

XRootD Server Plug-ins

XRootD Workshop
March 29-31, 2023

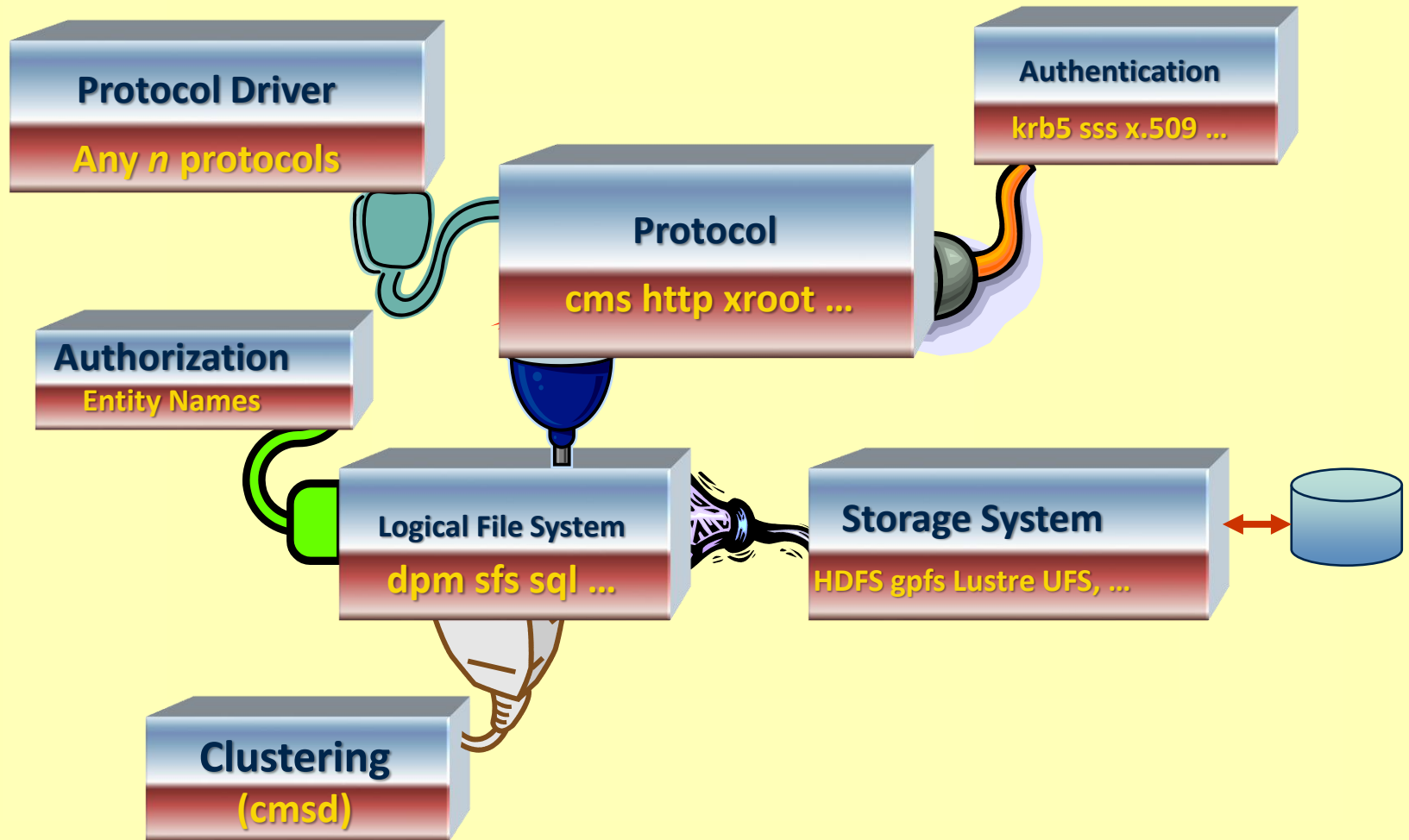
Andrew Hanushevsky, SLAC
<http://xrootd.org>



SLAC



XRootD Plug-in Architecture



Why Plug-ins?

- # Makes it much easier to
 - Adapt, customize, add new features
- # Any cons?
 - Need to know available plug-in points
 - These are documented but not in one spot
 - Described under the relevant directive
 - Usually xxx**lib** (e.g. xrootd.fslib)
 - However, we did make it a bit easier....

