Colorado State University

MANAGING SCIENTIFIC DATA WITH NDN

Chengyu Fan, Susmit Shannigrahi, Steve DiBenedetto,
Catherine Olschanowsky, Christos Papadopoulos

NDNcomm 2015
Sept 28, 2015 Los Angeles, CA
Supported by NSF #13410999 and NSF#1345236

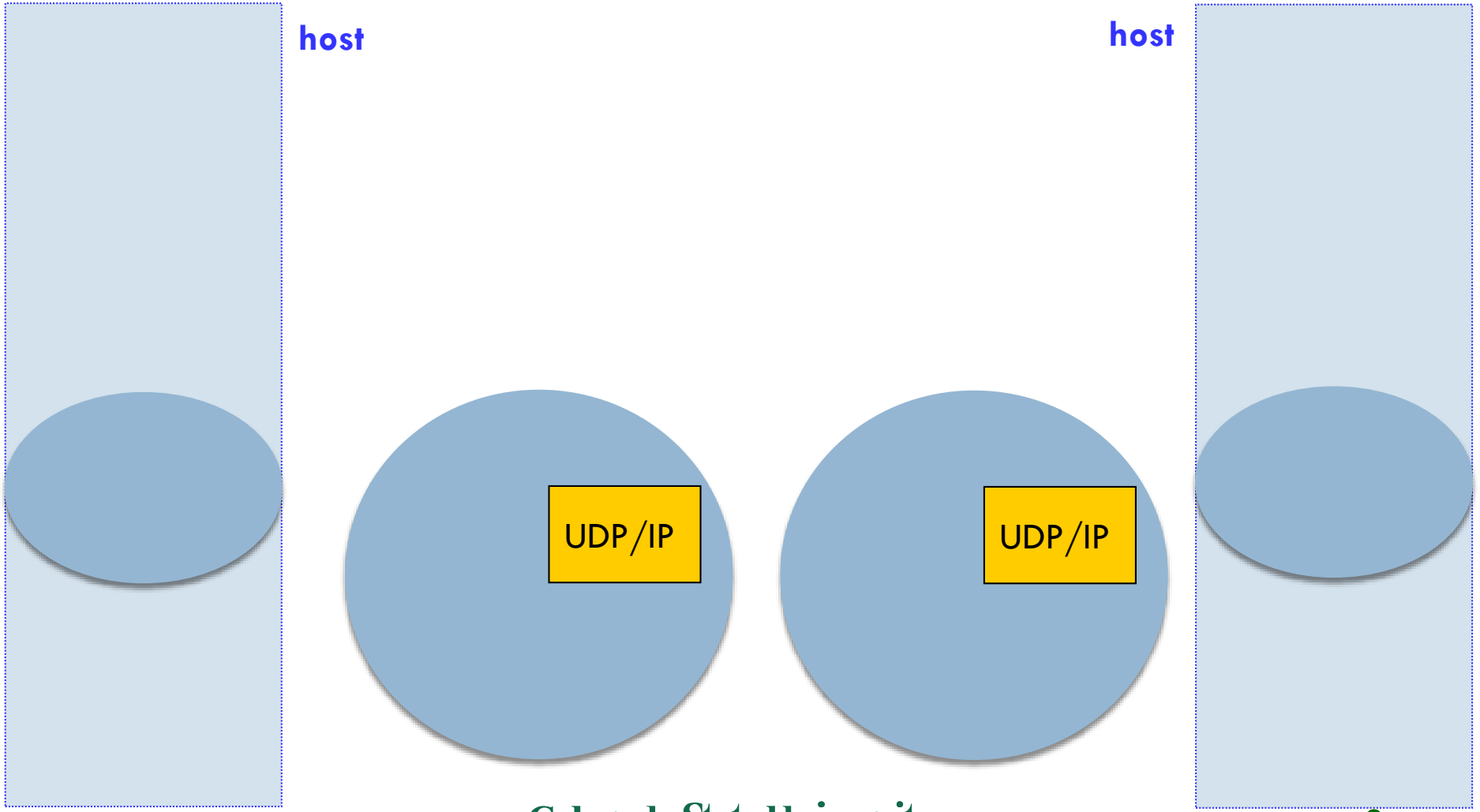
Introduction

- Scientific data is often very large and complex
  - Climate - CMIP5: 3.5 PB, CMIP6: 350PB-3EB
  - Physics - Atlas: 4 PB/Year
  - Astronomy, bioinformatics, others…
- Science infrastructure
  - Cutting edge hardware but often incompatible domain software (ESGF, xrootd, etc.)
  - Complexity, replication, redundancy

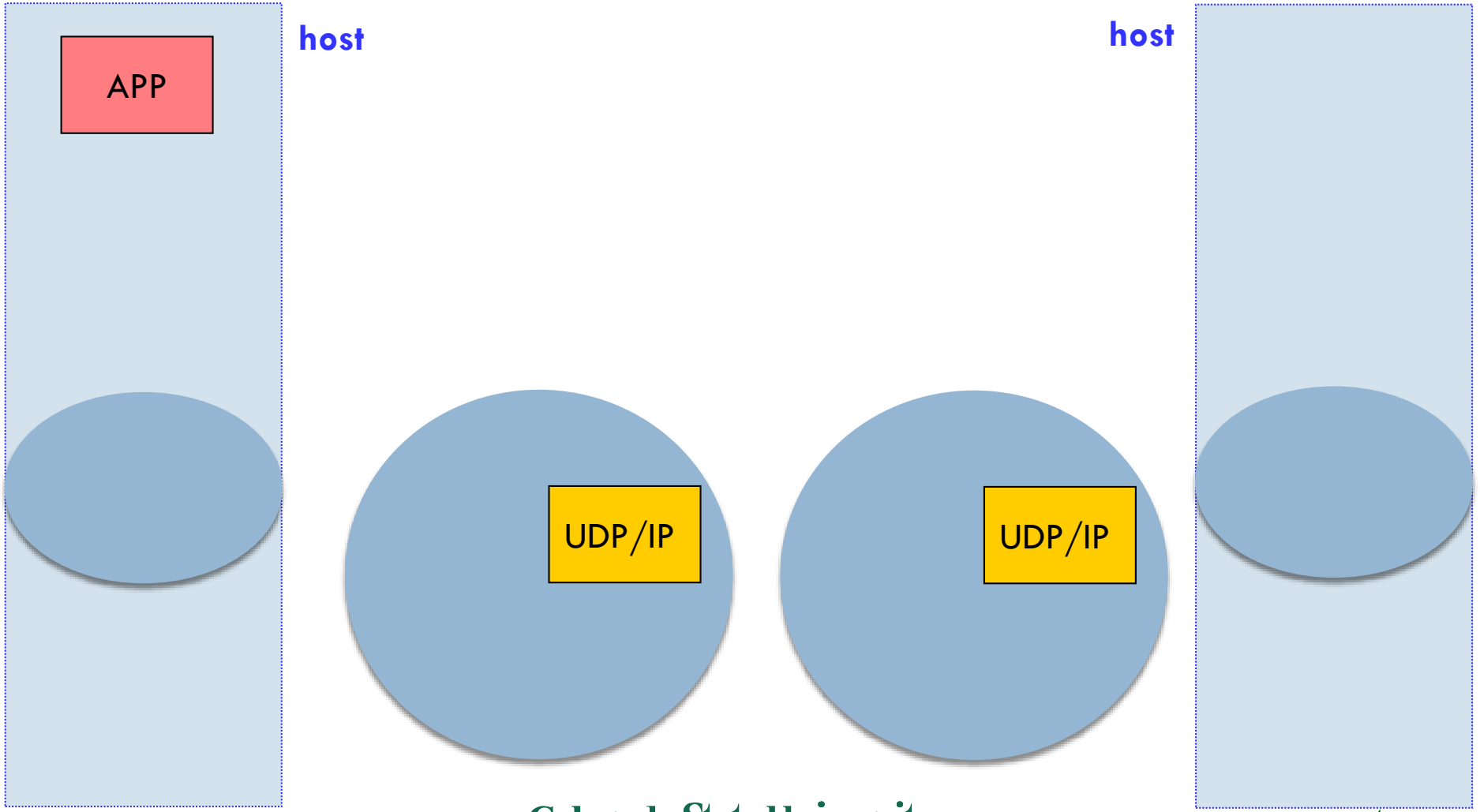
Our Project

- Build and deploy software to evaluate NDN in scientific applications over a dedicated hardware infrastructure
- Evaluate NDN in the context of:
  - Application services: publishing, discovery, retrieval, access control, load balancing, failover, caching, etc.
  - Network integration (OSCARS, SDN, etc.)
- Metrics
  - Performance, reduced complexity, ease of deployment, interoperability, reuse, efficiency, routing, security/trust, etc.

