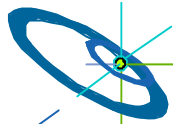


FTM

Global File Transfer and Management in a Network

Jim Almond, Peggy Lindner

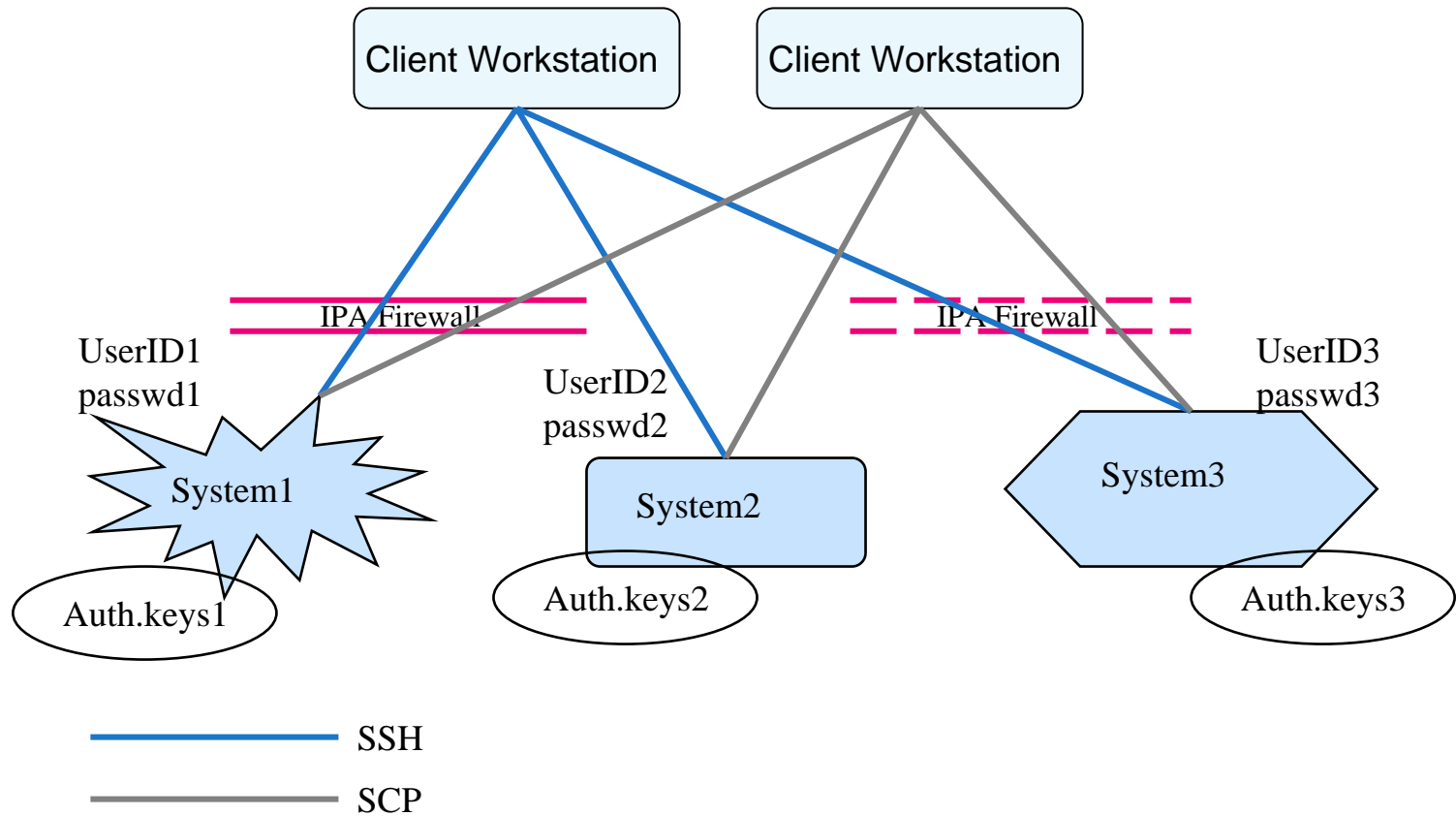
June 22, 2004



Almond/Lindner
Höchstleistungsrechenzentrum Stuttgart

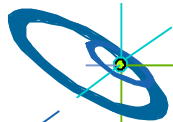
H L R I S 

Current Problems with Network Access



Overview, Goals, and Motivation

- Uniform and Secure Global Access to Distributed Unix Storage Servers
- Uniform, seamless User Interface for Unix and Windows clients
- Global user identity and authentication based on PKI Infrastructure
- Single integrated interface for *management, access, and execution* of files
- Access to distributed files by complex job workflows for High Performance Computing
- Use open, modular, Object oriented tools (java, SSL)
- Adaptable as a Grid element



