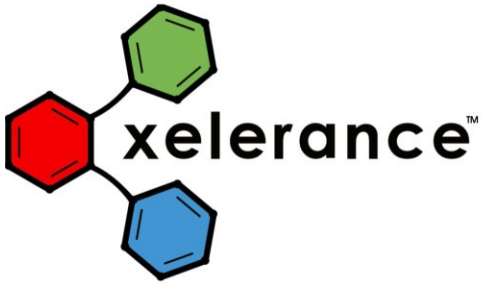
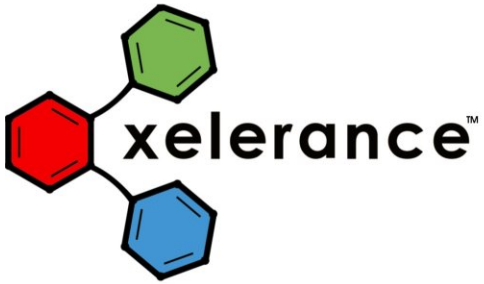


OCF level 2

CryptoAPI and Crypto frameworks – hardware offload

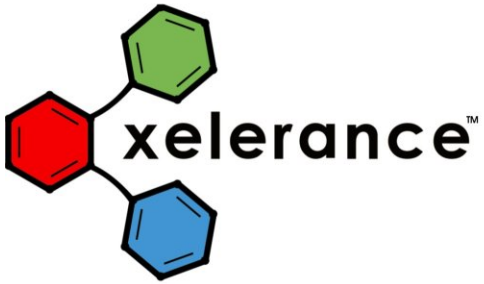
convenor

Michael Richardson
<mcr@xelerance.com>



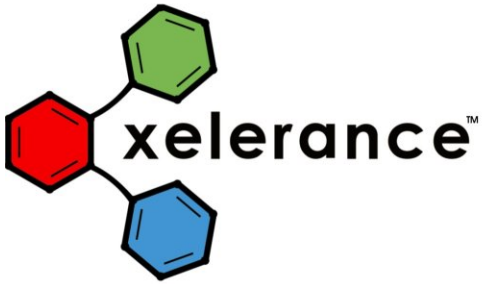
Agenda

- who is who/why do I care?
- how we got here
- where we propose to go
- requirements
- issues



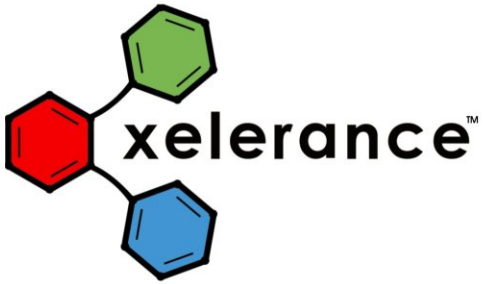
Who am I?

- Xelerance is open source company formed from parts of former FreeS/WAN team
- provides 3rd level defect support, and enhancement to Openswan on Linux
- under contract to Hifn to provide a better open source story for Vulcan and HIPP-I/II product line



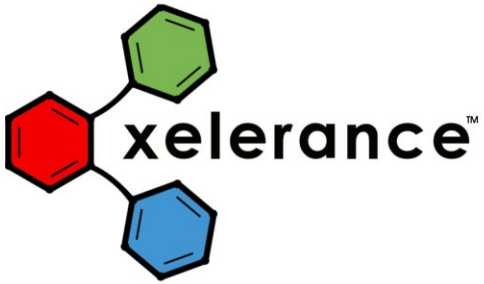
How we got here

- NRL IPsec 1993-1996 for BSDi/*BSD
- FreeS/WAN 1.x/2.x from 1996-2004 (crypto-war politics)
- KAME/BSD code 1997-present, in *BSD.
- OpenBSD imports parts of NRL IPsec, improves it, adding OpenBSD Cryptographic Framework
- Xelerance Openswan fork of FreeS/WAN 2.04, in 2003.



