

# Jacinto™ Automotive Processor

## Jacinto 7 Ethernet Switch

TI Confidential – NDA Restrictions

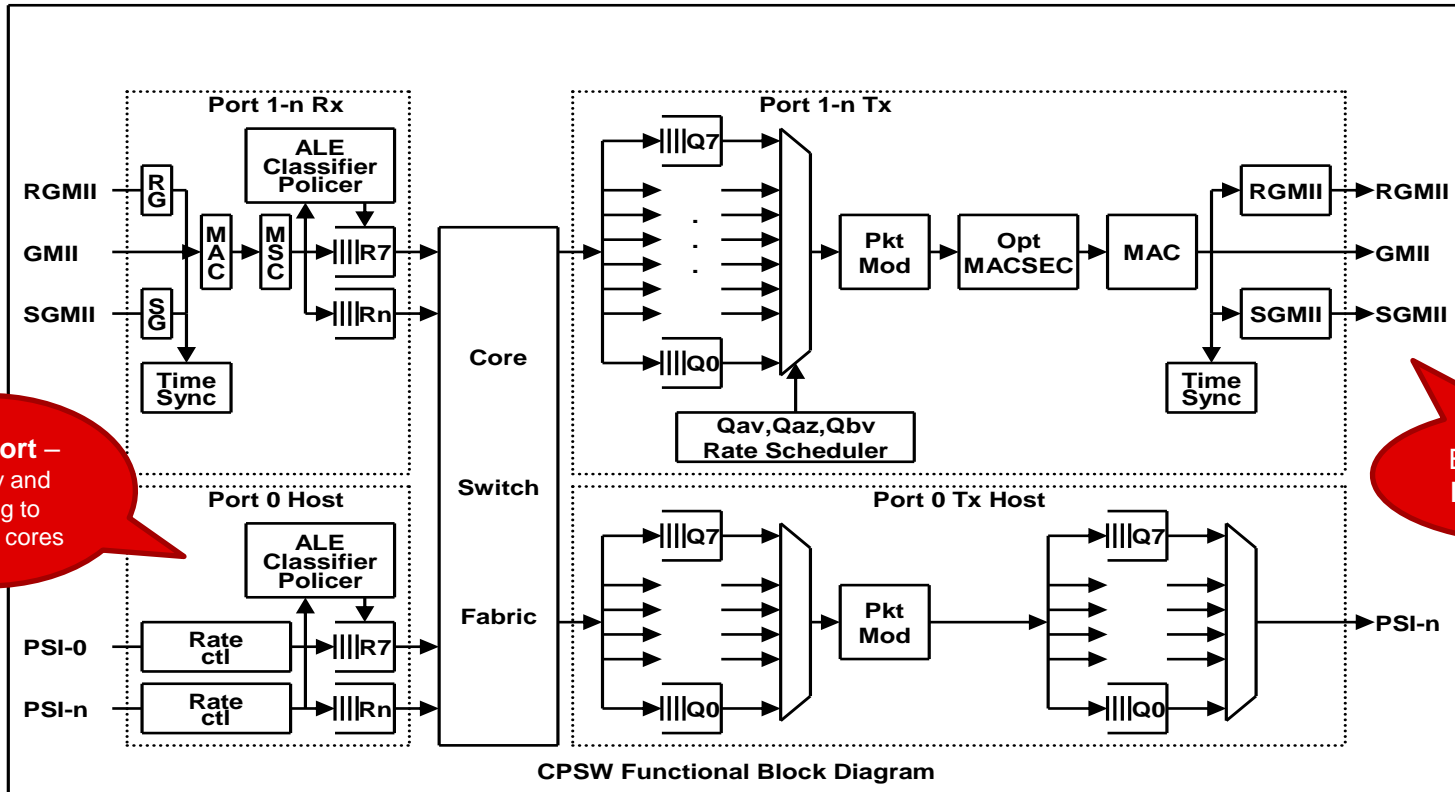
# Agenda

- Ethernet Switch IP Features Overview
- Deep Dive – Software Architecture
  - CPSW LLD
  - Ethernet Firmware
- EthFw Demos
- Jacinto 7 EVM – CPSW support
- Deep Dive – EthFw
  - Folder Structure
  - Docs organization
  - Sample Examples
  - CPSW and PHY integration
  - Switch configuration & Debugging

