NetCDF-Java version 2.2 Common Data Model

John Caron

Unidata/UCAR

Dec 10, 2004

Outline

1. Data Models
2. NetCDF-4 and NetCDF-Java 2.2
3. NcML & THREDDS

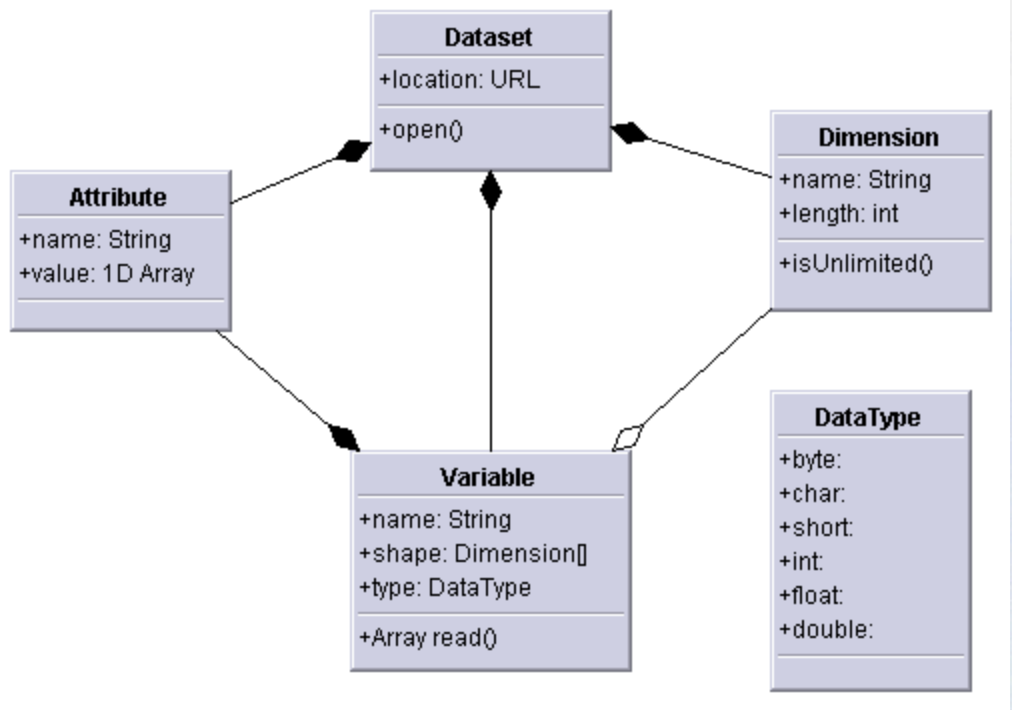
Acknowledgements

- **NetCDF-4**: Russ Rew, Ed Hartnett
- **THREDDS**: Ethan Davis, Ben Domenico, Yuan Ho, Robb Kambic
- **IDV**: Don Murray, Jeff McWhirter, Doug Lindholm
- **NcML**: Luca Cinquini, Ethan Davis, Stefano Nativi, Russ Rew, Bob Drach
- **HDF5**: Mike Folk, Quincey Kiozol, Robert McGrath
- **OpenDAP**: James Gallagher

Creating a Common Data Model from NetCDF, HDF5, OPeNDAP Data Models

NetCDF

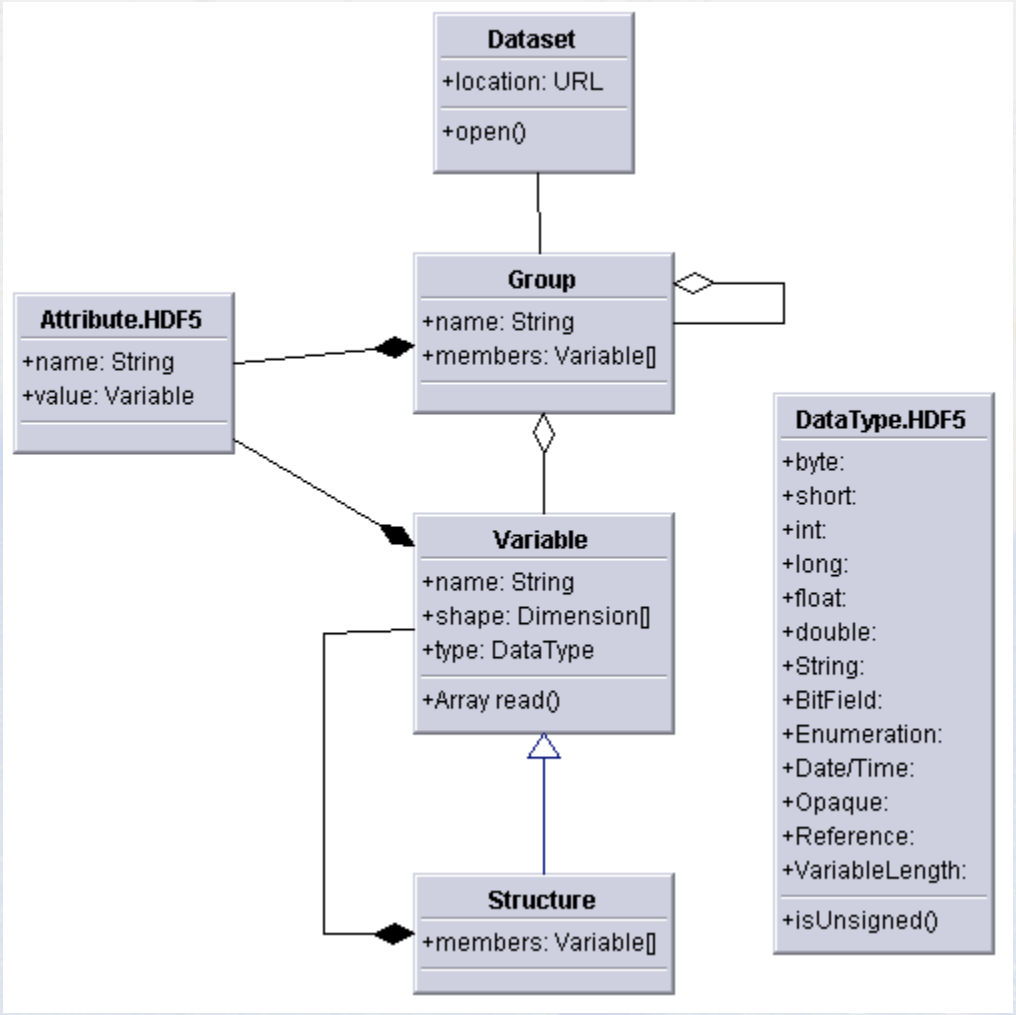
- Machine and OS independent file format for “self-describing” scientific data
- C library (Fortran, C++, Perl, IDL, MatLab, Python, Ruby), Java library
- Multidimensional arrays, efficient subsetting.
- > 20,000 downloads last year (of complete netCDF-3 source by distinct hosts)



