API, Virtio Messages for Virtio-ipsec-LA

Subha Venkataramanan, Srini Addepalli

- g-API Guidelines
- Virtio-ipsec-la
  - Categorization
  - Packet Flow
- g-APIs for IPsec
- sio Interface
  - Virtio IPsec default device definition
  - Virtio Messages for IPsec
- Relevant documents
- Architecture/PoC

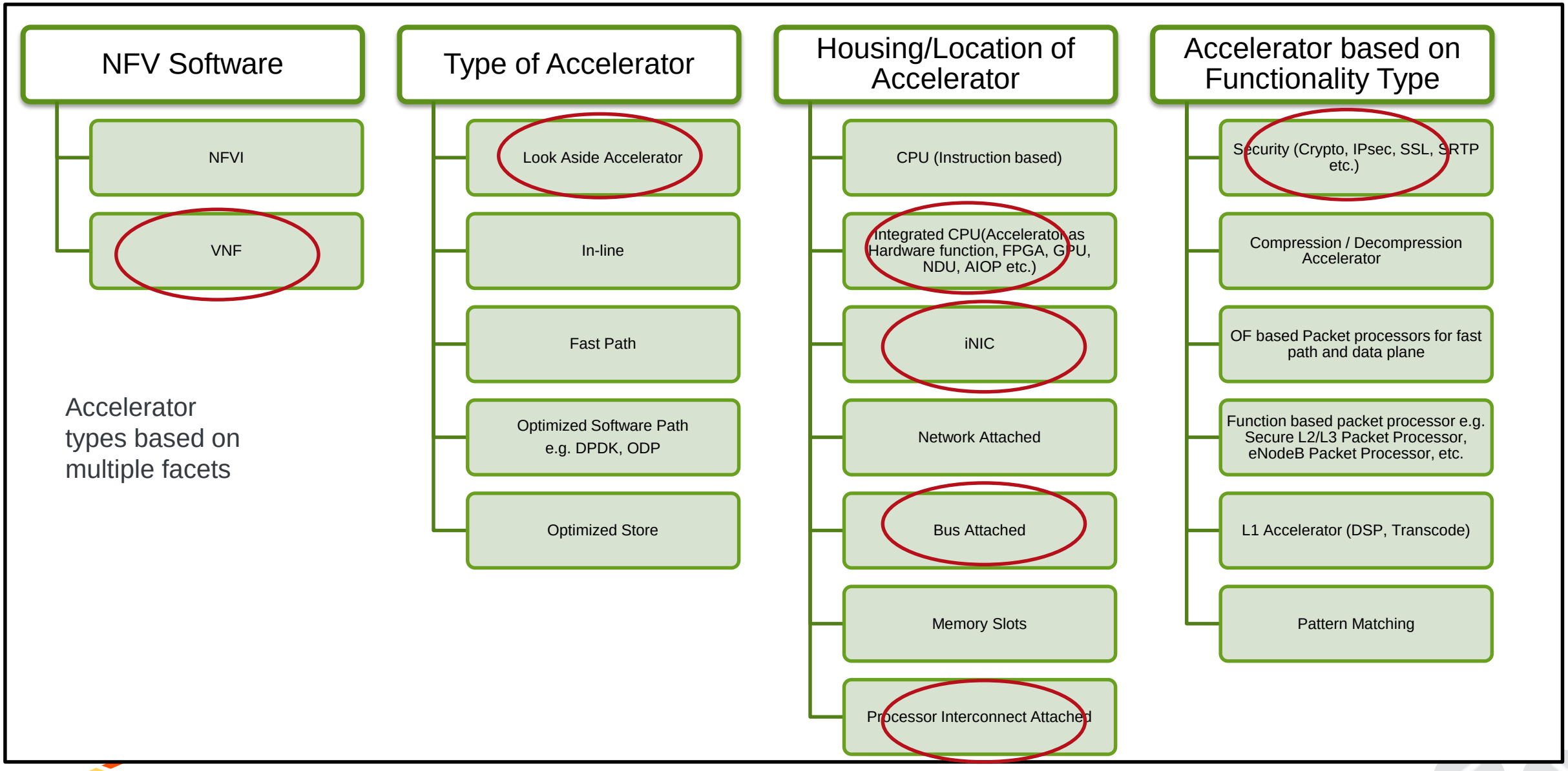
g-API Guidelines

C-functions	• APIs to be defined as C callable functions
Arguments as structures	• By defining arguments as structures, it is possible to add parameters passed easily without changing API definitions. • Allows for APIs to be extended with minimal code changes. • ***Not Applicable for Packet processing APIs***
Return Value	• Must return success or failure • May provide additional error details in the case of failure