The plug-in points

- # A lot and more plug-ins than points!
- # Get a list using **xrdpins** command

```
>xrdpins
Required >= 5.0 @logging
Optional >= 5.0 bwm.policy
Required >= 5.0 cms.perf
Required >= 5.0 cms.vnid
Optional >= 5.0 gsi-authzfun
Optional >= 5.0 gsi-gmapfun
Optional >= 5.0 gsi-vomfun
Required >= 4.8 http.exthandler
Required >= 4.0 http.secextractor
Required >= 5.0 ofs.authlib
Required >= 5.0 ofs.ckslib
Required >= 5.0 ofs.cmslib
Required >= 5.0 ofs.ctllib
Required >= 5.0 ofs.osslib
Required >= 5.0 ofs.preplib
Required >= 5.0 ofs.xattrlib
```

```
Optional >= 5.0 oss.namelib
Required >= 5.0 oss.statlib
Optional >= 5.0 pfc.decisionlib
Required >= 5.0 pss.cachelib
Required >= 5.0 pss.ccmlib
Required >= 5.0 sec.protocol
Required >= 5.0 sec.protocol-gsi
Required >= 5.0 sec.protocol-krb5
Required >= 5.0 sec.protocol-pwd
Required >= 5.0 sec.protocol-sss
Required >= 5.0 sec.protocol-unix
Untested >= 5.0 xrd.protocol
Required >= 5.0 xrdcl.monitor
Required >= 5.0 xrdcl.plugin
Required >= 5.0 xrootd.fslib
Required >= 5.0 xrootd.seclib
```

32 but actual 27

We are missing some
plug-ins of plug-ins.
Need a generalized way
to capture the hierarchy.
(future work)

Plug-in points explained I

@logging	Log message handler (server – cli option)
bwm.policy	Network bandwidth management policy
cms.perf	Performance monitor for cmsd (not script based)
cms.seclib	CMS-specific security plug-in
cms.vnid	Virtual network identifier generator for cms
gsi-authzfun	Specialized gsi authz function
gsi-gmapfun	Specialized gsi gridmap function
gsi-vomsfun	Specialized gsi VOMS function
http.exthandler	HTTP post processing handler
http.secextractor	HTTPS security information extraction
ofs.authlib	Authorization plug-in
ofs.ckslib	Checksum plug-in (individual and manager)
ofs.cmslib	Cluster management service client plug-in
ofs.ctllib	Specialized file system control plug-in
ofs.osslib	Storage system plug-in