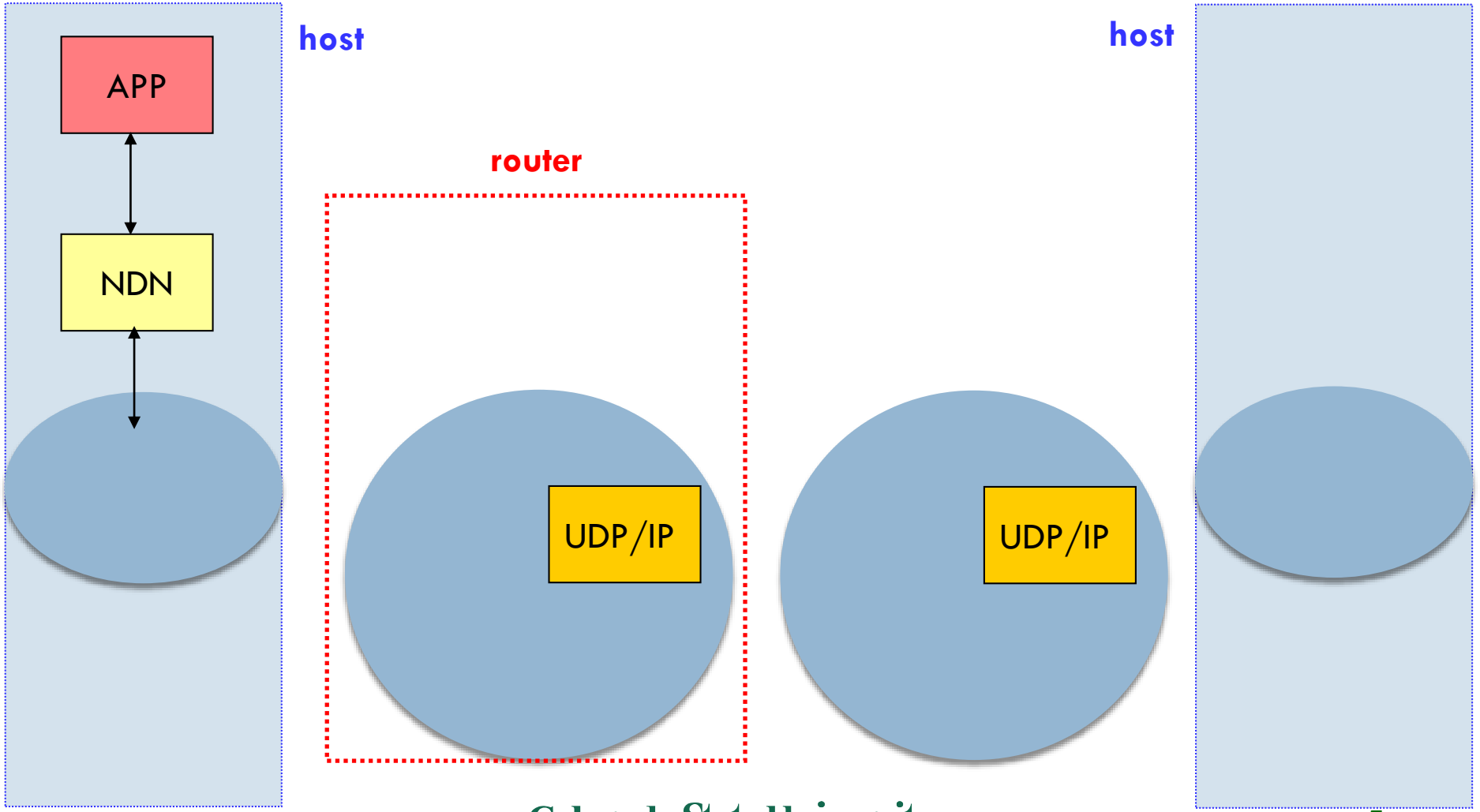
NDN Layer Structure



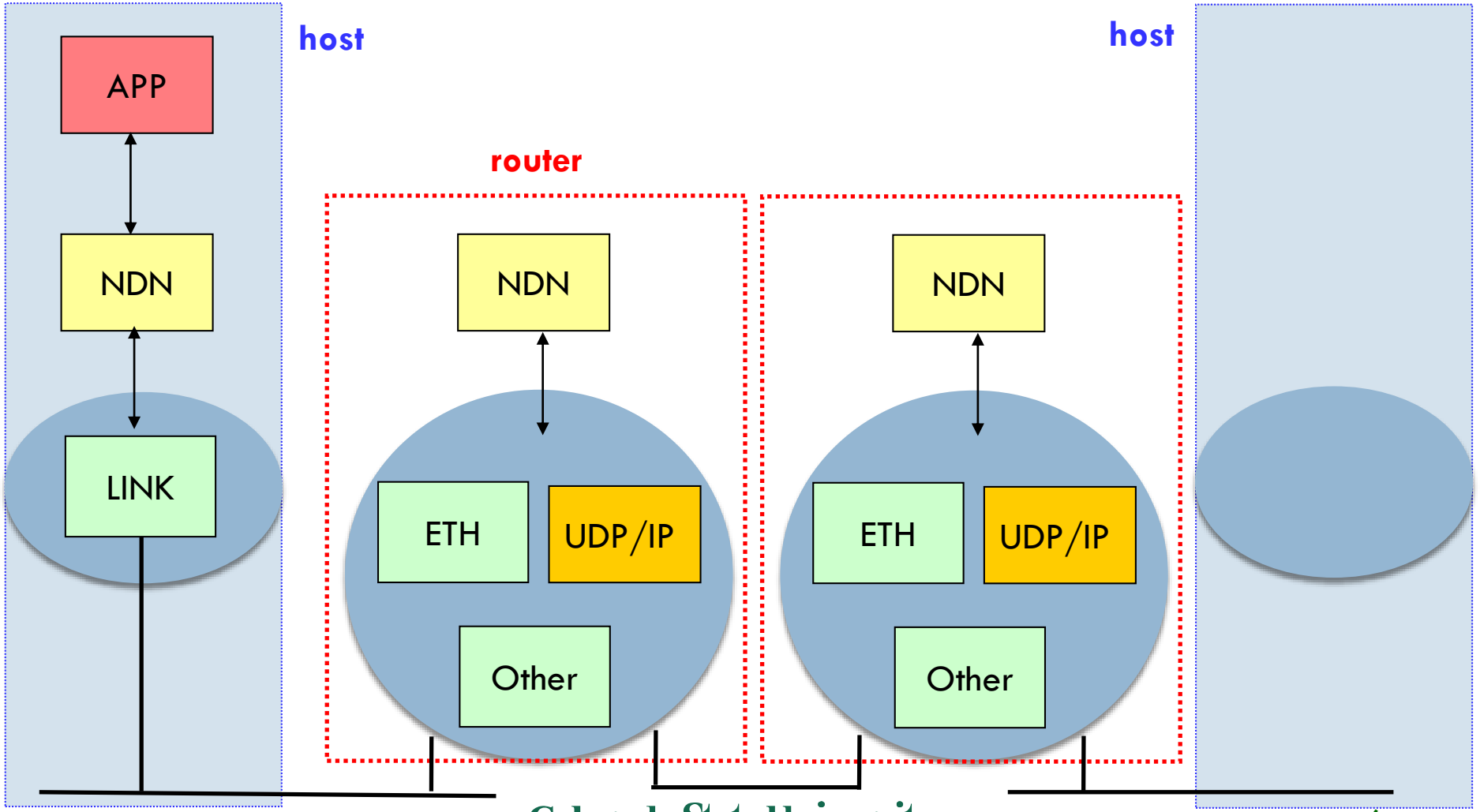
NDN Layer Structure



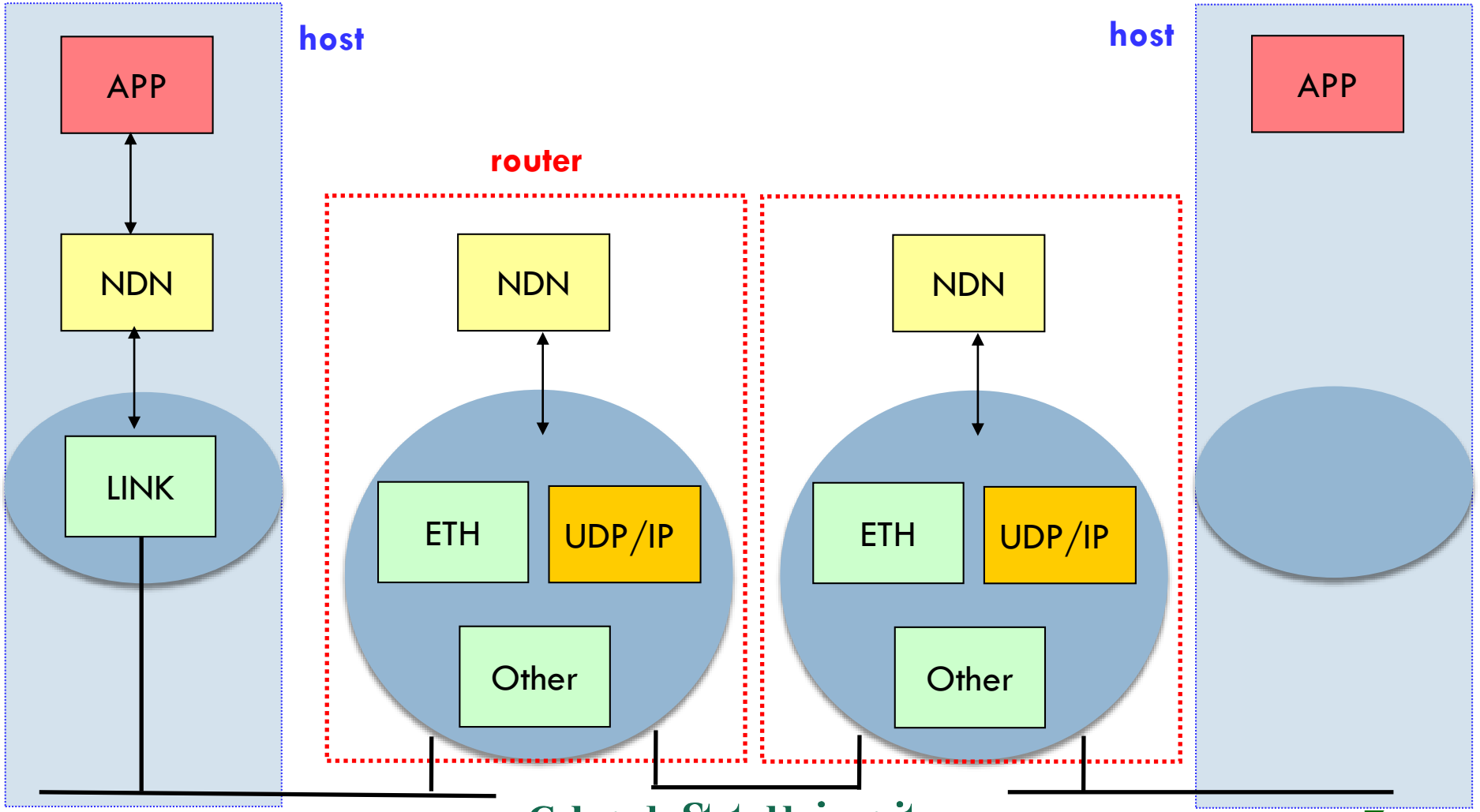
NDN Layer Structure



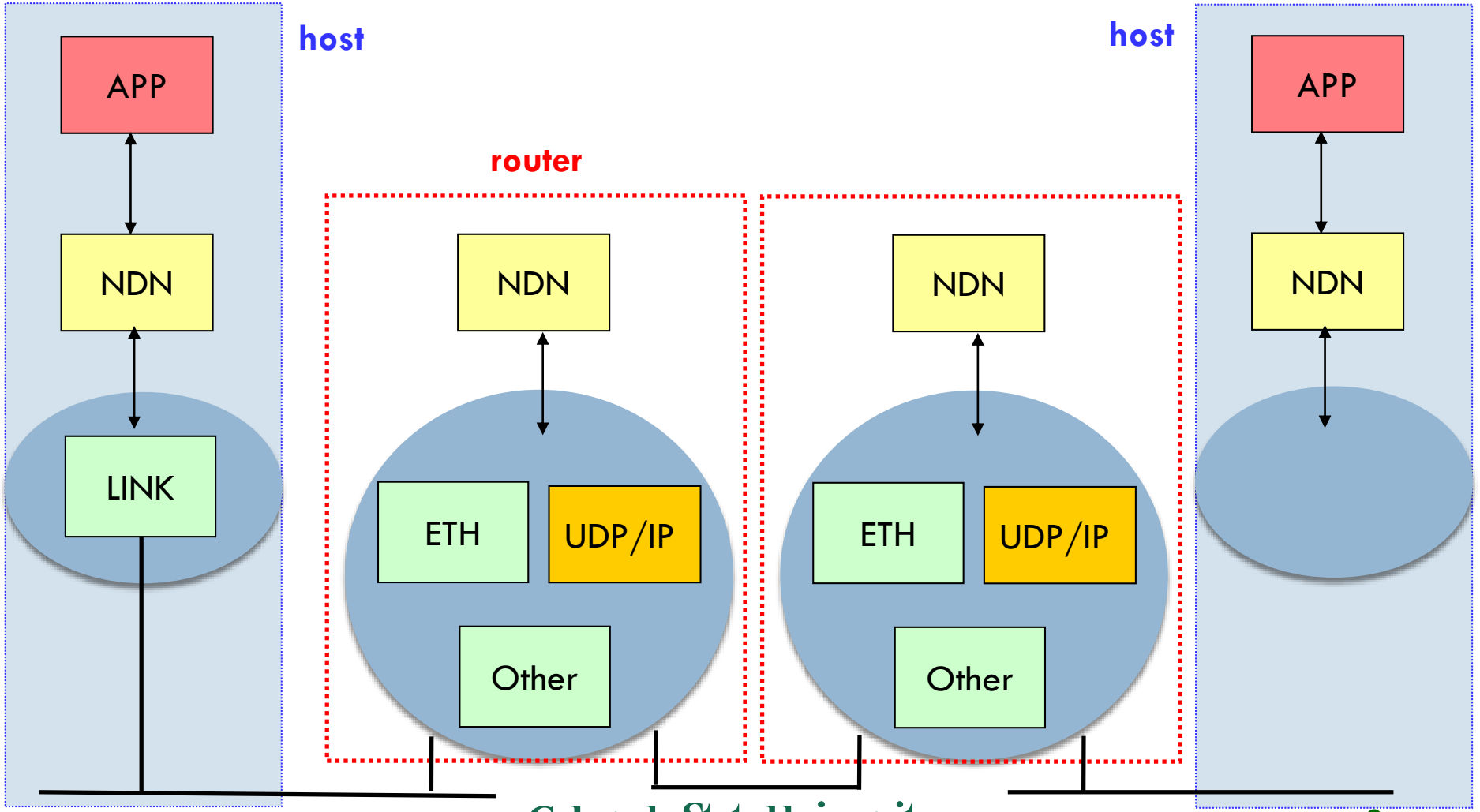
NDN Layer Structure



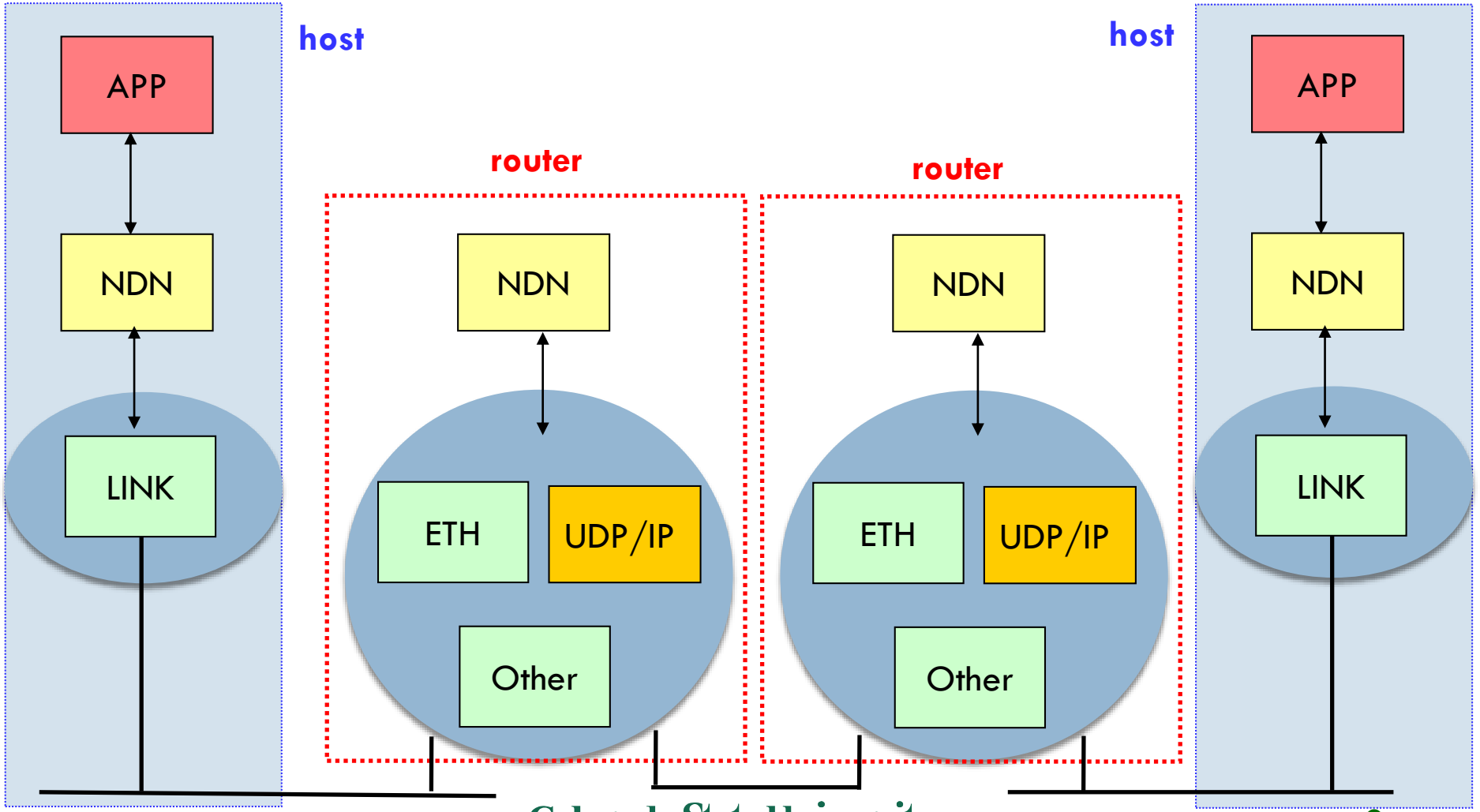
NDN Layer Structure



NDN Layer Structure



NDN Layer Structure



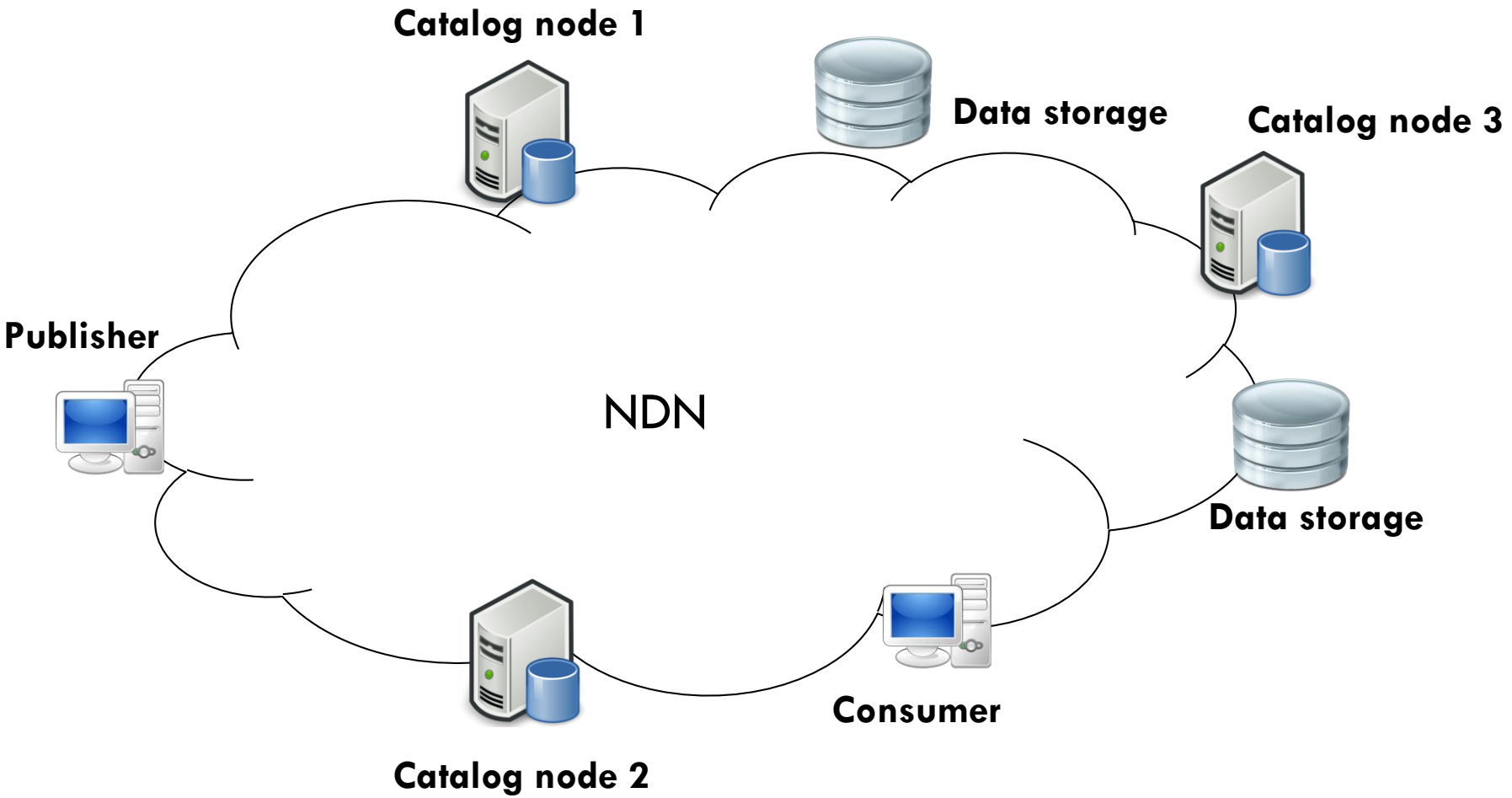
Methodology

- Investigate the use of NDN as a common platform for scientific data applications by:
  - Understanding data management challenges of various scientific domains
  - Developing and evaluating prototype applications that leverage NDN's features
  - Use prototypes to further drive NDN research

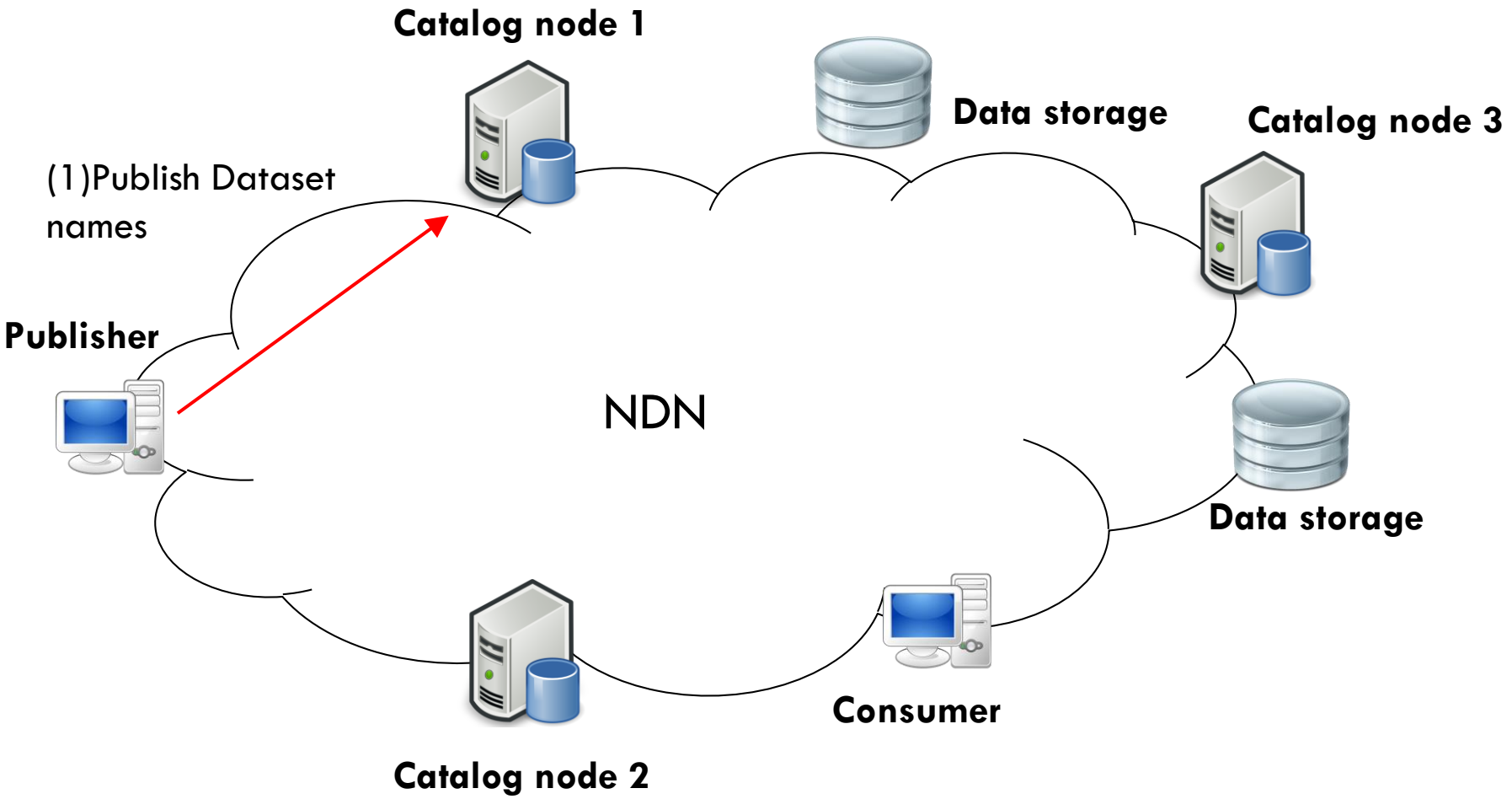
First Step – Build a Catalog

- Create a **shared resource** – a distributed, synchronized catalog of names over NDN
  - Provide common operations such as publishing, discovery, access control
  - Catalog only deals with name management, not dataset retrieval
  - Platform for further research and experimentation
- Research questions:
  - Namespace construction, distributed publishing, key management, UI design, failover, etc.
  - Functional services such as subsetting
  - Mapping of name-based routing to tunneling services (VPN, OSCARS, MPLS)