API Flags	• Synchronous or Asynchronous.
	  • In case asynchronous mode is requested, calling application should provide the callback function and argument that can be called by the API layer later when the response is ready.
	• Response Required or not
	  • In some cases, there may be additional steps that the API layer needs to do, to force a response from the underlying virtual accelerator. (e.g: Openflow)

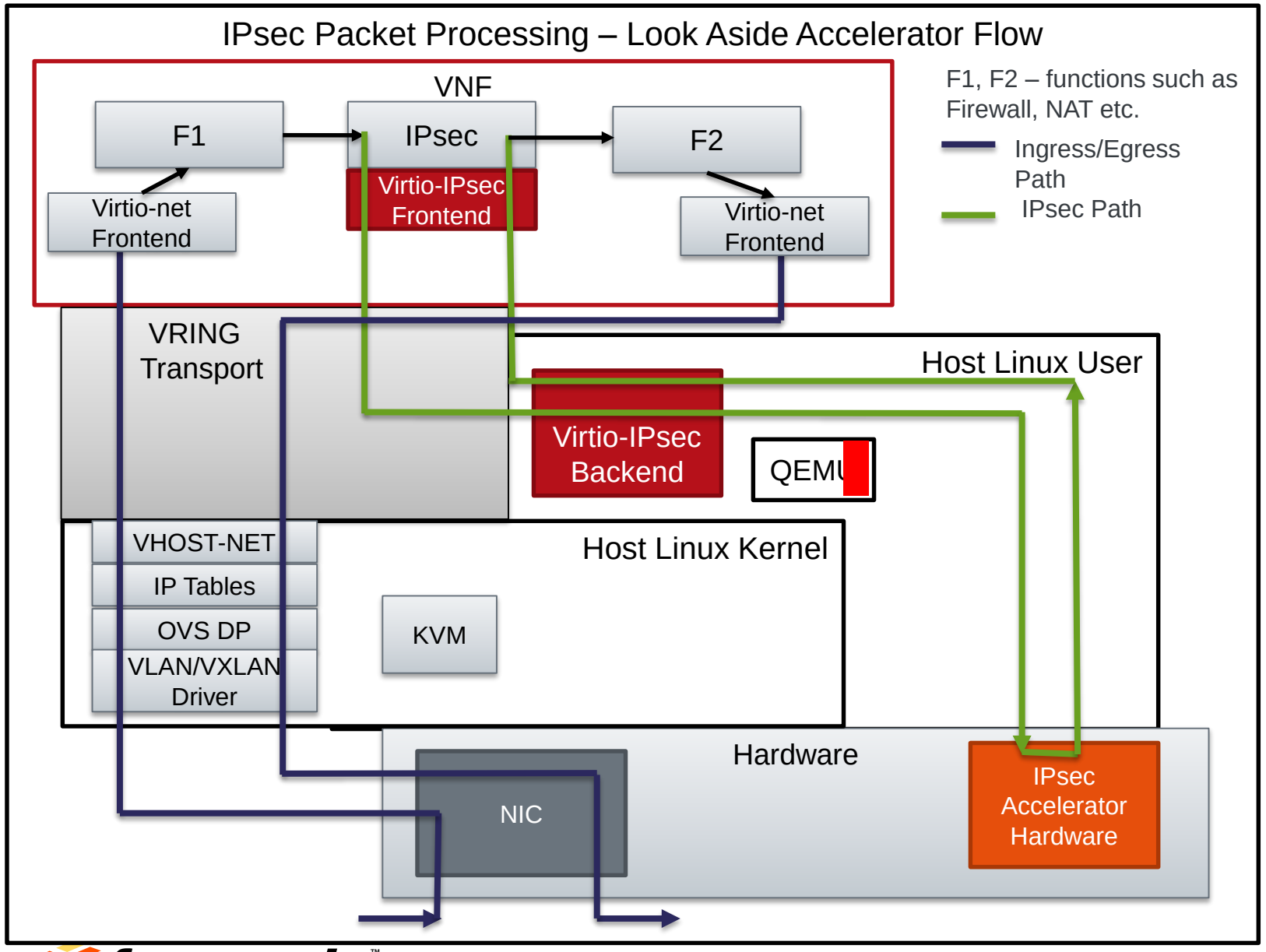
API Naming	• g_”function”_”type”_”object”_”action”. • For e.g., an IPsec look aside accelerator SA add function would be named as g_ipsec_la_sa_add()
Variable Naming	• Linux style of naming convention to be followed
Types	• Linux standard types to be used for data structures

