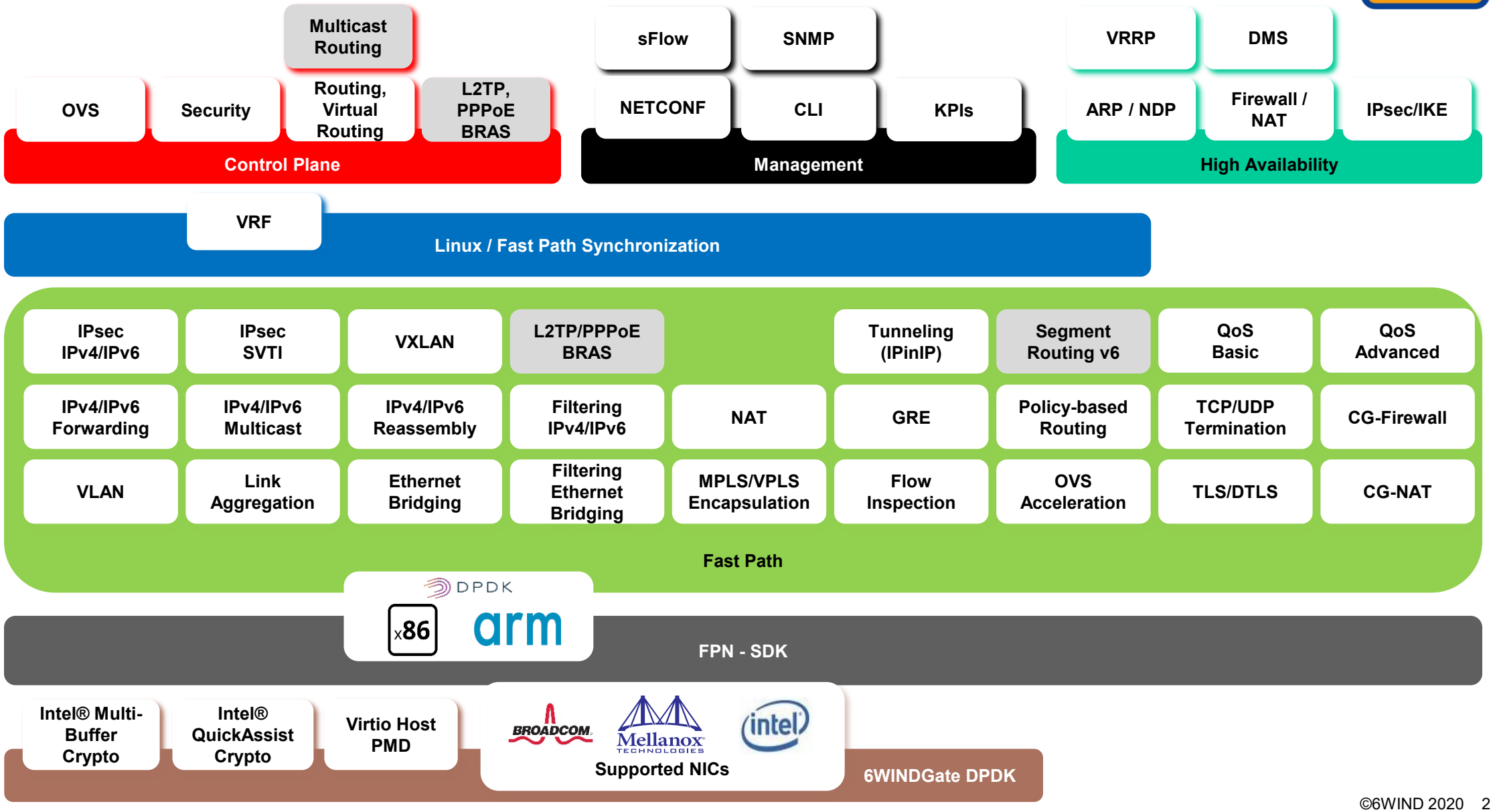
6WINDGate TCP

6WIND

#SPEEDMATTERS For Serious Networks

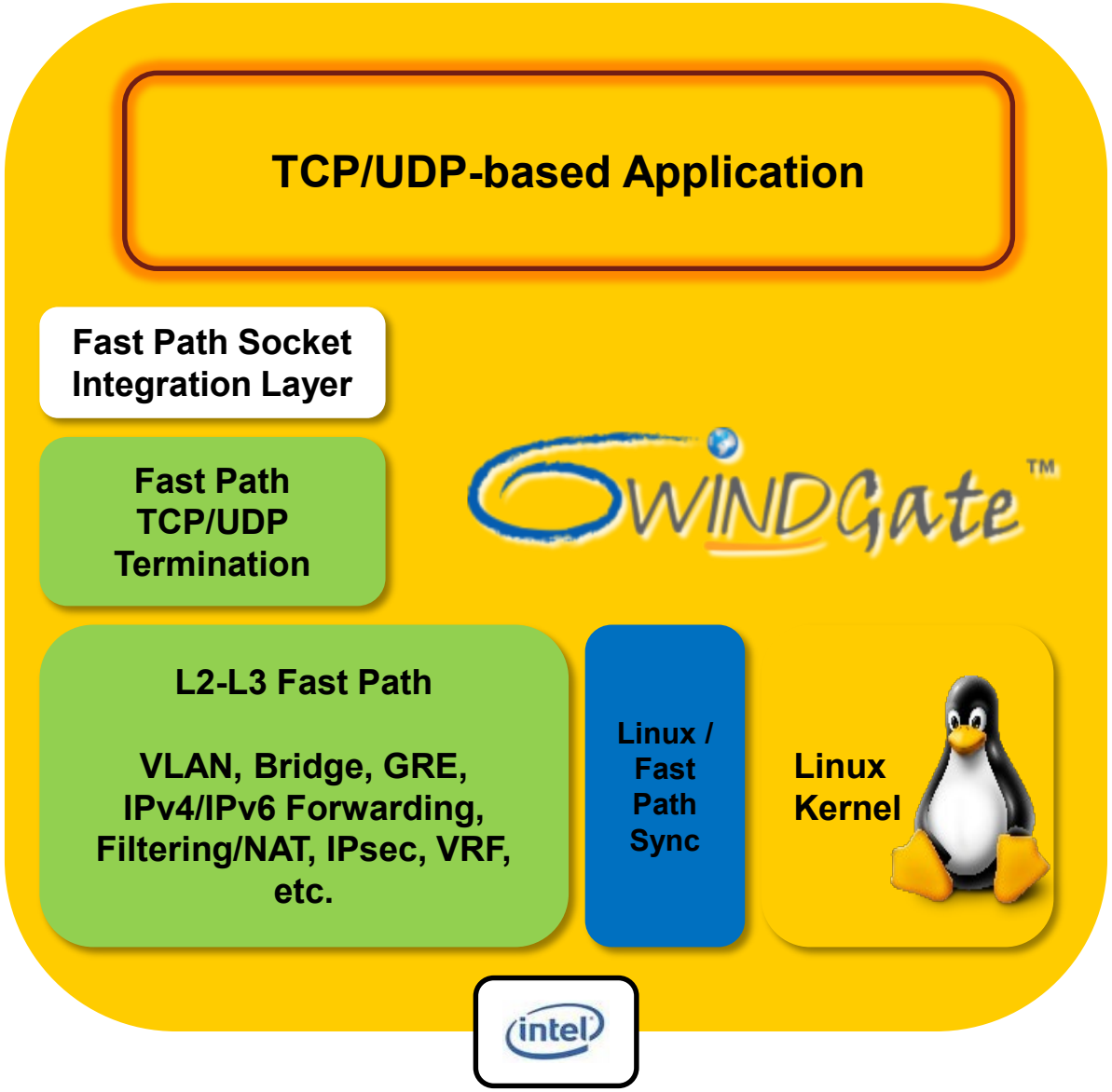
Accelerated Layer 2-4 Stacks Synchronized with Linux



Roadmap

6WINDGate TCP/UDP Termination

- **Software**
  - 6WINDGate source code license including the TCP modules and others 6WIND modules depending on the customer use case
- **Integrated with L2-L3 6WINDGate modules**
- **TCP stack configuration through dedicated CLI**
- **TCP/UDP-based application must be integrated with Fast Path Socket Integration Layer**

