Leveraging NFSv4 to Build a Federated File System Protocol

James Lentini
jlentini@netapp.com
NetApp, Inc.

Outline

- Introduction and overview
  - Motivation, background, and goals
  - Requirements, terms, and definitions
- Architecture and implementation
  - Basic resolution protocol
  - NFSv4 details
  - State of the standardization effort
- Conclusion

Introduction

- FedFs is an open protocol for a cross-platform, federated fileset namespace that can be used to build a very large file system.
  - FedFs is not a file system.
  - FedFs specifies how separate file servers can be joined together to create a common namespace.

Benefits of a Federated Namespace

- Simplified management
  - Clients only need to know how to mount the root (or some other part) of the namespace
- Replication: create copies of the namespace in different locations to provide
  - load balancing
  - high availablity
- Migration: change the physical container of a fileset transparently to clients

Requirements of the FedFs protocol

- Cross-platform: cross-vendor, cross-product, and cross-version
  - No customer lock-in
- Federated: control is decentralized
  - Admins retain control over their systems
- Leverages existing protocols
  - NFSv4, CIFS, LDAP, DNS
- No changes to existing protocols or client software

Influences

- IBM Almaden Research Center's Glamour Project

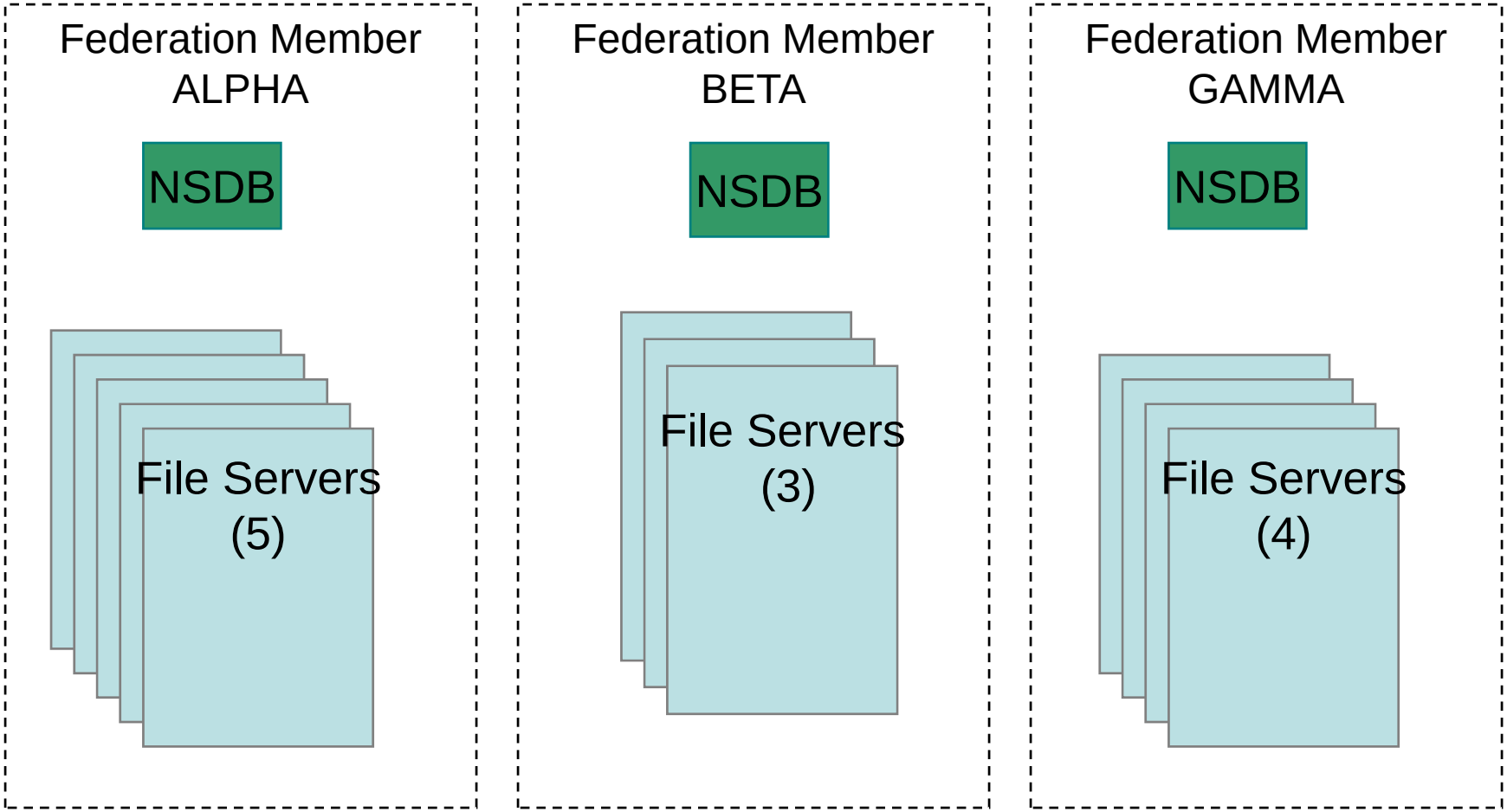
  http://www.almaden.ibm.com/StorageSystems/projects/glamour/

