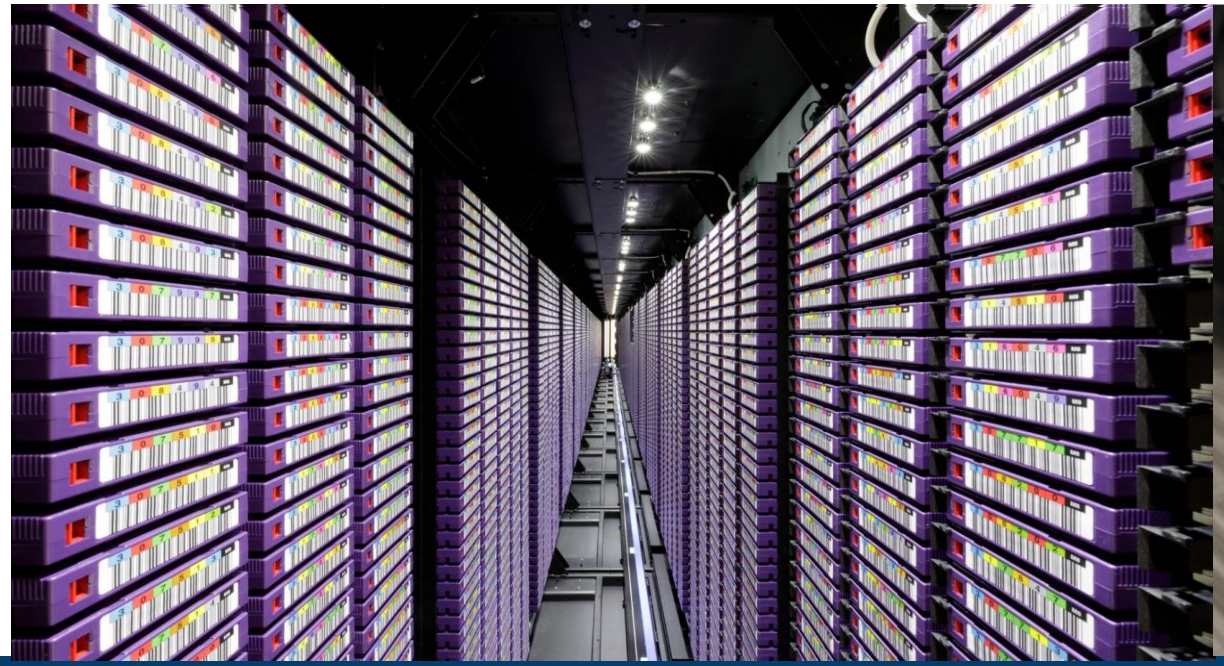


BACKUP & ARCHIVE OF HPC DATA

The Backup Infrastructure of the Jülich Supercomputing Centre

18.03.2025 I SALEM EL SAYED, **STEPHAN GRAF**, MARTIN LISCHWESKI & OLEG TSIGENOV

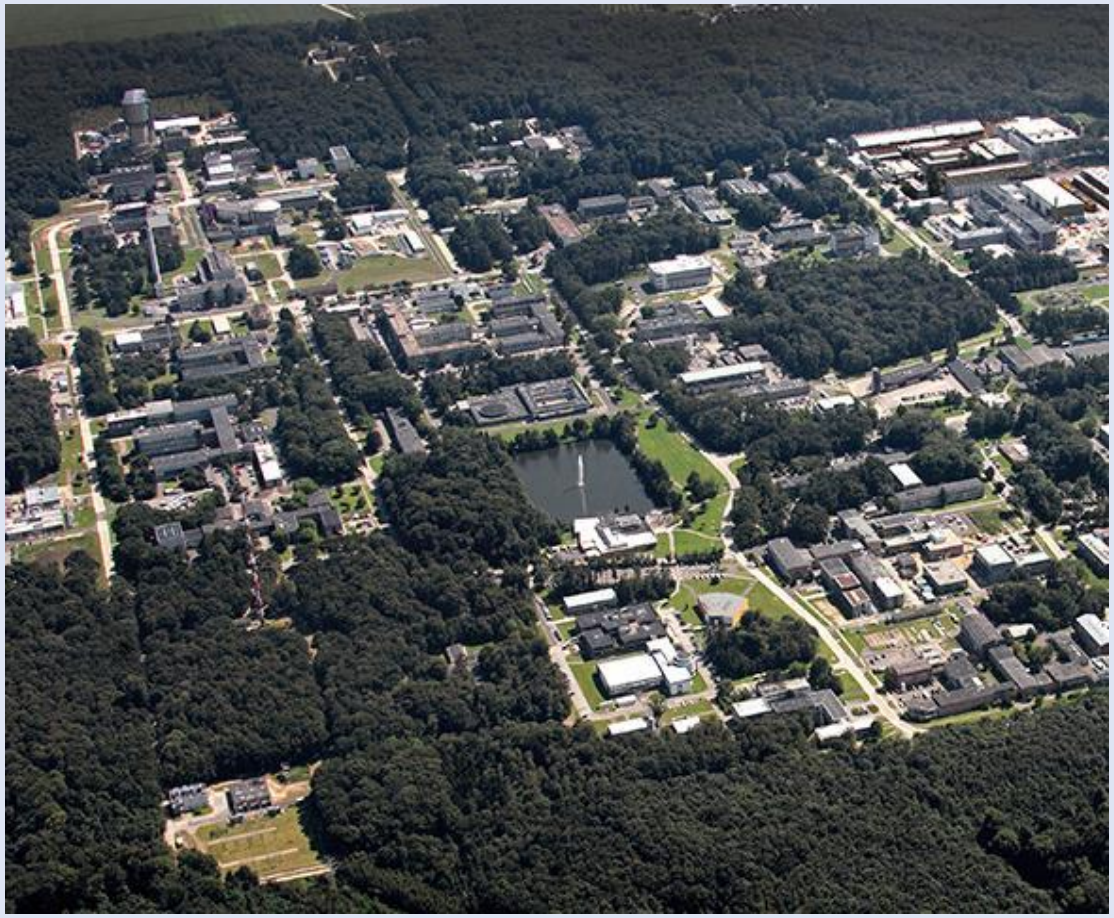
Member of the Helmholtz Association

JÜLICH
Forschungszentrum

TABLE OF CONTENTS

RESEARCH AND DEVELOPMENT

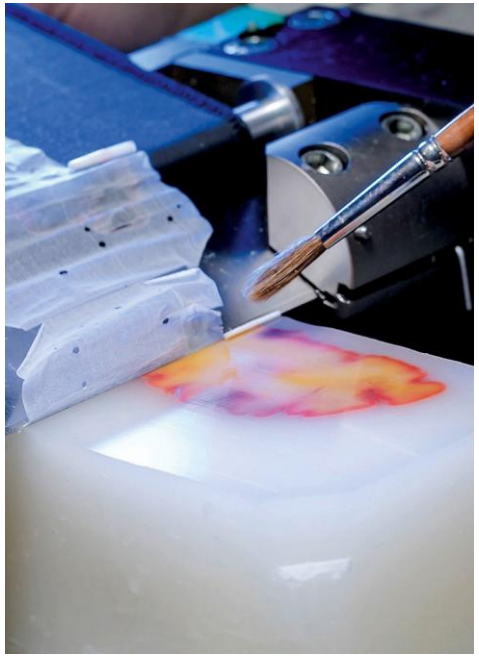
on 2.2 Square Kilometres



AT FIRST GLANCE

Founded in 1956

Revenue:
€ 987 million in 2023



Research priorities:
information, energy, bioeconomy

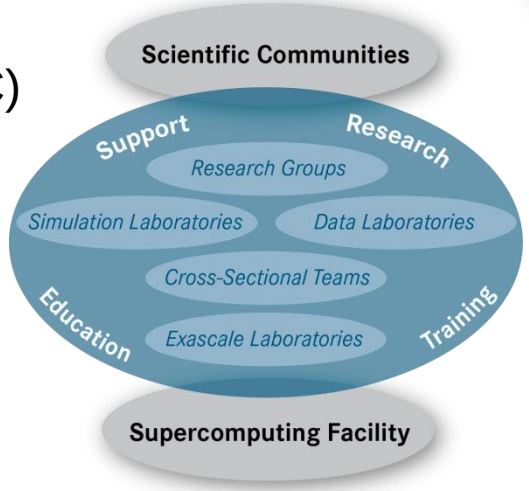
Research campus with 14 institutes and 18 branch offices in Germany and abroad

Shareholders:
Federal Republic of Germany (90%), federal state of North Rhine-Westphalia (10%)

Member of the Helmholtz Association

JÜLICH SUPERCOMPUTING CENTRE

- **Supercomputer operation for**
  - Centre – FZJ
  - Region – RWTH Aachen University
  - Germany – Gauss Centre for Supercomputing (GCS)
    John von Neumann Institute for Computing (NIC)
  - Europe – EuroHPC, PRACE, EU projects
- **Application support**
  - Unique support & research environment at JSC
  - Peer review support and coordination
- **R&D work**
  - Methods and algorithms, computational science, performance analysis and tools
  - Scientific Machine Learning and AI with HPC
  - Computer architectures, Co-Design, Exascale Labs together with IBM, Intel, NVIDIA
- **Education and training**



First Exascale supercomputer in Europe

DEEP

JÜLICH Forschungszentrum

JÜLICH SUPERCOMPUTING CENTRE

~300 Employees
(~280 FTE)

210 Scientists

22 PhD Students (+ externals)
27 Students (Bachelor/Master)

fz-juelich.de/jsc

also some JSC

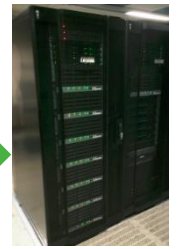
also some JSC

JSC

JUELICH STORAGE (JUST)