IP Overview

# ETHERNET SWITCH

# Ethernet Switch – CPSW Functional Block Diagram

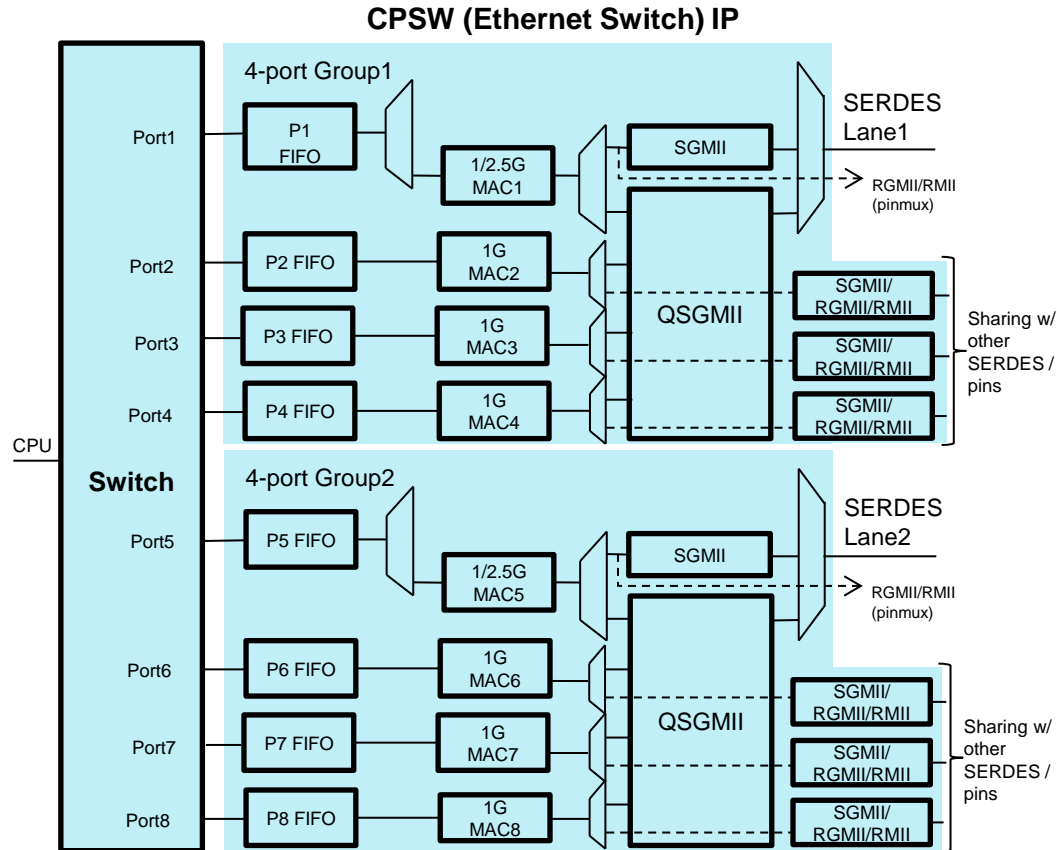


TI Confidential – NDA Restrictions

# Ethernet Switch – Feature Overview

## Top Level Features

- 9 Ports – 8 MAC Ports, 1 Host Port
- Support for RMII, RGMII, SGMII, QSGMII
- Packet Classification (64 classifiers) & filtering using L2/VLAN/L3
- InterVLAN routing support
- UDP/TCP Checksum offload
- Multicast/broadcast rate limiter
- Reset isolation
- Port Mirroring, Port Trunking



# Ethernet Switch – Auto Use-cases entitlement

## Security

- L2/L3 address whitelisting, L2 address blacklisting
- Advanced classification and Policing based on L2/L3 headers using Port number, Priority, ONU, DA, SA, VLAN inner/outer, EtherType, IPSA and IPDA
- Policing for rate limiting matched flows – three color marking on policed rates with drop controls
- Deep packet inspection via integrated compute cores

## Safety Features

- SECDED ECC protection of table entries
- Ram error detection and correction (ECC) with full end to end packet protection (CRC)

## AVB/TSN Features

- IEEE802.1Qav – Egress AVB rate shaping
- IEEE802.1Qbb and IEEE802.3x Flow control
- TSN Features
  - IEEE802.1Qbv/1Qaz time base scheduling
  - IEEE802.3br/IEEE802.1Qbu Frame Preemption

# Ethernet Switch – Feature Summary

## Key Building Blocks

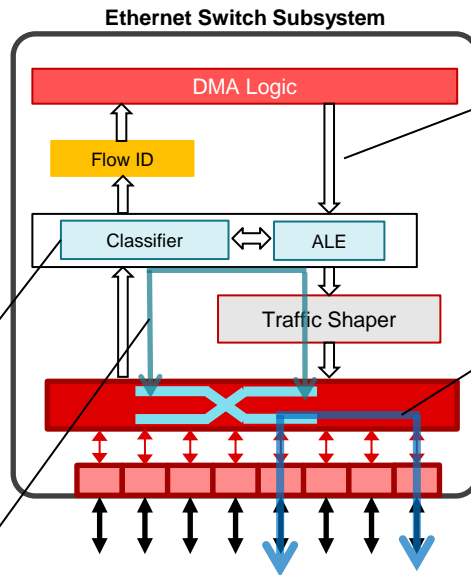
- ALE with 1k entries
- Classifier module for L2/L3 header inspection (96 HW classifier rules)
  - Ingress port: Rate-limit, filter
  - FlowID assignment for Host traffic
- AVB, TSN features support
  - Traffic shaper
- L2/L3 HW Checksum offload
- Inter-VLAN routing in HW

## Traffic to Host DMA port

- **Classifier** for header inspection
- **FlowID** for packet steering and multiple queue support

## Traffic to Host Inter-VLAN routing

- **ALE match**: traffic match to host port
- **Classifier**: Flow matching for HW offload
- **HW offload** for matched flow : packet header modification in HW



## Transmit from Host

- **Classifier** support for rate-limit per flow
- **TSN support**: Time aware scheduling, pre-emption, Traffic Shaping with 8 HW Priority queue/port
- **Per-port** priority regeneration support

## Forwarding Between Ports

- VLAN-aware line rate L2 switching
- **Per-flow** ingress port rate limiting, filtering
- **Per-port** broadcast, multicast rate limit

## Firewall, Filter

- L2-based filter (OUI deny, MAC authentication)
- Classifier based L2/L3 filtering
  - No match drop
  - MAC multicast range, IP CIDR mask filtering
- HW filter for untagged, IPV4, dual tag, IP fragment, L3 Next Header check

