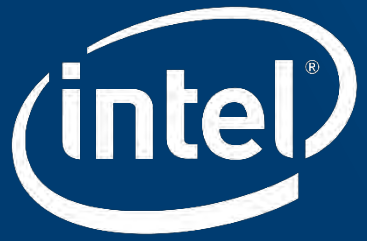


SECURING YOUR DATA ACROSS THE DATA CENTER

Accelerating IPSec for Oracle Database

Topics

Introductions

Threats and challenges to data center security

Security considerations at Oracle

A review of encryption options: TLS vs. IPSec

IPSec implementation options

Extensions and opportunities

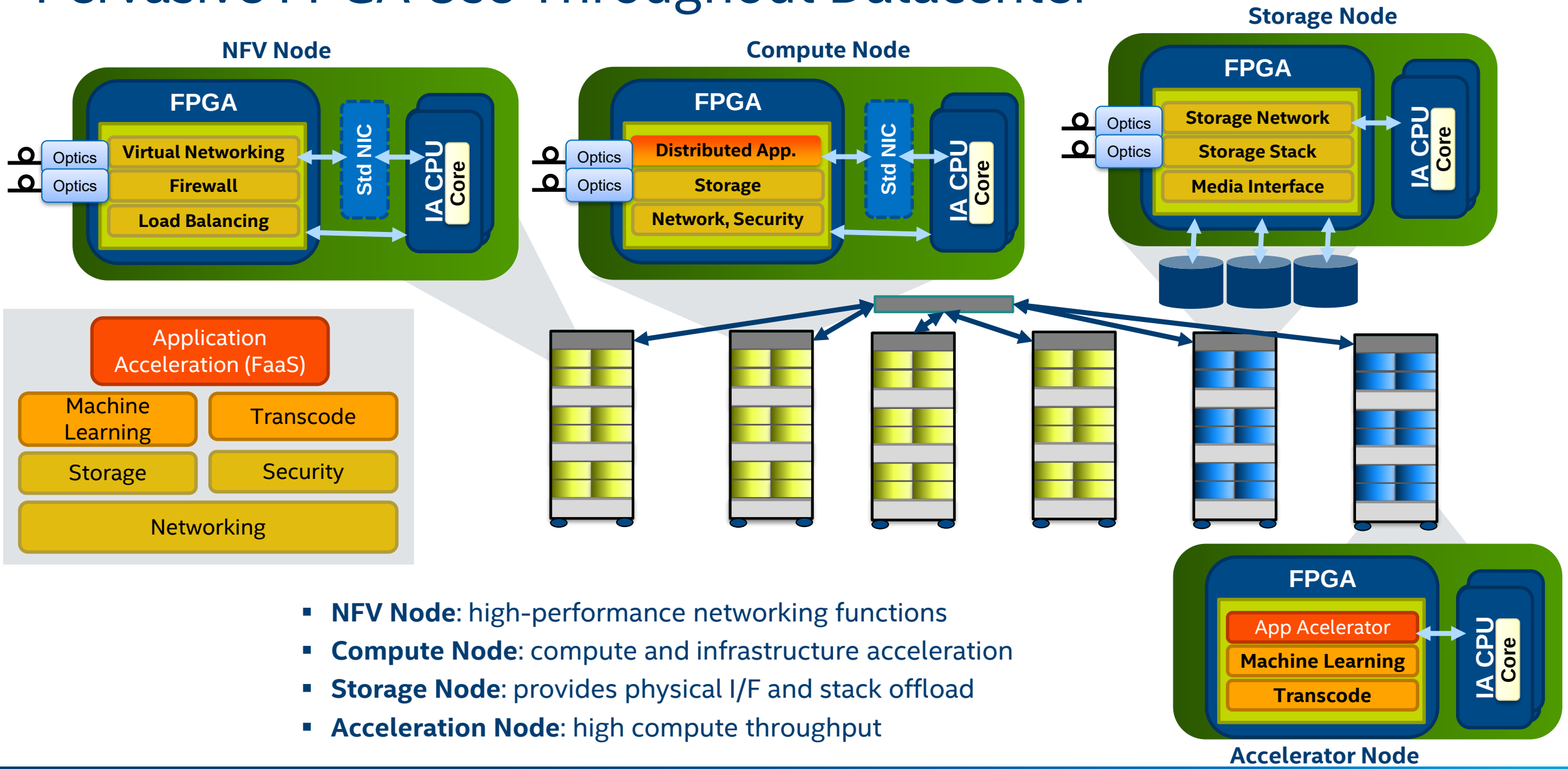
Presenters

Avneesh Pant
avneesh.pant@oracle.com
Director, Database Development
Virtual Operation System Group

Kumaran Siva
kumaran.siva@intel.com
Director, Data Center
Programmable Solutions Group

Nicola Tan
nicola.tan@intel.com
Senior Manager, Data Center Marketing
Programmable Solutions Group

Pervasive FPGA Use Throughout Datacenter



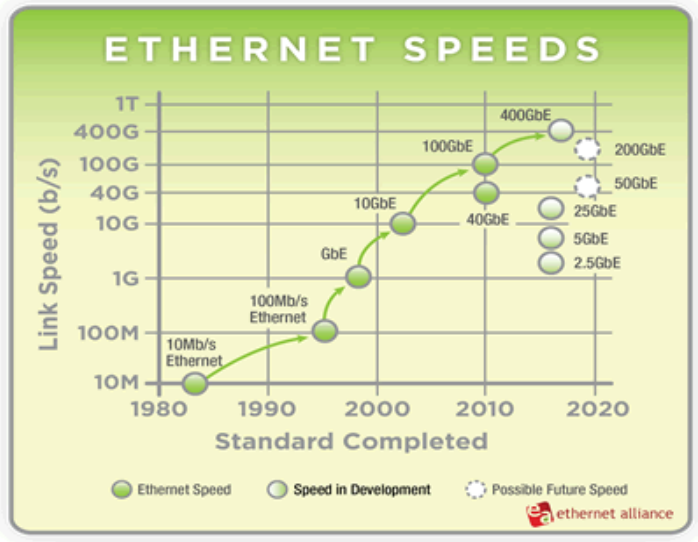
- **NFV Node**: high-performance networking functions
- **Compute Node**: compute and infrastructure acceleration
- **Storage Node**: provides physical I/F and stack offload
- **Acceleration Node**: high compute throughput

Forces **Impacting** Data Center Security

More data, moving faster, with greater security needs

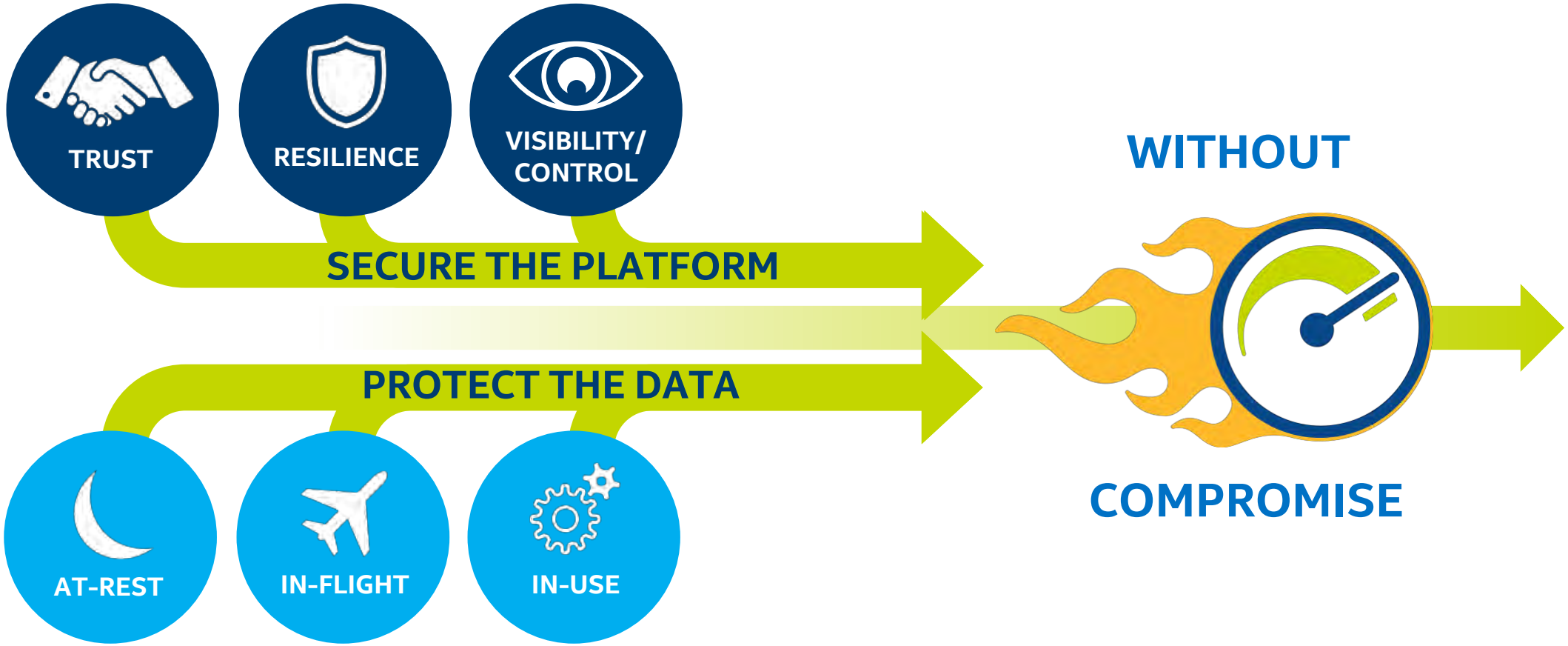
2+ ZB GLOBAL DATA TRAFFIC ANNUALLY BY 2020[1]