JUWELS + JUWELS Booster
2600+ Nodes

DEEP



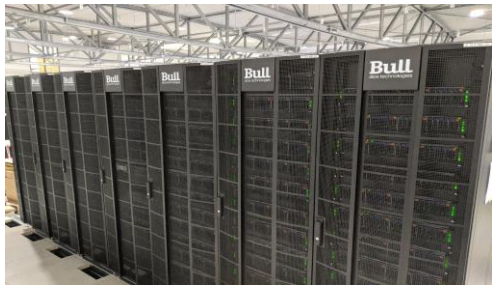
JARVIS

JUST

JURECADC
700+ Nodes

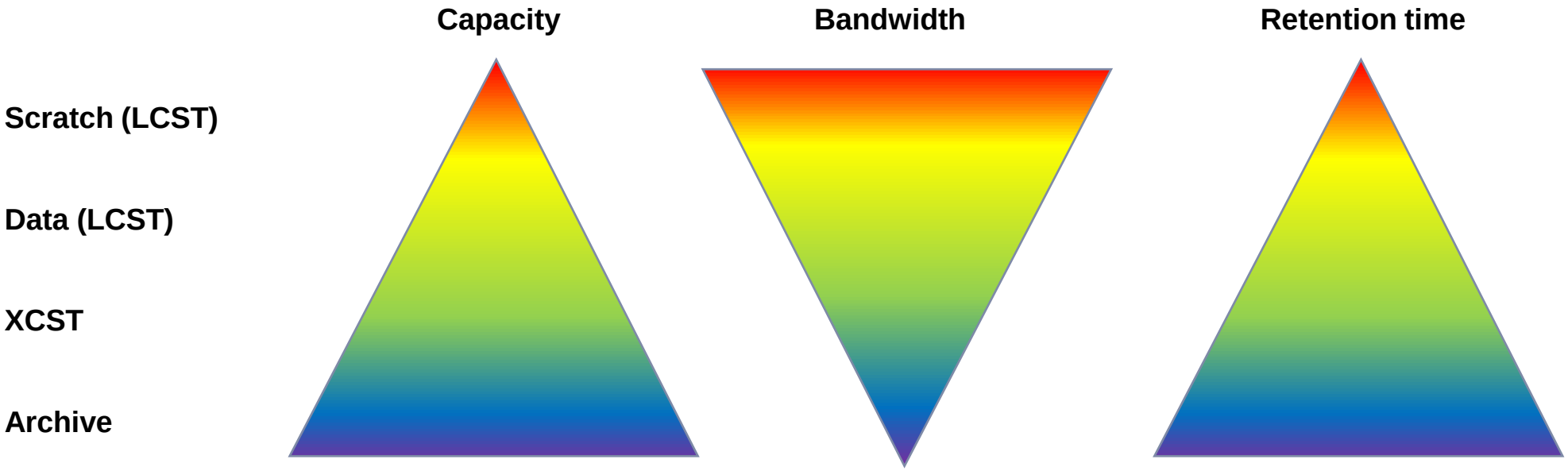
JUDAC

minions



JUSUF
182+ Nodes

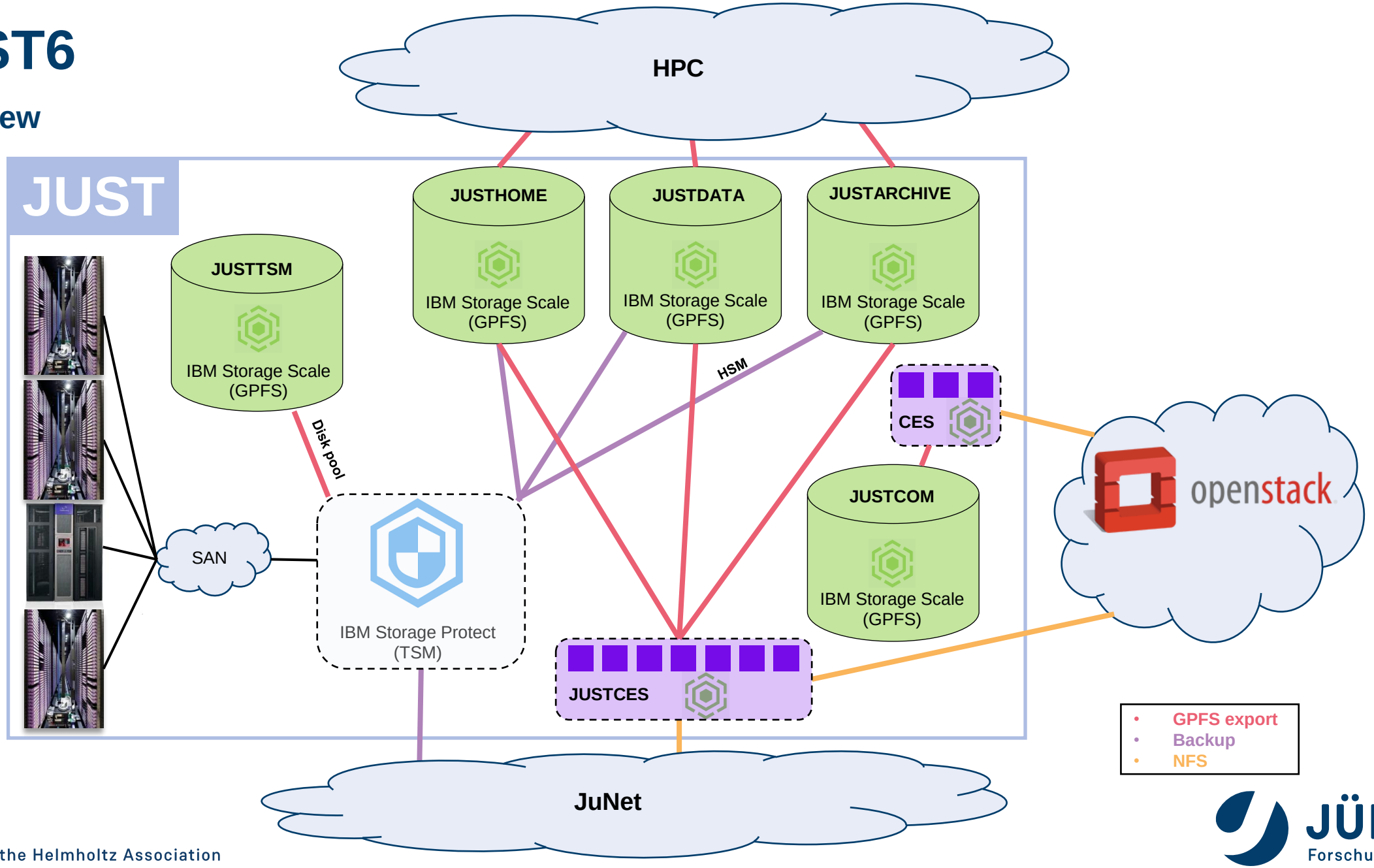
TIERED STORAGE OFFERING



- Large Capacity Storage Tier: IBM ESS/SSS Cluster (GNR, 6th Gen. of JUST, bandwidth optimized) → LCST
- Extended Capacity Storage Tier (XCST): GPFS Building Blocks (target: capacity) → XCST
- Archive: Tape storage (Backup + GPFS&TSM-HSM)

