Mitglied der Helmholtz-Gemeinschaft

UNICORE

JÜLICH
FORSCHUNGSZENTRUM

UFTP

Bernd Schuller and UNICORE Team
Federated Systems and Data division,
Jülich Supercomputing Centre

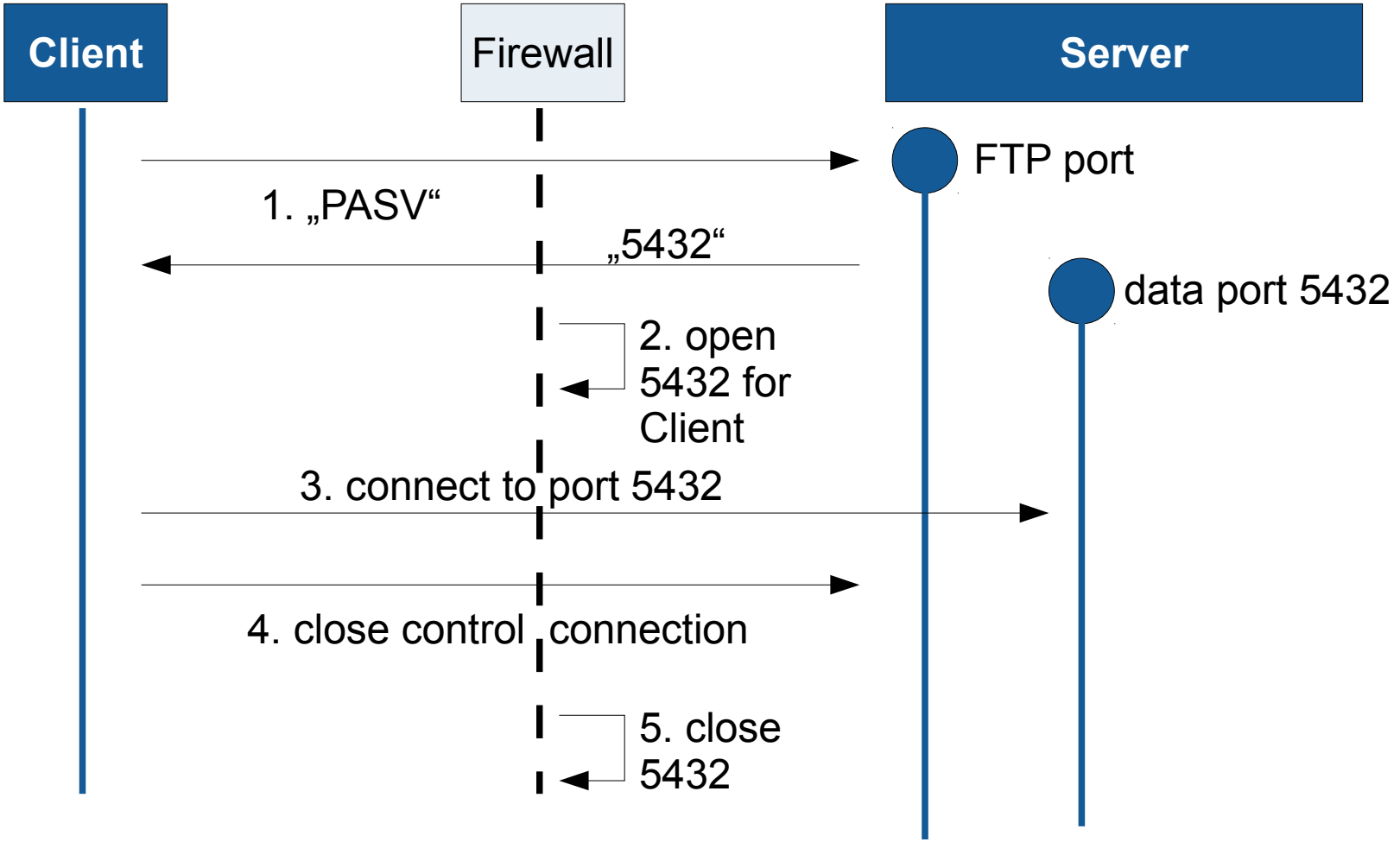
Common data transfer issues

JÜLICH FORSCHUNGSZENTRUM

- Firewall
  - Direct connections from the outside to the login node(s) are usually not allowed
  - Statically opening ports (or worse, port ranges) is a security risk

    → *need dynamic port opening technique*
- User authentication
  - AAI integration
  - User ID / group IDs mapping
- Avoid extra „hops“
  - e.g. when uploading data via web application