Virtio-ipsec-la



Note: Classification is Freescale Contribution to ETSI IFA-001 (Status: Accepted)

Virtio-ipsec-la Packet Flow



- More than Crypto
  - Algorithm Processing
  - Protocol header processing
    - Encapsulation
    - De-capsulation
    - ESP, AH
    - Anti-replay check
  - Checksum computation

g-API for IPSec

Data API

- **g_ipsec_la_packet_encap()**
  - Send a packet for encapsulation
- **g_ipsec_la_packet_decap()**
  - Send a packet for decapsulation
- **g_ipsec_la_mult_packet_encap()**
  - Send multiple packets for encapsulation
- **g_ipsec_la_multi_packet_decap()**
  - Send multiple packets for decapsulation

Control API

- **g_ipsec_la_capabilities_get()**
  - Get the capabilities of the underlying devices
- **g_ipsec_la_sa_add()**
  - Add SA
- **g_ipsec_la_sa_del()**
  - Delete SA
- **g_ipsec_la_sa_mod()**
  - Modify SA
- **g_ipsec_la_sa_flush()**
  - Flush SA
- **g_ipsec_la_sa_get()**
  - Read and Traversal SA
- **g_ipsec_la_notifications_hook_register()**
  - Register hooks for optional notifications such as Sequence number overflow or lifetime in kilobytes expiry etc.

