NcML Aggregation vs Feature Collections

NcML functionality

1. Modify the objects found in CDM files
   - Especially Attributes
   - Don't have to rewrite the files, esp on the server
   - Make modest changes "by hand"
2. Alternative to **ncgen, ncdump**
   - CDL in XML
3. Aggregations
   - Creating logical datasets out of file collections

NcML Aggregations

- **union**
  - Logical union of objects inside multiple files
- **joinNew**
  - Each file contains one slice – add new dimension
- **joinExisting**
  - Concat existing multidim arrays together
- **tiled** (never released)
  - Stitch horizontal tiles together
- **FMRC** (deprecated as of 4.2)

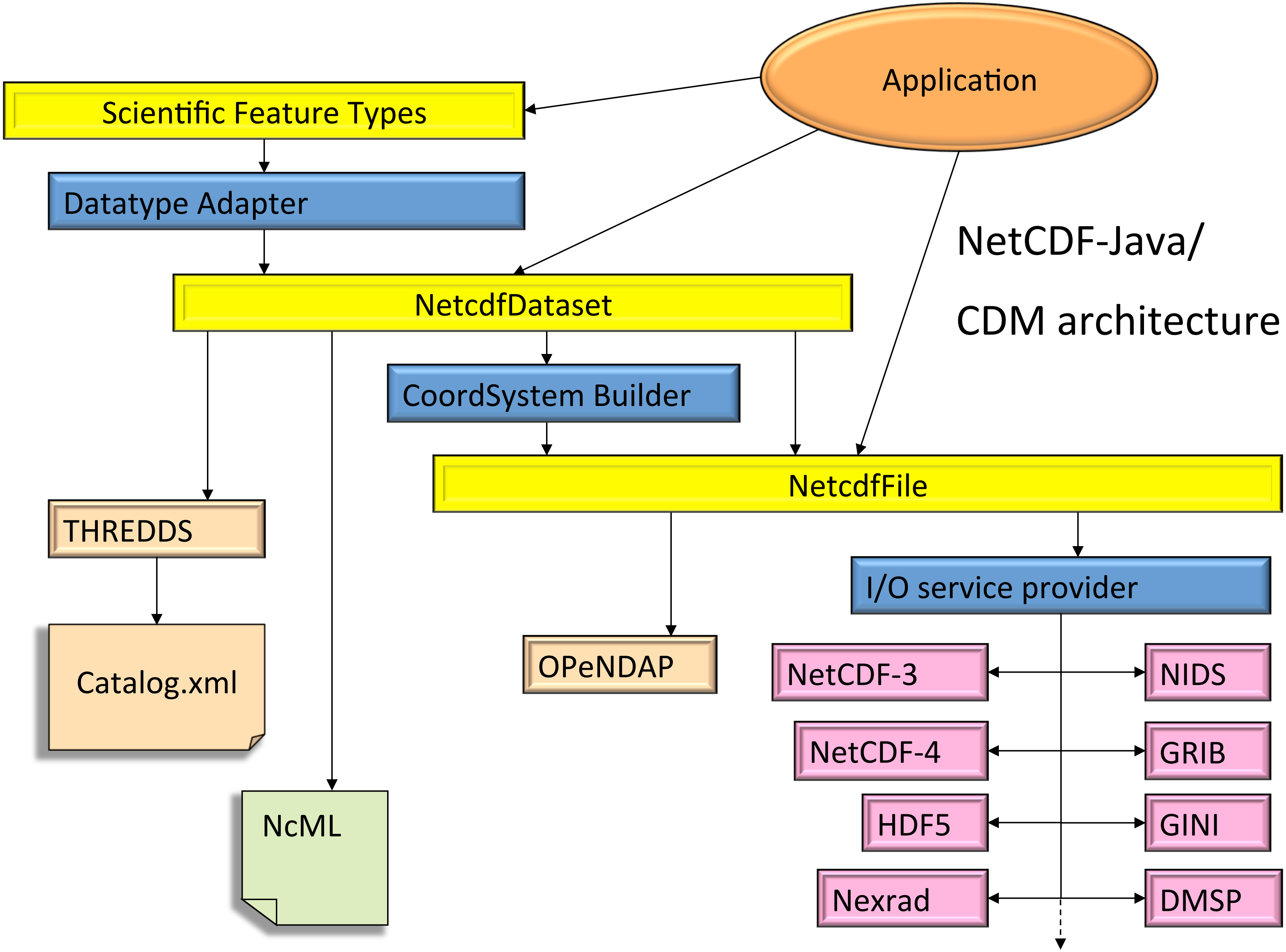
NcML Aggregations

- *Syntactic* aggregation
  - Manipulating multidimensional arrays
  - Doesn’t know what the data means
- High *homogeneity* requirements
  - Files must be identical, except for certain things
  - Difficult for Users to understand or to track
- Assemble at runtime
  - Inefficient for large collections
  - Caching strategies cause trouble for dynamic datasets

Feature Collections

- Feature Types
  - Coverage: grid, swath, FMRC, *ugrid*, image**
  - Discrete: point, station, profile, etc
  - Radial
- Understands more of the dataset semantics
  - Mostly about the coordinate systems
- Can "do the right thing" without user
  - Handle more inhomogeneity in the files

* Coming soon



NetCDF-Java/

CDM architecture

What do you need to know?

- NcML aggregation works and is stable code
  - Must know how your collection of files varies
  - Use it for simple cases
- Feature Collections are still work in progress
  - Much easier to work with
  - All future development is here
  - You have our attention
- If you have GRIB files
  - You want to use GRIB feature collections (4.3)

Good Luck!