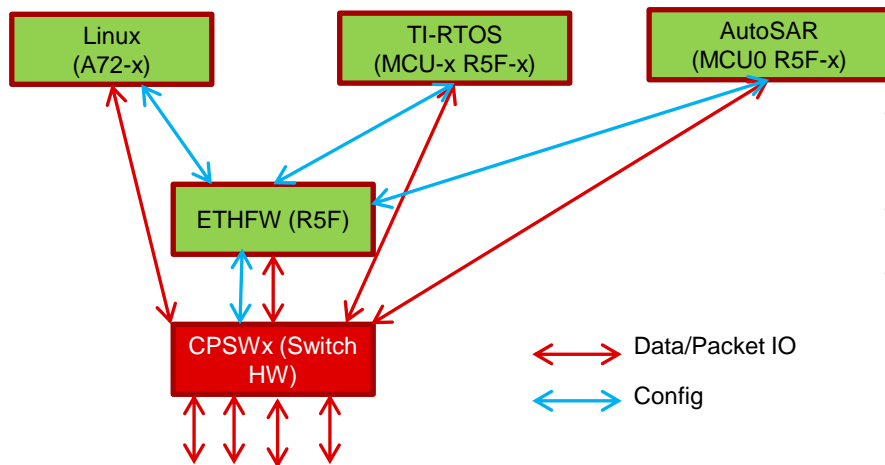
Deep Dive –Software Architecture

# ETHERNET SWITCH



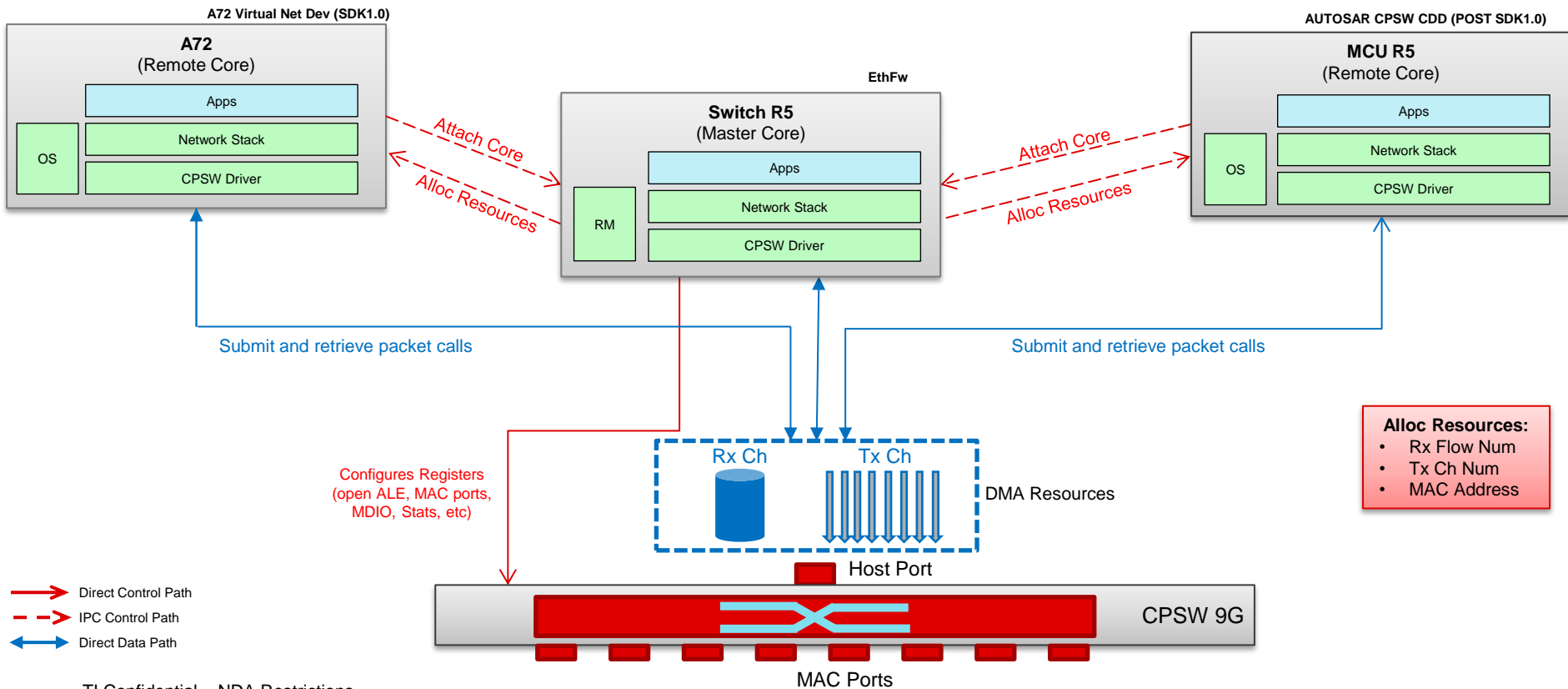
# Ethernet Switch – Data Flow

Ethernet Firmware Supporting Multiple Clients



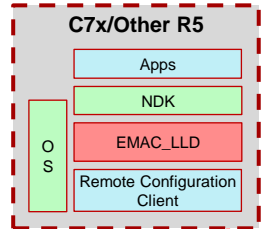
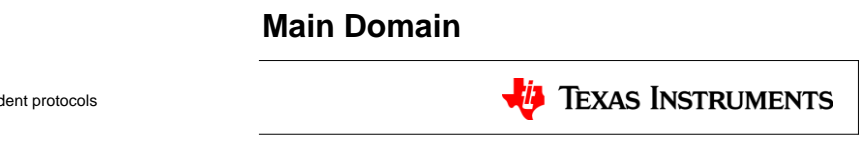
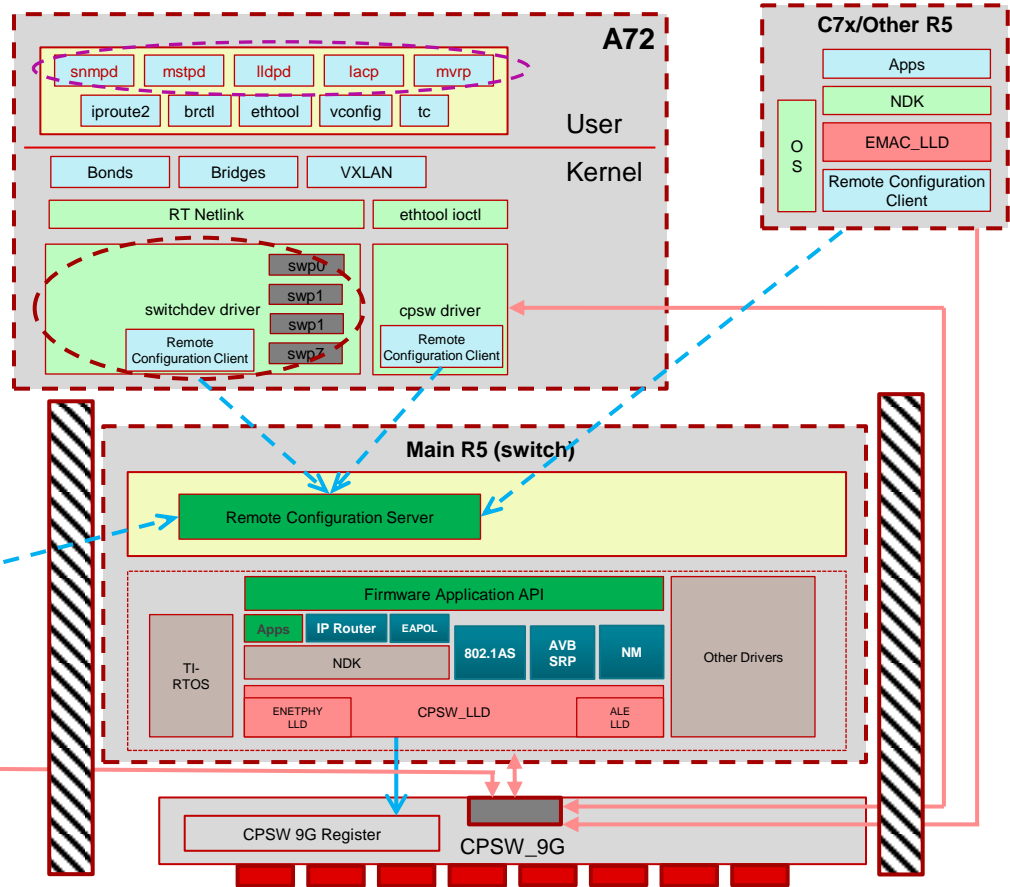
