OpenAFS' Road to IPv6

Andrew Deason

Sine Nomine Associates

European AFS & Kerberos Conference 2014

It’s not as hard as it sounds!

Trimming down tasks

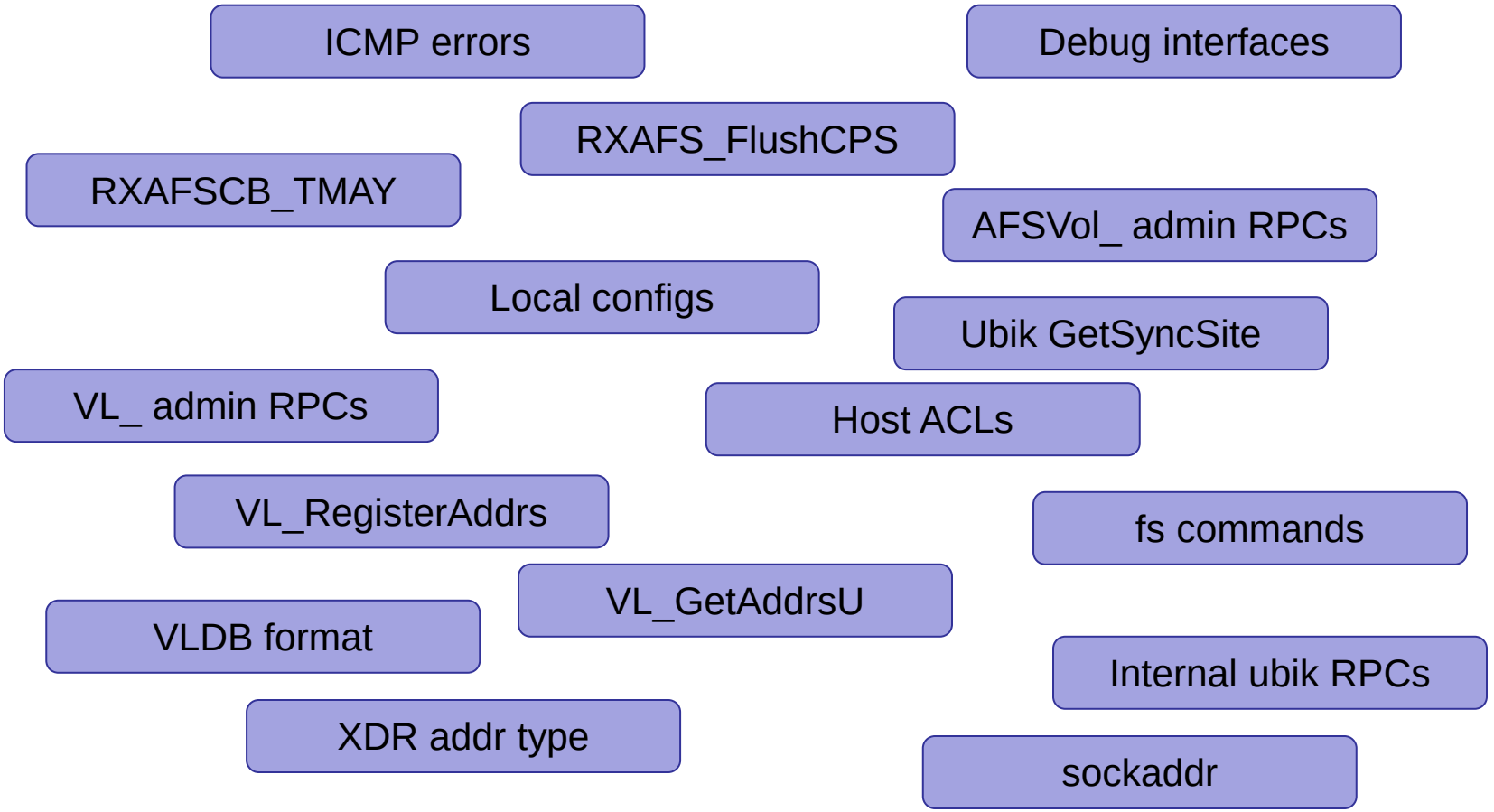
SINE NOMINE ASSOCIATES

- Varying levels of “need to do”
  - Debug interfaces, optional features, …
- Varying levels of difficulty
  - Design vs “just replace v4 addr with v6”
- Varying levels of stability
  - Standards vs internal