JUST6

Overview



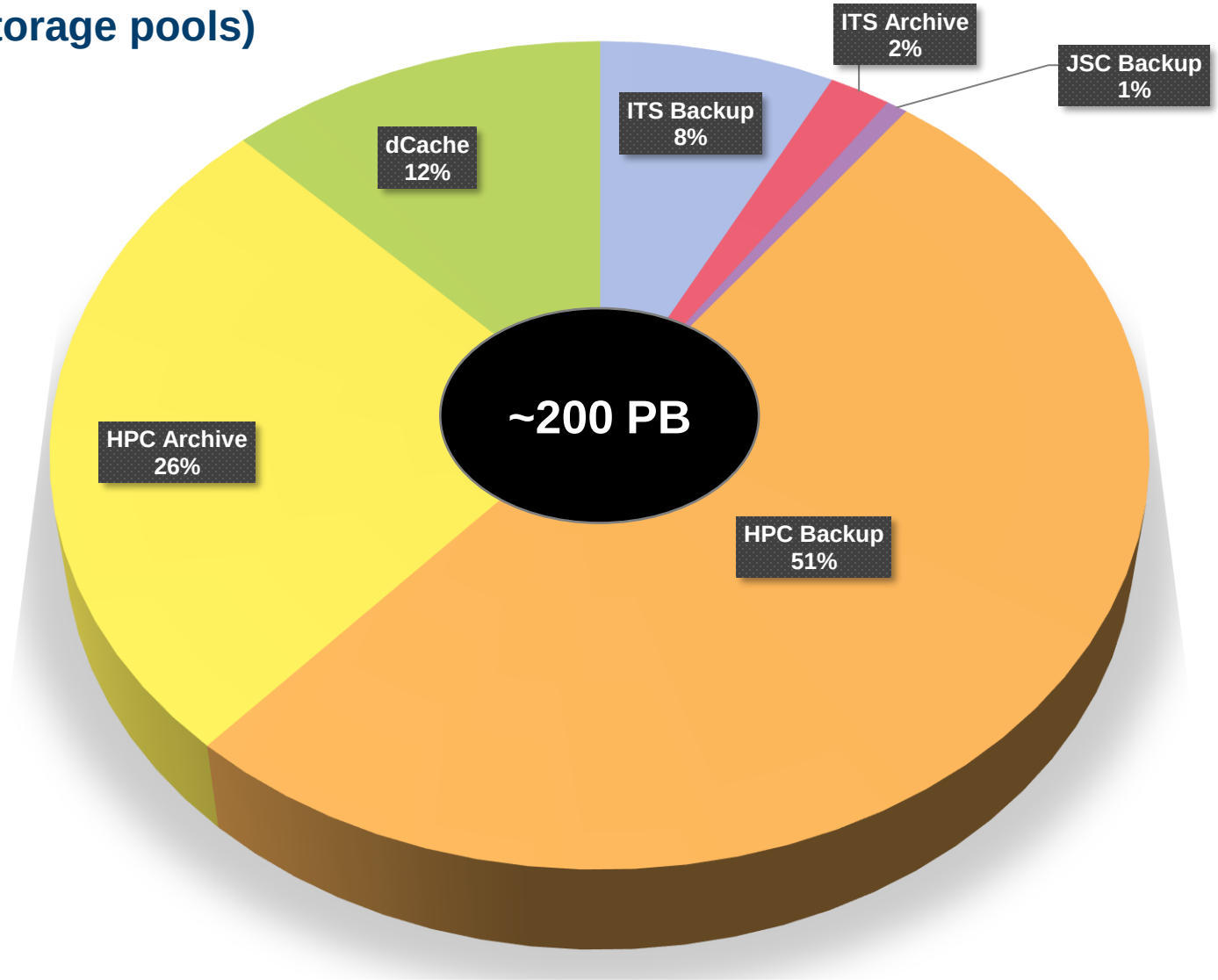
HPC USER DATA (USAGE)

File System	Shell Variable	Description	INODES	Capacity	HSM	(mm)Backup
home	**$HOME**	Users HOME File Systems	76.212.261	20 TB		✓
project1	**$PROJECT**	Compute Project File System	782.196.564	4.1 PB		✓
scratch	**$SCRATCH**	Compute Project Scratch File System	239.926.611	11 PB		
data1	**$DATA**	High Bandwidth and large Capacity File System	272.208.632	32 PB		✓
largedata2	**$LARGEDATA**	Large Capacity (Disk based, deprecated)	45.302.753	7.3 PB		✓
arch1 arch2	**$ARCHIVE**	Archive File System (Tape)	29.346.284	~47 PB	✓	✓
		Total:	**1.445.193.105**	**~100 PB**		

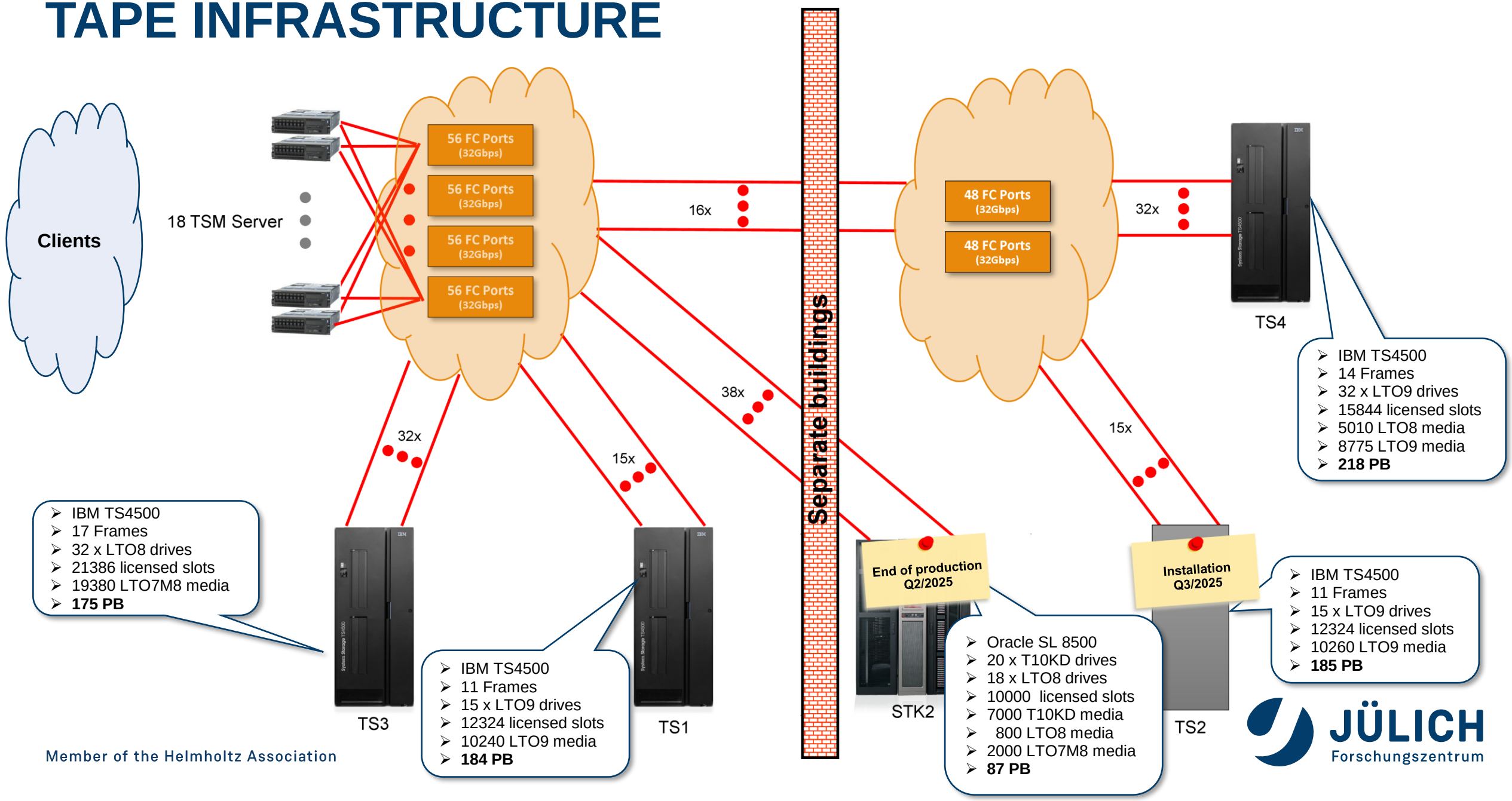
Goal is to have a daily backup!