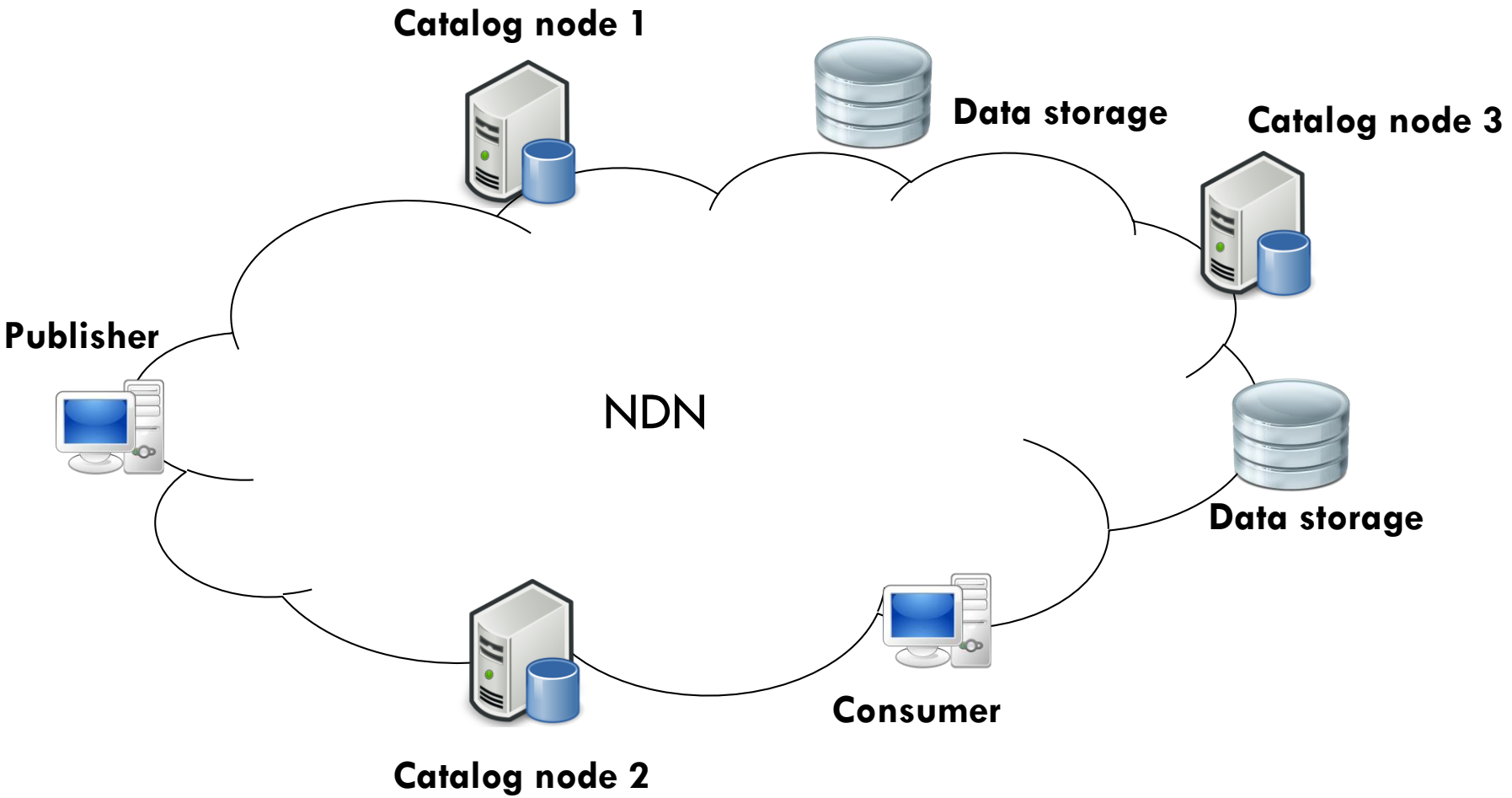
Overview of Catalog Workflow



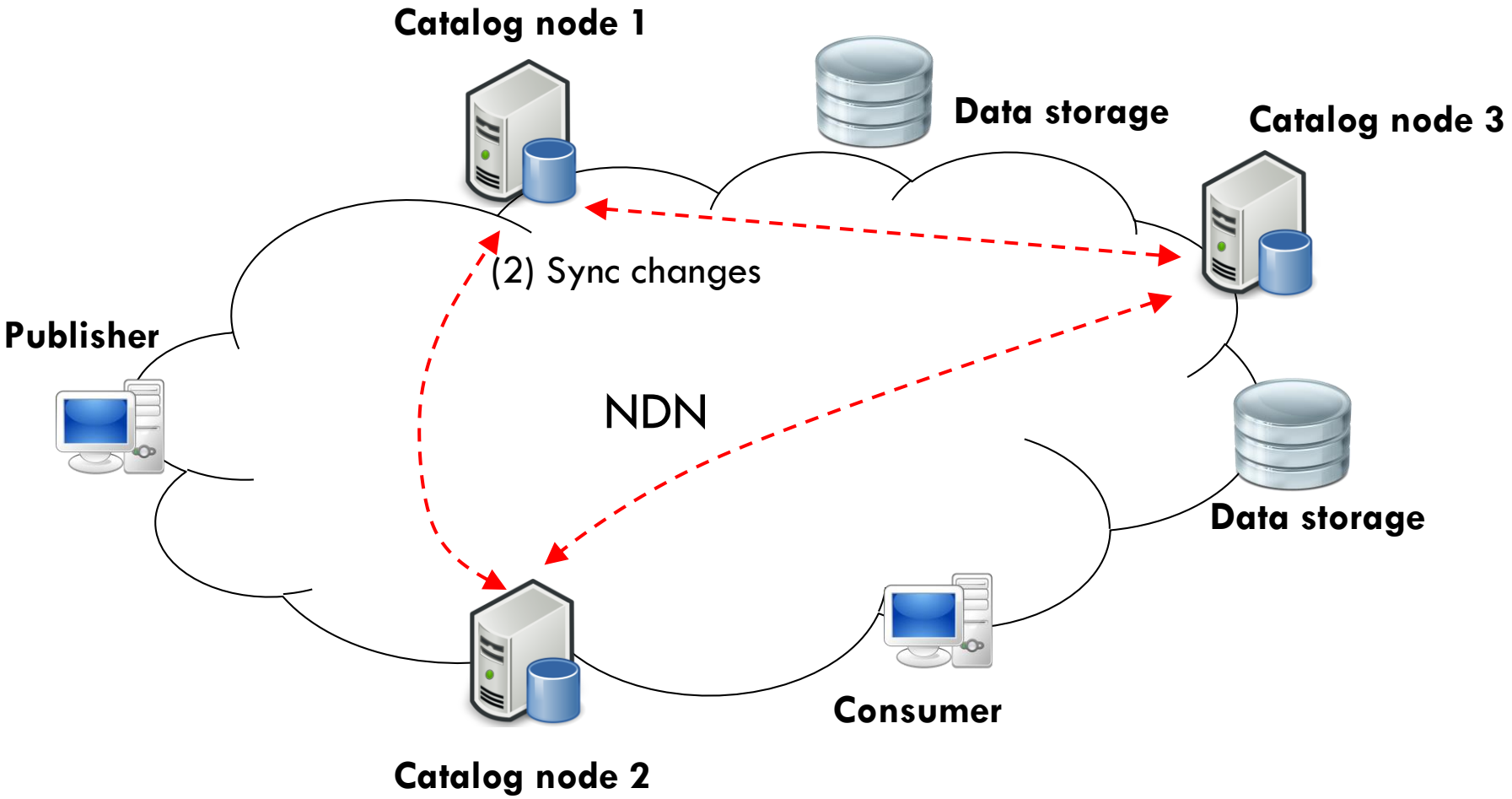
Overview of Catalog Workflow



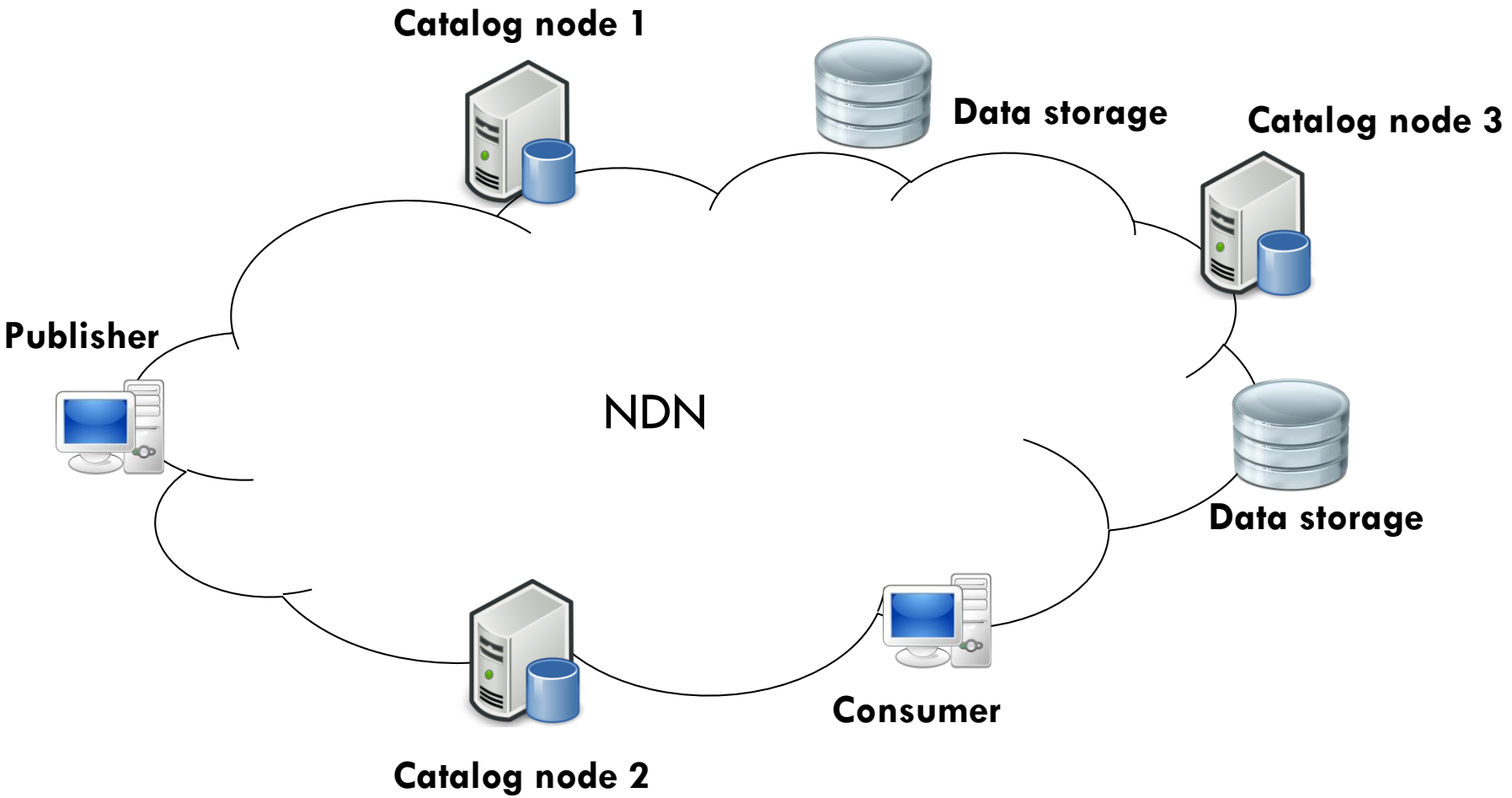
Overview of Catalog Workflow



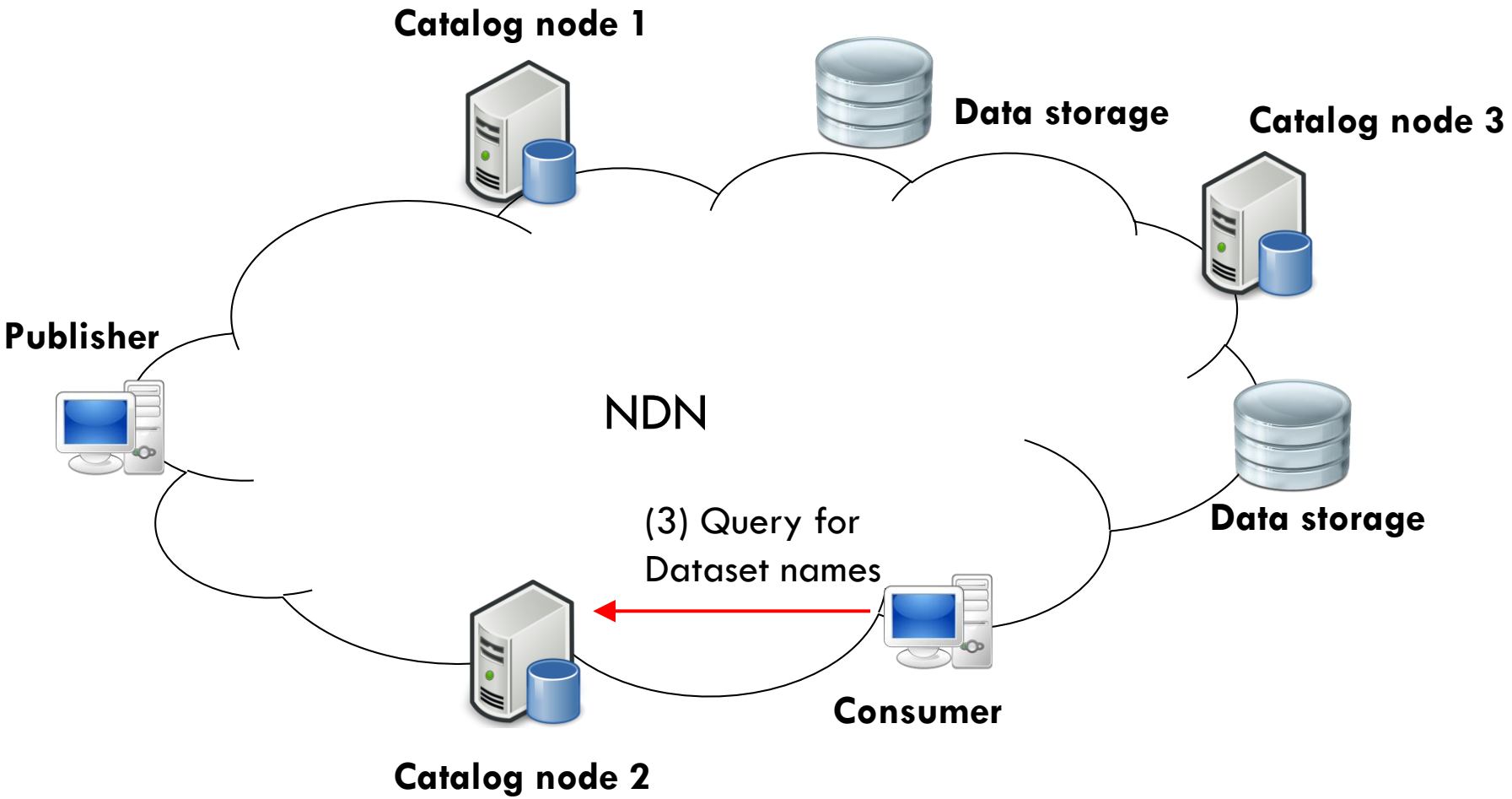
Overview of Catalog Workflow



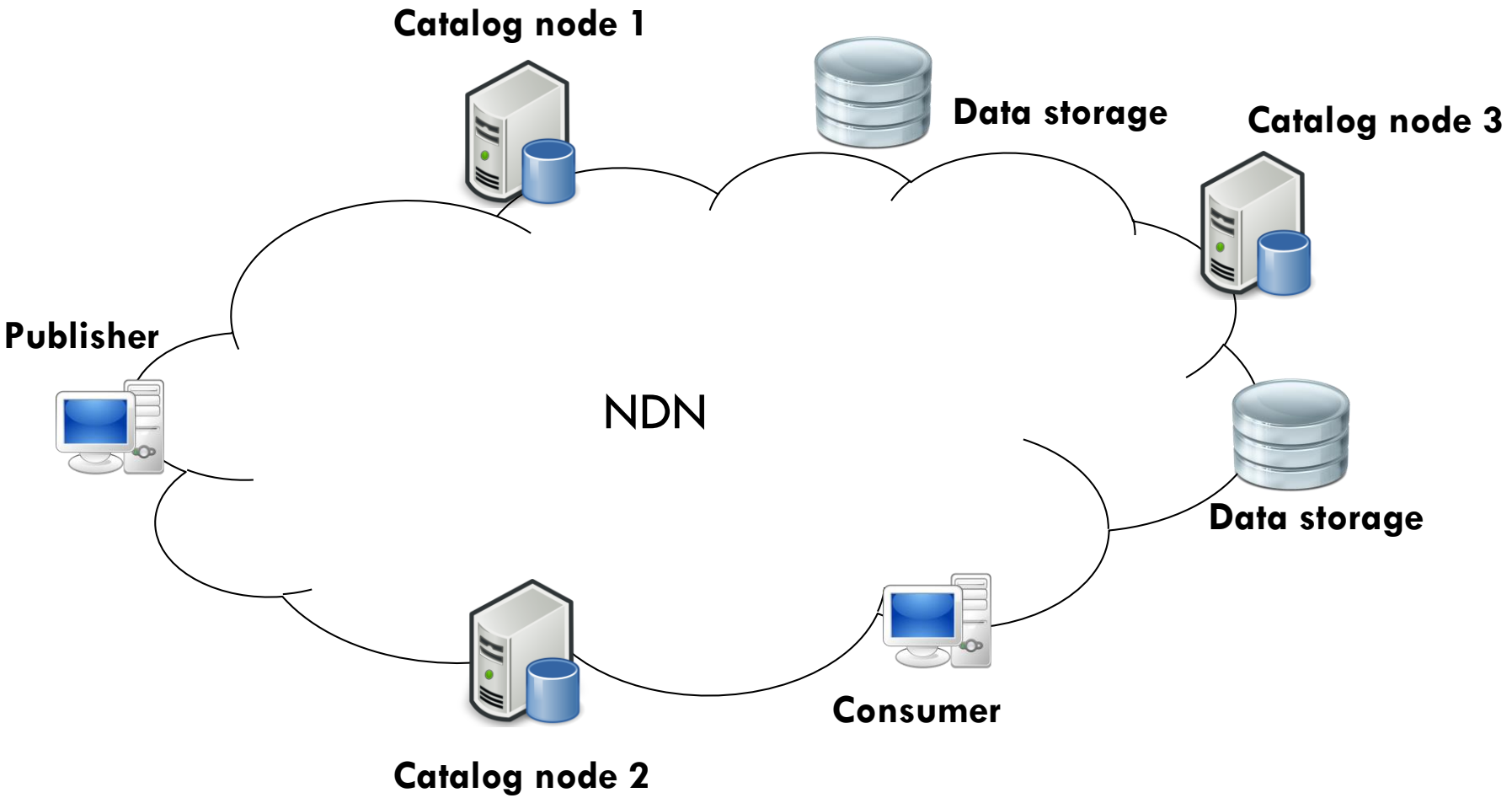
Overview of Catalog Workflow



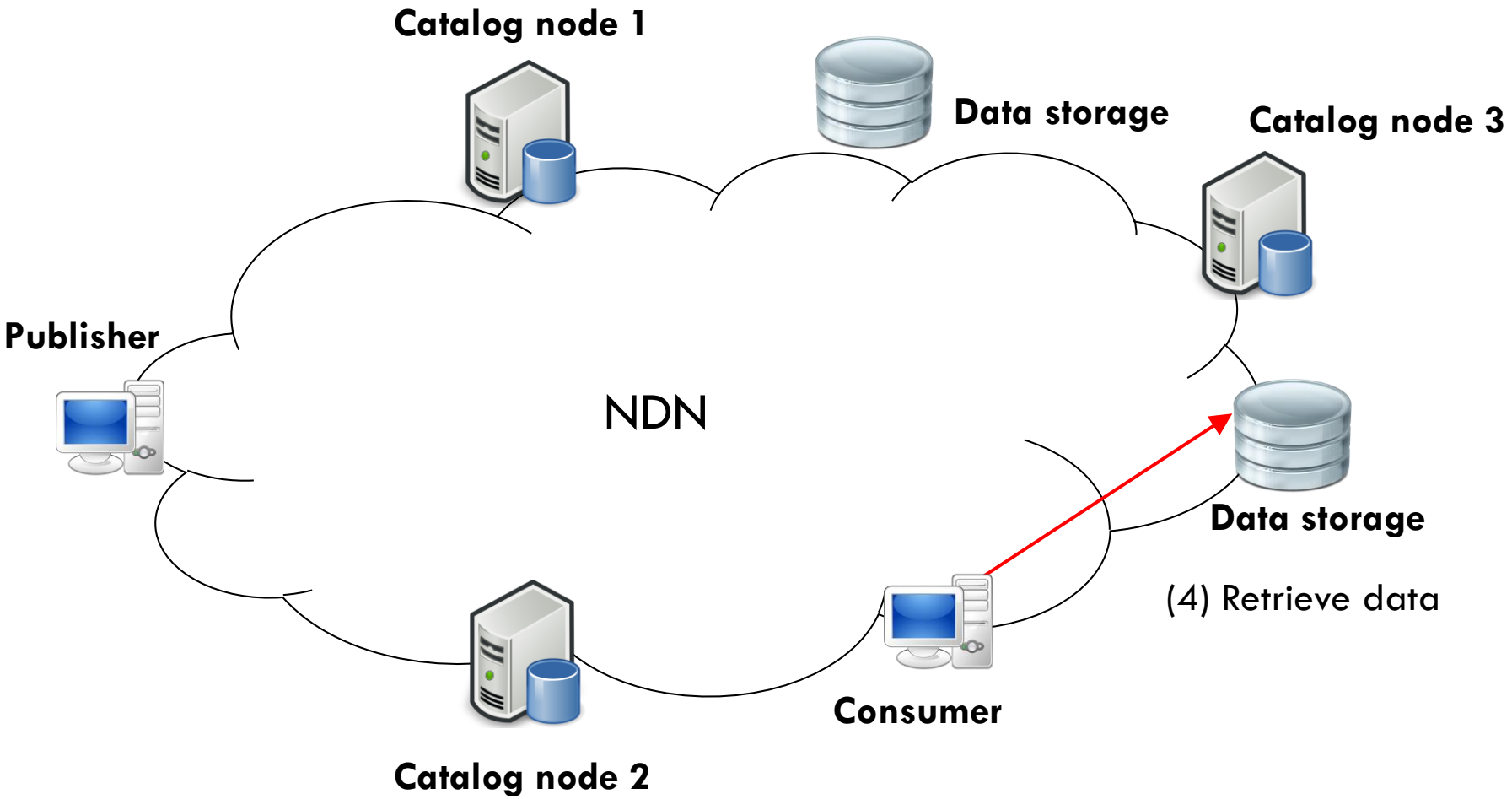
Overview of Catalog Workflow



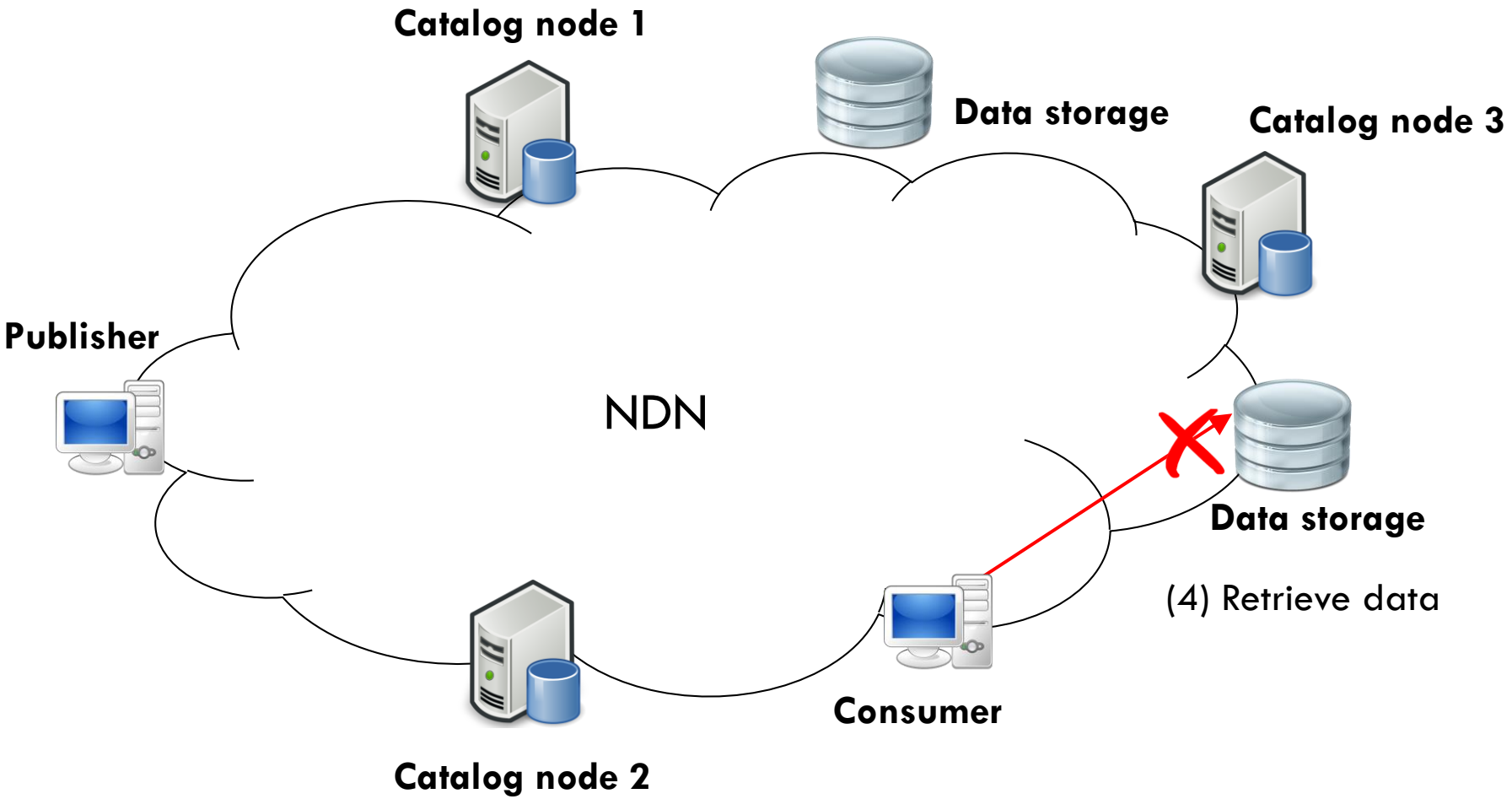
Overview of Catalog Workflow



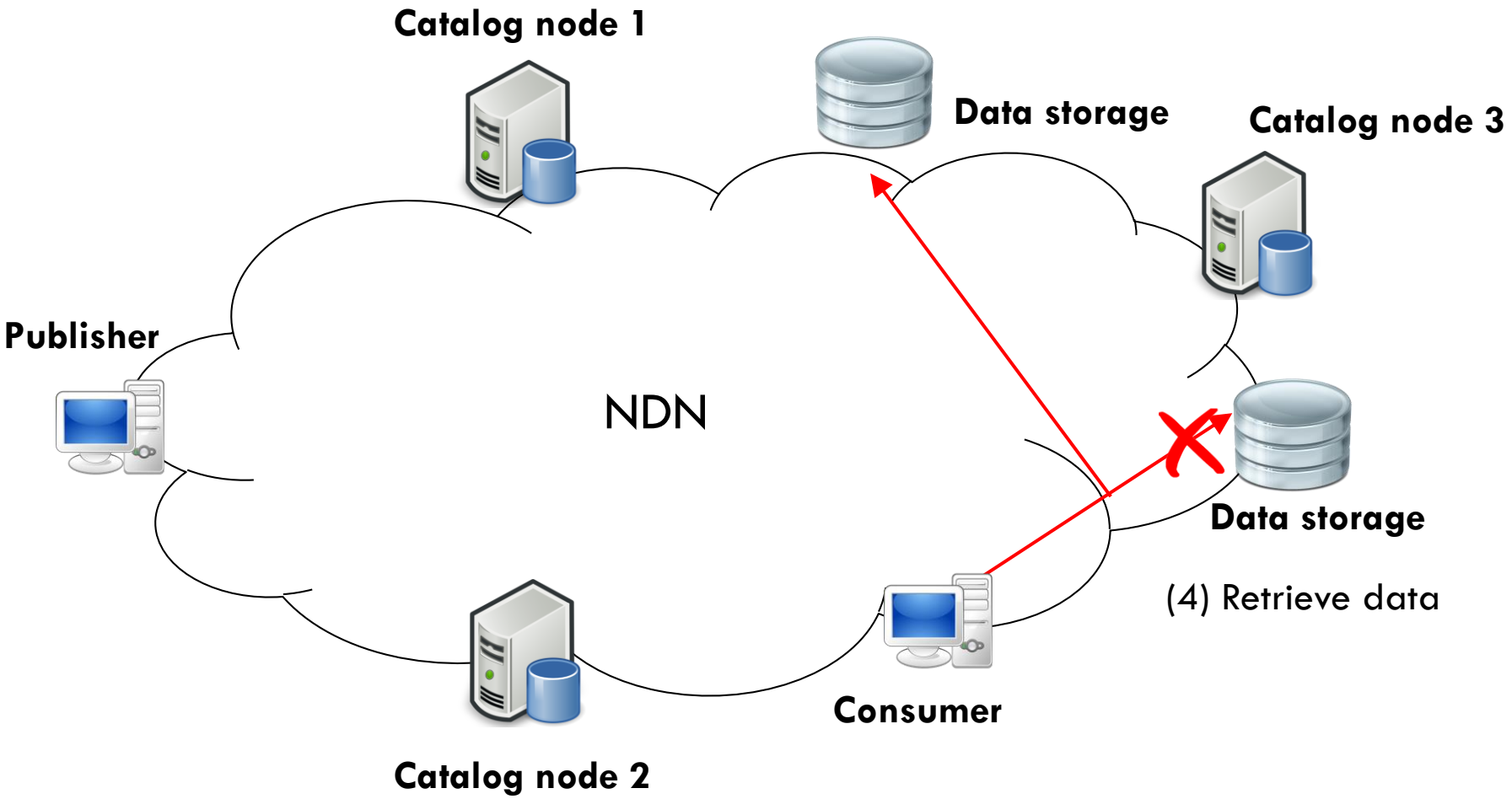
Overview of Catalog Workflow



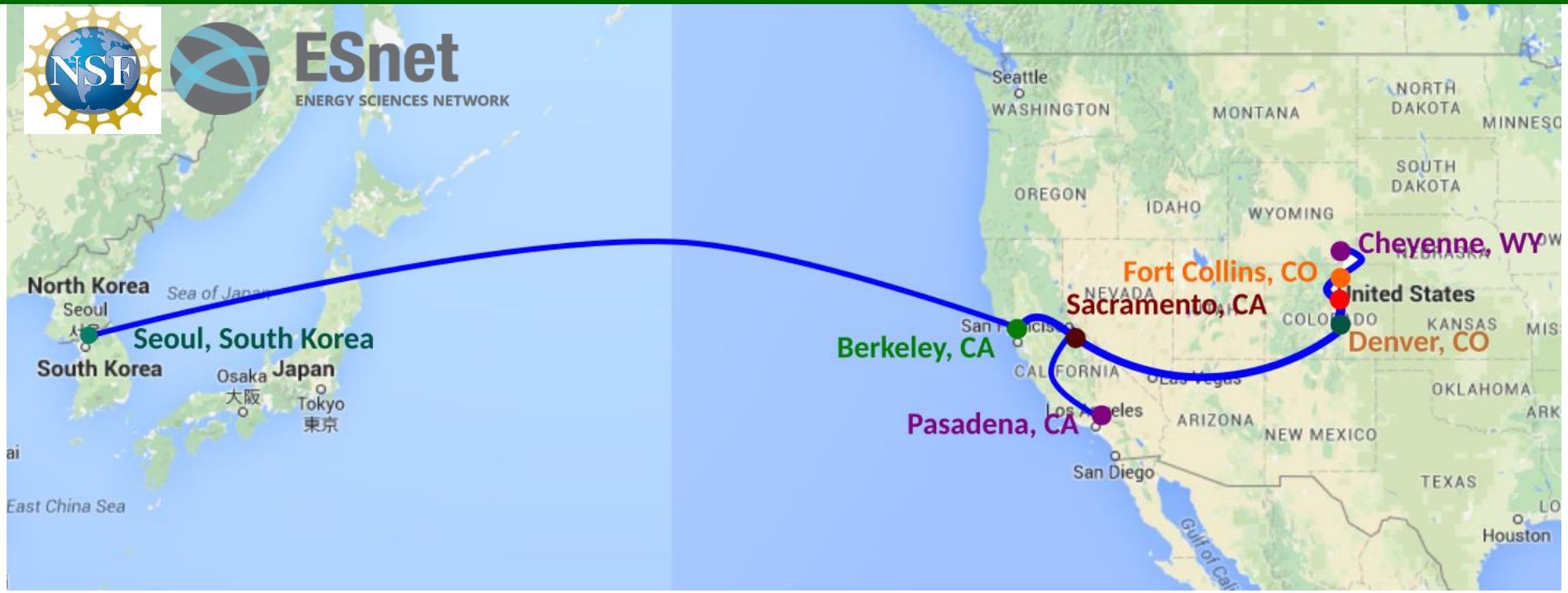
Overview of Catalog Workflow



Overview of Catalog Workflow



NDN-Science Testbed



- NSF CC-NIE campus infrastructure award
  - 10G testbed (courtesy of ESnet, UCAR, and CSU Research LAN)
- Currently ~50TB of CMIP5, ~70TB of HEP data

Demos

- Search
- Publication and Sync
- Access control