Solving the firewall issue: using passive FTP to open ports

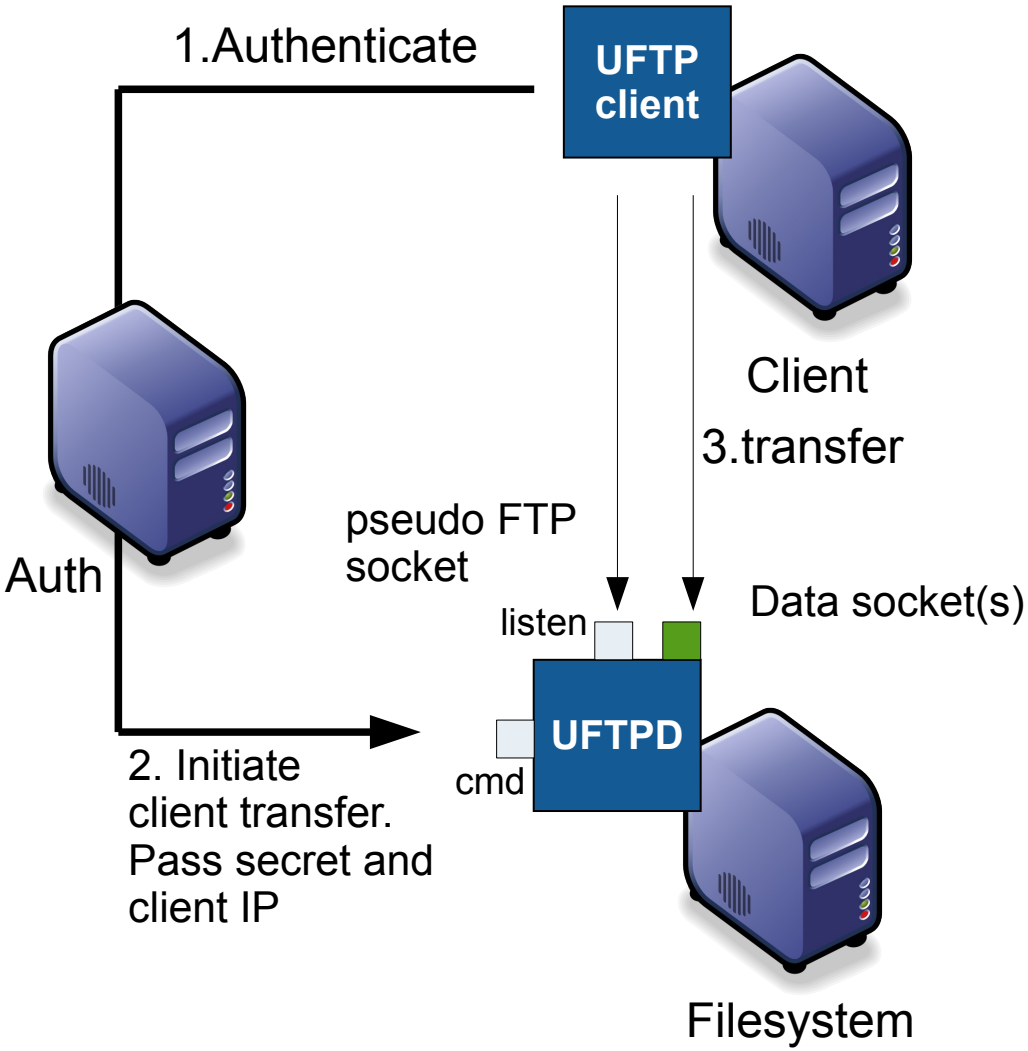


UFTP = passive FTP plus separate AuthN

- FTP by itself is insecure:
  - Users log in using username/password
- UFTP adds a secure channel from client to server which is used for additional security measures:
  - Authenticate clients
  - Map user ID / group IDs
  - Initiate data transfers
- Requires an secure „command port“ in addition to the FTP port

UFTP components

- UFTPD server
  - Pseudo-FTP port (open in firewall) for clients
  - Local command port (SSL protected) used by Auth server
  - Run as root w/ setuid
- UFTP client
  - Authenticate
  - Connect to UFTPD
  - Send/receive data
- Auth server
  - Client authentication
  - User ID mapping



