

# THREDDDS Technical Summary

John Caron

May 6, 2002

# Who Are We?

## Data Providers

- NOAA/NGDC
- NOAA/PMEL
- UAH/ITSC
- CRAFT/CAPS
- UnivOK
- FNMOC/GODAE
- Lamont/IRI
- NOAA/NOMADS
- UnivWM/SSEC
- NOAA/CDC
- NCAR

## Data Clients

- LAS
- EDMII
- INGRID
- MetApps

## Data Centers

- GCMD
- DLESE
- NSDL

## Technologies

- ESML
- GrADS
- DODS
- ADDE

# Workshop Overview

- Technical Summary
- Themes
  - Data Providers
  - Application Developers
  - Discovery Centers
  - Metadata/OpenGIS
- Breakout Sessions
- What's next?

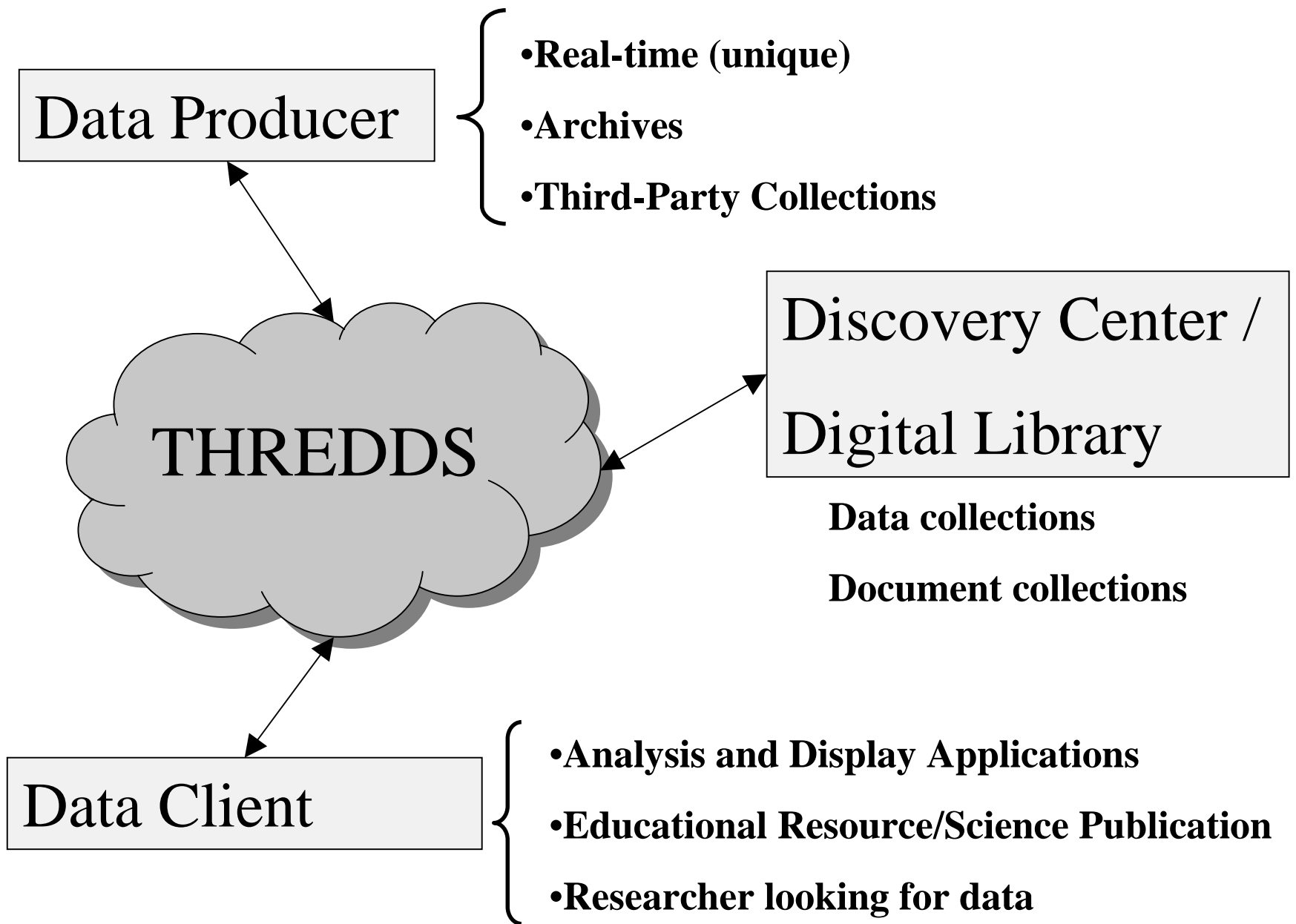
# Technical Summary

- High-level Overview
  - Technology Directions
  - Development Phases
- Phase 1 Details
  - Dataset Catalog XML
  - What is a Dataset?
  - Dataset Description XML
- THREDDS Data Model

# Mission Statement

- “Develop a software framework that allows educators and researchers to **publish**, **locate**, **analyze**, and **visualize** a wide variety of Earth science data.”

# THREDDDS Overview



# Working Definitions

- ***Data Producer***: “owns” a collection of datasets, makes them available on-line.
- ***Data Client***: software that reads data, documents that access data.
- ***Discovery Center***: provides browse and search services across multiple collections.
- ***3<sup>rd</sup> Party Providers***: provide logical dataset collections / additional metadata.

# Technology Focus

- Data available on-line, via Internet.
- Acquire data, not just pictures of data.
  - THREDDS clients create pictures, etc.
- Framework for “loosely coupled” systems
  - Distributed, exposed semantics
  - Emphasis on Services and APIs
- “Human in the loop” automation tools

# Current Technology Choices

- Metadata: XML over HTTP
- Data: DODS, ADDE, NetCDF servers
  - efficient access to entire dataset metadata
  - efficient data subsetting
  - “access protocol” analogous to file format
- Prototype client libraries in Java

# Future Technology Choices?

- SOAP for RPC communication
- WSDL for service definitions ?
- OpenGIS for services, data types ???
- Flexibility in supplying metadata records
  - DC vs DLESE vs DIF vs ISO

