Introduction to HDF5

Quincey Koziol
Director of Core Software & HPC
The HDF Group

Goal

- Introduce you to HDF5's:
  - Background and History
  - Data model
  - Programming model
  - File Format
  - Future Developments

BACKGROUND

What is HDF5?

- HDF5 == Hierarchical Data Format, v5
- Open **file format**
  - Designed for high volume or complex data
- Open source **software**
  - Works with data in the format
- A **data model**
  - Structures for data organization and specification

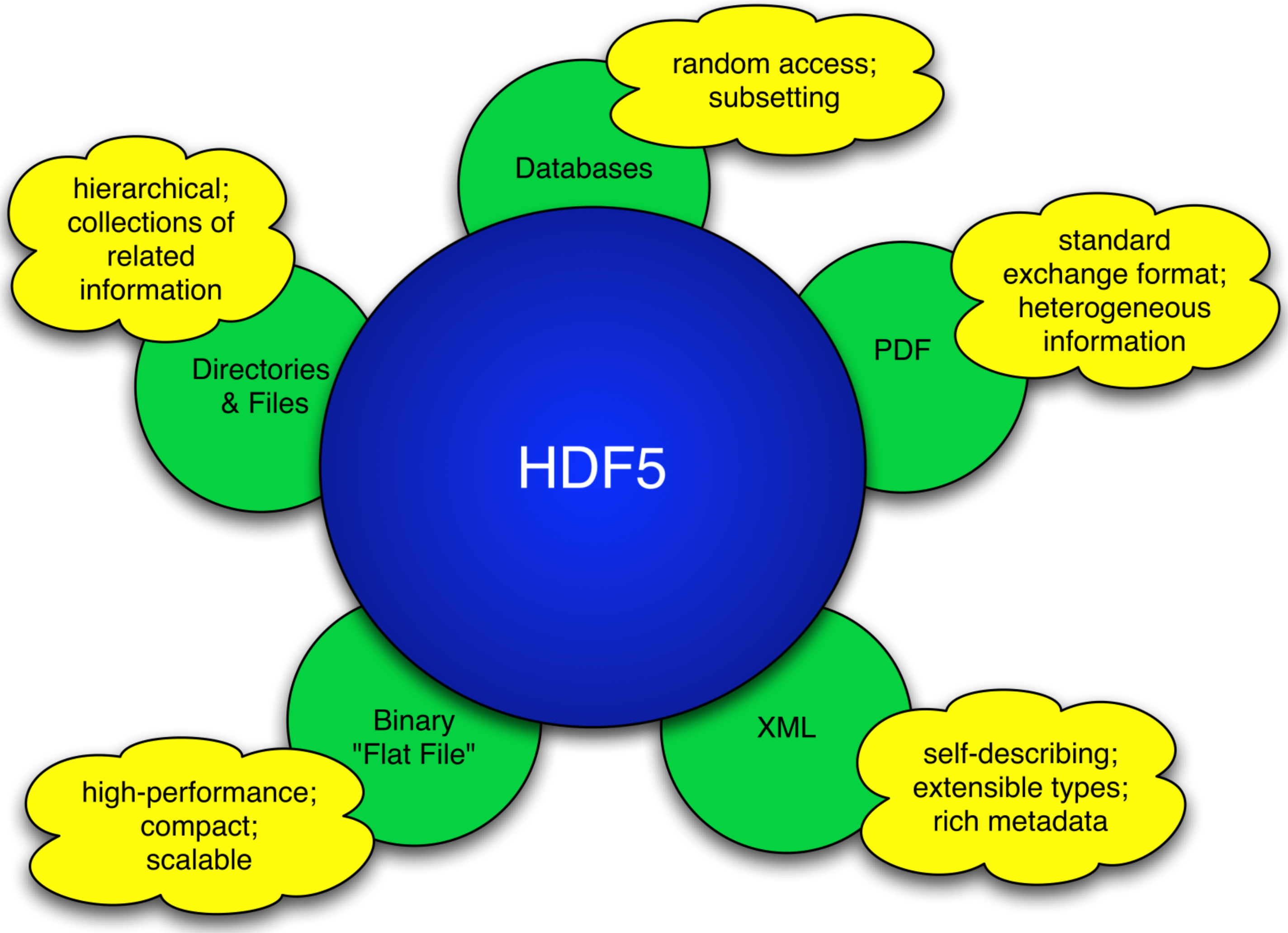
HDF5 Technology Platform

- HDF5 Abstract Data Model
  - Defines the “building blocks” for data organization and specification
  - Files, Groups, Links, Datasets, Attributes, Datatypes, Dataspaces

- HDF5 Software
  - Tools
  - Language Interfaces
  - HDF5 Library

- HDF5 Binary File Format
  - Bit-level organization of HDF5 file
  - Defined by HDF5 File Format Specification

HDF5 is like …


random access;
subsetting
Databases
hierarchical;
collections of
related
information
Directories
& Files
HDF5
PDF
standard
exchange format;
heterogeneous
information
Binary
"Flat File"
high-performance;
compact;
scalable
XML
self-describing;
extensible types;
rich metadata

What is HDF5?

- **A versatile data model** that can represent very complex data objects and a wide variety of metadata.
- **A completely portable file format** with no limit on the number or size of data objects stored.
- **An open source software library** that runs on a wide range of computational platforms, from cell phones to massively parallel systems, and implements a high-level API with C, C++, Fortran 90, and Java interfaces.
- **A rich set of integrated performance features** that allow for access time and storage space optimizations.
- **Tools and applications** for managing, manipulating, viewing, and analyzing the data in the collection.

Why use HDF5?

- Challenging data:
  - Application data that pushes the limits of what can be addressed by traditional database systems, XML documents, or in-house data formats.
- Software solutions:
  - For very large datasets, very fast access requirements, or very complex datasets.
  - To easily share data across a wide variety of computational platforms using applications written in different programming languages.
  - That take advantage of the many open-source and commercial tools that understand HDF5.
  - Enabling long-term preservation of data.

Who uses HDF5?

