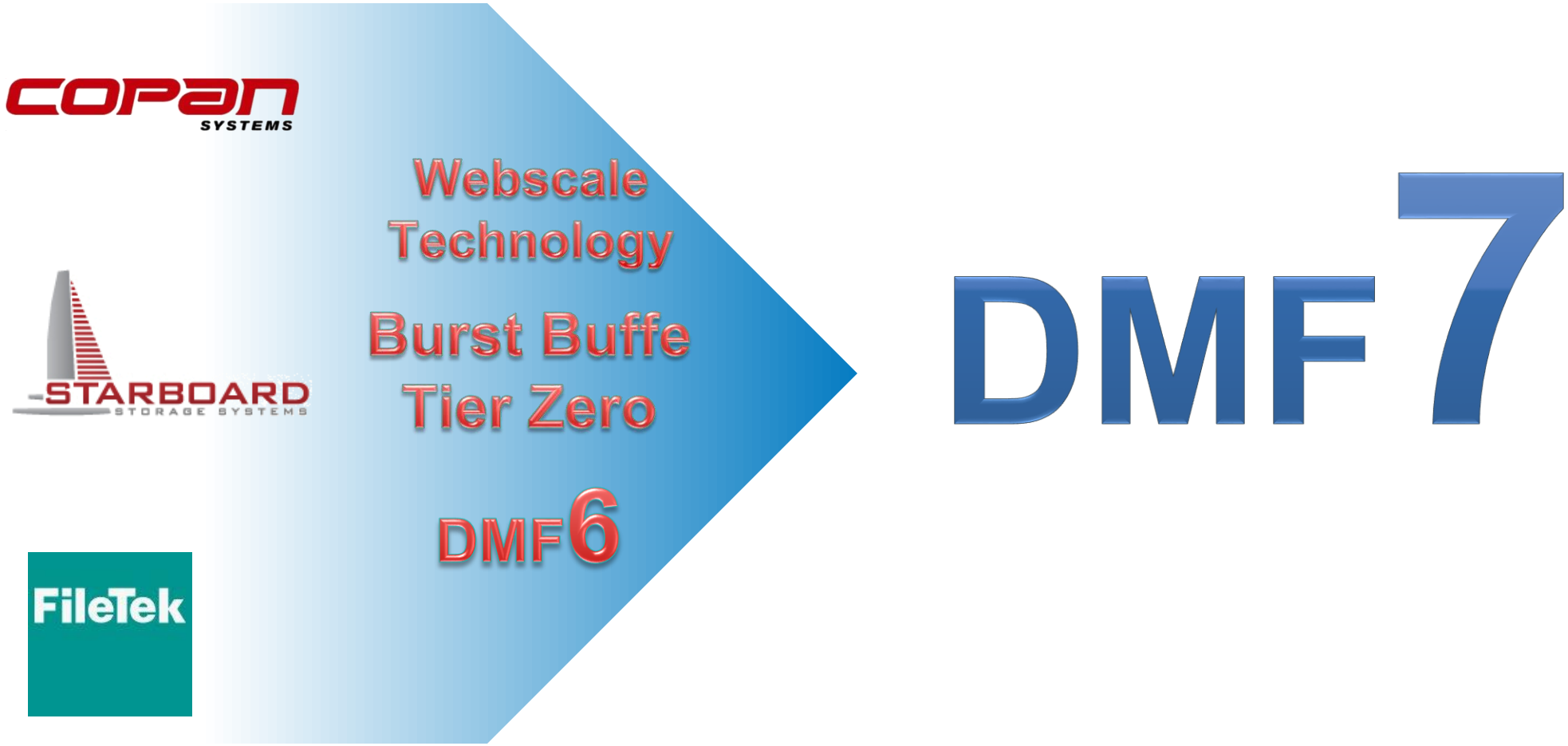
Gerald Hofer
hofer@sgi.com
Zsolt Ferenczy / SE
zsolt.ferenczy@hpe.com

HPC (Compute/Storage) Roadmap

Forward-Looking Statements

This document contains forward looking statements regarding future operations, product development, product capabilities and availability dates. This information is subject to substantial uncertainties and is subject to change at any time without prior notification. Statements contained in this document concerning these matters only reflect Hewlett-Packard Enterprise's predictions and / or expectations as of the date of this document and actual results and future plans of Hewlett-Packard Enterprise may differ significantly as a result of, among other things, changes in product strategy resulting from technological, internal corporate, market and other changes. This is not a commitment to deliver any material, code or functionality and should not be relied upon in making purchasing decisions.