Plug-ins points explained II

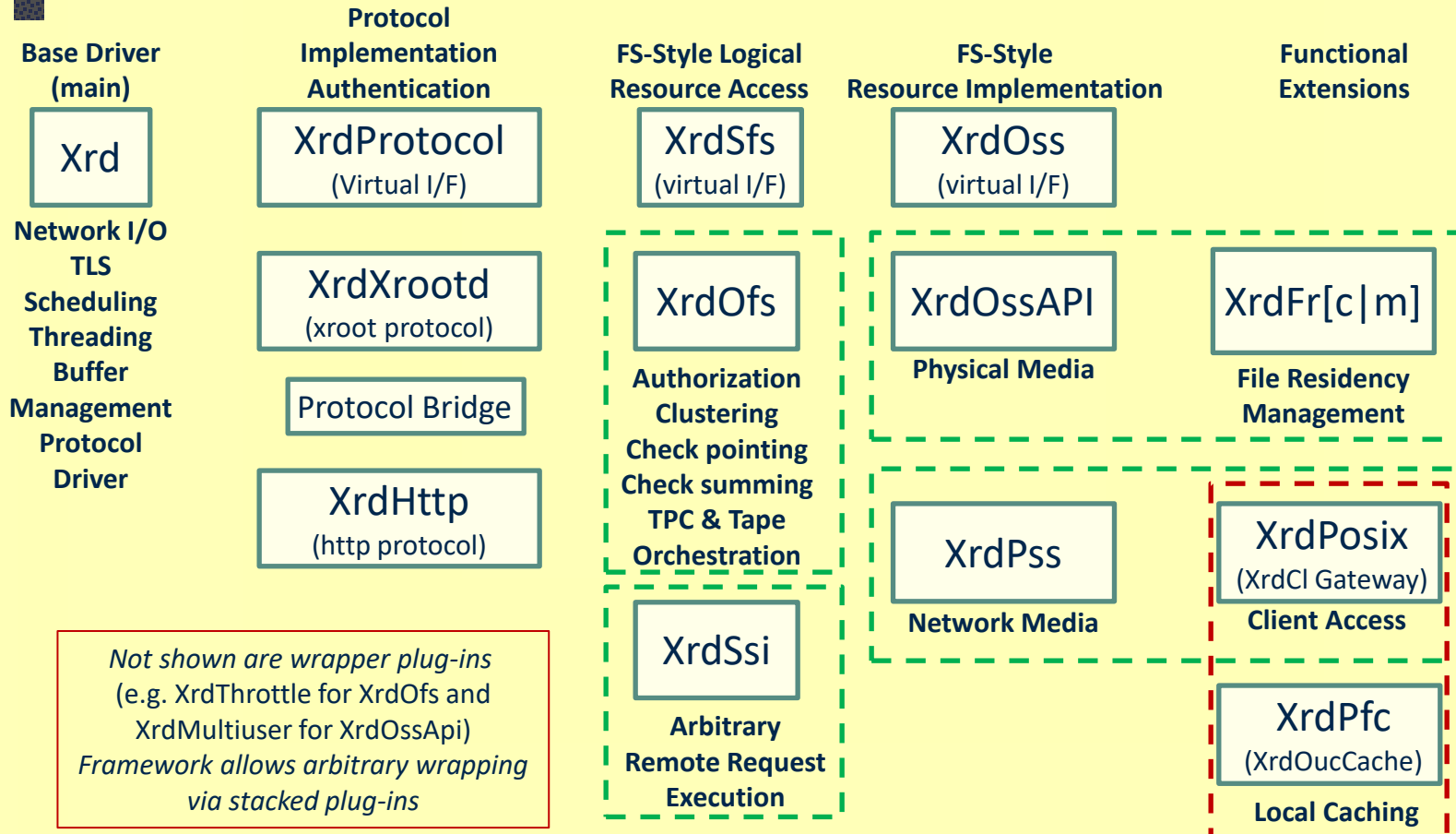
ofs.preplib	Prepare request plug-in
ofs.xattrlib	Extended attribute handler plug-in
oss.namelib	Name mapping plug-in
oss.statlib	Functional stat() plug-in
pfc.decisionlib	Cache purging decision plug-in
pfc.osslib	Cache-specific storage system plug-in
pss.cachelib	Cache implementation plug-in
pss.ccmlib	Cache context management plug-in
pss.namelib	Prosy-specific name-to-name plug-in
sec.protocol	Authentication protocol plug-in (overloaded)
ssi.cmslib	SSI clustering plug-in
ssi.loglib	SSI logging plug-in
ssi.svclib	SSI service implementation plug-in

Plug-ins points explained III

xrd.protocol	Communications protocol plug-in (overloaded)
xrd.tpcmonlib	Connection monitoring plug-in
xrdcl.monitor	Client-side action monitor plug-in
xrdcl.plugin	Client-side API implementation plug-in
xrootd.diglib	Remote debugging plug-in
xrootd.fslib	File system plug-in
xrootd.seclib	Security manager plug-in
ztn.tokenlib	Auth token implementation for ztn protocol

A total of 36 plug-in points

Architectural Plug-in Interplay



Supplied Plug-ins

- # Supplied plug-in have a name format
 - `libXrdname-v.so`
 - *name* is the function name (not always obvious)
 - *v* is the major version number, currently 5
- # General plug-in rules
 - Plug-ins should never be linked against!
 - Never specify the *-v* portion in a config file
 - Doing so defeats automatic version matching
 - It will come back to bite you!