NetCDF-3 Data Model

HDF5

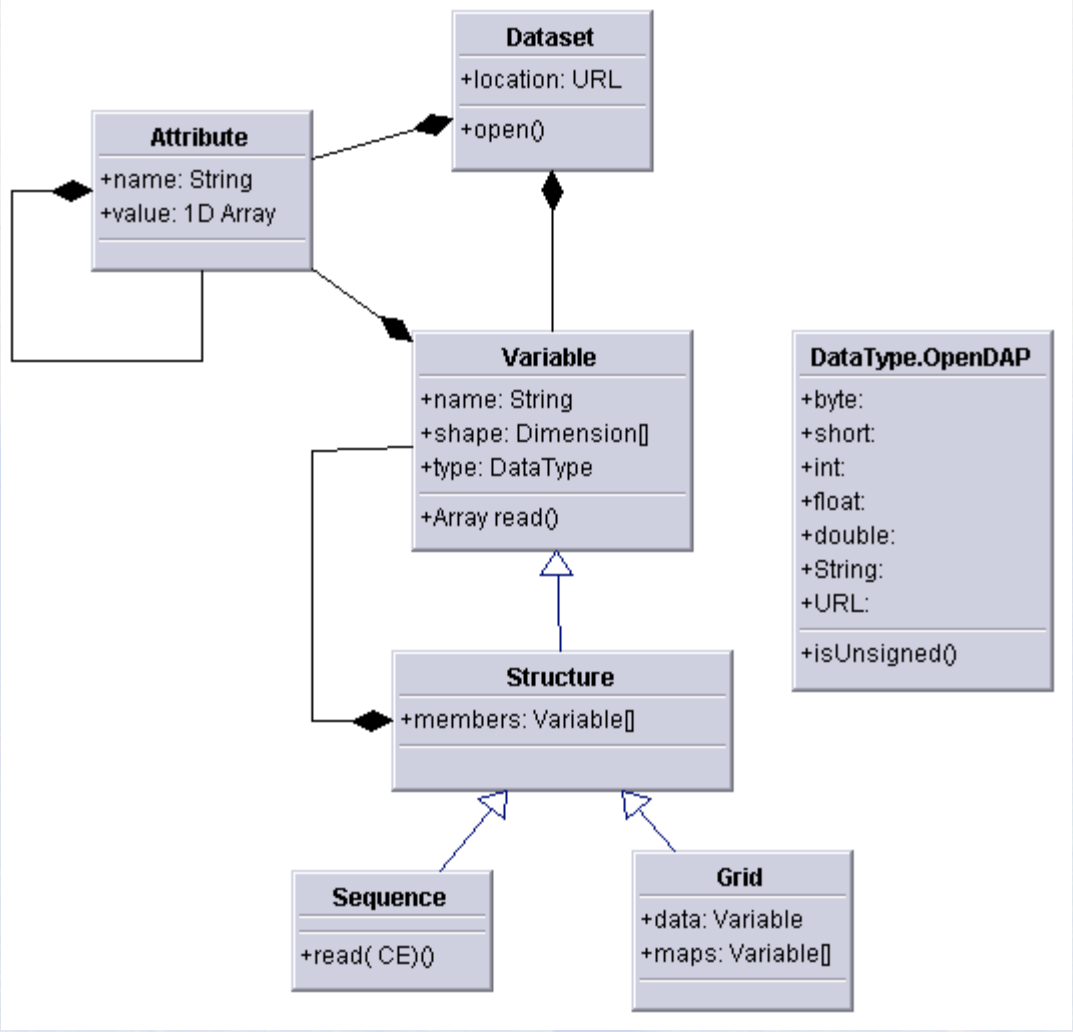
- Machine and OS independent file format for “self-describing” scientific data (NCSA)
- C library (Fortran, Java, others??)
- Evolution from HDF4, but not compatible.
- HDF-EOS, HDF5-EOS
- Standard formats for EOSDIS, ASCI, NPOESS
- Parallel-IO, chunked storage, compression filters, many data types.



