Integrating high speed detectors at Diamond

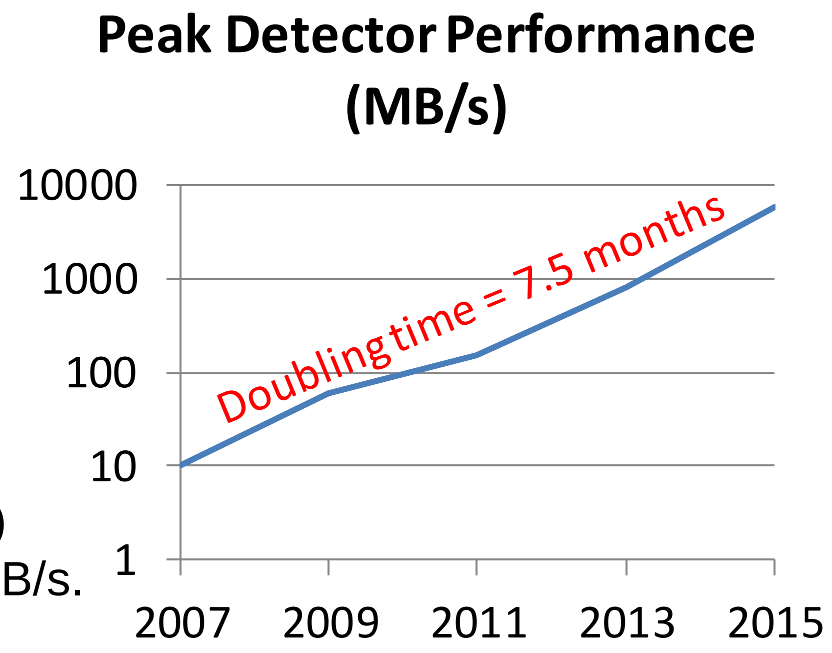
Nick Rees,, Mark Basham, Frederik Ferner,
Ulrik Pedersen, Tom Cobb,
Tobias Richter, Jonathan Thompson…
(Diamond Light Source),
Elena Pourmal (The HDF Group)

Introduction

- History
- Detector developments
  - Parallel detectors
  - Spectroscopic detectors
- HDF5 developments
  - HDF5 1.8.11 (Available now):
    - Dynamically loaded filter libraries
    - Direct write of compressed chunks
  - HDF5 1.10 (Being integrated):
    - New dataset indexing: Extensible array indexing.
    - SWMR
    - VDS
    - Journaling

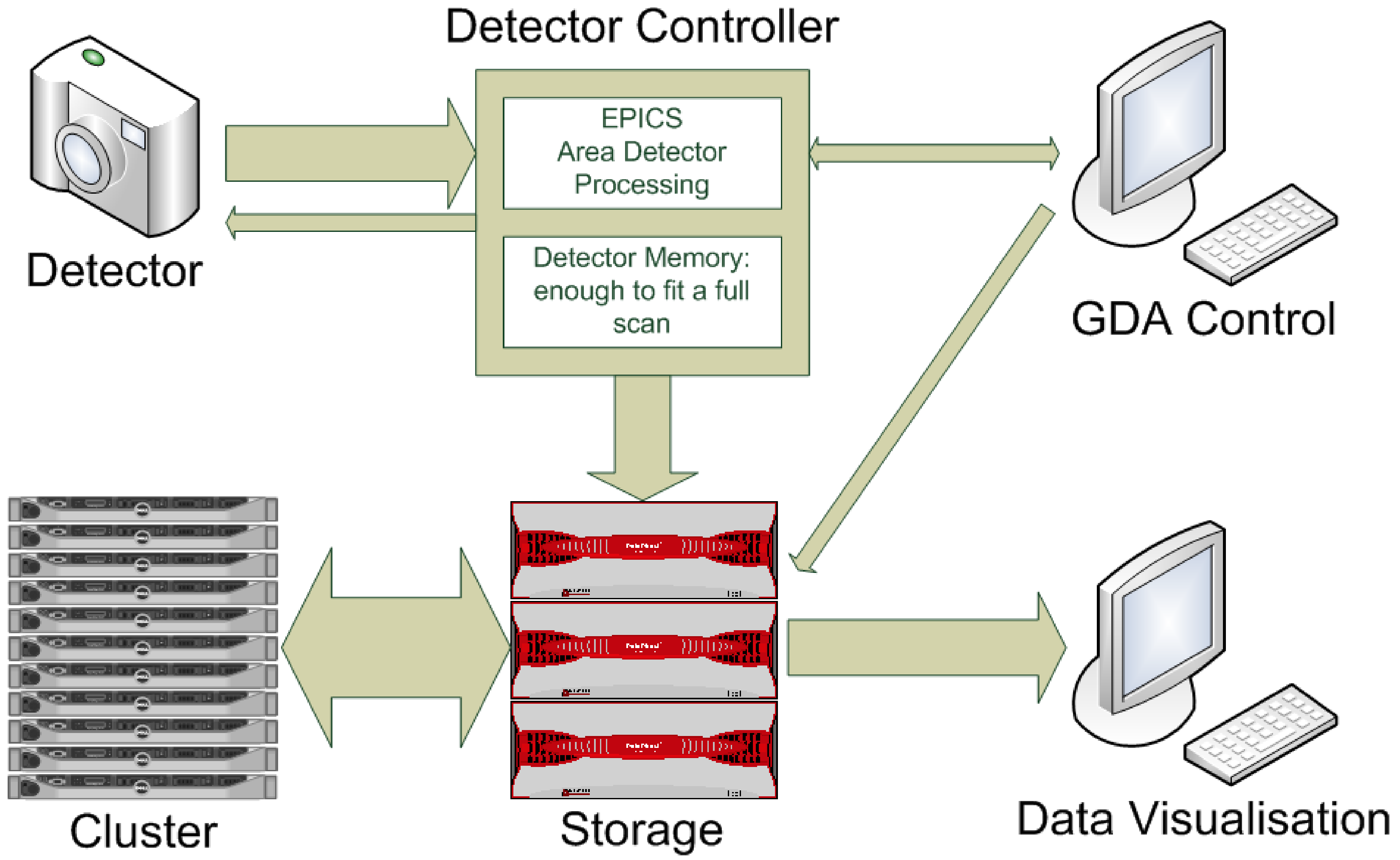
History

- Early 2007:
  - Diamond first user.
  - No detector faster than ~10 MB/sec.
- Early 2009:
  - first Lustre system (DDN S2A9900)
  - first Pilatus 6M system @ 60 MB/s.
- Early 2011:
  - second Lustre system (DDN SFA10K)
  - first 25Hz Pilatus 6M system @150 MB/s.
- Early 2013:
  - first GPFS system (DDN SFA12K)
  - First 100 Hz Pilatus 6M system @ 600 MB/sec
  - ~10 beamlines with 10 GbE detectors (mainly Pilatus and PCO Edge).
- Late 2015:
  - delivery of Percival detector (6000 MB/sec).