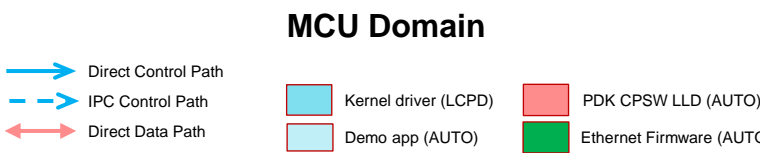
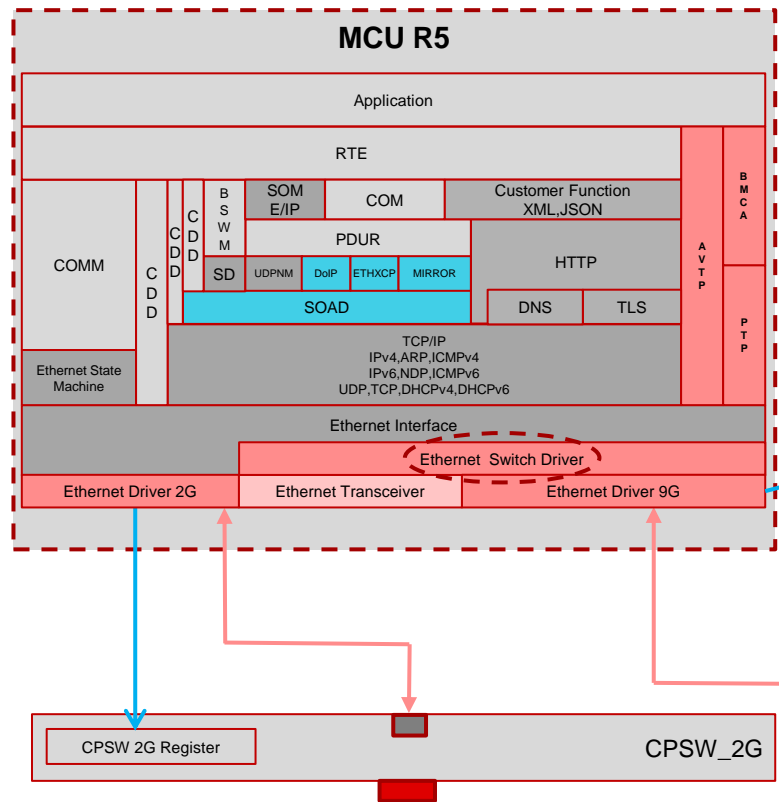
- Multiple CPUs on the SoC can perform network IO simultaneously using the switch once the switch is configured via the FW
- Subsequent packet IO can happen directly between the CPU specific network stack and switch HW via UDMA
- Switch configuration