DMF6 to DMF7 | **Why?** Influx of New Ideas



Hewlett Packard Enterprise

DMF6 to DMF7 | **Why?** Drivers for Change

- Highly scalable, distributed database for metadata
  - Single database for all metadata
  - Objects reference able outside of any given filesystem
  - Capture filesystem state changes in real time
- Data Policy engine in addition to Space Management
  - Integration with Job Schedulers and Workflow Managers
- New storage technologies are changing the landscape
  - Flash/SSD
  - Cloud and Object Storage
- Backward compatibility
  - There is close to Exabyte under management
  - Many installations depend on POSIX access

DMF6 to DMF7 | **New Architecture** How to Get There

Replace the DMF database

- R & D underway for several years
- Current shipping DMF has abstracted the database layer
- Use a hyper-scalable DB engine capable of handling 100s of billions of objects
- Bring the namespace into the DMF database

Move from a scan based model to an event and DB driven model

- Listen to event stream from the managed filesystem
- Policy decisions made in the database not in the filesystem
- Make DMF independent from front end filesystem
- Unified DMF policy engine for all filesystem types

DMF6 to DMF7 | **New Architecture** DMF7 Change Log

– Use DMAPI events to drive a XFS change log

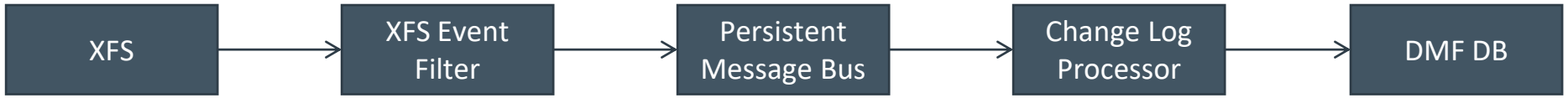
– Implement a DMAPI close event

– All events enabled on managed filesystems

– Events not needed for HSM activity are asynchronous

– Removes the need to scan the filesystem to drive the policy engine

– Removes the need to xfsdump the filesystem to preserve the namespace



DMF6 to DMF7 | **New Architecture** DMF7 Database

- Leverage open-source hyper-scale technology
- Design for 100s of billions of objects
- Store metadata and namespace inside DMF
- Re-create the namespace without xfsdump
- Enable new workflows and policy types
  - Tier “up”
  - Versioning
  - Policy by namespace
  - User defined metadata

DMF6 to DMF7 | **New Architecture** DMF to Drive HPC Workflows

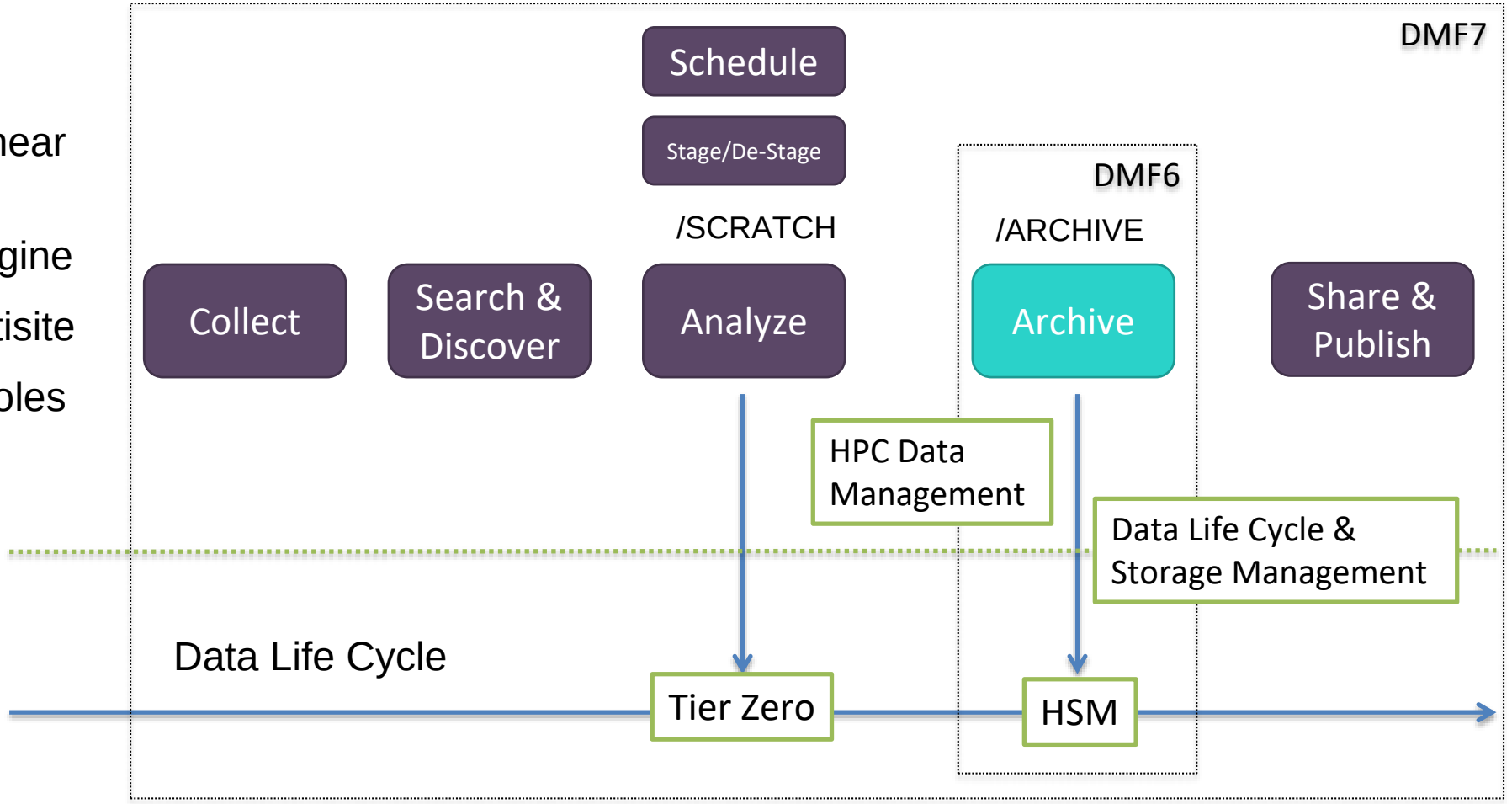
– Integration with Tier Zero: up-tiering, staging & offloading capabilities

