Innovation from Within: The Story of Meta's Host Network Interface and fbnic Driver

2024-07-19

Alex Duyck

Agenda

Why Build Our Own NIC?

The Meta Host Networking Interface

Software Development Status

The fbnic Driver

What Comes Next?

Demo

Why Build Our Own NIC?

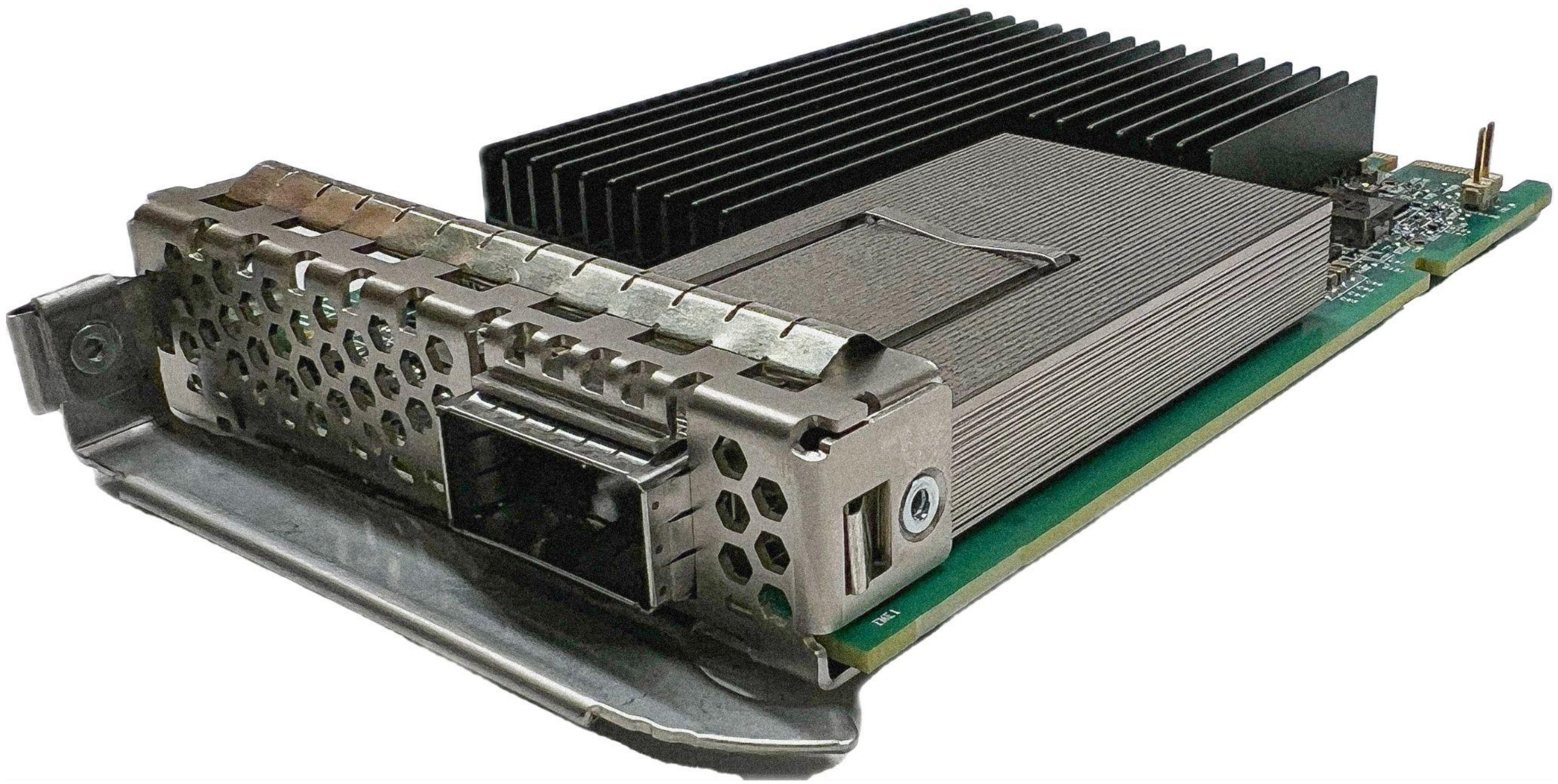
Why build Our Own NIC?