# Ethernet Switch – Data Flow Detailed

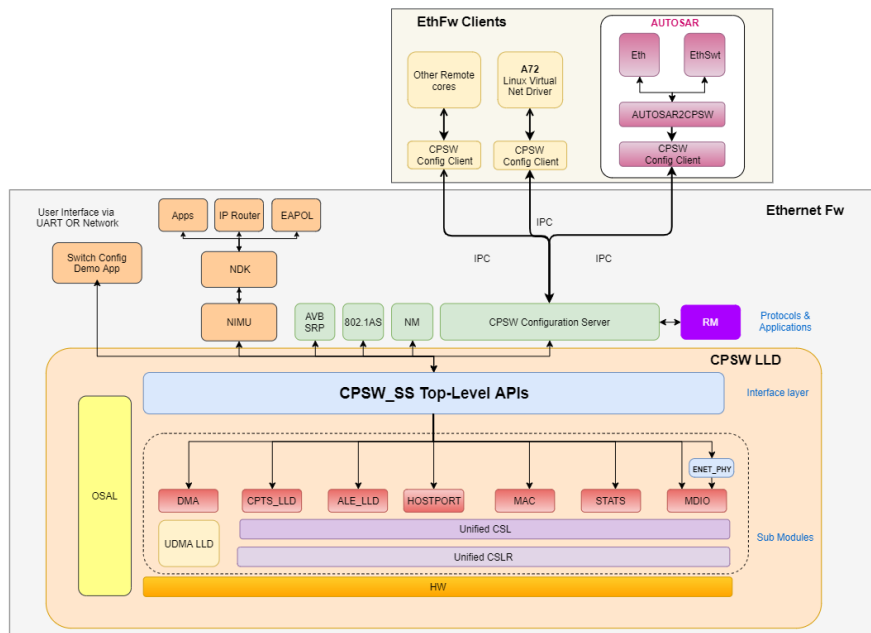


TI Confidential – NDA Restrictions

# Ethernet Switch – Ref. AUTO Use-case



# Ethernet Switch – Software Stack



EthFw Software Stack

## EthFw Submodules

- **CPSW LLD** – driver layer for CPSW9G IP
- **CPSW Remote Configuration Server**
  - CPSW Remote Configuration RTOS Client
- **Switch Resource Manager**
- **Switch Resident Protocols**
  - TCP/IP NDK
- **Switch Configuration App**

Software Releases

# ETHERNET SWITCH

# Ethernet Switch – EthFw 1.0: IS / IS NOT

## SDK 1.0

- Basic Switching
- VLAN
- RGMII
- 1G/100M
- Host Port Packet Tx/Rx
- Switch R5 as packet source/sink
- IPC based switch configuration from other cores
- Linux virtual netdev driver integration
  - A72 data path
- TCP/IP stack (NDK)
- Multicast switching
- InterVLAN routing
- Network Security
- Classifier/Classifier
- QoS/Packet Priority regeneration
- Traffic Shaping

## Post SDK 1.0

- Timesync (802.1ASrev)
- Time aware scheduling (802.1Qbv) - EST
- SGMII
- RMII
- QSGMII and 2.5 Gbps support
- Port Mirroring
- Port Trunking
- IET (Interspersing express traffic)
- ENET PHY – power mgt
- Switch compliance testing
- Time sensitive networking

OS	CPSW 2G (Direct driver)	CPSW 9G (Virtual driver)
TI RTOS	Y	SDK 1.0
Linux	Y	SDK 1.0
AUTOSAR	Y	Post SDK 1.0
QNX	SDK 1.0	Post SDK 1.0