– Scalable metadata in near real-time & data sets

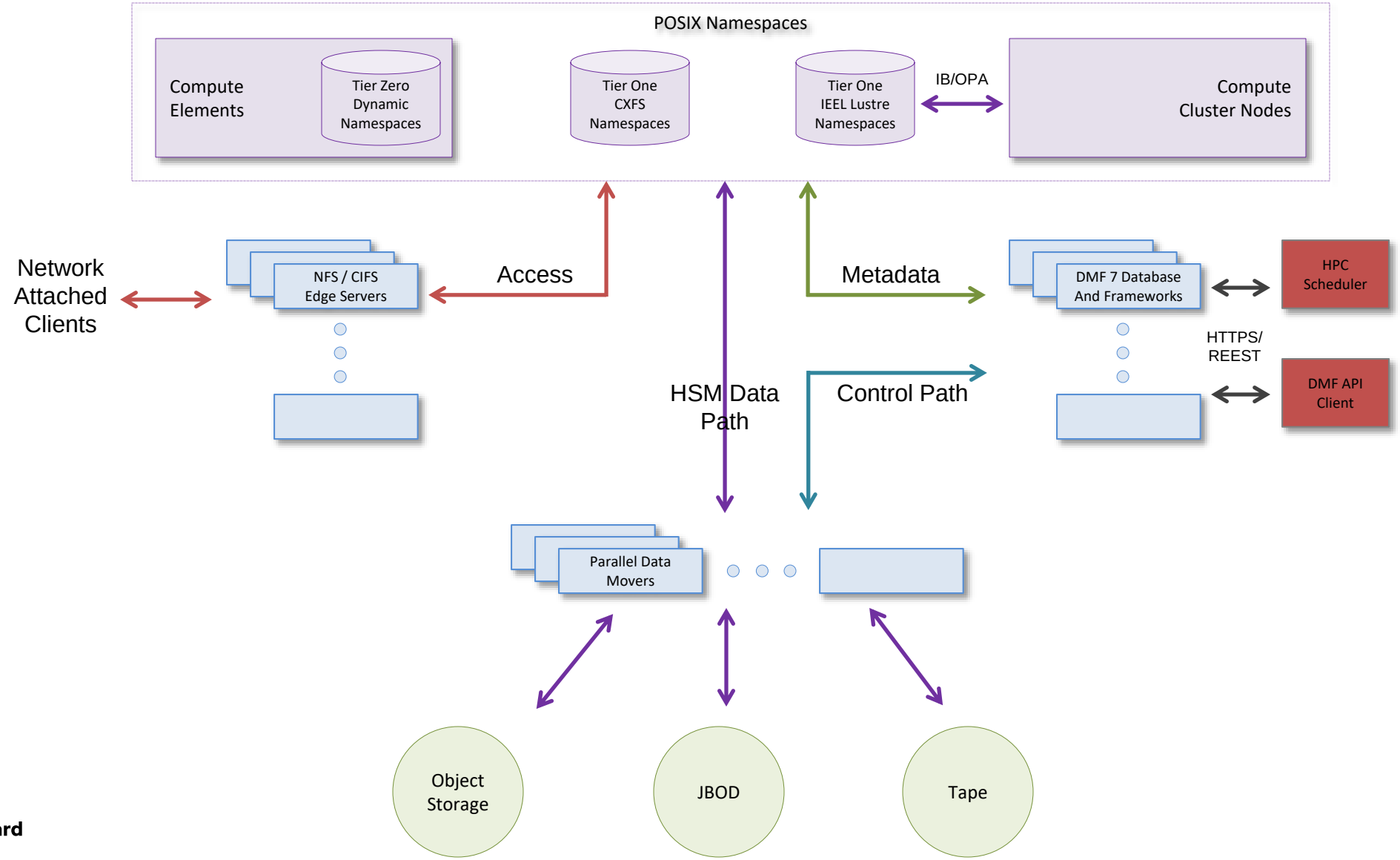
– Flexible data policy engine

– Data replication & multisite

– User management & roles



DMF6 to DMF7 | **New Architecture** DMF7 Reference Diagram



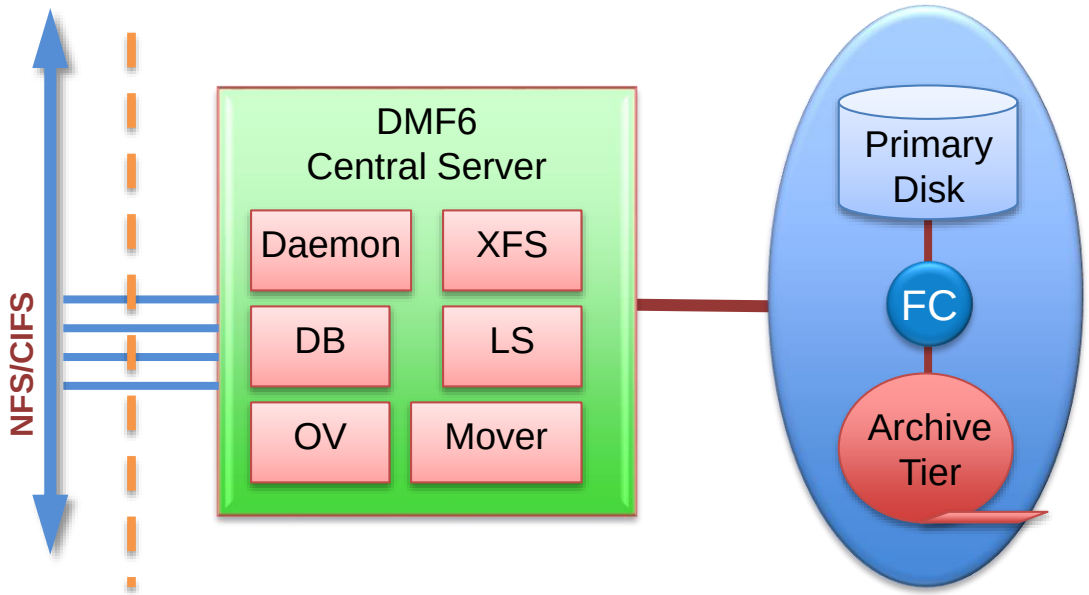
DMF6 to DMF7 | **Cluster** DMF7 Database and Frameworks

- DMF7 scales out and distributes and front end
- A minimum of 3 nodes are needed for the DMF7 DB cluster
- The size of the cluster is scaled out as the managed object count and filesystem transaction rates increase
- This cluster is added to an existing Basic or Parallel DMF system
  - Existing DMF Server, CXFS MDS, PDMO nodes, and Edge Servers continue to be used
- A high speed network interconnects the filesystem servers with the DB cluster
  - The DB cluster does not need access to the managed filesystems