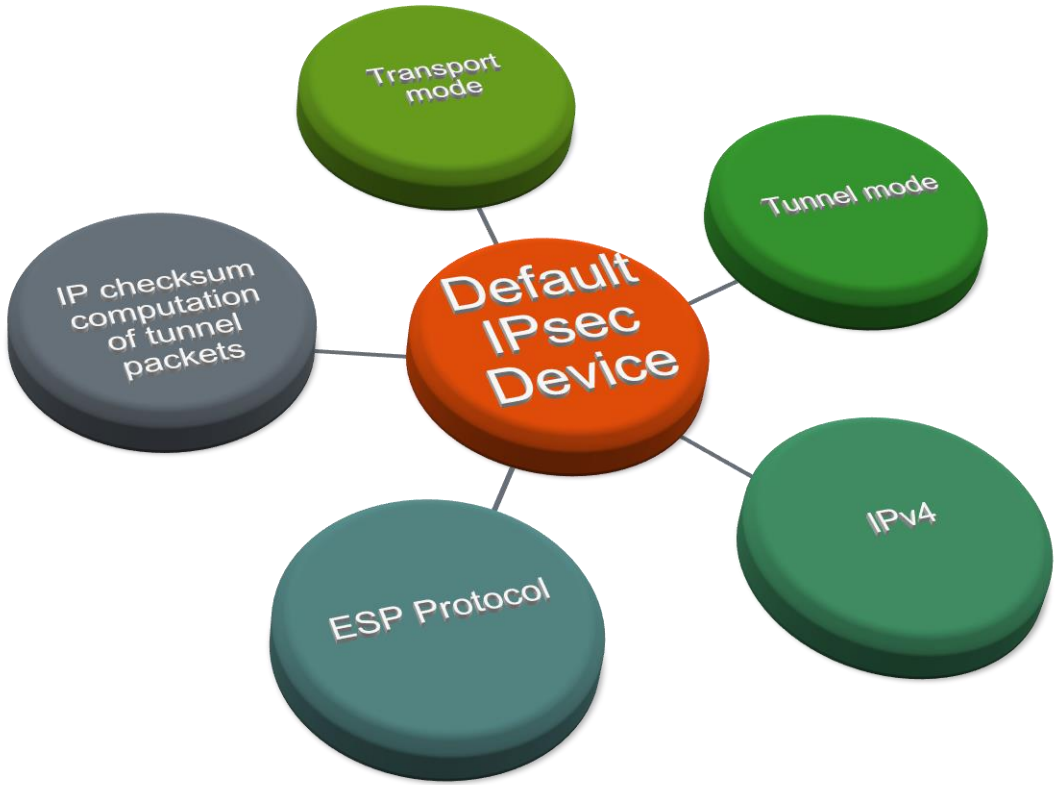
Management API

- **g_ipsec_la_get_api_version()**
  - Get the API version
- **g_ipsec_la_avail_devices_getinfo()**
  - Get the information on available devices
- **g_ipsec_la_active_devices_getinfo()**
  - Get the information on active devices
- **g_ipsec_la_open()**
  - Open a device
- **g_ipsec_la_close()**
  - Close a device
- **g_ipsec_la_group_create()**
  - Create a logical group for grouping SAs
- **g_ipsec_la_group_delete()**
  - Delete a logical group

DPACC Requirements/IPsec API mapping

#	DPACC Requirement	Compatibility	Comments
1	MUST be as high a performing design as possible	✓	No structures in the Data processing APIs
2	MUST provide portability for the applications (Both Source and Binary) - Source code portability across CPU architectures Binary portability within a CPU architecture (vNF VM binary in case of QEMU and vNF container binary in case of Linux containers).	✓	APIs don't use any particular implementation based structures and all APIs are normal C functions. All tunable parameters are queried and informed using API functions.
3	MUST be scalability in performance and design	✓	Supports any number of accelerators exposed by host Linux Tunable for high capacity systems
4	MUST be written in a portable language	✓	Implemented in 'C'
5	MUST support legacy VNFs	✓	
6	MUST supply all code within the DPACC design to be open sourced	✓	In the works
7	MUST NOT supply code in binary form with only one exception	✓	Code to be submitted in source form
8	MUST NOT use non-upstreamed host kernel modules or modifications for core DPACC system	✓	Yes. No host Kernel Changes or Modules
9	MUST document the API and code with Doxygen	✓	Will be made available when code is submitted

sio (virtio) Interface – Ipsec-LA Device



Registers

- Number of Version Registers
- Version 1
- Version 2
- Version n
- Guest's Preferred Version
- Device Queue Information
- Guest's Selected Queues

Queues

- Control Queue
- De-capsulation Qs [1..n]
- Encapsulation Qs[1..n]
- Notification Queue (Optional)

Feature Bits

- VIRTIO_IPSEC_F_SG_BUFFERS
- VIRTIO_IPSEC_F_AH
- VIRTIO_IPSEC_F_WESP
- VIRTIO_IPSEC_F_SA_BUNDLES
- VIRTIO_IPSEC_F_UDP_ENCAPSULATION
- VIRTIO_IPSEC_F_TFC
- VIRTIO_IPSEC_F_ESN
- VIRTIO_IPSEC_F_ECN
- VIRTIO_IPSEC_F_DF
- VIRTIO_IPSEC_F_ANTI_REPLAY_CHECK
- VIRTIO_IPSEC_IPV6_SUPPORT
- VIRTIO_IPSEC_F_SOFT_LIFETIME_BYTES_NOTIFY
- VIRTIO_IPSEC_F_SEQNUM_OVERFLOW_NOTIFY
- VIRTIO_IPSEC_F_SEQNUM_PERIODIC_NOTIFY
- VIRTIO_RING_F_INDIRECT_DESC
- VIRTIO_RING_F_EVENT_IDX

