

GSSIAM: Identity-Managed Lustre Mounts

Sebastien Buisson

Principal Engineer – Lustre Architect – DDN

Di Wang

Principal Engineer – Google



GSSIAM: Identity-Managed Lustre Mounts

- Context: cloud multi-tenancy, zero trust
- GSSIAM: the hybrid RPC model
- Handshake and mount flow
- Enforcement: dynamic nodemap, ProjID fence
- Interfaces, status

Cloud Identity and Access Management Model: Zero-Trust

- Identity-based access
 - Node+service identification
- Contextual-based decision
 - For every request, assessment of identity/resource/context
- Access control
 - File system access based on directory-level attributes for finer grain.
 - Policies defined at directory level.

Bridging IAM With High Performance File System

- High performance requirement
 - Not met by other strong authentication mechanisms
 - Kerberos
 - Shared Secret Key
 - krb5p/skpi flavors imply encryption of metadata and bulk data
 - Even krb5n/skn flavors involve GSS wrapping & signing overhead for every request.
- Need:
 - Authenticate once.
 - Then metadata/data RPCs with no extra cost.

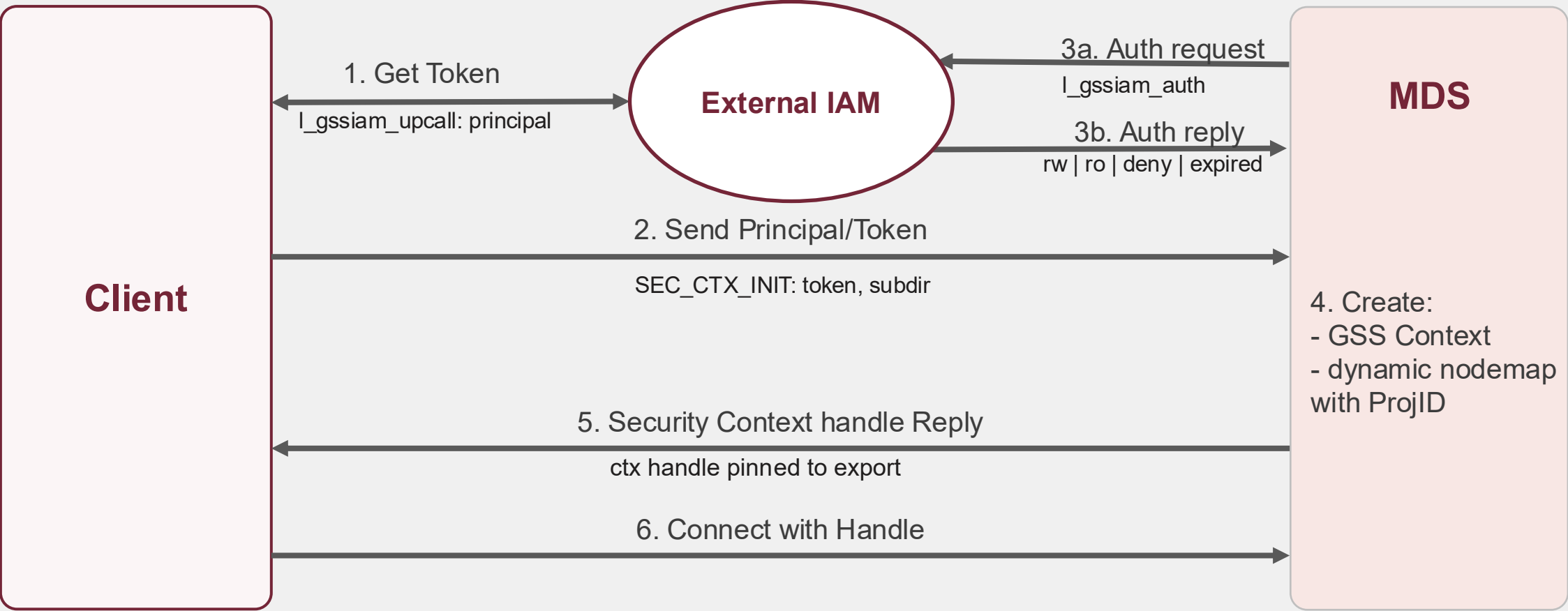
Bridging IAM With High Performance File System – cont'

- Nodemap trusts the client node
 - NID as identity
 - containers / K8s: local root untrusted
 - one tenant per fileset, not one user
 - Static mapping, admin-maintained
 - client-to-tenant churn in the cloud
 - Cloud identities: external, short-lived
- Need:
 - User level authorization
 - Flexibility in permission definitions

GSSIAM Introduces Hybrid RPC Model

- GSS framing: SEC_CTX_INIT only
 - Authentication to fetch IAM token
- *null* framing: all data and metadata RPCs
 - No GSS wrapping/signing
- Identity pinned to the server export
 - no per-RPC crypto, no per-RPC lookup