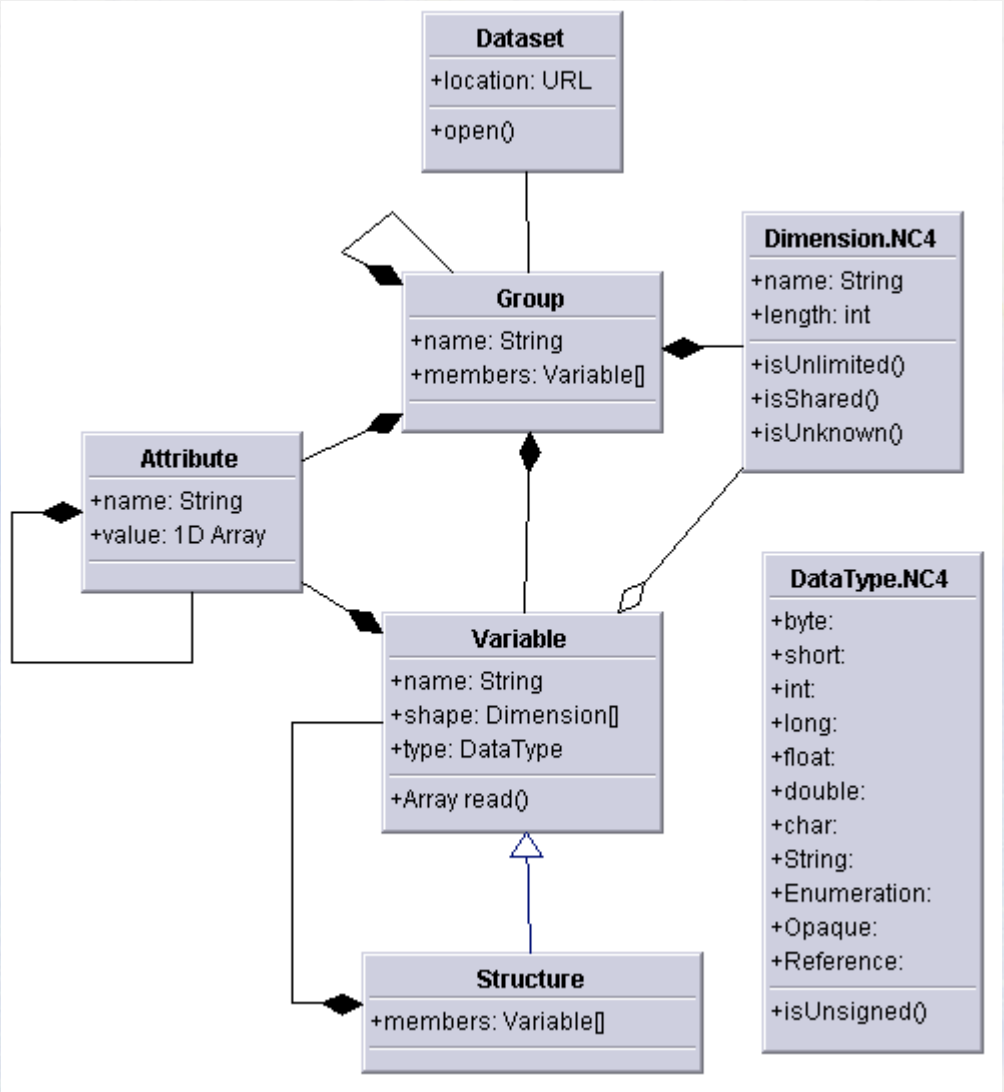
HDF5 Data Model

OPeNDAP

- Client-server protocol for scientific data access
- C++ client and server, Java client and server libraries.
- NetCDF-OPeNDAP client most popular (80/20)
- Current version 2.0 NASA ESE standard
- Working on new 4.0 protocol spec.
- Peter Cornillon (PI), James Gallagher (lead), et al, from Univ. Rhode Island



OpenDAP Data Model

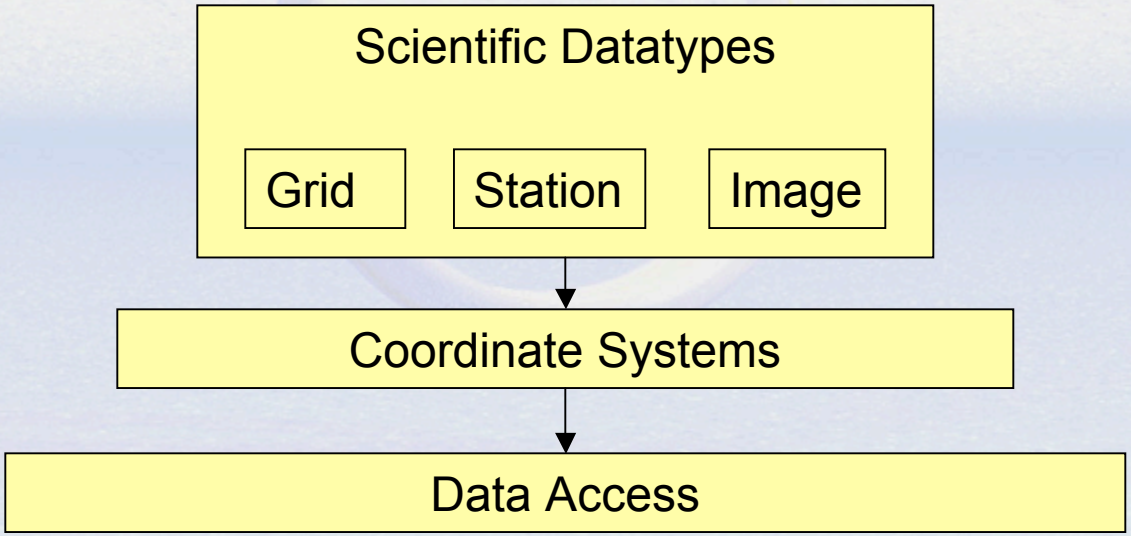


Common Data Model (CDM)