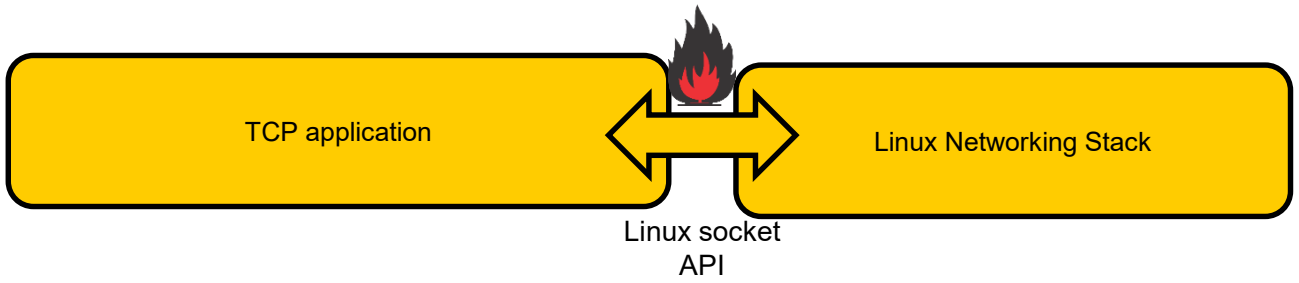


6WINDGate TCP Implementation

- **Stack implementation**
  - Redesigned and fully optimized for multi-core execution environments
  - Highly scalable processing of parallel establishment / shutdown of TCP connections
  - High performance data exchanges on a huge number of TCP/UDP sockets on established TCP connections
  - Event driven notifications from the stack to the application
  - Plugin support for custom TCP/UDP applications including bridged and routed transparent proxy support with configurable IP bypass lists

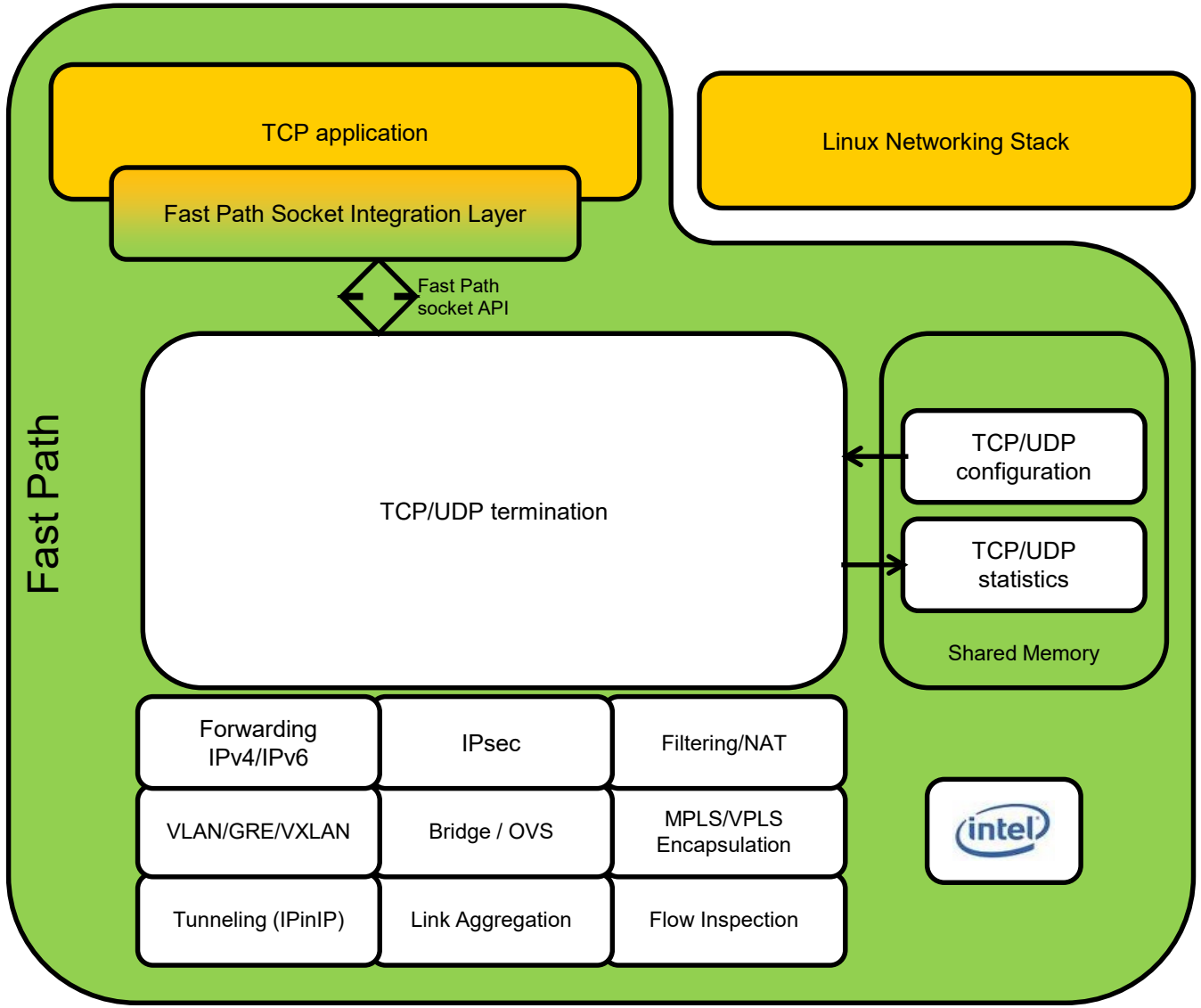
- **Socket API**
  - Full support of TCP and UDP sockets over IPv4 and IPv6
  - POSIX-compliant socket API and Zero-copy based socket APIs
  - VRF-aware sockets
  - Netstat like support to dump state and statistics of the sockets