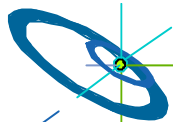
Functional Overview

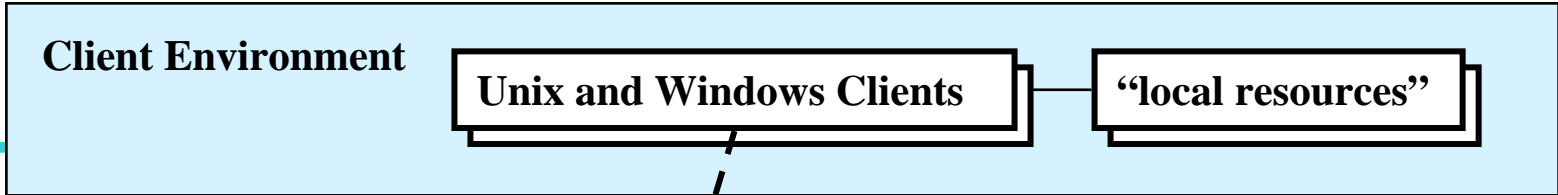
- Integrates file *access, management, and execution* in a single system (can replace ssh, scp, and ftp)
- Functional semantics are like Unix for file and directory resources:
 - Supports multiple targets, wildcards, global navigation
 - File management: list, make, remove, find, cat, rename - - -
 - File Access: Get to client, Put from client, third-party Transfers