How We got here (2)

- CryptoAPI 0.1 used by disk encryptors from 1999-2003.
- Linux 2.6 native xfrm code committed just before October 31 (2003) 2.6 feature freeze, includes “new” CryptoAPI 1.0FreeBSD imports OpenBSD Cryptographic Framework (OCFl1), improves it (2004)
- Snapgear (David Mccullough) ports OCF to Linux, adapts KLIPS to use it (2005)
- Xelerance imports OCF code to KLIPS (2006Q2).

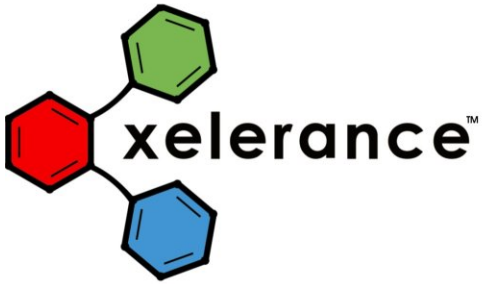


Why something new?

OCF level 1 (my term) deals with offload of specific algorithms (AES, 3DES, HMAC-SHA1), but not with the entire packet transform.

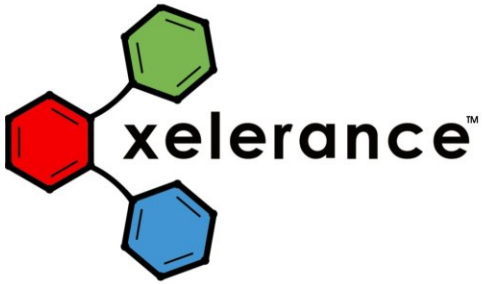
It does this asynchronously, but is not yet integrated into Linux.

There is also an effort called **acrypto**.



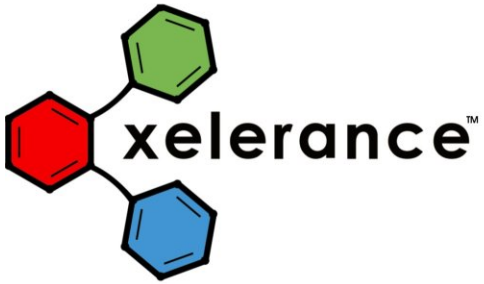
OCF level 2

- newer cryptographic hardware can do the entire ESP encrypt or decrypt
- distinguish between look-aside (DMA bus-mastering) vs inline (contained in NIC card)
  - inline efforts out of scope for now.



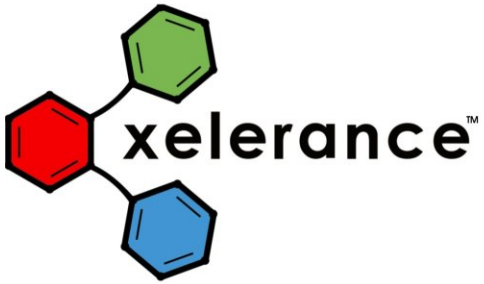
Discussion list

- ocf2-discuss@hifn.xelerance.com
- join by visiting
  https://hifn.xelerance.com/mailman/listinfo



