

StoRM : a Manager for Storage Resource in Grid

Introduction

A Storage resource can be composed by different storage systems: disk only systems, tape archiving systems, or by a combination of both. The basic logical entities of a storage resource are space and file. Space must be allocated when a new file have to be stored into a storage resource, and files could be dynamically removed to create the necessary space. This is the main goal of Storage Resource Managers (SRMs), middleware services whose function is to provide dynamic space allocation and file management of shared storage components. SRMs services agree on a standard interface to hide storage dependent characteristics and to allow interoperability between different storage systems. Nowadays, Storage Area Network (SAN) solutions are commonly deployed at LHC centres, and parallel file systems such as GPFS and Lustre allow for reliable, high-speed native POSIX I/O operations.

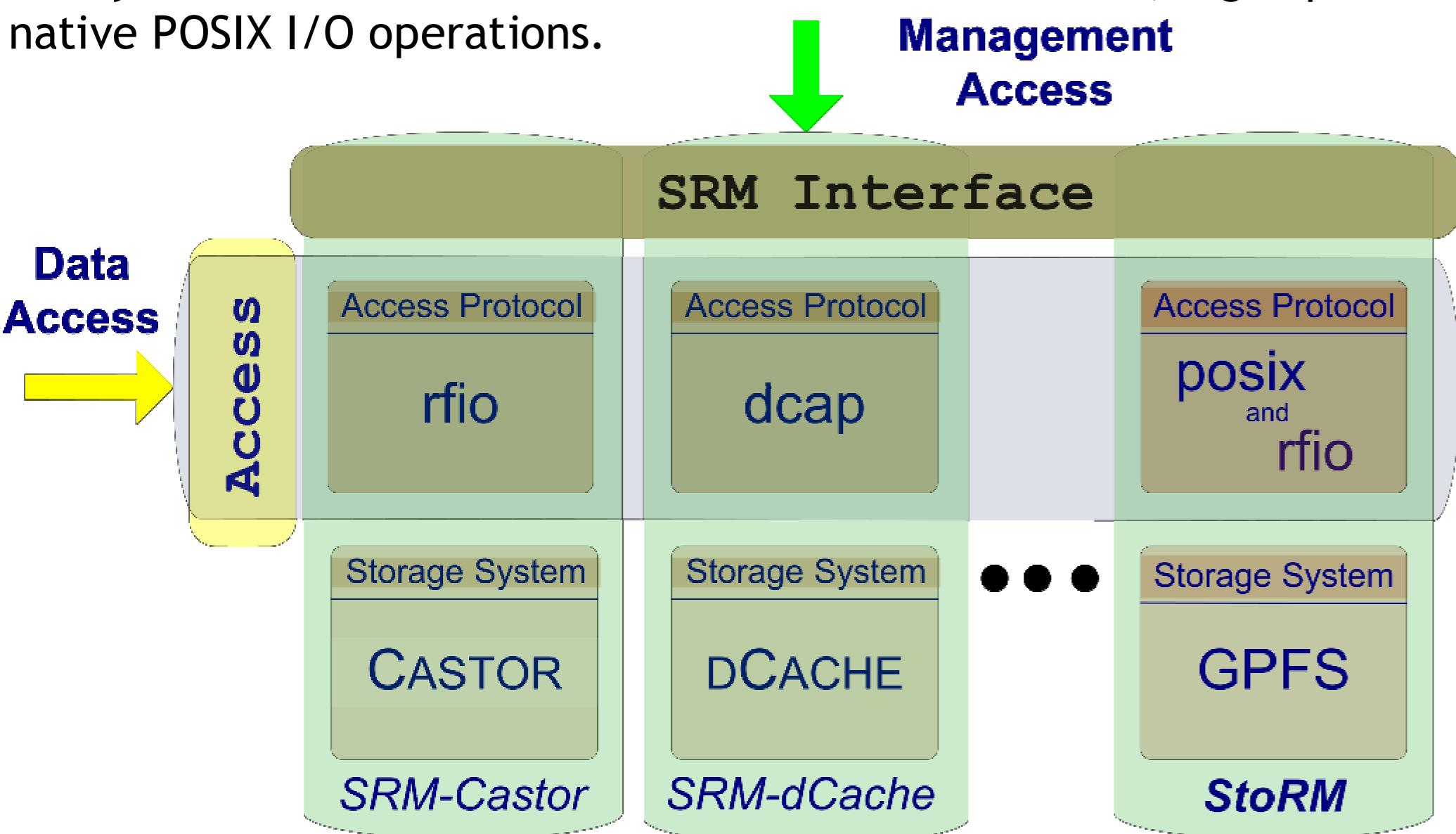


Fig. 1 : SRM interface allows a standard management of heterogeneous resources

StoRM (STorage Resource Manager) is a flexible and high-performing implementation of Storage Resource Manager (SRM) standard interface version 2.2. It is designed to be easily adapted to different underlying storage system via a plug-in mechanism, and, in particular, it leverage on the advantages of high performance storage systems based on cluster file system such as GPFS file system from IBM and Lustre from Sun Microsystems. StoRM is designed to support guaranteed space reservation and direct access (native POSIX I/O call), as well as other standard Grid access protocols.