- Retrieval and failover

Conclusions

- IP encourages common **host** access, not common **data** access methods
  - Does not encourage interoperability at the application level
- NDN has the potential to unify the service interface required by scientific applications
  - Science testbed and prototypes to test hypothesis and drive research and experimentation
- Ready-to-try catalog, we invite you to try it with your data
  - Catalog is general, supports a variety of applications
  - Currently CMIP5 and HEP applications
  - UI for data search and retrieval.

Thanks

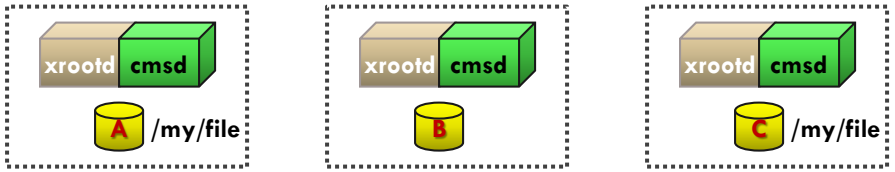
Our sponsors: NSF and ESnet

Join us @

http://www.netsec.colostate.edu/mailman/listinfo/ndn-sci

Backup Slides

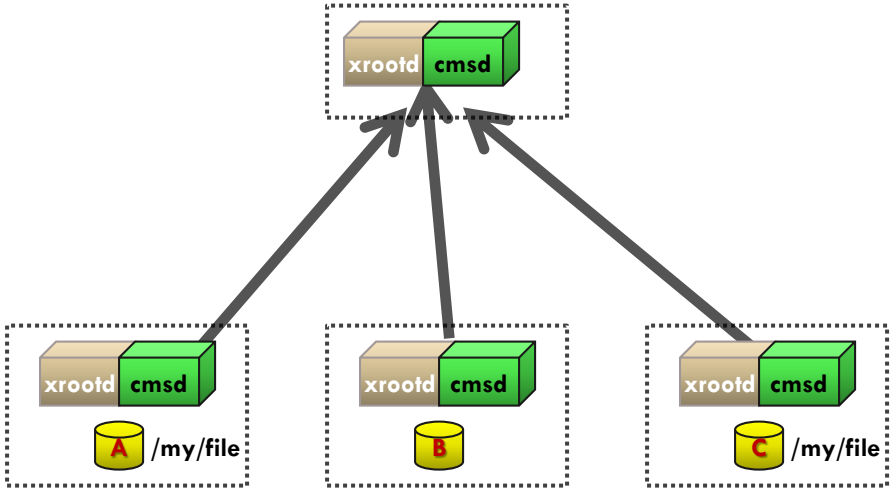
Current Example: xrootd



Data Servers

- Fragile, fairly complex middleware

Current Example: xrootd

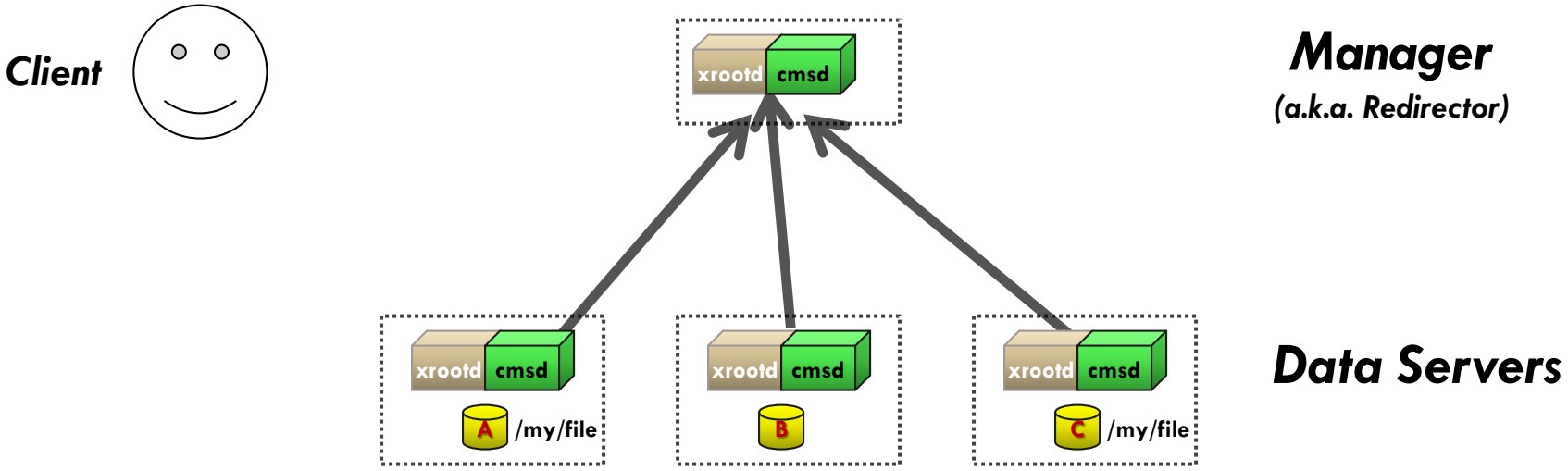


Manager
(a.k.a. Redirector)