# Phase 1– Data Catalogs

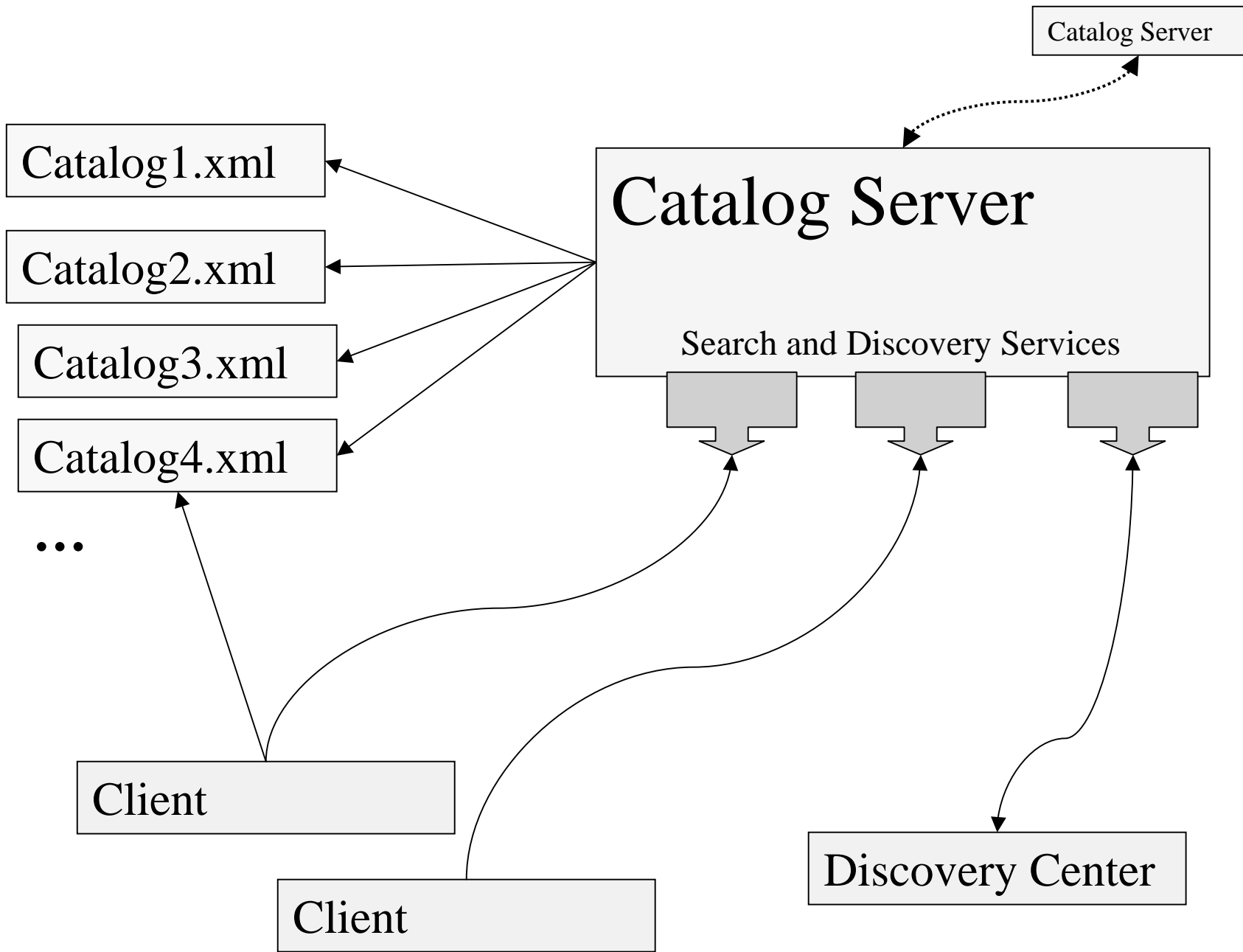
1. XML documents on web servers
2. Minimal metadata
  - mostly from Data Provider's POV
3. Create catalog generators for existing data archives and servers
4. Prototype clients: feasibility, performance
5. Get Providers using the tools, make it work(!) for important, real datasets

# Phase 1 Issues

- Dataset granularity
  - Number of Datasets, size of Catalogs
  - User “mental model” mismatch
- Real-time Dataset frequency
  - Catalog updating

# Phase 2 – Catalog Servers, DL

- Catalog Servers
  - consolidate individual catalogs
  - Active process allows e.g. notification, push, record transfer to Discovery Centers, etc
  - provide geographic / keyword search
  - other web services, prob. using SOAP
- Augmented metadata for Discovery Centers



# Phase 2 Issues

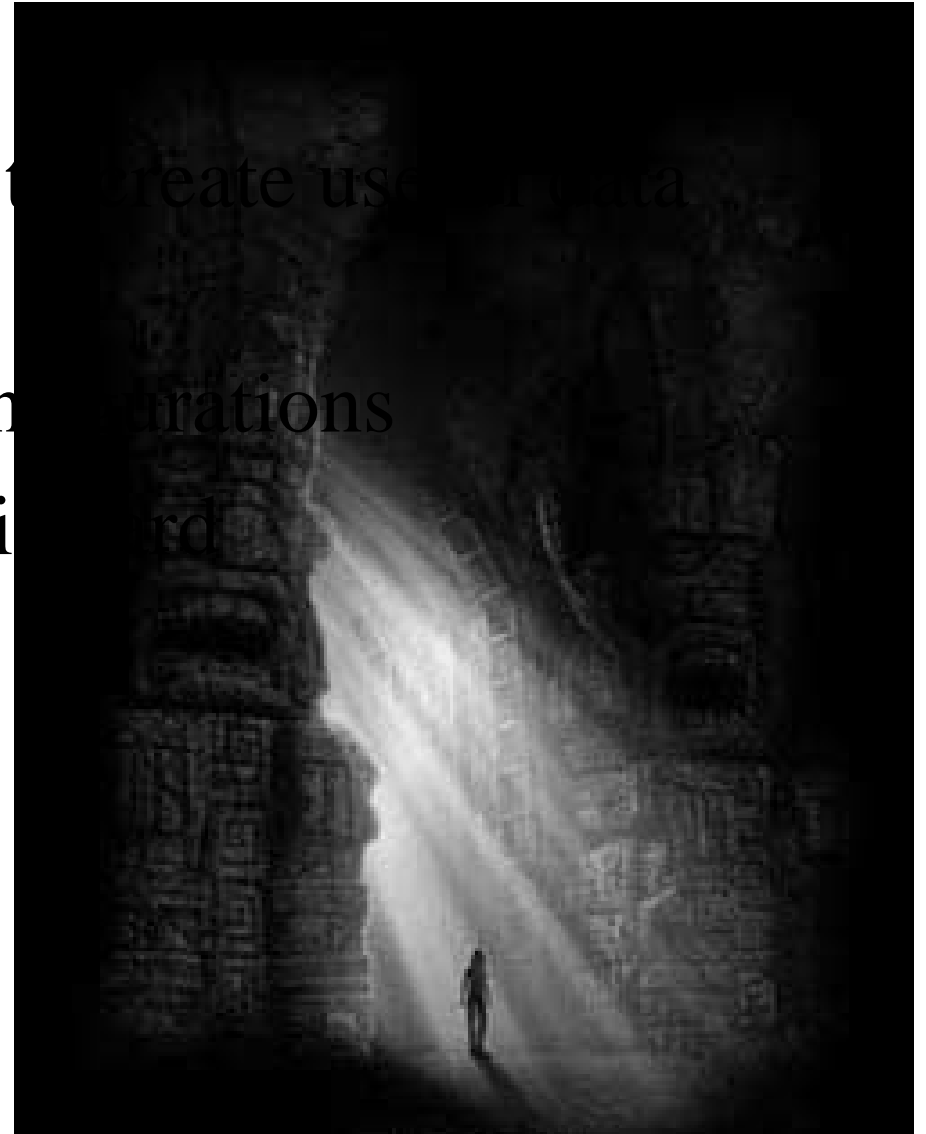
- Efficient, flexible queries
  - Large # datasets overwhelming
  - Efficient geography search
  - What should be returned: dataset granularity
- How to create desired metadata records
- Distributed data: propagate changes

# Phase 3 – Data Semantics

- Create tools that allow data classification and data attribute ontologies
  - Standard quantity controlled vocabularies
- Allow “third-parties” to classify, annotate, logically group, “add value” to datasets
- “Collaborative Knowledge Building Environment”