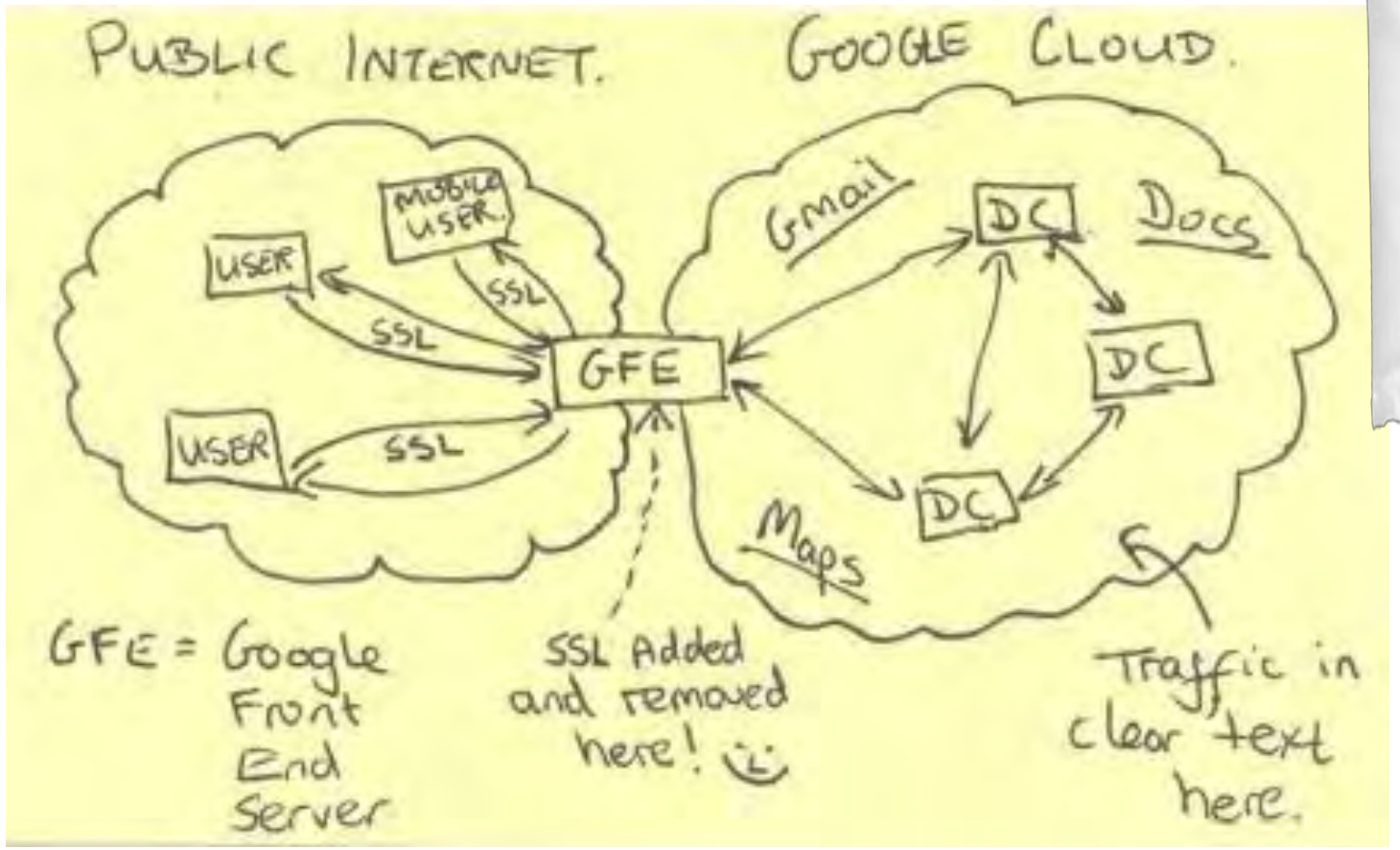
50B CONNECTED DEVICES BY 2020[2]



Intel Data Center **Security Strategy**

Effective security is built on a foundation of trust

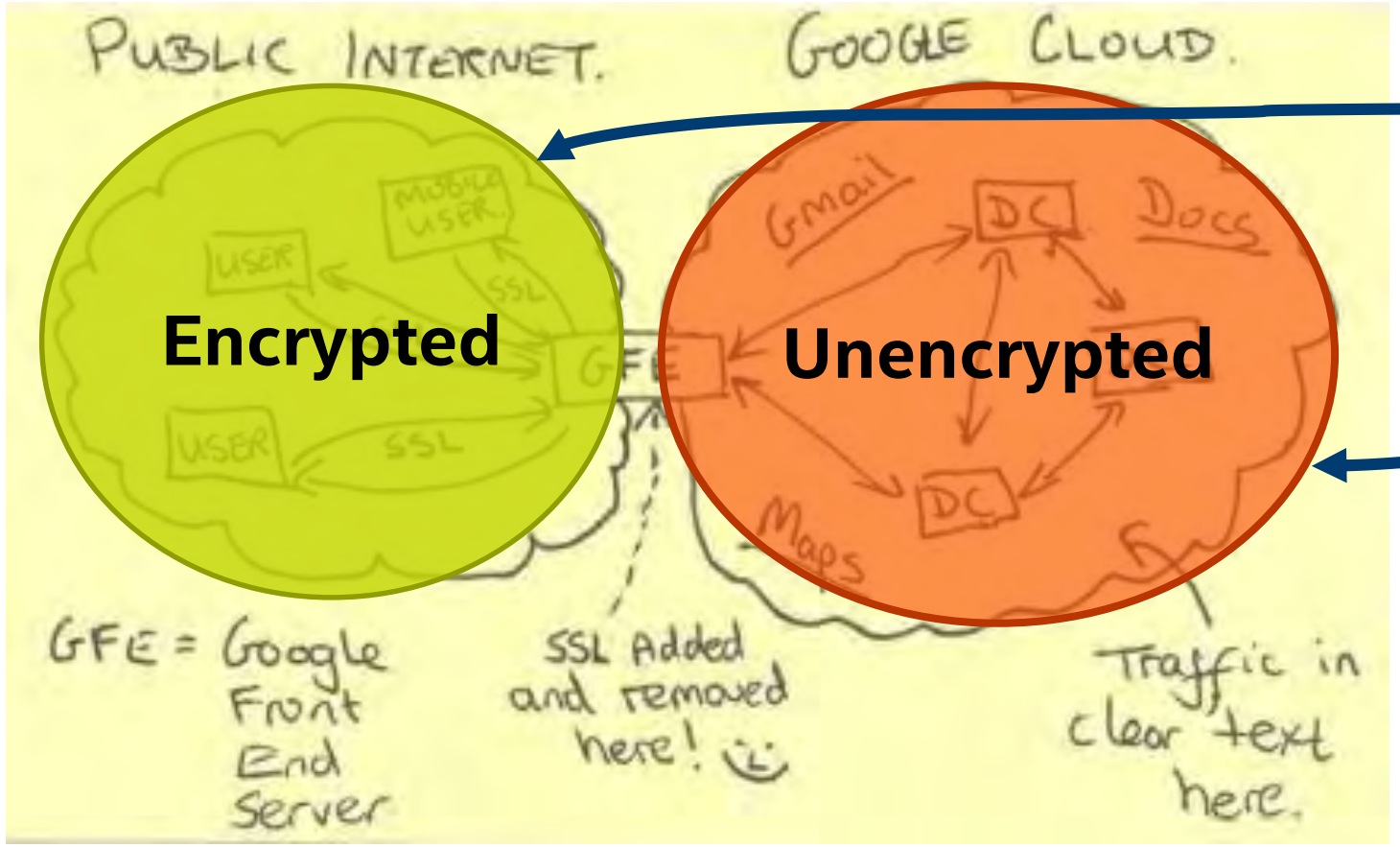


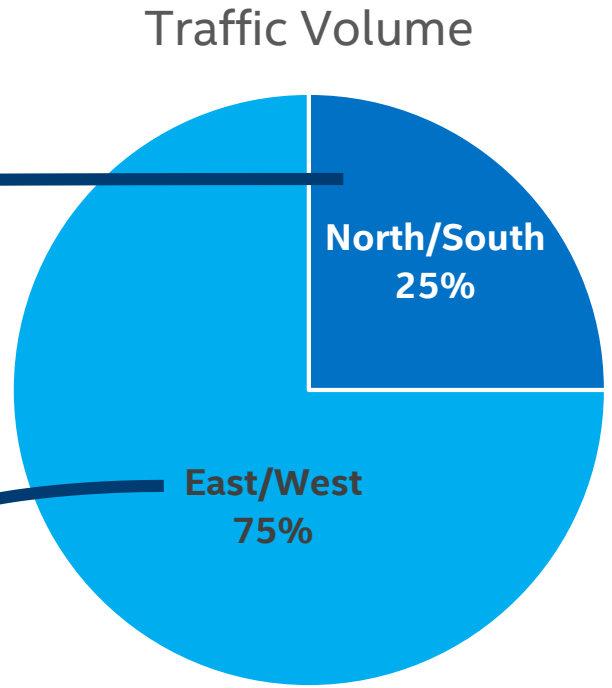


The NSA Hacked Google and Yahoo's Private Networks

More documents from the Edward Snowden leak show that the National Security Agency has tapped Google and Yahoo's cloud networks to access massive amounts of data, including from Americans.

https://www.theatlantic.com/politics/archive/2013/10/nsa-hacked-google-and-yahoos-private-networks/354570/





Performance issues have been a **chief inhibitor** to encryption adoption

- Longer keys
- Complex algorithms
- Exponential data growth

Why security?