General plan

- First, just focus on basic client and server, assume v4/v6
- Then, all of the “we can test” sites can test
- Now we have momentum for everything else

IPv6 Tasks

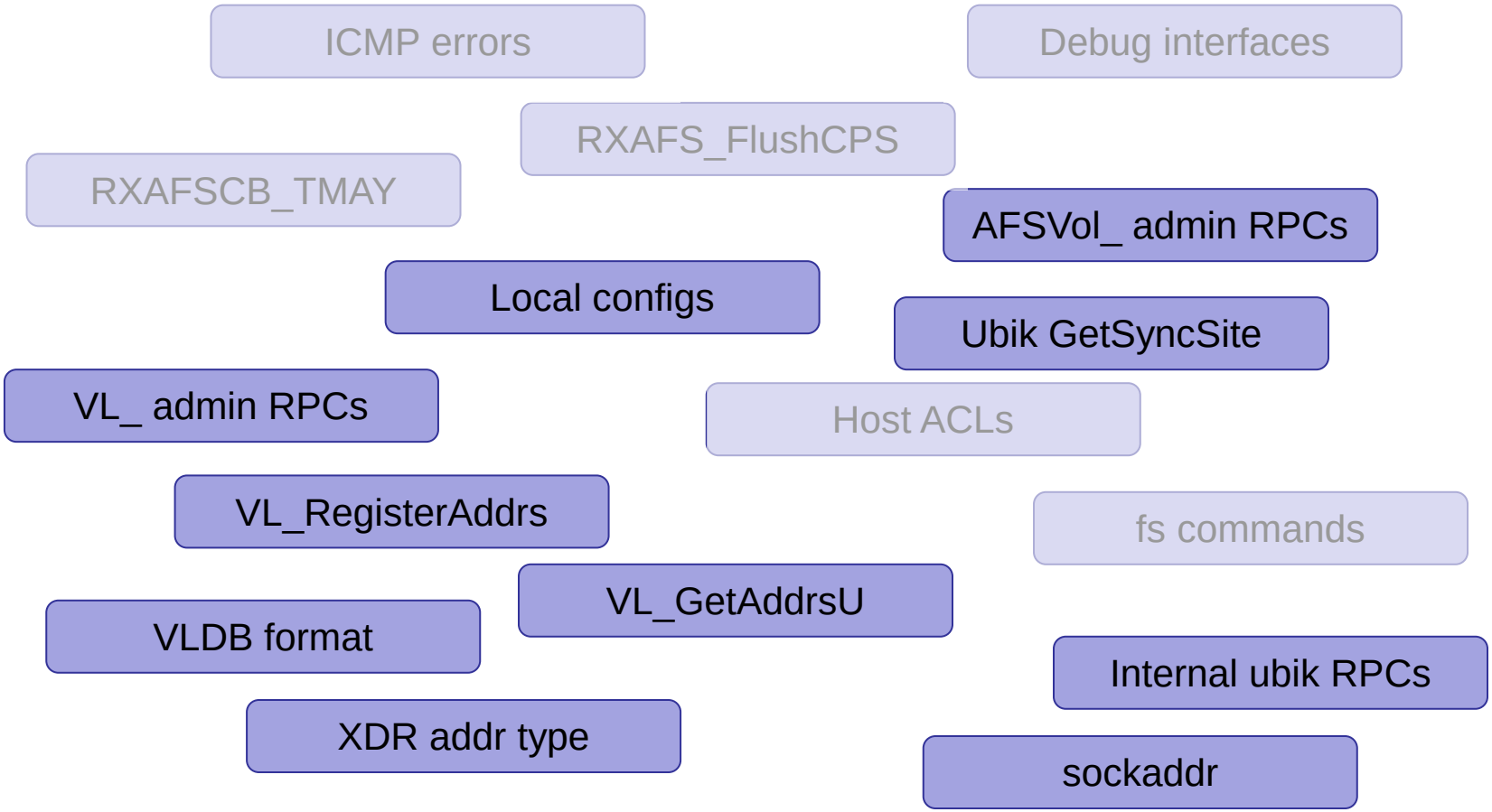

ICMP errors
Debug interfaces
RXAFS_FlushCPS
RXAFSCB_TMAY
AFSVol_ admin RPCs
Local configs
Ubik GetSyncSite
VL_ admin RPCs
Host ACLs
VL_RegisterAddrs
fs commands
VL_GetAddrsU
VLDB format
Internal ubik RPCs
XDR addr type
sockaddr