# Phase 3 Issues

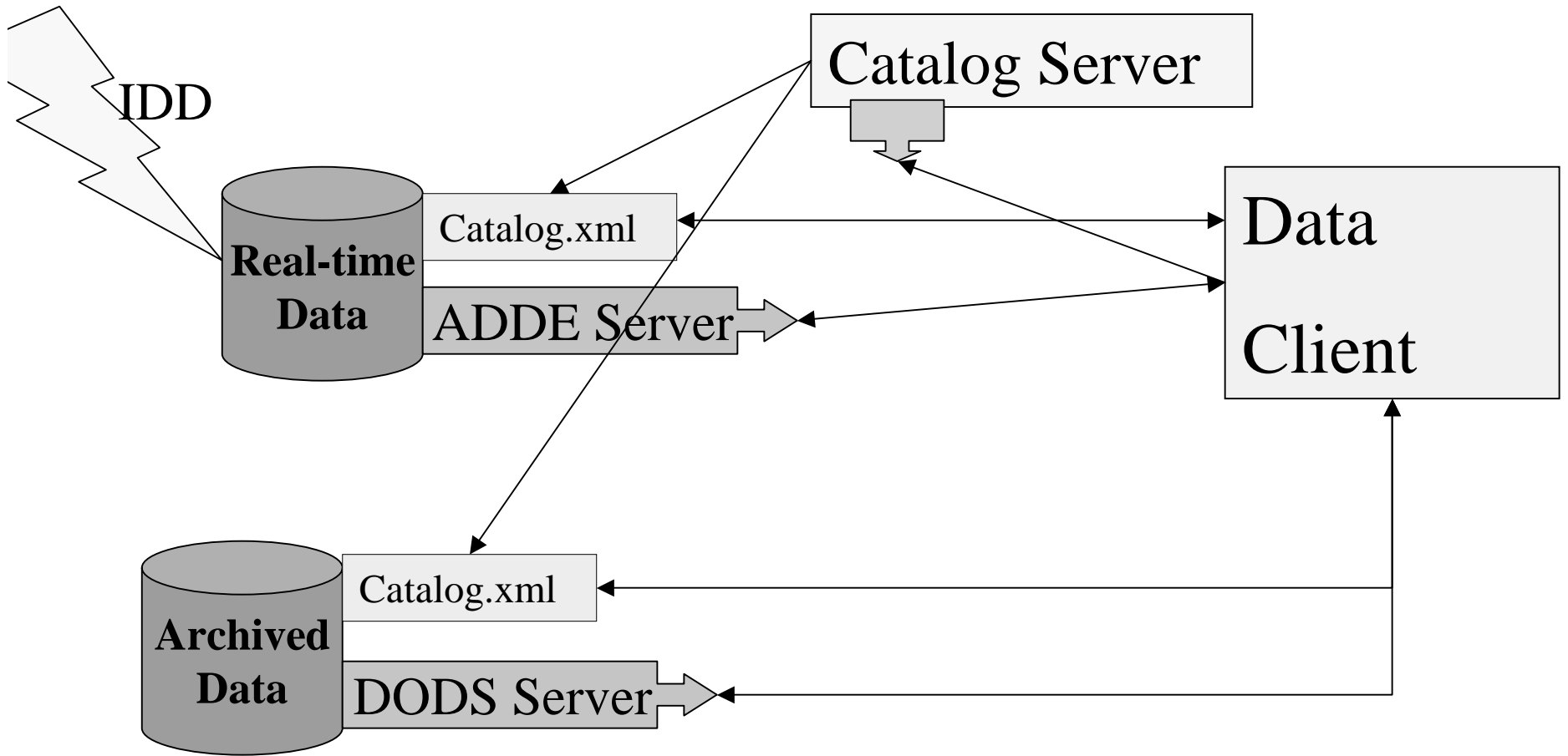
- Data common enough to create useful data classification?
- Managing multiple configurations
- Creating quality tools in the field
- Uncharted territory



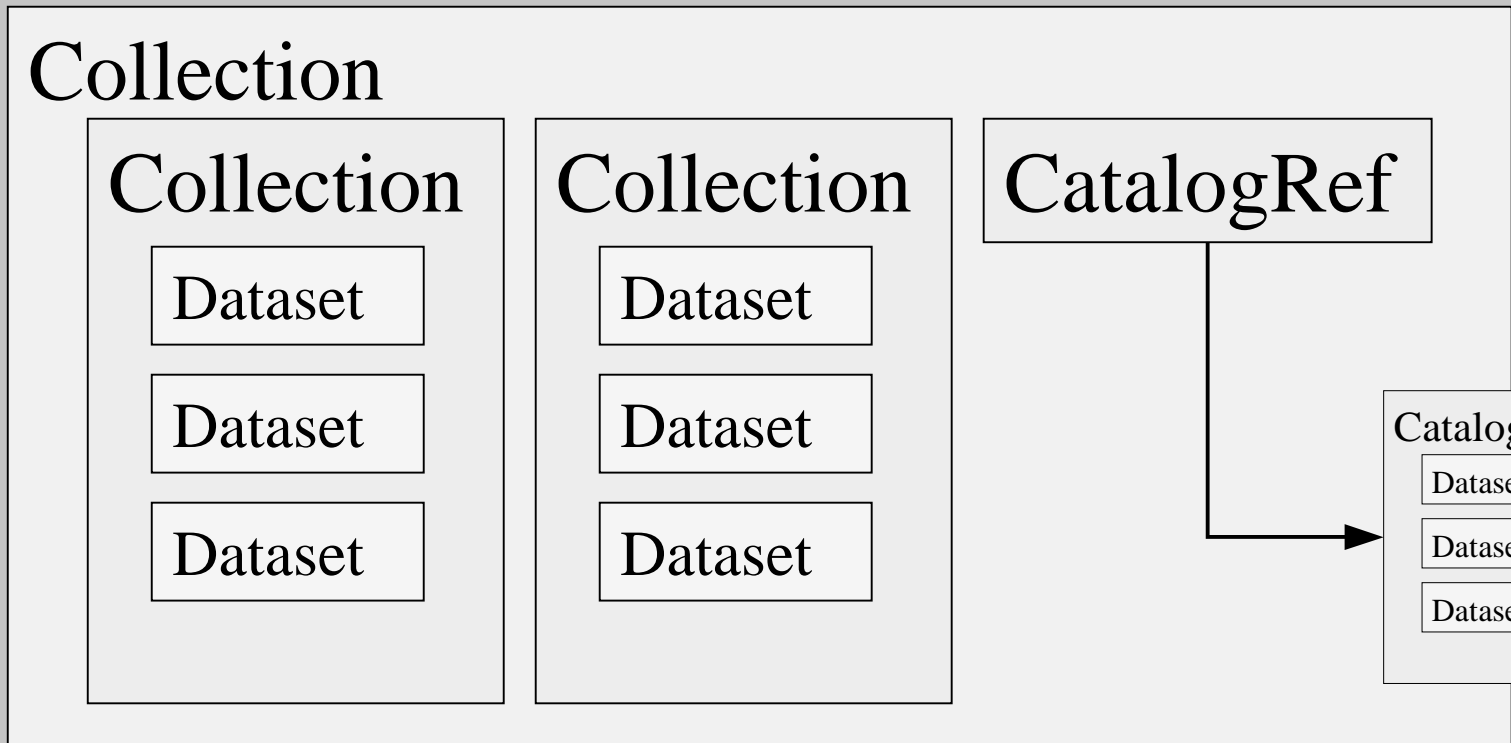
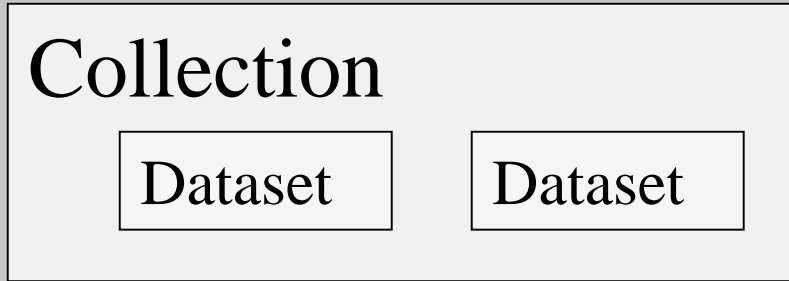
# Development – Phase 4

