



# AFS/OSD Project

R. Belloni, L. Giammarino,  
A. Maslennikov, G. Palumbo, H. Reuter,  
R. Toebbicke

# Contents

- Goals of the Project
- Solution and benefits
- SCSI T10 OSD Standard ■
- Components of the Client/Server Architecture
- Additional Elements
- Interactions between Components
- Client and Server Details
- Conclusions and Immediate Goals

# Goals of the Project

## Improve AFS performance and scalability

- Principal issues that have to be addressed:
  - Throughput of a single fileserver is too low
  - AFS volumes are limited to single disk/lun/filesystem
  - Large files cannot be striped over the multiple servers
  - No RW replication

# Solution and benefits

## Extend AFS to support object-based file management

- File Server will be less loaded
  - Client requests, security checks, data transfers and platform-dependent object-related meta-data will be handled by device servers
- Finer-grained performance tuning
  - Support for object mirroring (RAID-1) & striping (RAID-0)
- Better Scalability
  - Support for location-independent identification of objects

# SCSI T10 OSD Standard

- SCSI Object-based Storage Management Standard
  - **OBSD** = Object-based Storage Device supporting OSD Standard
  - Exposes stored data as user objects and partition objects with attributes
- Command Descriptor Block (CDB)
  - Fixed format for specifying commands to execute on OBSDs
- Attributes & Capabilities
  - Meta-data used to describe objects characteristics
  - Fixed format used by OBSDs for authorization checks
  - Capabilities can be signed for security reasons (credentials)
- Implementation Notice: OBSD access via Rx-protocol
  - Lightweight, connection-oriented, RPC-based protocol over UDP
  - Allows for multiple simultaneous connections (seen ~250k)

# Client/Server Architecture

## AFS Client Extended

- Extended for T10-OSD support
- Communicates directly with OBSDs
- Manages data read/write streams

## AFS File Server Extended

- Extended for T10-OSD support
- Tracks objects location
- Creates Capabilities

## OBSD

- Accepts CDBs & parses them
- Checks capabilities for authorization
- Executes commands & manages stored objects

# Additional Elements

## ➤ File Descriptor (FD)

- Exchanged between File Server and Clients
- Describes how files are split into segments composed of objects
- Describes segments and objects
- Stored where normally AFS file's data are stored