Oracle RAC (Real Applications Cluster) traffic currently requires a private network, so traffic goes in the clear today

As we scale to multi-tenant Cloud environments, we have multiple tenants sharing the same physical infrastructure

Attack vectors that need to be considered:

- Protecting tenant payload from identity theft, providing privacy (encryption) of tenant/application data, protection from replay attacks, integrity protection
- Protecting the tunneling protocol header itself (TCP, UDP headers, tunneling protocol headers)

Networking protocol considerations for Oracle traffic

Oracle application transactions are typically massive numbers of request-response exchanges that expect reliable, ordered delivery of data

Why not use TCP for these transactions (and TLS from user space)?

- Large number of peers, so using a TCP socket for each pair of communicating peers would result in a connection explosion

TCP sockets provide stream semantics (no message boundaries). Oracle application complexity is significantly reduced with datagram semantics (message boundaries managed by the transport)

Why not UDP?

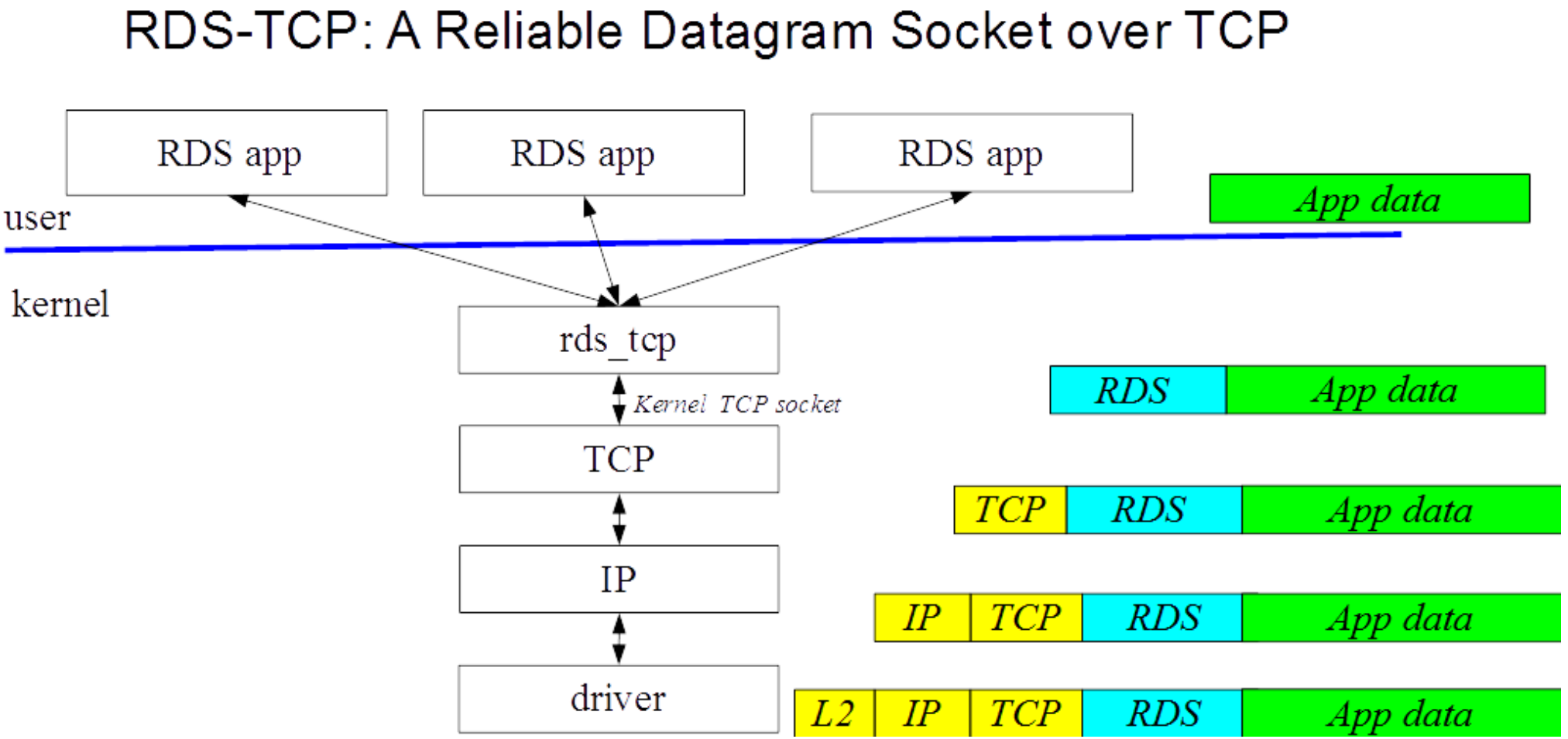
UDP provides datagram semantics. A single UDP socket can be used to send/receive packets to/from multiple peers. DTLS provides AAA (Authentication, Authorization, Accounting) and privacy for UDP sockets.

Drawbacks to using UDP:

- UDP does not have intrinsic congestion management, so application is burdened with complex congestion management logic
- Oracle application data tends to be 8K or larger packets so that we have to choose between:
  - performance impact of IP Fragmentation, or,
  - track MTU on in user space and manage a layer that can reliably send/receive MTU sized records

What we ideally want is “Reliable UDP socket”

RDS-TCP: A Reliable Datagram Socket over TCP



Security considerations for kernel managed TCP and UDP sockets

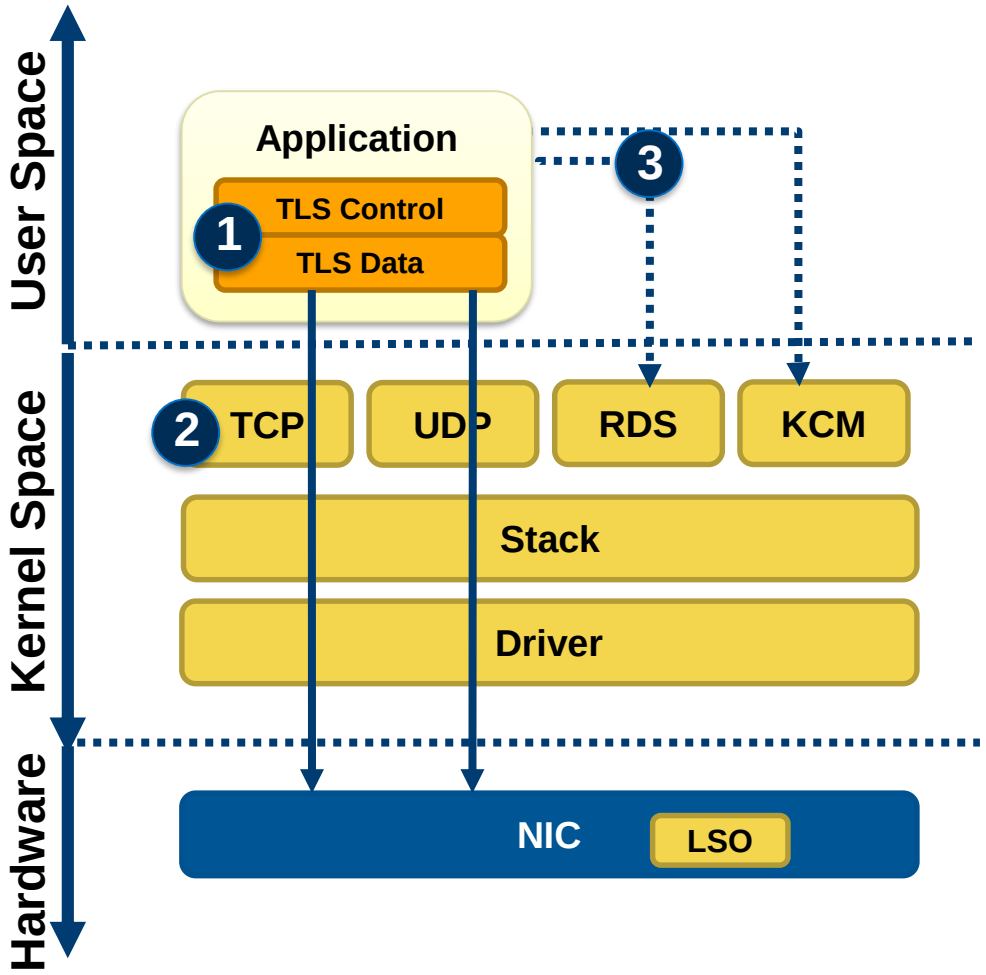
TCP socket in the RDS-TCP architecture is a kernel managed socket, and we do not currently have a standards-compliant TLS implementation in the kernel

TLS is a complex protocol, kernel implementations of TLS have steep challenges

Lack of security for the tunneling protocol header also exists for other data-center protocols like VXLAN, RoCEv2 etc

We do have mature, standards-compliant implementations for IPsec in the kernel that provide AAA and privacy at the IP layer of the network stack.