- Examples of HDF5 user communities
  - Astrophysics
  - Astronomers
  - NASA Earth Science Enterprise
  - Dept. of Energy Labs
  - Supercomputing centers in US, Europe and Asia
  - Financial Institutions
  - NOAA
  - Manufacturing industries
  - Many others
- For a more detailed list, visit
  - http://www.hdfgroup.org/HDF5/users5.html

Brief History of HDF

1987 At NCSA (University of Illinois), a task force formed to create an architecture-independent format and library:

AEHOO (All Encompassing Hierarchical Object Oriented format)

Became HDF

Early 1990's NASA adopted HDF for Earth Observing System project

1996 DOE's ASC (Advanced Simulation and Computing) Project began collaborating with the HDF group (NCSA) to create "Big HDF" (Increase in computing power of DOE systems at LLNL, LANL and Sandia National labs, required bigger, more complex data files).

"Big HDF" became HDF5.

HDF5 was released with support from DOE Labs, NASA, NCSA

2006 The HDF Group spun off from University of Illinois as non-profit corporation

The HDF Group

The HDF Group Mission

To ensure long-term accessibility of HDF data through sustainable development and support of HDF technologies.

Goals of The HDF Group

- Maintain and evolve HDF for sponsors and communities that depend on it

- Provide support to the HDF communities through consulting, training, tuning, development, research

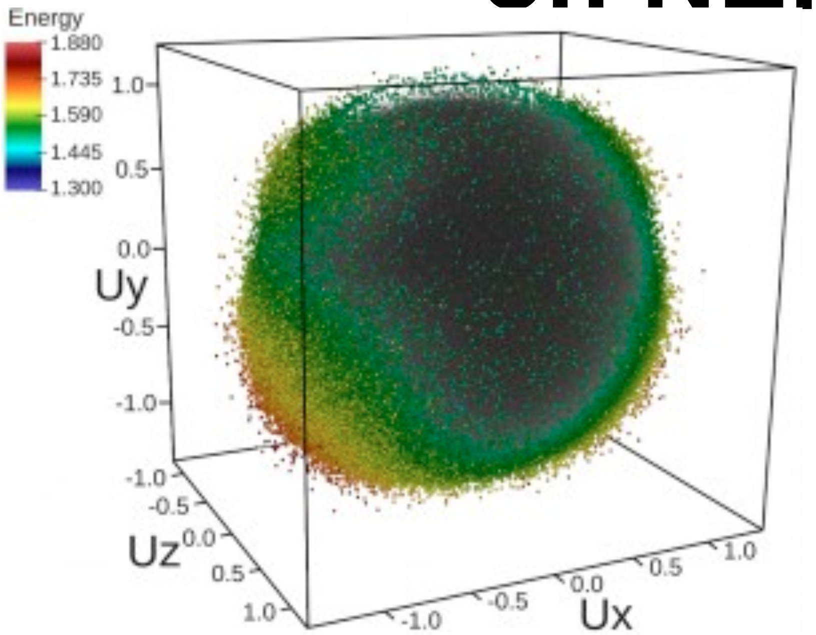
- Sustain the company for the long term to assure data access over time

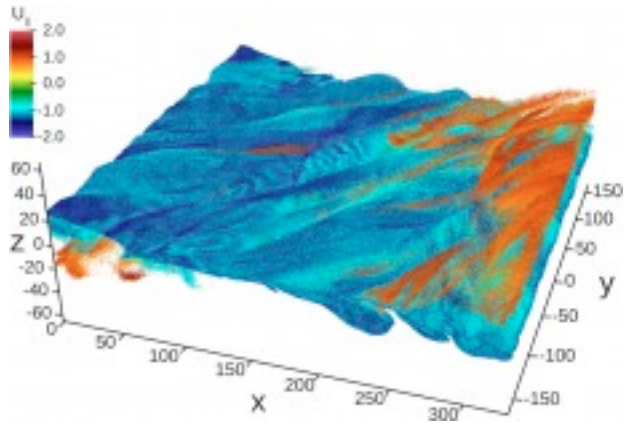
The HDF Group Services

- Helpdesk and Mailing Lists
  - Available to all users as a first level of support: help@hdfgroup.org
- Priority Support
  - Rapid issue resolution and advice
- Consulting
  - Needs assessment, troubleshooting, design reviews, etc.
- Training
  - Tutorials and hands-on practical experience
- Enterprise Support
  - Coordinating HDF activities across departments
- Special Projects
  - Adapting customer applications to HDF
  - New features and tools
  - Research and Development

Cool recent application

Trillion Particle Simulation on NERSC’s hopper system





- VPIC with 100,000 nodes on hopper
- Achieved 27GB/s sustained rate to each 32TB HDF5 file (out of 35GB/s theoretical peak)
- http://1.usa.gov/Le0JF8