Requirements

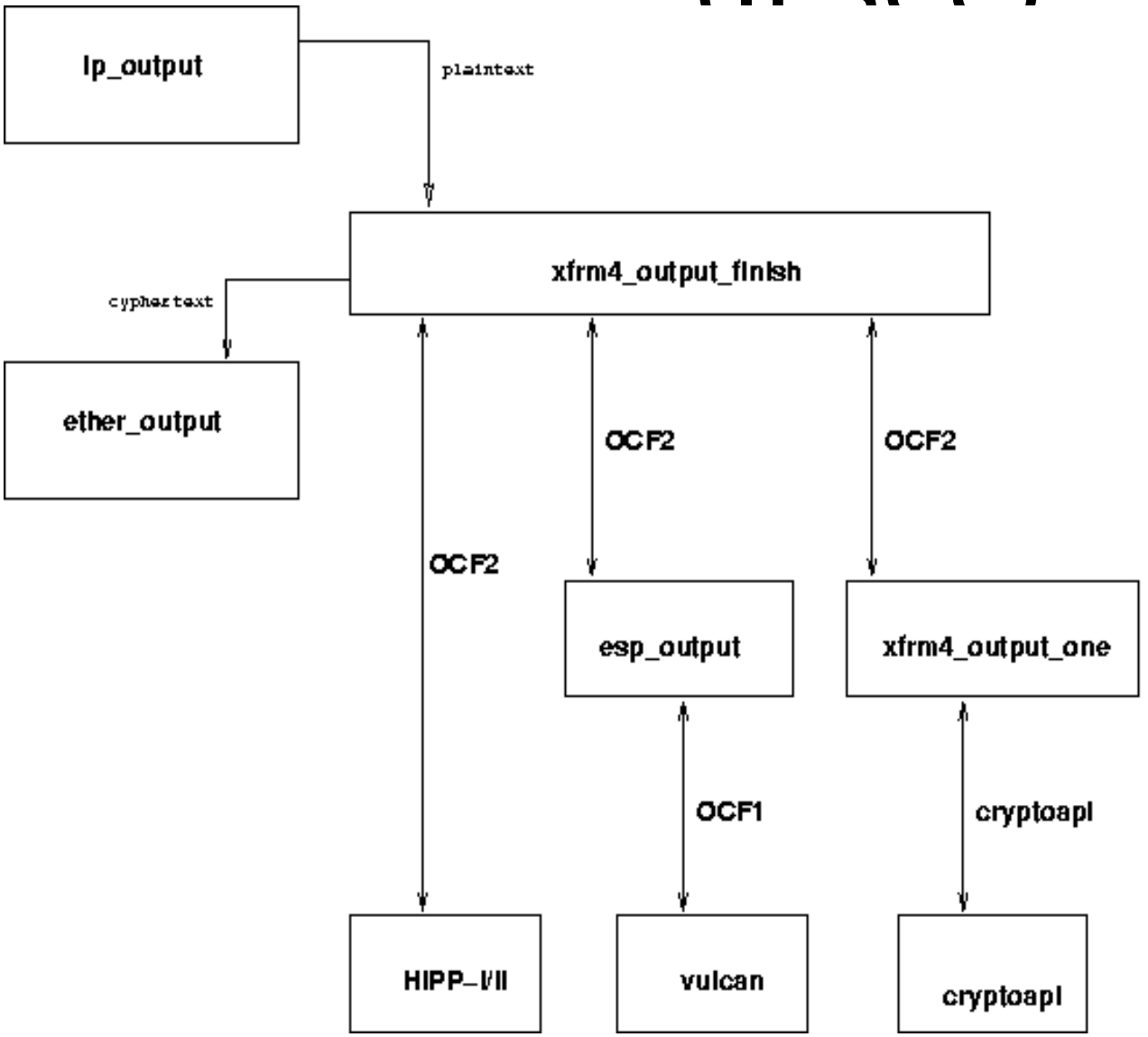
- maintain synchronous cryptoapi.
  - Keep it simple to use for existing uses
  - Keep it simple to test/debug
- be able to batch operations on a single packet: crypto, hmac, compress, other?
  - maybe codec as well.
- support software-only case (99% of users), and do it as efficiently as if there was no hardware offload