- UMICH CITI NFSv4 Project

  http://www.citi.umich.edu/techreports/reports/citi-tr-06-1.pdf

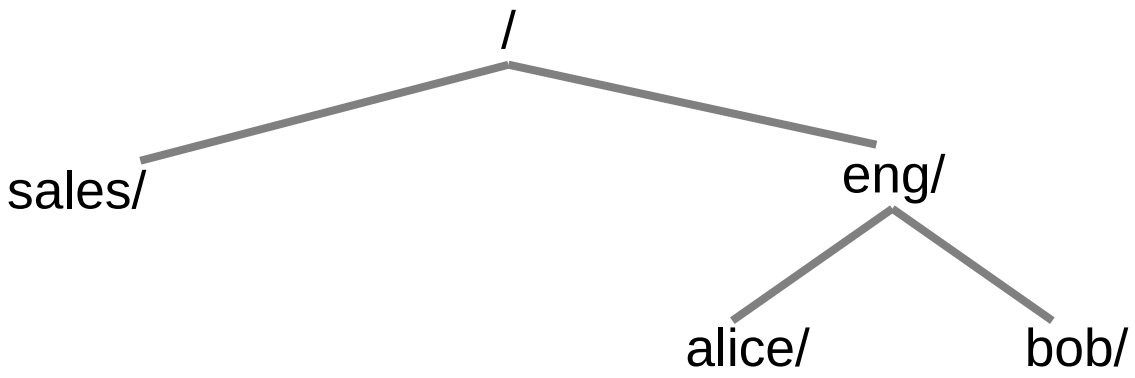
- DCE/DFS, CMU AFS, …

An Example Federation



An example federated namespace

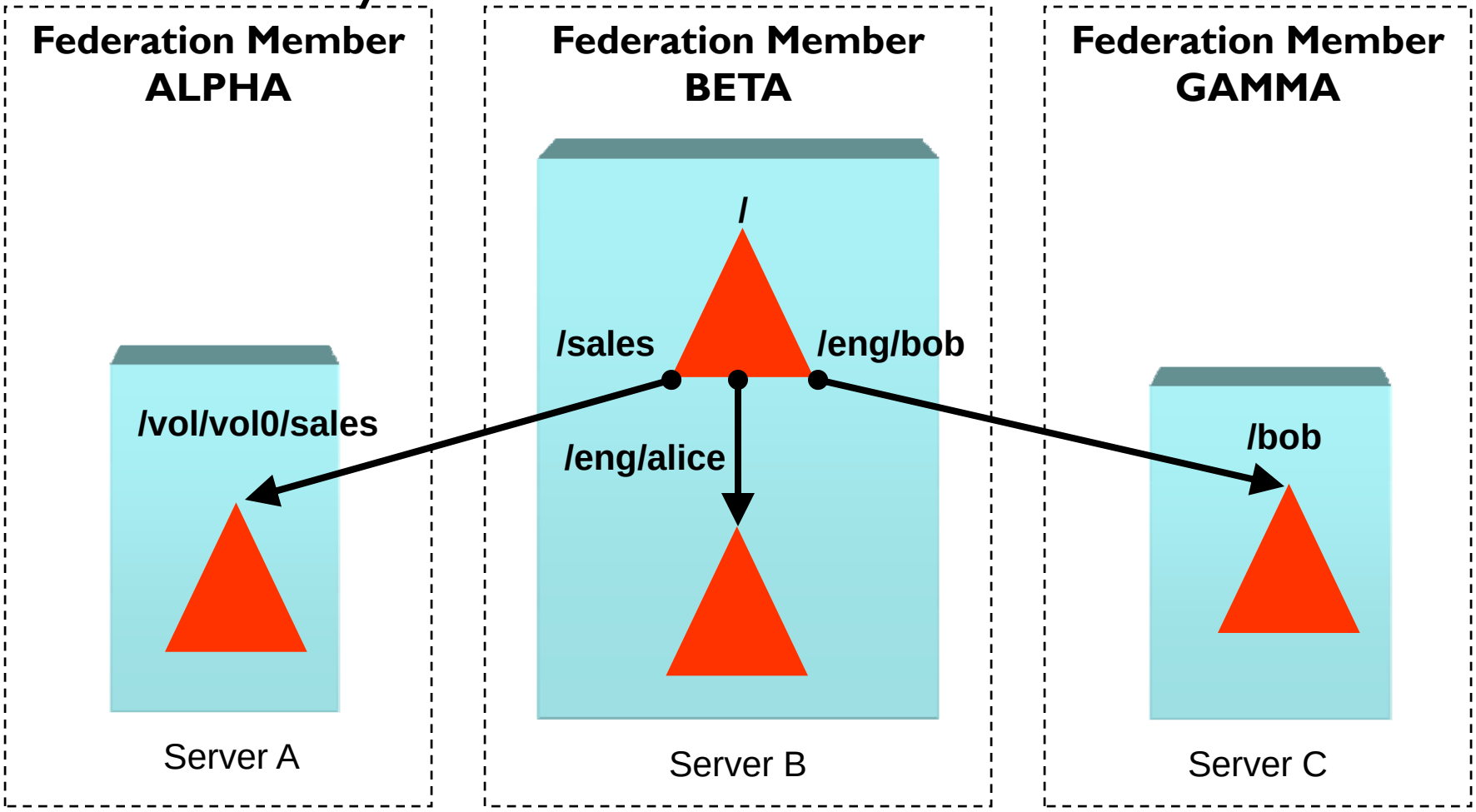
- The illusion:



- A simple hierarchical namespace is what we want the client (and user) to see.
- Behind the scenes, things may be somewhat more complicated…

An example federated namespace

- The reality:



Terms and definitions

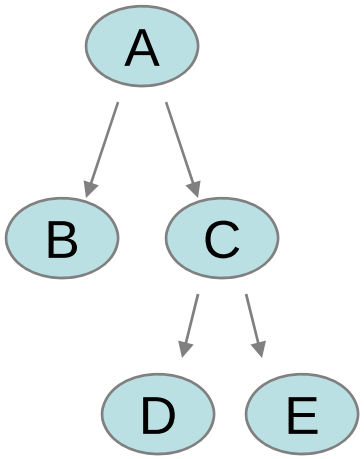
- Fileset: a directory tree (volume)
- FSN (fileset name): a fileset identifier that is independent of the representation of the fileset
  - Each FSN contains an FsnUuid (a UUID) and an NSDB location
- FSL (fileset location): network location of a fileset instance
- Junction: an object that provides a way for one fileset to reference another
- NSDB (namespace database): a service that tracks the mapping between FSNs and FSLs; implemented with LDAP

Outline

- Introduction and overview
  - Motivation, background, and goals
  - Requirements, terms, and definitions
- Architecture and implementation
  - Basic resolution protocol
  - NFSv4 details
  - State of the standardization effort
- Conclusion