- Supply Chain / Logistics
  - Tighter ownership of the supply chain
  - Ability to multi-source the same hardware
  - Greater control of costs
- Maintenance
  - Goal of reducing mean time to resolve issues
  - Ownership of firmware
  - Access to internals of hardware
- Ability to innovate in areas we are interested in
  - Internal teams to support efforts
  - Overall faster iteration times
- “Owning Our Own Destiny”

The Meta Host Networking Interface

The Meta Host Network Interface

- Our Own Foundational NIC Designed by Meta for Meta
  - Feature set focused on Meta's use cases
  - Multihost NIC which we iteratively designed
    - A single host QEMU/driver in H1 2021
    - A single host FPGA in H1 2022
    - Multi-host NIC ASIC in H2 2023
  - Plan for production use by 2025



Meta’s Use Cases

- Meta is primarily data center focused
  - No virtualization use cases (currently)
    - Containers vs VMs
  - Little if any IPv4 on the network
  - IPv6/IPv6 tunnels
  - Per-host traffic isolation
  - BMC Support / No wake-on-lan
  - Congestion avoidance
  - Header data split
    - Support future use cases such as 4K page flipping
    - Can be reused for things like Device Memory TCP
  - Small average packet size, but exploring jumbo frames

Software Development Status

Software Development Status

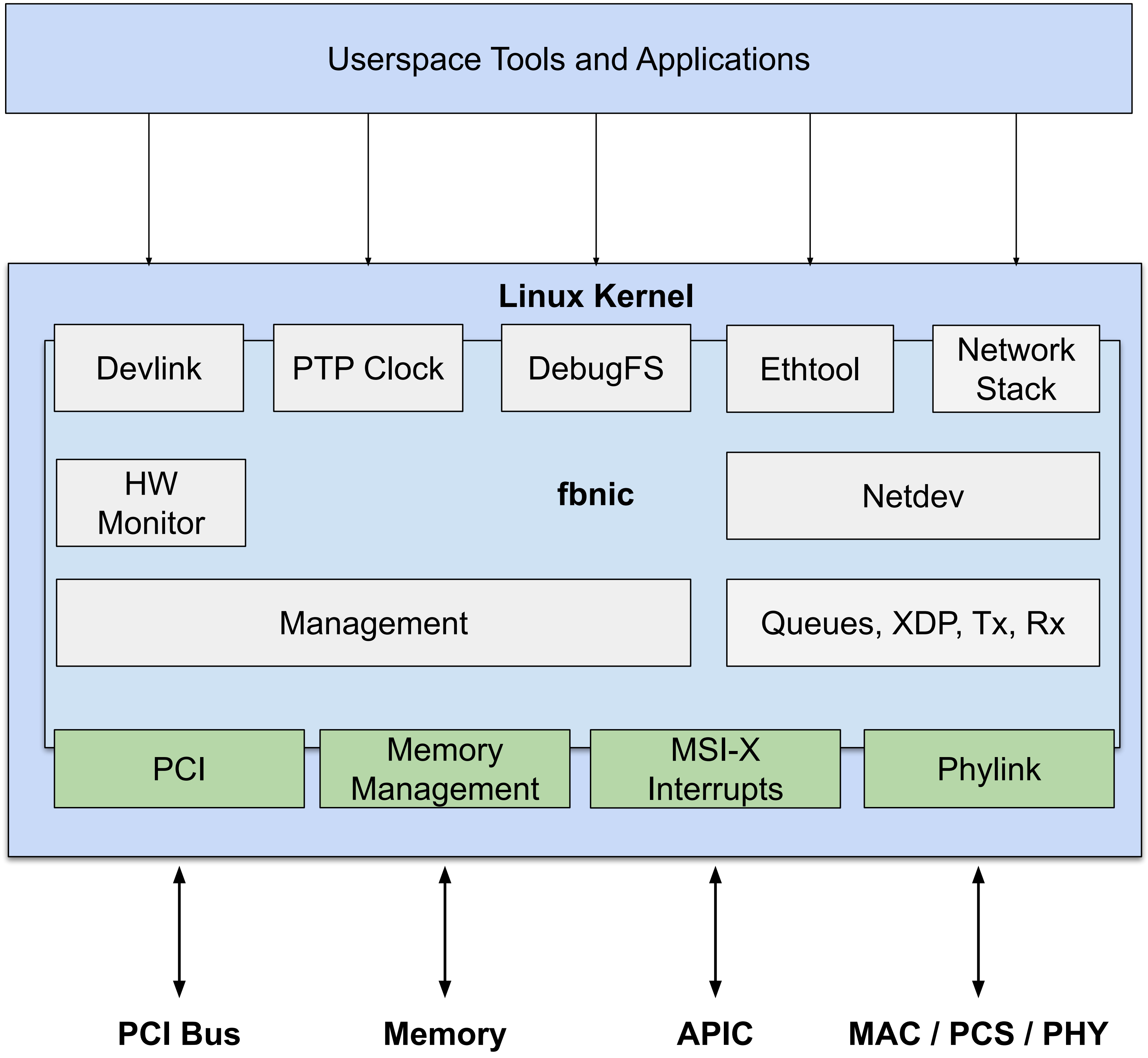
- Linux Drivers (fbnic)
    - Out-of-tree Driver
        - Lead vehicle for development and iteration
        - Ongoing testing on multiple platforms
    - In-Kernel Driver
        - Submitted first version of fbnic and got a story on LWN
            - https://lwn.net/Articles/969383/
            - v5 was accepted 07/15
                - Minimal driver that can load, link, and pass traffic
                - Should be included in 6.11 kernel
        - Will be used in production
- UEFI Driver
    - Successfully provisioning systems w/ minimal issues