 - Execution of files
- Enhancements
 - Control of overwrite and backup of existing resources
 - Optional compression and encryption during transfer
 - Third-party transfer among StorageServers & organizations
 - Transfer of multiple resources to .ZIP file
 - Optional limits on transfers



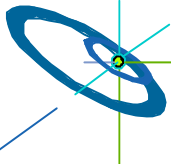
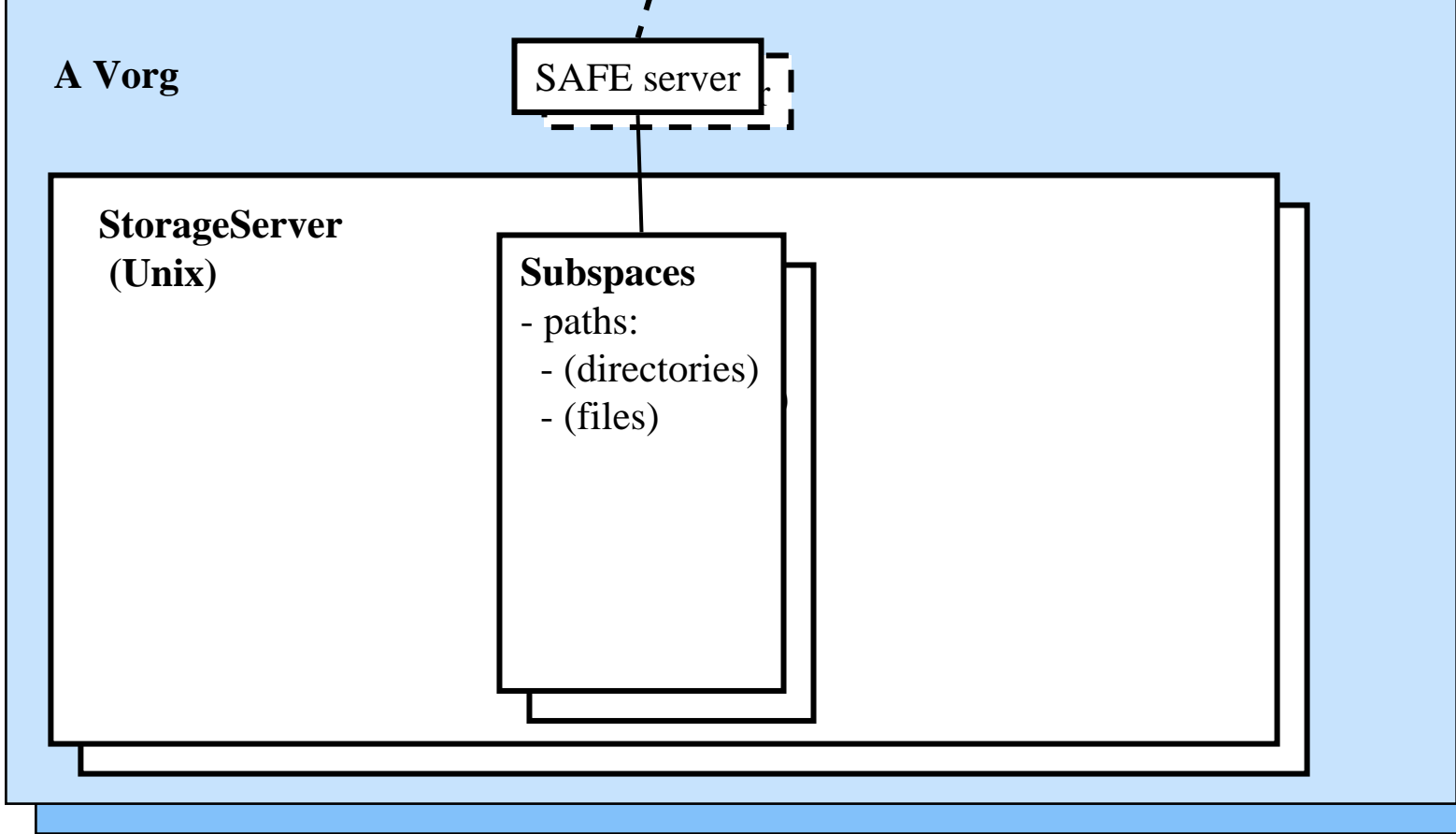
Global Terms for Designating Files and Directories:

