Enstore and The Farm

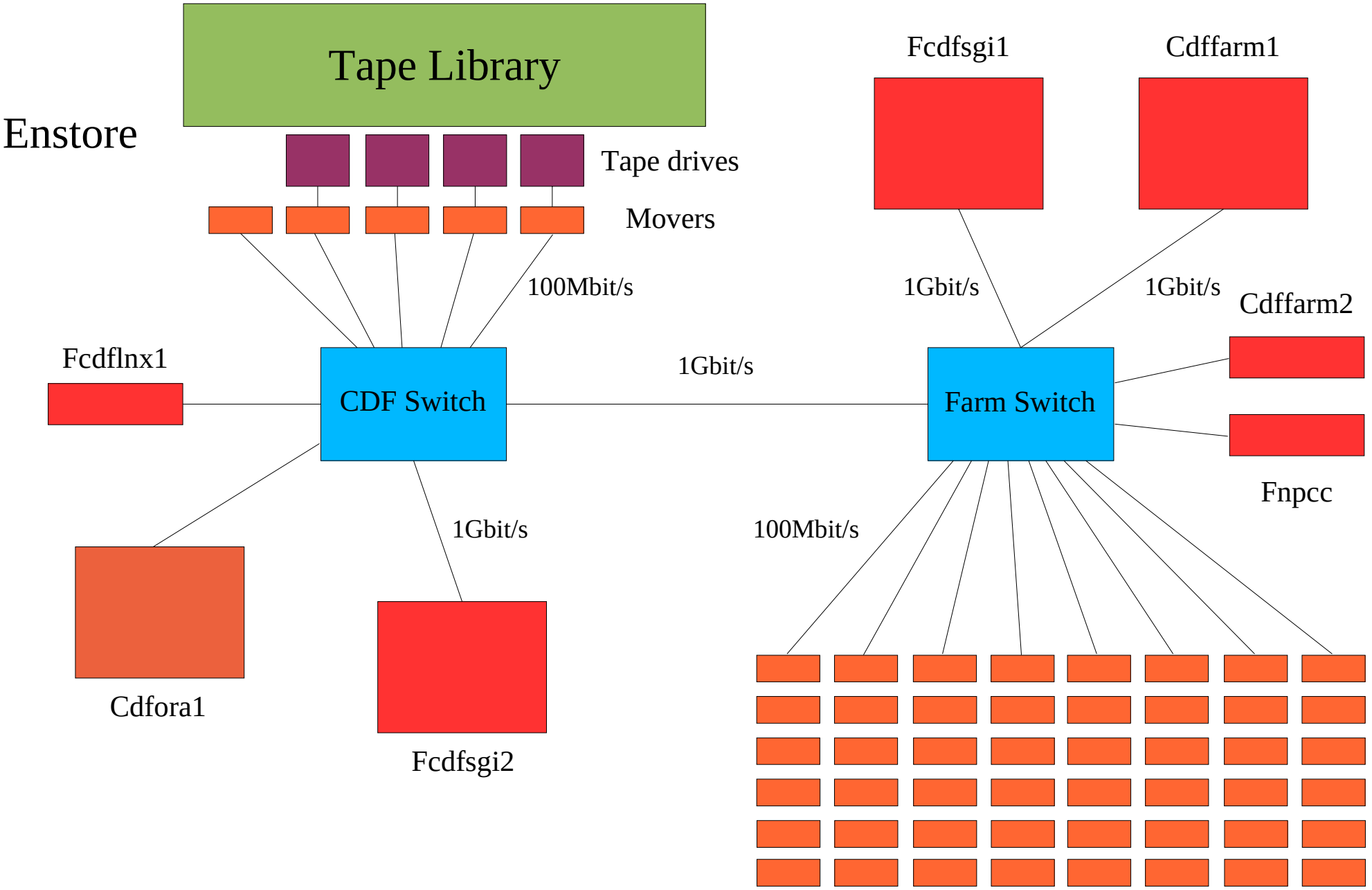
Miroslav Šiket
for the Farm Group

December 12th, Fermilab

Enstore

- Enstore Provides
  - Direct access to tapes over IP
  - High Speed - using multiple movers (single tape at 10/10MB/s)
  - Not attached to any particular I/O node
  - Caching and optimizing internally requests
  - Quasi sequential access on file level
  - Mapping between tapes and tape families
  - Priority based management for the tape drives