Data Servers

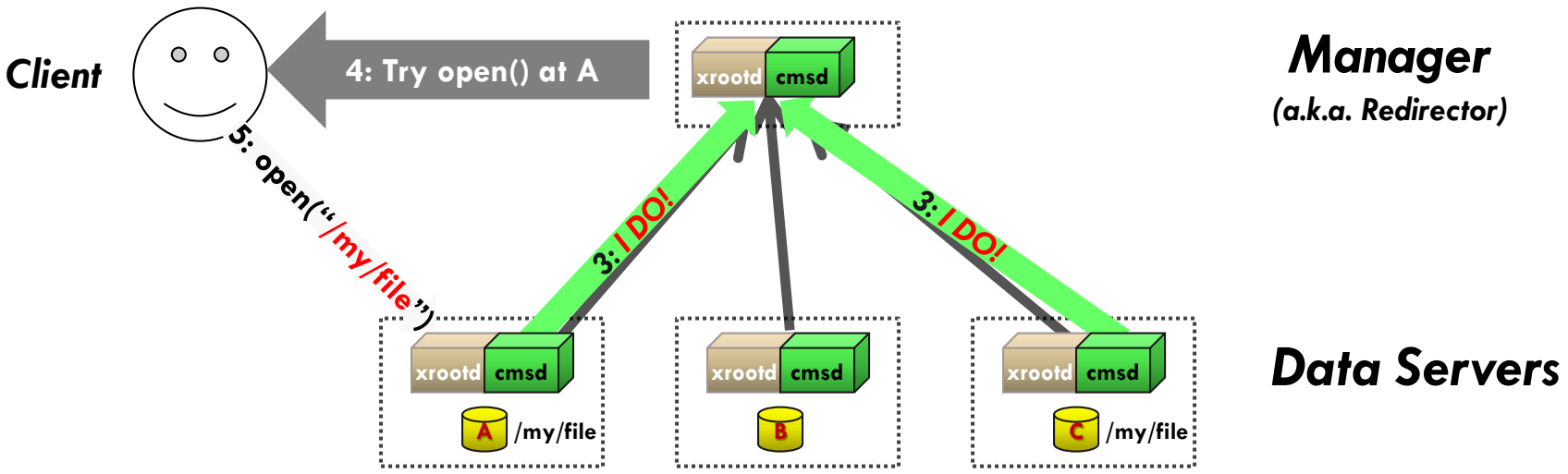
- Fragile, fairly complex middleware

Current Example: xrootd



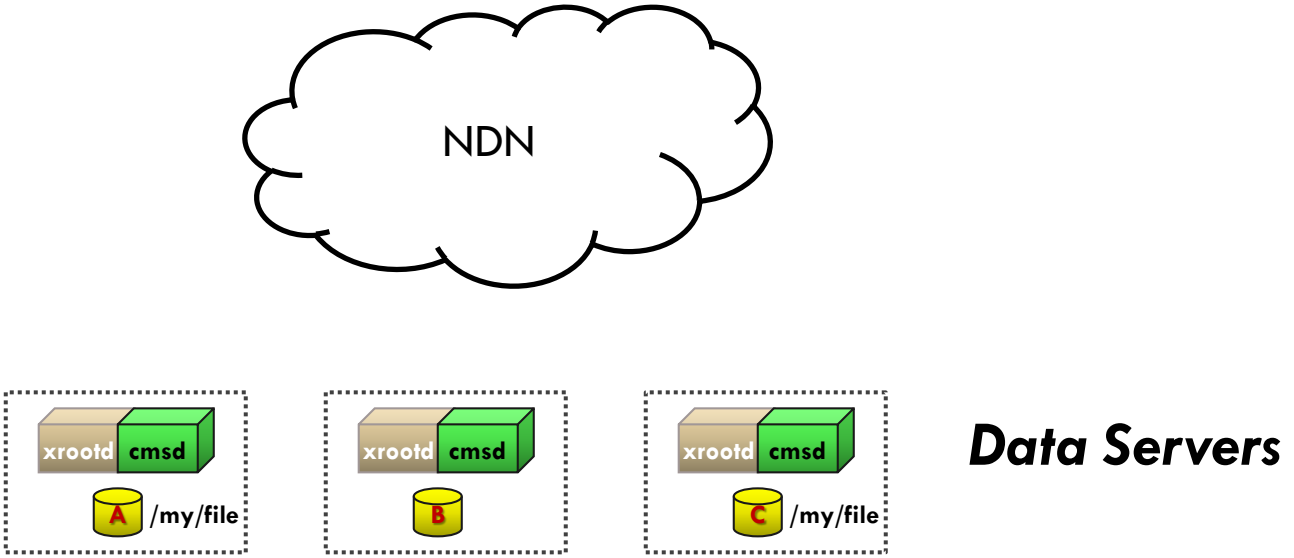
- Fragile, fairly complex middleware

Current Example: xrootd



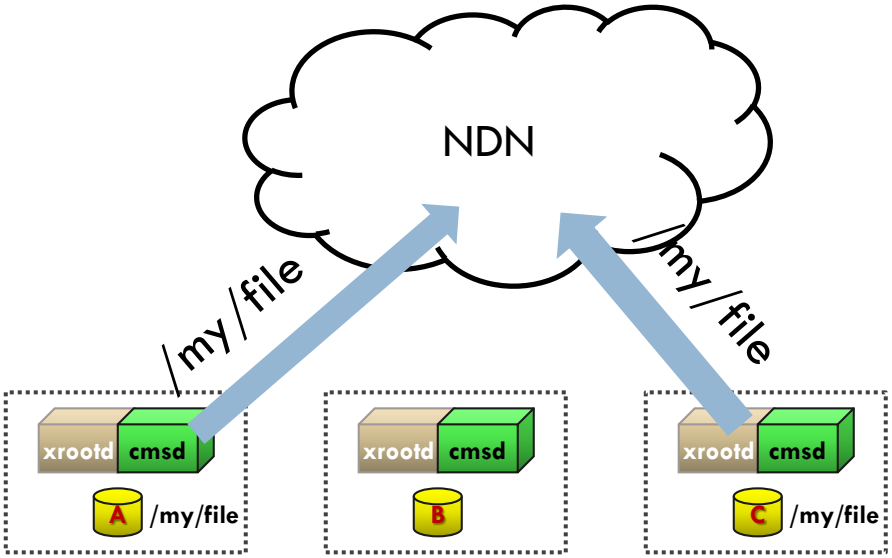
- Fragile, fairly complex middleware

xrootd under NDN



- Significantly reduced system complexity
- Better service abstraction

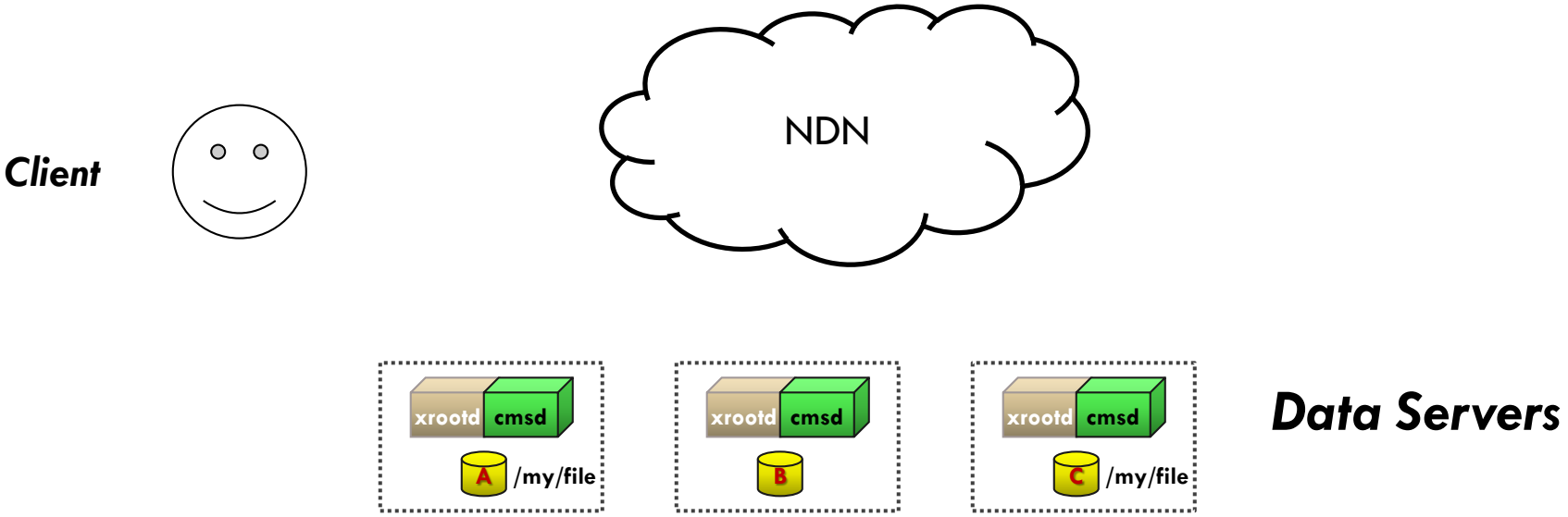
xrootd under NDN



Data Servers

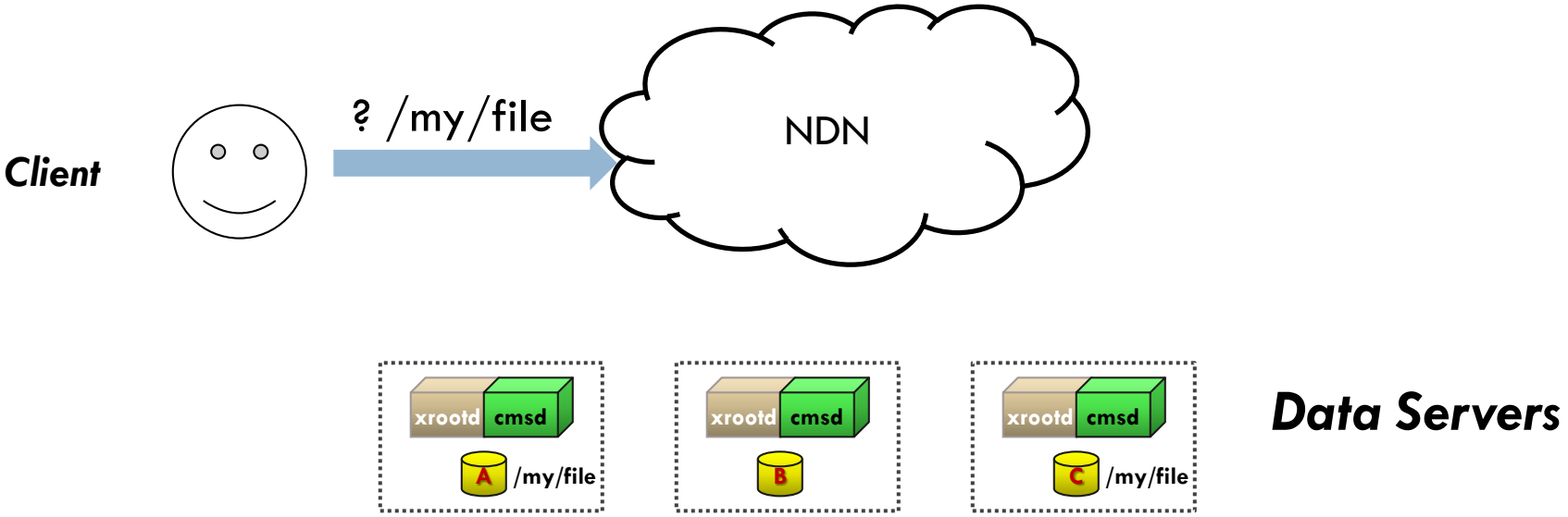
- Significantly reduced system complexity
- Better service abstraction

xrootd under NDN



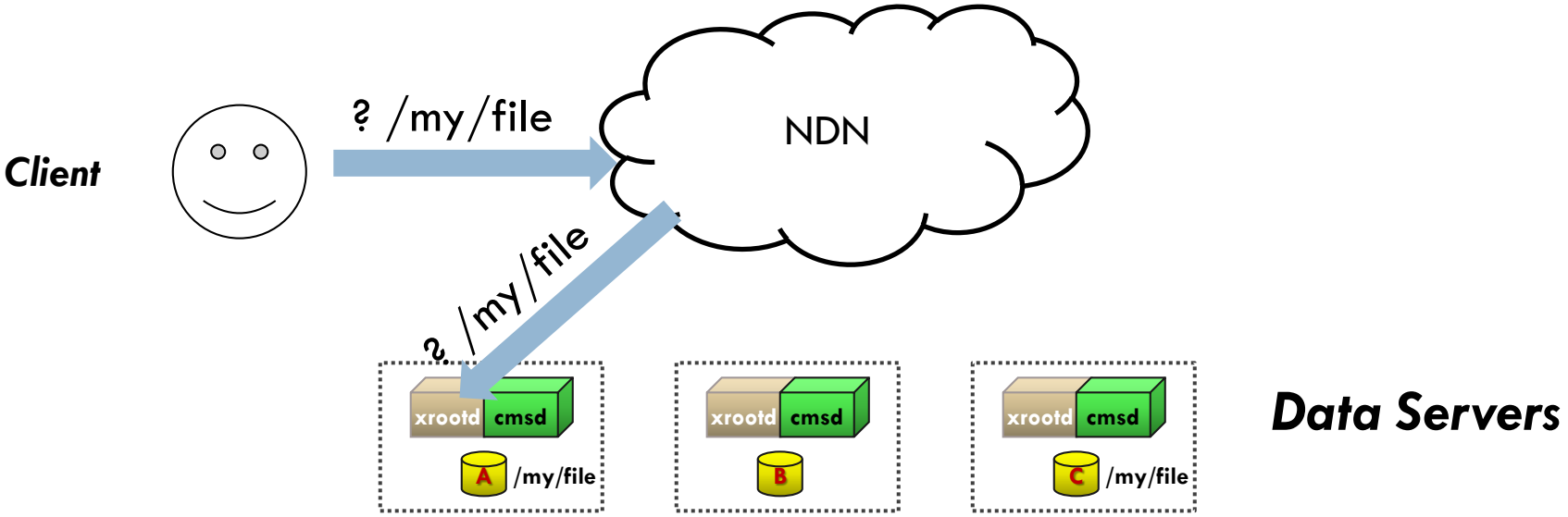
- Significantly reduced system complexity
- Better service abstraction

xrootd under NDN



- Significantly reduced system complexity
- Better service abstraction

xrootd under NDN



- Significantly reduced system complexity
- Better service abstraction

Data Publication

Catalog

Publisher

1) Listening on /<catalog-prefix>/publish

Data Publication

Catalog

1) Listening on /<catalog-prefix>/publish

Publisher

2) Generate NDN names for datasets/services

Data Publication

Catalog

Publisher

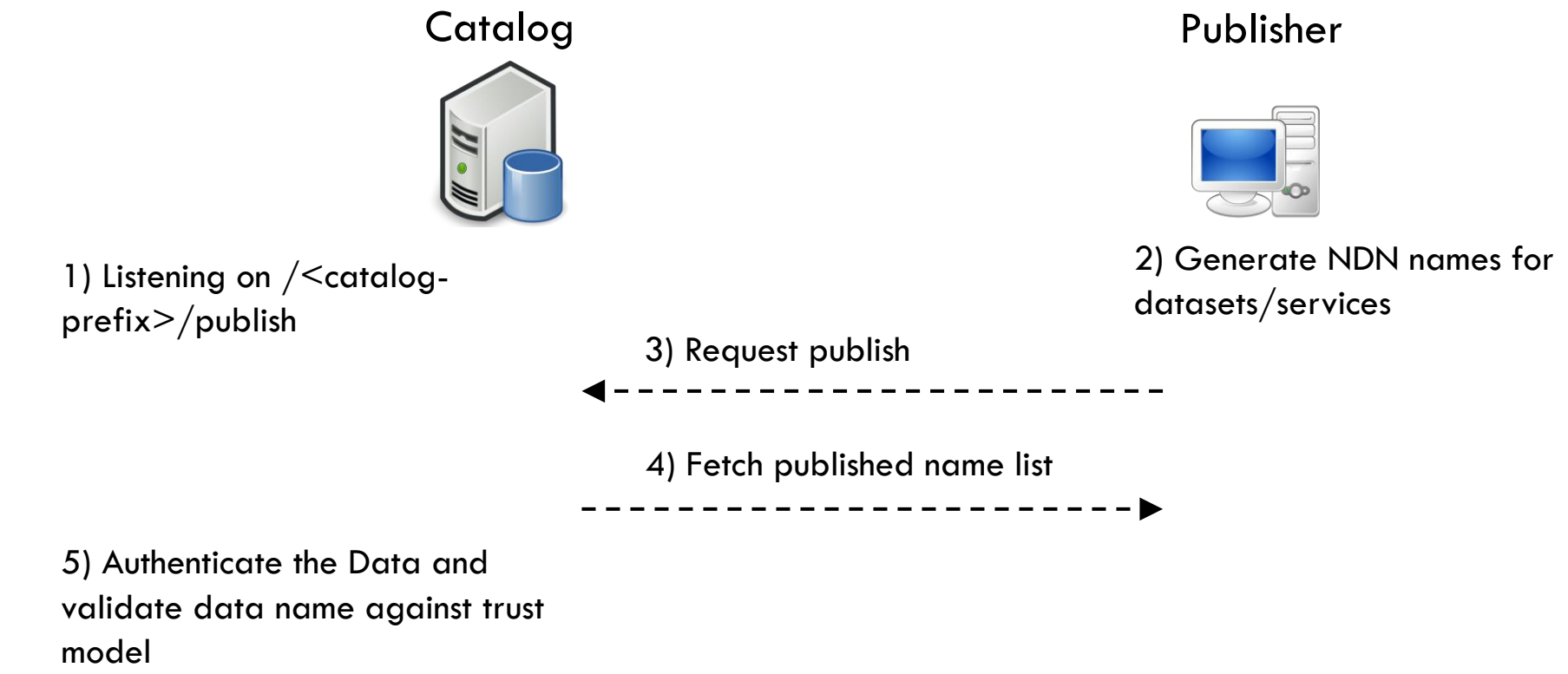
1) Listening on /<catalog-prefix>/publish

2) Generate NDN names for datasets/services

3) Request publish

Data Publication

Data Publication



Data Publication

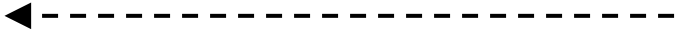
Catalog

Publisher

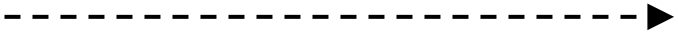
1) Listening on /<catalog-prefix>/publish