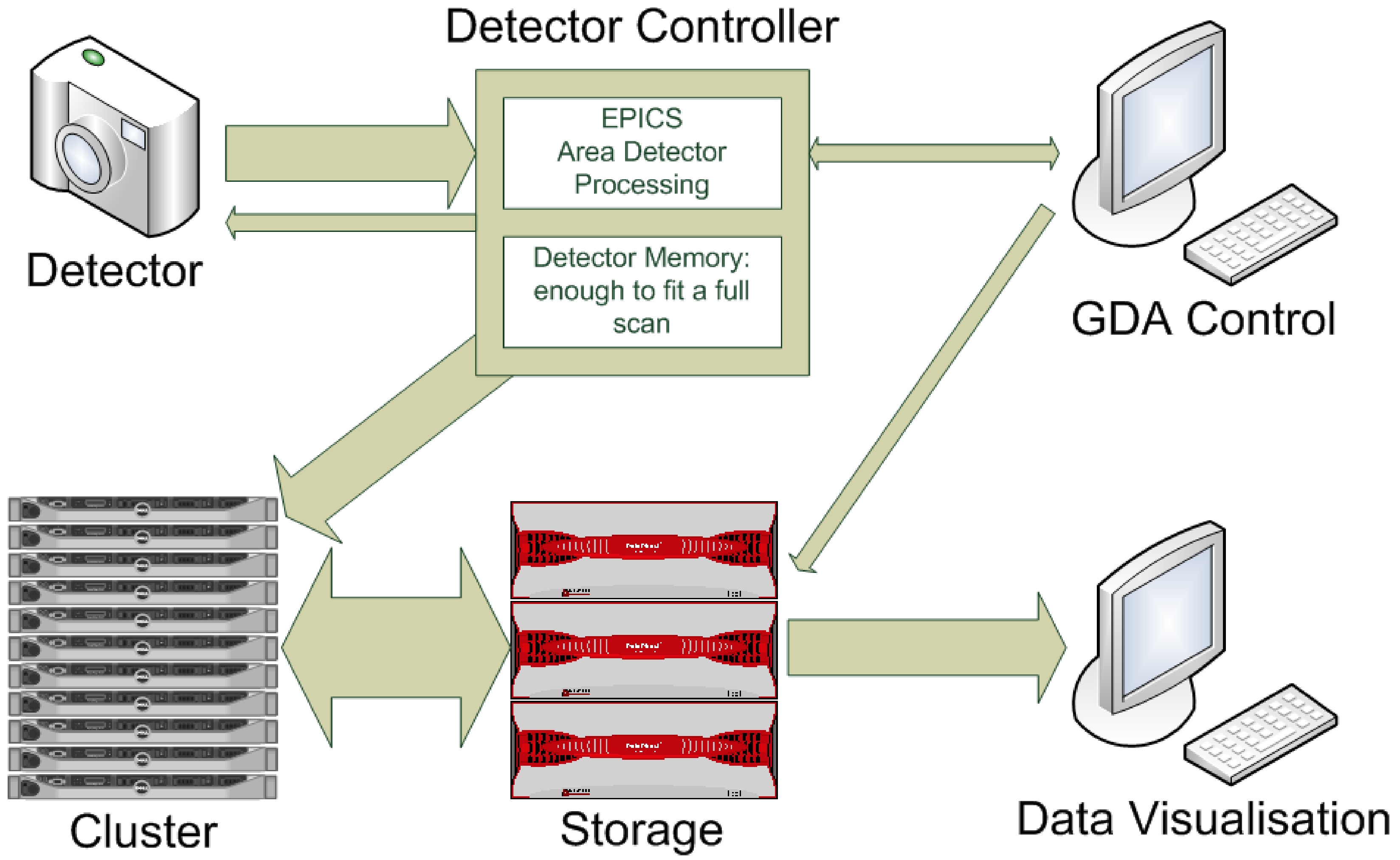


DETECTOR DEVELOPMENTS

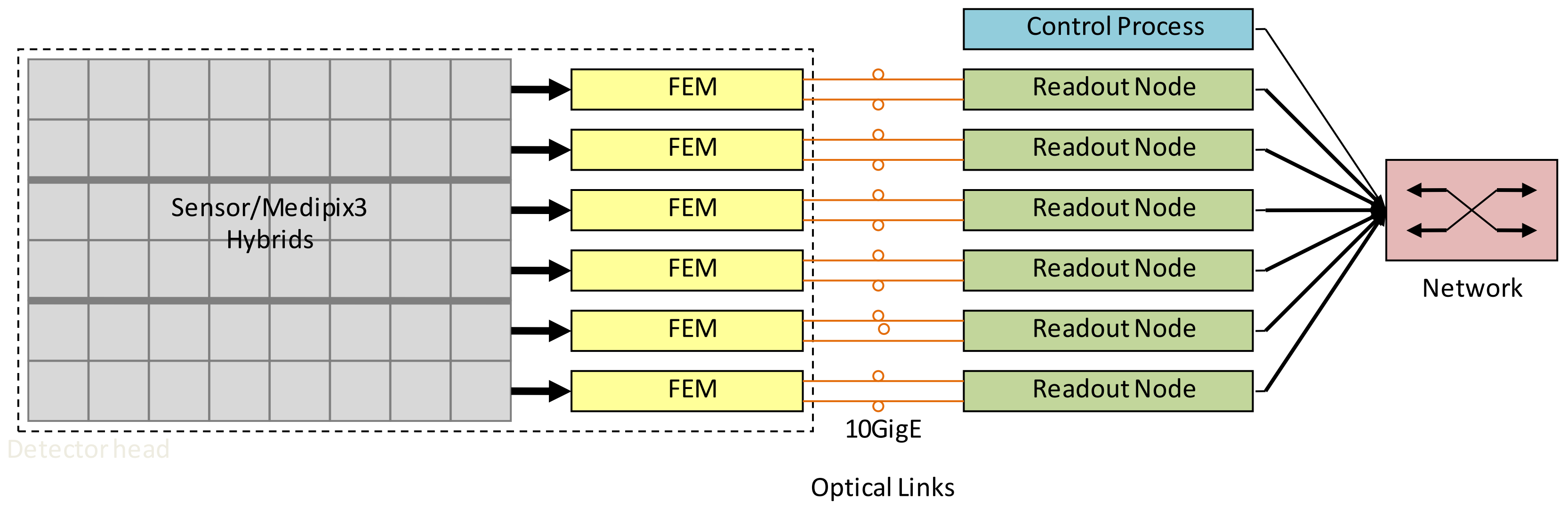
Diamond Detector Model



Potential EPICS Version 4 Model

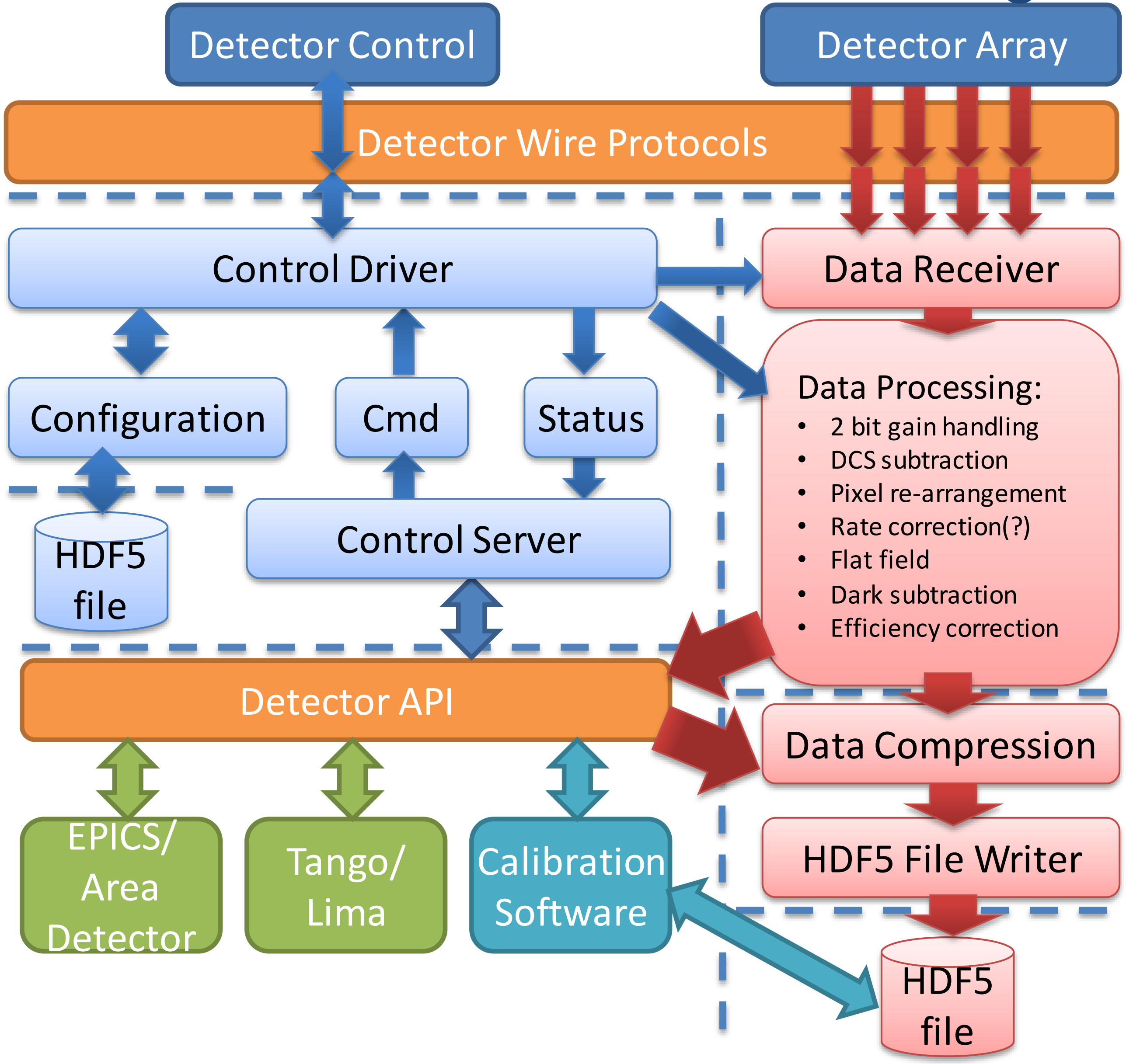


Basic Parallel Detector Design



- Readout nodes all write in parallel
- Need a mechanism to splice data into one file.

Detector Block Diagram



Actual/potential network or CPU socket boundaries

Detector Control Software

Detector Data Stream (n copies)

Documented Controlled Interfaces

Beamline Control Software

Detector Engineer Software

Spectroscopic Detectors

- areaDetector is poorly named…
  - Base class is asynNDArrayDriver, but this name is not so catchy…
    - NDArray* classes provide basic functionality
    - Core plugins derive from NDPluginDriver and many will work with any NDArray.
    - Most popular plugins are the file writing plugins that get data to disk.
  - Basic areaDetector class is really NDDriver
    - Provides methods for reading out a typical areaDetector
    - The methods aren't so good for other types of detectors, e.g.:
      - Spectroscopic (MCA like) detectors.
      - Analogue (A/D like) detectors.