“Less necessary” Tasks

SINE NOMINE ASSOCIATES

- Only do what we need for a client to access files in AFS
- Some obviously optional
  - AFSCB_TMAY, just get rid of addresses
- Some helpful, but not required

IPv6 Tasks (less necessary)

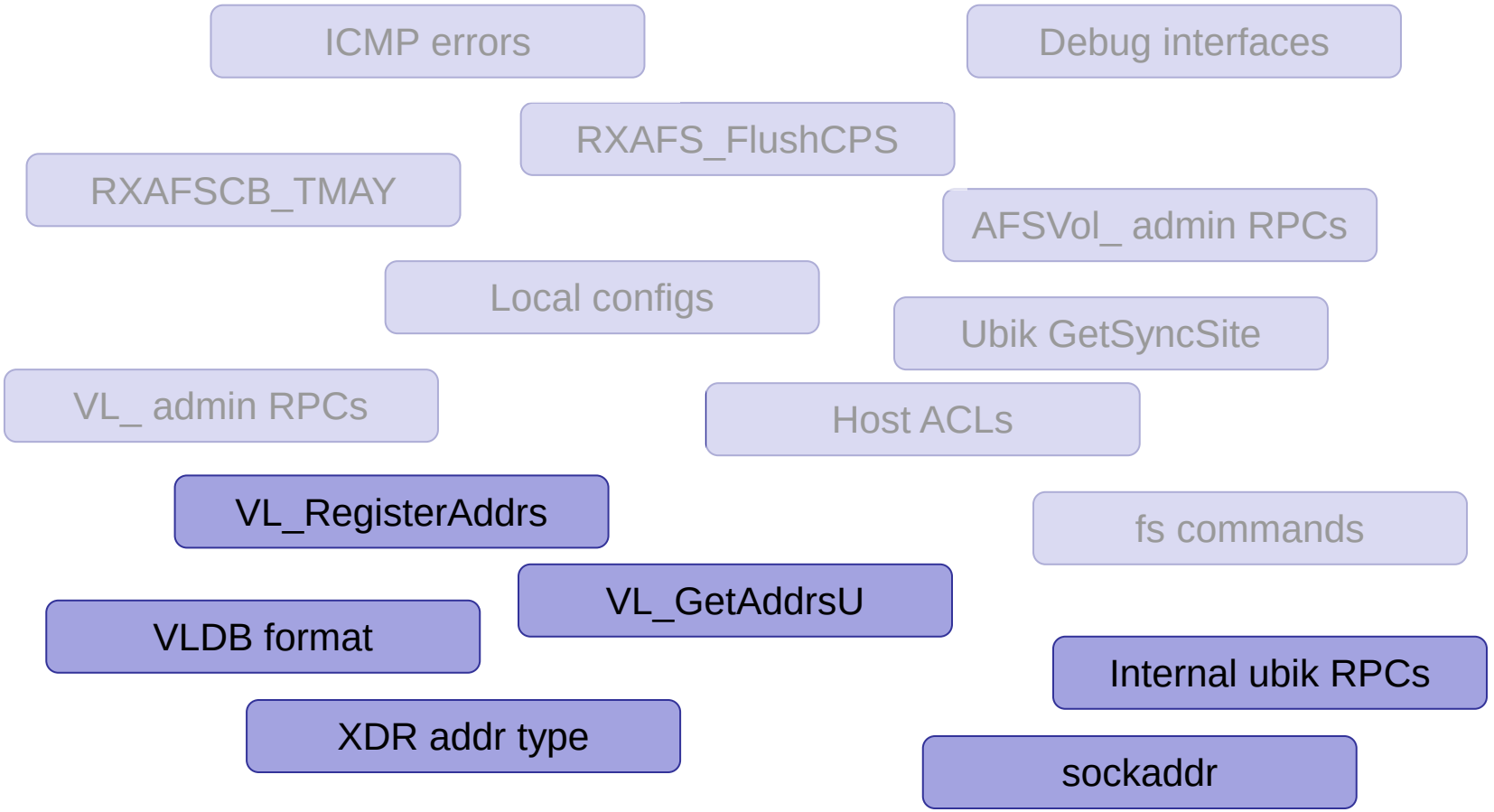


“v4/v6 servers”