2) Generate NDN names for datasets/services

3) Request publish



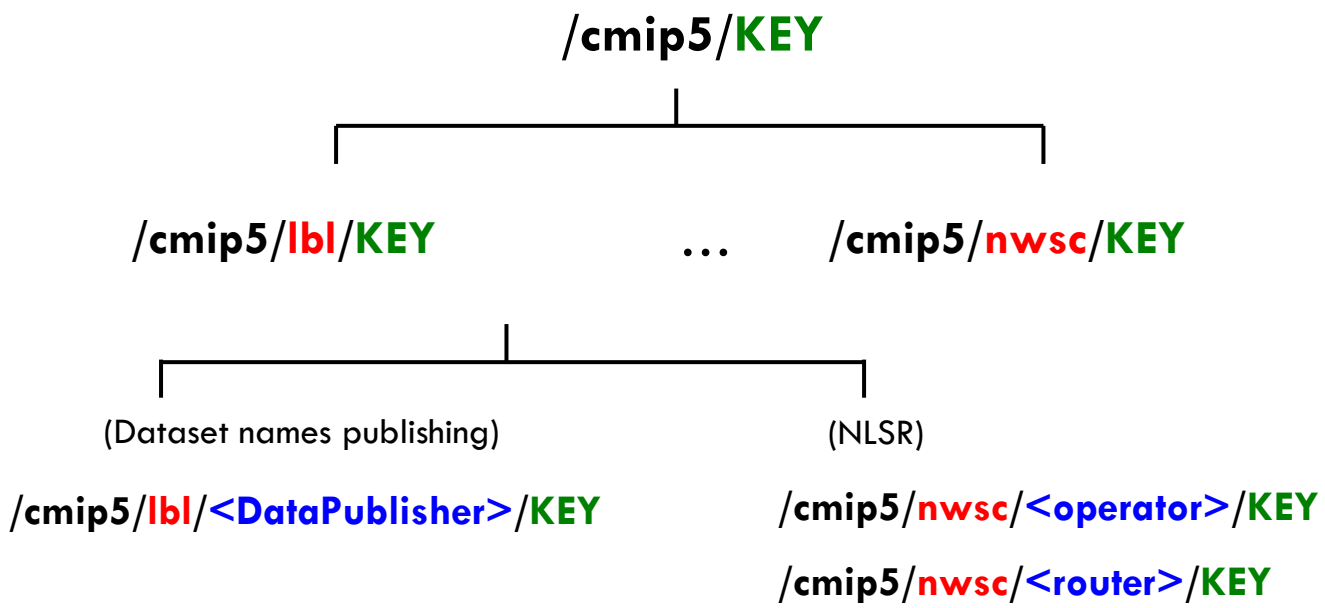
4) Fetch published name list



5) Authenticate the Data and validate data name against trust model

6) Share names with other catalogs

Keys for ndn-atmos

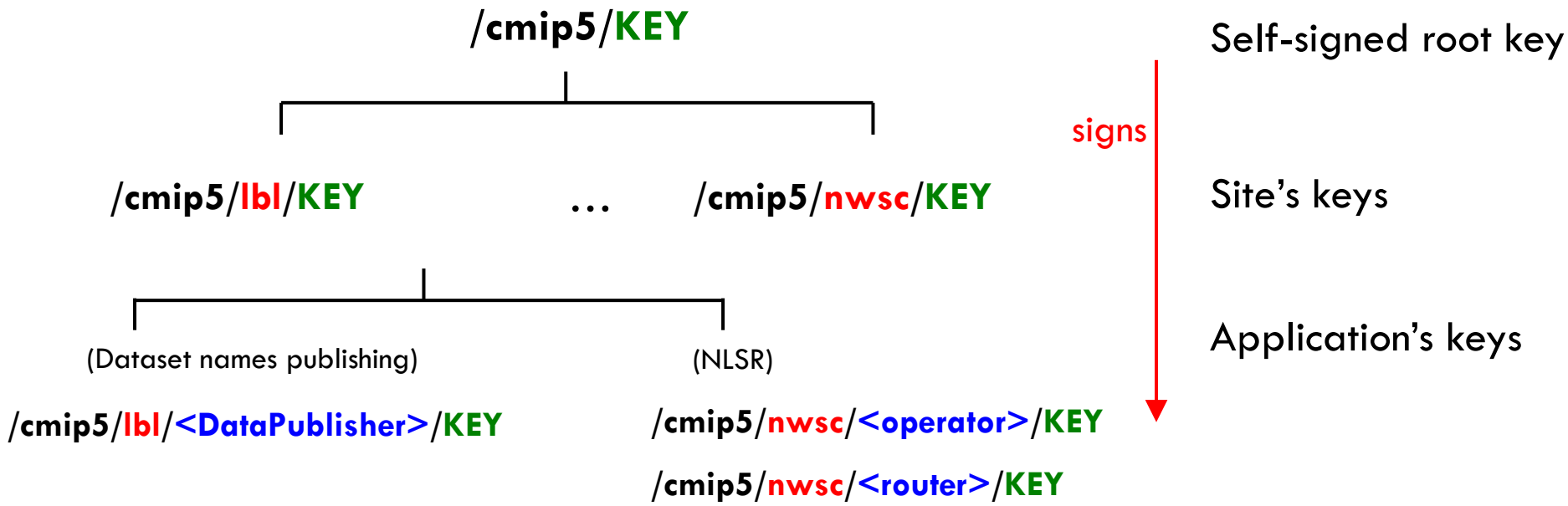


Self-signed root key

Site's keys

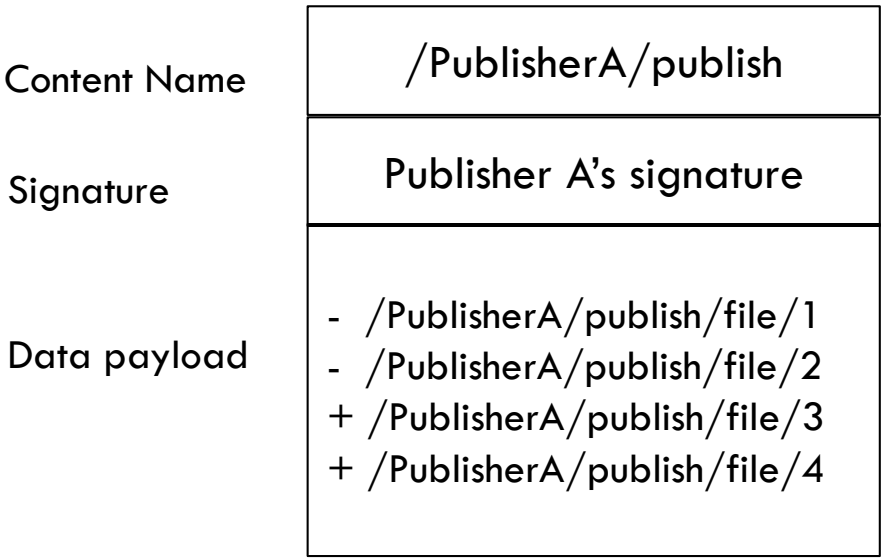
Application's keys

Keys for ndn-atmos



Trust Model

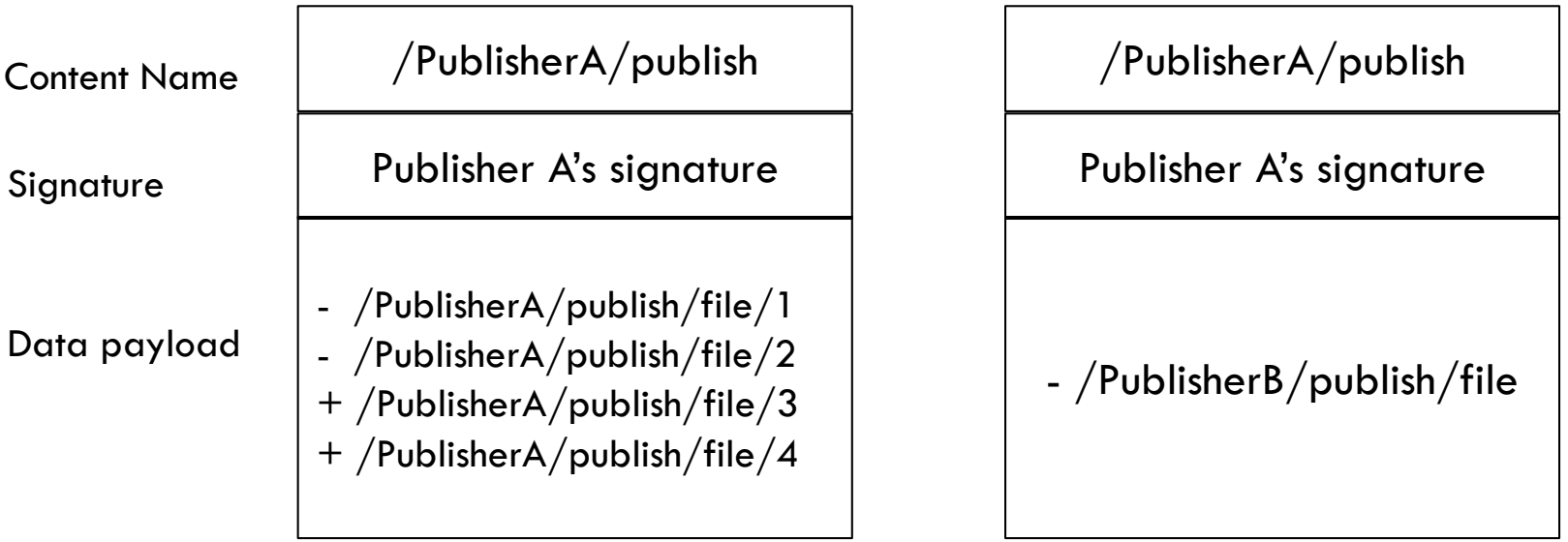
- Only namespace owners are allowed to publish data
  - Data provenance built into the data packet



Valid publish message

Trust Model

- Only namespace owners are allowed to publish data
  - Data provenance built into the data packet

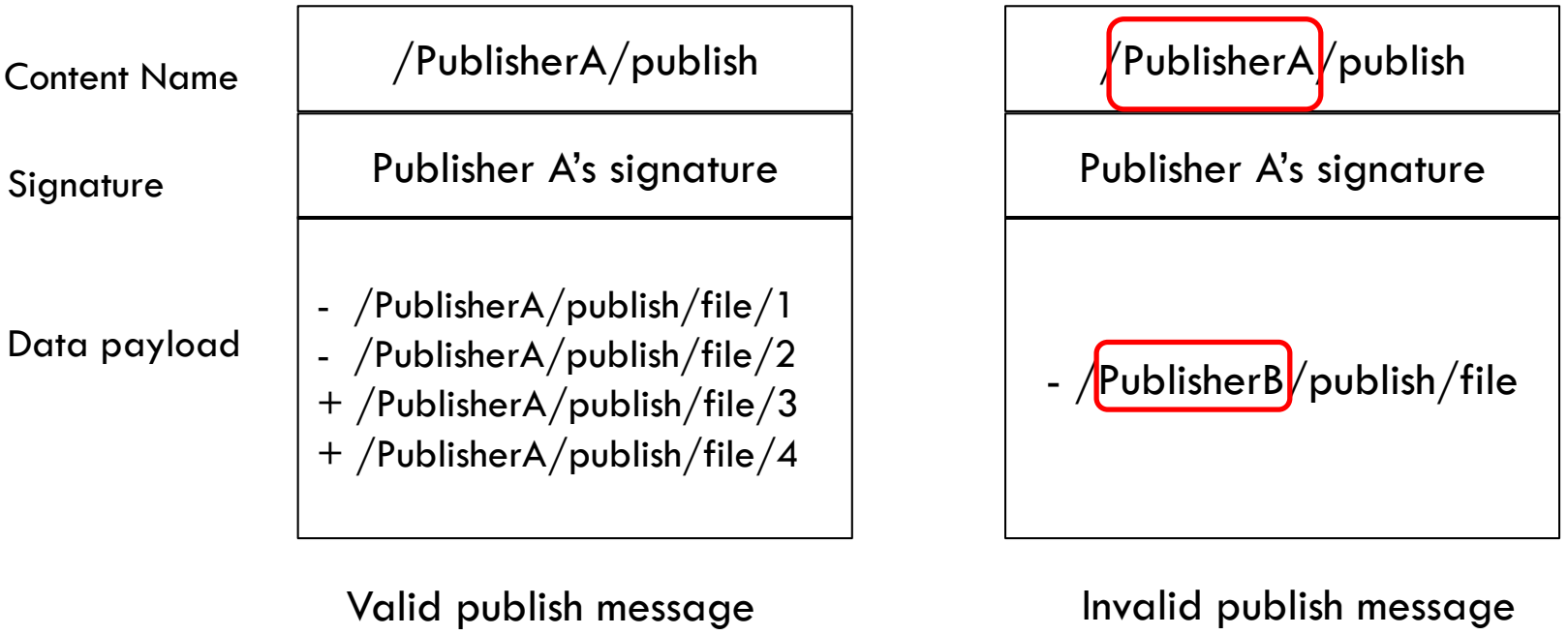


Valid publish message

Invalid publish message

Trust Model

- Only namespace owners are allowed to publish data
  - Data provenance built into the data packet

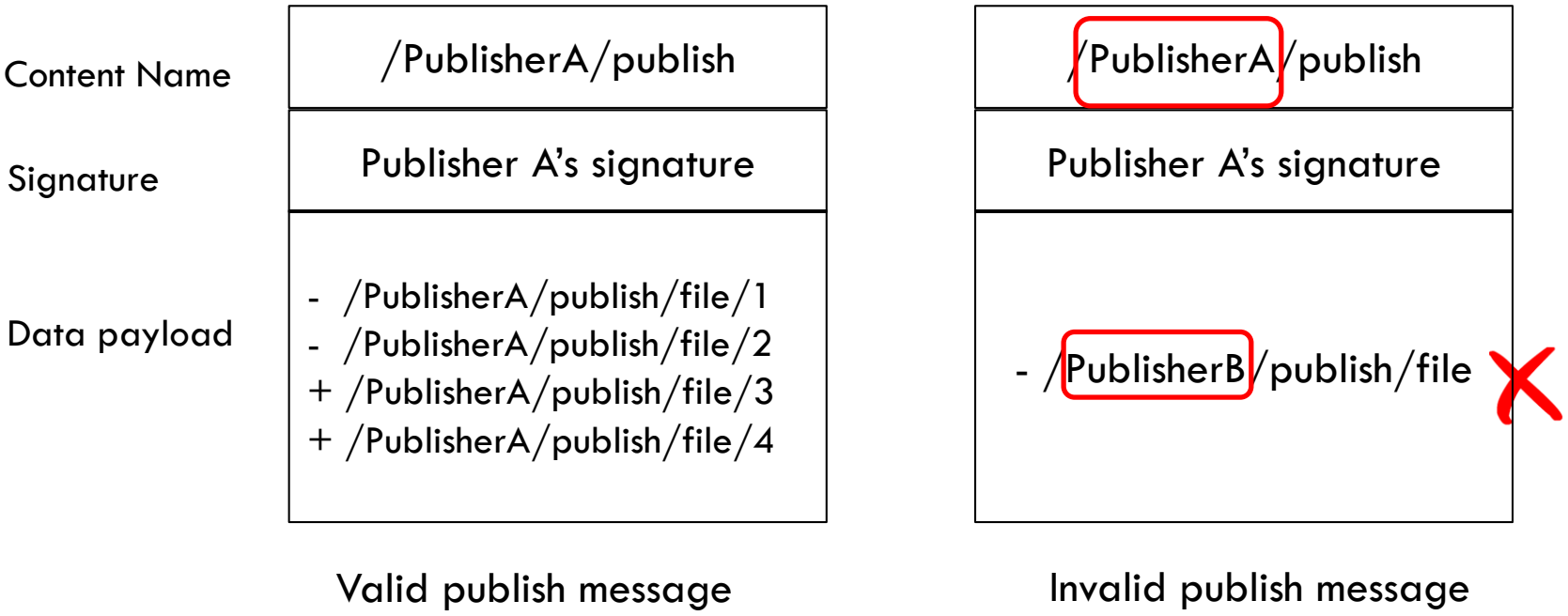


Valid publish message

Invalid publish message

Trust Model

- Only namespace owners are allowed to publish data
  - Data provenance built into the data packet



Name Discovery

Catalog

Consumer

1) Listening on /<catalog-prefix>/query

Name Discovery

Catalog

Consumer

1) Listening on /<catalog-prefix>/query

2) Query with parameters (model=cmip5 AND frequency=6hr)

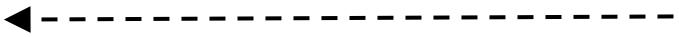
Name Discovery

Catalog

Consumer

1) Listening on /<catalog-prefix>/query

2) Query with parameters (model=cmip5 AND frequency=6hr)



3) Query local DB; Packetize results under /<catalog-prefix>/query-results/<params>

Name Discovery

Catalog

Consumer

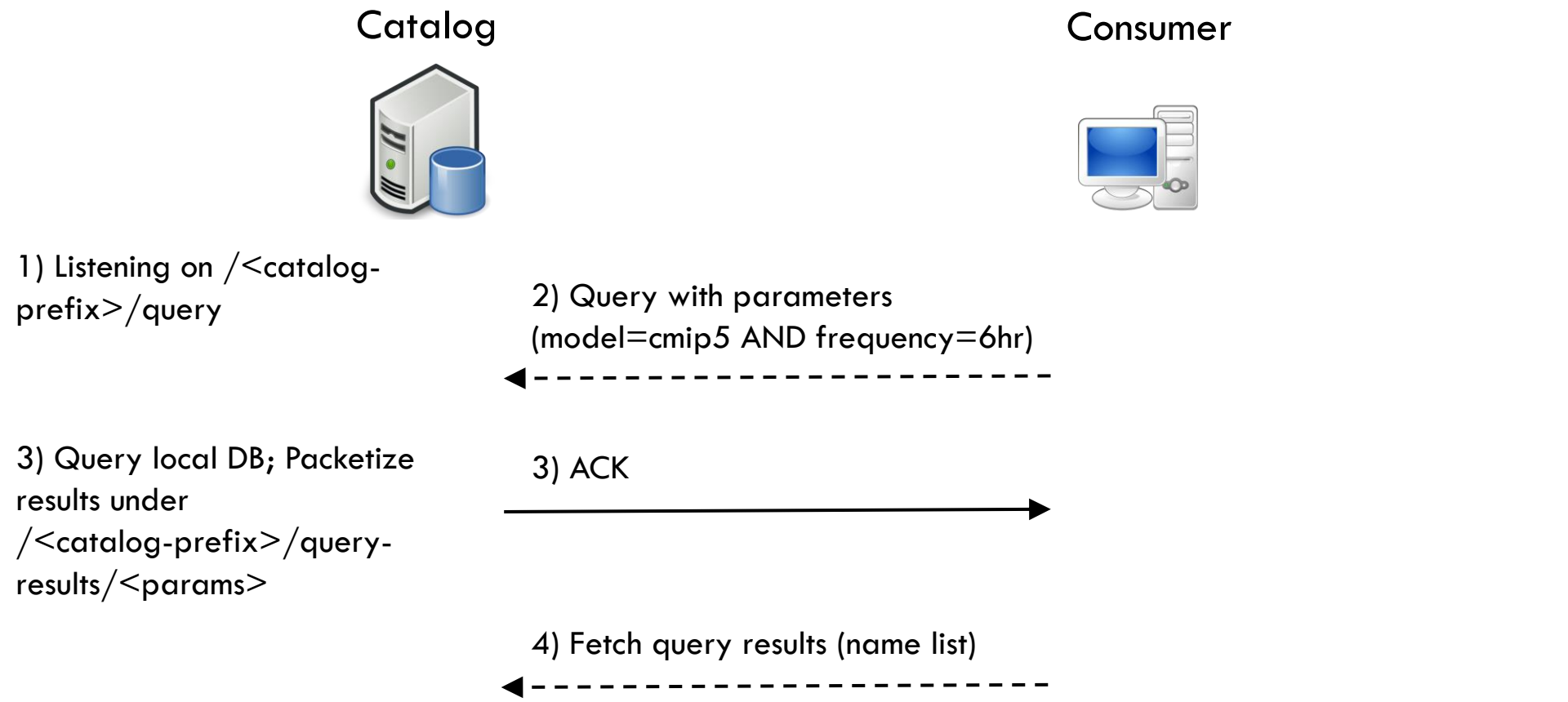
1) Listening on /<catalog-prefix>/query

2) Query with parameters (model=cmip5 AND frequency=6hr)

3) Query local DB; Packetize results under /<catalog-prefix>/query-results/<params>

3) ACK

Name Discovery



Name Discovery

Catalog

Consumer

1) Listening on /<catalog-prefix>/query

2) Query with parameters (model=cmip5 AND frequency=6hr)

3) Query local DB; Packetize results under /<catalog-prefix>/query-results/<params>

3) ACK

4) Fetch query results (name list)

5) Fetch desired dataset(s) or re-query

Data Publication

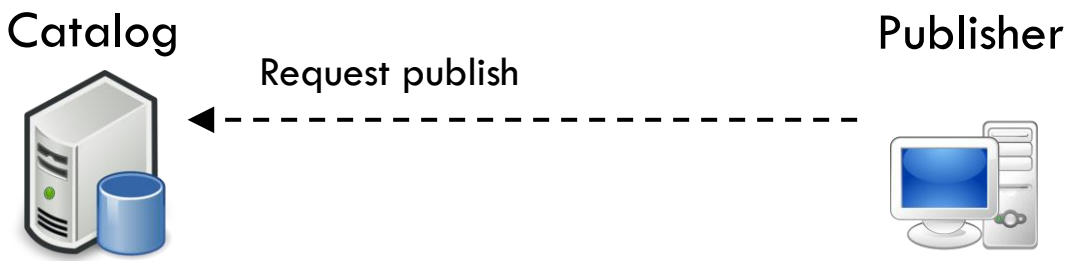
- Catalog
  - Accept publish requests:
    /<catalog-prefix>/publish
  - Authenticate and retrieve data names from publisher
  - Sync names with other catalogs
- Publisher
  - Generate NDN names for datasets/services
  - Inform catalog of names to add/remove

Catalog

Publisher

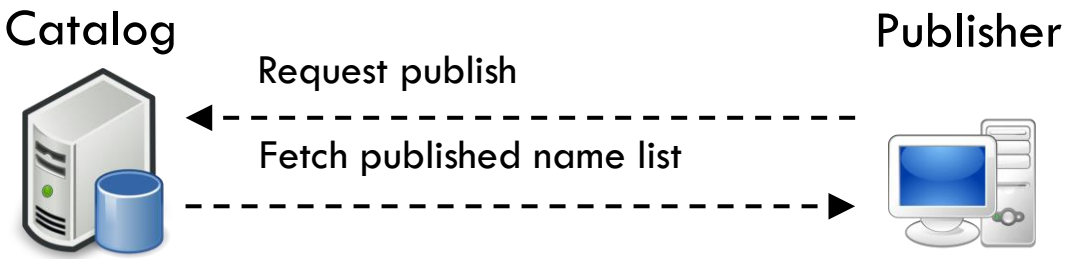
Data Publication

- Catalog
  - Accept publish requests:
    /<catalog-prefix>/publish
  - Authenticate and retrieve data names from publisher
  - Sync names with other catalogs
- Publisher
  - Generate NDN names for datasets/services
  - Inform catalog of names to add/remove