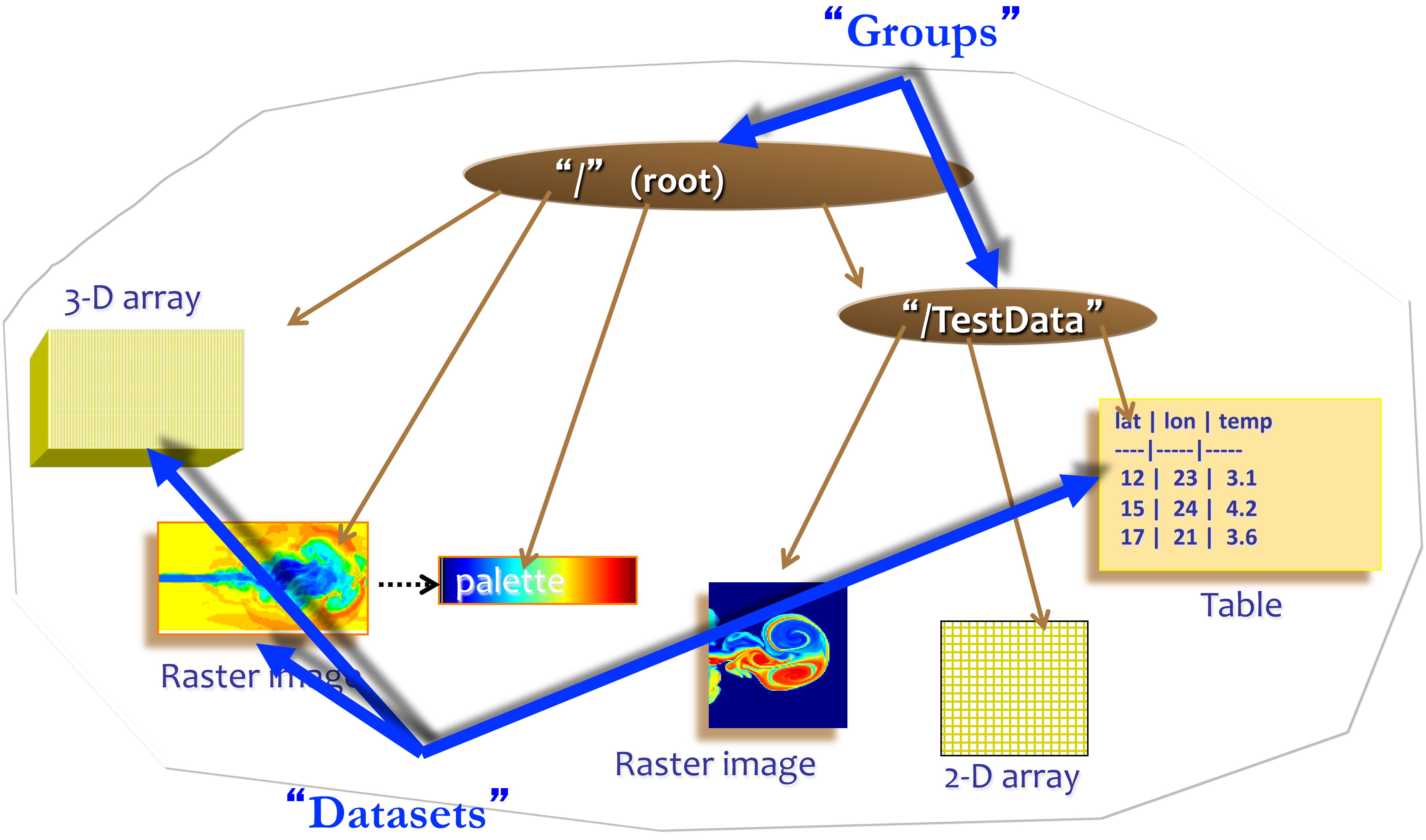
HDF5 DATA MODEL

HDF5 Data Model

- Groups – provide structure among objects
- Datasets – where the primary data goes
  - Data arrays
  - Rich set of datatype options
  - Flexible, efficient storage and I/O
- Attributes, for metadata

Everything else is built essentially from these parts.