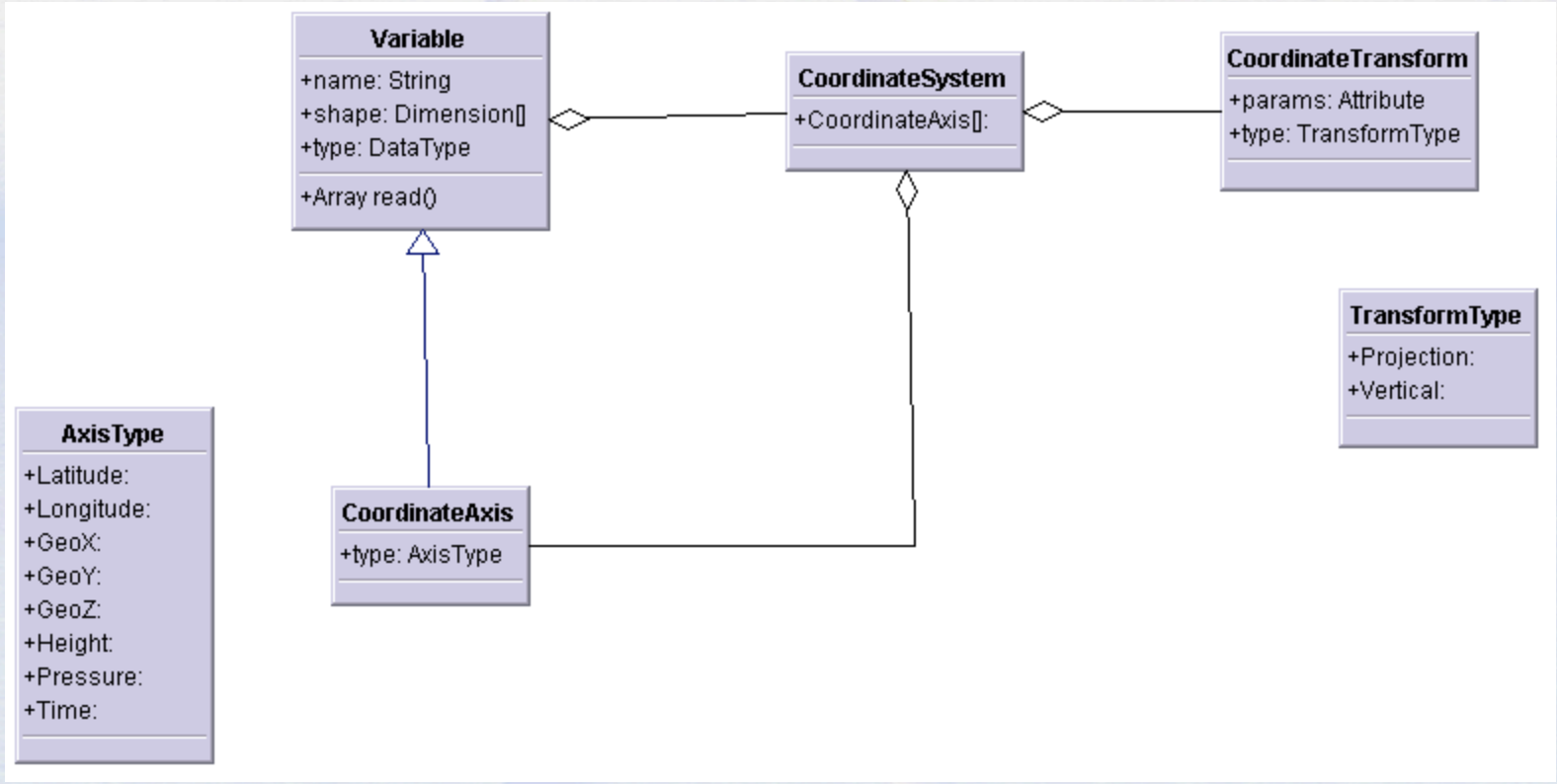
Abstract Data Models

- An API is the interface to the Data Model for a specific language
- A file format is a persistence format for the Data Model.
- A data access protocol plays roughly the same role as a file format.
- The Abstract Data Model removes the details of any particular API and the persistence format.

Common Data Model Layers



CDM Coordinate Systems

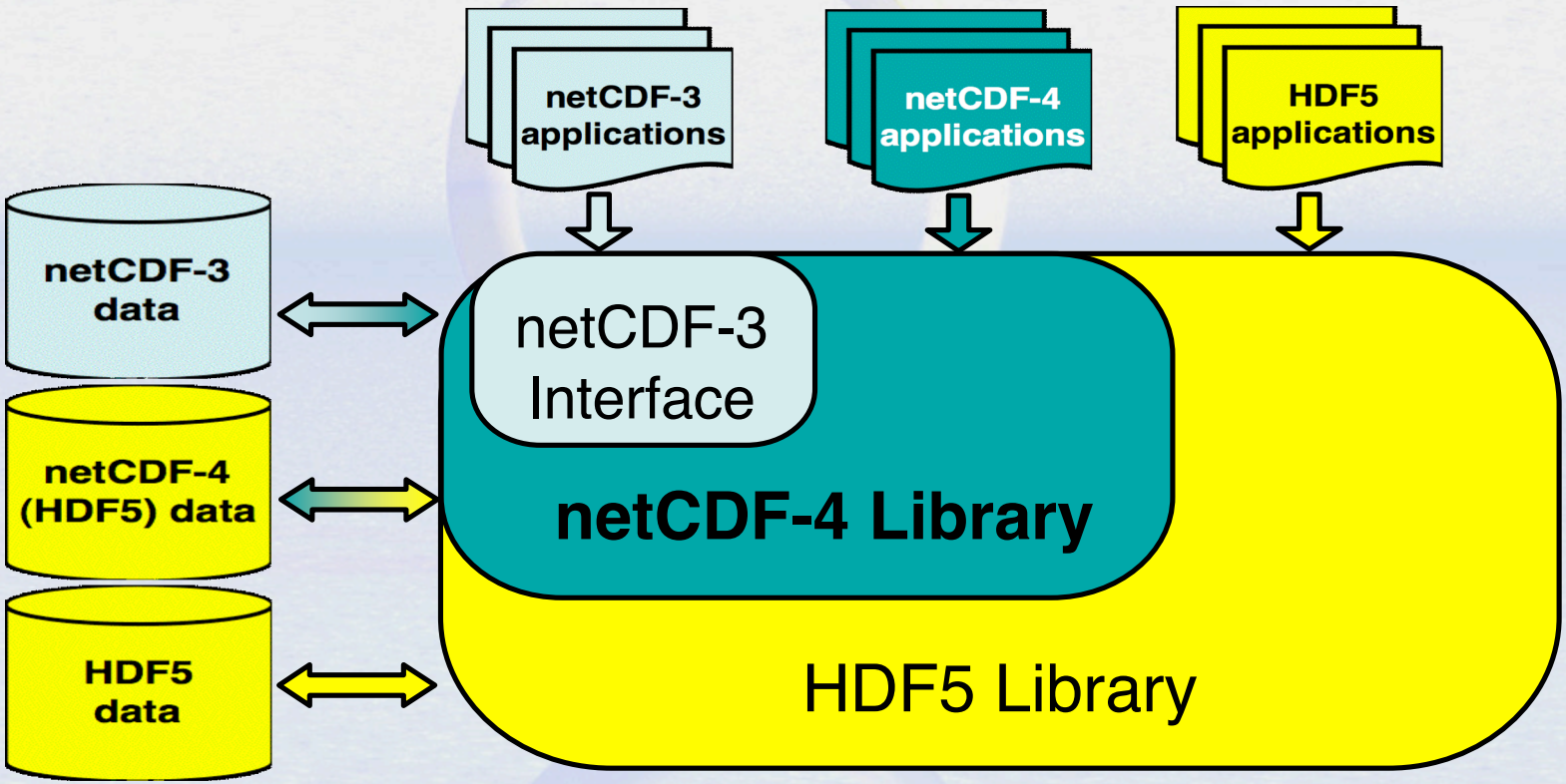


Implementing the CDM: Netcdf-4 NetCDF-Java 2.2

NetCDF-4

- Project funded by NASA to create new version of netCDF using the HDF5 file format.
- “Extend and merge” netCDF and HDF5:
  - Widespread use and simplicity of netCDF-3
  - Generality and performance of HDF5
- Specifically, we are funded to create netCDF-4 C library API, using HDF5 library underneath.
- Russ Rew (PI), Ed Hartnett