- Solve Israeli-Palestinian conflict
- Develop Cold Fusion
- Red Sox play Chicago Cubs in World Series

# Dataset Inventory Catalogs



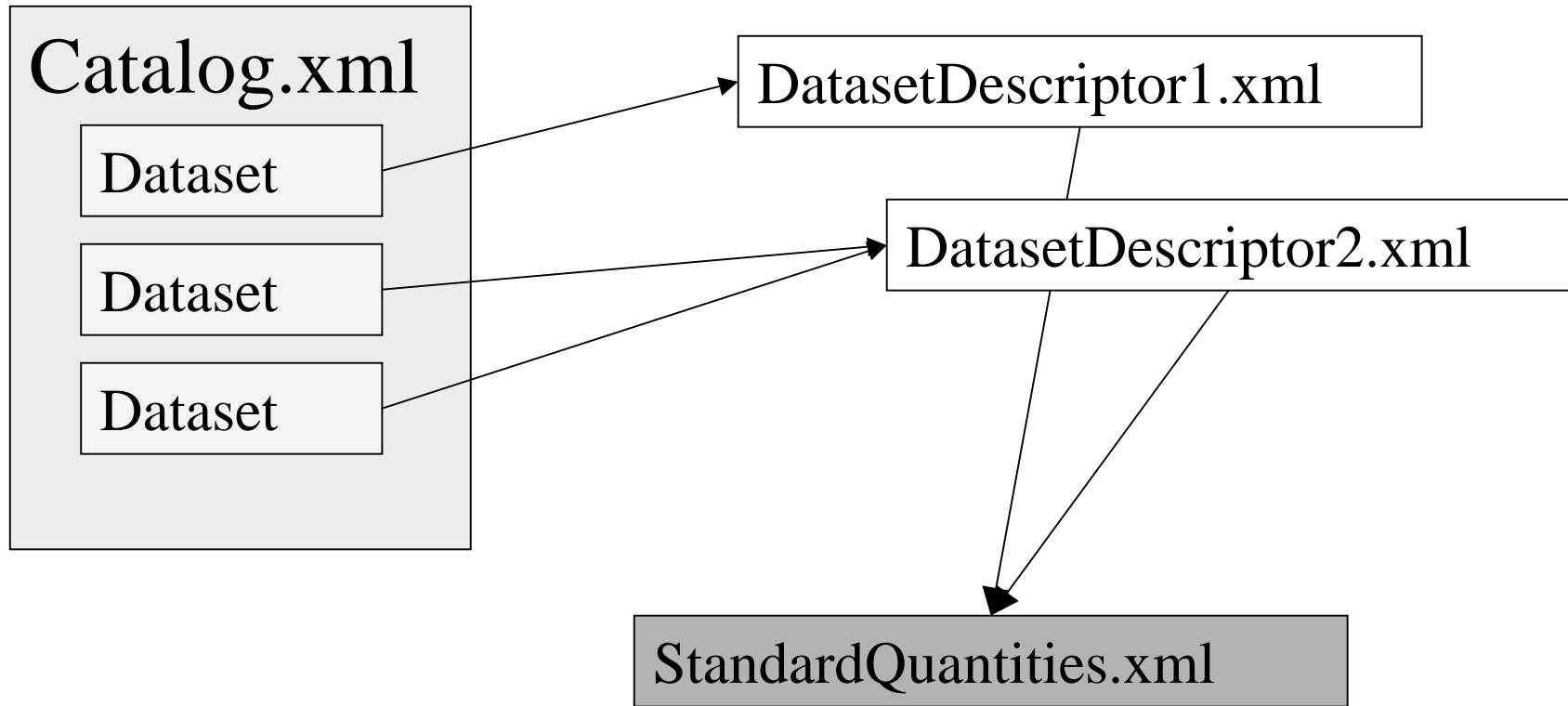
# Dataset Inventory Catalog



# Dataset Catalog XML

- Group datasets into hierarchical collections
- Dataset has a displayable name and a URL
  - Optional link to a *Dataset Description*
- Optional comments, links to more info.
- May link to other catalogs, these become nested collections

# PICats



XML over HTTP

# Dataset Catalogs Examples

- XML
- THREDDS Data Viewer
  - GRIDS
  - Images
  - Stations

# What is a dataset?

“Any meaningful collection of data”

“Data in a single physical file” (NetCDF)

“The abstract thingy that lies behind a URL”  
(DODS)

“Grouping of fields, approximately collocated  
in space and time, to enable derived  
quantities and other analysis”

# Issue: Dataset Granularity

- Discovery Centers don't want huge numbers of identical records.
- Data Providers want to comprehensively catalog their data.
- Data Clients can be confused/frustrated if granularity is wrong.

# THREDDS Granularity POV

- Want to allow users to browse catalogs
  - fast enough UI to get/ display/ browse catalog
- Want to use catalogs for DL entries, search
  - Must prevent overwhelming details
  - Use catalog, collection, individual dataset?
- DODS, ADDE allow efficient subsetting

# Granularity solutions

1. Datasets should be logically aggregated into the “right” level of granularity
  - DODS Aggregation Server
2. Dataset grouping
  - Catalog Collections
  - Nested Catalogs can be generated on-the-fly
  - Dataset Descriptions allow grouping of ADDE elements into datasets
3. Different Types of Catalogs?

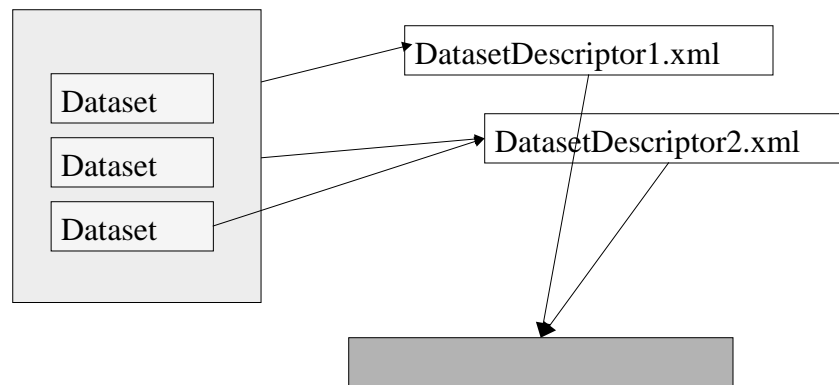
# DODS Aggregation Server

- Java-DODS Servlet under Tomcat
- Integrated THREDDS Catalog
- Aggregates DODS datasets or NetCDF files:
  1. One time step per file
  2. Variables written to separate files
  3. Concatenate multiple-time step files

# Catalog Summary

- Hierarchical collections of datasets
- Nested catalogs using XLink
- Minimal metadata requirements; retrofit existing servers
- Complex metadata factored using XLink
- DTD had 5 iterations

# Dataset Description XML



# Goals

- Missing, non/standard “use” metadata
  - Standardize, clarify semantics for clients
- Data access parameters for ADDE
- Enable fast catalog search on space/time, standard quantities
  - Don’t have to read datasets, just metadata

# Data Types

- Grid, Image, Point
- Station is a compound type
- More extensive types eventually

# Grid DatasetDesc XML

- Describe Georeferenced Coordinate System
  - x, y, z coords: can derive bounding box
  - static for model datasets
- Describe Time Coordinates
  - Static for archives, changing for LDM feeds
- List of Fields
  - Optional mapping to Standard Quantity