Proposal for new ND Drivers

- Need a set of basic driver classes for other types of NDArrays
  - NDMCADriver (or NDSpectraDriver)
    - Generates 2-D array of energy vs detector channel
    - 3rd dimension can be time.
  - NDADCDriver (or ND DigitizerDriver)
    - Generates 1D array of values from a set of ADC's
    - 2nd dimension can be time.
- Each driver can feed existing plugins, but also could benefit from specialist plugins.

HDF5 DEVELOPMENTS

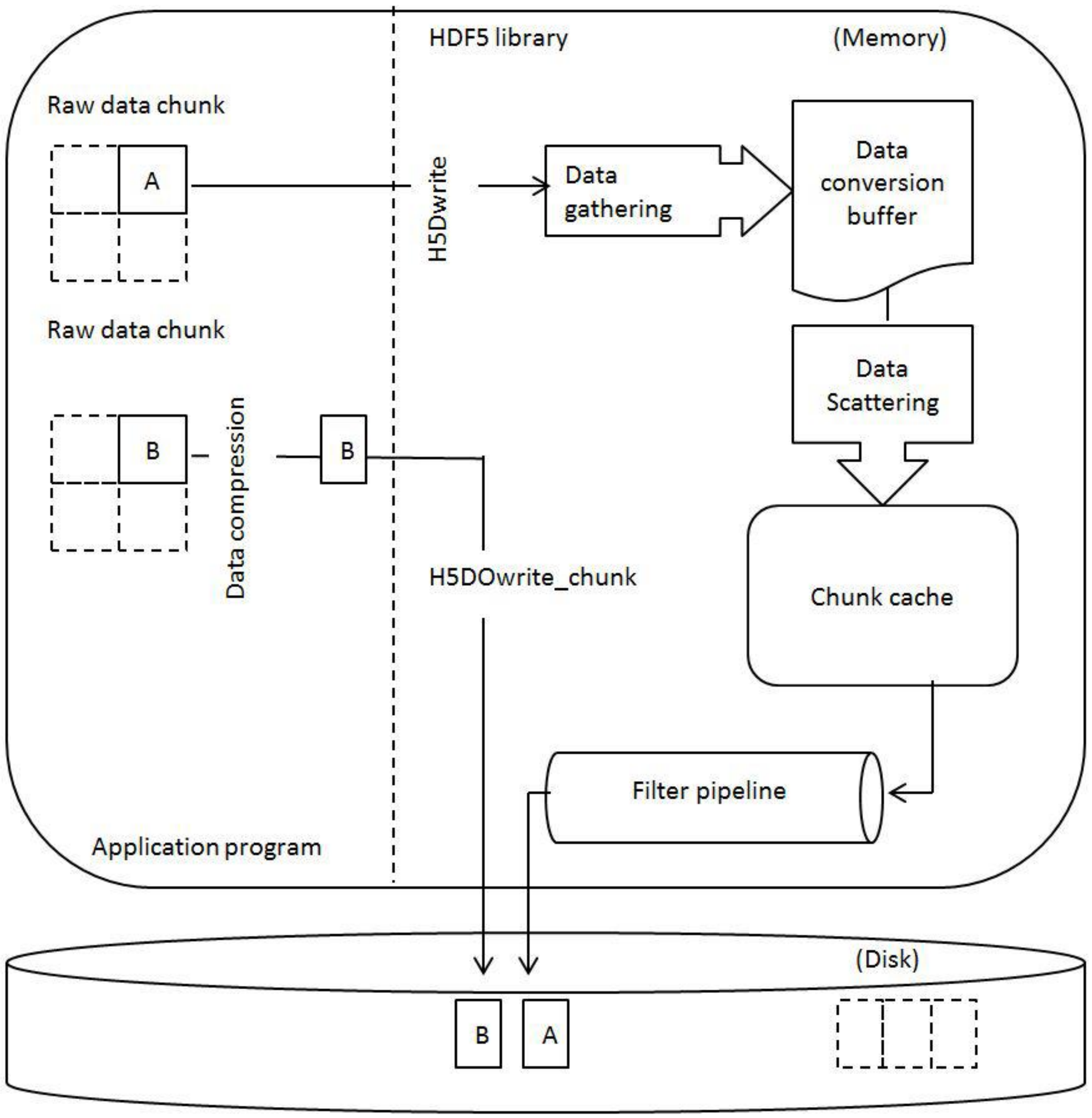
HDF5 key points

- HDF5 is mature software that grew up in the HPC environment.
- It is a widely used standard and has the richest set of high performance functionality of any file format.
- It allows rich metadata and flexible data formats
- It has some caveats we know about:
  - HDF5 is single threaded.
  - pHDF5 relies on MPI, which doesn't happily co-exist with highly threaded architectures like EPICS.
  - pHDF5 is not as efficient as HDF5
  - pHDF5 doesn't allow compression.
  - Files cannot be read while they are written

Recent Developments: Release 1.8.11

- H5DO_write_chunk
  - Funded by Dectris and PSI
  - Improves writing compressed data by:
    - Avoiding double copy of filter pipeline
    - Allowing optimised (e.g. multithreaded) compression implementations
- Pluggable filters
  - Funded by DESY
  - Allows users to provide filters as a shared library that is loaded at runtime.
  - Search path set by environment variable: HDF5_PLUGIN_PATH