Software Development Status

- Control FW
  - Ongoing development to debug I2C and Ethernet PHY issues
- QEMU
  - Host /w NIC
  - FW
  - BMC
  - All can be interconnected to provide full ecosystem
- CI / CD Framework
  - LNDT

The fbnic Driver

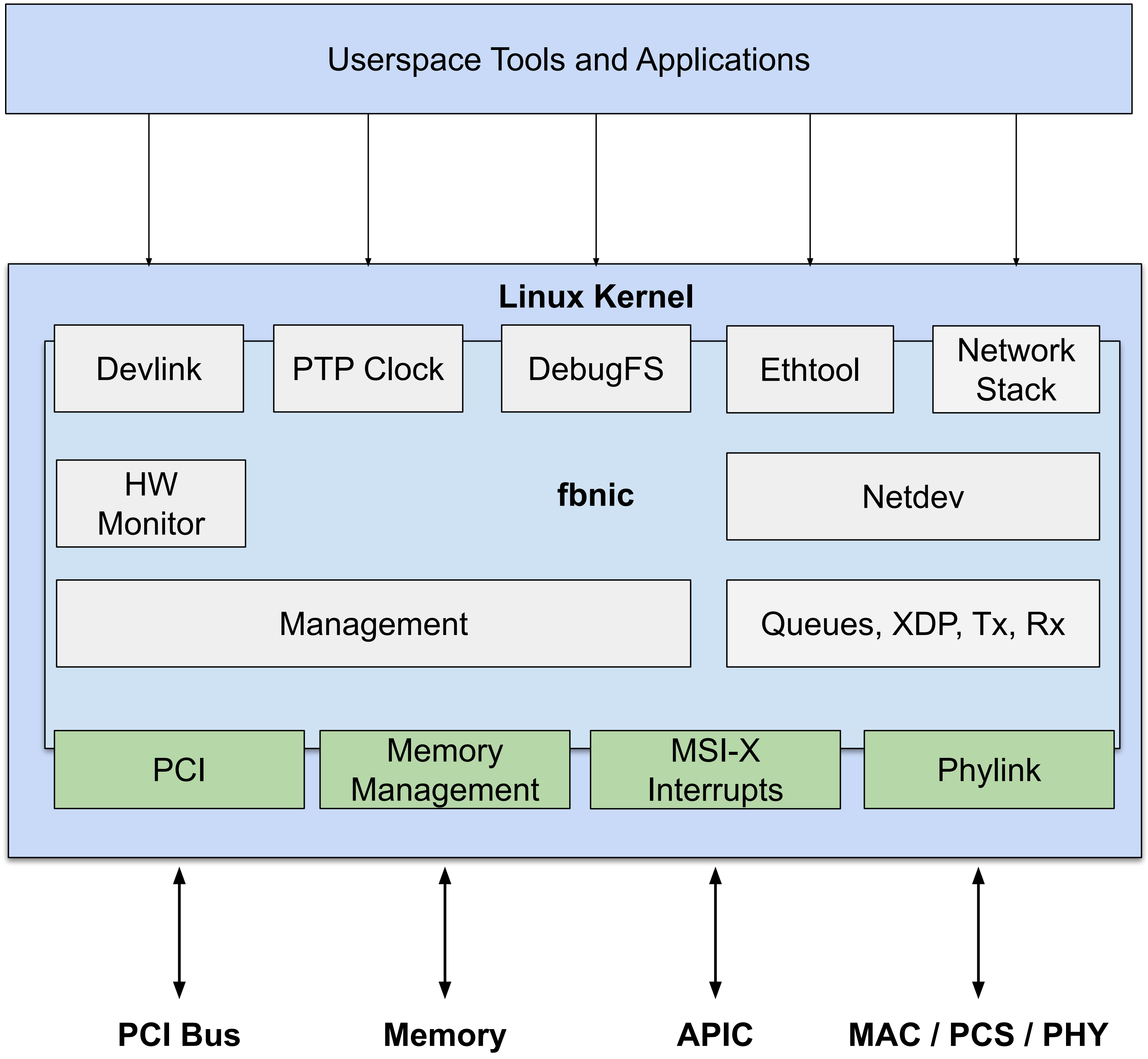
fbnic Linux Driver Architecture



Highlights:

- Devlink health reporter(s)
- DebugFS interface
- 128 Rx queues
- 256 Tx queues
  - 128 XDP Tx, 128 Tx
- Focus on software based offloads such as XDP and GSO Partial
- Allocate before change (ethtool -G/L)

fbnic Linux Driver Architecture



Highlights:

- Header queue / Payload queue
- Page pool w/ fragment based Rx
- Payload packing
- 8B purpose driven descriptors
- ECN support for internal Rx FIFOs
- EDT support for Tx queues
- Timestamping all Tx/Rx packets
- Consolidated write to enable interrupts and set coalescing values

What Comes Next?

Fbnic Future Patch Sets

- Devlink
  - Flash Update, Health Reporter, Info
- Ethtool
  - Interrupt moderation configuration
  - Register dump
  - EEPROM read/write
  - Ring size, count configuration
  - Rx classifier configuration
  - Self test
- HWMON
  - Voltage and Temperature Interfaces
- Debugfs
  - Support for dumping TCAMs and descriptor rings
- LED Configuration
- Phylink Support
  - CGMII, FEC, Multi Lane Support, Bit Error Rate Test, Loopback
- Tx/Rx Completion
  - TSO, USO, GSO_PARTIAL

Features For The Next Meta Host Network Interface

- Feature Set
  - HW GRO
    - Lightweight version of HW GRO
    - Context hints
  - Jumbogram Segmentation / GRO
  - MAC/PSC/Phy Cleanup
    - Migration to more phylink friendly model
  - PSP
  - SIOV

We’re Hiring!

- https://www.metacareers.com/jobs/1129307131655785/

Demo

Questions?