NetCDF-4 Architecture
NetCDF-4 C Library



NetCDF-4 and Java

- 100% Java library for netCDF-4 files possible?
  - Won't implement MPI parallel-IO
  - netCDF-4 features are a subset of HDF5
  - Reading easier than writing
- NetCDF-Java 2.1 already a 100% Java library for netCDF-3 files (and OPeNDAP)
- NetCDF-Java 2.2: read HDF5 to determine what netCDF-4 data model should be

Common Data Model

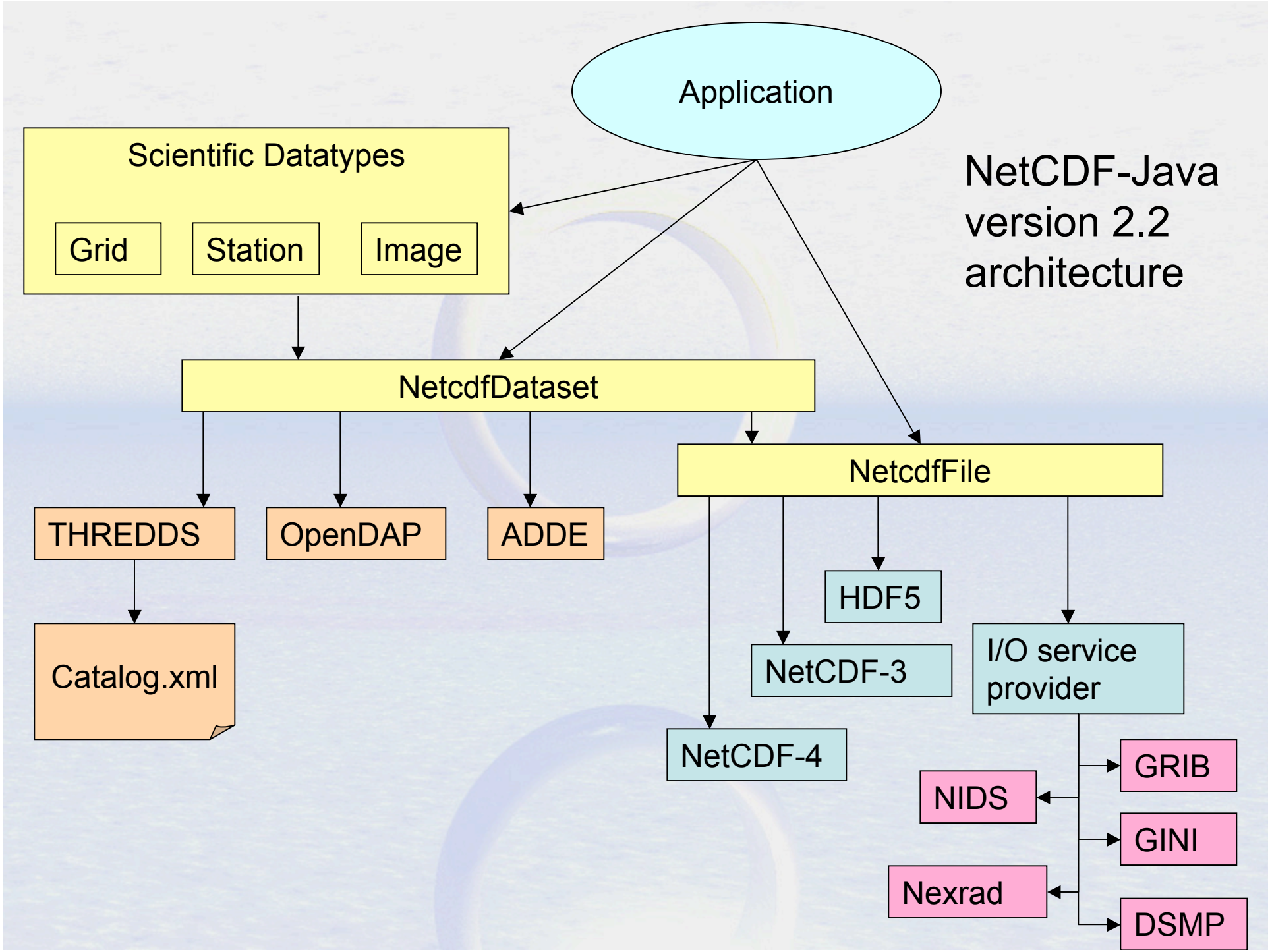
- NetCDF-Java 2.2: create one API (and data model) for access to netCDF-3, HDF5, and OPeNDAP: prototype for CDM.
- NetCDF, HDF5, and OPeNDAP groups are discussing a formal mapping between the three data models.
  - Opportunity to tweak the 3 data models to mitigate differences
  - Opportunity to make OPeNDAP 4.0 the remote access protocol for netCDF-4, and netCDF-4 the file persistence format for OPeNDAP.

Common Data Model

- NetCDF-Java 2.2 implements the CDM.
- NetCDF-4 C library will implement the CDM
- NetCDF-4 file format will be the persistence format for CDM.
- Caveats:
  - Not stable until C library and file format are finished (summer 05).

NetCDF-Java 2.2 (nj22)

- Alpha release: Nov 2004
- Beta release: Mar 2005
- Release: summer 2005


Application
Scientific Datatypes
Grid
Station
Image
NetCDF-Java
version 2.2
architecture
NetcdfDataset
NetcdfFile
THREDDS
OpenDAP
ADDE
HDF5
Catalog.xml
NetCDF-3
I/O service
provider
NetCDF-4
GRIB
NIDS
GINI
Nexrad
DSMP

I/O Service Provider Implementations

- DMSP (Defense Meteorological Satellite Program) from NGDC (*Ethan Davis*)
- GINI (national radar mosaic) (*Yuan Ho*)
- GRIB-1, GRIB-2 (*Robb Kambic*)
- NEXRAD level II (NCDC archives, CRAFT compressed)
- NEXRAD level III (partial) (*Yuan Ho*)
- NetCDF-3
- HDF5

Direct Grib reading – why?