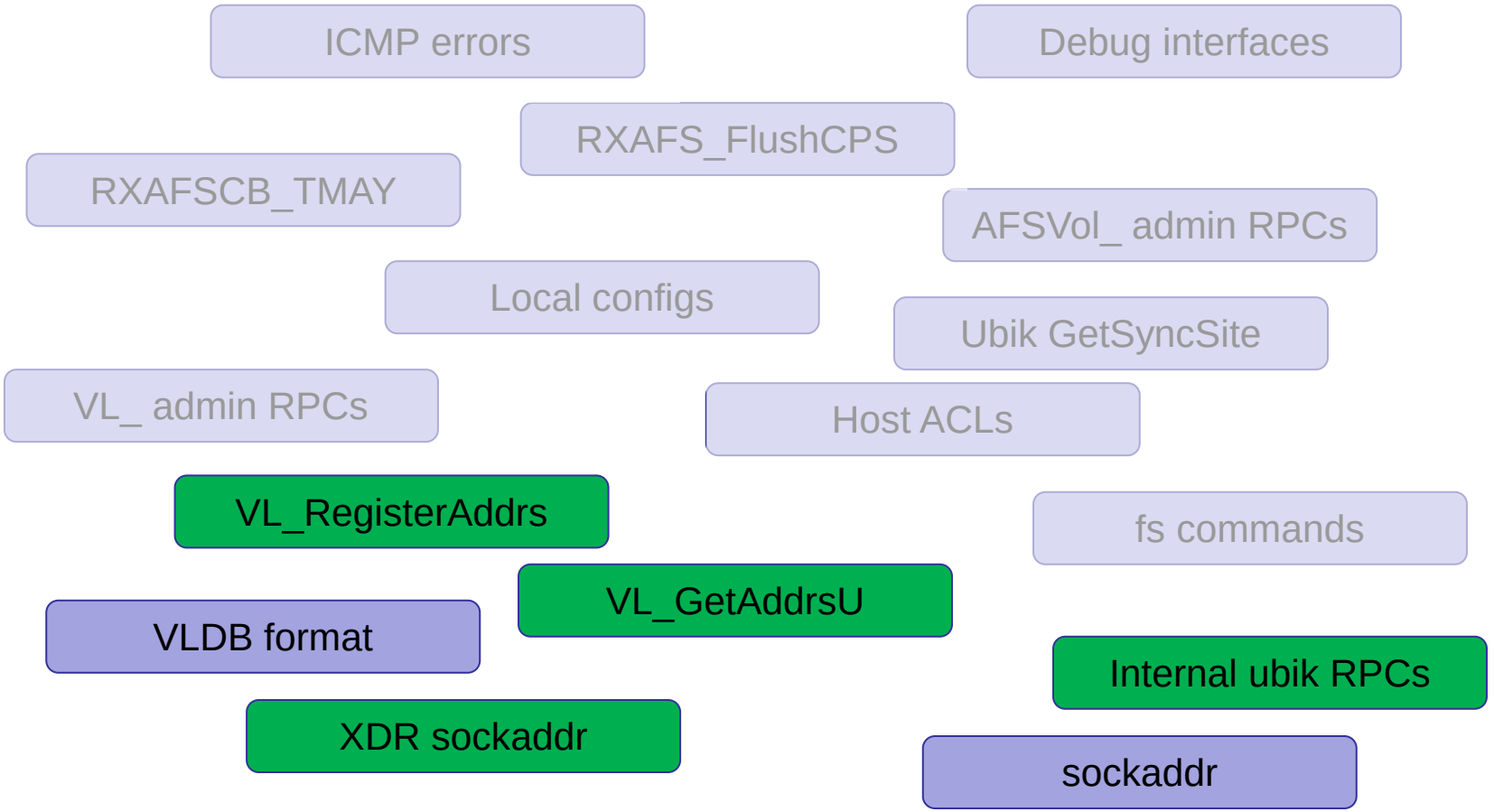
SINE NOMINE ASSOCIATES

- Assume all servers have at least one ipv4 address
- Simplifies many administrative RPCs