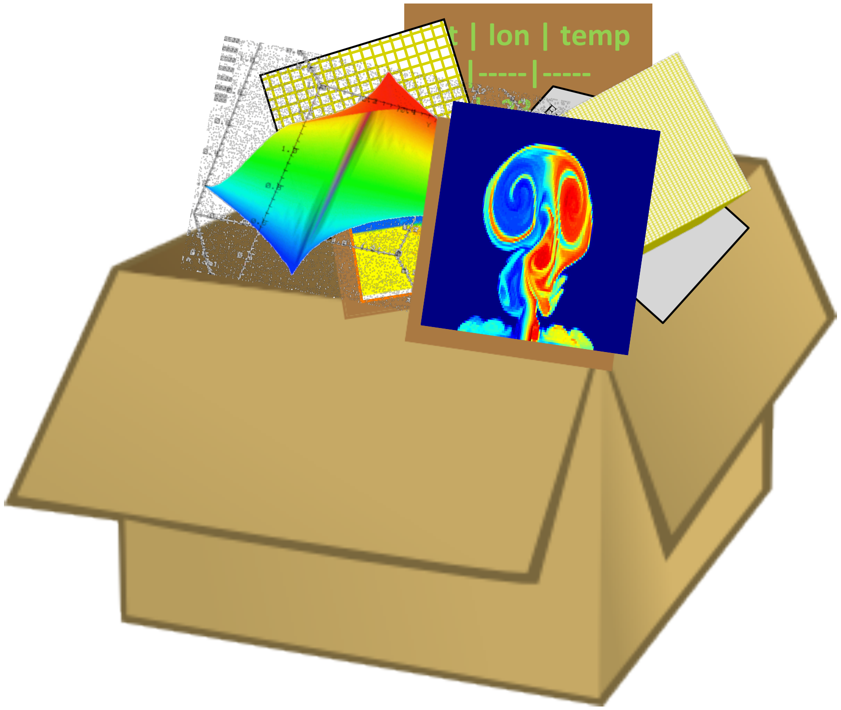
Structures to organize objects



lat	lon	temp
12	23	3.1
15	24	4.2
17	21	3.6

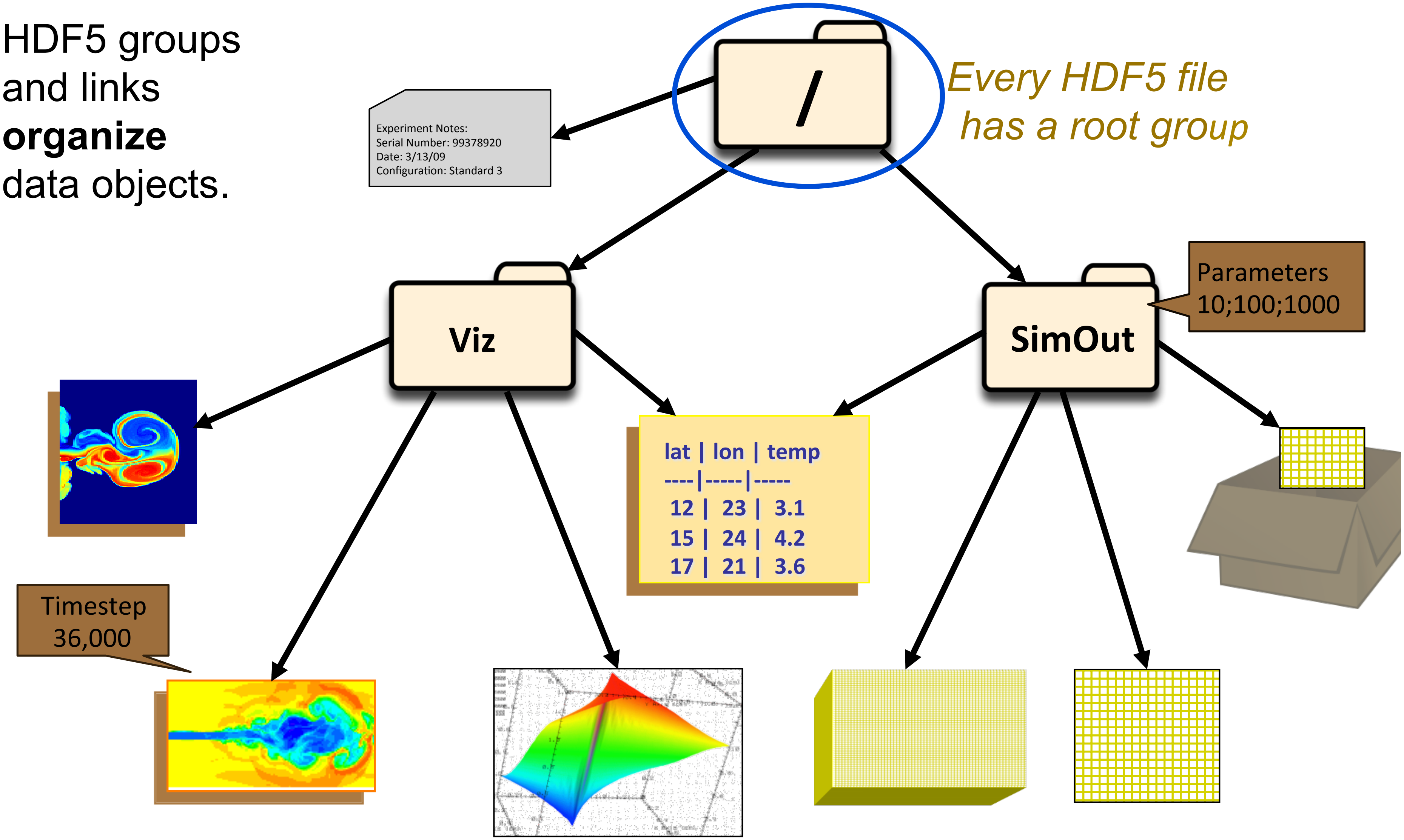
HDF5 File

An HDF5 file is a **smart container** that holds data objects.



HDF5 Groups and Links

HDF5 groups and links **organize** data objects.



Every HDF5 file has a root group

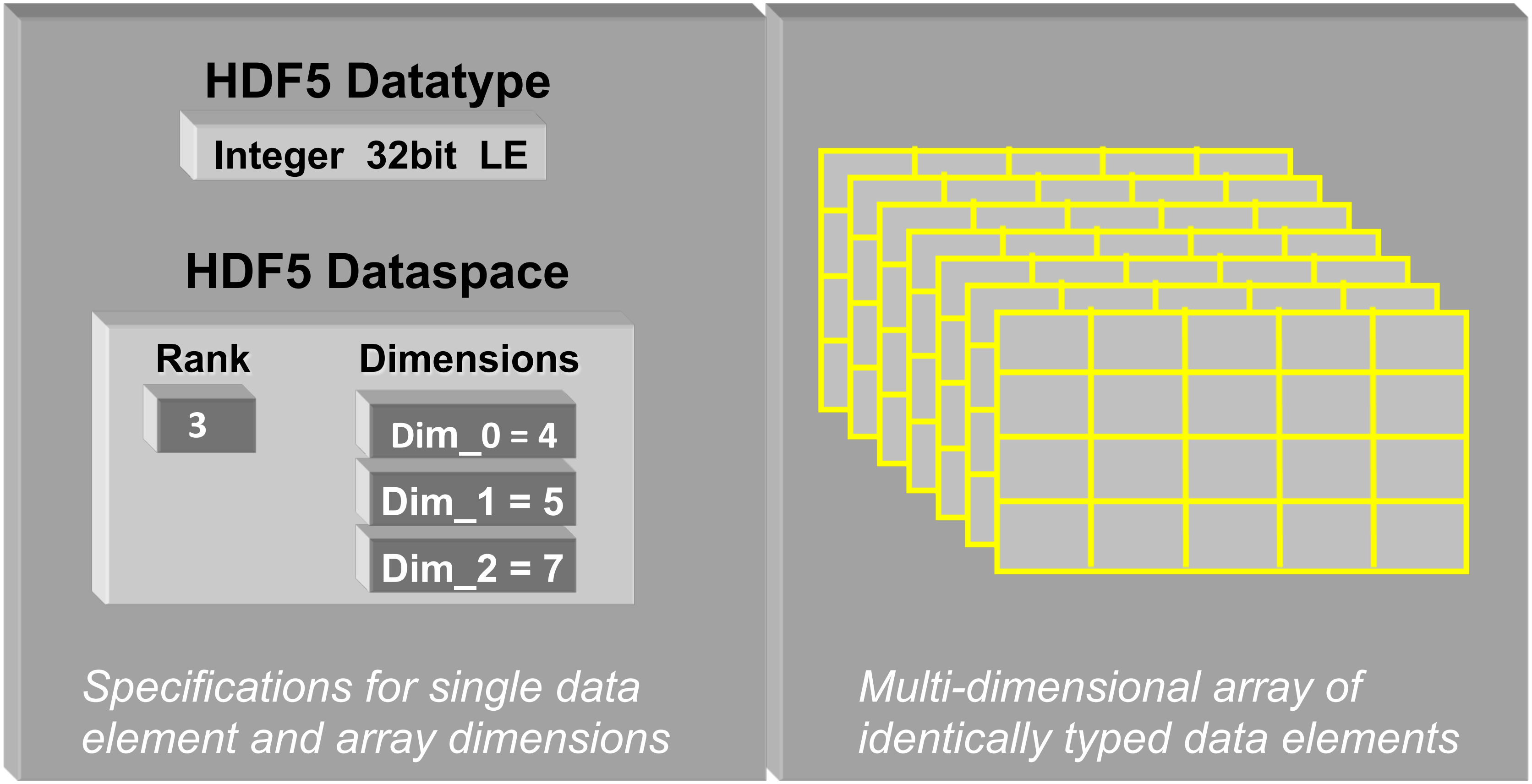
Experiment Notes:
Serial Number: 99378920
Date: 3/13/09
Configuration: Standard 3

Parameters
10;100;1000

Timestep
36,000

lat	lon	temp
12	23	3.1
15	24	4.2
17	21	3.6

HDF5 Dataset



HDF5 Datatype

Integer 32bit LE

HDF5 Dataspace

Rank: 3

Dimensions: Dim_0 = 4, Dim_1 = 5, Dim_2 = 7

Specifications for single data element and array dimensions

Multi-dimensional array of identically typed data elements

- HDF5 datasets **organize and contain** "raw data values".
  - HDF5 datatype describes individual data elements.
  - HDF5 dataspace describes the logical layout of the data elements.