OVERALL BACKUP & ARCHIVE USAGE

(used capacity of TSM storage pools)



TAPE INFRASTRUCTURE



TAPE@JÜLICH: SOME FACTS/FINDINGS

- No partitioning: TSM is the one and only client
- No media verification. Tape drives are always the bottleneck.
- We moved away from Oracle Enterprise to LTO
  - IBM Enterprise cartridges and LTO cartridges have different form factor
- Tape behavior in TSM for HPC data: no compression on tape

Cartridge Type	Capacity (filling state)	Capacity (100% full)	Specification
LTO9	40.9 T	~ 18 T	18TB
LTO8	27.3 T	~ 11.7 T	12 TB
LTO7M8	20.5 T	~ 8.83 T	9 TB
T10KD	15.3 T	~ 8.15 T	8.5 TB

MANAGING IBM LTO DRIVES IN LINUX AND STORAGE PROTECT

Member of the Helmholtz Association

JÜLICH
Forschungszentrum

USING TAPE LIBRARY IN LINUX

Tape Access

- Install lin_tape + lin_taped packages for the IBM TS4500
- **Udev** Rules to create persistent device name links
  Oracle T10KD drives:

```
SUBSYSTEM=="scsi_generic", MODE="0666"
SUBSYSTEM=="scsi_generic", SUBSYSTEMS=="scsi", ATTRS{vendor}=="STK*", ATTRS{model}=="T10*", PROGRAM="/lib/udev/scsi_id --page=0x80 --
whitelisted -d $tempnode", RESULT=="*579001234567*", SYMLINK+="L2D01"
```

- IBM LTO drives:

```
KERNEL=="TSMtape", MODE="0666"
KERNEL=="IBMchanger*", MODE="0666", ATTRS{serial_num}=="*78BB123*", ATTR{primary_path}=="Primary", SYMLINK+="TS3control"
KERNEL=="IBMtape*[0-9]", MODE="0666", ATTRS{serial_num}=="*10WT012345*", SYMLINK+="L3D01"
```

- Create module parameter file **`/etc/modprobe.d/lin_tape.conf`**

```
options lin_tape alternate_pathing=1
options lin_tape tape_reserve_type=persistent
```

- Module is loaded on boot by creating `/etc/modules-load.d/lin_tape.conf`

```
lin-tape
```

- Prepare systemctl service to start Oracle ACSLS service
  `/opt/Tivoli/tsm/devices/bin/rc.acs_ssi`

USING TAPE LIBRARY IN LINUX

TSM setup & Systemd dependencies

- Define library

```
define library TS1 libtype=scsi shared=yes RESETDrives=yes AUTOLabel=yes
```

- Define control path

```
DEFINE PATH tsmstk TS1 SRCTYPE=SERVER DESTTYPE=LIBRARY DEVICE=/dev/TS1control
```

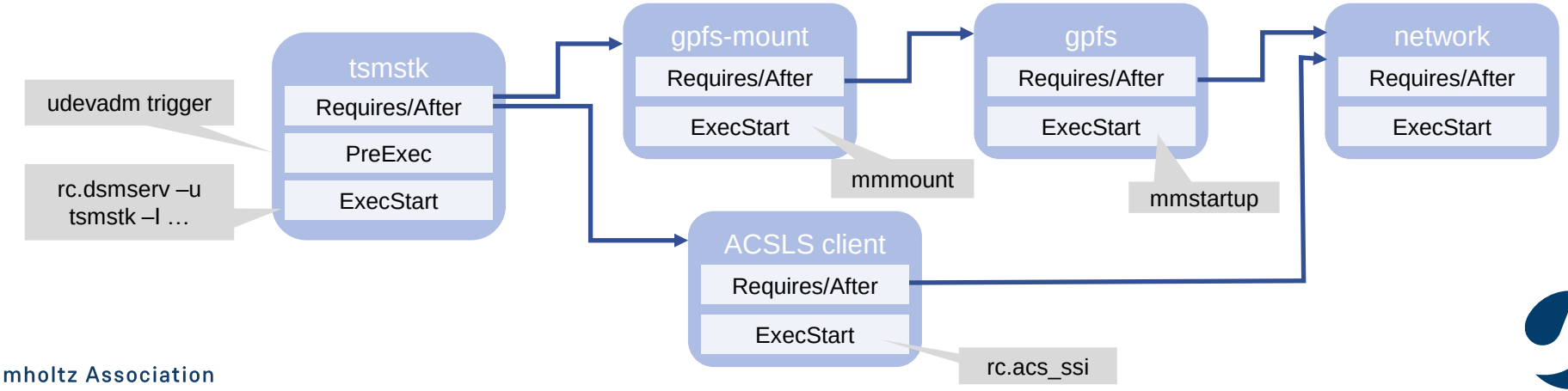
- Define drives

```
define drive TS1 L1DD01 serial=autodetect online=yes element=autodetect cleanfrequency=none
```

- Define path for all drives to all instances on the TSM shared tape library instance (tsmstk)

