

Developing UID Mapping and a Stand Alone Security Mechanism for Lustre Challenges and Successes

Joshua Walgenbach

September 16, 2013



INDIANA UNIVERSITY



UID/GID Mapping and Shared Keys

Simple interface for static mapping of client (NID, [UID/GID]) pairs to a canonical set on the file system.

Add a shared key GSSAPI mechanism to authenticate and protect the data stream.



UID/GID Mapping Project Overview

Divided into three parts

- UID/GID Management Module (nodemap)
- UID/GID Mapping in the MDT/OST
- Map Synchronization via LNET from the MGS

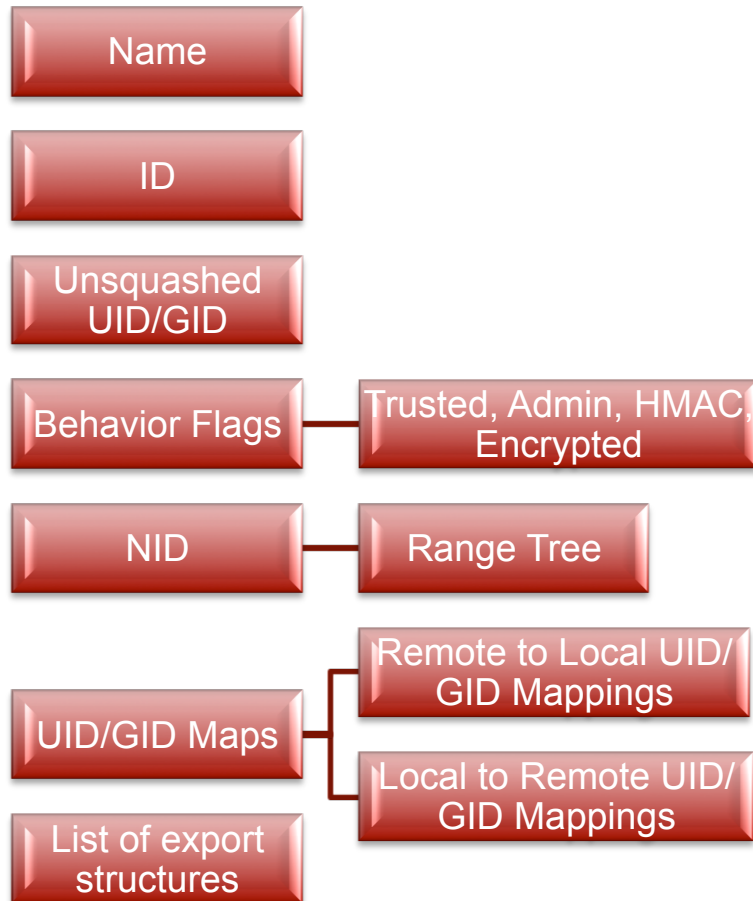


Nodemap Features

- Nodemaps defined as NID ranges and IDMaps
- Flags determine how ID Mappings are respected
 - Allow UID 0 to be unsquashed
 - Trust incoming UID/GIDs
 - Require HMAC/Encryption