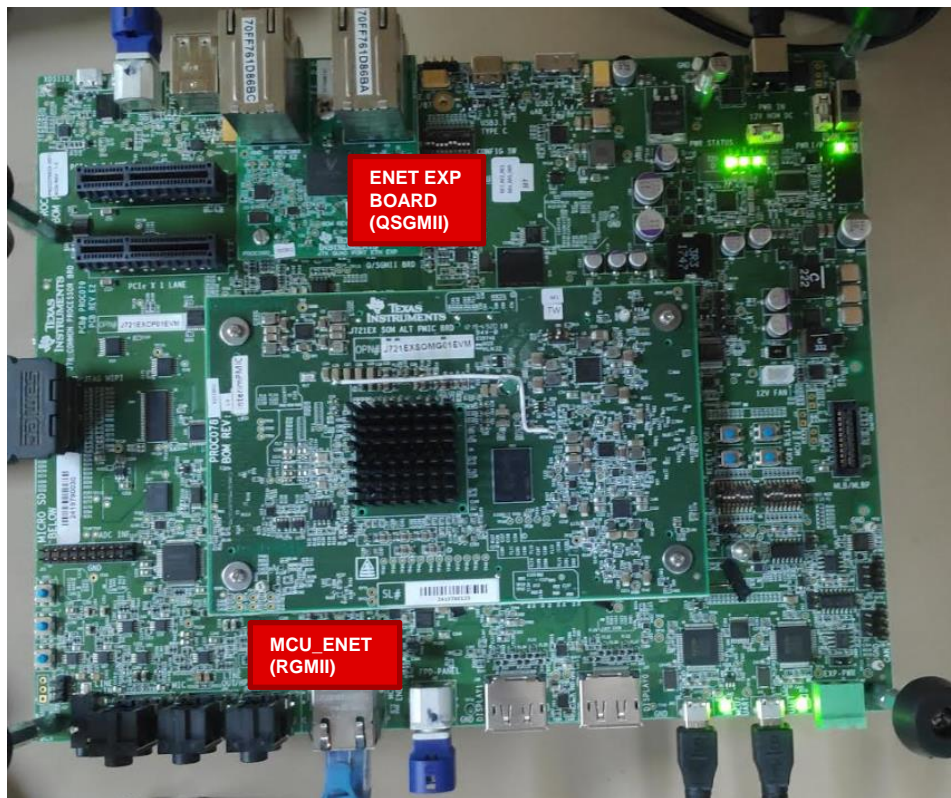
Jacinto 7 EVM: CPSW Support

# ETHERNET SWITCH

# Ethernet Switch – Jacinto 7 EVM Baseboard

- **MCU\_ENET**
  - DP83867 RGMII PHY
  - 1 Gbps
  - CPSW 2G MDIO bus
- **ENET\_EXP DB**
  - 1 x QSGMII Expansion Daughter Board
  - Not supported as SDK1.0

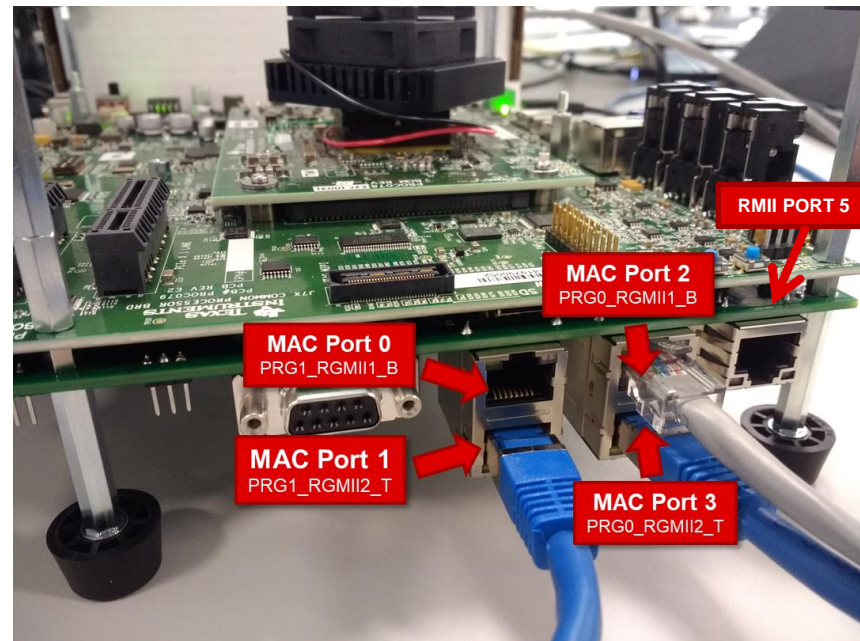
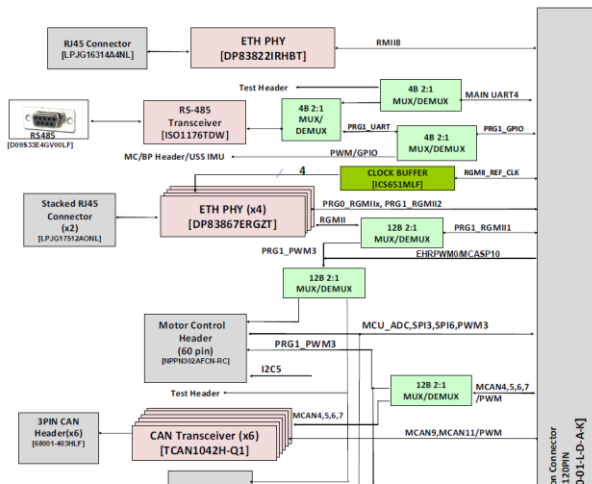
With SDK1.0 only CPSW2G supported on Baseboard, GESI DB is needed for CPSW9G





# Ethernet Switch – GESI Board

- 4 x RGMII PHYs
  - DP83867
  - 1 Gbps
  - CPSW 9G MDIO bus
- 1 x RMII PHY (Post SDK1.0)
  - DP83822
  - 100 Mbps
  - CPSW 9G MDIO bus