- “okay” for existing sites, maybe not for new sites

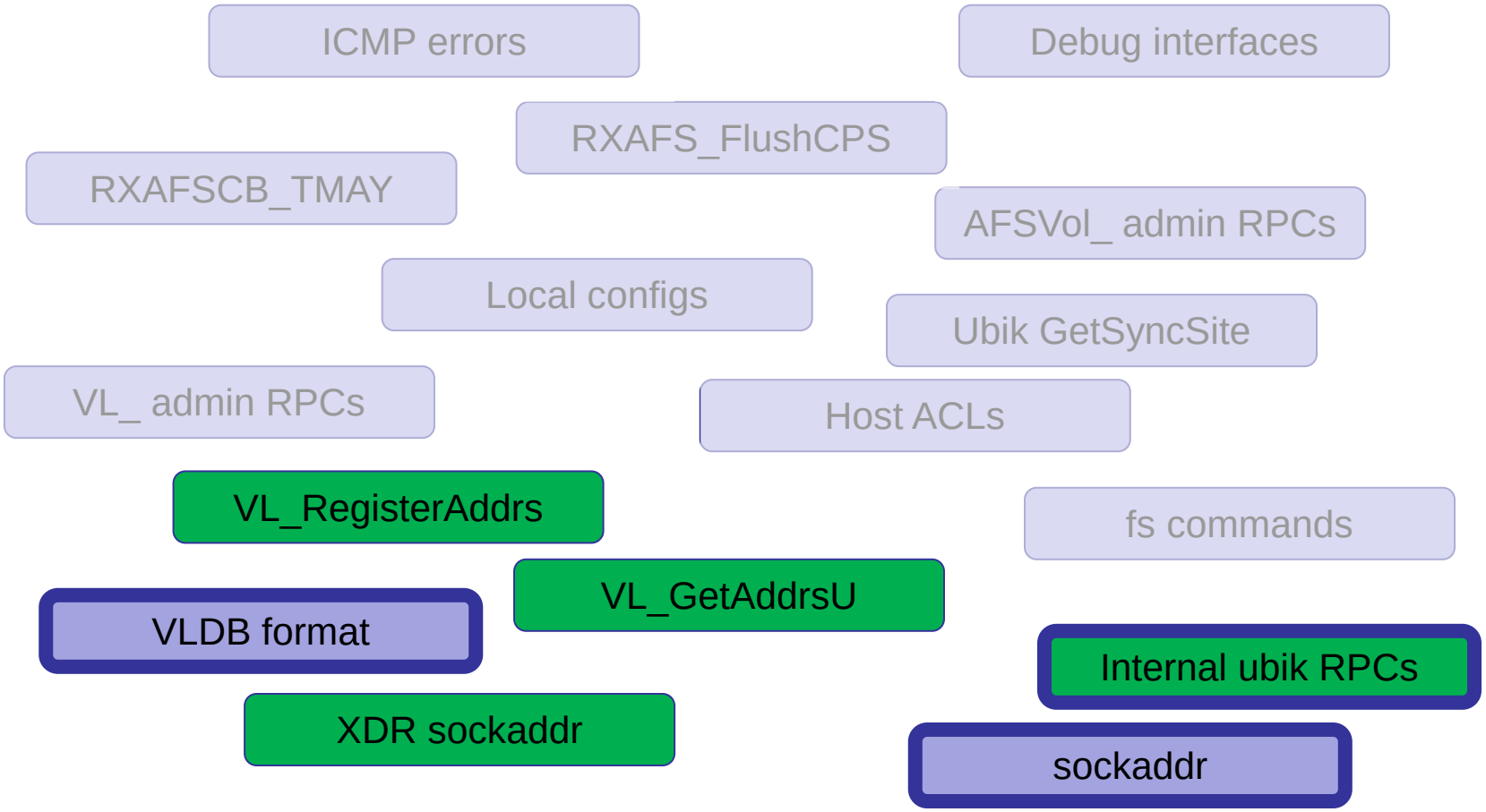
IPv6 Tasks (v4/v6 servers)