# Image DatasetDesc XML

- ADDE Servers needed to add “AccessPath” for fields, times
- Optional Coordinate Systems
  - Sattellite image navigation not always simple
- List of Times Coordinates
- List of Fields
- Allows fast access to ADDE holdings

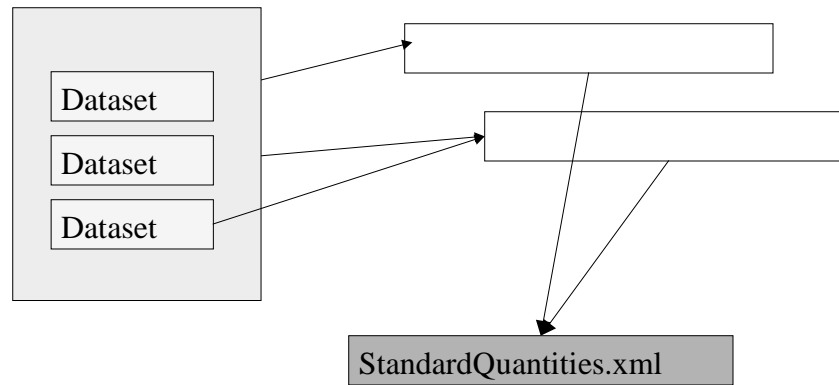
# Station DatasetDesc XML

- Real-time radar feed: frequent, intermittent, voluminous
- Dataset = {Station} X {Field} X {Time}
- Station = named location
- Times = logical ranges, e.g. “last 3 hours”
- Create query to get actual list of datasets
- Hope to generalize to other station datasets

# DatasetDesc Summary

- Optional, but needed for search, prob. DL
- Design not stable, complex
  - Assumptions may not work
- Building automatic generators
- Need to understand LAS, ESML, OpenGIS, other efforts

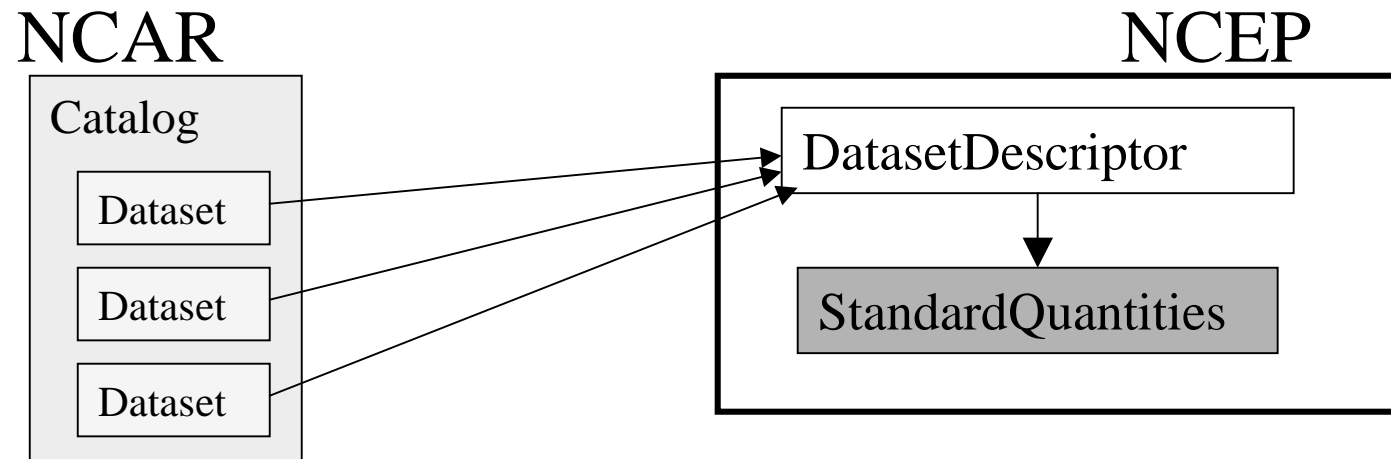
# Standard Quantity XML



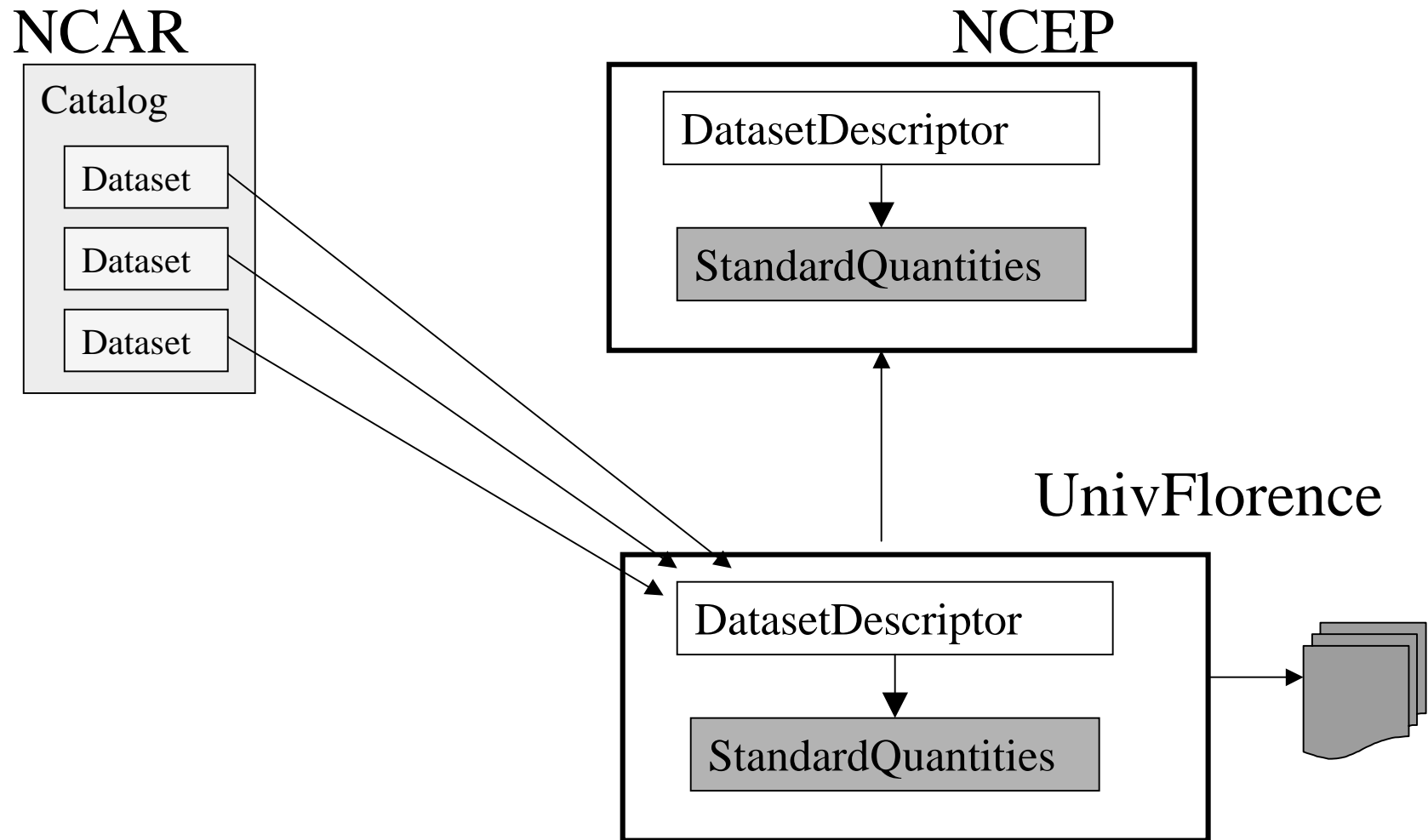
# Standard Quantities

- Mapping of dataset field names to controlled vocabulary
- Not necessarily centralized
- Expect authoritative vocabularies will emerge