GSSIAM Handshake Process



Then: null framing — no GSS wrapping, no per-RPC crypto

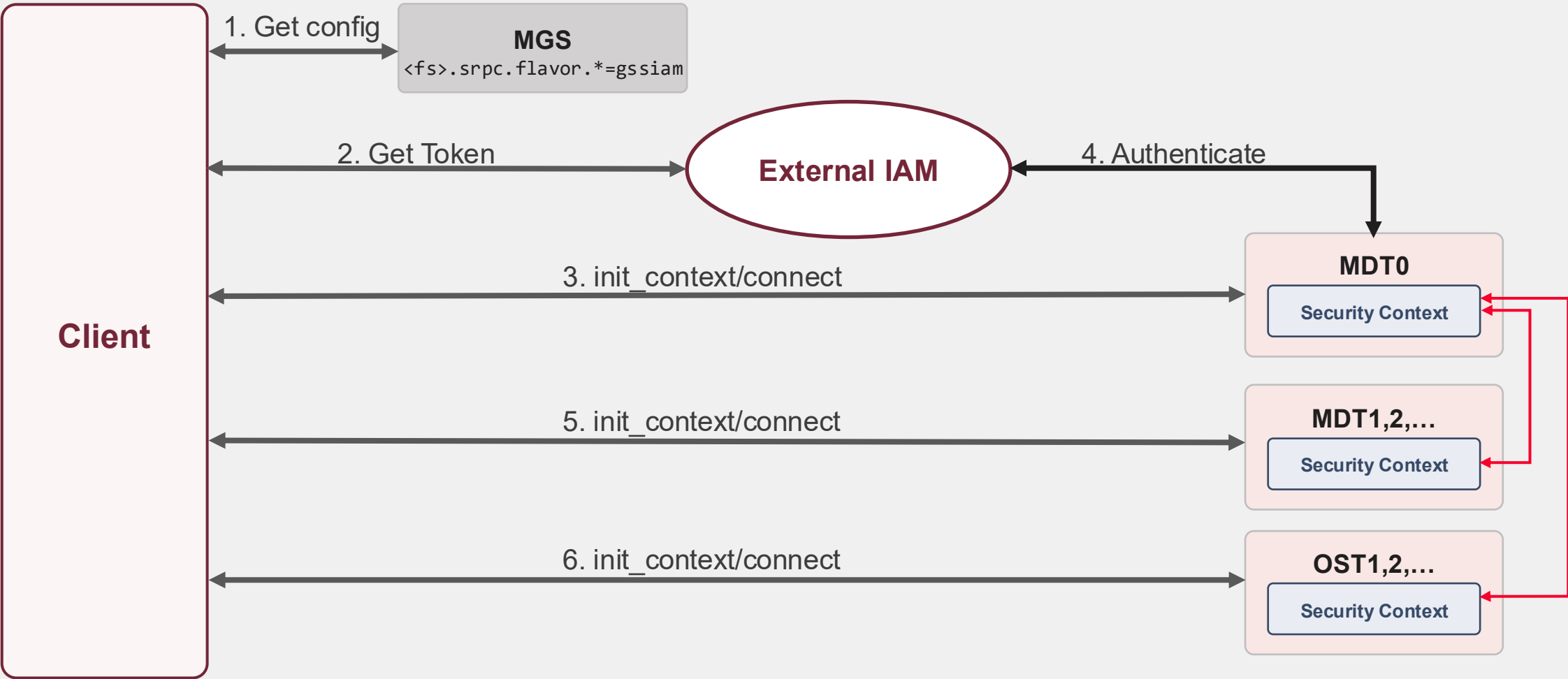
Communications With IAM In Userspace

- Helper paths are writable tunables
 - `sptlrpc.gssiam.gssiam_upcall`
 - `sptlrpc.gssiam_server.gssiam_auth_upcall`
- Downcall format
 - `<key> <mount_option>:<expire>:<approved options>:<principal>`
- In-tree helpers: mock references, to be replaced at your convenience

Dynamic Per-Identity Nodemap

- Child of the parent *gssiam* nodemap
- *fileset* → subdirectory confinement
- *ro/rw* enforced from the IAM verdict
- *projid_set* RBAC role withdrawn
- Project ID to enforce server-side check
 - $\text{user_projid} \neq \text{file_projid} \Rightarrow \text{-EHRNG}$

Mount Process Flow



MDT0-Leader Model

- Remote targets do not call IAM themselves
- Only MDT0 talks to the IAM server
- Other MDTs and OSTs forward verification to MDT0
 - `MDS_GET_INFO` → MDT0
- Bounded load on the external IAM
 - One authentication per identity — not per target, not per RPC
- Asynchronous Policy Refresh
 - Background heartbeat on MDT0
 - `sptlrpc.gssiam_server.gssiam_refresh_interval`
 - Lazy consistency, no per-RPC policy check

GSSIAM Client-Side Interface

- New mount option to specify IAM principal
 - `-o user_principal=<principal>`
- New mount option to disable root squash
 - `-o disable_rootsquash`
 - Only if IAM allows
 - Server root squash enforced by default

GSSIAM Can Work With Any IAM

- Lustre never reads the token.
- Opaque buffer, no parsing
 - `GSSIAM_MAX_TOKEN_LEN = 4096`
 - base64url-encoded into helper argv
- Any bearer-style credential fits
 - OAuth2 / OIDC, cloud instance identity, Vault
- No provider-specific code in Lustre

Status and Community

- Design document
 - <https://wiki.whamcloud.com/display/PUB/Lustre+mount+IAM+integration+HLD>
- LU-19921: patch series in review on master
 - authored by Di Wang, Google
 - 10 patches landed so far, 6 in-flight
- Targeted for the 2.18 timeframe
- Bring your own IAM — feedback wanted



Thank you!

Questions & discussion

ddn.com