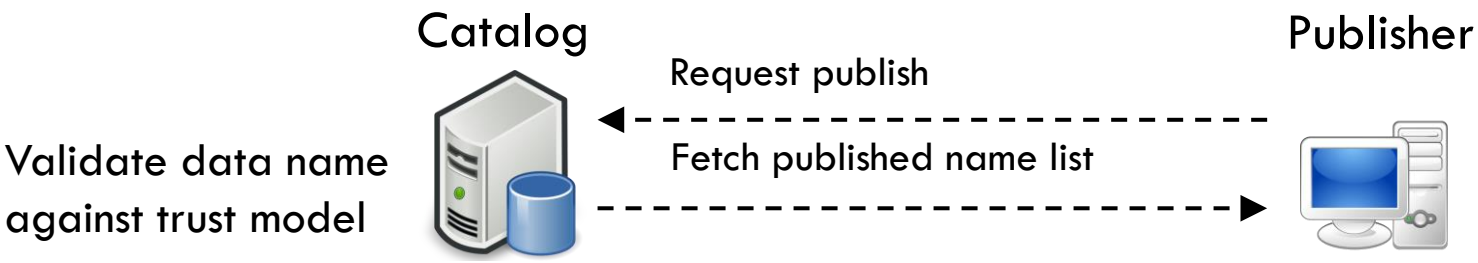
Data Publication

- Catalog
  - Accept publish requests:
    /<catalog-prefix>/publish
  - Authenticate and retrieve data names from publisher
  - Sync names with other catalogs
- Publisher
  - Generate NDN names for datasets/services
  - Inform catalog of names to add/remove



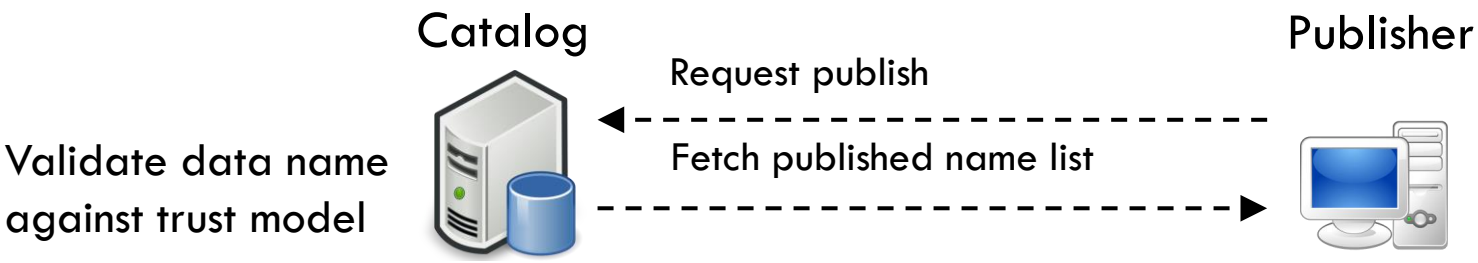
Data Publication

- Catalog
  - Accept publish requests:
    /<catalog-prefix>/publish
  - Authenticate and retrieve data names from publisher
  - Sync names with other catalogs
- Publisher
  - Generate NDN names for datasets/services
  - Inform catalog of names to add/remove



Data Publication

- Catalog
  - Accept publish requests:
    /<catalog-prefix>/publish
  - Authenticate and retrieve data names from publisher
  - Sync names with other catalogs
- Publisher
  - Generate NDN names for datasets/services
  - Inform catalog of names to add/remove



Name Discovery

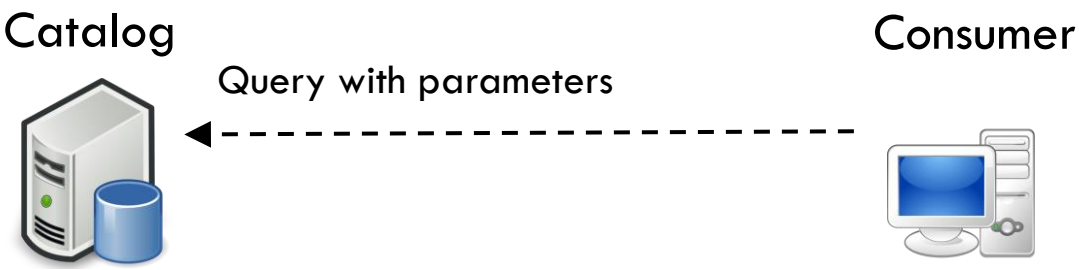
- Catalog
  - Accept queries on
    /<catalog-prefix>/query
  - Query local DB
  - Packetize the returned names under
    /<catalog-prefix>/query-results/<params>
- User
  - Query catalog for names with specified components
    - e.g.: model=cmip5 AND frequency=6hr
  - Fetch generated name list
  - Fetch desired dataset(s) or re-query

Catalog

Consumer

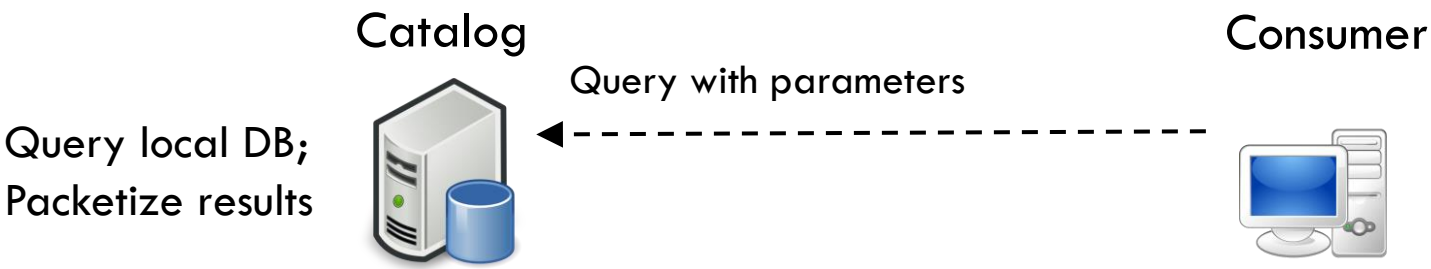
Name Discovery

- Catalog
  - Accept queries on
    /<catalog-prefix>/query
  - Query local DB
  - Packetize the returned names under
    /<catalog-prefix>/query-results/<params>
- User
  - Query catalog for names with specified components
    - e.g.: model=cmip5 AND frequency=6hr
  - Fetch generated name list
  - Fetch desired dataset(s) or re-query



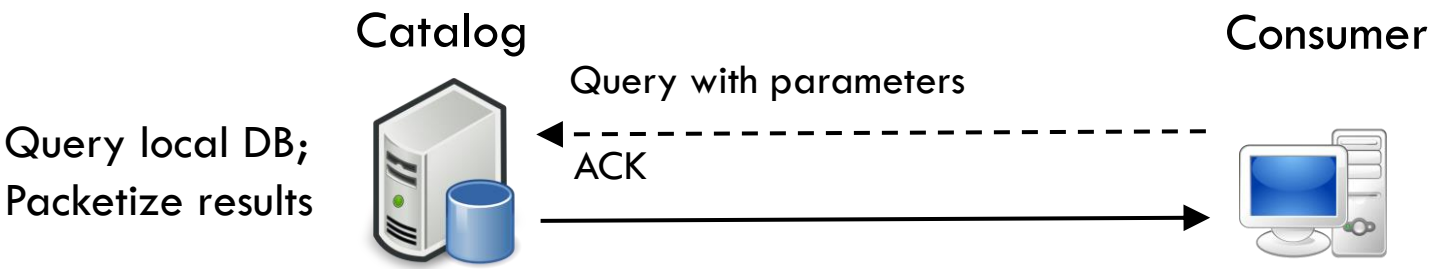
Name Discovery

- Catalog
  - Accept queries on /<catalog-prefix>/query
  - Query local DB
  - Packetize the returned names under /<catalog-prefix>/query-results/<params>
- User
  - Query catalog for names with specified components
    - e.g.: model=cmip5 AND frequency=6hr
  - Fetch generated name list
  - Fetch desired dataset(s) or re-query



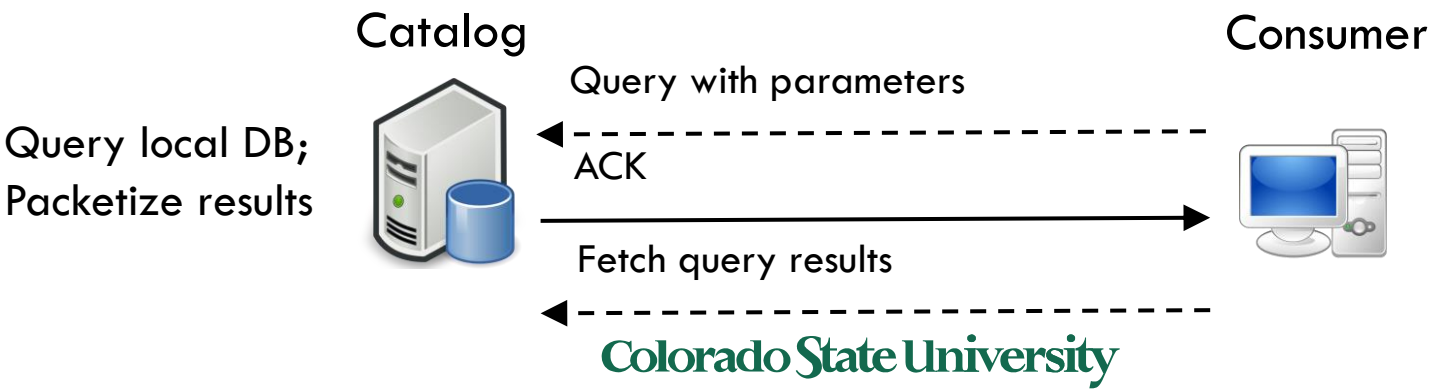
Name Discovery

- Catalog
  - Accept queries on /<catalog-prefix>/query
  - Query local DB
  - Packetize the returned names under /<catalog-prefix>/query-results/<params>
- User
  - Query catalog for names with specified components
    - e.g.: model=cmip5 AND frequency=6hr
  - Fetch generated name list
  - Fetch desired dataset(s) or re-query



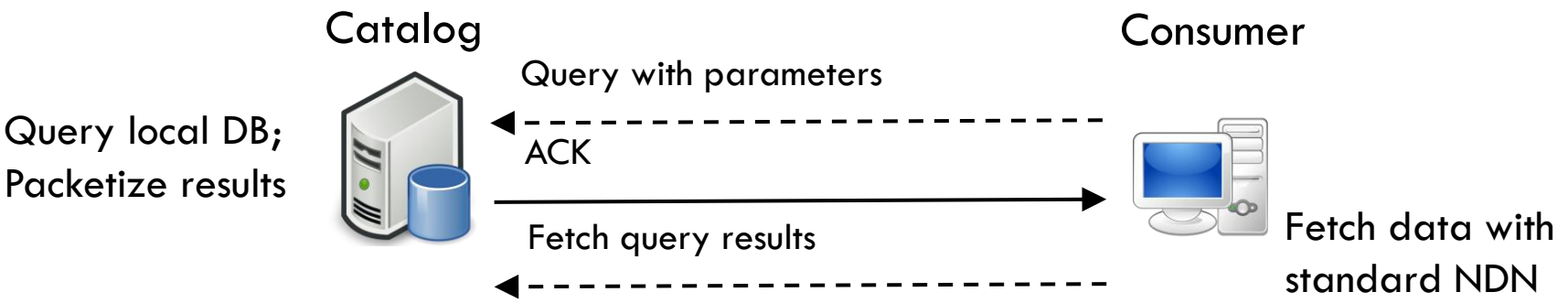
Name Discovery

- Catalog
  - Accept queries on /<catalog-prefix>/query
  - Query local DB
  - Packetize the returned names under /<catalog-prefix>/query-results/<params>
- User
  - Query catalog for names with specified components
    - e.g.: model=cmip5 AND frequency=6hr
  - Fetch generated name list
  - Fetch desired dataset(s) or re-query



Name Discovery

- Catalog
  - Accept queries on /<catalog-prefix>/query
  - Query local DB
  - Packetize the returned names under /<catalog-prefix>/query-results/<params>
- User
  - Query catalog for names with specified components
    - e.g.: model=cmip5 AND frequency=6hr
  - Fetch generated name list
  - Fetch desired dataset(s) or re-query



Name Discovery Optimization

- Avoid maintaining state between user and catalog
  - Enables graceful failover

- Catalog
  - Accept queries on

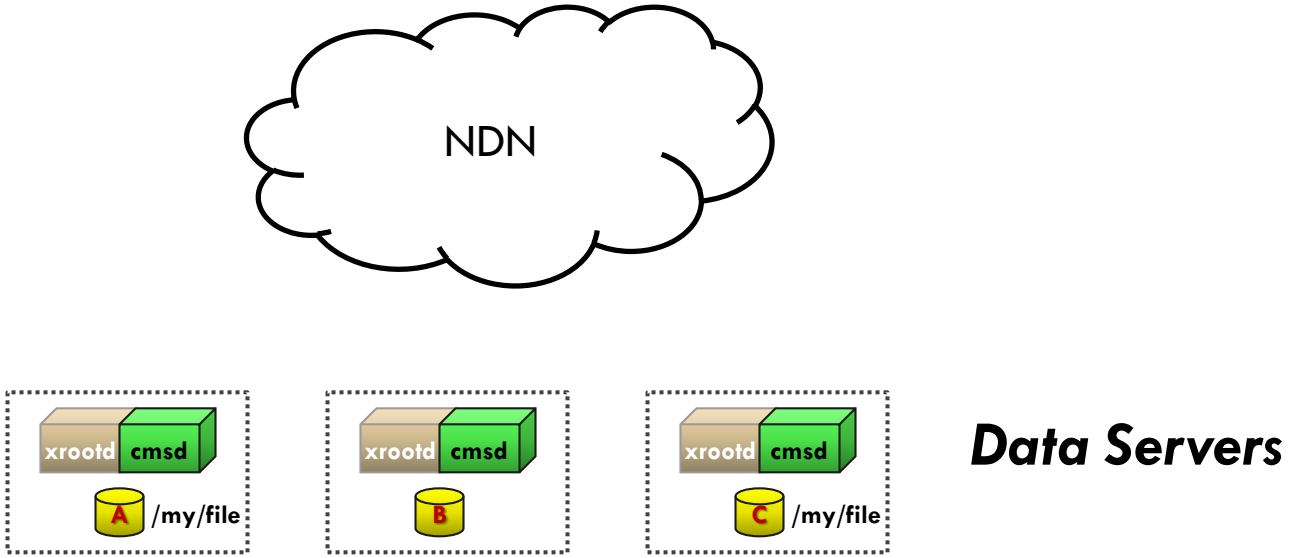
    /<catalog-prefix>/queryParams
  - Query local DB
  - Packetize the returned names under

    /<catalog-prefix>/queryParams/seg#
  - In case of failure, queries get redirected to another catalog

- Consumers
  - Can query any catalog instances
  - Can transparently failover to another catalog

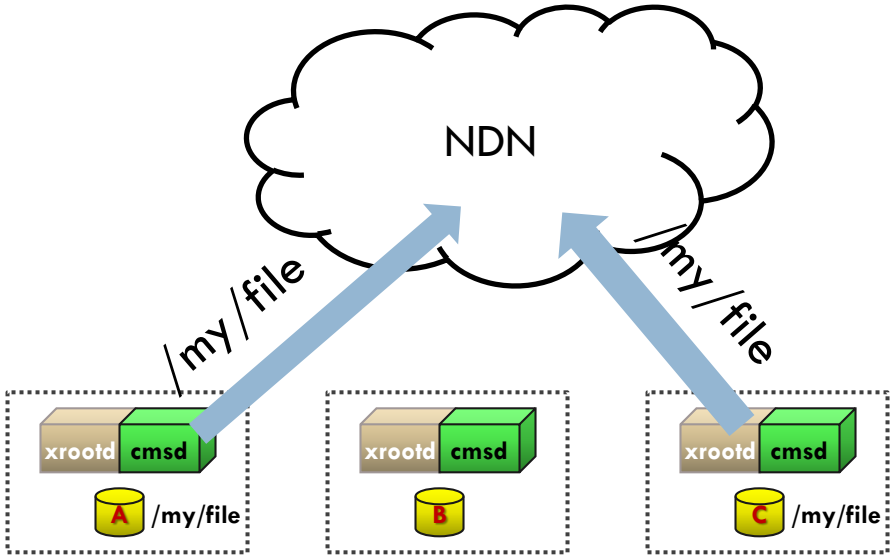
Simplified xrootd Under NDN

- NDN integrates discovery, failover, retrieval …
  - Provides a better abstraction to the applications



Simplified xrootd Under NDN

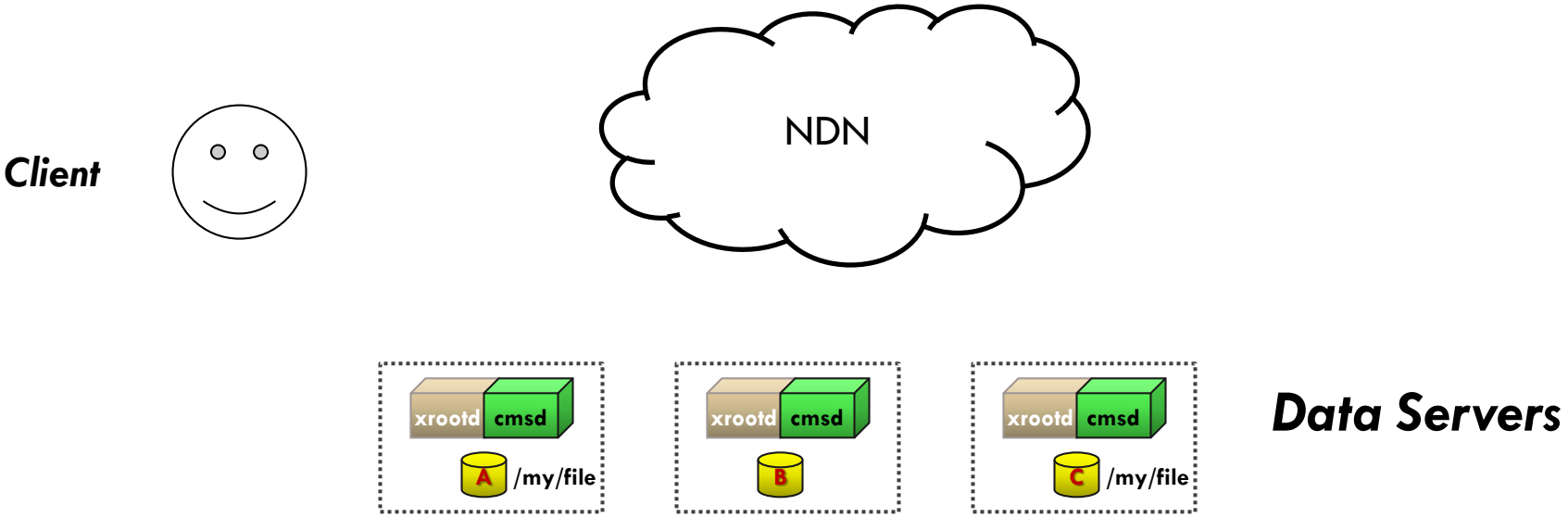
- NDN integrates discovery, failover, retrieval …
  - Provides a better abstraction to the applications