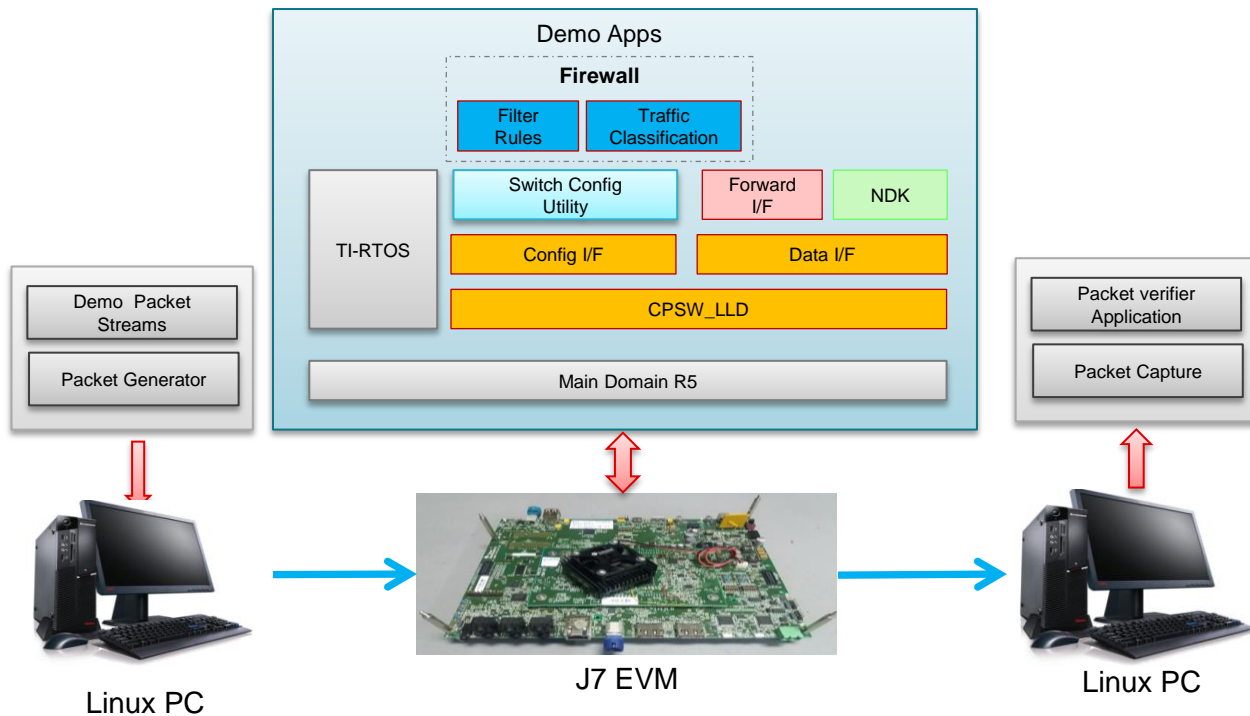


TI Confidential – NDA Restrictions

Demos

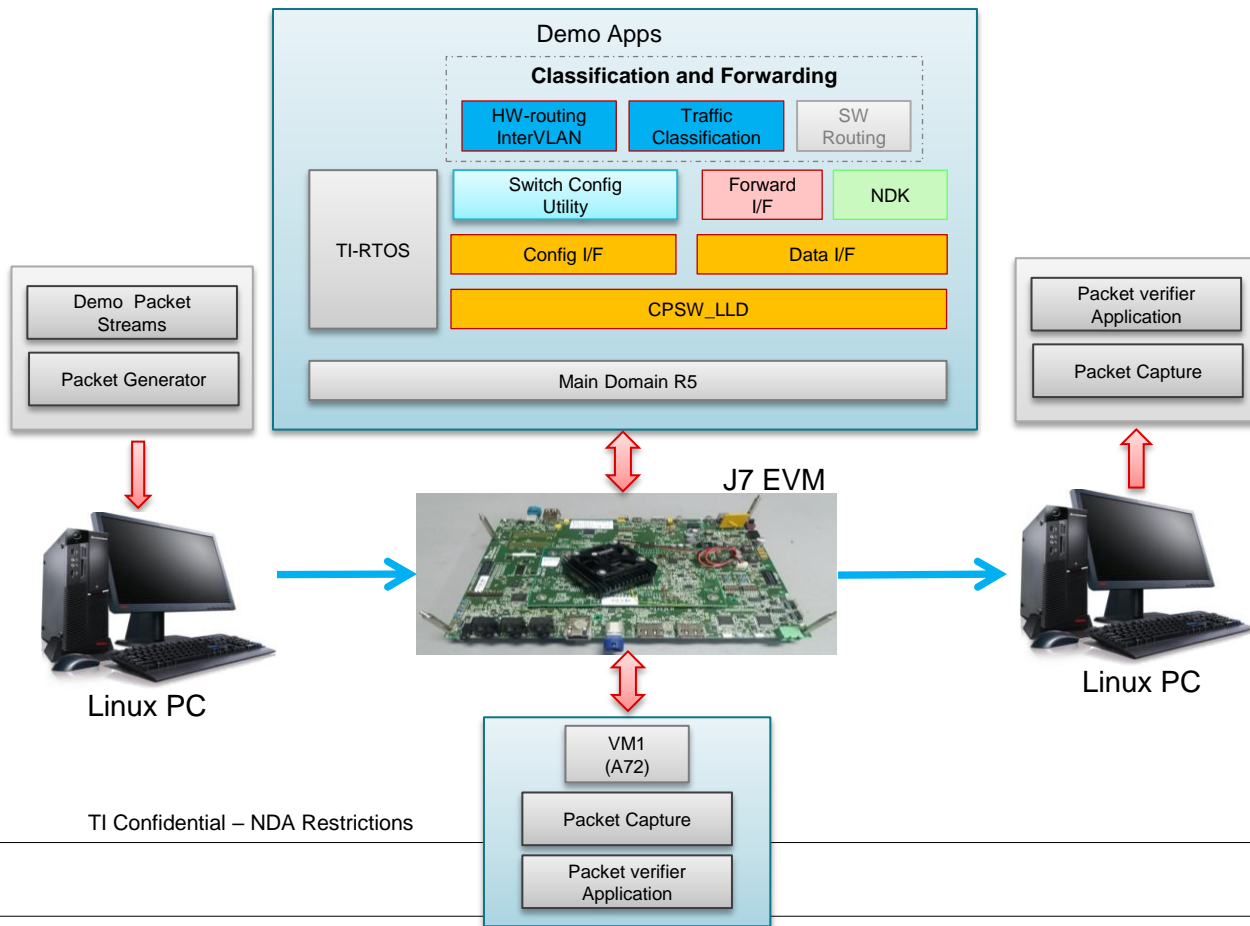
# ETHERNET SWITCH

# Ethernet Switch – Firewall



- **Demonstrate**
  - Packet classification Network security, firewall
    - Filter on L2 header: OUI, SA, VLAN
    - Filter on L3 header: frag, SA/DA
- **Differentiation**
  - Network security based on L2/L3 header fields (wire rate)
- **Timeline: SDK 1.0**