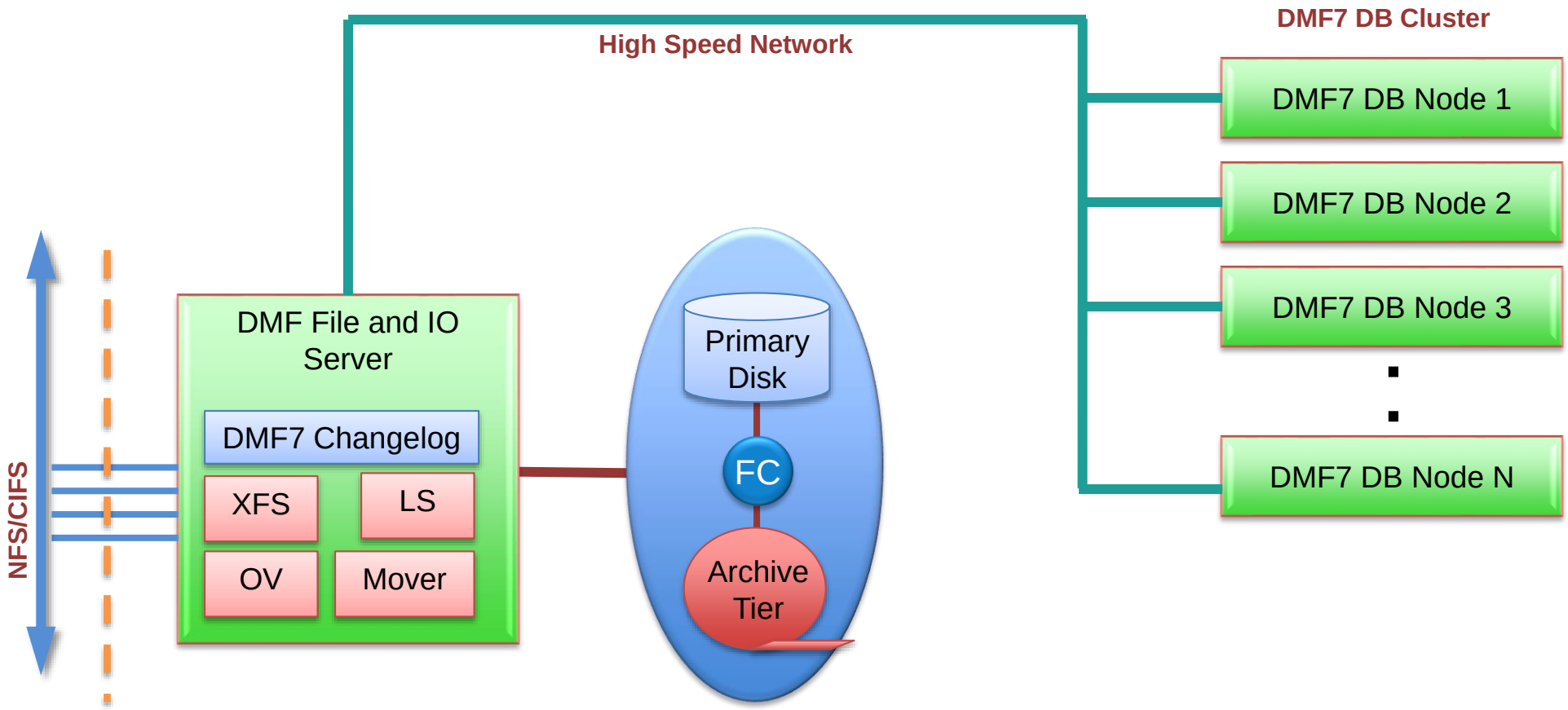
DB Node Minimum Specification:

- High clock CPU
- 128GB of memory