HDF5 Dataspace

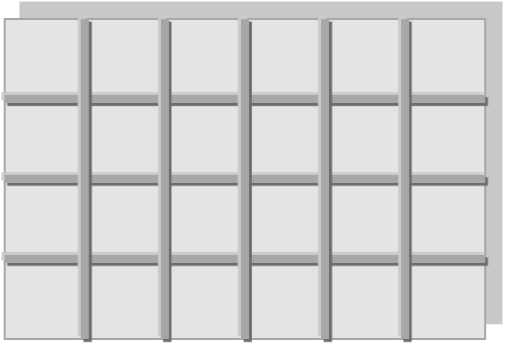
- Describes the logical layout of the elements in an HDF5 dataset
  - NULL
    - no elements
  - Scalar
    - single element
  - Simple array (*most common*)
    - multiple elements organized in a rectangular array
      - rank = number of dimensions
      - dimension sizes = number of elements in each dimension
      - maximum number of elements in each dimension
        - may be fixed or unlimited

HDF5 Dataspace

Two roles:

Dataspace contains spatial information

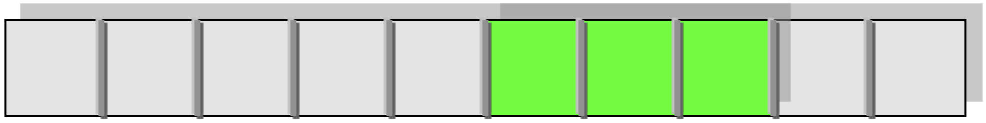
- Rank and dimensions
- Permanent part of dataset definition



Rank = 2

Dimensions = 4x6

Partial I/0: Dataspace describes application’s data buffer and data elements participating in I/O



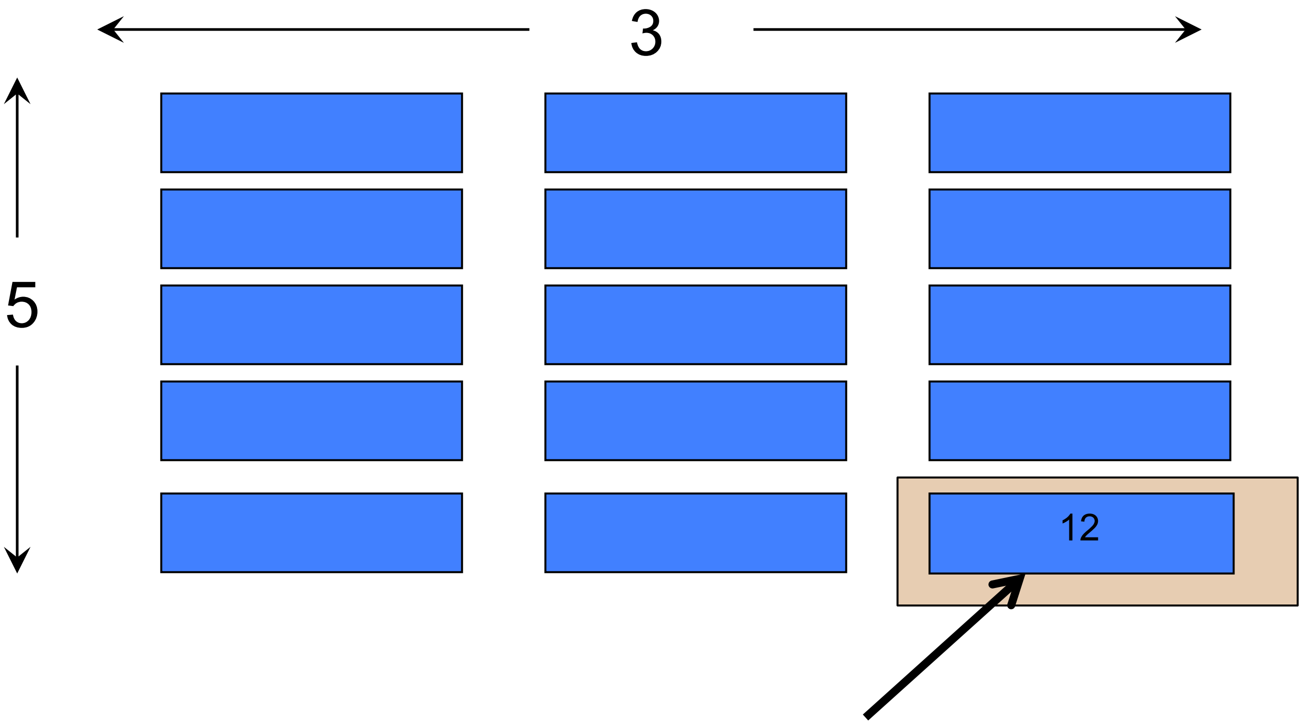
Rank = 1

Dimension = 10

HDF5 Datatypes

- Describe individual data elements in an HDF5 dataset
- Wide range of datatypes supported
  - Integer
  - Float
  - Enum
  - Array
  - User-defined (e.g., 13-bit integer)
  - Variable length types (e.g., strings)
  - Compound (similar to C structs)
  - Many more …