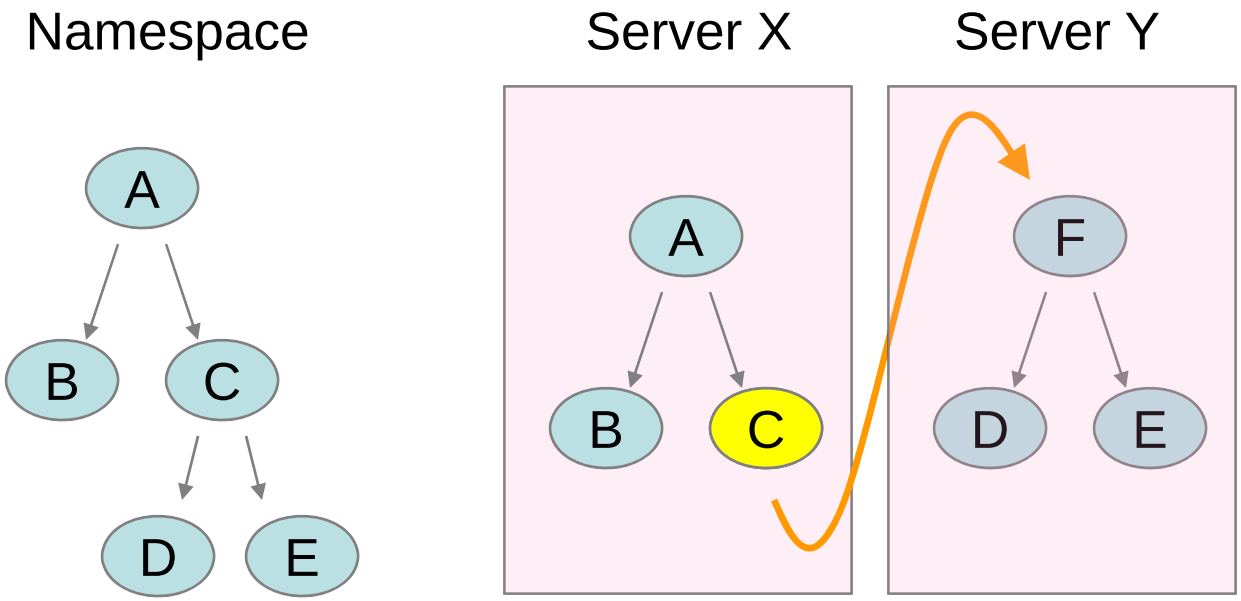
Namespace example

Namespace



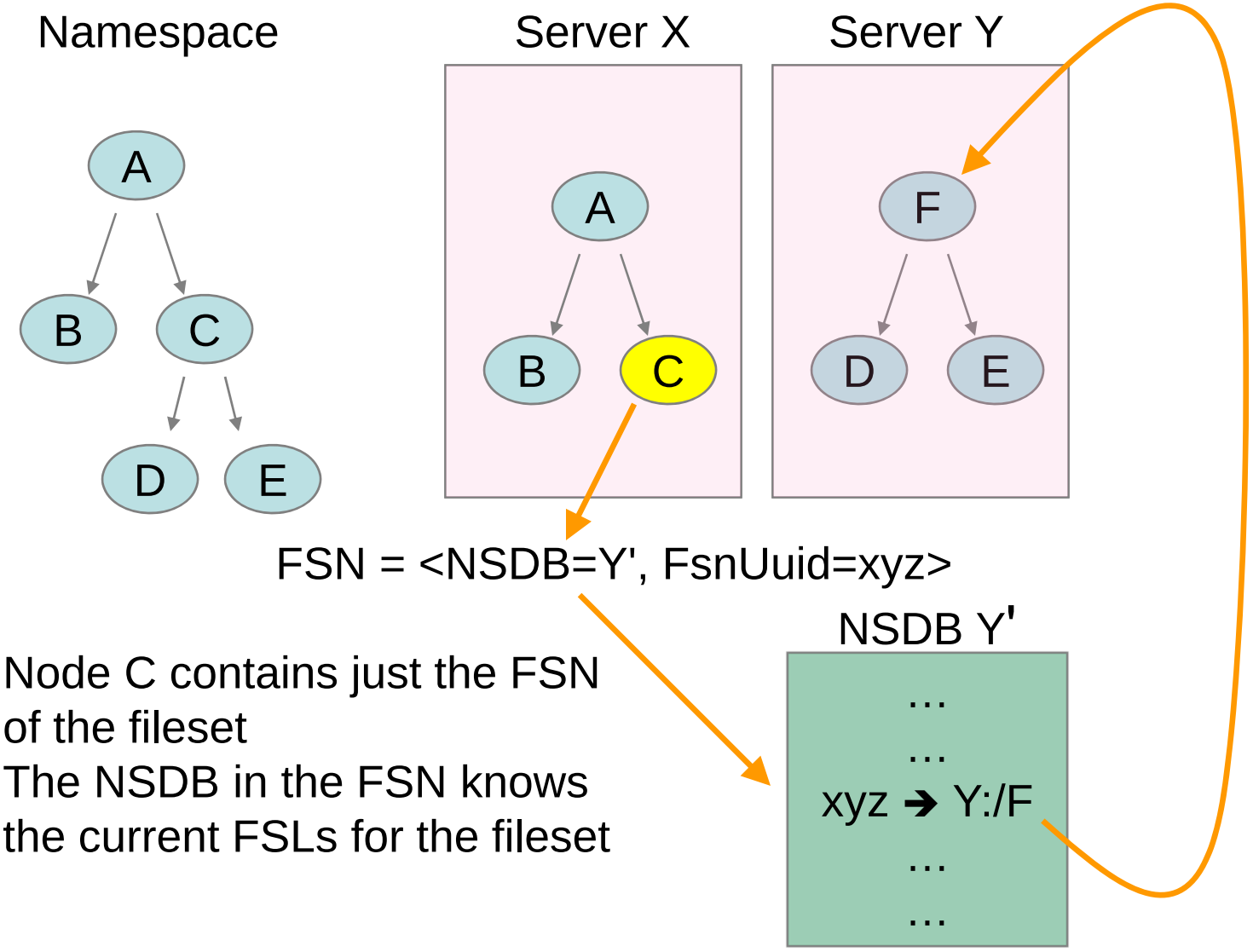
- Goal: store nodes A and B on server X and nodes C, D, and E on server Y

Naïve approach



- Implementation of the namespace is split across two servers.
- Server X knows to redirect accesses from node C to Y:/F.
- Problem: A local change on one server may require changes on another
  - X's Node C must be updated when Y's admin changes the location of F

FedFs approach



- Node C contains just the FSN of the fileset
- The NSDB in the FSN knows the current FSLs for the fileset