- 4TB of internal SSD

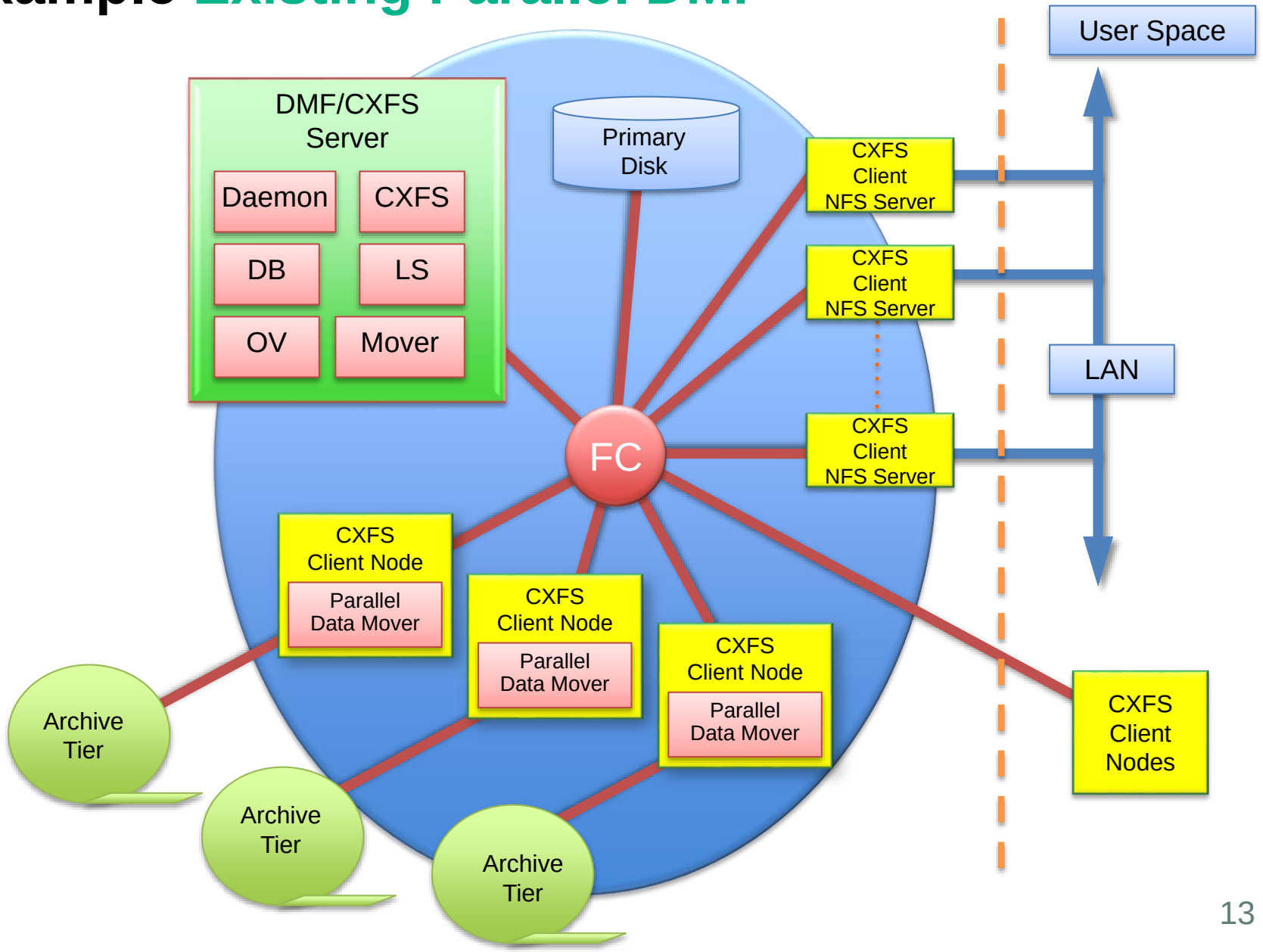
DMF6 to DMF7 | **Example** Existing Basic DMF