Chunk write mechanism



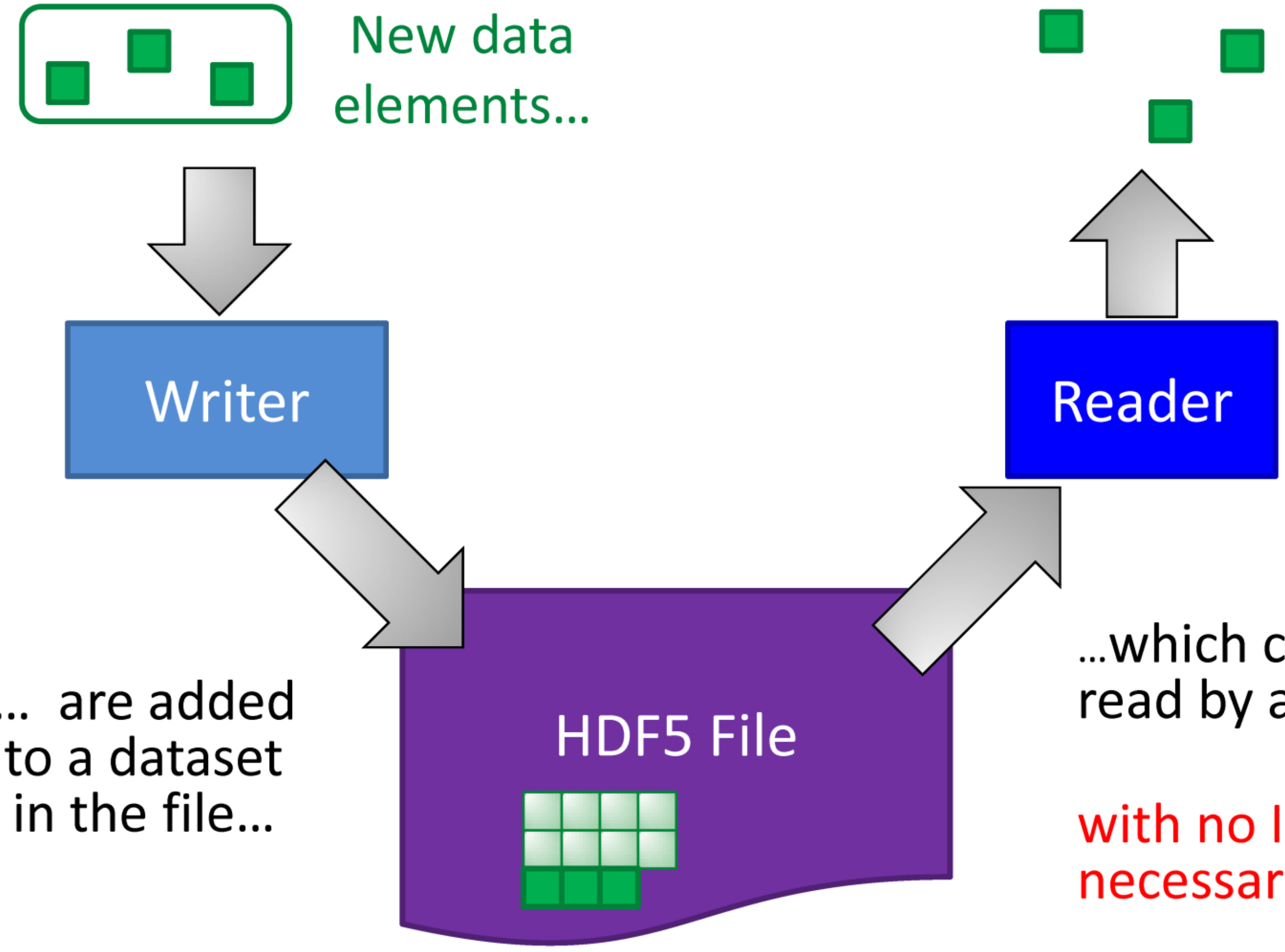
diamond

Current developments: Release 1.10

- File format changes that need a major release:
  - Improved dataset indexing:
    - New B-Tree implementation
    - Extensible array indexing
  - Journaling
  - Virtual Object Layer
  - Single Writer Multiple Reader (SWMR)
    - Funded by Diamond, Dectris and ESRF
  - Virtual Data Set
    - Funded by Diamond, DESY and Percival Detector
- Beta release July 2015

CONCURRENCY:
SINGLE-WRITER/MULTIPLE-READER

Goal



… are added to a dataset in the file…

…which can be read by a reader…

with no IPC necessary.

Status

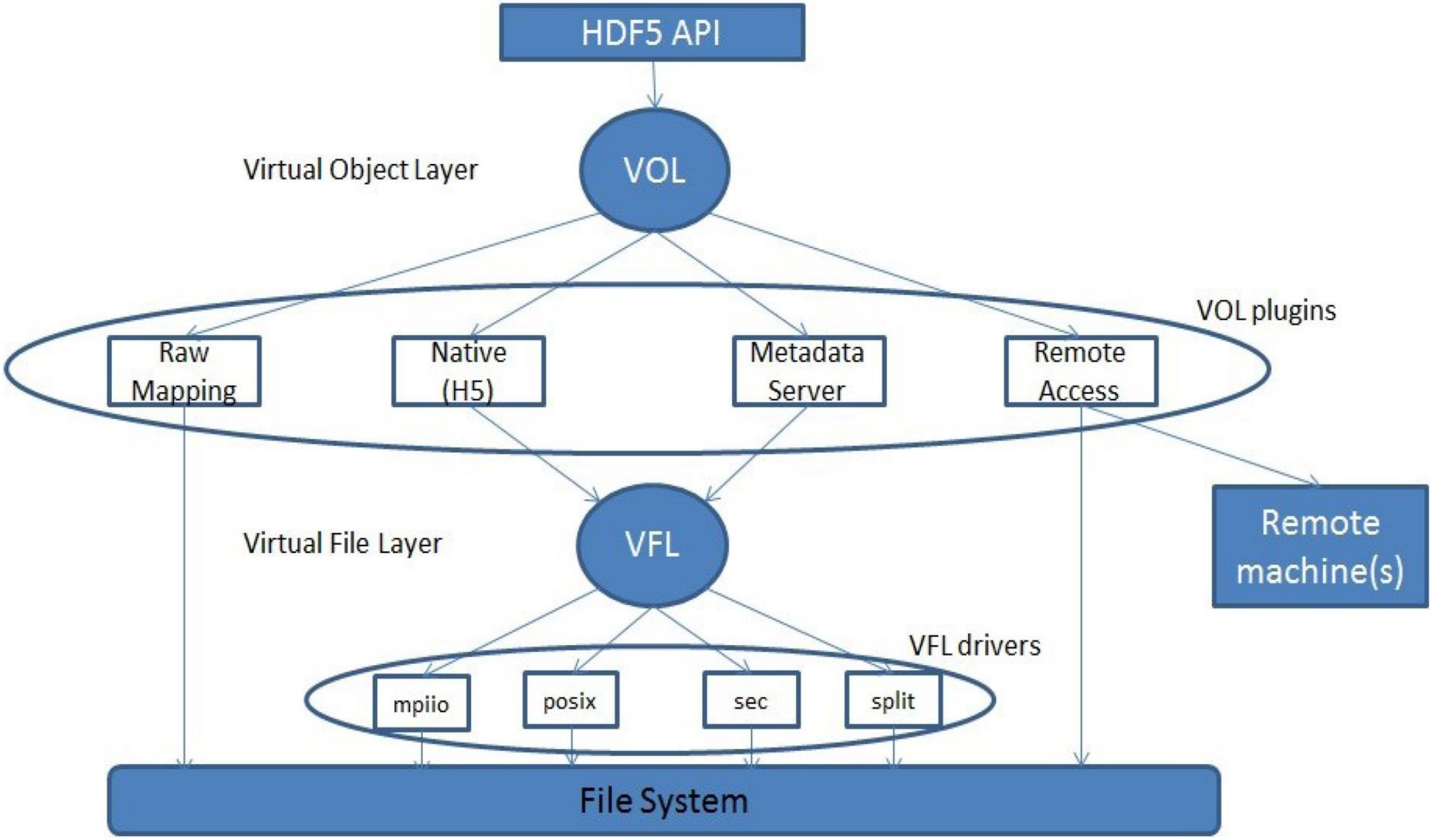
- Implemented for raw data “append only” scenario
  - No creation or deletion of the datasets, groups, and attributes is allowed at this time
- Product is under integration
  - Works on GPFS, Lustre, Linux Ext3, Ext4, FreeBSD USF2, OS X HDFS+
  - Documentation http://www.hdfgroup.org/HDF5/docNewFeatures/
  - Source ftp://ftp.hdfgroup.uiuc.edu/pub/outgoing/SWMR/