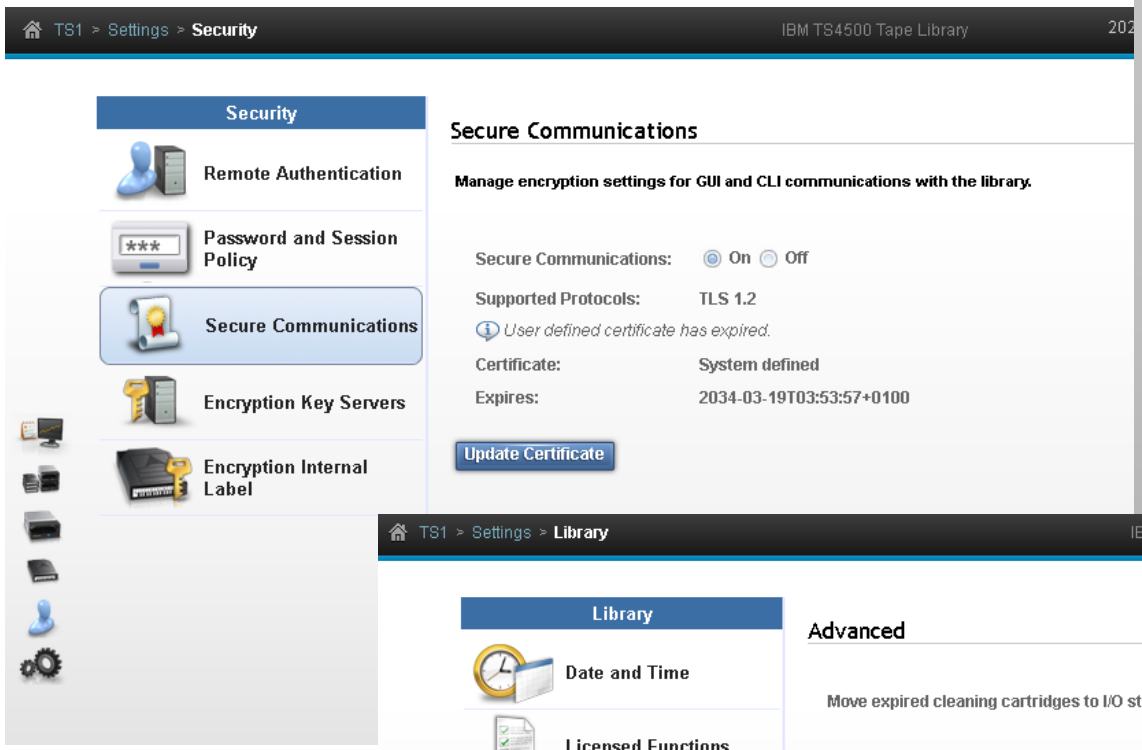
```
define path tsmarch L1D01 srctype=server desttype=drive online=yes device=/dev/L1D01 autodetect=yes library=TS1
```

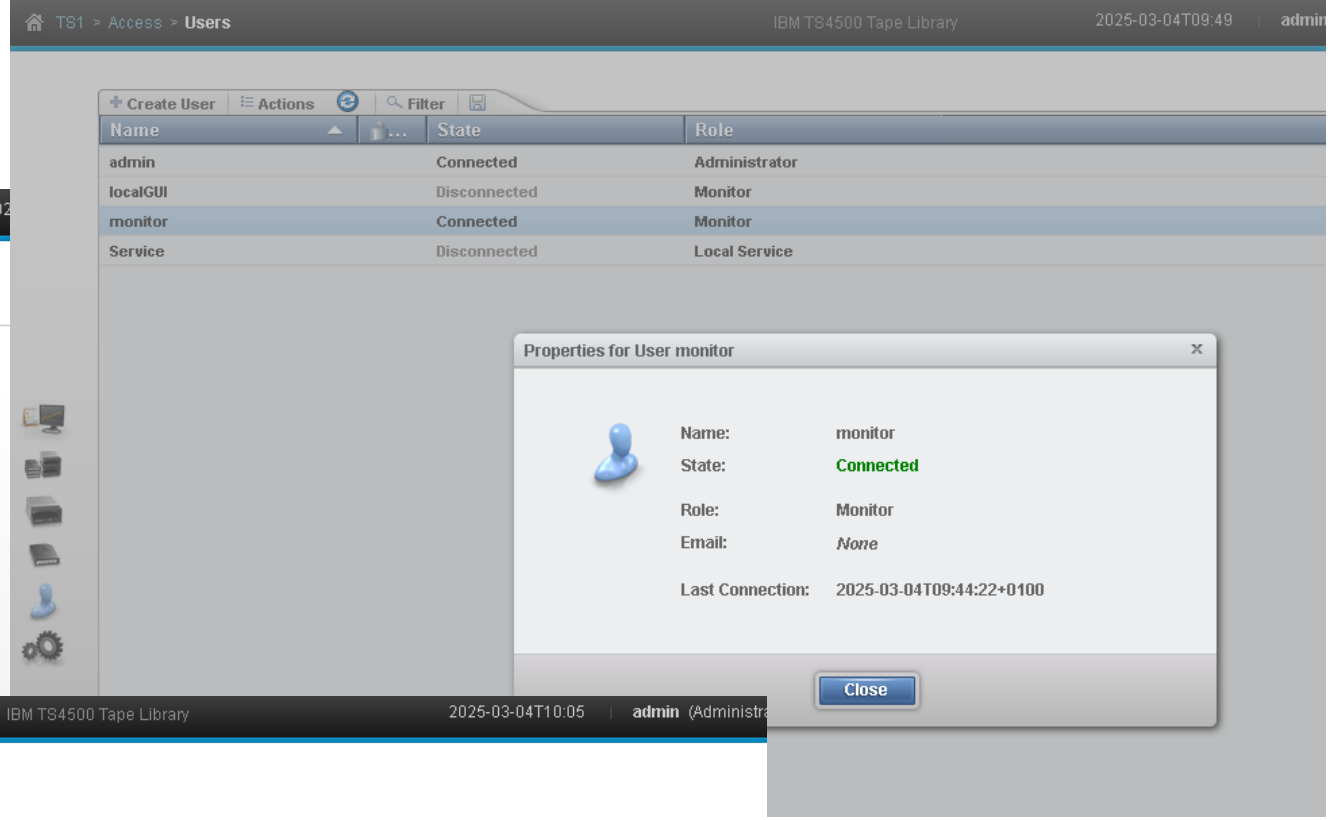
- **Use systemd to start, stop and solve dependencies**