Encrypting User Data With TLS/DTLS



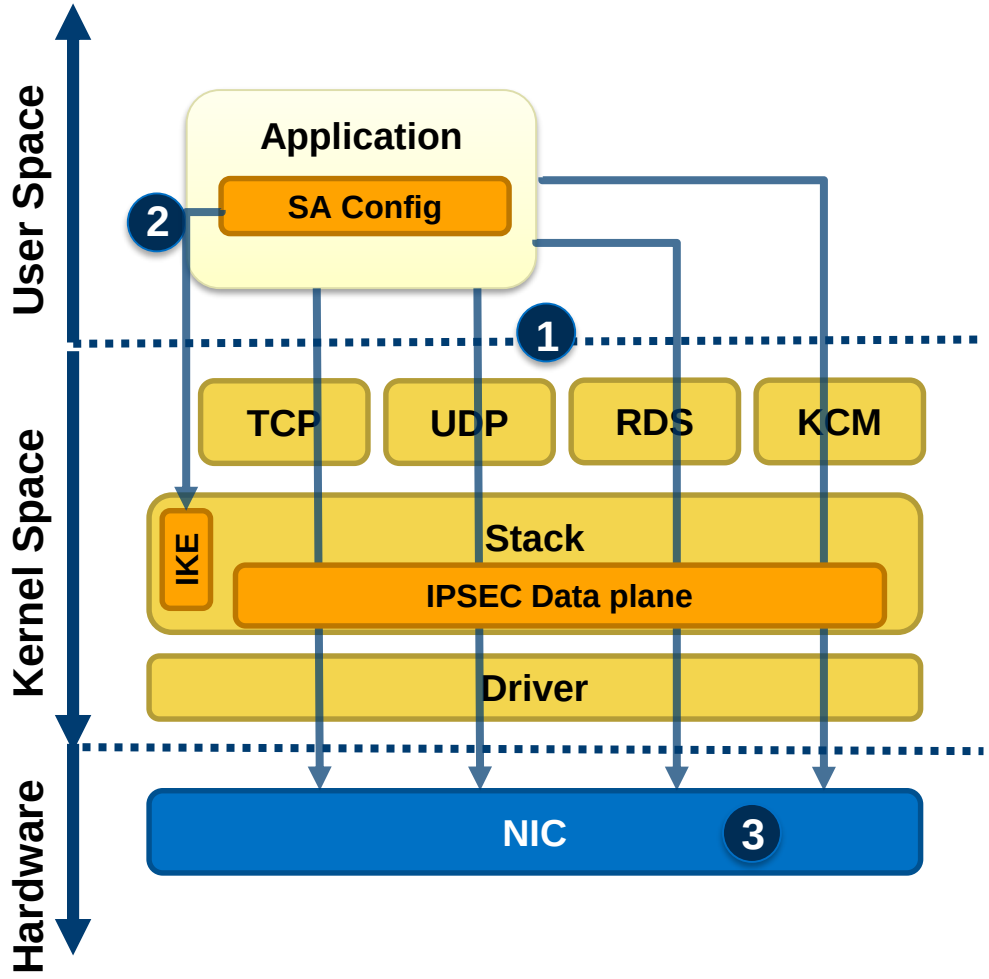
North South use-case with TLS/DTLS

- Implements DTLS/TLS in user space using open source implementation
- Uses TCP or UDP sockets in Kernel
- Allows Large Segment Offload in NIC

Challenges With TLS

- DTLS/TLS control and data plane are complex and tightly coupled (1)
- Does not protect against TCP attacks (2)
- Does not support RDS, KCM today → need to bring into Kernel for RDS, KCM support (3)

Encrypting User Data With IPSEC



IPSEC Provides Kernel Encryption

- Works across broad socket types 1

Separable Control & Data

- Standard interface for key management 2

IPSEC Processing After NIC limits offloads

- Large Segment Offload cannot be used 3

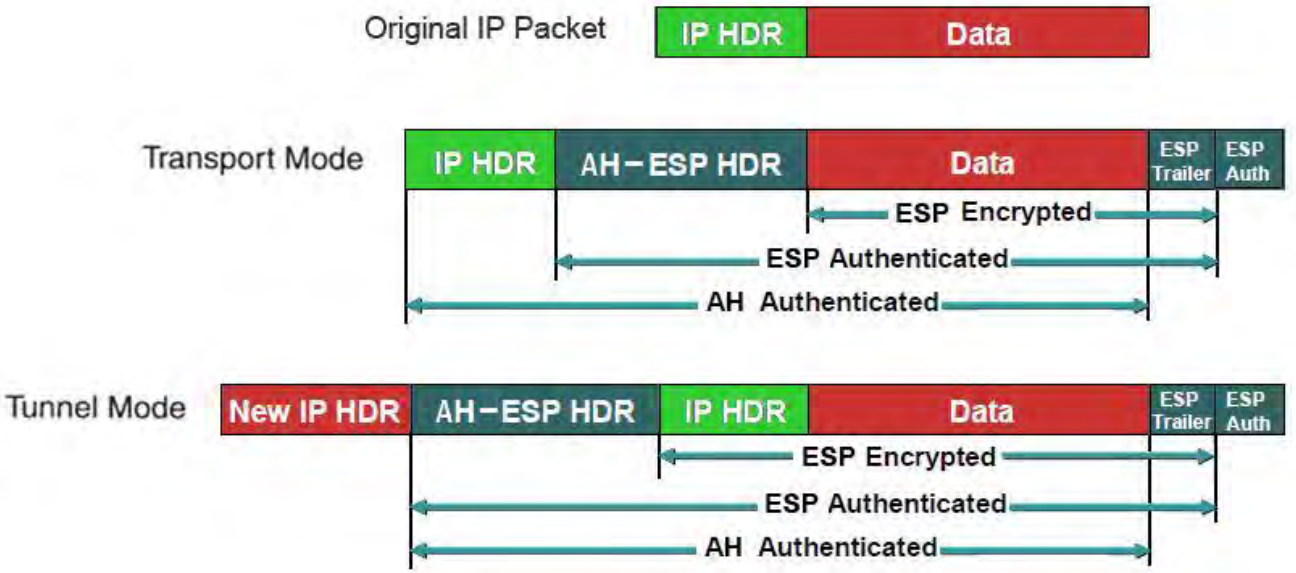
IPSec Background

Internet Protocol Security (IPSec) architecture is a set of protocols that ensures data security of an IP network

IPSec ensures:

- **Integrity** – data has not been changed
- **Confidentiality** – data is not readable by third parties
- **Authentication** – data comes from intended source

Works at the network level, not application specific



Intel Products for Encryption & Compression

Standard
For standard algorithms that do not require further acceleration or CPU offload

Compute Intensive Crypto & Compression
For lookaside offload acceleration on standard algorithms
symmetric crypto, public key encryption, de/compression

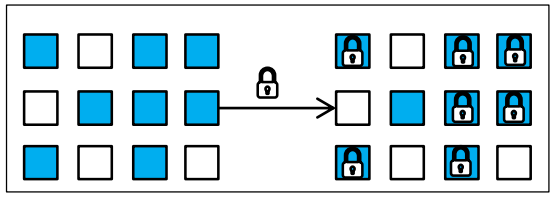
Custom Algorithms or Inline Processing
For desired flexibility in crypto/compression algorithms or for inline processing capability