Proposed Scheme



Accessing the Tapes

- Farm Input
  - Client Worker Node requests a file, which is delivered directly to the client
  - Need several (min 2) tape driver to achieve 20MB/s throughput
  - No need to cache it somewhere on a disk
  - New requests are queued and organised inside Enstore for optimization
  - Works fine with many streams at the same time or with many requests (more than 1 tape)

Accessing the Tapes cont.

- Farm Output
  - Concatenation is done at Worker Nodes
  - Staging the data to the tapes is done in parallel from concatenation worker nodes
  - Is organised into the tape families by Enstore accordingly
  - Delivers high throughput - setting up priorities to ensure that at least 2 tapes are available for writing will deliver requested throuhput

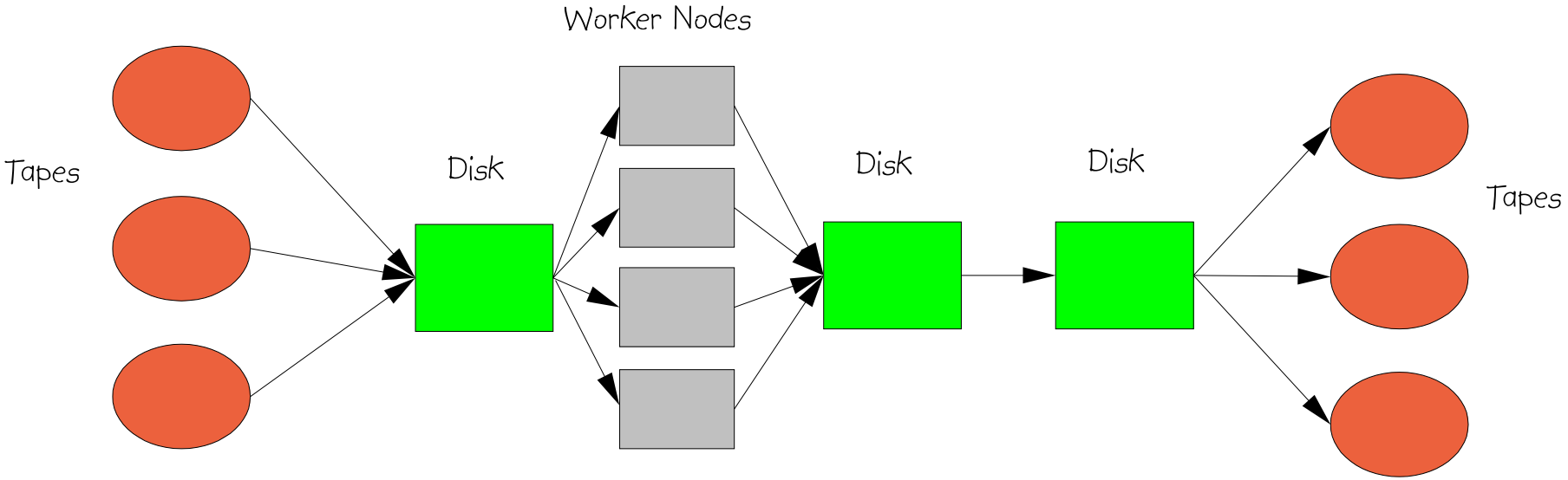
Impact on DFC

- Farm Input
  - Farm will continue to access DFC in the same way, in case of a file based approach even simpler (internally)
- Farm Output
  - 1. Farm will either move data to tape in a file based approach and in DFC will be only a logical structure to encompass filesets (post mortem approach)
  - 2. Farm will create filesets from the concatenation units – about 10GB – that will result in filesets stretching over tapes (deterministic approach)
  - 3. Filesets would become tape equivalent (post mortem extended approach)