FedFs Steps

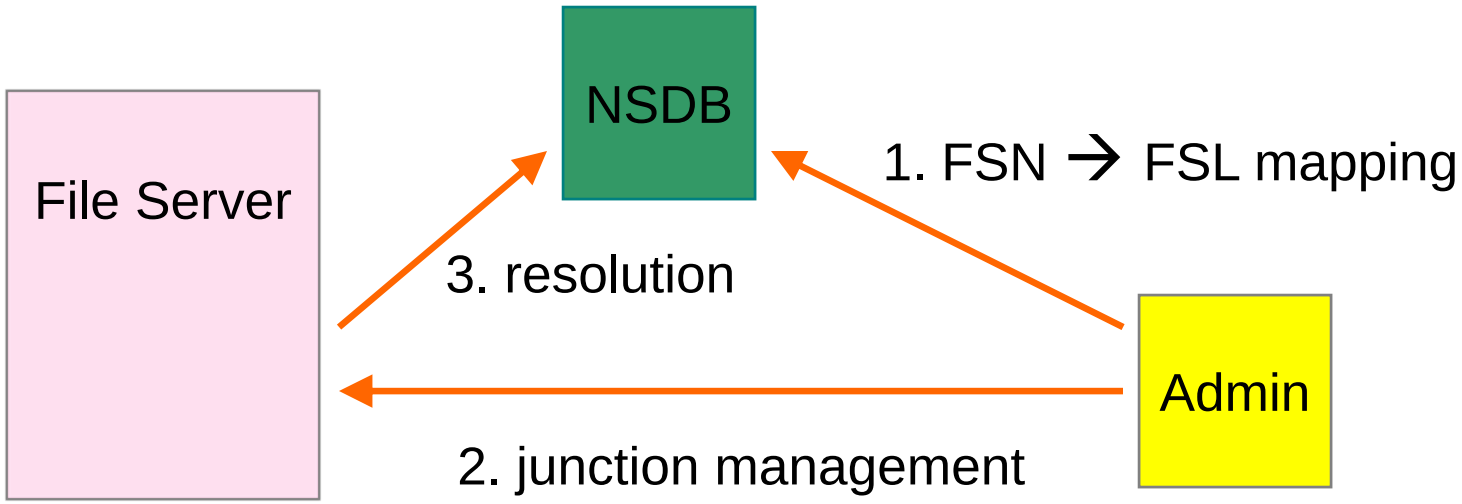
- Setup:
  - Admin creates FSN to FSL mapping(s) in NSDB
  - Admin creates junction on server
- On a client access:
  - Server determines if location is a junction
  - Server resolves junction's FSN to an FSL using the FSN's NSDB
  - Server returns a referral to the client

Three sub-protocols

NFS Client

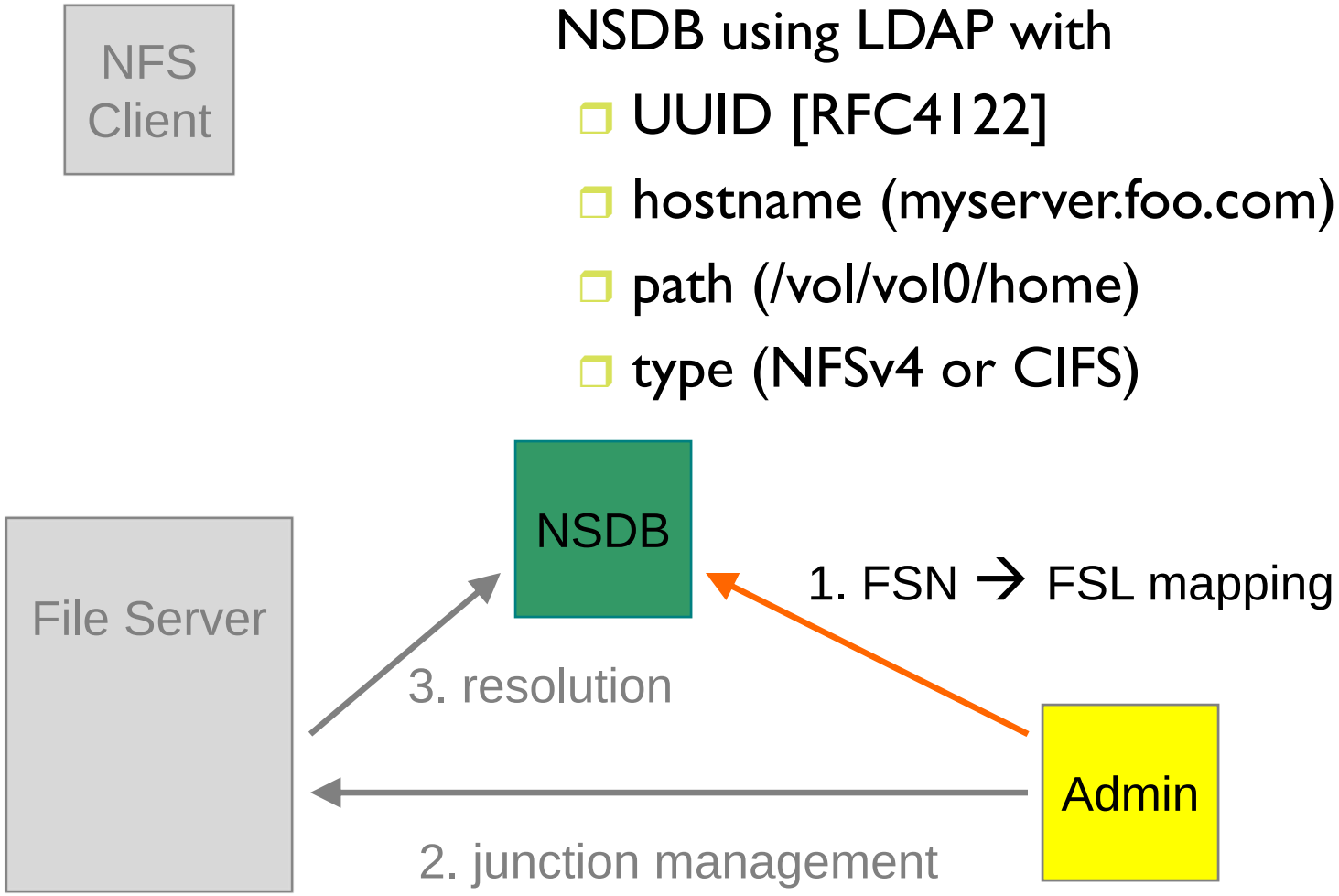
1. Admin to NSDB (FSN → FSL mapping)
2. Admin to server (junction management)
3. Server to NSDB (resolution)

Note: no changes to client protocols



FSN → FSL Mapping

- Admin creates an FSL entry in the NSDB using LDAP with
  - UUID [RFC4122]
  - hostname (myserver.foo.com)
  - path (/vol/vol0/home)
  - type (NFSv4 or CIFS)