Available Plug-ins I

libXrdAccSciTokens

- SciToken authorization
 - Via ofs.authlib usually stacked

libXrdBwm

- Network Bandwidth Monitor
 - Via xrootd.fslib is now deprecated
 - However serves as a pedagogical example

Available Plug-ins II

libXrdCryptossl

- Internal gsi ssl factory plug-in

libXrdCksCalczcrc32

- Compute zlib crc32
 - Via ofs.ckslib zcrc32

libXrdCmsRedirectLocal

- Redirect client to local file when possible
 - Via ofs.cmslib (is internally stacked)

Available Plug-ins III

libXrdHttp

- http[s] protocol
 - Via xrd.protocol

libXrdHttpTPC

- http[s] third party copy
 - Via http.exthandler

libXrdMacaroon

- Macaroon authorization (https only)
 - Via http.exthandler & ofs.authlib

Available Plug-ins IV

libXrdN2No2p

- Object id support using Posix file systems
 - Via oss.namelib
 - This plug-in has never really been used

libXrdOfsPrepGPI

- General purpose prepare coordinator
 - Via ofs.preplib

Available Plug-ins V

libXrdOssCsi

- Integrity at rest file system support
 - Via ofs.osslib or pfc.osslib (must be stacked)

libXrdOssSIgpfsT

- Offline file support for gpfs
 - Via oss.statlib

Available Plug-ins VI

libXrdPfc

- Xcache implementation
 - Via pss.cachelib
- libXrdBlacklistDecision
 - Caching for files not on a black list
 - Via pfc.decisionlib

Available Plug-ins VII

libXrdPss

- Proxy server implementation
 - Via ofs.osslib

libXrdSec

- Security framework (usually not replaced)
 - Via xrootd.seclib using “default”

libXrdSecProt

- Request signing implementation
 - Via sec.level (internally loaded when specified)

Available Plug-ins VIII

libXrdSecgsi

- Gsi authentication

- Via sec.protocol gsi

- libXrdSecgsiAUTHZVO

- Gsi authz funtion via **-authzfun** gsi parameter

- libXrdSecgsiGMAPDN

- Gsi gridmap function via **-gmapfun** gsi parmeter

Available Plug-ins IX

libXrdSeckrb5

- Kerberos V authentication
 - Via sec.protocol krb5

libXrdSecpwd

- Password authentication
 - Via sec.protocol pwd

libXrdSecsss

- Simple Shared Secret authentication
 - Via sec.protocol sss

Available Plug-ins X

libXrdSecunix

- Old NFS-style authentication
 - Via sec.protocol unix

libXrdSecztn

- Token (SciToken or WLCG style) firewall
 - Via sec.protocol ztn

Available Plug-ins XI

libXrdSsi

- Scalable Service Interface implementation
 - Via xroot.fslib
- libXrdSsiLog
 - Logging extensions for SSI
 - Via command line `-l@pluginlib` option

libXrdVoms

- Virtual Organization Management System
 - Via `gsi -vomsfun` parameter and `http. secextractor`

Available Plug-ins XII

libXrdXrootd

- Xroot protocol implementation
 - Via xrd.protocol (unnecessary)
 - Note this protocol is always loaded

What about client plug-ins?

- We don't cover these
 - libXrdClProxyPlugin
 - libXrdClRecorder
- And a few other possible ones in this talk.

Plug-in Stacking

- # Many plug-ins can be stacked
 - When stacked a plug-in can hand off work
 - Case 1: Preprocess hand off dirty work to next PI
 - Case 2: Prescreen and handle or hand off
 - Typically uses
 - Multi-mode authorization
 - New features for old storage systems
- # Stacking uses a common specification
 - *xxx.yyylib ++ rest_of_parms*

Writing plug-ins

- # Requirements are documented in hh files
 - Most plug-ins have a common boilerplate
 - How they are loaded, stacked, versioned
 - Server side plug-ins are never unloaded
 - Loading always happens at start-up time
- # Just remember if you want an enhancement
 - First see if you can write a plug-in for it

Major Release 6 **Warning!**

- # You may need to do source changes
 - XrdOucString.hh & XrdSysPlatform.hh issue
 - Contained “using namespace std” Yikes!
 - Will be removed but severe implications
 - Source changes needed if you implicitly relied on this
 - Involves any file that also included ...
 - XrdOfsTrace.hh XrdOssApi.hh XrdOucGMap.hh
XrdOucTrace.hh XrdSutAux.hh
 - Unfortunately most XrdCrypto...hh
 - Next major release is 6.0.0 this year

Conclusion

XRootD plug-in architecture rocks!

- Unfortunately, there is a lot there
 - This presentation merely introduced you to it
 - We are working on making this less daunting

Our core partners



Community & funding partners *(not a complete list)*



Funding from US Department of Energy contract DE-AC02-76SF00515 with Stanford University