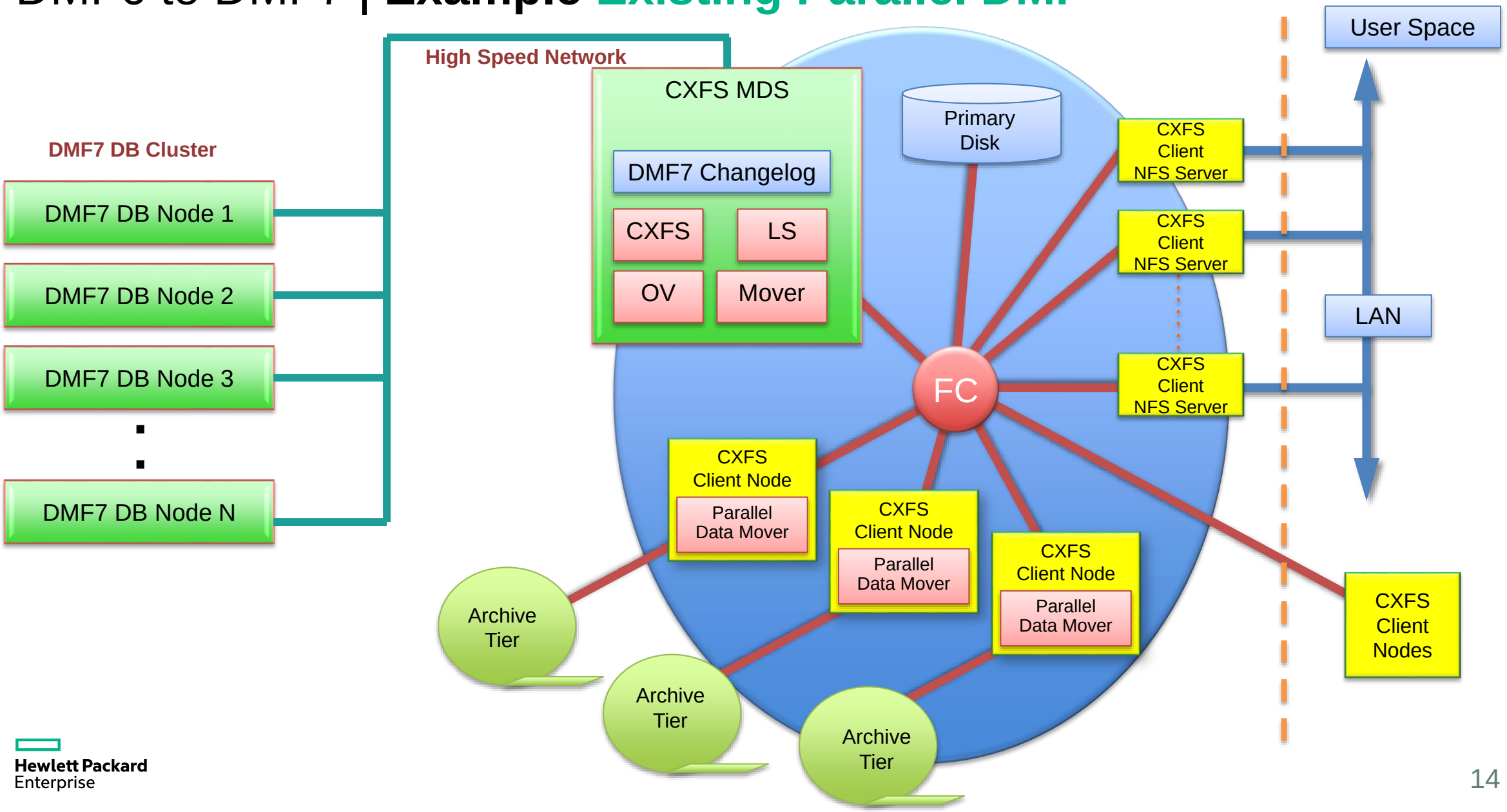
DMF6 to DMF7 | **Example** Existing Basic DMF