sio Interface
Virtio Messages

VIRTIO_IPSEC_DATA_GENERIC

- **VIRTIO_IPSEC_SEPERATE_INPUT_OUTPUT_BUFFERS**
  - Input provided in separate buffers and processed output expected in different buffers
- **VIRTIO_IPSEC_IN_MEMORY_BUFFER_REPLACEMENT**
  - Input data in the input buffers will be replaced with processed data

VIRTIO_IPSEC_CTRL_GENERIC

- **VIRTIO_IPSEC_CTRL_GET_CAPABILITIES**
  - Get capabilities
- **VIRTIO_IPSEC_CTRL_SET_CAPABILITIES**
  - Set capabilities
- **VIRTIO_IPSEC_CTRL_SET_GUEST_ENDIAN**
  - Set the guest endian

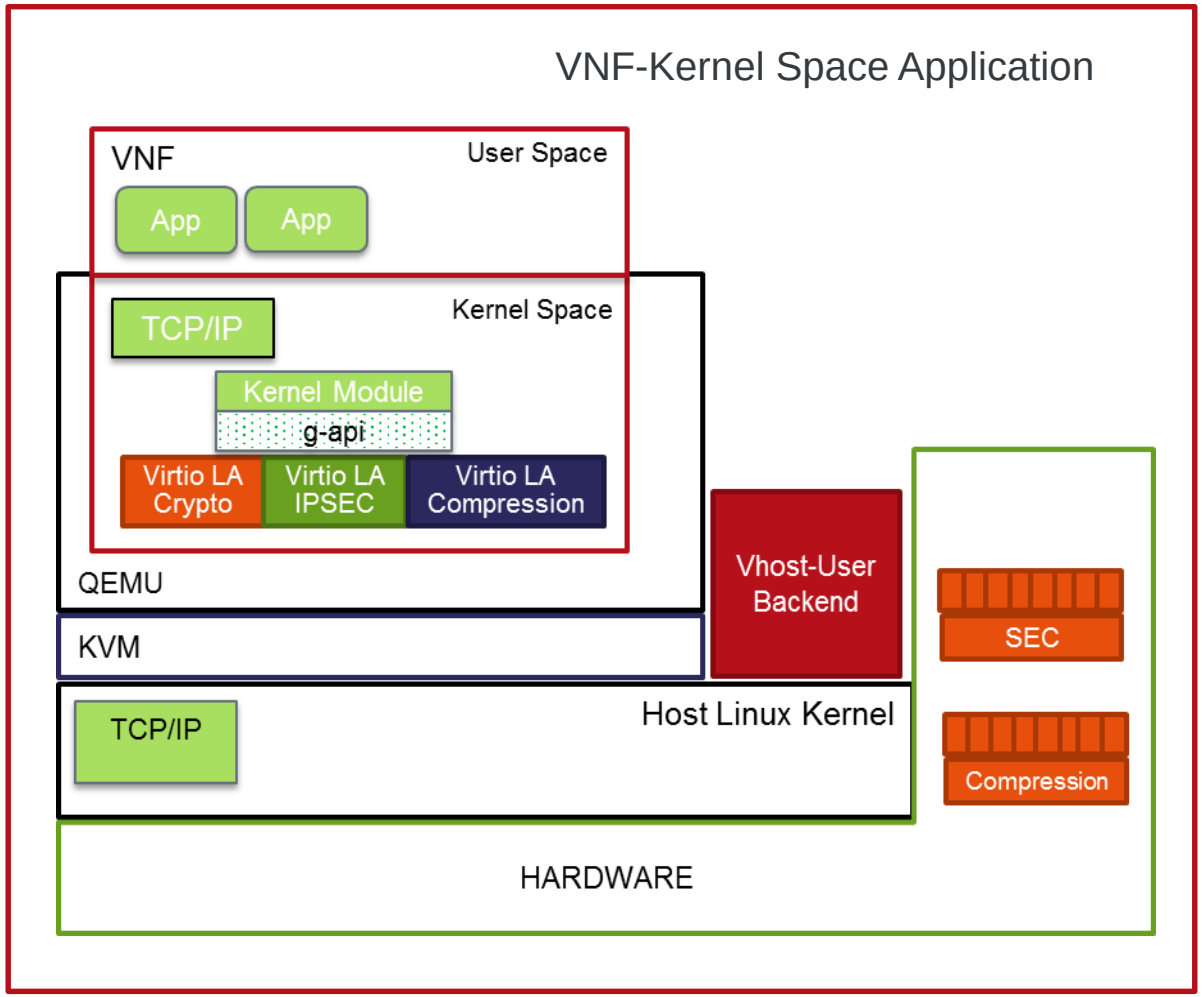
VIRTIO_IPSEC_CTRL_SA

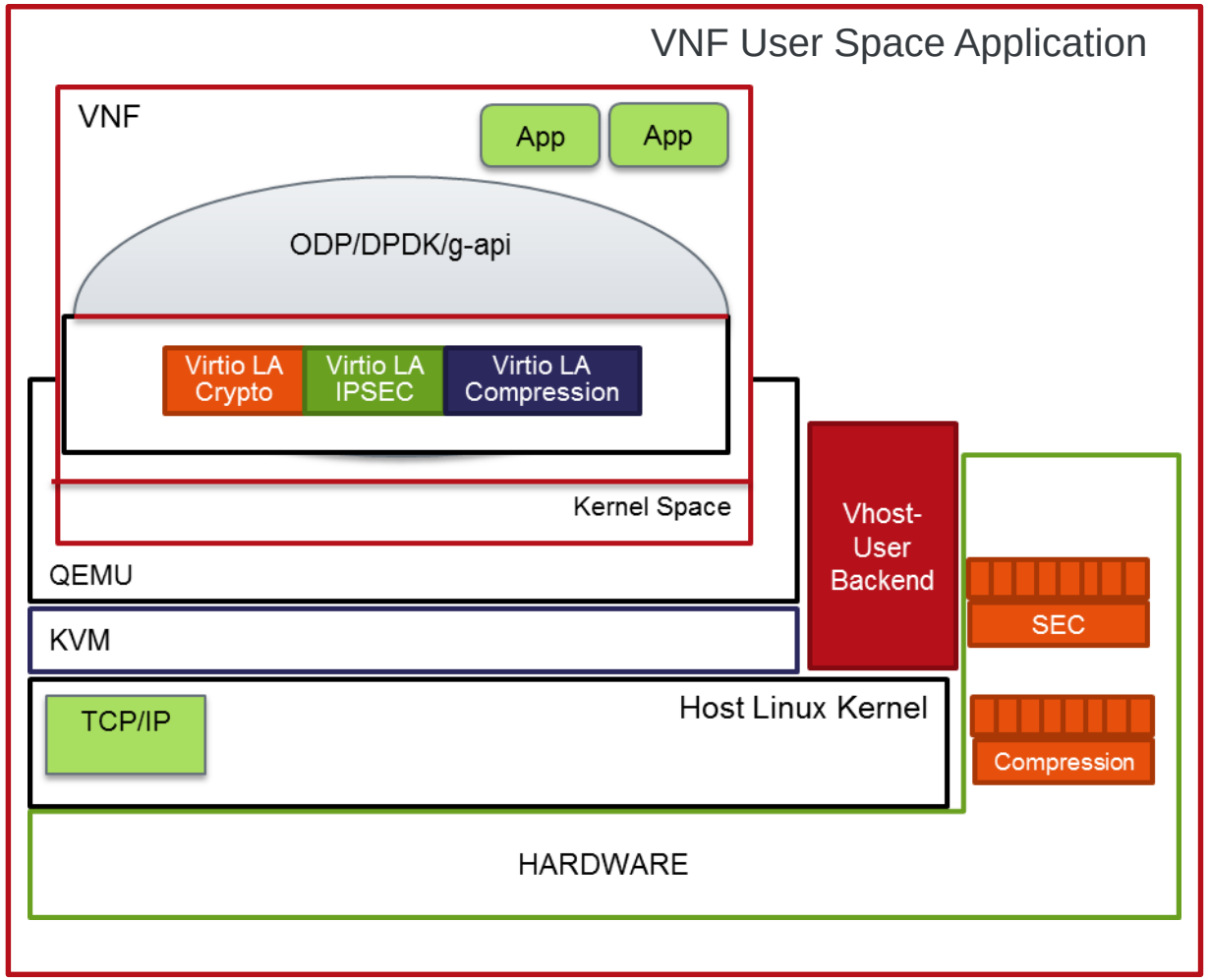
- **VIRTIO_IPSEC_CTRL_ADD_GROUP** Add a group
- **VIRTIO_IPSEC_CTRL_DELETE_GROUP** Delete a group
- **VIRTIO_IPSEC_CTRL_ADD_OUT_SA** Add an outbound SA
- **VIRTIO_IPSEC_CTRL_DEL_OUT_SA** Delete Outbound SA
- **VIRTIO_IPSEC_CTRL_UPDATE_OUT_SA** Update Outbound SA