- *Vorg*:
 - A Virtual Organization containing resources available to the user based on a registration (user ID, password)
 - Access within Vorg handled by one or more “*Safe*” (Secure Access File Executive) servers
 - May contain multiple StorageServers
- *StorageServer*: a storage platform within a Vorg
- *Subspace*: a storage area on a StorageServer analogous to directories or folders
 - Seamless: such as “HOME” or “TMP”
 - Path such as */users/myhome*
- *Pathname*: path (analogous to unix or windows) either *absolute* or *relative* to Subspace



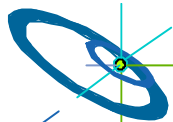


Optional firewall

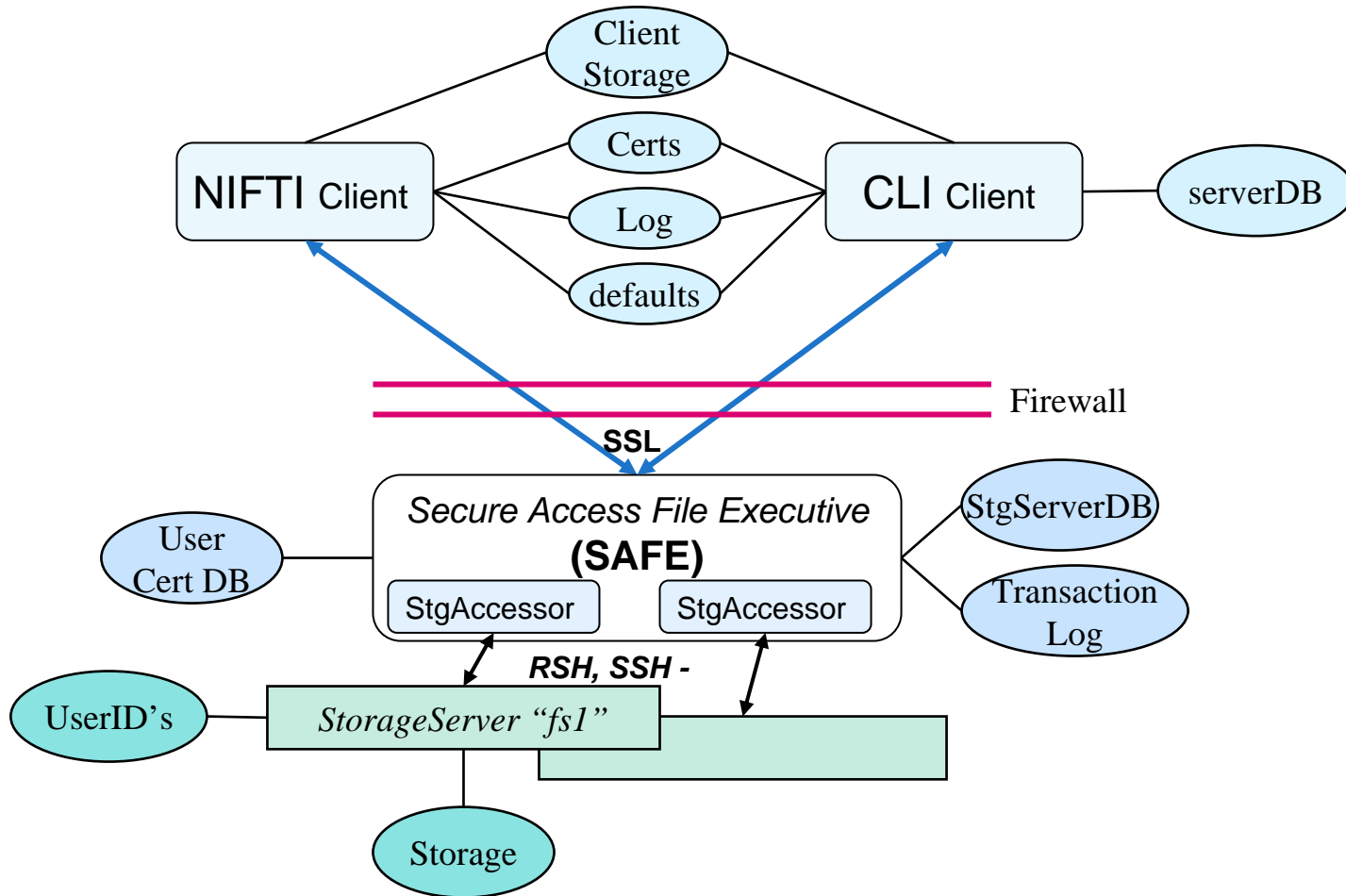


Security Issues

- Secure Access File Executive (SAFE) server controls access based on user's certificate;
 - Allows access through (or replaces) firewall.
- X509 certificate as global user identity
- Client authenticates server to prevent spoofing
- Connections to resources in a Vorg uses SSL
 - Internal connections can use rsh, ssh, etc.
- All requests and protocol traffic is encrypted
 - data traffic is *optionally* encrypted



Architectural Overview



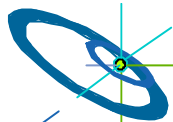
The Command Line Interface – for Unix clients

- *Single Command* mode makes connection for each command (useful for testing and for scripts)
 - `unix> Fpwd`
`@134.56.14.44:fs1/HOME`
 - `unix> Fcd @:fs2/indat`
change to Subspace indat on StorageServer fs2 at same Vorg
 - `unix> Fls -v :foo?`
`-rw-r--r-- 1 myID myGroup 17440 Dec 15 22:43 foo9`
 - `unix> Fget -ow 2 :foo9 :adir lclDir`
Copies foo9 and adir to lclDir on client (overwrite not allowed)
 - `unix> Frm -v :foo?`
Removed: foo9

