Storage4PUNCH – Overview

Based on two storage technologies

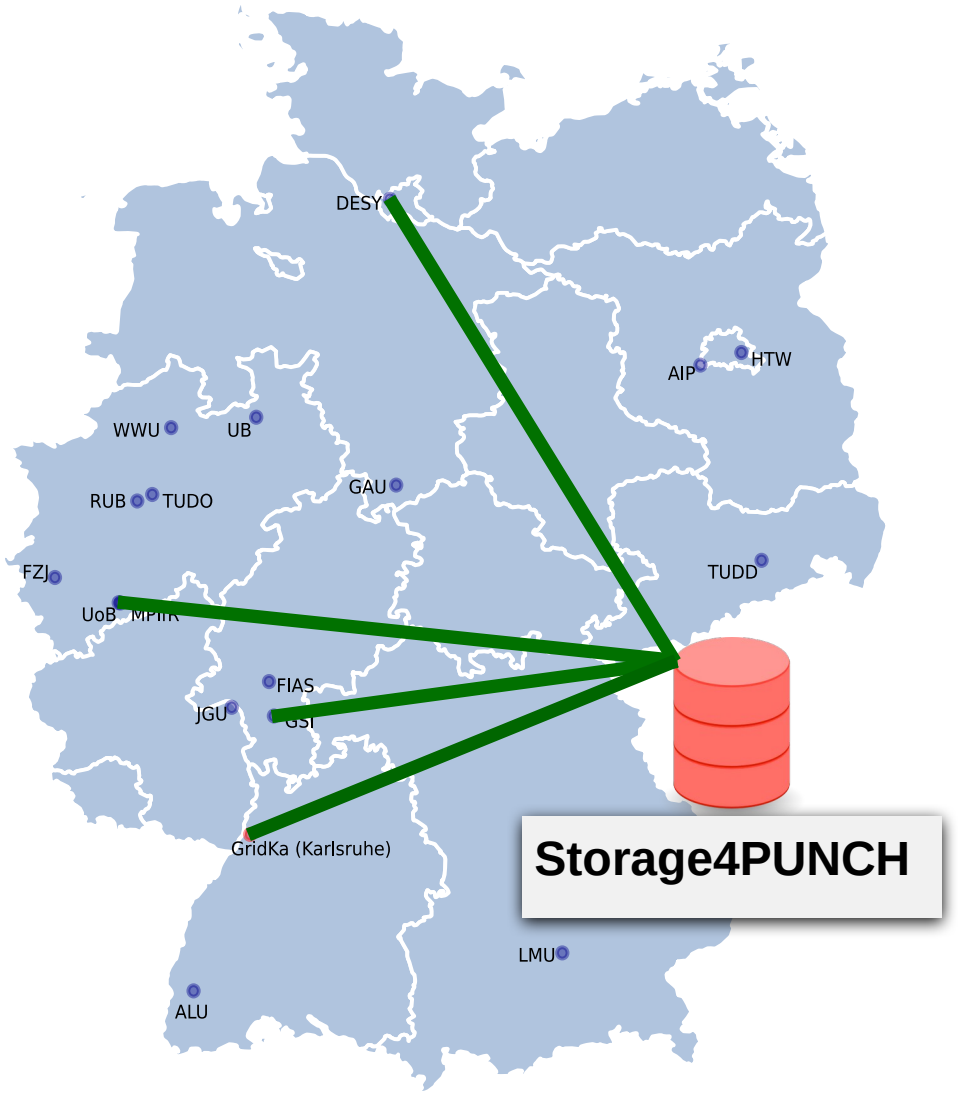
- Dcache: Instances at DESY & KIT
- XrootD: Instances at U Bonn & GSI