Security challenges and their resolution

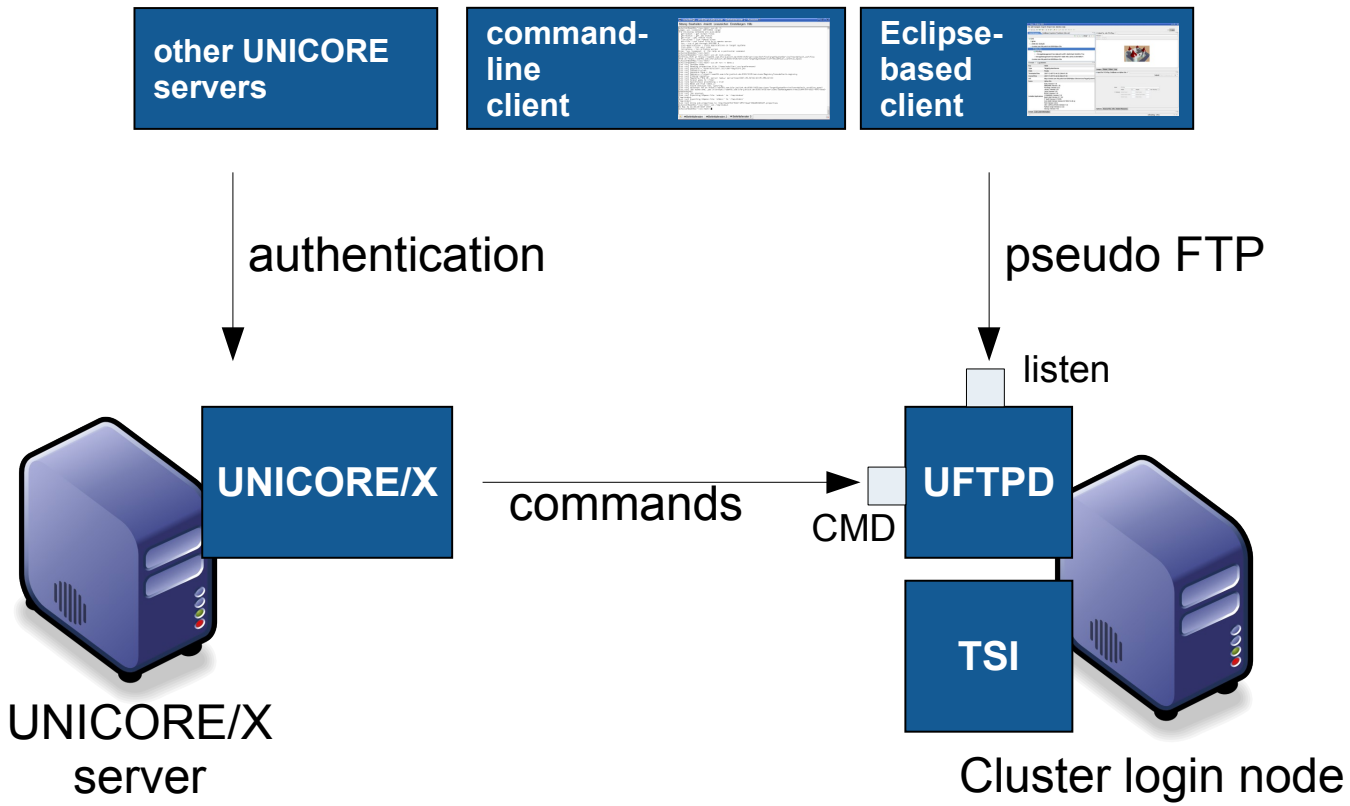
JÜLICH FORSCHUNGSZENTRUM

- uftpd server runs with root privileges (because it needs to access files from all users)
  - Switch effective user/group ID before file access
- Sending commands via the Command channel allows local users to read/write files under any user ID
  - Command port **never** accessible outside the firewall
  - Protect it using client authenticated SSL and ACL file
- Attacker might connect to the newly opened sockets on the uftpd server
  - Client IP is checked, and a secret key is required for authentication
- Data channels might be sniffed
  - Optional symmetric encryption (64 bit key, blowfish algorithm)

Applications and use cases

- File transfer and data staging in UNICORE
- Standalone use (client with auth server)
- Integrate into web applications
  - Planned master thesis: Data access and sharing at JSC (UFTP+AAI+HPC storage cluster)
- More?