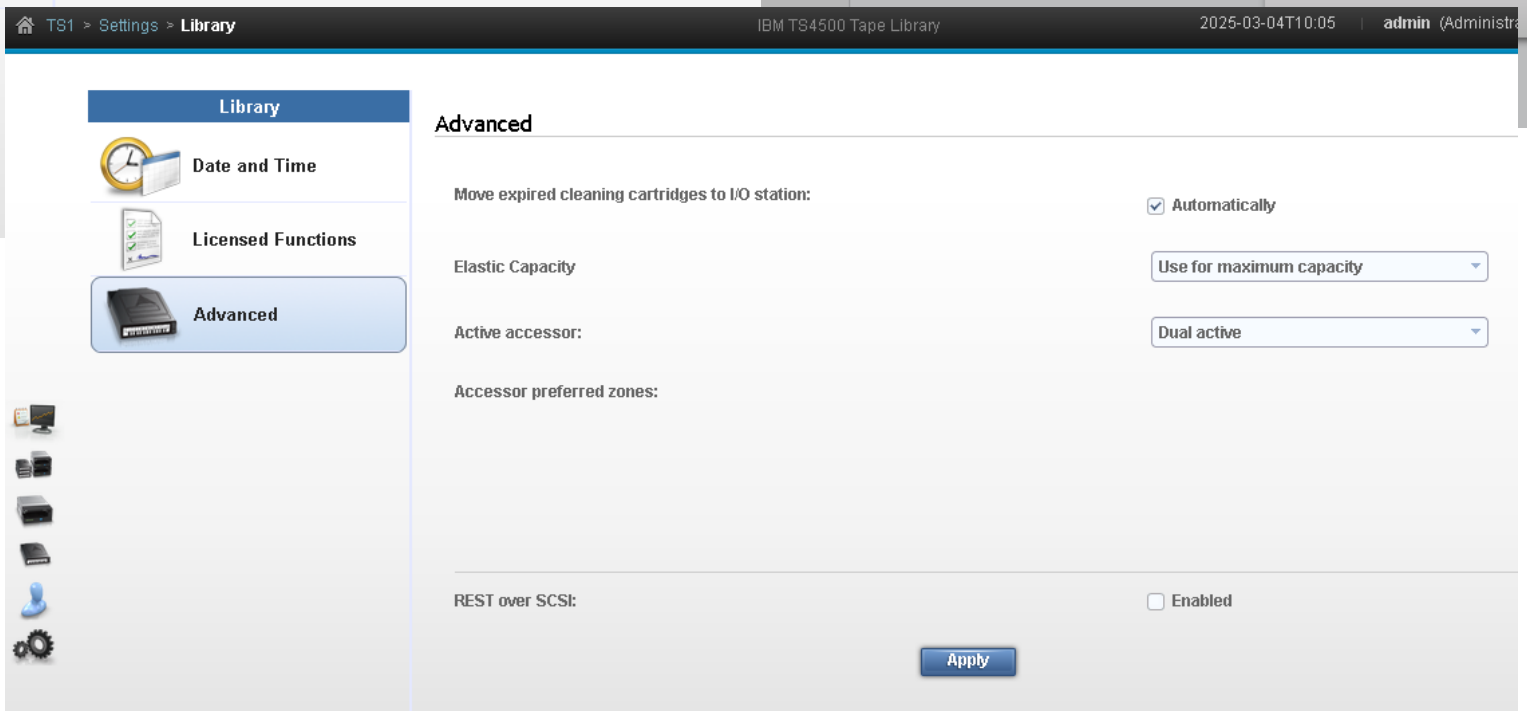


USING TS4500 REST API FOR MONITORING

ENABLE REST API







TS4500 REST OVER ETHERNET

Custom python script

```
#!/usr/bin/env python3
"""
TS4500 REST over Ethernet
"""

import argparse
import datetime
import json
import requests
import urllib3

urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

VERBOSE = False

PASSWORD_FILE = "/root/.ts4500.credentials"

def get_password() -> str:
    """return password from file"""
    with open(file=PASSWORD_FILE, mode="r", encoding="UTF-8") as
file:
        return file.read().strip().split(':')
```

```
class ts4500rest:
    """
    REST over Ethernet interface to the tape library
    - will return json object: self.json_data
    """

    json_data = []

    def __init__(self, ip_number) -> None:
        """get JSESSIONID aka COOKIES"""
        (username, password) = get_password()
        self.ip_number = ip_number
        url = "https://" + ip_number + "/web/api/v1/login"
        payload = {"user": username, "password": password}
        session = requests.Session()
        session.post(url=url, json=payload, verify=False)
        self.cookies = session.cookies.get_dict()
        if self.cookies is None:
            raise RuntimeError("Cookies not found")

    def get_data(self, target="library"):
        """return json_data for target"""
        url: str = "https://" + self.ip_number + "/web/api/v1/" + target
        request_output = requests.get(url=url, cookies=self.cookies, verify=False, timeout=60)
        self.json_data: list[dict[str, str]] = json.loads(request_output.content.decode())
        return self.json_data

    def get_events(self, days: int = 0):
        """
        Fetch and filter event list of target library
        - return: list of event dictionary
        """
        target: str = "events"
        if days > 0:
            start_time = datetime.datetime.now() - datetime.timedelta(days=days)
            target += "/?after=" + start_time.astimezone().replace(microsecond=0).isoformat()
        return self.get_data(target=target)
```