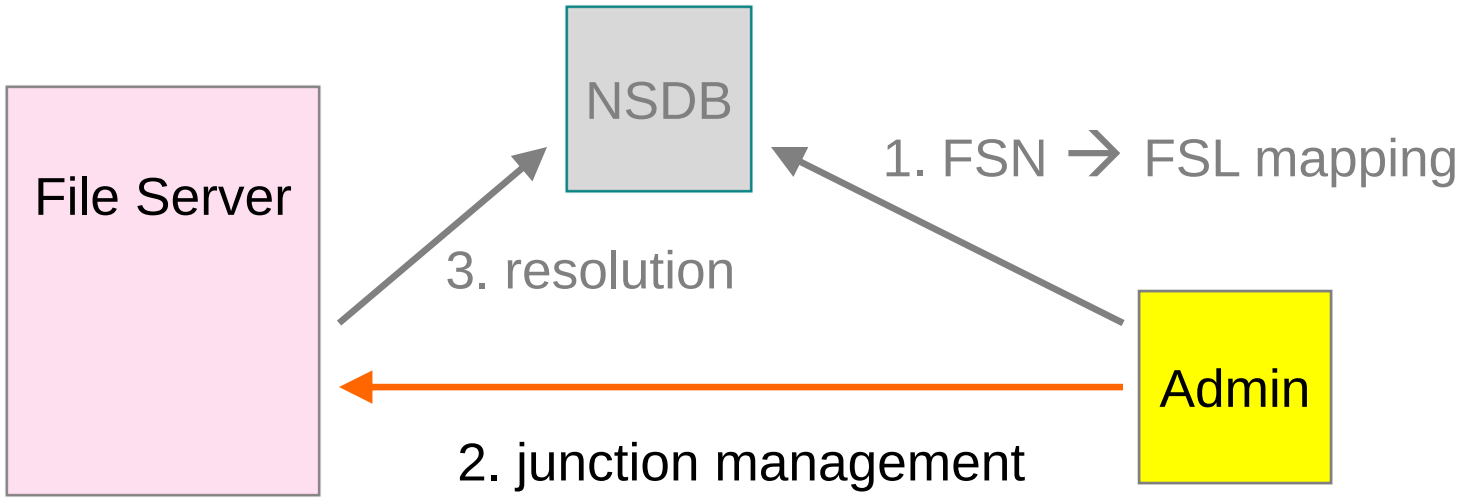
What is a FedFs Junction?

- A filesystem object used to link a directory name in the current fileset to the root of the target fileset
- A leaf object of a fileset
- An object that stitches together the federated namespace

Junction Management

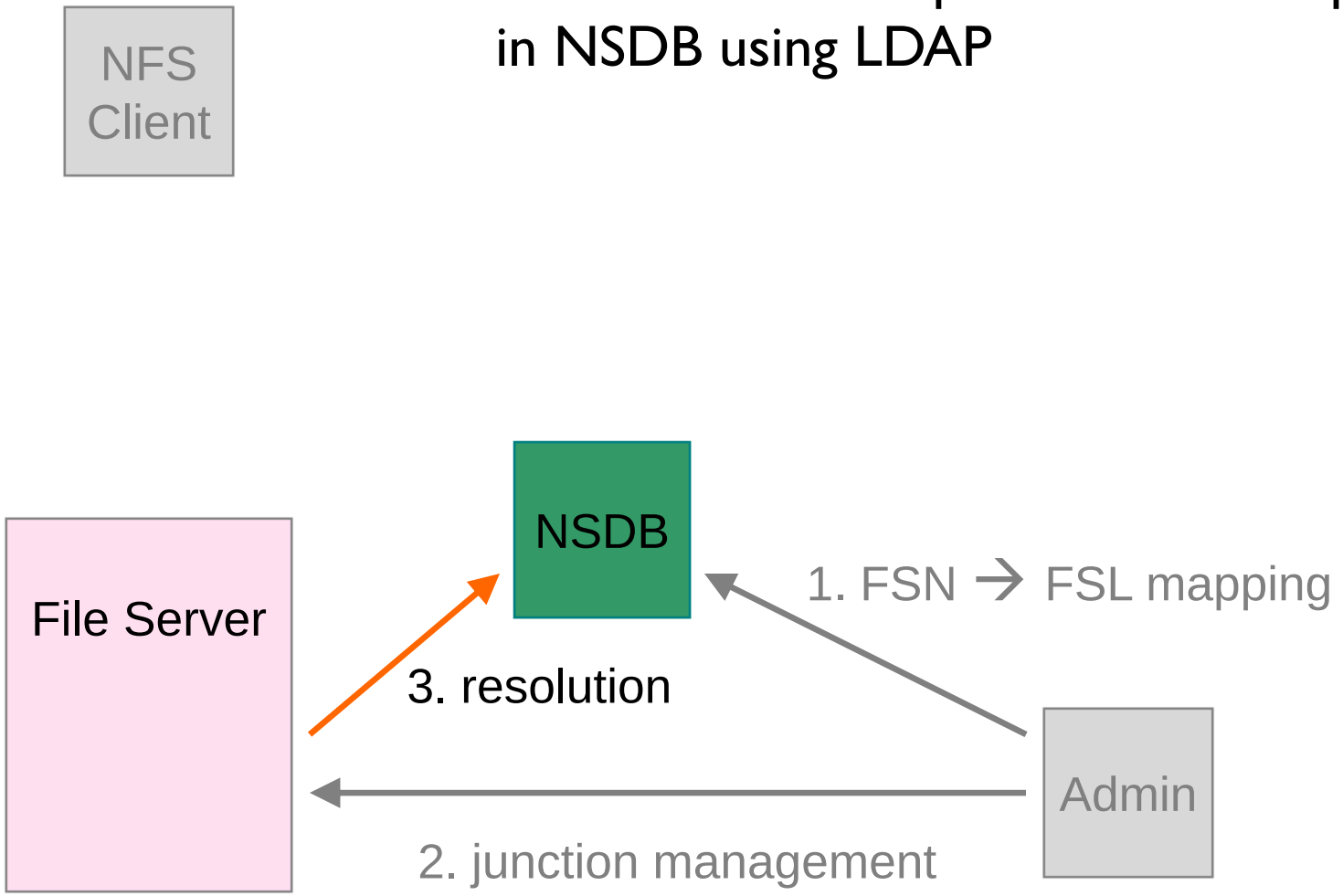
NFS Client

- Admin uses an ONC RPC protocol to
  - Create junctions
  - Delete junctions
  - Lookup FSNs