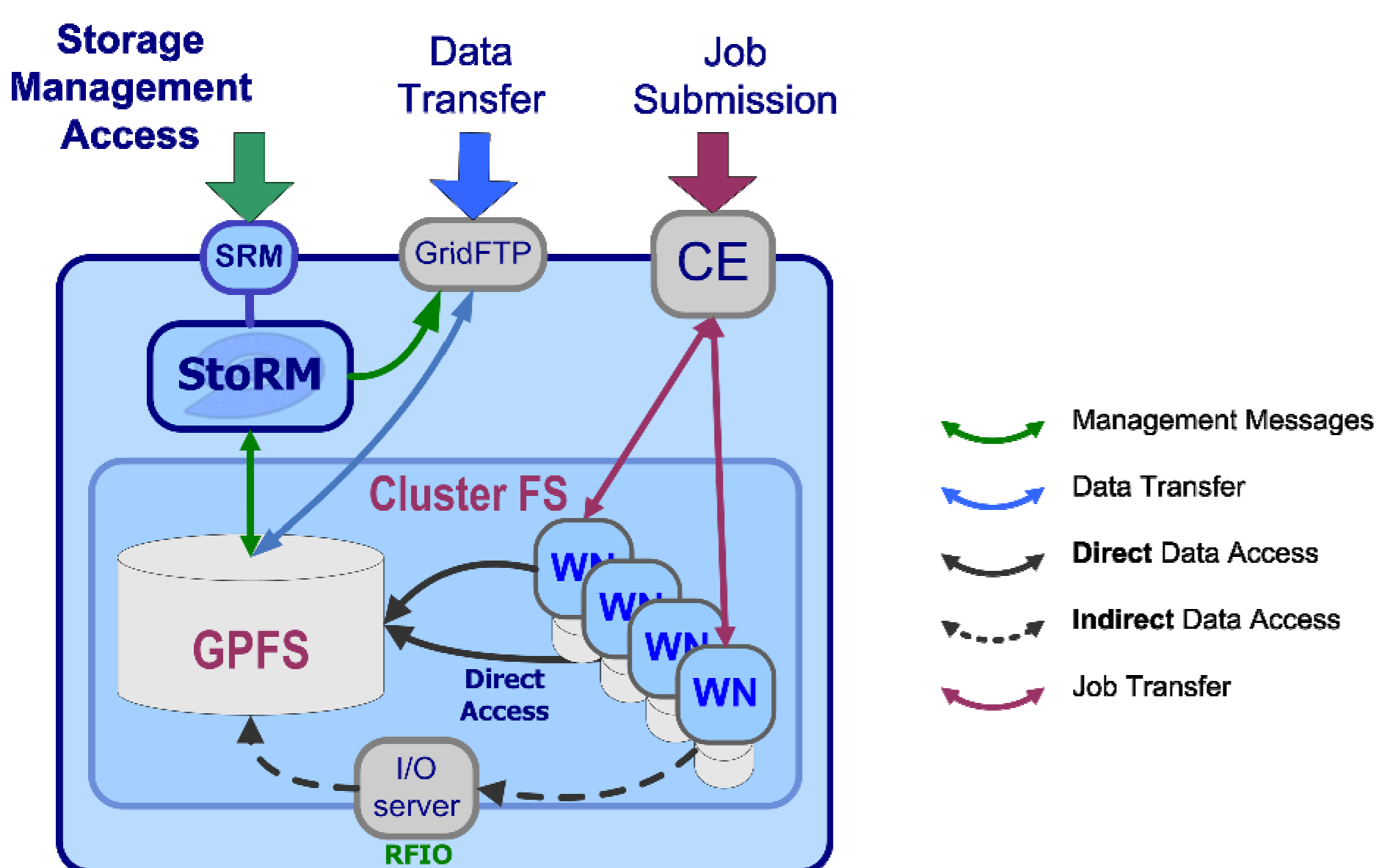


Fig. 2 : StoRM in a site with GPFS file system allows direct access (file protocol)

Architecture

StoRM has a multi-tier architecture. There are two main levels named Front-End (FE) and Back-End (BE).