- Grib is WMO standard, NCEP model data
- NetCDF/Grib file size = 6.6 to 40
  - Grib-1 has scale/offset compression
  - Grib-2 has JPEG2000 (wavelet), complex compression
- Existing decoder (grib2nc)
  - needs predefined CDL
  - No Grib-2 decoder
- Want the convenience of netCDF API without actually writing a netCDF file.

ucar.grib library

- Standalone Java library to read Grib files
  - Author: Robb Kambic
  - Grib-1: started with JGrib library, but rewrote
  - Grib-2: from scratch, uses jpeg2000 library
- Grib file = collection of Grib records.
- Write index file first time it reads Grib file.
- Tested with only IDD/NCEP data so far.
- Goal: allow others to extend by adding new tables without programming.
- Basis for future Grib decoders.

ucar.nc2.iosp.grib

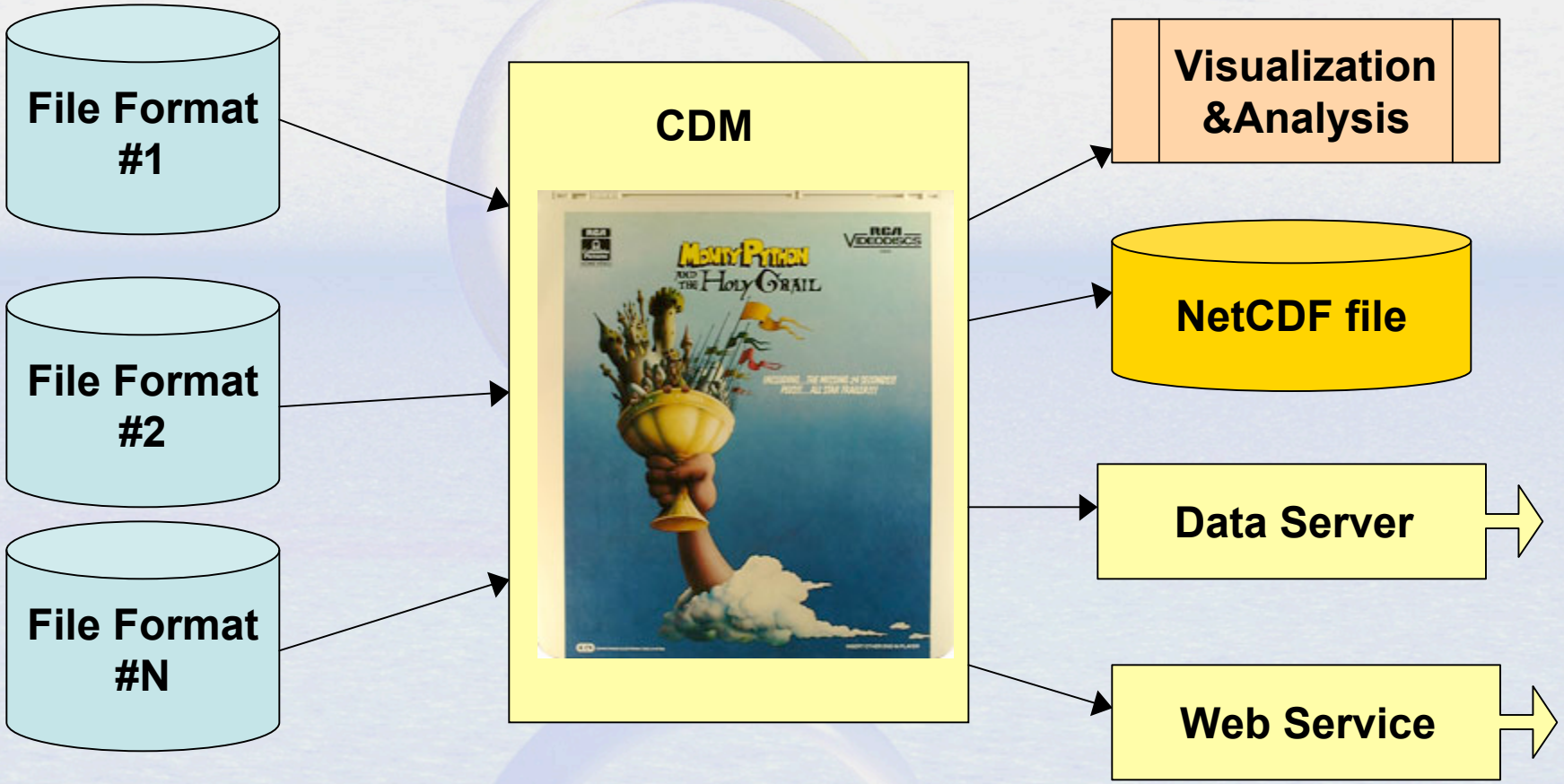
- Creates NetCDF / CDM objects on the fly.
- Collection of 2D arrays (Grib records) -> 5D dataset (netCDF). (not foolproof)
- Add CF-1 and _Coordinate Conventions.
- Looks like a CF compliant netCDF file.
- Can use FileWriter to write to netCDF file.

I/O Service Provider

Implement this interface:

```
public interface IOServiceProvider {

  boolean isValidFile( RandomAccessFile raf);

  void open( RandomAccessFile raf, NetcdfFile ncfile);

  Array readData( Variable v2, List section);

  // only if you use Structures
  Array readNestedData( Variable v2, List section);

}
```

