



Go further, faster®



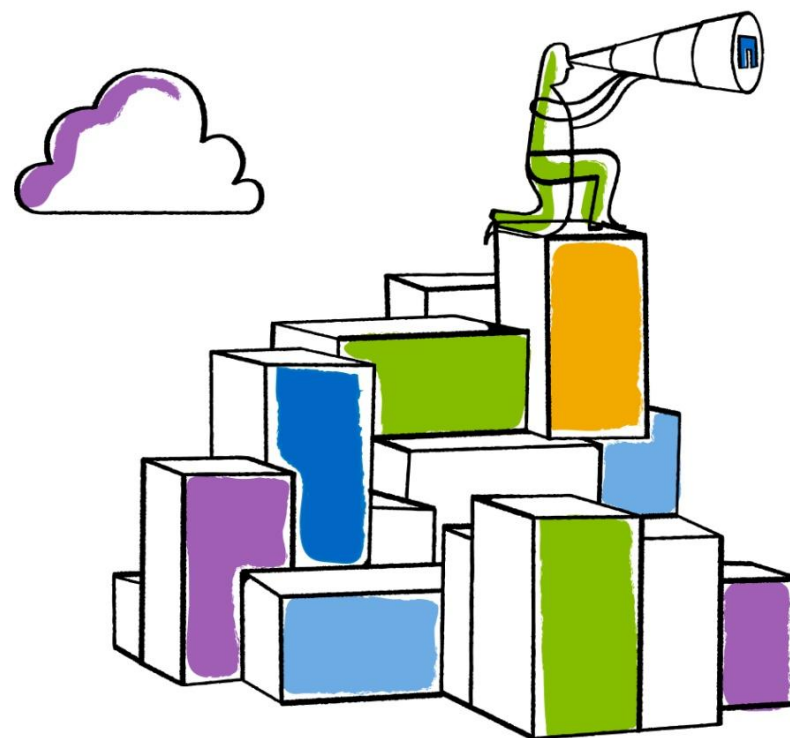
# Object Storage

## IETF 81

Mike Eisler

mre dash ietf at eisler dot com

July 25, 2011





# Overview

- draft-dipankar-nfsv4-pathless-objects-02
- Proposes new operations for accessing pathless objects
- Recent developments in the Linux kernel makes even more interesting



# New Stuff in Linux

- Captured in <http://lwn.net/Articles/375888/>

- Linux kernel now has two new system calls

```
int name_to_handle(const char *name, struct
    file_handle *handle);
int open_by_handle(struct file_handle
    *handle, int flags);
```

- These are supported on local file systems today (XFS)
  - Use cases include backup/restore/user level NFS servers
- It is reasonable to expect these to be supported on the NFS file system



# Why would filehandle system calls be useful for the NFS file system?

- Reduced overhead by reducing or eliminating need to access directories
- Industry experience with this model
  - Facebook’s Haystack experiences use this model (via internal modifications to Linux that are similar to the new system calls)
  - eiomail.com’s direct-to-nfs (<http://www.strangebizmo.com/articles/directnfs/>)



## What does this have to do with draft-dipankar-nfsv4-pathless-objects-02?

- There is really no difference between an Object Identifier and a filehandle
- Without these system calls, draft-dipankar-nfsv4-pathless-objects-02 would be embodied as a user-level NFS client
  - Competes with restful http object access protocols (e.g. CDMI, S3, etc.)
    - Not a real win
- With these system calls the built in NFS client can be leveraged to provide object access
  - Existing Linux system calls could be overloaded to provide other features of objects



# Proposal for Moving forward

- If/when Linux supports the system calls over NFS target draft-dipankar-nfsv4-pathless-objects to next minor version of NFSv4
  - NFSv4.2 is not proposed



*Thank you*



# Q&A