TS4500 REST OVER ETHERNET

Custom python script output example

```
# calls in MAIN
lib_session = ts4500rest(ip_number=ip_number)
lib_json_data = lib_session.get_data(target="library")
lib_json_data = lib_session.get_data(target="drives")
lib_json_data = lib_session.get_data(target="frames")
events_json_data = lib_session.get_events(days=31)
```

```
[root@just-master just_admin_scripts]# python3 ts4500rest.py
|           | type | SN |
|-----------|------|----|
| Library | TS1 | 78XXXXX |

| Drive     | Location | Type | SN |
|-----------|----------|------|----|
|     | F4C1R1 | 3588-F9C | 78      |
|     | F4C1R2 | 3588-F9C | 78      |
|     | F4C1R3 | 3588-F9C | 78      |
|     | F4C1R4 | 3588-F9C | 78      |
|     | F4C2R1 | 3588-F9C | 78      |
|     | F4C2R2 | 3588-F9C | 78      |
|     | F4C2R3 | 3588-F9C | 78      |
|     | F4C2R4 | 3588-F9C | 78      |
|     | F4C3R1 | 3588-F9C | 78      |
|     | F4C3R2 | 3588-F9C | 78      |
|     | F4C3R3 | 3588-F9C | 78      |
|     | F4C3R4 | 3588-F9C | 78      |
|     | F4C4R1 | 3588-F9C | 78      |
|     | F4C4R2 | 3588-F9C | 78      |
|     | F4C4R3 | 3588-F9C | 78      |
|     | F4C4R4 | 3588-F9C | 78      |
|     | F8C1R1 | 3588-F9C | 78      |
|     | F8C1R2 | 3588-F9C | 78      |
|     | F8C2R1 | 3588-F9C | 78      |
|     | F8C2R2 | 3588-F9C | 78      |
|     | F8C2R3 | 3588-F9C | 78      |
|     | F8C2R4 | 3588-F9C | 78      |
|     | F8C3R1 | 3588-F9C | 78      |
|     | F8C3R2 | 3588-F9C | 78      |
|     | F8C3R3 | 3588-F9C | 78      |
|     | F8C3R4 | 3588-F9C | 78      |
|     | F8C4R1 | 3588-F9C | 78      |
|     | F8C4R2 | 3588-F9C | 78      |
|     | F8C4R3 | 3588-F9C | 78      |
|     | F8C4R4 | 3588-F9C | 78      |
```

```
| Frame     | Type | SN | Slots | Drives | IO Stations |
|-----------|------|----|-------|--------|-------------|
| F1 | 3584-S55 | 78      | 1054 | 0 | 0 |
| F2 | 3584-S55 | 78      | 1320 | 0 | 0 |
| F3 | 3584-S55 | 78      | 1320 | 0 | 0 |
| F4 | 3584-L55 | 78      | 882 | 16 | 2 |
| F5 | 3584-S55 | 78      | 1320 | 0 | 0 |
| F6 | 3584-S55 | 78      | 1320 | 0 | 0 |
| F7 | 3584-S55 | 78      | 1320 | 0 | 0 |
| F8 | 3584-D55 | 78      | 882 | 14 | 2 |
| F9 | 3584-S55 | 78      | 1320 | 0 | 0 |
| F10 | 3584-S55 | 7      | 1320 | 0 | 0 |
| F11 | 3584-S55 | 7      | 1320 | 0 | 0 |

Last 10 events:
{'ID': 6533, 'severity': 'information', 'time': '2025-03-04T11:10:27+0100', 'location': None, 'user': 'monitor', 'description': 'A user logged in with REST from 10.201.9.82', 'state': None, 'errorCode': '0834'}
{'ID': 6532, 'severity': 'information', 'time': '2025-03-04T10:55:26+0100', 'location': None, 'user': 'monitor', 'description': 'A user logged out with REST from 10.201.9.82', 'state': None, 'errorCode': '0838'}
{'ID': 6531, 'severity': 'information', 'time': '2025-03-04T10:44:23+0100', 'location': None, 'user': 'monitor', 'description': 'A user logged in with REST from 10.201.9.82', 'state': None, 'errorCode': '0834'}
{'ID': 6530, 'severity': 'information', 'time': '2025-03-04T10:25:17+0100', 'location': None, 'user': 'monitor', 'description': 'A user logged out with REST from 10.201.9.82', 'state': None, 'errorCode': '0838'}
{'ID': 6529, 'severity': 'information', 'time': '2025-03-04T10:14:26+0100', 'location': None, 'user': 'monitor', 'description': 'A user logged in with REST from 10.201.9.82', 'state': None, 'errorCode': '0834'}
```

```
{'ID': 6528, 'severity': 'information', 'time': '2025-03-04T10:13:16+0100', 'location': None, 'user': 'monitor', 'description': 'A user logged out with REST from 10.201.9.82', 'state': None, 'errorCode': '0838'}
{'ID': 6527, 'severity': 'information', 'time': '2025-03-04T10:02:52+0100', 'location': None, 'user': 'monitor', 'description': 'A user logged in with REST from 10.201.9.82', 'state': None, 'errorCode': '0834'}
{'ID': 6526, 'severity': 'information', 'time': '2025-03-04T10:02:09+0100', 'location': None, 'user': 'admin', 'description': 'A logical library  setting was changed', 'state': None, 'errorCode': '0803'}
{'ID': 6525, 'severity': 'information', 'time': '2025-03-04T10:02:06+0100', 'location': None, 'user': 'admin', 'description': 'A logical library  setting was changed', 'state': None, 'errorCode': '0803'}
{'ID': 6524, 'severity': 'information', 'time': '2025-03-04T09:55:08+0100', 'location': None, 'user': 'monitor', 'description': 'A user logged out with REST from 10.201.9.82', 'state': None, 'errorCode': '0838'}
```

```
# for monitoring: check events of severity 'error'

def get_active_errors(json_data):
    """return list of active errors"""
    errors = []
    for event in json_data:
        if event["severity"] == "error":
            errors.append(event)
    return errors
```

OUTLOOK

- Finalize data migration from Oracle SL8500 until Ende of Q2/2025
- Dismantel Oracle SL8500 afterwards
- Install new TS4500 at the same location and bring it in production

Installation of Exascale supercomputer JUPITER will be finished soon

New Storage for JUELICH ExaByte FOR ExaFLOP

Talk tomorrow at 1pm

- A new large HPC System with new data for backup and archive

GREETINGS FROM THE STORAGE TEAM

IBM Tape User Day 2025
18. März 2025

09:00	Eintreffen	
10:00	Begrüßung	Stefan Neff, IBM
10:05	Storage und Tape Technologie - Whats New	Sepp Weingand, IBM
10:45	S3-To-Tape: IBM Scale & Archive und IBM Deep Archive	Nils Haustein, IBM
11:30	Backup & Archive of HPC Data @FZ Jülich	Stephan Graf, FZ-Jülich
12:00	Mittagspause	
13:00	AirGap Tape & Safeguarded Tape - Status und Outlook	Sepp Weingand, IBM
13:30	Safeguarded Tape - Kundenbeispiel	Uwe Schreiber, SVA System Vertrieb Alexander GmbH
14:00	eXstor Safeguard Tape & News about eXstor Library and Media Manager	Bruno Friess, eXstor GmbH
14:30	Kaffeepause	
15:00	BACKUP EAGLE® IBM TS4500 Reporting und Überwachung	Stefan Schröder Geschäftsführer . CTO - Schmitz RZ Consult GmbH
15:30	News aus dem Support	Erwin Zwemmer, IBM
16:00	Archive REST API	Nils Haustein, IBM
16:30	Sneak-Preview	Stefan Neff, IBM
16:55	Abschluss Diskussion & Verabschiedung	Stefan Neff & Sepp Weingand, IbM