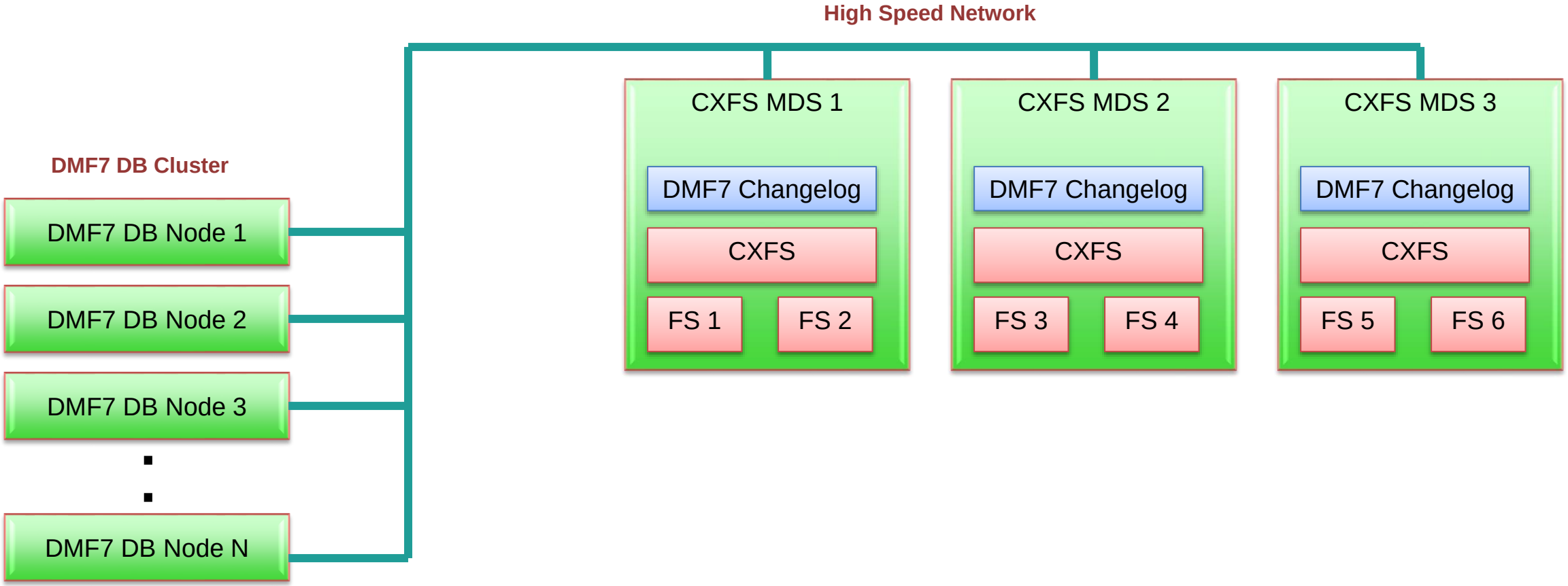
DMF6 to DMF7 | **Example** Existing Parallel DMF



DMF6 to DMF7 | **Example** Existing Parallel DMF



DMF6 to DMF7 | **Example** DMF7 Parallel Front End



DMF6 to DMF7 | **Prerequisites** for Converting to DMF 7.1

- Running DMF 6.7,
- No database errors in dmaudit or verifymsp
- External backup of the database and file systems
- All DMF managed filesystems are XFS
- Using only the OpenVault mounting service
  - TMF will not be available with DMF7
- All migrated data stored only in a Library Server
  - Tape, JBFS (ZWS), or MAID VGs
- DMF not controlled by an external application via the dmusrcmd API
  - Arcitecta MediaFlux, Drastic MediaReactor, etc