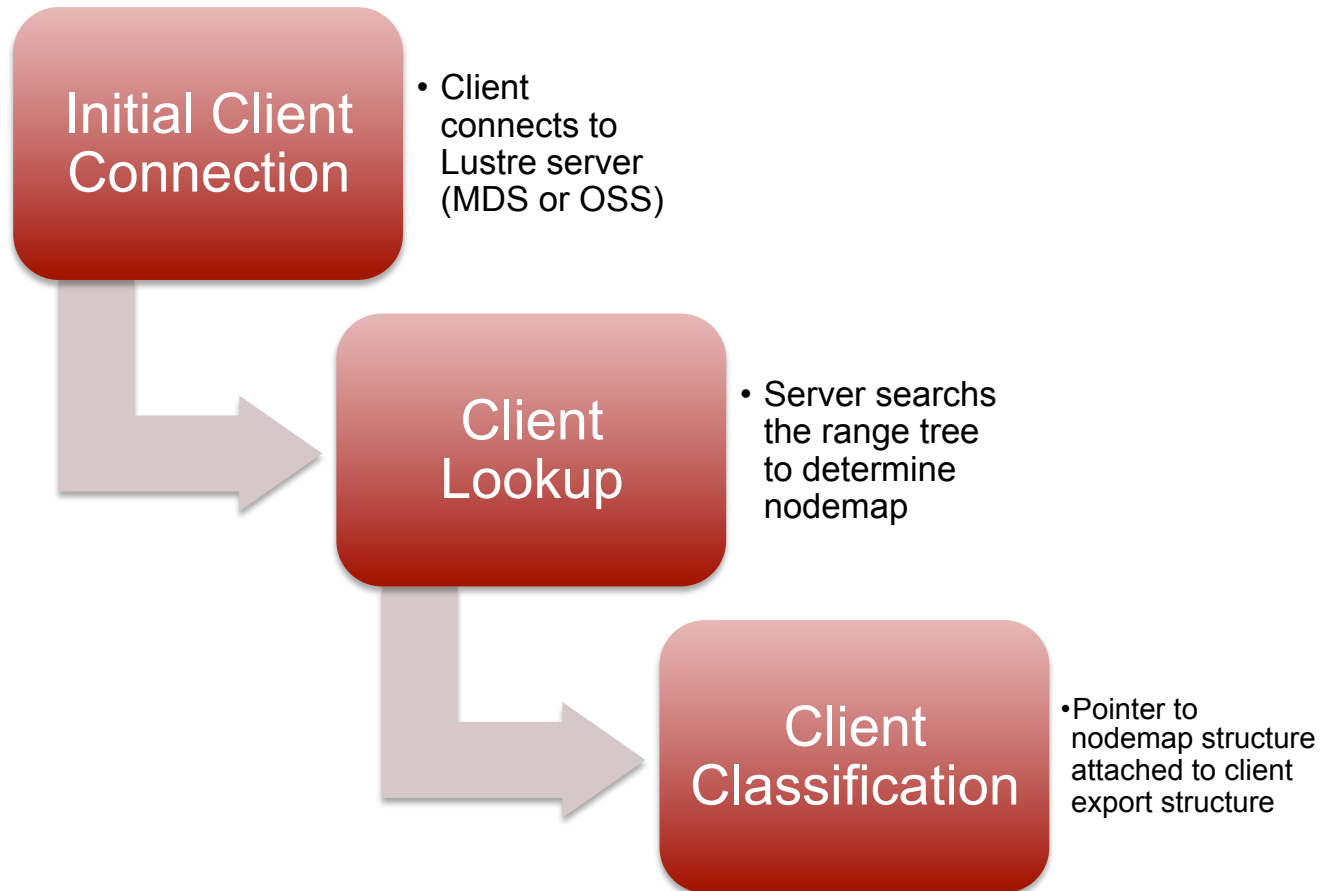


Nodemap Layout



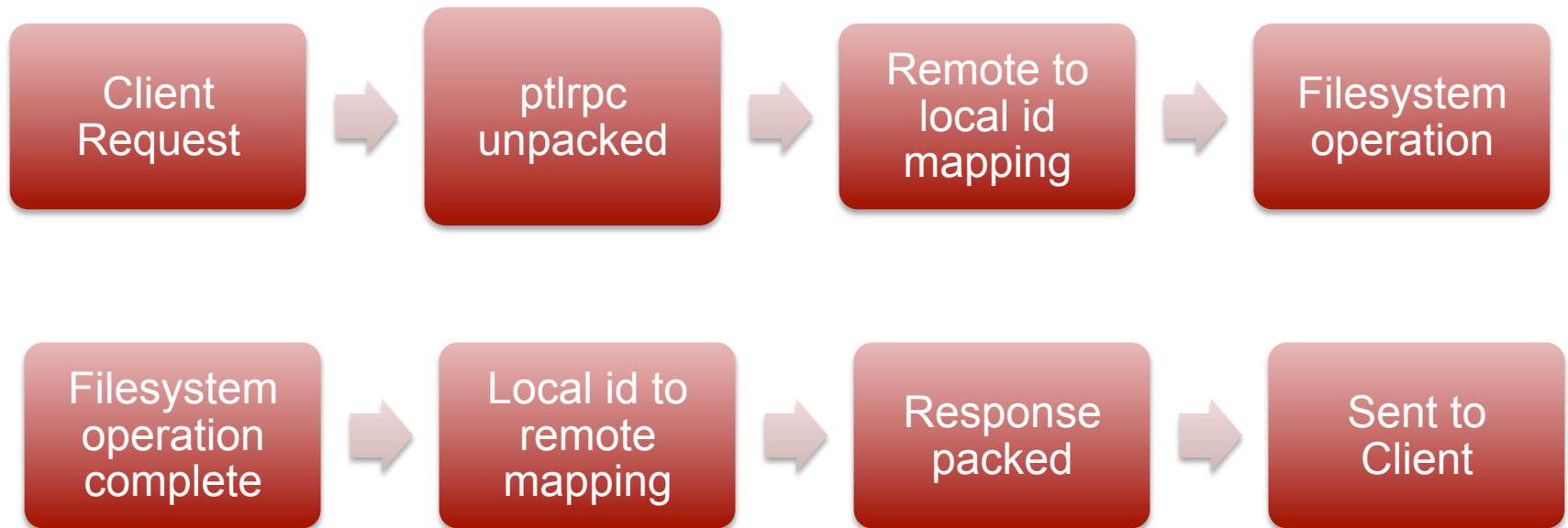


Client connection with nodemap



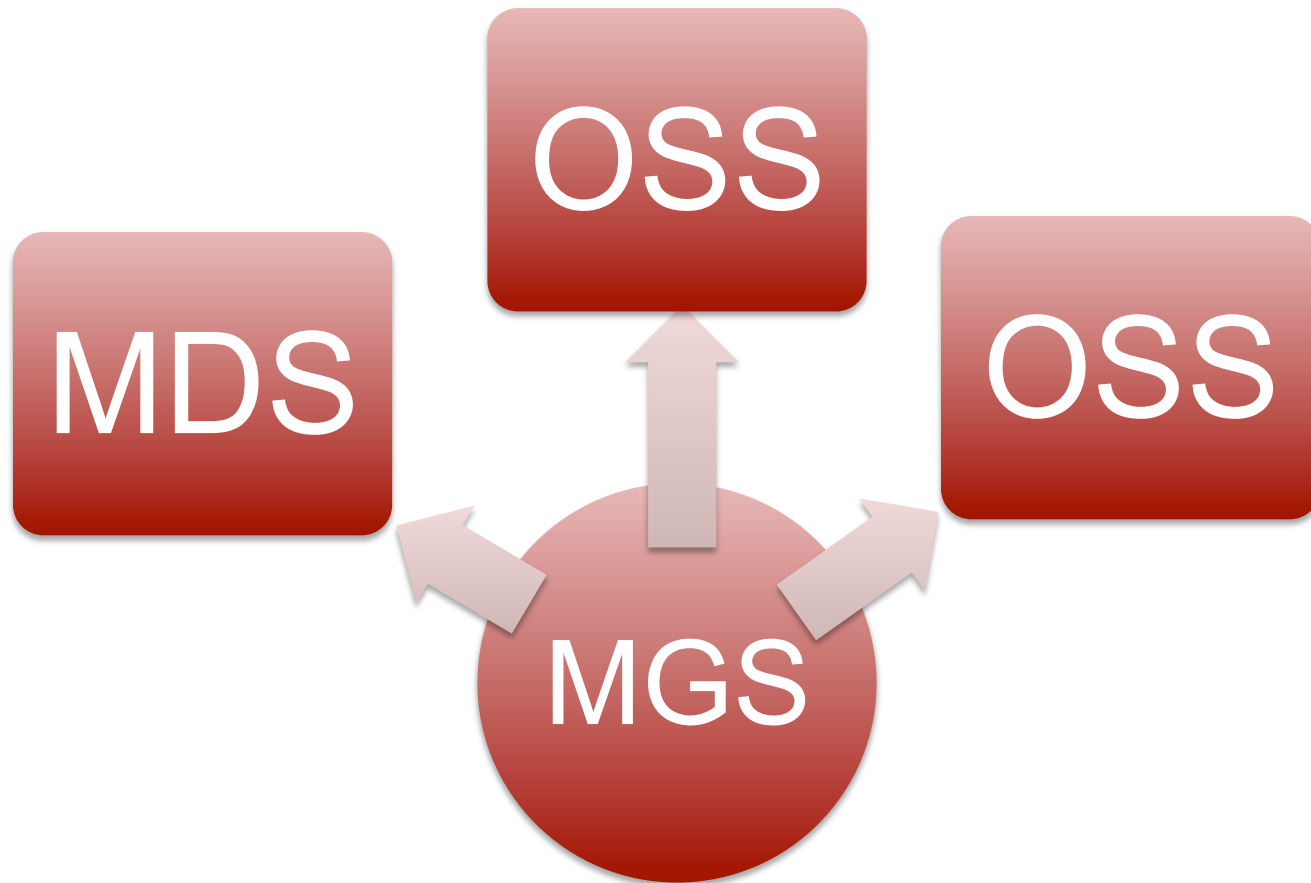


Filesystem Request Lifecycle





Nodemap Distribution





IU DevPAC

- Development Project Approval Committee

Volunteers from OpenSFS members to review the project and approve completion of milestones.



Challenges

- Manageable commits
- Code style is more than what is written down
- Expect the unexpected



Challenges

- Plowing a new field
 - Design documents
 - Where do you branch?
 - lustre-release and lustre-dev
 - Where do you commit?
- Tired reviewers (Thank you)



Successes

- Scope Statement, Solution Architecture, High Level Architecture approved by OpenSFS
- The UID/GID management module has been committed after a first set of code reviews.
- Work has begun on mapping the the MDT and OST
- Tireless reviewers (Thank you!)



Thank you

- The OpenSFS Board for supporting this work
- The IU DevPAC for all their work in helping guide us to completion and inclusion in the release tree
- Reviewers