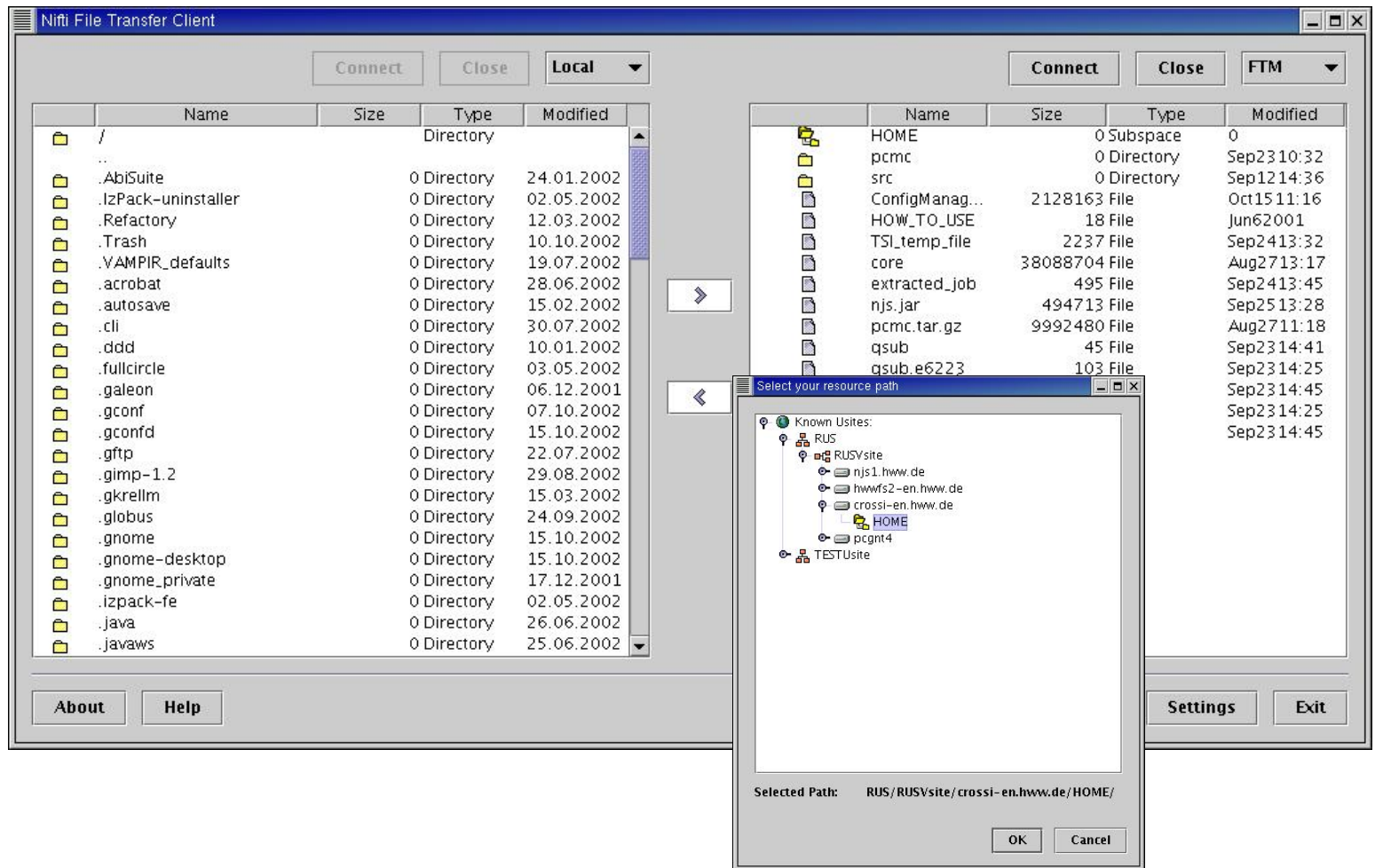


The Command Line Interface – for Unix clients

- *Session* mode makes single user sign-on and SSL connection for multiple requests (like ssh)
 - `unix> Fconnect @:fs2/indat`
connect to subspace indat on fs2 at default Vorg
 - `:fs2/indat> ls -l foo?`
`-rw-r--r-- 1 myID myGroup 17440 Dec 15 22:43 foo9`
 - `:fs2/indat> pwd`
`@134.56.14.44:fs2/indat`
 - `:fs2/indat> lpwd`
`/userhomes/myhome`
 - `:fs2/indat> lcd lclDir`
cd to lclDir on local client
 - `:fs2/indat> get foo?`
Copies remote foo9 to lclDir
 - `:fs2/indat> rm foo9`