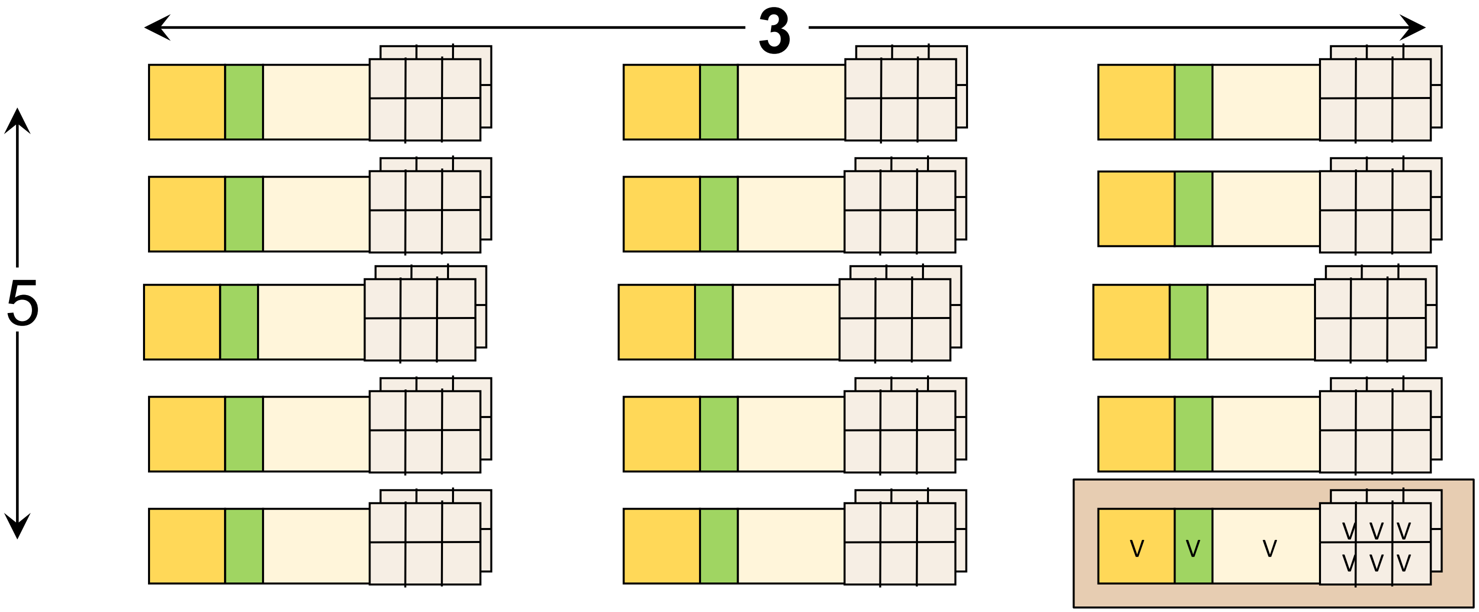
HDF5 Dataset



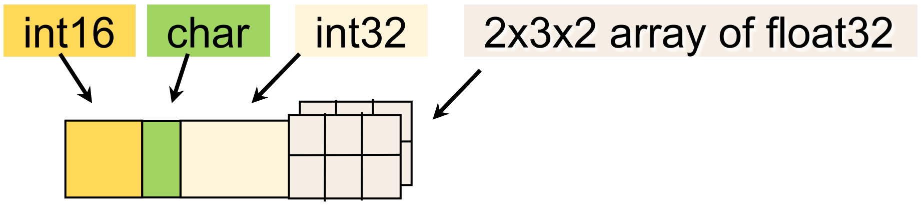
Datatype: 32-bit Integer

Dataspace: Rank = 2
Dimensions = 5 x 3

HDF5 Dataset with Compound Datatype



Compound Datatype:



int16

char

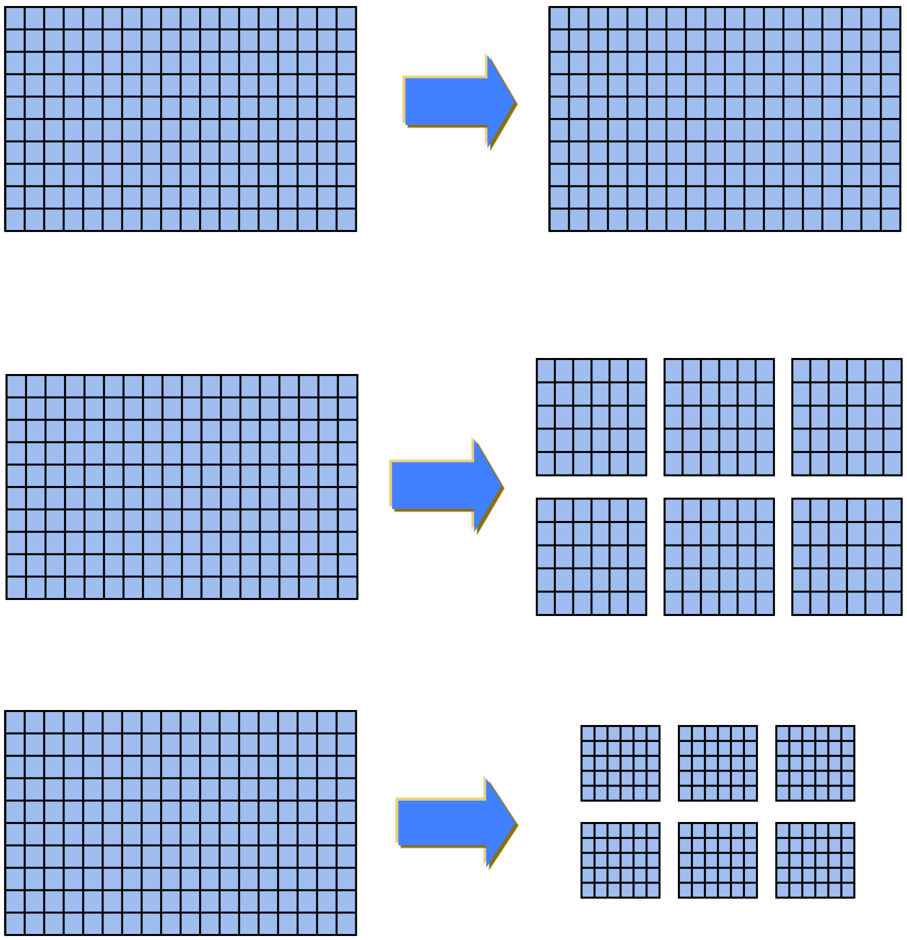
int32

2x3x2 array of float32

Dataspace: Rank = 2

Dimensions = 5 x 3

How data is stored?