IPv6 Tasks (easy/consensus)



IPv6 Tasks (no standards)



IPv6 Tasks

SINE NOMINE ASSOCIATES

- "Easy" doesn't mean "trivial"

- Many tasks have at least 3 solutions:
  - The "wrong" way (kluges)
  - The "right" way
  - The practical way (middle ground)

RPCs

- Prereq: ext-union
- draft-keiser-afs3-xdr-union-06
- General agreement, but needs implementation

RPCs

- XDR sockaddr
  - Wrong: opaque[16]
  - Right: nested ext-union
  - Practical: ext-union
- RPCs
  - Wrong: opaque[16]
  - Right: special vector types
  - Practical: arrays of the XDR sockaddr

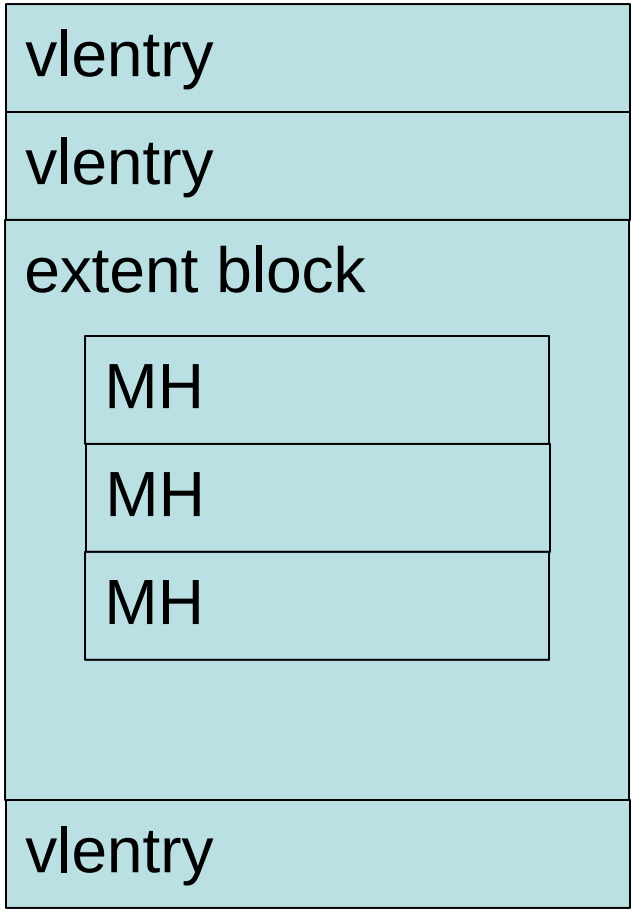
sockaddr

- Wrong: 128-bit storage
- Right/Practical: sockaddr or new struct
- Usually “easy” but tedious
- Tedious
- Tedious

VLDB

- Wrong: stuff v6 addrs in existing v4
- Right: completely new db
- Practical: new address records in extent blocks

Extent blocks



Extent blocks

- Currently for MH (multi-homed) records
- Unreferenced data is ignored
  - …as long as we VLCONTBLOCK
- Anything at all can go in here

Extent blocks

- Add any new arbitrary data, and existing vlservers don’t break
- Entire new db, broken up by 8k blocks
- Speed not critical

Extent blocks

- What to put in them? XDR addrs
  - Or opaque[16]s if you want to be “wrong”
- Maybe more extensible/arbitrary server data
- One block per server

Beyond

- With that done, move on to v6-only
- Debugging, may be useful during dev
  - rxdebug, maybe temporary solution
- Optional features as needed