FSN Resolution

- File server looks up FSN to FSL mappings in NSDB using LDAP



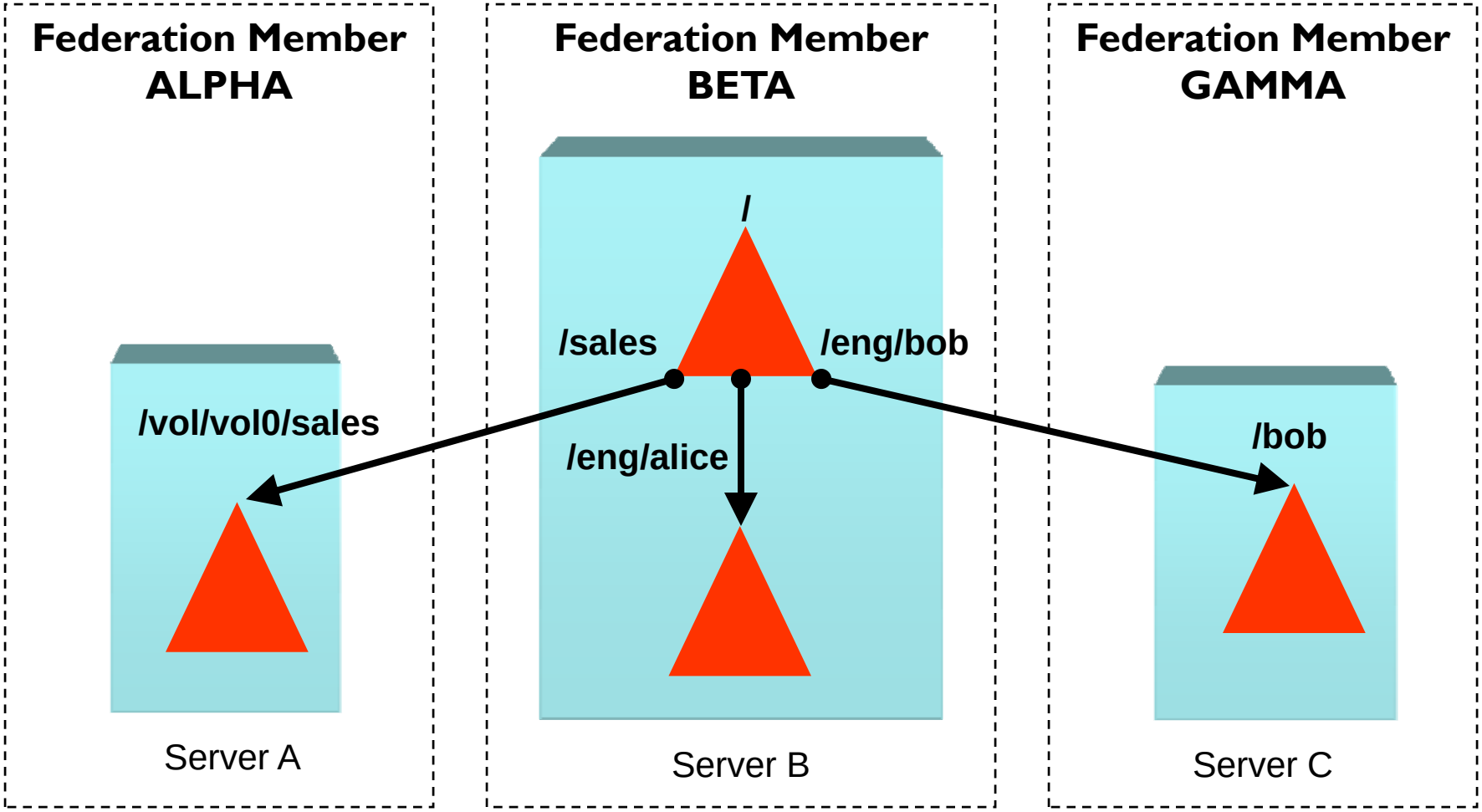
NFSv4 Referrals (1)

- Referrals are an NFSv4 feature that permit a server to redirect a client to another location, possibly on another server.
- The server refers a client to another location by returning an ERR_MOVED. The client can discover the object's new location via the information in the
  - fs_locations attribute (v4)
  - fs_locations_info attribute (v4.1)
- Referrals supported in Linux as of 2.6.20

NFSv4 Referrals (2)

- Client sends PUTROOTFH, LOOKUP (sales), GETFH
- Server sends ERR_MOVED error to client
- Client sends GETATTR (fs_locations)
- Server sends fs_locations attribute with
  - fs_root -- path on current server
  - one or more pairs of
    - Server -- target server
    - Rootpath -- path on target server
  - fs_locations_info attribute extends fs_locations with additional information on replicas
- Client mounts rootpath from target server

An example federated namespace



Referral Example (1)

- Client mounts server B:/
- User does “cd sales”