DMF6 to DMF7 | **Conversion Outline**

- Upgrade base OS
- Upgrade the ISSP software stack
- Convert filesystems
- Convert the database
- Convert dmf.conf

DMF6 to DMF7 | **Conversion** OS and Software Upgrade

- ISSP 4.0/DMF 7.0 servers will be available on RHEL 7.3 and SLES12 SP2
  - DMF 7.1 and future versions will follow the RedHat and SLES release schedules
- DMF7 software packaging will be different from DMF6
  - Expectation is to remove the ISSP 3.X software and then install the ISSP 4.X software after the base OS has been upgraded
- ISSP 4.0 changes SGI Enhanced XFS allowing it to co-exist with the OS vendor provided XFS on the same system.
  - The SGI XFS filesystem type will change to “EXFS”
  - Existing DMF managed filesystems will need to be updated to EXFS type before they can be used with DMF7
  - This conversion only changes the filesystem type in the super block
  - Tools will be provided to change filesystems between XFS and EXFS
  - EXFS remains on-disk format compatible with community v4 XFS
- Obtain CXFS8 and DMF7 licenses keys
  - Discussions are underway to determine if ISSP 4.0 can be moved to HPE “AutoPass” licensing or continue to use the current SGI “LK” licensing

DMF6 to DMF7 | **Conversion** Filesystem and Daemon DB

- The DMF7 Object DB is a combination of the DMF6 daemon DB and the metadata stored in the managed filesystem
- The Object Database has to be created as part of the conversion process
- A filesystem scan is needed to extract the name space
- All extended attributes stored in the filesystem will be read and stored in the object database
- The conversion will extract data stored in the legacy DMF6 attribute
  - dmtag value, BFID, managed regions, state, etc
- The legacy DMF attribute is not used by DMF7

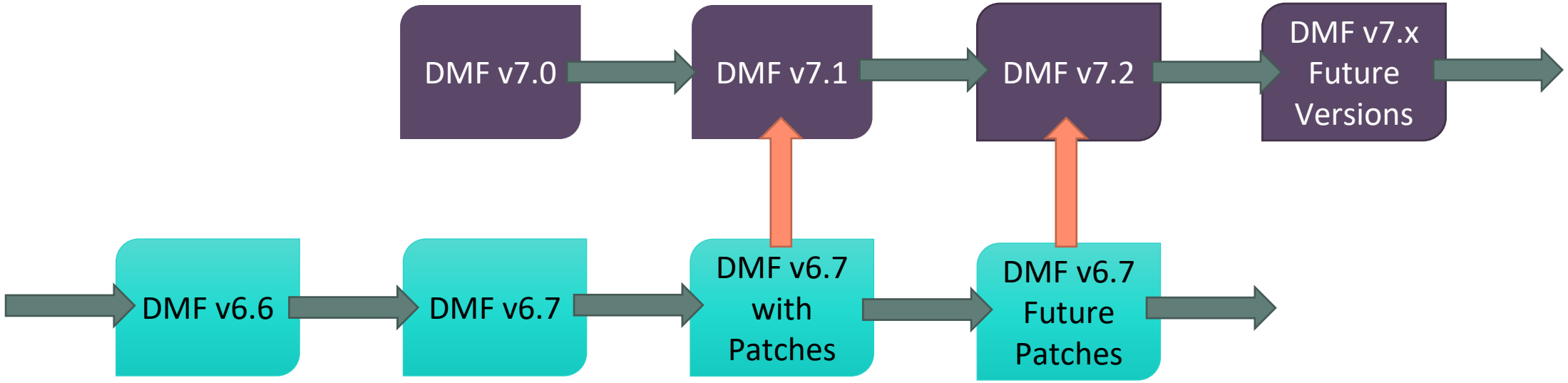
DMF6 to DMF7 | **Conversion** **Library Server DB**

- The LS CAT and VOL databases will be sorted directly in the DMF7 database
- Conversion will extract the RAIMA DB records and store them in the DMF7 DB using native tables
- Links between the Object tables and the Library Server tables tables will be created
- This process has no affect on any data stored by DMF

DMF6 to DMF7 | **Conversion** Configuration and Policy

- DMF7 stores configuration in the database
- The configuration used by the Library Server is still under discussion
  - May be kept in dmf.conf format or moved to the DMF7 database (ZooKeeper) for the 7.1 release
  - The end goal is for all configuration to be stored in the database
- The DMF7 policy engine is a superset of DMF6 capabilities
  - Syntax is different from DMF6
  - Existing policies will be converted to DMF7 syntax
- OpenVault configuration remains unchanged

DMF6 to DMF7 | **Roadmap** Getting to DMF7



Hewlett Packard
Enterprise

Thank you