VIRTUAL OBJECT LAYER

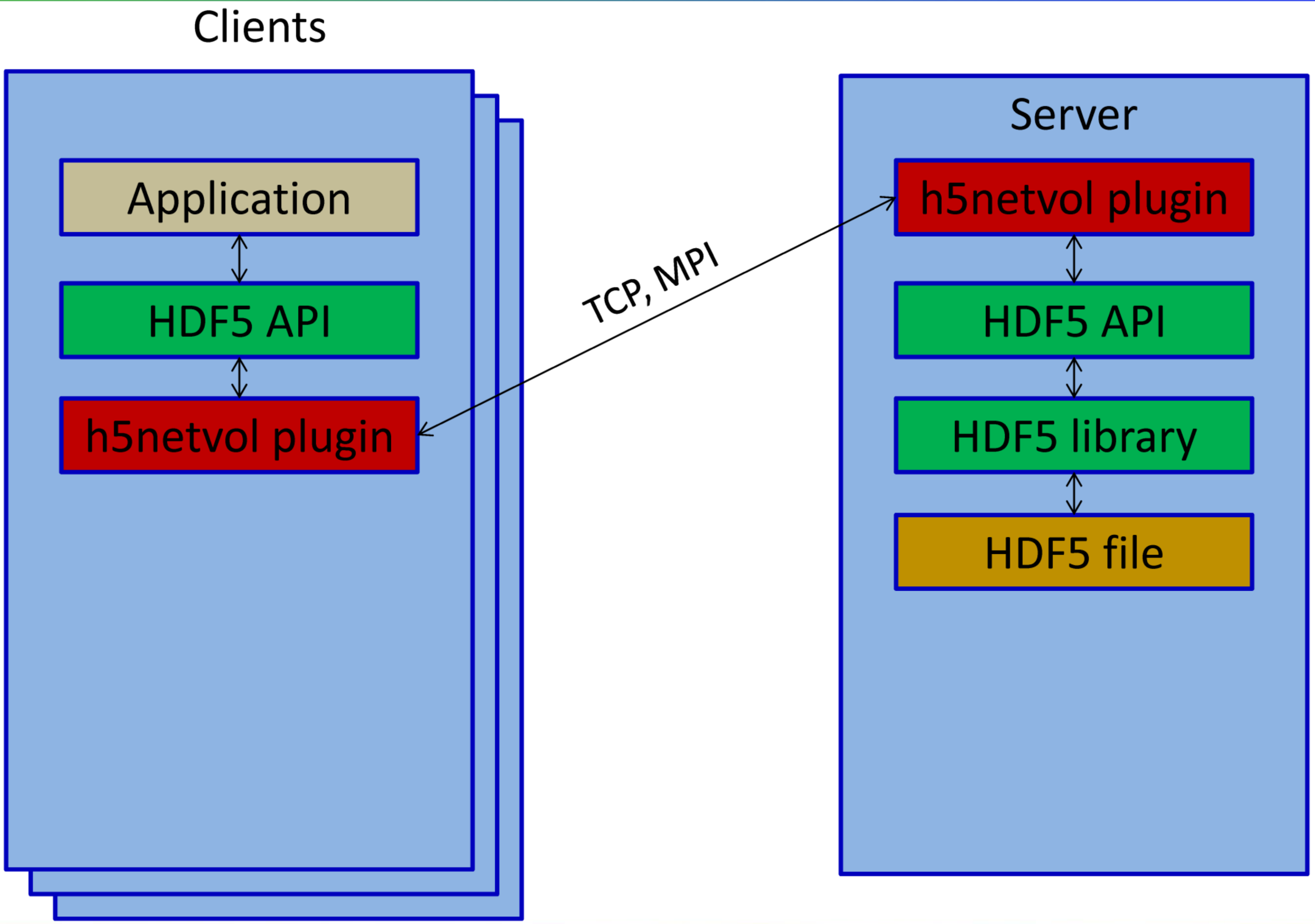
Virtual Object Layer

- Goal
  - Provide an application with the HDF5 data model and API, but allow different underlying storage mechanisms
- New layer below HDF5 API
  - Intercepts all API calls that can touch the data on disk and routes them to a Virtual Object Driver
- Potential Object Drivers (or plugins):
  - Native HDF5 driver (writes to HDF5 file)
  - Raw driver (maps groups to file system directories and datasets to files in directories)
  - Remote driver (the file exists on a remote machine)