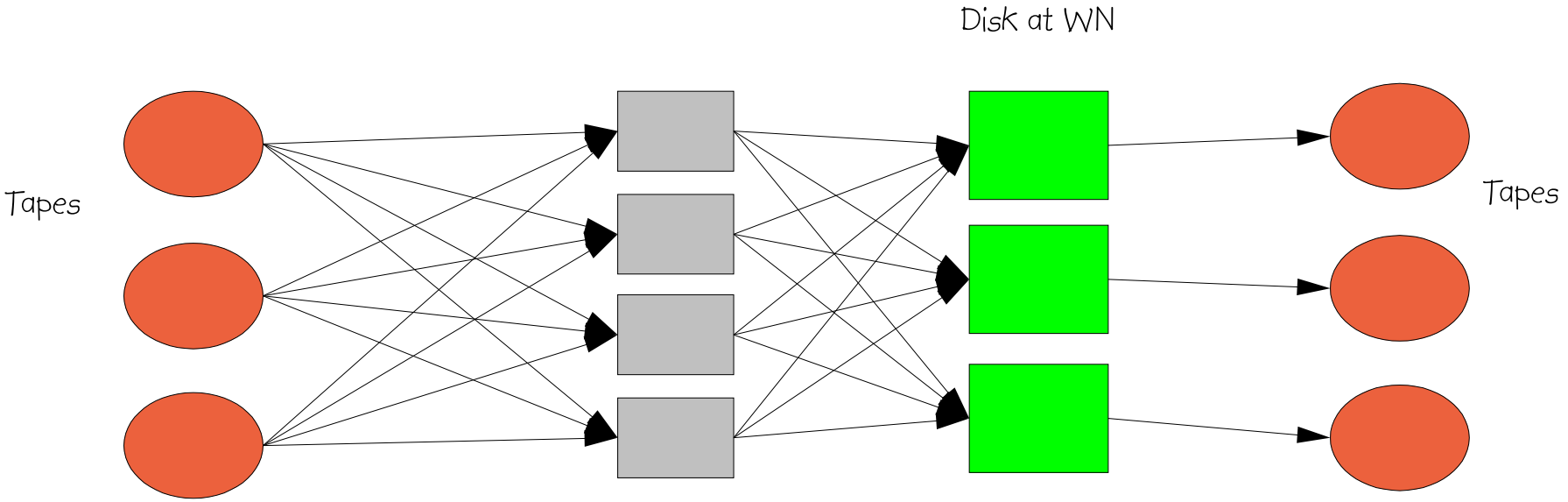
Minimal Upgrade Solution

- Upgrade DIM at fcdfsgi1 to work with Enstore
  - Advantage of this solution is that the Farm does not have to change its software – would continue to access the input data in the same way as before
  - Output would be written back to fcdfsgi1 as before, but from the worker nodes – requires stager locally on the concatenation worker nodes
  - In either case fcdfsgi1 would serve as a front-end to the Enstore and would provide data to farm via DIM managed disk space and would organize filesets on the output as well
  - This solution scales with increasing number of streams purely – requires large cache disk space and will not deliver higher throuhput in the future, on the other hand provides data cache to the Farm and therefore fast access when needed

Current approach



Proposed approach



Stageing in process

Concatenation Process

Stageing out process

Proposed vs. Minimal

- Proposed solution requires some work on the Farm, but is scalable and oriented for the longer term and reduces two bottlenecks in the processing chain
- Minimal solution is available almost immediately, but requires fcdfsgi1 to do the data movement between Enstore and Farm + large disk cache
- In the case of an proposed solution an upgrade of the tape drives to a higher throughput one needs to upgrade the network accordingly

Tentative Schedule

- Get concatenation on the worker nodes with Kahuna/DIM in place by the mid of January (already working on that with DH)
- At the end of January test reading/writing data from/to Enstore
- In February implement/test interaction between Enstore and the Farm
  - Includes different approaches:
    - Direct data transfers from/to Enstore from the Worker Nodes
    - Caching at Fcdfsgi1 with Kahuna/DIM
    - Caching at dfarm (disk farm – using disks at the Farm Worker Nodes)
- End of February start using Enstore for writing output and/or reading raw data as well, depending on the status of the previous steps