The FE exposes the service interface. It receives the requests, manages the client communications, and forwards a processed structure to the server.

The BE is the core of StoRM. It processes the requests, executes necessary tasks to perform operations like security, metadata, space and file management and returns the result to the client.

The wrappers to different file system types (at the moment GPFS from IBM, the standard Posix FS and XFS from SGI) provides a common interface, hiding file system peculiarities. StoRM takes also advantage of special features provided by the underlying file system, like ACL support and file system block pre-allocation (feature available with GPFS and XFS).

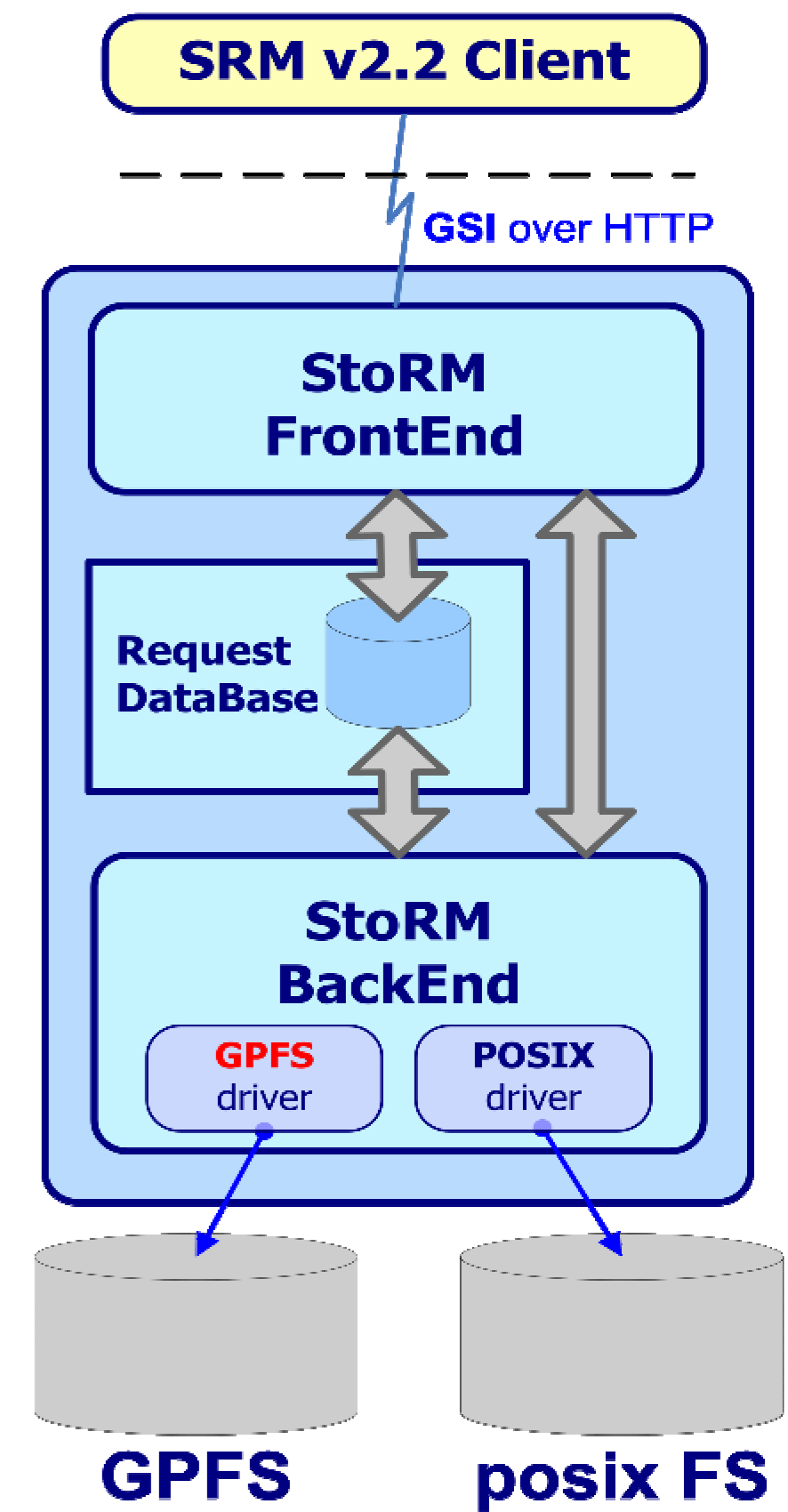


Fig. 3 : Multi-tier architecture of StoRM

Usage scenario

- 1) The user requests *SrmPrepareToGet* to a SURL to obtain a file pinned and ready to be accessed.
- 2) StoRM then queries the Authorization Sources to verify if the user can perform the specified operation on the SURL
- 3) StoRM queries LCMAPS to obtain local user account corresponding to the grid identity of the requester
- 4) The file system wrapper enforces permissions by setting a new ACL on the physical file