Intel® Xeon® Processor E5 and E7 Families

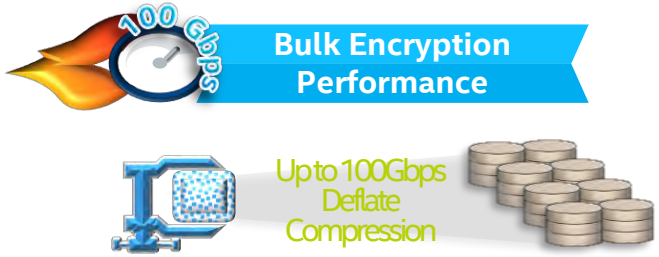
Intel® QuickAssist Technology

Intel FPGAs

Example: AES-256 bit encryption using Intel® AES-NI

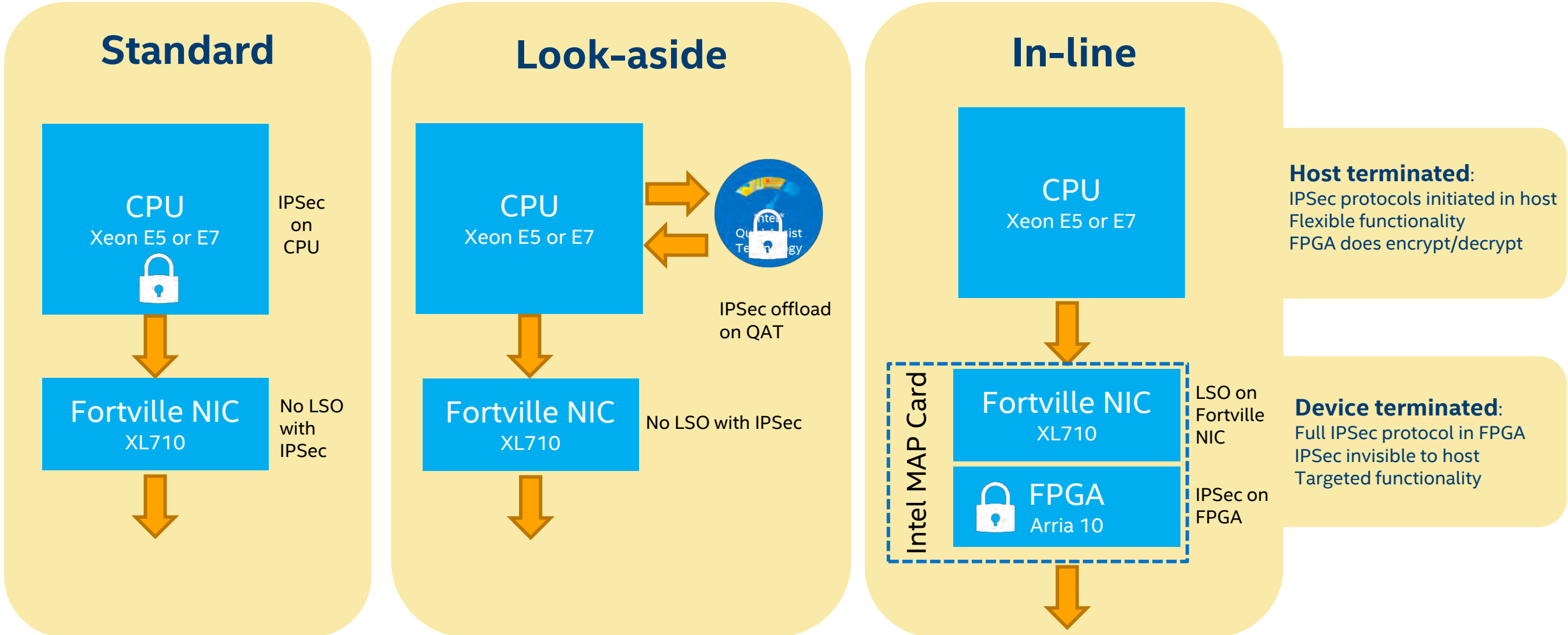


Example: Bulk operations

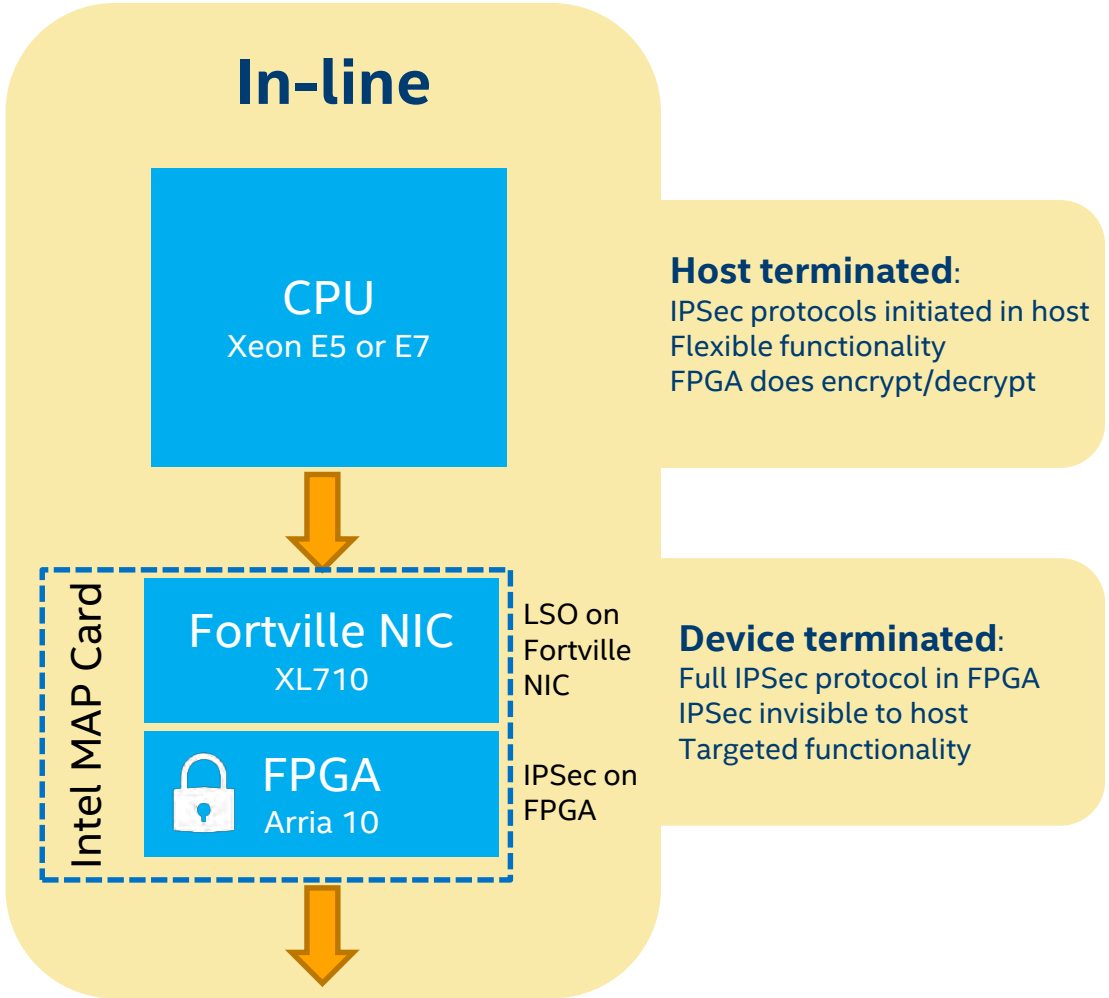


Example: Customer or Geo Specific algorithm

Three Models for IPSec Implementation



In-line IPSec Acceleration



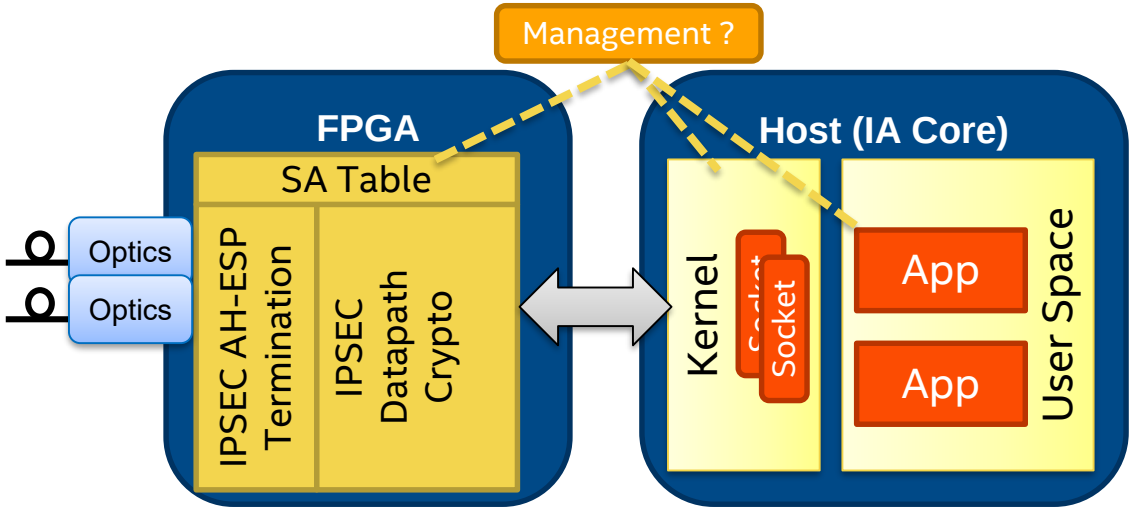
Key benefits:

- Minimizes CPU overhead
- Minimizes latency
  - Less PCIe passes
- Exploits order of operations limitations
  - Encryption occurs after passing through NIC
  - NIC offloads (checksum, LSO, tunneling) not permitted in other IPSec implementations

DEMO

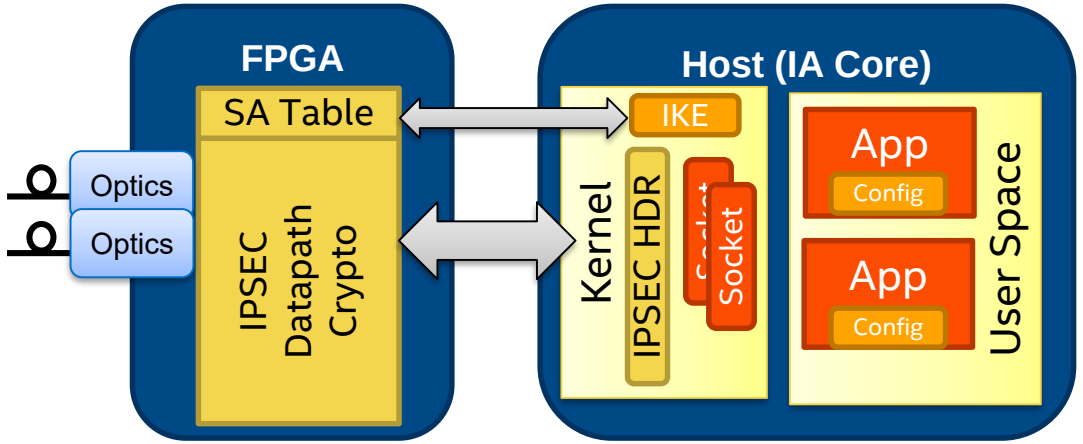
Inline IPSec Implementations

Device Terminated



- Terminates IPSEC headers in hardware
- Bump the wire, transparent to host
- Main challenge is control plane

Host Terminated



- Software terminates IPSEC headers
- Leverages IPSEC control plane infrastructure
- Interface to applications is simplified

Kernel support

What needed to happen to software to enable this?