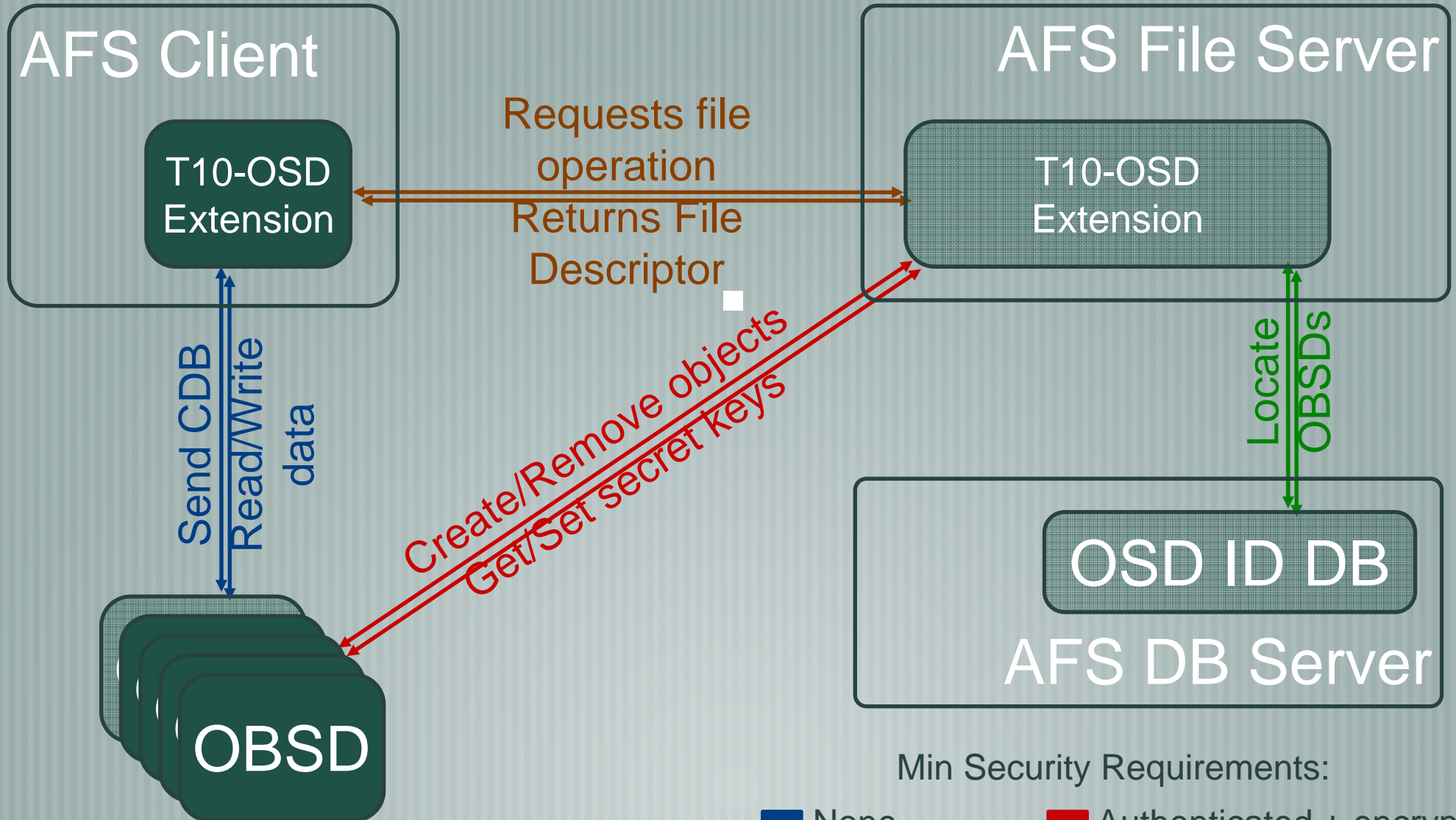
## ➤ OSD ID Replicated Databases

- Used by File Server to track OBSDs
- Use Ubik framework

## ➤ Security Framework

- Used by all components
- Makes checksums of capabilities using secret keys to create credentials
- Avoids credential forgery and capabilities alteration

# Interactions Between Components



Min Security Requirements:

- None
- Authenticated
- Authenticated + encrypted
- AFS Specific



# Client and Server Details

## ➤ AFS Client Extended - Details

- Analyzes the status info of the requested file to determine if it uses OSD
- Receives/analyzes the FD and contacts OBSDs to read/write objects
- Takes advantage of mirroring and striping techniques

## ➤ AFS File Server Extended - Details

- Analyzes a file's data and meta-data to determine if it uses OSD
- Interacts with OSD ID Databases and OBSDs to create new objects based on a policy description file and using a load-balancing algorithm
- Creates capabilities, generates credentials and returns the FD to the client

## ➤ Important Open Issue

- T10-OSD is not rich enough to support AFS volume replication and link counts

# OBSD Server Details

RX-protocol API

*Multi Threaded*

## Network Layer

*Interface to RX-calls for receiving CDB commands*

OSD-CDB API

*T10-OSD  
Compliant*

## OSD Command Interpreter

*Interpreter of CDB commands and security enforcer*

Logical Unit  
API

*Thread Safe*

## Logical Unit Layer

*Manager of attributes, partitions and user objects*

File Storage  
API

## File Storage Layer

*Interface to underlying storage technology (Linux FS, Namei...)*

# Conclusions & Immediate Goals

## Still a prototype, but with a working base

- Provide a basic stable high performance version in August (version 1.0) ■
- Enhance & add features
  - Support for Mirroring & Striping
  - Support for other protocols/technologies
  - Support for AFS volume replication and link counts

# References

- AFS/OSD twiki access - currently available upon request  
Public access by August
- Andrew File System - <http://www.openafs.org>
- SCSI T10 Technical Committee - <http://www.t10.org>
- RX protocol - <http://web.mit.edu/kolya/afs/rx/rx-spec>

Thank You for your Attention!