Buffer in memory

Data in the file

Contiguous (default)

Data elements stored physically adjacent to each other

Chunked

Better access time for subsets; extendible

Chunked & Compressed

Improves storage efficiency, transmission speed

HDF5 Attributes

- Typically contain user metadata
- Have a name and a value
- Attributes “decorate” HDF5 objects
- Value is described by a datatype and a dataspace
- Analogous to a dataset, but do not support partial I/O operations; nor can they be compressed or extended

HDF5 SOFTWARE

HDF5 Home Page

HDF5 home page: http://hdfgroup.org/HDF5/

- Latest release: HDF5 1.8.12 (1.8.13 coming May 2014)

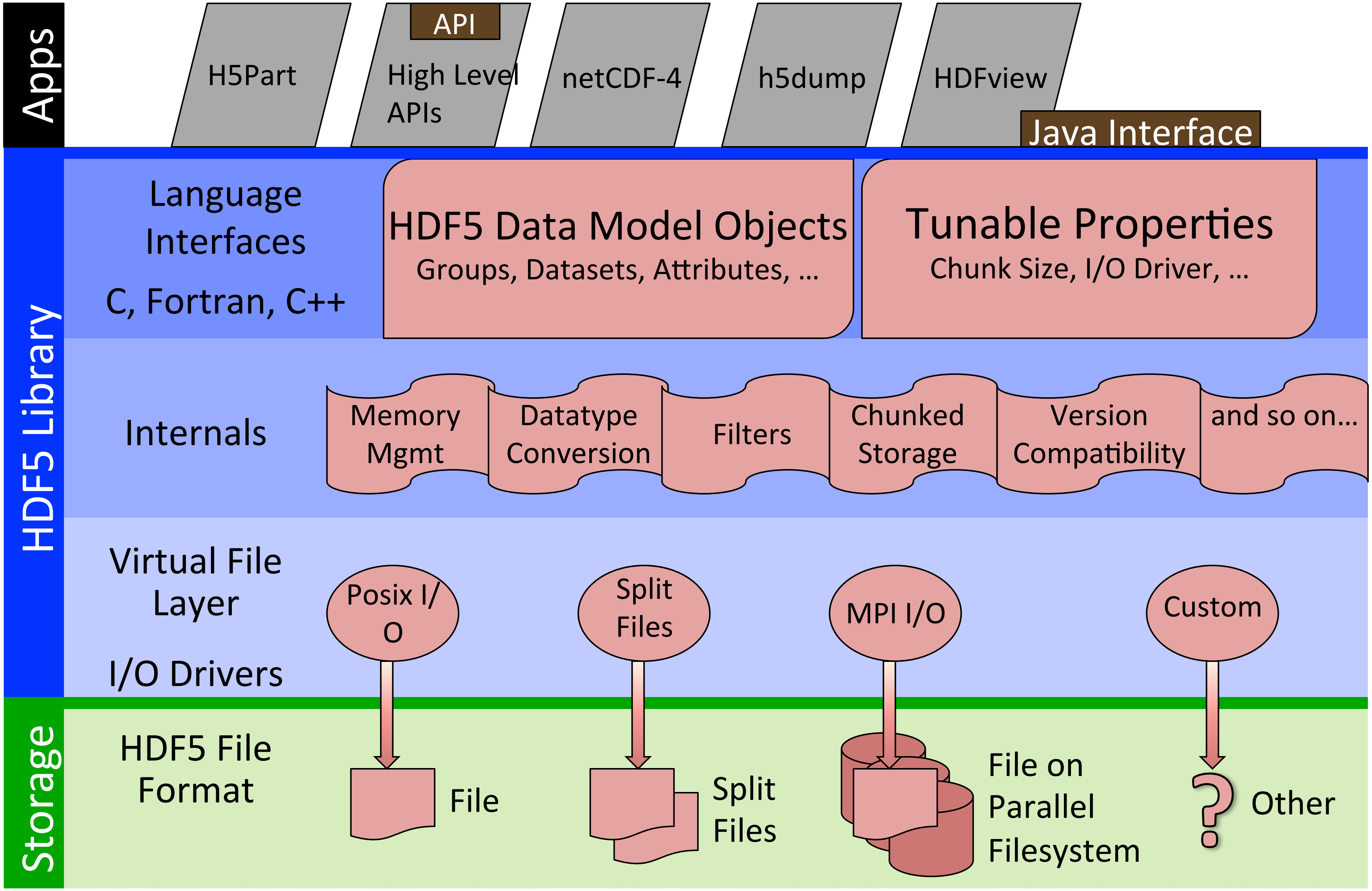
HDF5 source code:

- Written in C, and includes optional C++, Fortran 90 APIs, and High Level APIs
- Contains command-line utilities (h5dump, h5repack, h5diff, ..) and compile scripts

HDF5 pre-built binaries:

- When possible, include C, C++, F90, and High Level libraries. Check ./lib/libhdf5.settings file.
- Built with and require the SZIP and ZLIB external libraries

HDF5 Software Layers & Storage



Useful Tools For New Users

`h5dump`:

Tool to "dump" or display contents of HDF5 files

`h5cc, h5c++, h5fc`:

Scripts to compile applications

`HDFView`:

Java browser to view HDF5 files

http://www.hdfgroup.org/hdf-java-html/hdfview/

HDF5 Examples (C, Fortran, Java, Python, Matlab)

http://www.hdfgroup.org/ftp/HDF5/examples/

HDF5 PROGRAMMING MODEL AND API

General Programming Paradigm

- Object is opened or created
- Object is accessed, possibly many times
- Object is closed

- Properties of object are optionally defined
  - ✓Creation properties (e.g., use chunking storage)
  - ✓Access properties

The General HDF5 API

- C, Fortran, Java, C++, and .NET bindings
- IDL, MATLAB, Python (H5Py, PyTables)
- C routines begin with prefix H5*?*