Upstream Linux kernel stack infrastructure - a series of patches, that started with https://lwn.net/Articles/710591/ with several follow-ons.

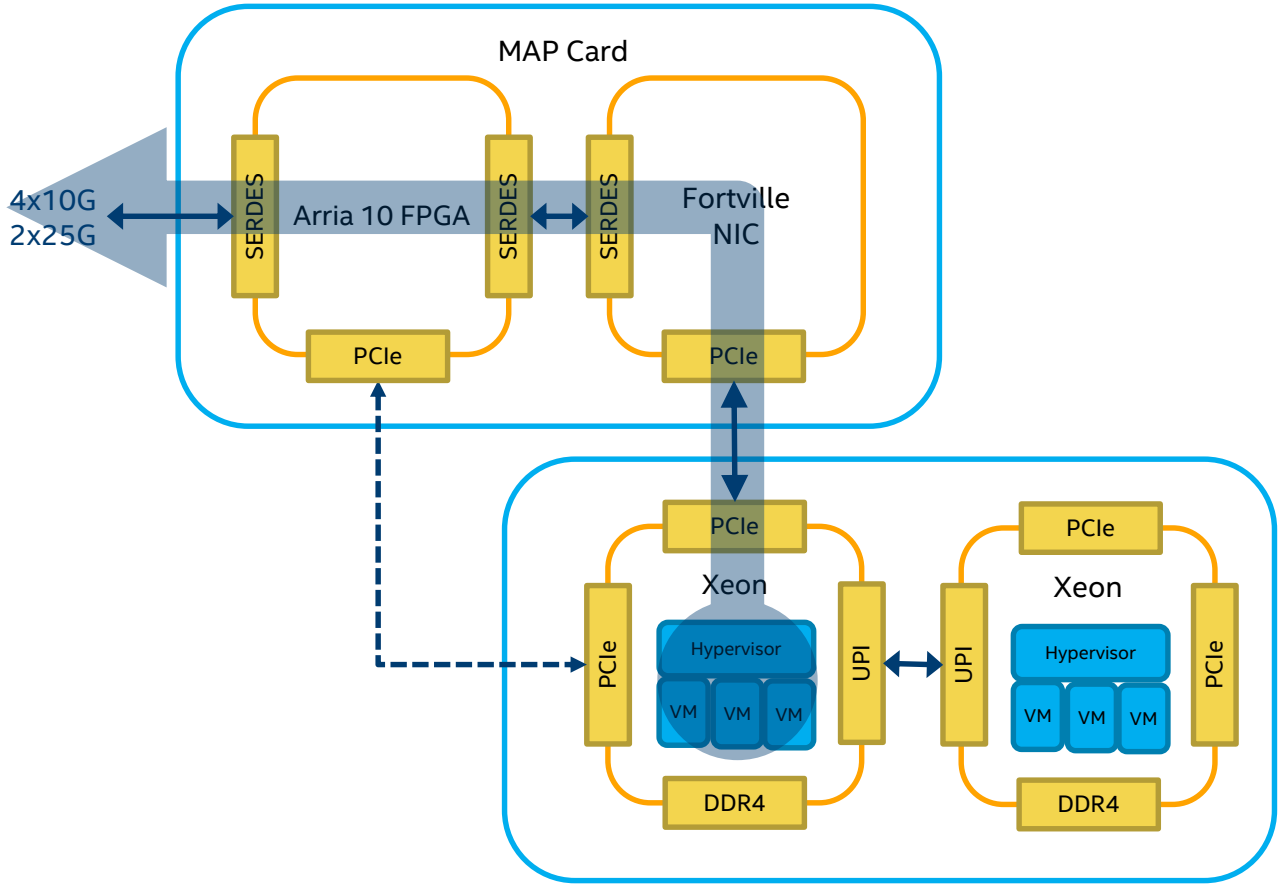
Contributors: Steffen Klassert, Ilan Tayari, **Sowmini Varadhan (Oracle)**

IPsec offload, part one

From: Steffen Klassert <steffen.klassert-AT-secunet.com>
To: David Miller <davem-AT-davemloft.net>, <netdev-AT-vger.kernel.org>
Subject: [PATCH RFC ipsec-next] IPsec offload, part one
Date: Wed, 4 Jan 2017 09:23:45 +0100
Message-ID: <1483518230-6777-1-git-send-email-steffen.klassert@secunet.com>
Cc: Steffen Klassert <steffen.klassert-AT-secunet.com>, Sowmini Varadhan <sowmini.varadhan-AT-oracle.com>, Ilan Tayari <ilant-AT-mellanox.com>
Archive-link: Article

```
his is the first part of the IPsec offload work we
alked at the IPsec workshop at the last netdev
onference. I plan to apply this to ipsec-next
fter this round of review.

atch 1 and 2 try to avoid skb linearization in
he ESP layer.

atch 3 introduces a hepler to seup the esp trailer.

atch 4 prepares the generic network code for
Psec GRO. The main reason why we need this, is
hat we need to reinject the decrypted inner
acket back to the GRO layer.

atch 5 introduces GRO handlers for ESP, GRO
an enabled with a IPsec offload config option.
his config option will also be used for the
pcomming hardware offload.

avid, patch 3 touches generic networking code.
s it ok to integrate such a generic preparation
atch into an IPsec pull request, or do you
refer to get it as a separate patch?
```

System Architecture



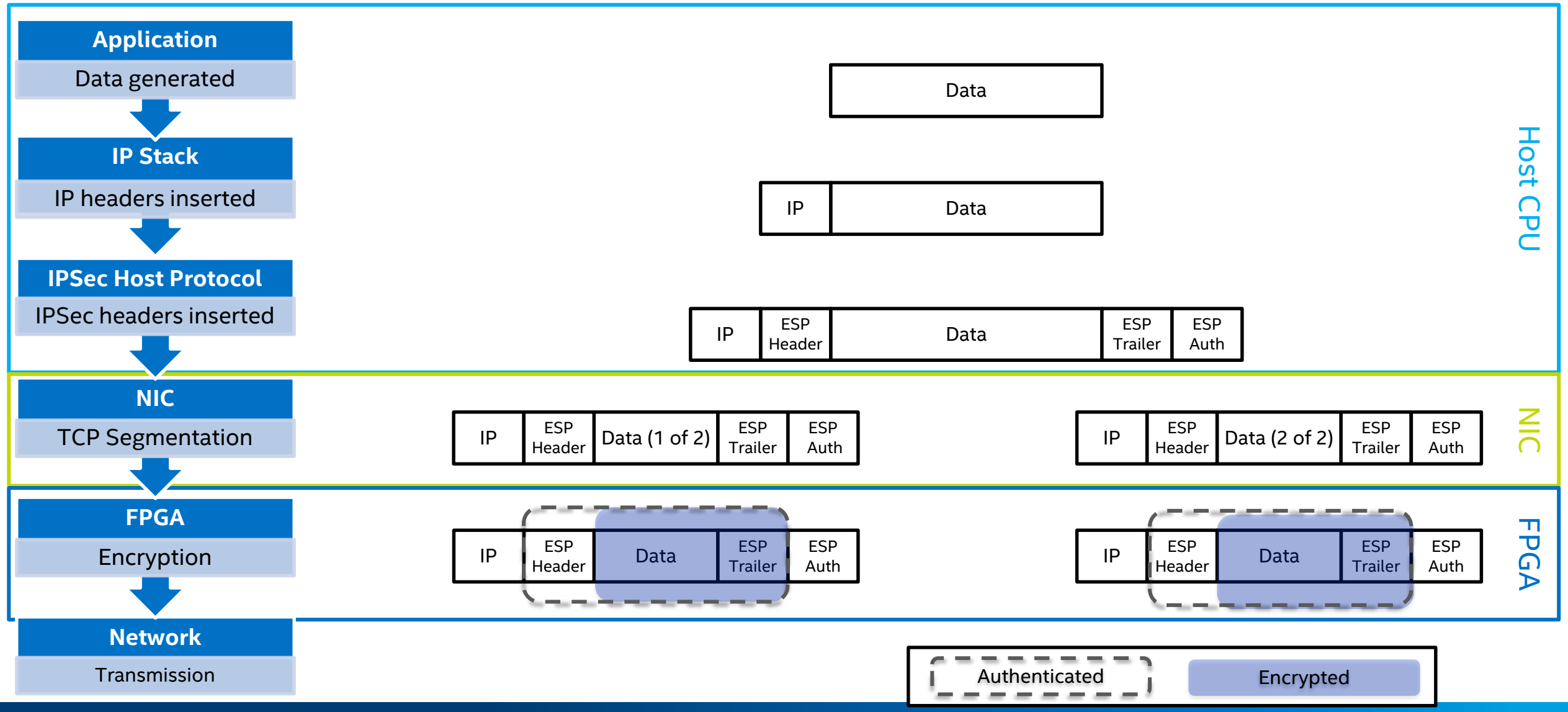
Key Functions

Arria 10 FPGA
- IPSec encryption and authentication

Fortville NIC
- DMA Ring Interface
- Queue management
- MAC, VEB, VLAN

Host
- Fortville driver
- IPSec driver
- Host termination of IPSec protocol
- Security association configurations