Goal: N + M instead of N * M things on your TODO List



NcML
THREDDS

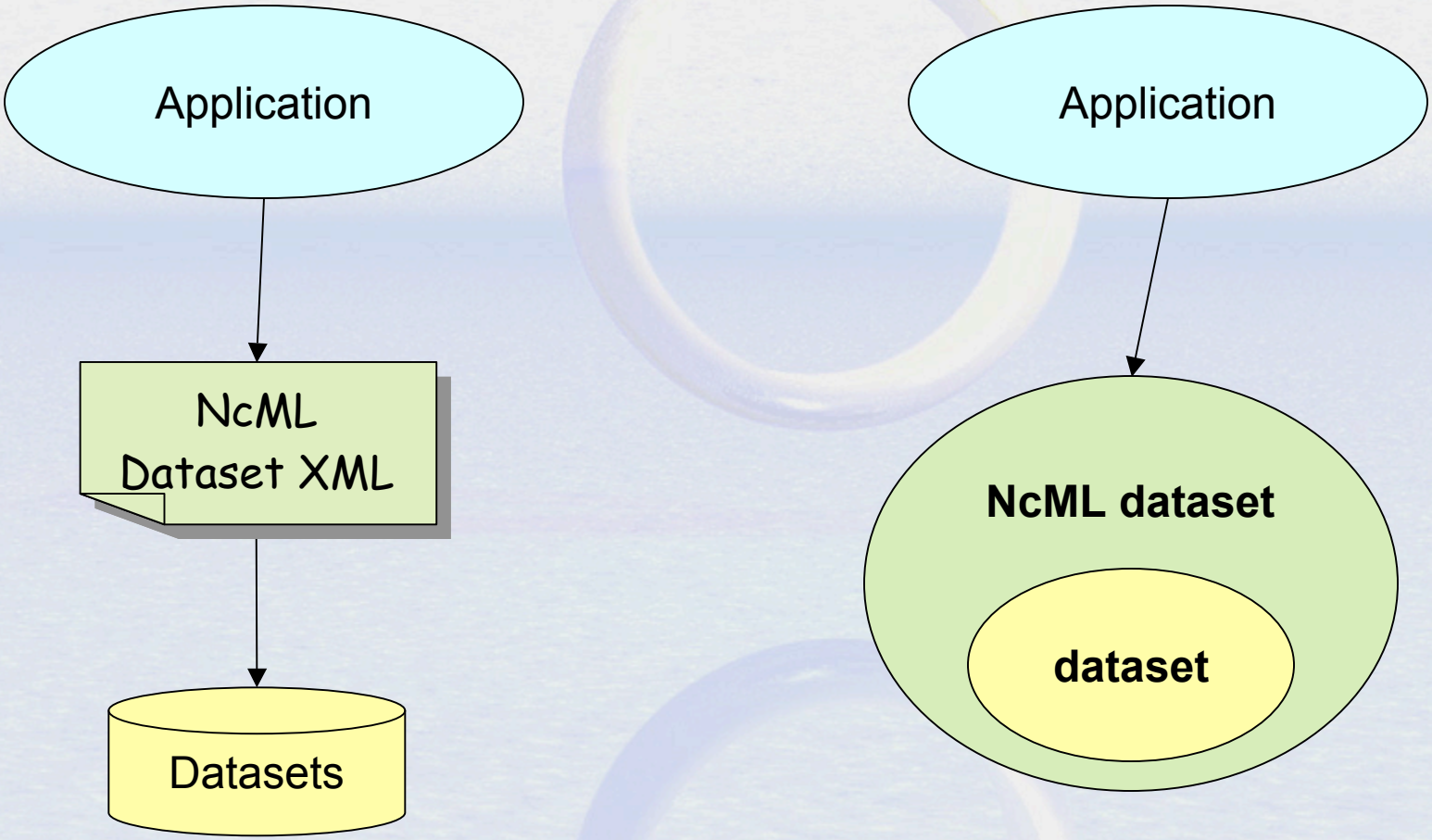
NcML - NetCDF Markup Language

- XML representation of netCDF metadata
- Create new files, like ncgen uses CDL
- Modify existing datasets
  - Add, delete, rename Attributes, Dimensions, Variables, Groups
  - Create logical sections of existing variables.
  - Create unions and aggregations of multiple existing datasets.

NcML example

```
<?xml version="1.0" encoding="UTF-8"?>

<netcdf xmlns="http://www.unidata.ucar.edu/schemas/netcdf/ncml-2.2"
   location="test/data/nids/N0R_20041119_2147">

  <dimension name="azimuth" length="367" />
  <dimension name="gate" orgName="bin" length="230" />

  <attribute name="latitude" type="double" value="39.786" />

  <variable name="Reflectivity" shape="azimuth gate" type="byte">
    <attribute name="units" type="String" value="dBZ" />
  </variable>

</netcdf>
```