NFS server B

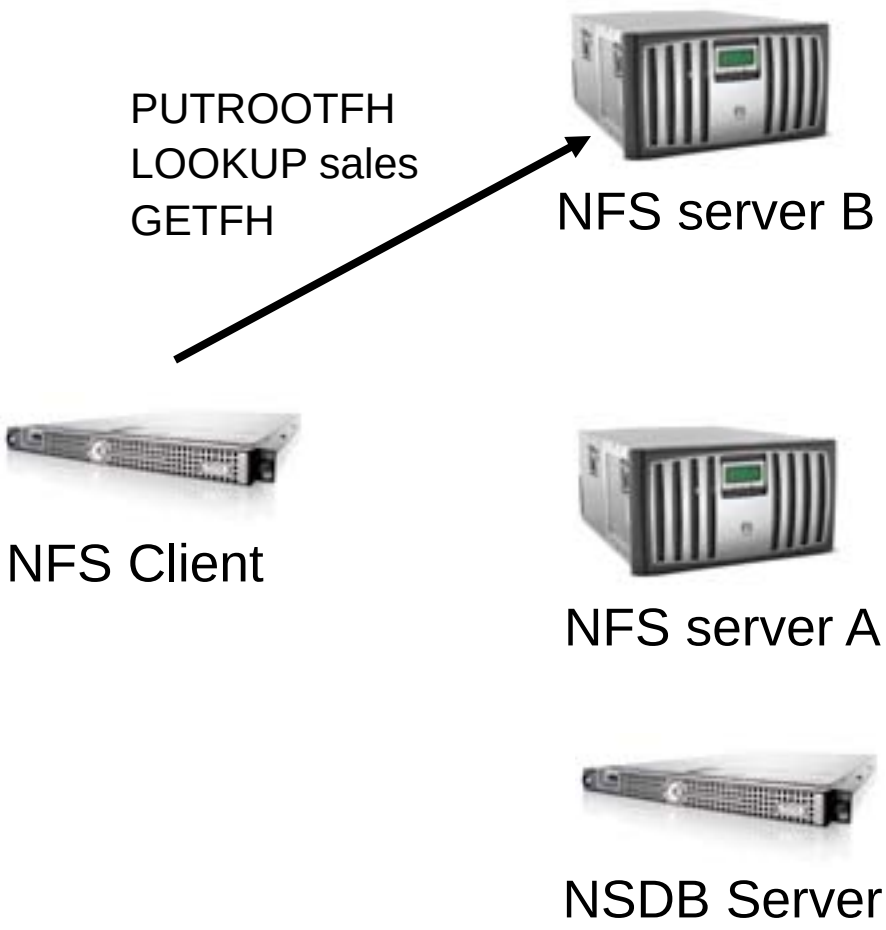
NFS Client

NFS server A

NSDB Server

Referral Example (2)

- Client mounts server B:/
- User does “cd sales”



Referral Example (3)

NFS server B

- Client mounts server B:/
- User does “cd sales”
- NFS server B determines sales is a junction

NFS Client

NFS server A

NSDB Server

Referral Example (4)

- Client mounts server B:/
- User does “cd sales”
- NFS server B determines sales is a junction
- NFS server B queries NSDB for FSL

NFS server B

LDAP query

NFS Client

NFS server A

NSDB Server

Referral Example (5)

- Client mounts server B:/
- User does “cd sales”
- NFS server B determines sales is a junction
- NFS server B queries NSDB for FSLs

NFS server B

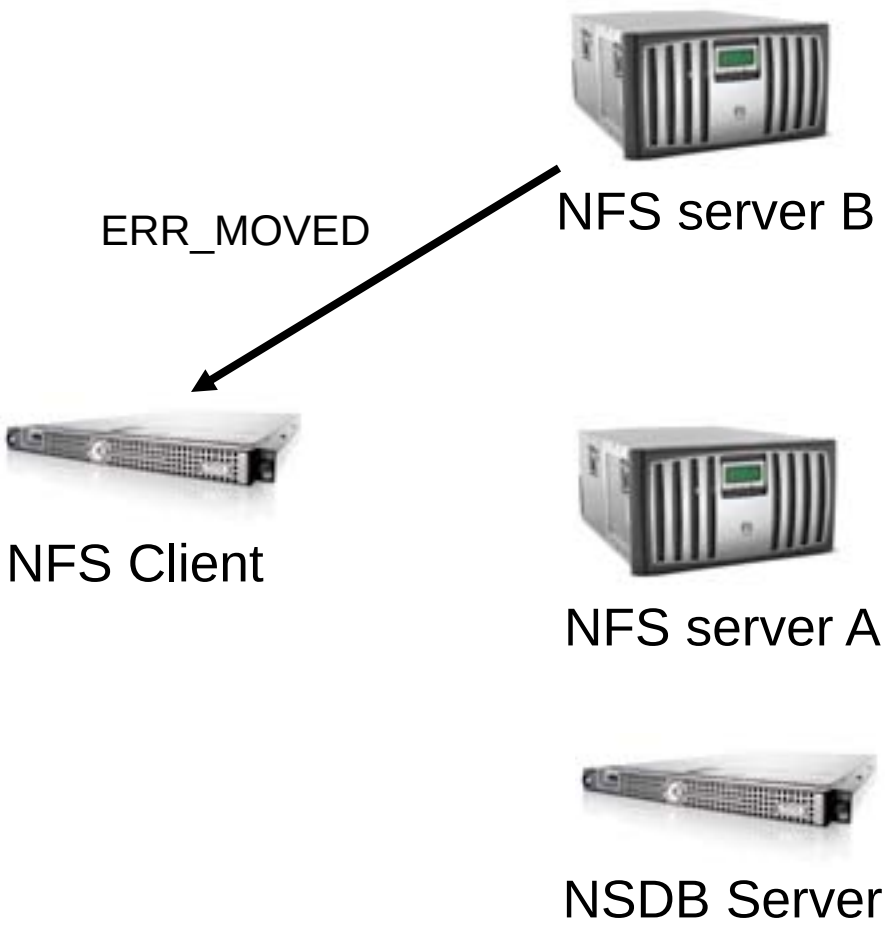
LDAP reply

NFS Client

NFS server A

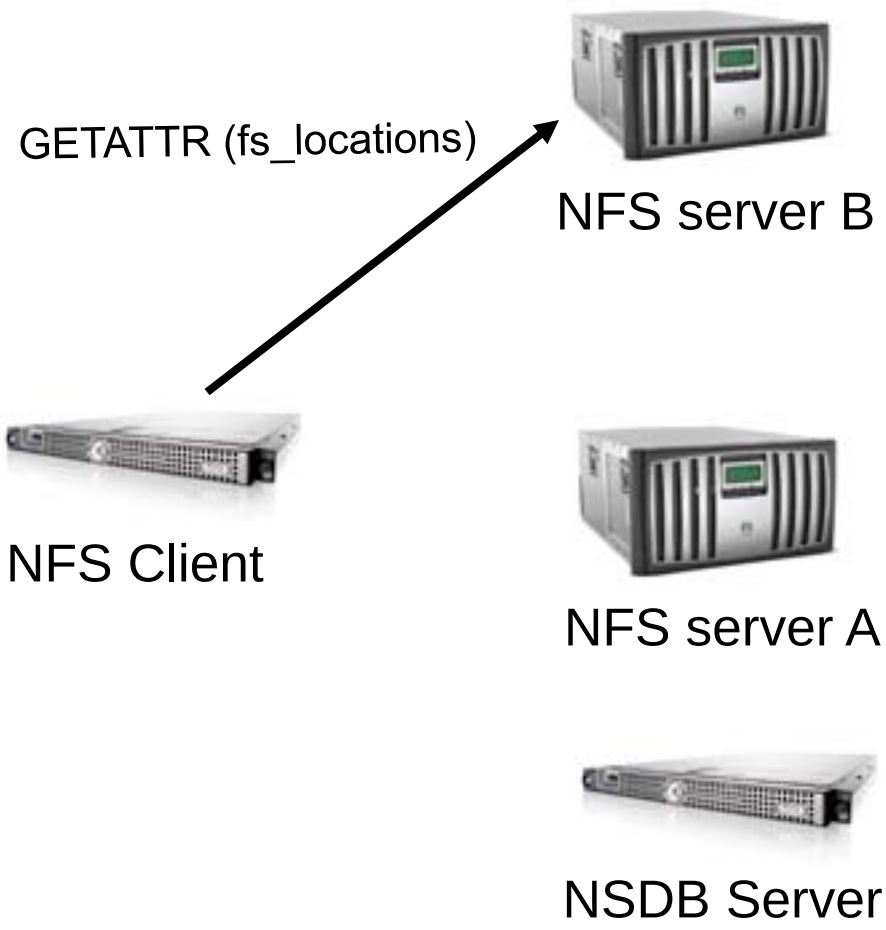
NSDB Server

Referral Example (6)