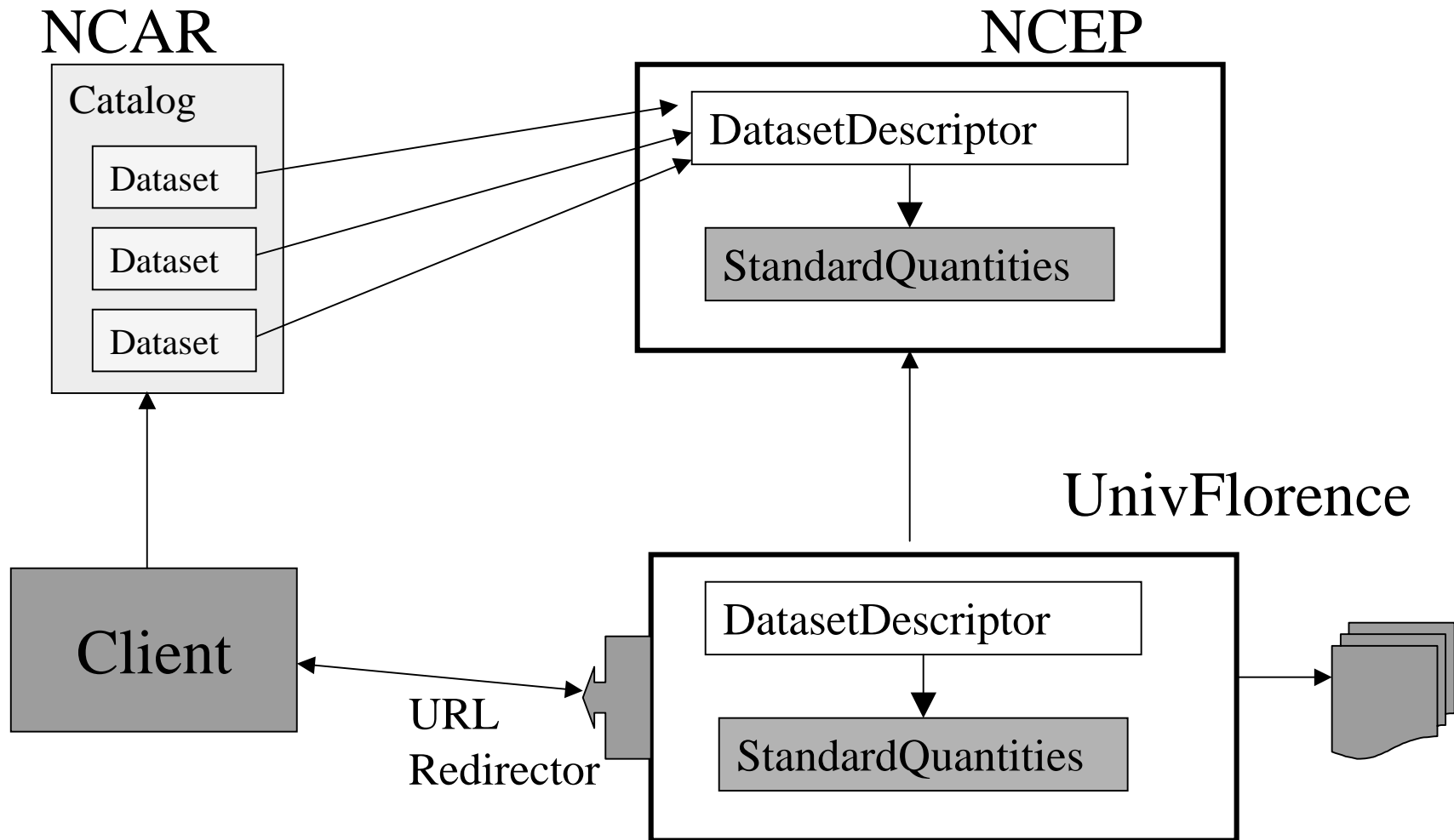
# Example 3<sup>rd</sup>-Party Metadata



# Redefine SQs w/ Data Provider

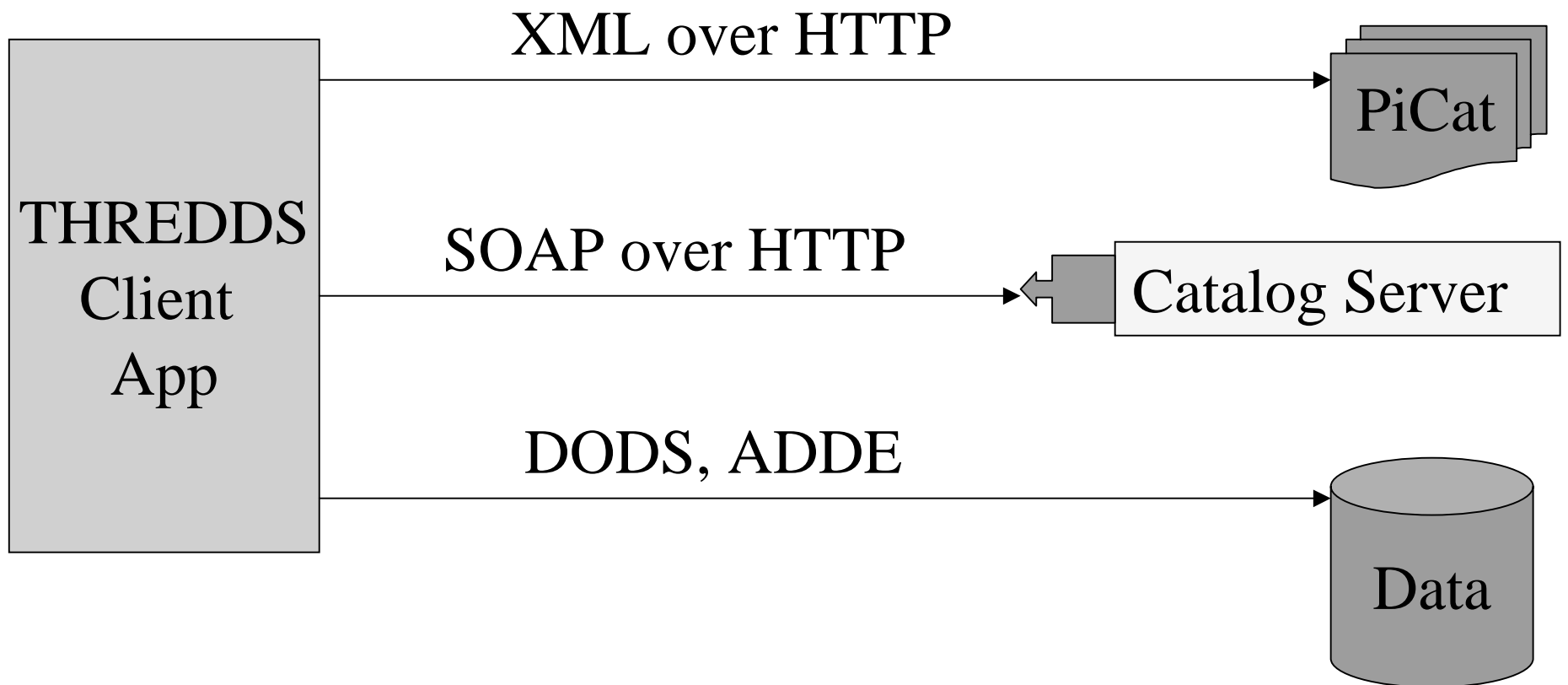


# Redefine SQs w/ Data Client

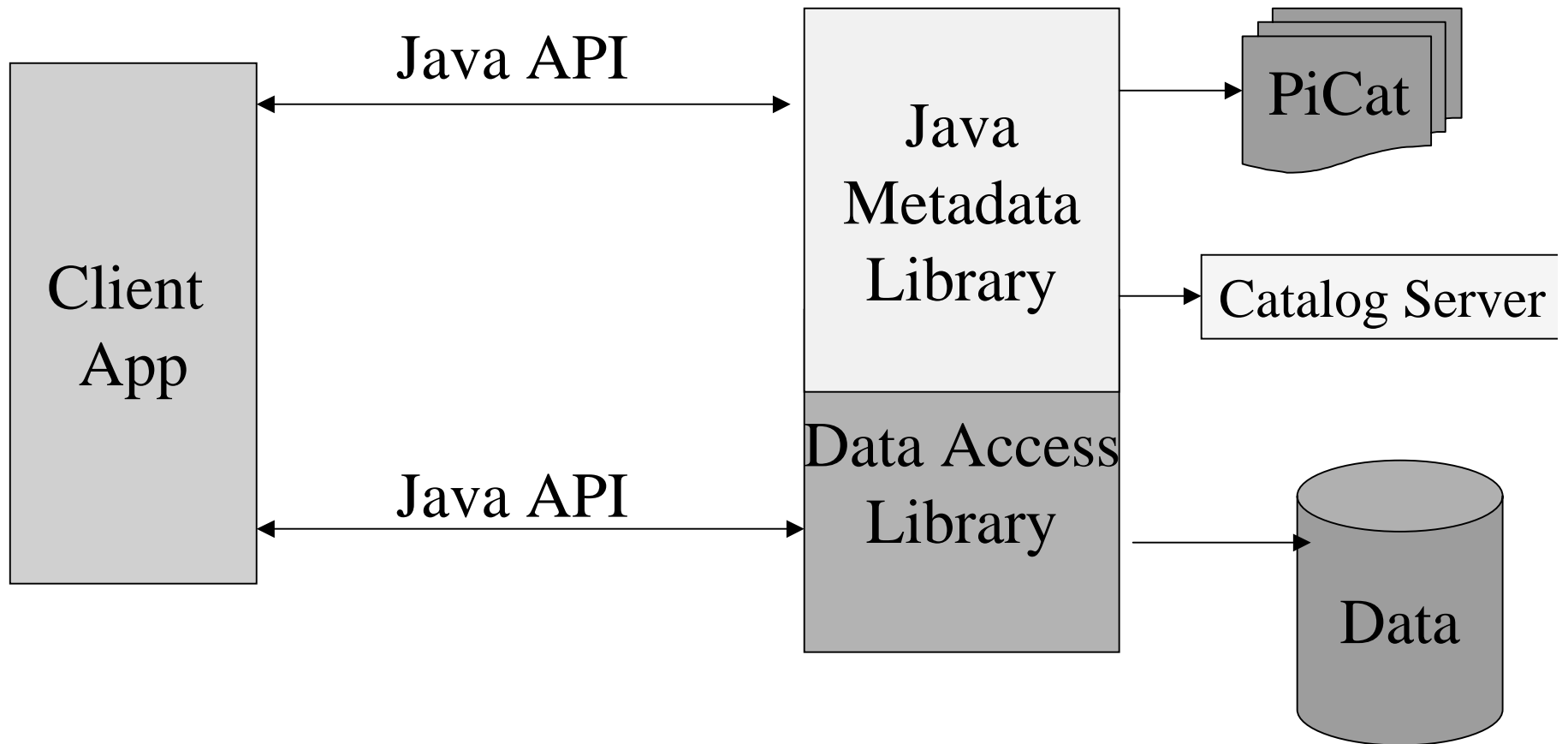


# THREDDS Data Model

# Direct Client Access

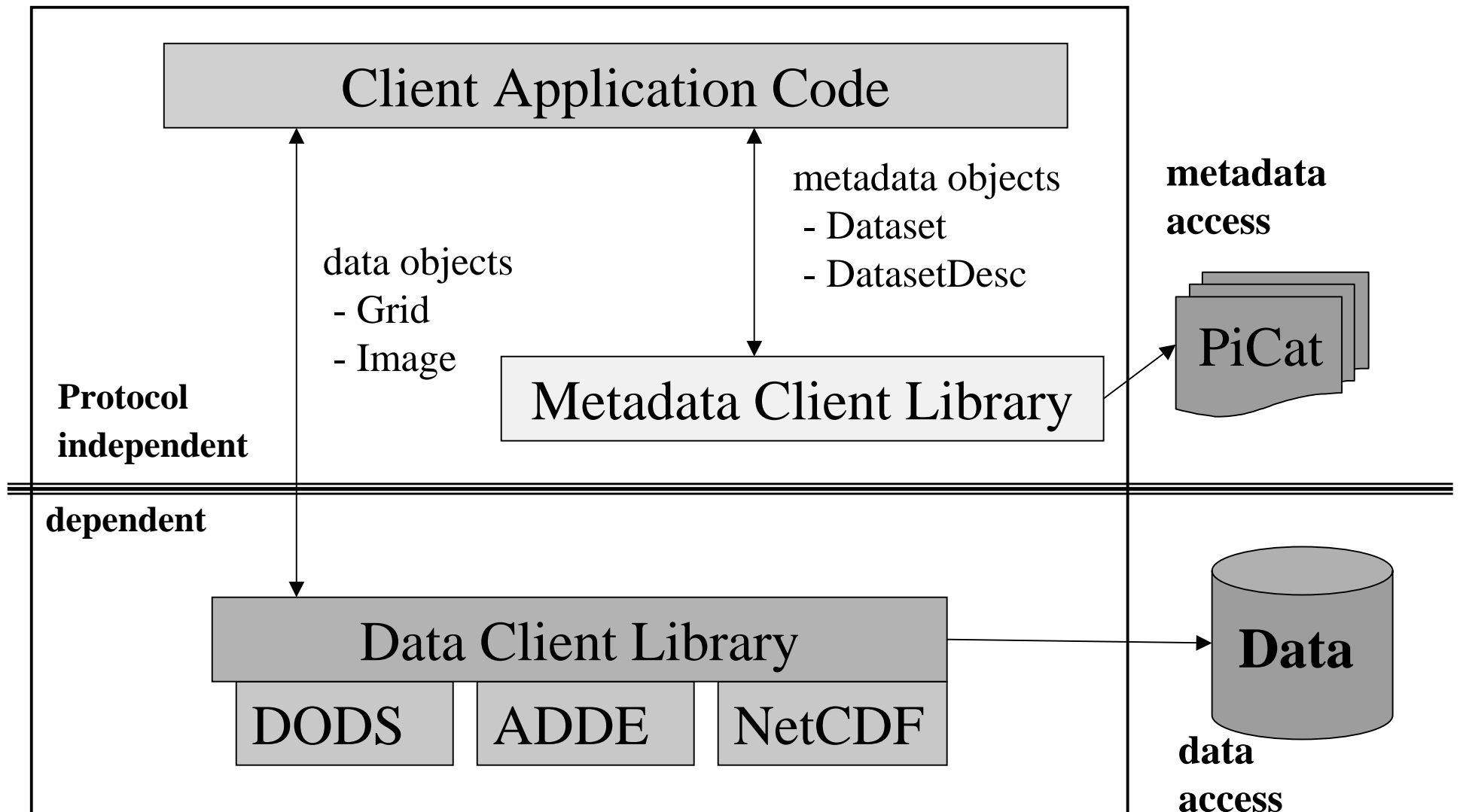


# Java Library Client Access



# Data vs. Metadata

Crossing the protocol boundary



# What is the Data Object Model?

1. None: we only do metadata, clients must do
2. Adopt OpenGIS or ISO model, convert data
  - Can it handle subsetting? Any clients?
  - Stefano: modifications to handle subsets
3. Adopt VisAD model, convert data
  - Java only
4. Ad-hoc minimal model (current)
  - Subsetting, hide protocol, Grid, VisAD Adapters
5. Partial, provide helper classes