- **VIRTIO_IPSEC_CTRL_READ_OUT_SA** Read Outbound SA
- **VIRTIO_IPSEC_CTRL_READ_FIRST_N_OUT_SAs** Read first N outbound SAs
- **VIRTIO_IPSEC_CTRL_READ_NEXT_N_OUT_SAs** Read next N Out SAs
- **VIRTIO_IPSEC_CTRL_ADD_IN_SA** Add an inbound SA
- **VIRTIO_IPSEC_CTRL_DEL_IN_SA** Delete Inbound SA
- **VIRTIO_IPSEC_CTRL_UPDATE_IN_SA** Update Inbound SA
- **VIRTIO_IPSEC_CTRL_READ_IN_SA** Read Inbound SA
- **VIRTIO_IPSEC_CTRL_READ_FIRST_N_IN_SAs** Read first N SAs
- **VIRTIO_IPSEC_CTRL_READ_NEXT_N_IN_SAs** Read Next N SAs
- **VIRTIO_IPSEC_CTRL_FLUSH_SA** Flush SAs within a group
- **VIRTIO_IPSEC_CTRL_FLUSH_SA_ALL** Flush all SAs
- **VIRTIO_IPSEC_GET_HEADROOM_TAILROOM_SIZE** Get the headroom, tailroom size

Details

- Documents in git repository
  - https://github.com/fsl-dpacc-poc/docs.git
- G-API guidelines:
  - api_guidelines_xx.doc
- G-APIs for Virtio IPsec
  - Freescale-IPsec-Virtual-Accelerator-gapi-xxxxx.doc
- Virtio IPsec Messages:
  - Freescale-IPsec-Virtual-Accelerator-xxxx.doc

Architecture Model





- PoC Implementation for Kernel Space IPSec to make use of g-api (Currently in focus)
- PoC Implementation for User space IPsec Application to use ODP-dpdk

DPACC PoC (Virtio-IPSec) – Contributions welcome

- VNF Side
  - Linux Kernel based IPSec
    - FSL is working on this
  - User space IPSec DP
    - Which Kit to use
      - NO HW dependency
      - Virtio Drivier framework.
      - ODP over DPDK
      - -→ DPDK as code base to start?
      - → Contributions welcome
- Open Stack Support
  - Advertize accelerator capabilities
  - Libvirt changes
  - NOVA scheduler changes
  - Contributions welcome…
- Backend
  - QEMU Changes to introduce virtio-ipsec-la device.
  - Vairous vendors can implement this.
  - FSL contribution on Layerscape (ARM)

Thank You