? is a character corresponding to the type of object the function acts on

Example Functions:

H5D :	**D**ataset interface	*e.g.,* **H5Dread**
H5F :	**F**ile interface	*e.g.,* **H5Fopen**
H5S :	data**S**pace interface	*e.g.,* **H5Sclose**

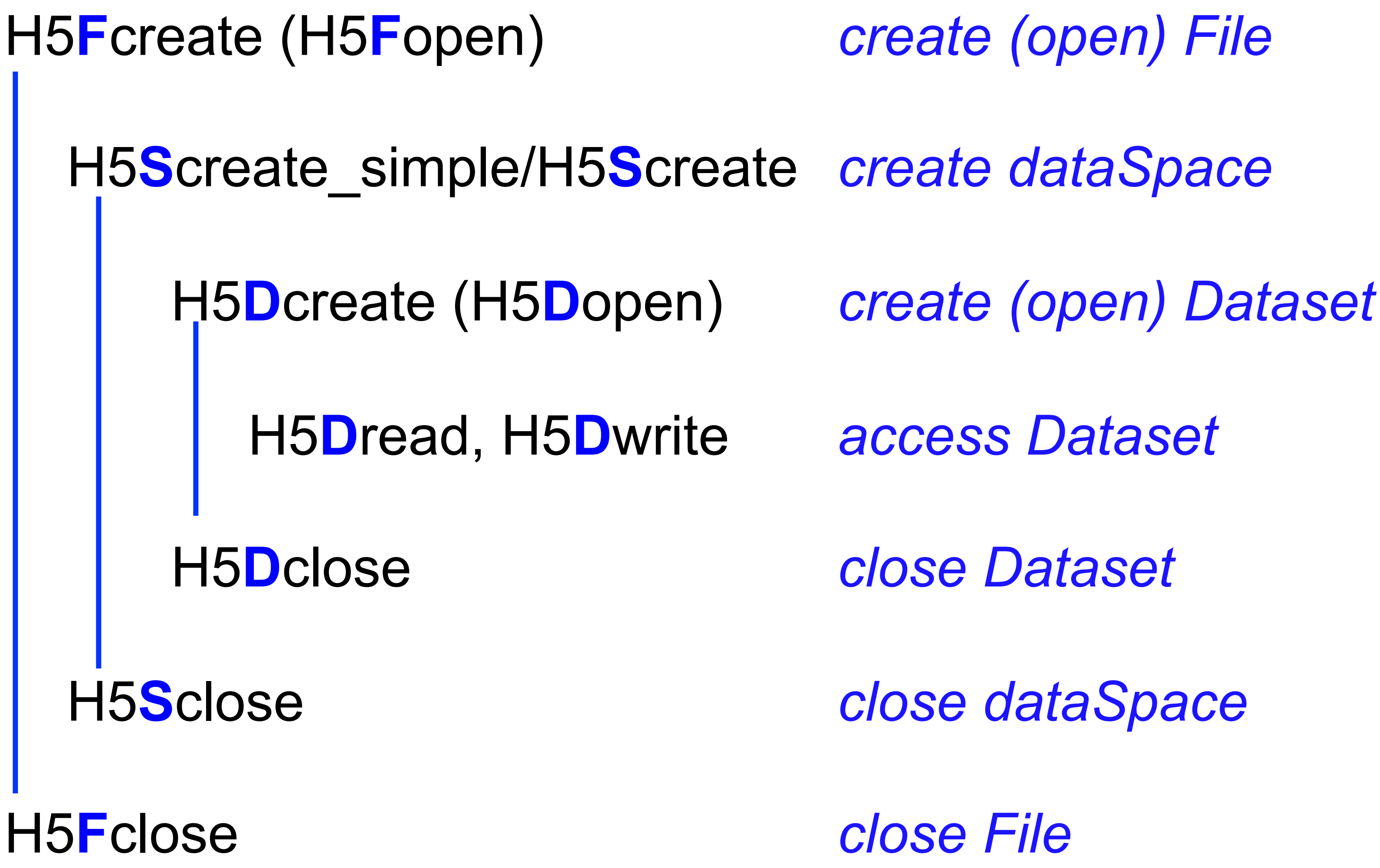
The HDF5 API

- For flexibility, the API is extensive
  - ✓ 300+ functions

Victorinox
Swiss Army
Cybertool 34

- This can be daunting… but there is hope
  - ✓ A few functions can do a lot
  - ✓ Start simple
  - ✓ Build up knowledge as more features are needed

Basic Functions



H5**F**create (H5**F**open)	*create (open) File*
H5**S**create_simple/H5**S**create	*create dataSpace*
H5**D**create (H5**D**open)	*create (open) Dataset*
H5**D**read, H5**D**write	*access Dataset*
H5**D**close	*close Dataset*
H5**S**close	*close dataSpace*
H5**F**close	*close File*

Other Common Functions

Data**S**paces:	H5Sselect_hyperslab (Partial I/O) H5Sselect_elements (Partial I/O) H5Dget_space
Data**T**ypes:	H5Tcreate, H5Tcommit, H5Tclose H5Tequal, H5Tget_native_type
Groups:	H5Gcreate, H5Gopen, H5Gclose
Attributes:	H5Acreate, H5Aopen_name, H5Aclose, H5Aread, H5Awrite
Property lists:	H5Pcreate, H5Pclose H5Pset_chunk, H5Pset_deflate

HDF5 FILE FORMAT

HDF5 File Format

- Defined by the *HDF5 File Format Specification.*

[*http://www.hdfgroup.org/HDF5/doc/H5.format.html*](http://www.hdfgroup.org/HDF5/doc/H5.format.html)

- Specifies the bit-level organization of an HDF5 file on storage media.

- HDF5 library adheres to the File Format, users do not need to know the details of this information.

HDF5 Roadmap

- Concurrency
  - Single-Writer/Multiple-Reader (SWMR)
  - Internal threading
- Virtual Object Layer
- Native HDF5 client/server

- Performance
  - Scalable chunk indices
  - Metadata aggregation and Pa
  - Asynchronous I/O
  - Variable-length records
- Fault tolerance
- Parallel I/O
- I/O Autotuning

Thank You!

Questions?