# Data Model – Goals and Issues

- Use existing/emerging standards
  - Efficient handling of large datasets
  - Simple things are simple
  - Clients that can use it
- Hide the protocols, clean API
- Possible to create C++ library eventually

# THREDDS Tech status

<http://www.unidata.ucar.edu/projects/THREDDS/tech/>

# Dataset Characteristics

- Data Type : semantics
  - Grid, Image, Point, Station
- Server Type : access protocol
  - DODS, ADDE, NetCDF, Catalog
- Data frequency
  - Static: archived datasets
  - Dynamic, periodic: NCEP model output
  - Dynamic, frequent: NEXRAD level 3

# PiCAT Generator Tools

1. ADDE Cataloger
  - ADDE, Image, ~ 2 min
2. Catalog Generator/Scanner
  - DODS, Grid, ~ hour
3. Dynamic Station Catalog Generator
  - Catalog, Station, ~ 10/sec

# Data Clients

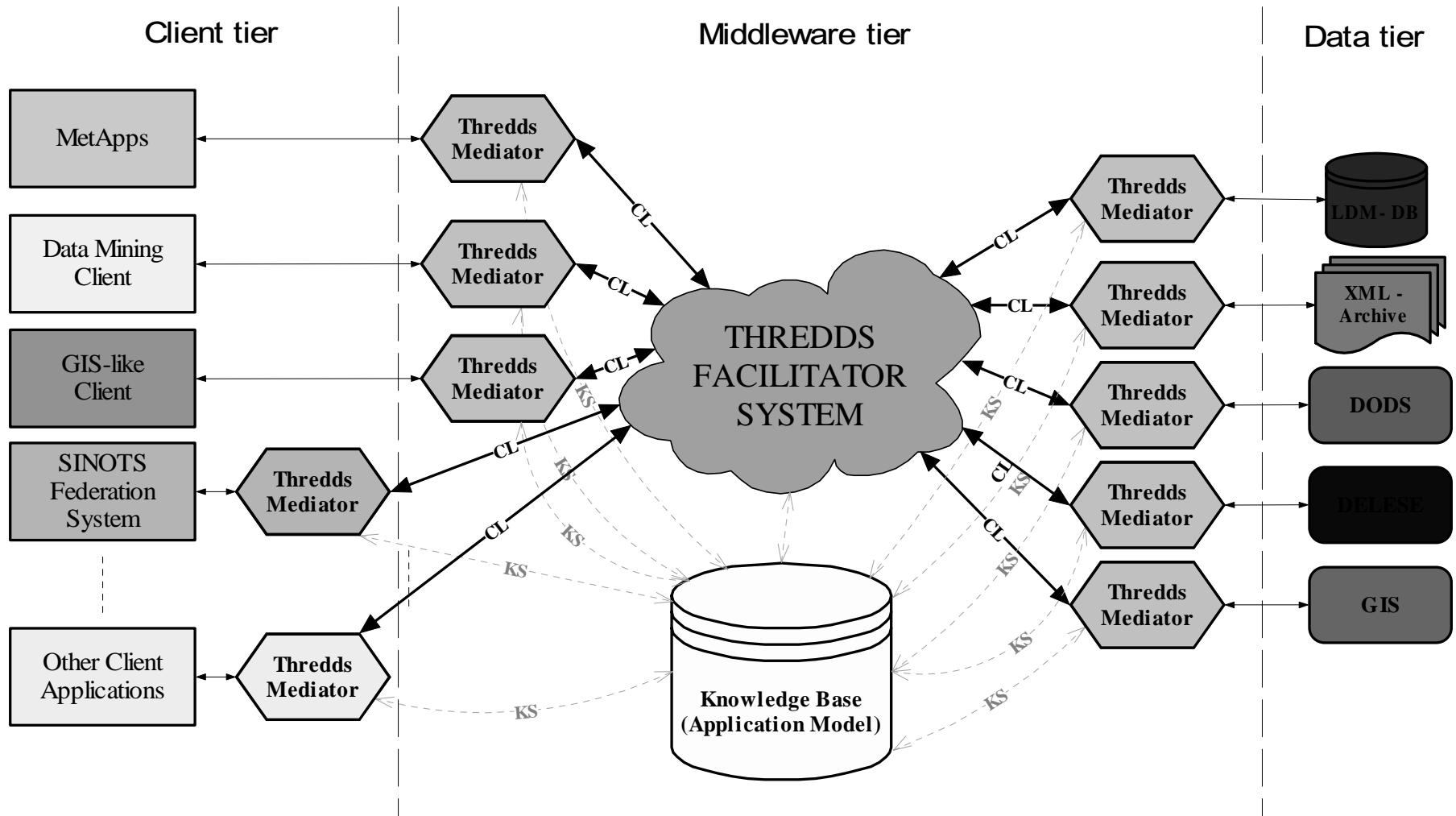
- **Fat: Integrated Data Viewer (IDV)**
  - Full functionality
- **Medium: Thredds Data Viewer**
  - Under 2Mb download – Applet possible (?)
  - Specialized applets even smaller ~ 100 Kbs
- **Thin: Web Browser**
  - Client is on the server, provides HTML service

# Application Development

- Integrated Data Viewer (IDV)
- UI Widgets
- Java library
- What about non-Java applications?

# Nativi/SINOTS: modified ISO

- Use OpenGIS/ISO model
  - for GIS data
  - Spatial and Temporal Reference Systems
  - coordinate transformations
- Add THREDDS Extensions for Grids
  - Extends the Coverage model for gridded data



# Dataset Model - Issues

- Physical vs. logical datasets
  - DODS Aggregation Server
  - Catalogs for ADDE data must aggregate
- Satellite data – moving coordinate system
  - Bounding box?
- DatasetDesc assumes {levels} X {times}

# Data Client POV

- Needs to know server name
- Needs to query server to find what datasets are available.
  - DODS: may need to know “root paths”.
  - ADDE: may be very slow.
- Figure out meaning of the data from dataset name?
  - May work for tightly coupled client/server, but not a general solution for distributed data.

# Issues

- What is a dataset: Data Granularity
- Missing, non/standard metadata
- Missing context
- What should DL URL point to?
- Non-Java development

# What is a THREDDS dataset?

In the context of a THREDDS catalog:

- Syntactically, it's an XML element
- Minimally, it has a display name and a URL.
- It's the smallest thing (atom) a user can select.
- It's the only thing a user can select in the current client library.

# Grid DatasetDesc XML

- Dataset = (name, dataType, {Field}, {Times})
- Field = (name, unit, accessPath, SQ\*)
- Times = (name, unit, ({coord}|Regular))
- Coord = (name, unit, value, accessPath)
- Regular = (start, incr, npts)