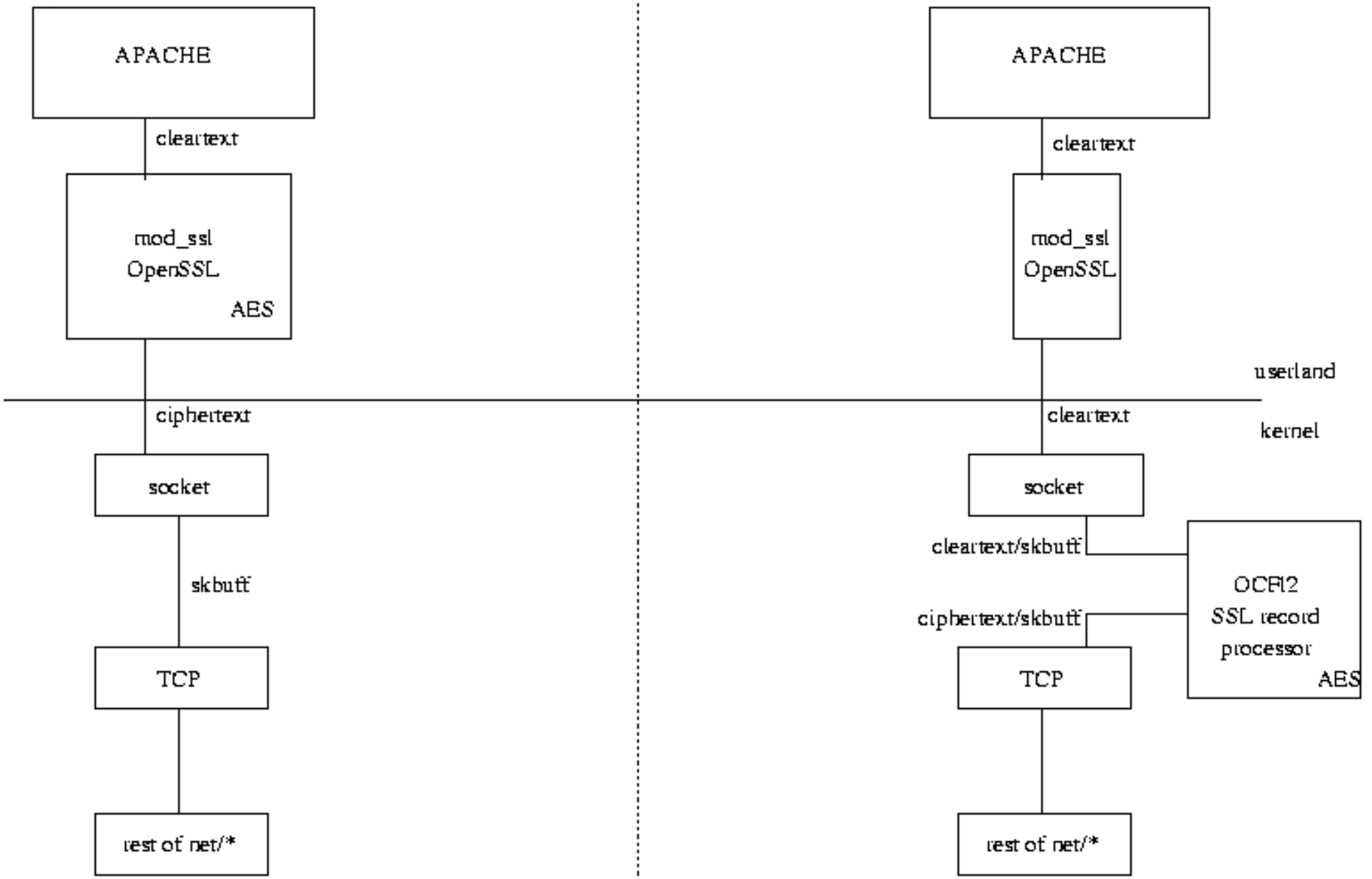
Requirements (2)