UFTP deployment

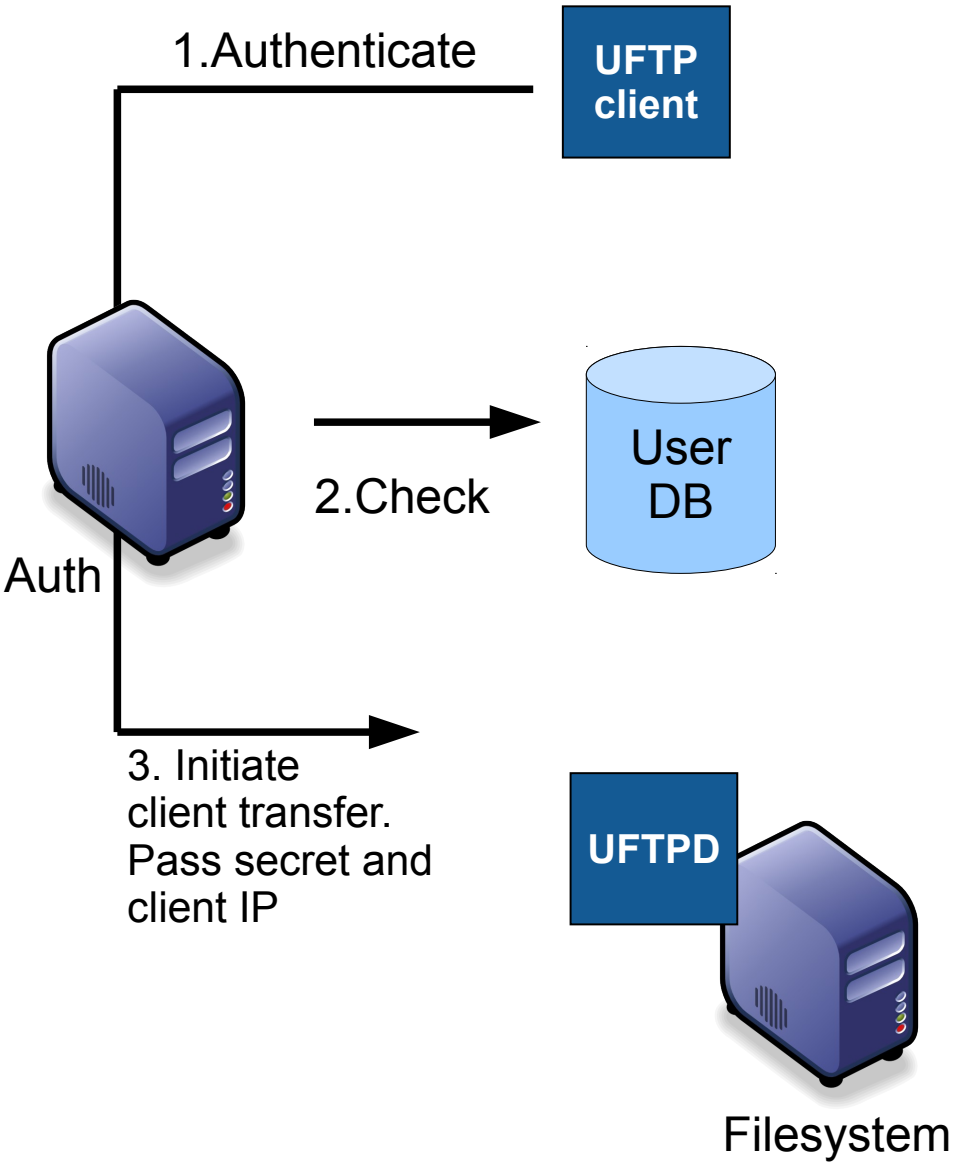


Standalone „auth server“

- Authentication
  - Username is given
  - Password check
  - sshkey check
- User „database“
  - uid, gid
  - Password, ssh public key

→ maps username given by client to attributes such as uid and gid

- RESTful service



Outlook – current work in progress

JÜLICH FORSCHUNGSZENTRUM

- Standalone client
  - Packaging (deb, rpm), documentation, usability
  - Support for more UFTP features: rsync, shell mode, encryption, compression
  - Better usability
  - Support for SSH-Agent, DSA keys
- Standalone auth Server
  - Multiple keys per user
  - Run in UNICORE container and behind Gateway
- Mid term: make UFTP more compliant with standard FTP
  - Want to use non-Java FTP code (e.g. Python or Perl) on the client
  - Even possible to mount in local file browser etc.

UFTP summary

- Fast file transfer library similar to FTP
- Firewall friendly and secure
- Fully integrated into UNICORE for data staging and client/server data movement
- Standalone use possible
- Flexible integration options (portals, …) via separate authentication server
- Implementation in Java
- Available as tgz, rpm, deb