NcML Datasets



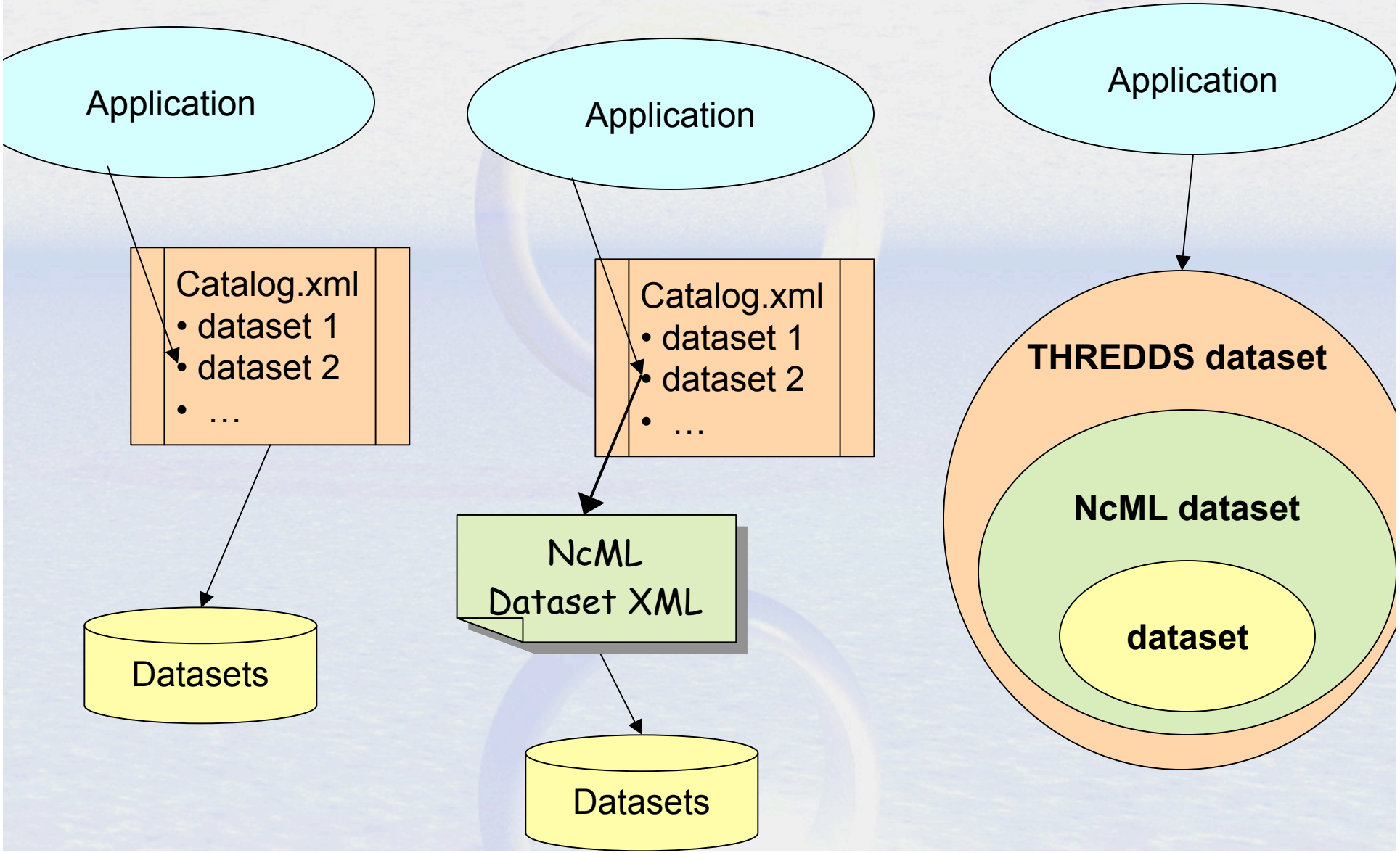
THREDDS Datasets

- nj22 library accepts URLs like

```
thredds:http://server:8080/thredds/catalog.xml#datasetId
```

- THREDDS metadata can be used to know how to read the dataset.
- THREDDS metadata can be added to the Dataset as global attributes.
- NcML can be applied to a collection of datasets in a THREDDS catalog

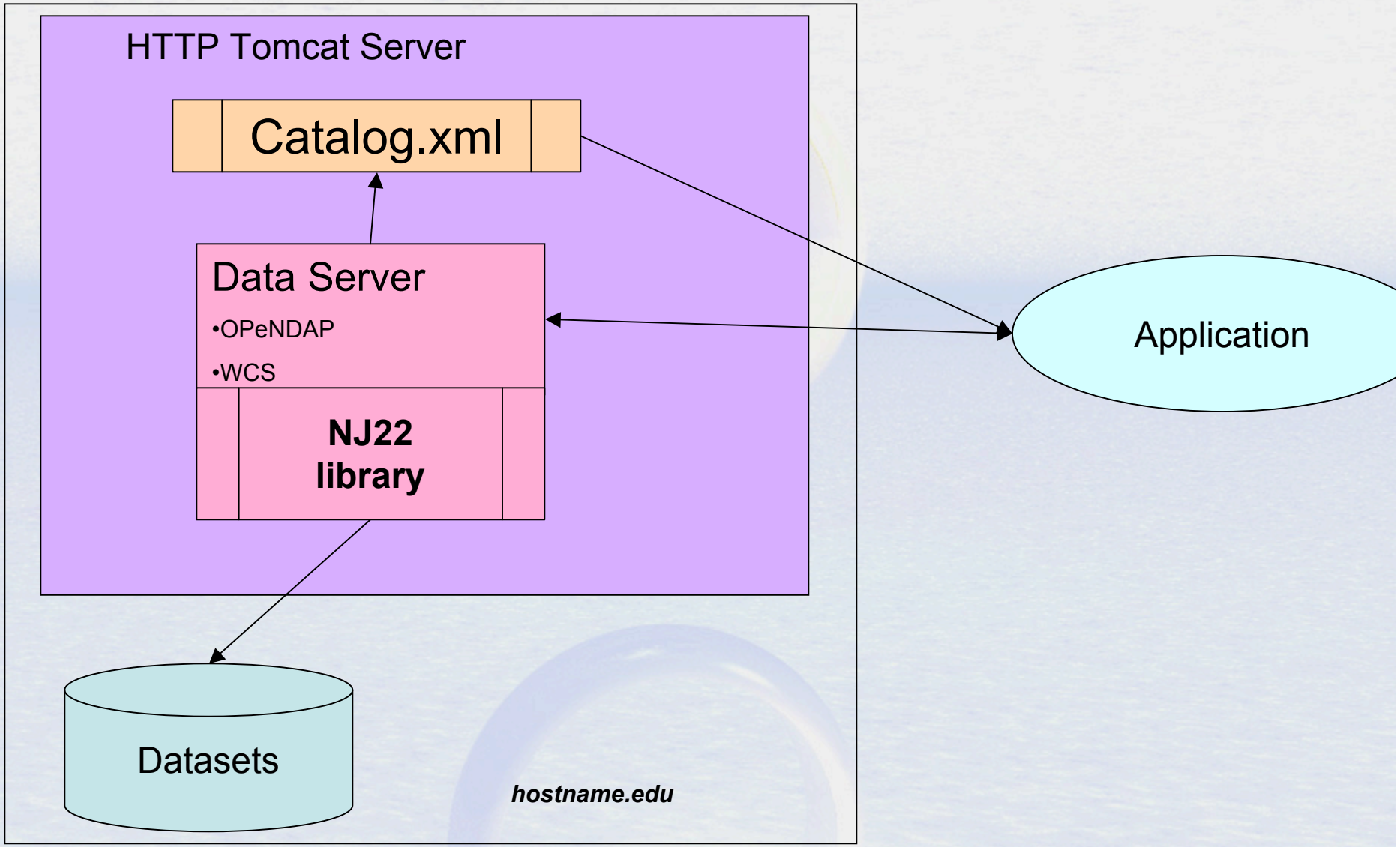
THREDDS Datasets



Limitations

- Currently this functionality is available only through the netCDF-Java library.
  - NcML will probably eventually become available in the C library.
  - Not sure about THREDDS catalogs
- So your client has to be written in Java

THREDDS Data Server



Summary

- NetCDF-4 will have an extended data model based on experience with netCDF-3, HDF5 and OPeNDAP.
- Lack of shared Dimensions biggest problem in mapping to other models.
- Currently available in alpha version of netCDF-Java 2.2 library.

Next Time

- Coordinates
- Scientific Data Types
- OpenDAP as remote access protocol for netCDF-4?

Warning! Danger!

- This is alpha quality, API still evolving!
- Please use and influence us:
  - Testing with real datasets
  - Convention parsing
  - IOServiceProvider