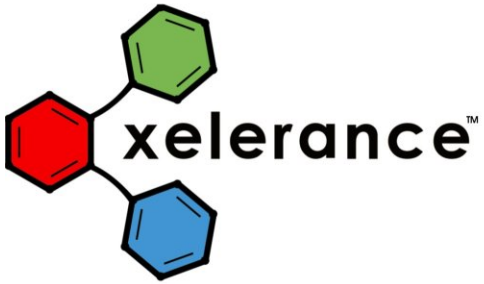
- packet offload for IPsec (KLIPS and 2.6 “native” /NETKEY)
- algorithm offload for disk encryptors
- permit future SSL record mode in kernel
-

OCF level 2 diagram (IPsec)



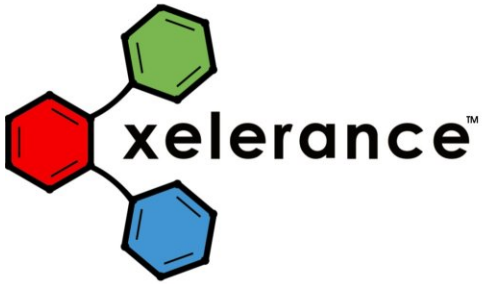
OCF level 2 diagram (SSL)





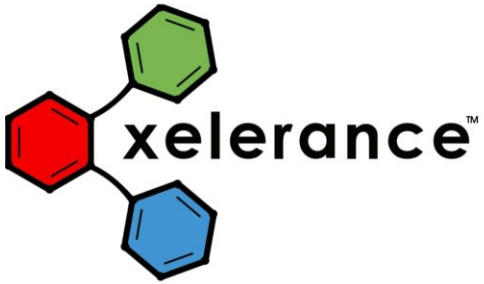
Issues

- keeping latency low
- how to best use multiprocessors
- DMA access to memory is often so bad as to make hardware irrelevant
- binary only drivers: how much support to provide/permit?



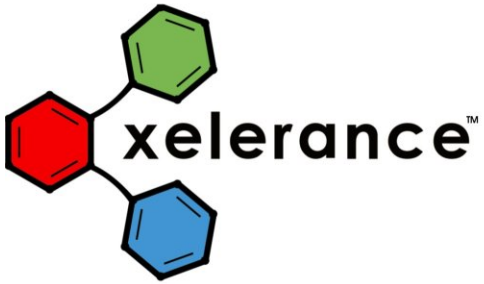
Opportunities

- using extra processors to do crypto work
- using custom things (DRC?) do help, at hypertransport rates
-



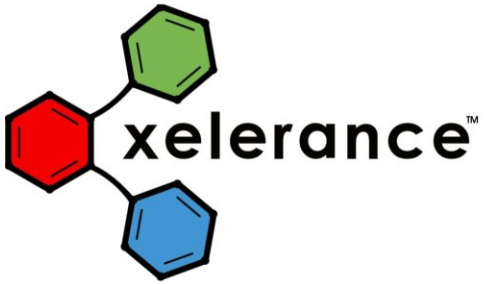
KLIPS and NETKEY ("2.6 native") merge

- many prefer KLIPS style of packet classification, diagnostic interfaces, and firewall interaction
- insides of KLIPS sucks
- replace insides of KLIPS with calls to XFRM code



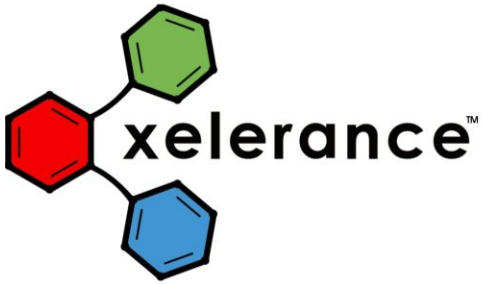
KLIPS and OCF1

- existing OCF1 code is patch to KLIPS to use algorithm accelerator
- refactor this into an OCF2 machine



Work to date

- OCFl1 merge (from snapgear)
- Version of KLIPS that uses xfrm code done



Work to do

- version of xfrm packet classifier that talks to OCF2
-