Packet Walk (ESP Transport Mode)



Intel® Arria 10 FPGA-based SmartNIC

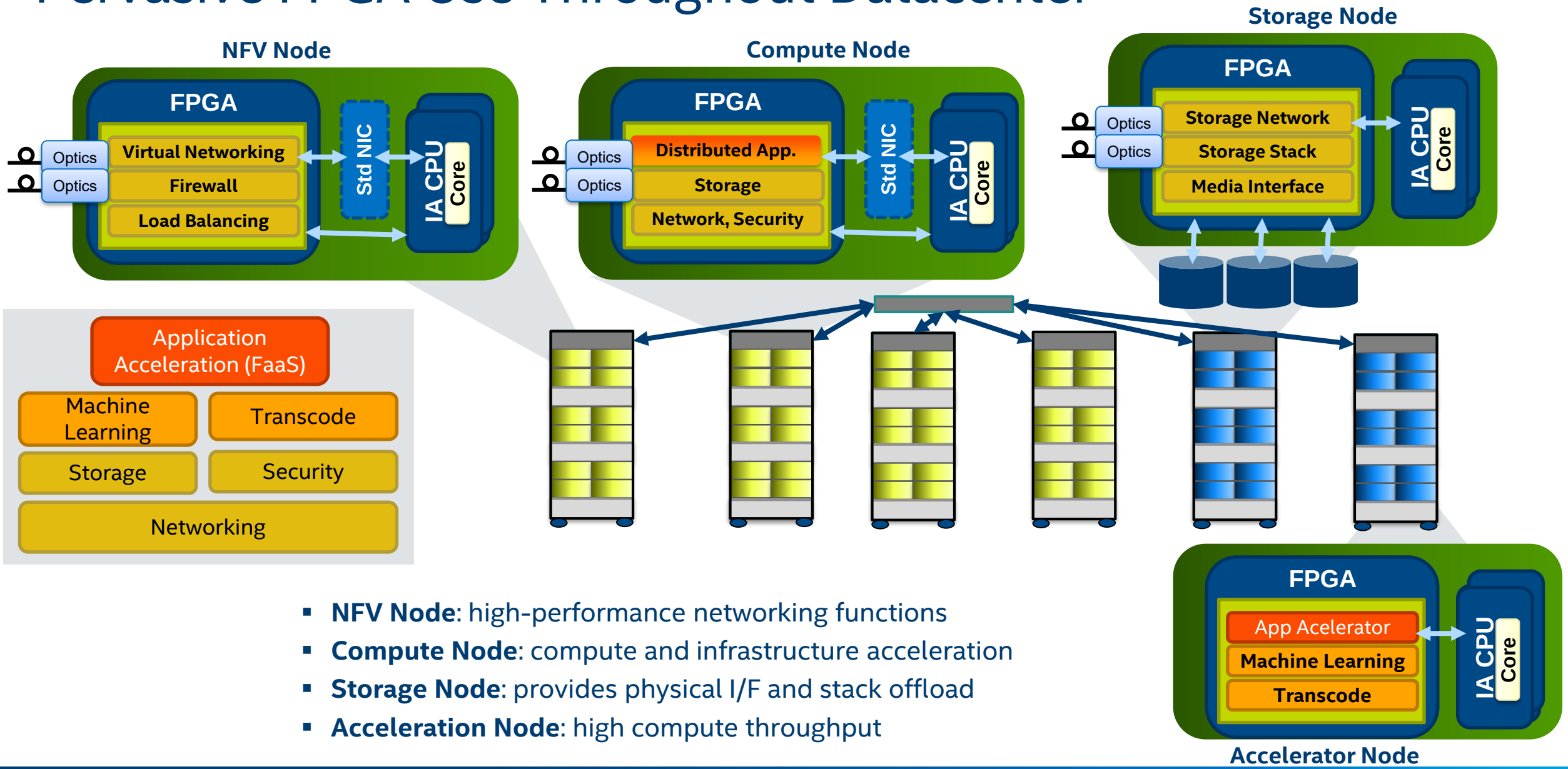
Arria 10 (FPGA)
- Intel Arria 10 1150 GT, 20nm
- Accelerated function IP and gearbox

QSFP-A
4x 10G or
4x 25G QSFP28

QSFP-B
4x 10G

XL710
- Intel Fortville Ethernet NIC
- 10/40 GbE

Pervasive FPGA Use Throughout Datacenter



- **NFV Node**: high-performance networking functions
- **Compute Node**: compute and infrastructure acceleration
- **Storage Node**: provides physical I/F and stack offload
- **Acceleration Node**: high compute throughput

Takeaways

Securing East/West traffic is vital to an overall data center security strategy

Inline IPSec acceleration offers high throughput, low latency encryption

Intel FPGAs offer flexibility for multiple uses across the data center

For more information: IPSec Workshop at Netdev 2.2 (Nov 8-10, Seoul)

BACKUP

FPGA Technology Introduction

Partial Reconfiguration

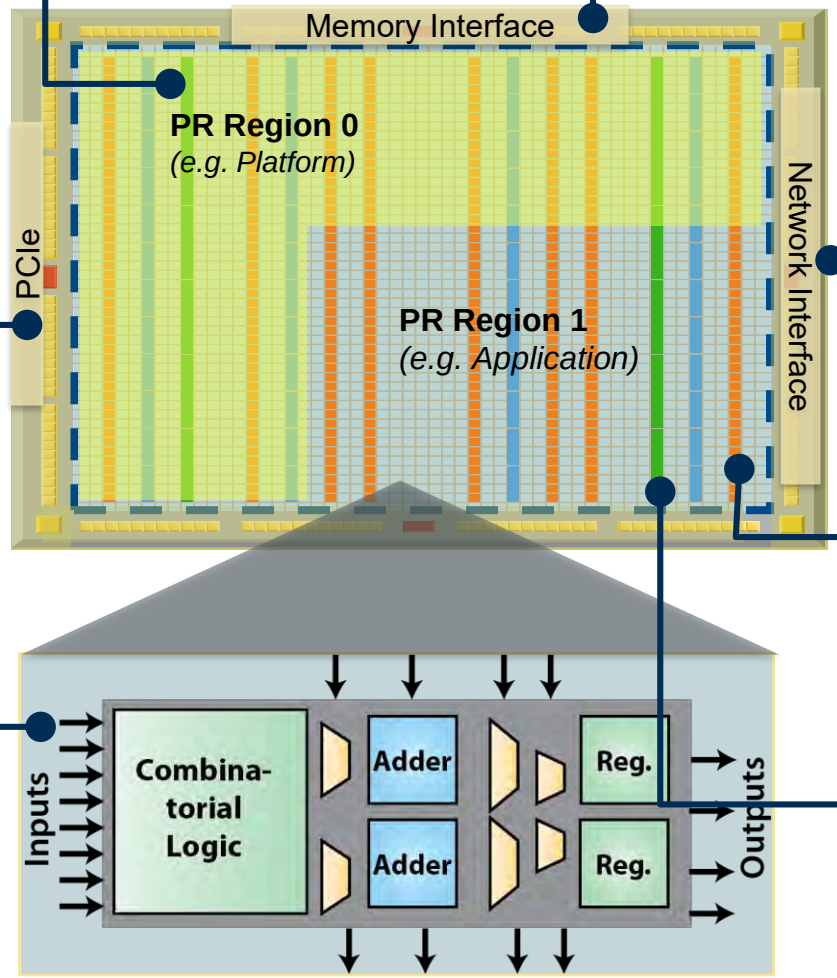
- Allows separate regions

PCIe Host Interface

- Hardened + Soft host interface
- Hardened PCIe controller
- Soft interface allows different use models and drivers

Logic Elements

- Main programmable component
- Millions of logic elements
- Simple logic, adders, and registers
- Interconnect with configurable fabric



Memory Interfaces

- Configurable high performance memory interfaces
- Hardened controllers

Network Interface

- Configurable network interfaces
- Hard/soft interfaces

Memory Blocks

- Thousands of 20Kb memory blocks
- Allows processing to stay on-chip

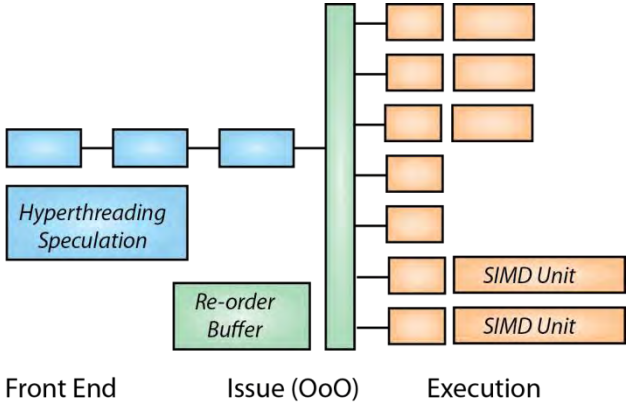
Variable Precision DSP Blocks

- Allows FPGA to perform compute intensive functions

Where FPGAs Fit In?