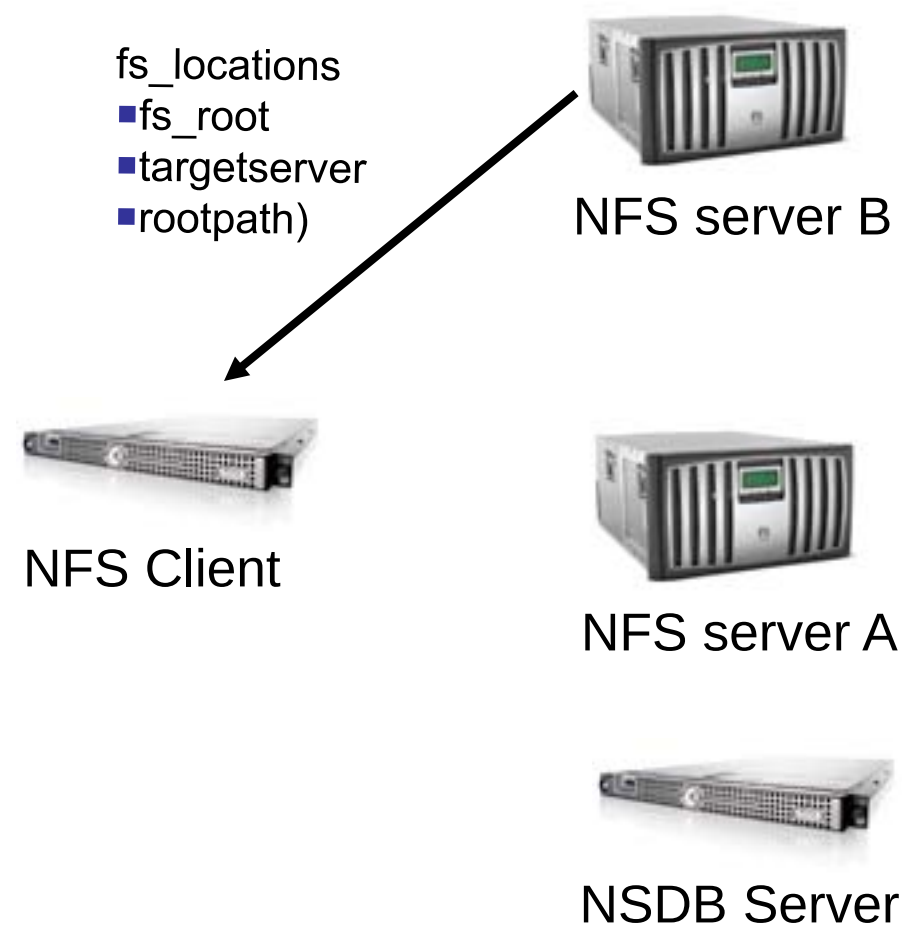
- Client mounts server B:/
- User does “cd sales”
- NFS server B determines sales is a junction
- NFS server B queries NSDB for FSLs
- NFS server B returns ERR_MOVED

Referral Example (7)



- Client mounts server B:/
- User does “cd sales”
- NFS server B determines sales is a junction
- NFS server B queries NSDB for FSLs
- NFS server B returns ERR_MOVED
- Client gets fs_locations

Referral Example (8)



- Client mounts server B:/
- User does “cd sales”
- NFS server B determines sales is a junction
- NFS server B queries NSDB for FSLs
- NFS server B returns ERR_MOVED
- Client gets fs_locations

Referral Example (9)

NFS server B

PUTROOTFH

NFS Client

NFS server A

NSDB Server

- Client mounts server B:/
- User does “cd sales”
- NFS server A determines sales is a junction
- NFS server B queries NSDB for FSLs
- NFS server B returns ERR_MOVED
- Client gets fs_locations
- Client mounts NFS server A

FedFs Standardization (1)

- Informal group with participants from several organizations
  - Weekly meetings
  - Open community list: federated-fs@sdsc.edu
- Open source NSDB
  - http://snsdb.sourceforge.net
- Four IETF drafts
  - Requirements for Federated File Systems

    https://datatracker.ietf.org/drafts/draft-ellard-nfsv4-federated-fs/
  - NSDB Protocol for Federated Filesystems

    https://datatracker.ietf.org/drafts/draft-tewari-nfsv4-federated-fs-protocol/
  - Admin Protocol for Federated Filesystems

    https://datatracker.ietf.org/drafts/draft-ellard-nfsv4-federated-fs-admin/
  - Using DNS SRV to Specify a Global File Name Space with NFS version 4

    https://datatracker.ietf.org/drafts/draft-everhart-nfsv4-namespace-via-dns-srv/

FedFs Standardization (2)

- Requirements agreed upon
- Common terms and definitions agreed upon
- Substantial progress on the protocol drafts
  - Three sub-protocols defined
  - Continuing discussion over additional protocols
- This work will be incorporated into the IETF NFSv4 working group
  - documents will be re-published as WG drafts
  - charter will be updated
- NetApp has a working prototype that was demonstrated at the IETF NFSv4 Working Group Meeting in Dublin

Outline

- Introduction and overview
  - Motivation, background, and goals
  - Requirements, terms, and definitions
- Architecture and implementation
  - Basic resolution protocol
  - NFSv4 details
  - State of the standardization effort
- Conclusion

Conclusions

- The FedFs project has made considerable progress toward an open standard for a global, federated namespace.
  - Standards drafts with community support
  - A proof-of-concept demonstration
- Next steps:
  - Formal standardization through IETF process
  - Leveraging the federated namespace for new features
    - replication
    - migration

Questions?

Backup: Discovery protocol?

How does a client find the root fileset?

1. Using DNS-SRV
2. Client asks a server
3. Client asks an NSDB