Data Servers

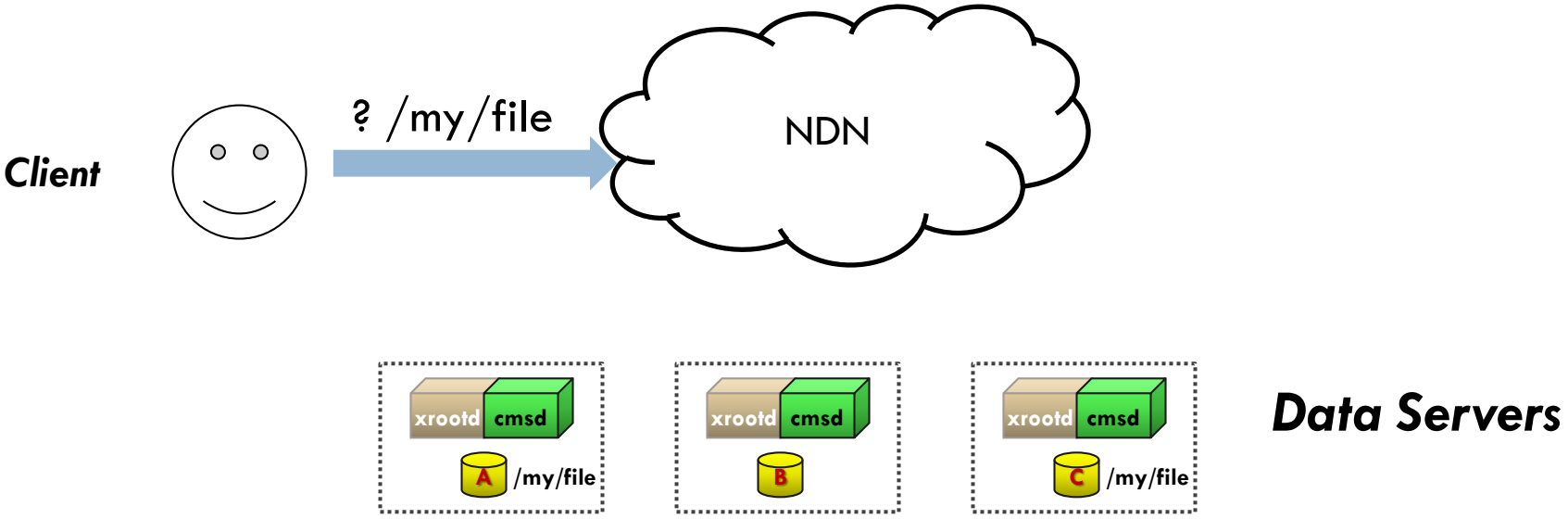
Simplified xrootd Under NDN

- NDN integrates discovery, failover, retrieval …
  - Provides a better abstraction to the applications



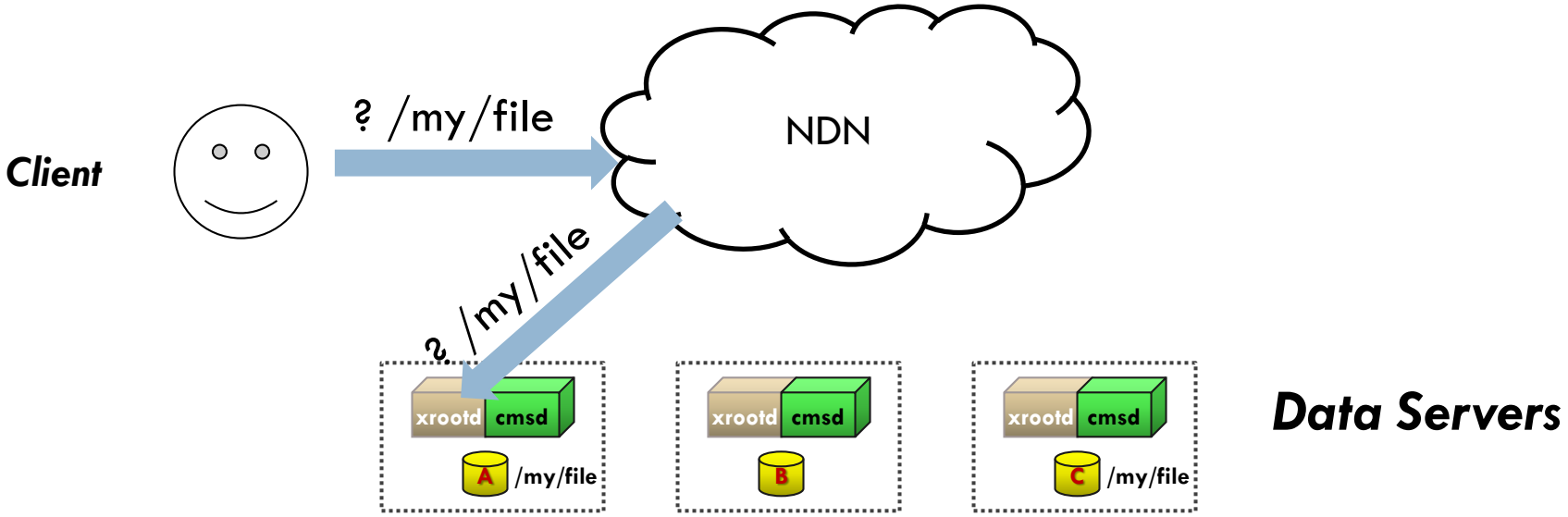
Simplified xrootd Under NDN

- NDN integrates discovery, failover, retrieval …
  - Provides a better abstraction to the applications



Simplified xrootd Under NDN

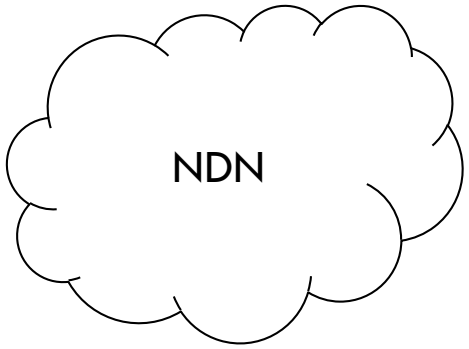
- NDN integrates discovery, failover, retrieval …
  - Provides a better abstraction to the applications



Name Discovery Challenges

- Users may need to discover content/services without knowing a the full NDN name prefix structure
  - NDN names are contiguous prefixes
  - Users may only know a few disjoint name components (e.g. frequency=6hr)
  - But can not use wildcards for name discovery

User wants: /CMIP5/output1/VA/6hr/2016

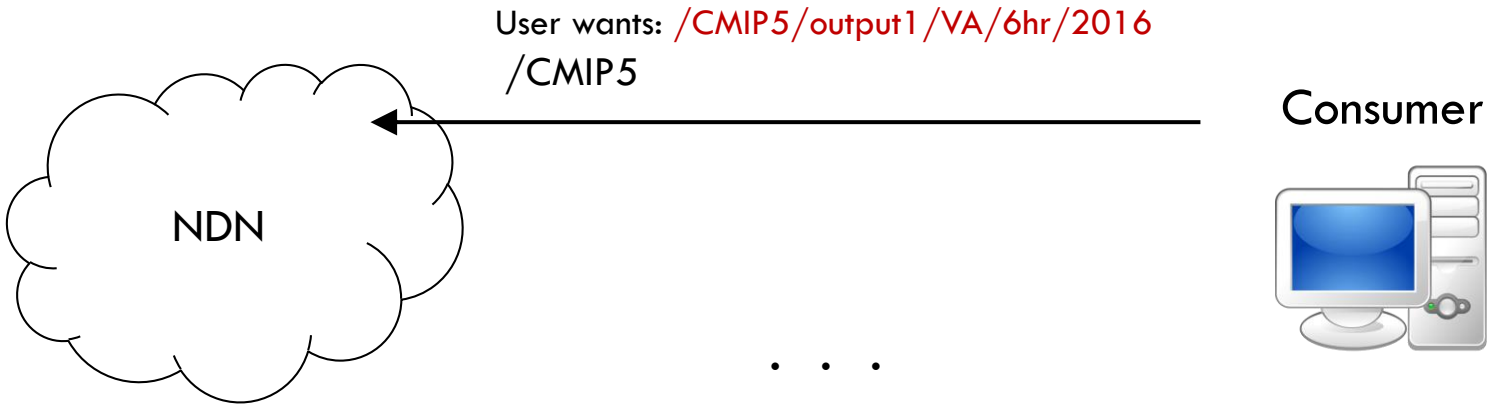


. . .

Consumer

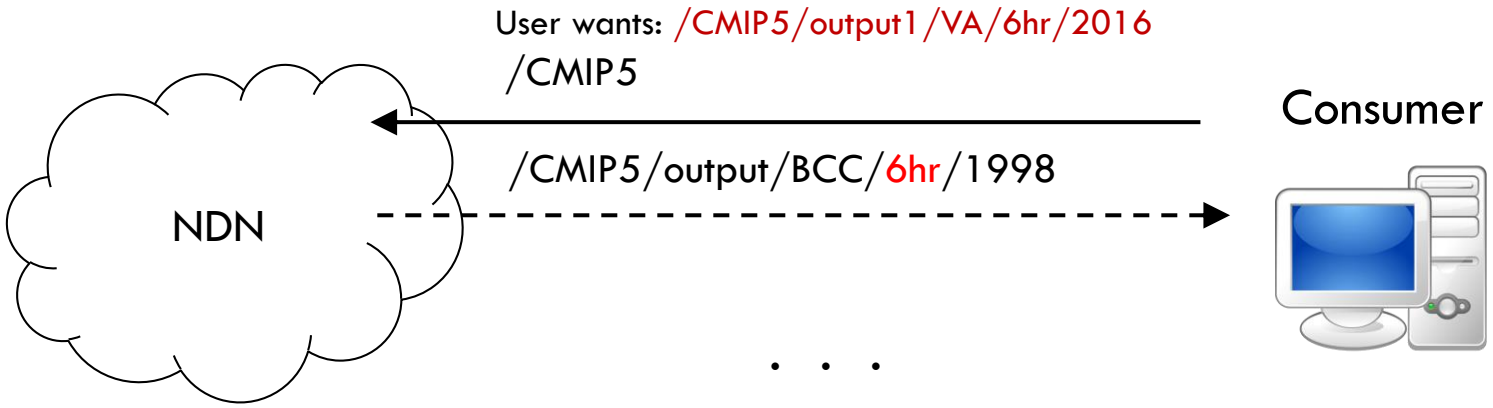
Name Discovery Challenges

- Users may need to discover content/services without knowing a the full NDN name prefix structure
  - NDN names are contiguous prefixes
  - Users may only know a few disjoint name components (e.g. frequency=6hr)
  - But can not use wildcards for name discovery



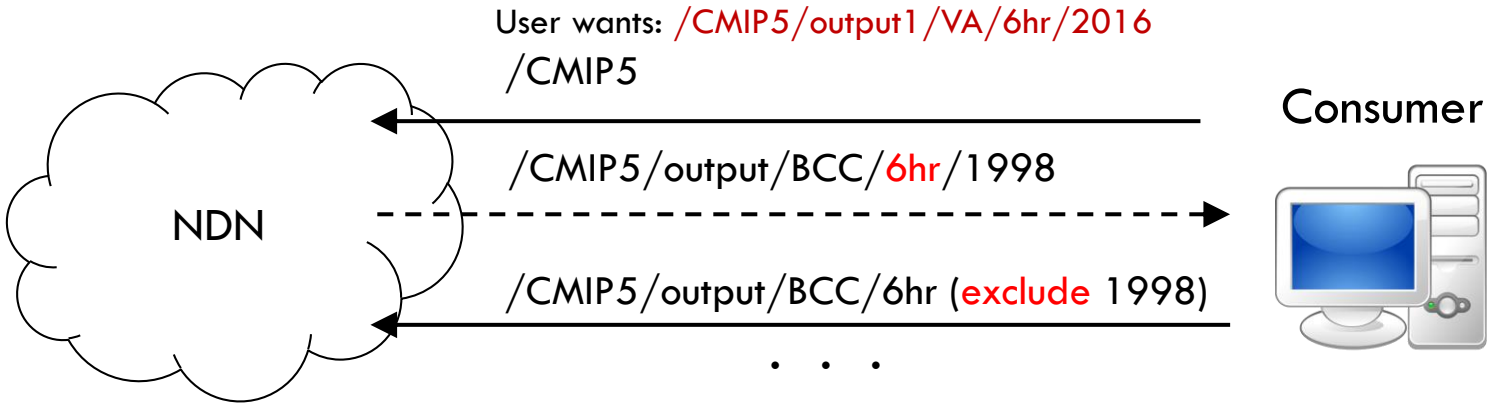
Name Discovery Challenges

- Users may need to discover content/services without knowing a the full NDN name prefix structure
  - NDN names are contiguous prefixes
  - Users may only know a few disjoint name components (e.g. frequency=6hr)
  - But can not use wildcards for name discovery



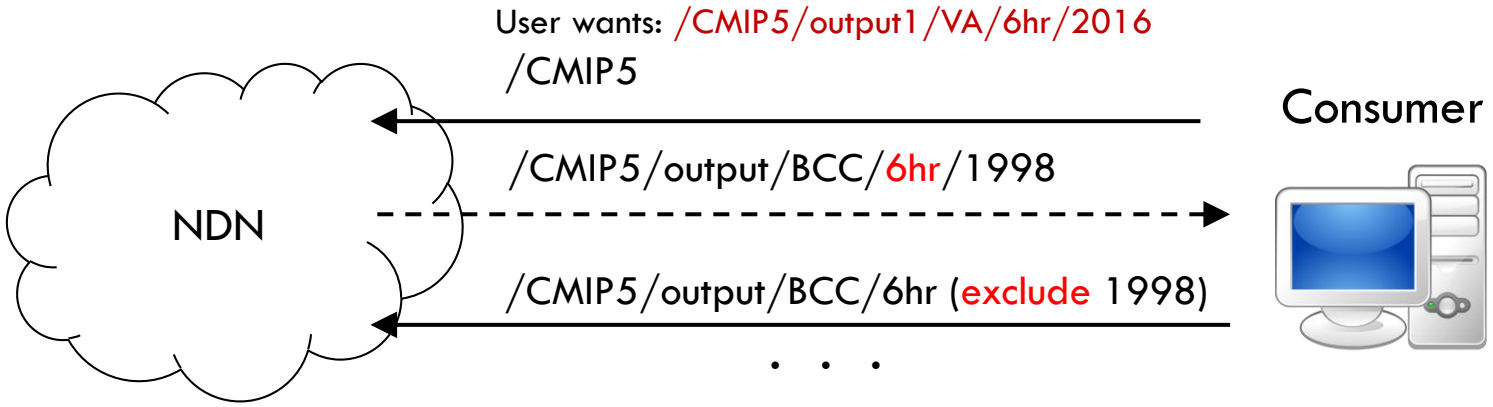
Name Discovery Challenges

- Users may need to discover content/services without knowing a the full NDN name prefix structure
  - NDN names are contiguous prefixes
  - Users may only know a few disjoint name components (e.g. frequency=6hr)
  - But can not use wildcards for name discovery



Name Discovery Challenges

- Users may need to discover content/services without knowing a the full NDN name prefix structure
  - NDN names are contiguous prefixes
  - Users may only know a few disjoint name components (e.g. frequency=6hr)
  - But can not use wildcards for name discovery



May take too many requests to find desired data or service

NDN Support for Big Science

- NDN Names separate data from hosts
  - Discovery: Names directly translate to network queries
  - Failover: Network can get verifiable data from anywhere
  - Retrieval: Data can be fetched from optimal source(s)

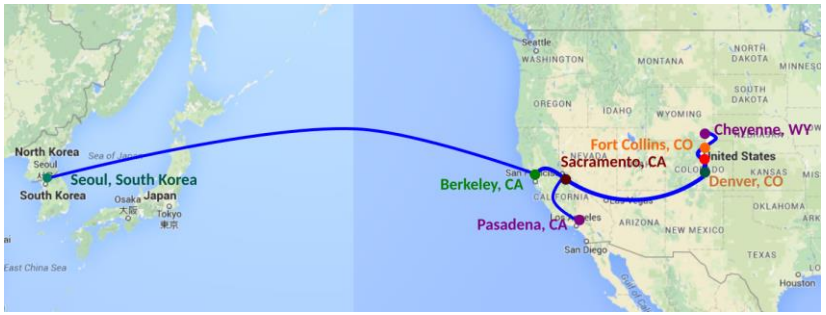
- Investigate the use of NDN as a platform for scientific data applications
  - Understand data management challenges of various scientific domains
  - Develop prototype applications to leverage NDN's built-in features
  - Use these applications as case studies to drive NDN research aspects

Summary

- NDN improves scientific data management at scale
  - Apps benefit from transparent multipath, automatic failover, etc.
  - Built-in security provides publisher provenance
- Names are the common building block for content and services
  - Names are flexible: can refer to static content or dynamic services
- Catalog supports efficient publication, non-contiguous name discovery
  - Users can discover content and services with minimal a priori knowledge
  - Catalog validates publication requests for authorization

Managing Scientific Data with NDN

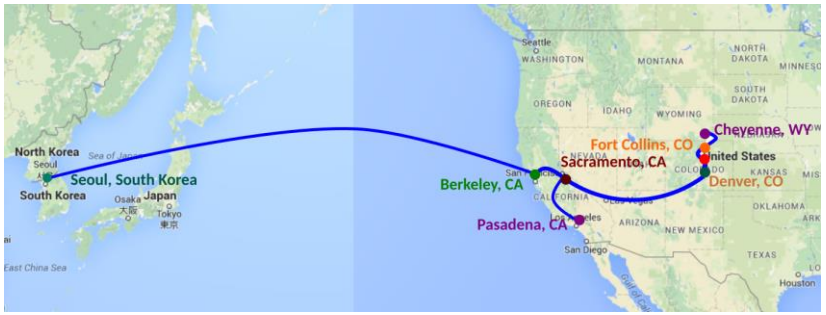
- Distributed, synchronized catalog of names and services
  - Common functionality: publishing, discovery, access control, etc.
  - Search and retrieval UI
  - Platform for further research and experimentation
- Research questions:
  - Namespace construction, distributed publishing, key management, UI design, failover, etc.
  - Functional services such as subsetting
  - Mapping of name-based routing to tunneling services (VPN, OSCARS, MPLS)



- Science testbed
  - 10G testbed (courtesy of ESnet, UCAR, and CSU Research LAN)
- Nodes strategically located near scientific data (climate +HEP)
- CC-NIE NSF award

Managing Scientific Data with NDN

- Name-based Internet architecture
  - Name the data, not the host
  - All data digitally signed
  - Unifies and pushes common functionality to the network: publishing, discovery, access control, etc.
- Data Intensive applications
  - Automatic pervasive in-network caching, parallel retrieval, automatic failover and more
  - Simpler alternative middleware implementation e.g., ESGF, xrootd



- Science testbed
  - 10G testbed (courtesy of ESnet, UCAR, and CSU Research LAN)
  - CMIP5 and HEP data
- CC-NIE NSF award