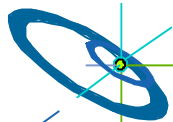


The Nifti GUI Interface – for Unix and Windows clients

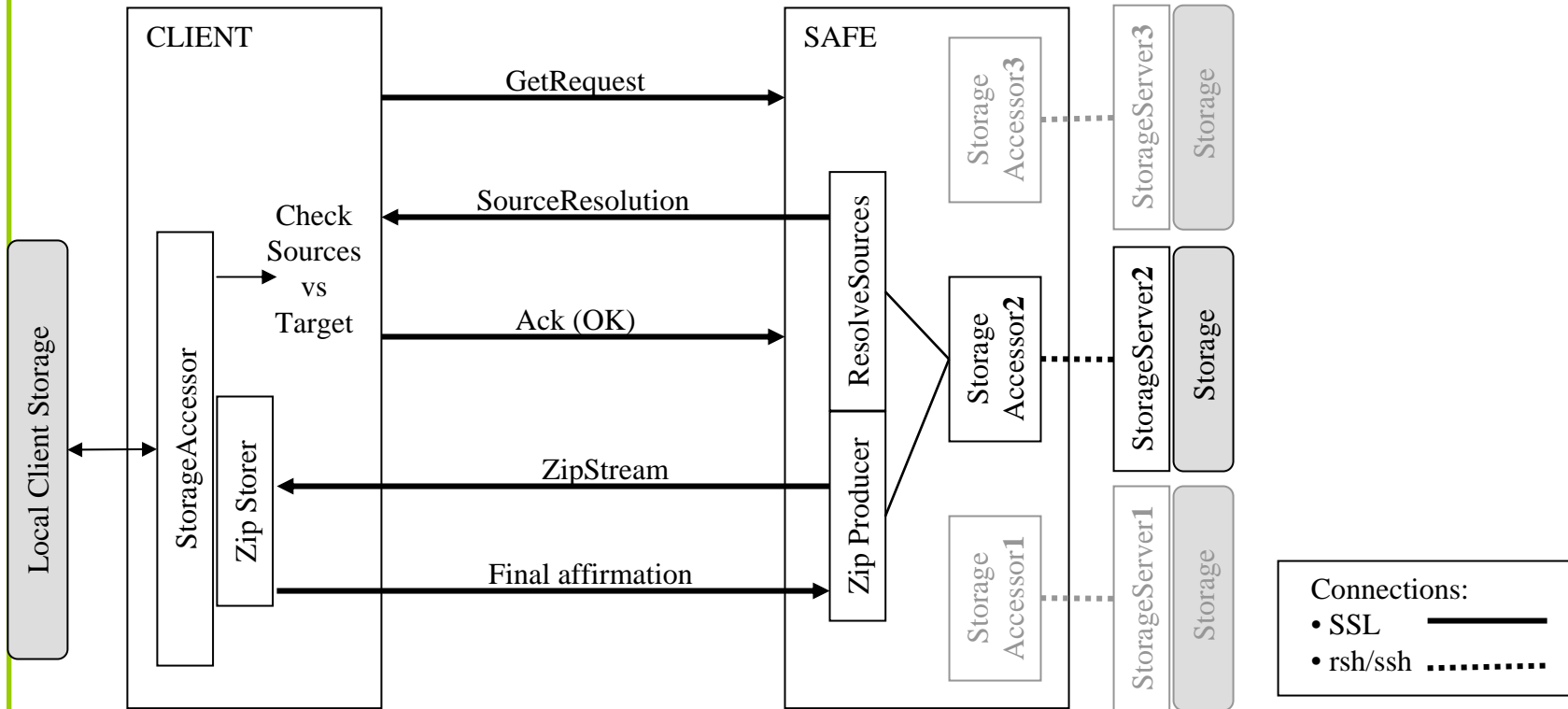


FTM Compared to SCP/SSH/FTP

- Single interface for access and transfer
- Meant for users in a Public Key Infrastructure (PKI)
 - Single user identity for all Vorgs
 - Single secure firewall interface independent of IP addresses
 - No passwords on servers
 - No key setup on individual servers
- Broader file functionality
 - Third party data transfer
 - Control of overwrite and backup
 - Subspace shortcuts
 - Optional encryption, compression
- Settable limits for data volume and number of files

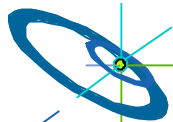


FTM architecture; the GET Protocol



Computational workflow based on FTM

- *SAPHIR* (Secure Access Portal to HPC Internet Resources)
- Complex jobs and data requirements as XML *jobSpecn*
- Workflow engine carries out requests in the *jobSpecn* using *FTM*:
 1. *Create a workspace on target platform*
 2. *Assemble required data from client and servers*
- including data from embedded jobs (=> dependency model)
 3. *Execute computational processes, either interactive or batch*
 4. *Distribute results to servers; return specified results to client*
- Client-centered model Facilitates interactions with the workflow, But requires client to remain connected
- Server-centered workflow engine under development



Saphir GUI interfaces for job development and execution