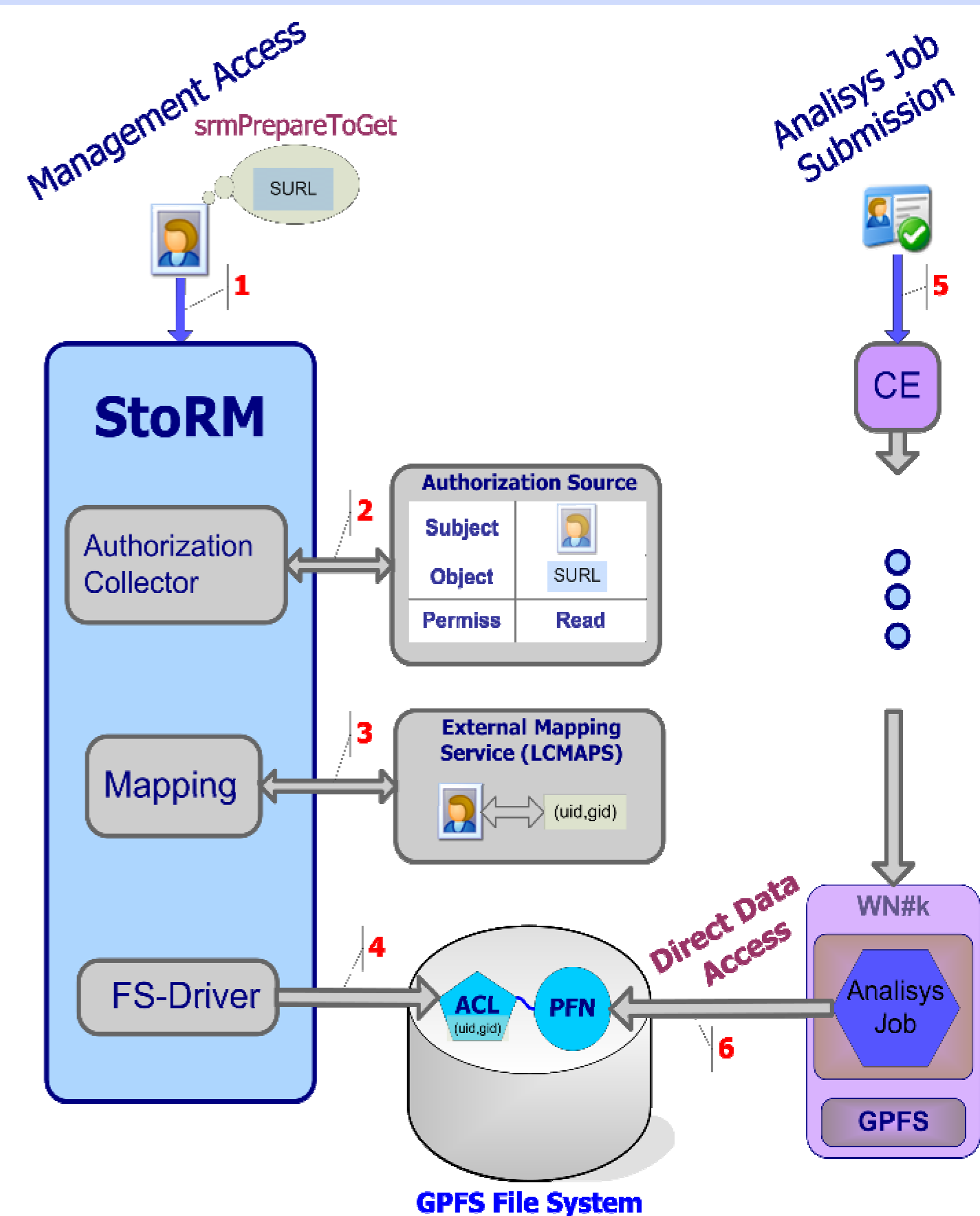


Fig. 4 : A simple usage scenario for accessing a file.

- 5) The user Job can be executed into the worker node.
- 6) The application can perform a standard POSIX call to access the file into/from the storage system