Architecture



- **TCP application performance suffers from Linux networking stack bottlenecks**

Architecture



- **Fast Path TCP/UDP termination**
  - TCP/UDP protocols are processed in the Fast Path
  - Full featured TCP/UDP stack using BSD-like socket API
  - Timers are re-designed to get more scalability
  - Locks are removed
  - Memory footprint is reduced
- **Performance**
  - Scale: 8M active concurrent TCP sockets
  - Throughput: 40+ Gbps
  - CPS: 1.47M TCP connections per second
  - TPS: 7.1M TCP transactions per second
  - Latency TTFB: 24 µs
- **Optimized Fast Path TCP/UDP socket implementation**
  - Using event-based socket callbacks
  - Latency of socket calls is minimized

6WINDGate TCP Features Details

- **Available**
  - TCP_SACK and TCP_FACK
  - TCP_QUICKACK
  - Socket options to retrieve/Set TTL, MSS, TOS, DF bit
  - Reno, New Reno
  - ECN support (RFC 3168)
  - TCP Protection against wrapped sequence number
  - TCP Appropriate Byte Counting (RFC 3465)
  - TCP Segment Offload (TSO) support
  - Window Scaling
  - L2 bridge hook for transparent proxy
  - UDP transparent proxy
  - TLS v1.2/v1.3 support
  - DTLS v1.2/v1.3
- **20/Q1**
  - Per socket rate-limit (SO_MAX_PACING_RATE)
  - Initial congestion window per route (initcwnd)
  - Cubic congestion algorithm
  - TCP early retransmit (RFC 5827)
  - TCP Fast Open (RFC 7413)
  - Optimizations
    - Mbuf clone support: avoid copy on transmit side
    - Bulk API
  - L2 bridge plugin enhancement for Transparent proxy
    - Support L2 flow association with socket (ETH/VLAN)
  - TCP syn cookies
- **20/Q3 and next**
  - PATH MTU discovery and ICMP support
  - Duplicate SACK (RFC 3708), challenge ack limit
  - TCP fast RTO (RFC 5682)
  - Slow Start After Idle (RFC 5681)
  - L2 bridge plugin enhancement for Transparent proxy
    - QinQ
    - VxLAN
    - PPPoE
    - GTP-U
    - L2TPv2/v3
  - Full transparency: Socket creation in connected state