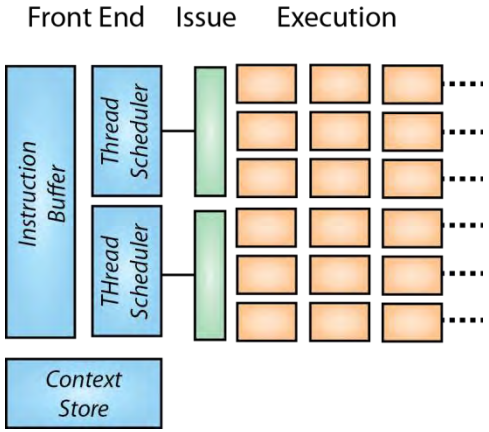
FLEXIBILITY ⟵ ⟶ EFFICIENCY

CPU



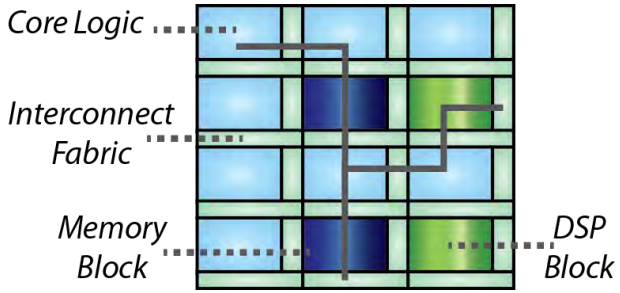
- Balanced architecture: *Good enough most workloads*
- Good single thread & throughput perf.
- Fastest cadence

GPU



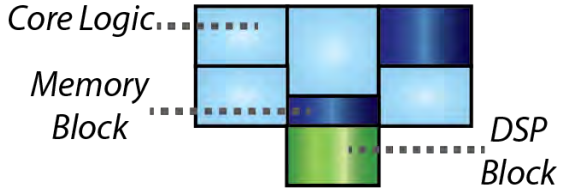
- Focused on compute throughput
- Many low performance threads
- High memory throughput
- Purpose made programming tools

FPGA



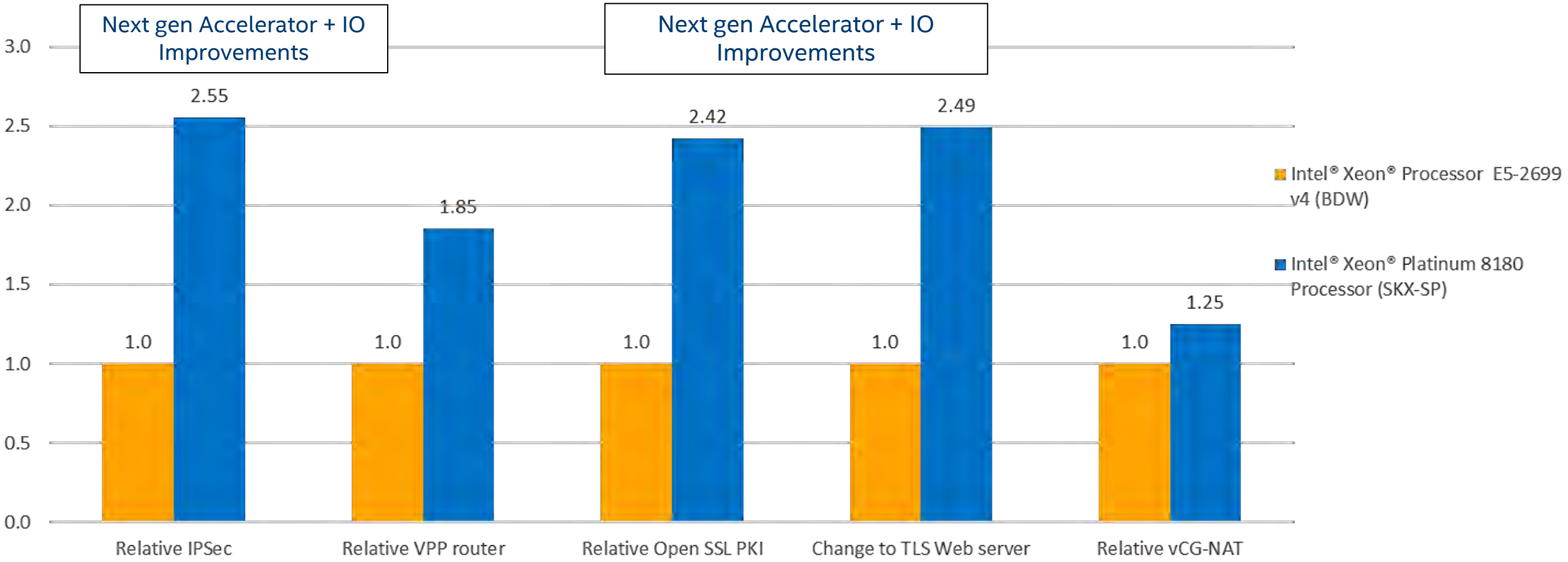
- Full custom pipeline
- Capable of networking and compute
- High memory throughput
- Change cadence in months → rapidly changing needs
- Requires sophistication

ASIC



- Fixed function
- High efficiency – only blocks that are needed
- Change cadence in years → needs stable standards
- Expensive: minimum volume for viability
- Requires sophistication

Application Level Comparison – Skylake vs. Broadwell



"Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance. Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to http://www.intel.com/performance/datacenter. Configurations: see backup

Intel® QuickAssist technology

Optimize the platform beyond processor ISA algorithm performance with hardware for additional scale and workload efficiency

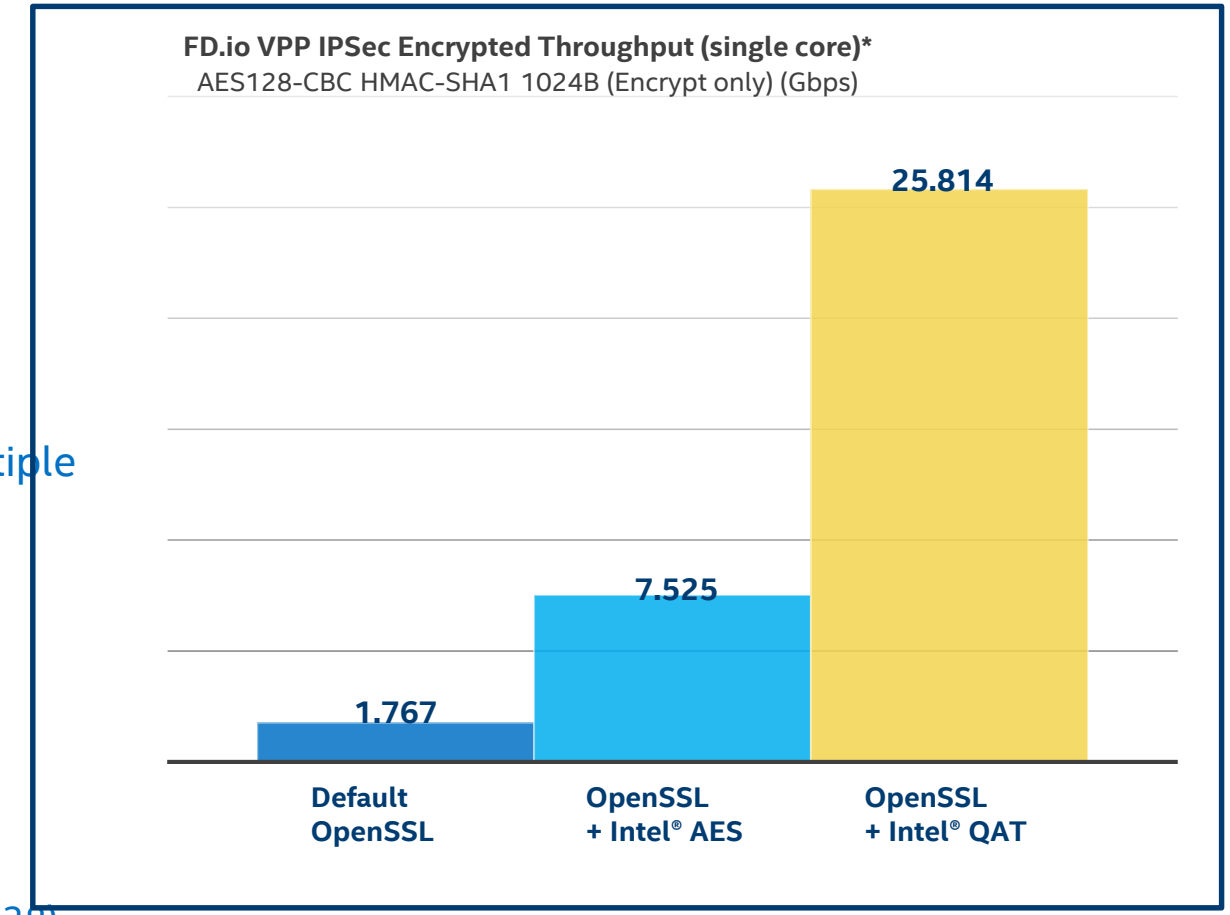
- Cryptography (cipher and authentication operations)
- Public key (RSA, Diffie-Hellman, and elliptic curve cryptography)
- Compression and decompression (DEFLATE and LZS)

Technology available in form factors and packages to meet multiple market requirements for cost, form factor, power, flexibility, etc)

PCIe Card – Intel and 3rd party (e.g. Intel® QuickAssist Adapter)

Chipset Option (e.g. Intel® Communications Chipsets 89xx)

Integration with CPU as SoC (e.g. Intel® Atom™ Processor C2000, Intel® Atom™ Processor C3338)



* See backup slide "VPP IPSec Performance Configuration" for details

Ideal choices for solutions targeting crypto and compression heavy workloads