Token based access using PUNCH AAI

Supported protocols: WebDAV & XrootD

Client tools

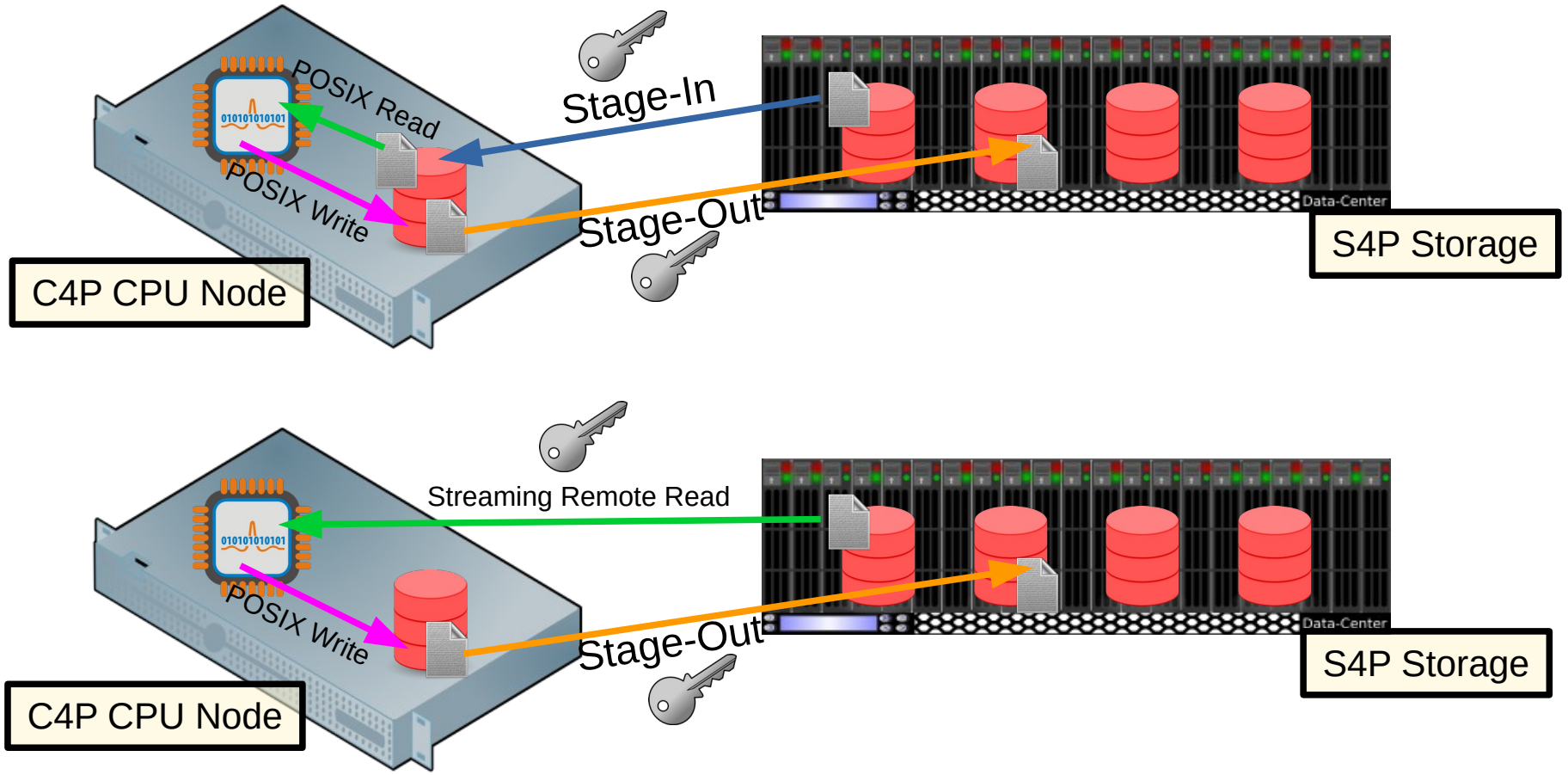
- gfal2
- davix
- Plain curl
- All are quite complex, some “easier” wrappers are under development



S4P -- Accessing / Processing Files

Distributed storage is typically not POSIX accessible

Application needs special features or files are staged to local POSIX compliant storage at run time



AAI Constrains

PUNCH AAI / Helmholz AAI based on Unity

Present configuration is very limited

- All PUNCH members can read/write “everywhere” on S4P
- No “public” access without any token

Usage of AAI groups now possible

- Required XrootD code contribution via PUNCH4NFDI
- Could use small number of groups
  - e.g. write permissions in certain directories only for a “writer or admin group”
  - Limited read access for groups
  - Each S4P instance needs to be configured to support group scheme

Scope based approach

- Permissions are encoded in tokens itself
- No configuration of individual S4P instances
- However significant enhancements of Unity AAI stack required to support it
  - A first development request towards “scope based approach” recently submitted

Data Management Components

PUNCH metadata catalog

- Originally developed for LQCD community, supports generic metadata schemata for other applications
- Can serve as file catalog (and does for LQCD community)
- In (pre-) production use for the ILDG VO with Indigo IAM (“WLCG” AAI solution) as token issuer
- Client tools still rough

Rucio testbed

- Rucio is data management middleware used in HEP (ATLAS, Belle II, CMS) and beyond (SKA, CTAO)
- Started to set up a PUNCH test instance with the help of HIFIS colleagues
  - Bonn, DESY & KIT instance are configured
  - Presently no automatic population of PUNCH users, adding users presently manual operation