Virtual Object Layer



Network VOL Plugin



Other Capabilities

- Allows concurrent access, even by multiple writers
  - Could even be useful on a single machine
- Includes locking scheme that can be used to control access to objects

DATA INDEXING

Indexing and HDF5

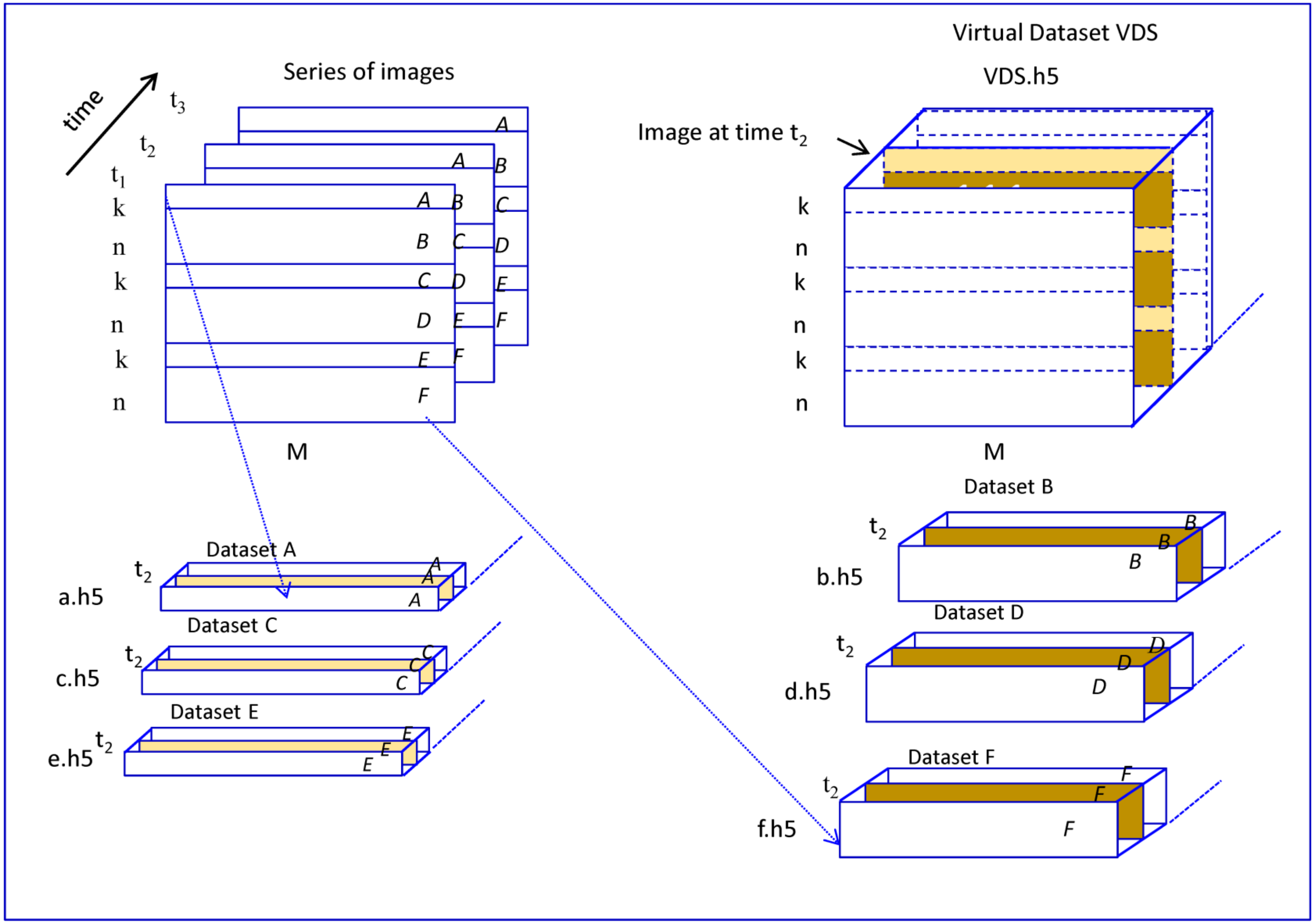
- New APIs for indexing and querying of both structure and contents of HDF5 containers
- H5Q API defines query to apply to a container
  Create/combine queries (OR, AND)
  - Basic operators supported (≤ , ≥ ,=, ≠) on either dataset/attribute values, link/attribute names
- HDF5V API retrieves data
- HDF5X API adds third-party indexing plugins

VIRTUAL DATASET

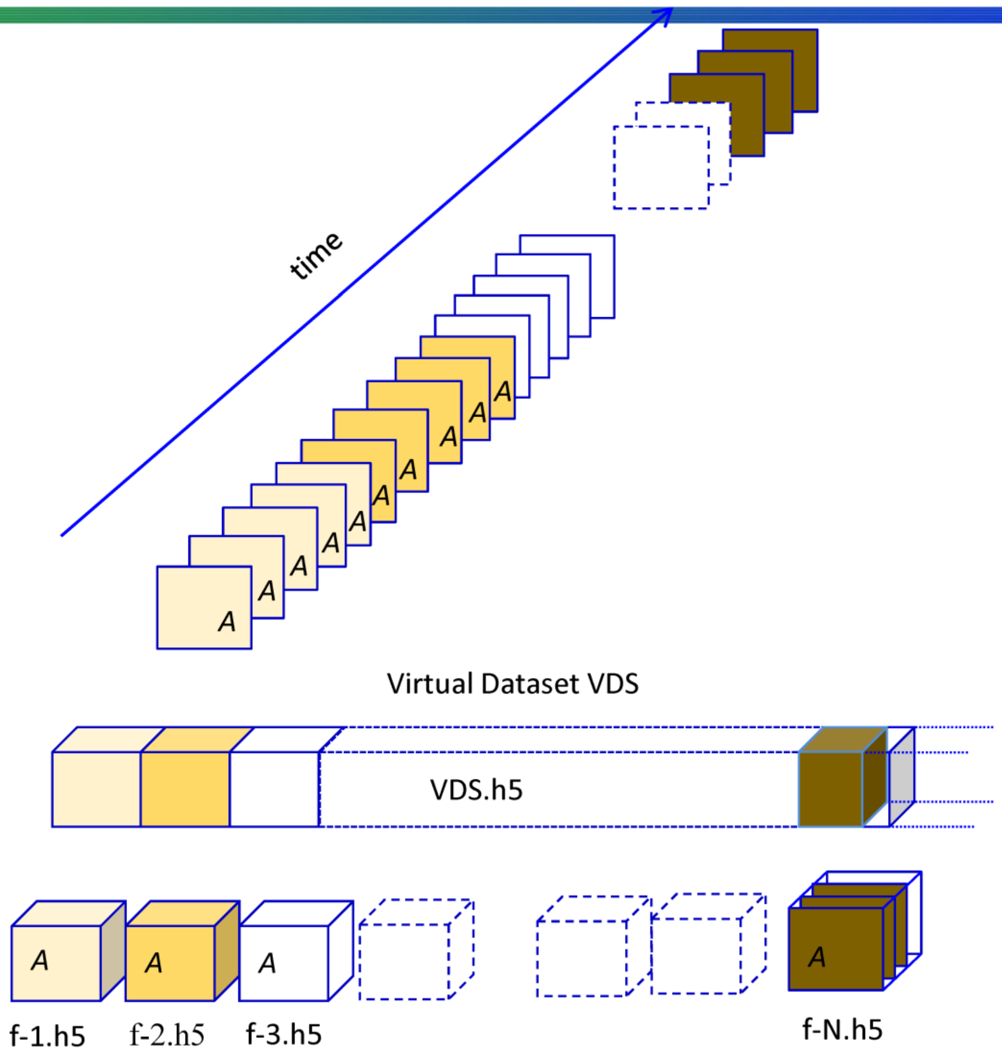
Challenge

- How to view data stored across the HDF5 files as an HDF5 dataset on which normal operations can be performed?
  - High-level approach
    - Special library that applications like MATLAB and H5Py will need to use
    - Example : THREDDS Data Server based on OPeNDAP http://www.unidata.ucar.edu/software/thredds/current/tds/TDS.html
  - Native HDF5 implementation
    - Transparent to applications

Virtual Dataset: Excalibur Detector Use Case

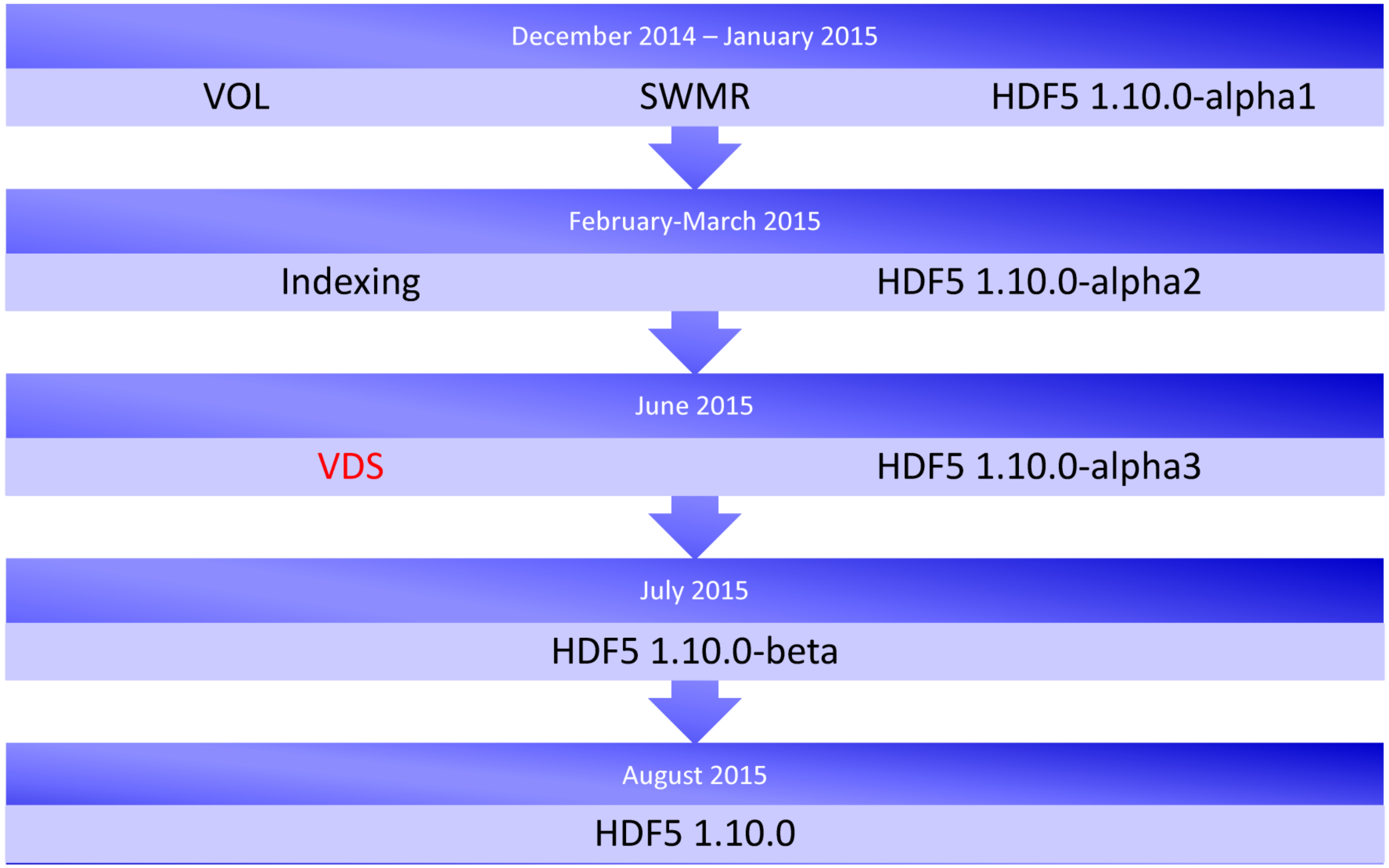


Example: "Printf" Source Generation



File names are generated by "printf" capability

HDF5 1.10.0 Roadmap



Features and release dates are tentative; may change

Thank you for your attention…