# Ethernet Switch – Classification and InterVLAN Routing



- **Demonstrate**

- Packet classification
  - L2 header: addr, EtherType, VLAN, priority, port number
  - L3 header: SA/DA
- InterVLAN routing
  - Wire rate, up-to 4 routes

- **Differentiation**

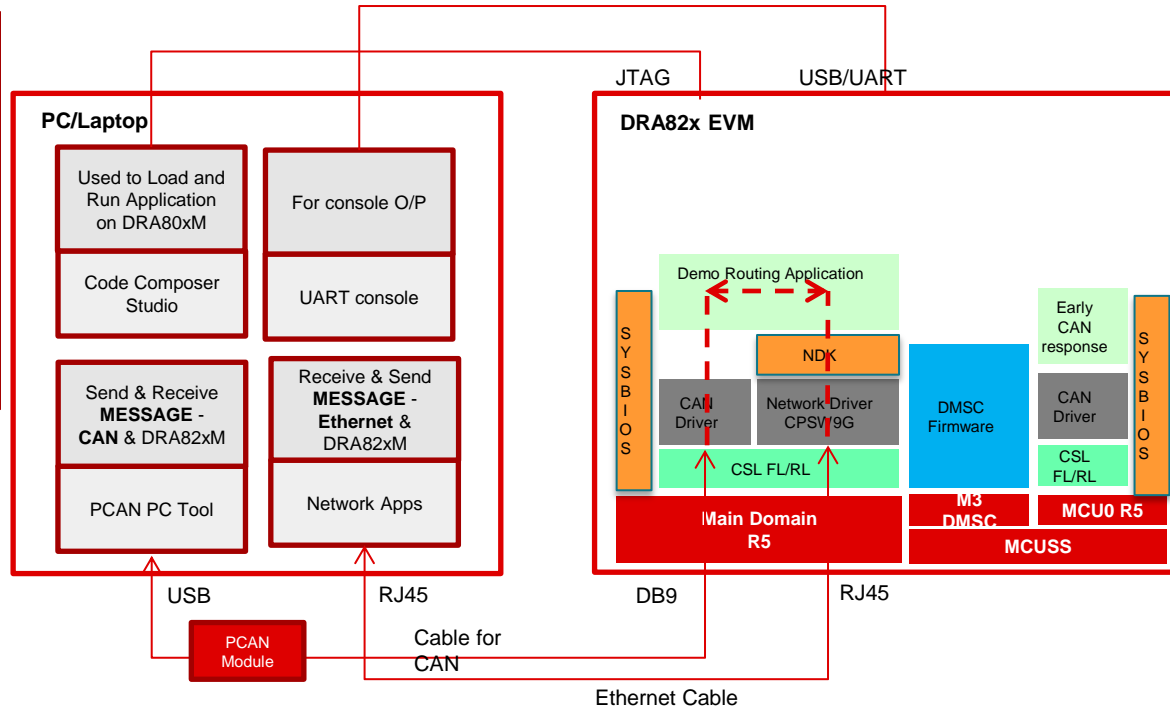
- Ethernet sharing across multiple cores / VMs
- HW packet header inspection/classification
- InterVLAN routing in HW at line rate

- **Timeline:** SDK 1.1

# Gateway Demo: CAN ↔ Ethernet

- Demonstrates Latency and CPU Load for routing application
- Shows early CAN response
- AUTOSAR MCAL driver
- RTOS driver for CAN + Linux for ETH:

- Routing SW using just R5F – Cortex A free for other application usage
- <100us driver routing latency

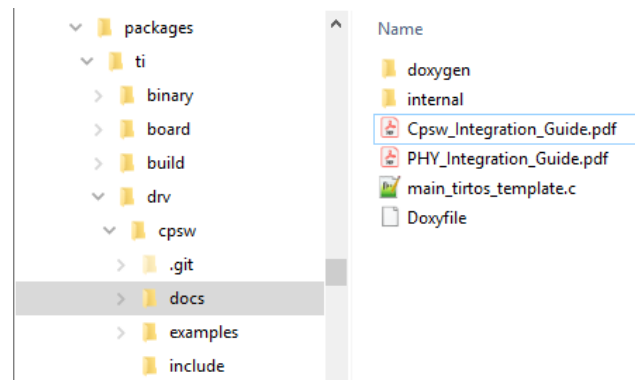


Deep Dive – Ethernet Firmware

# ETHERNET SWITCH

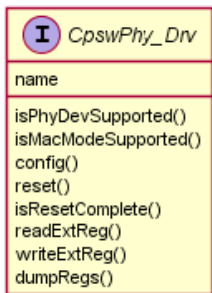
# Ethernet Switch – CPSW Integration

- Integration into app:
  - CPSW configuration parameters: ALE, MAC, Host, MDIO
  - DMA configuration parameters
  - Print/tracing functions
  - CPSW init and open
  - Resource Manager attach
  - IOCTLs and helper functions
  - MAC/PHY configuration
  - Close sequence
- NDK integration C-file template
- CPSW LLD examples: loopback, sanity, NIMU



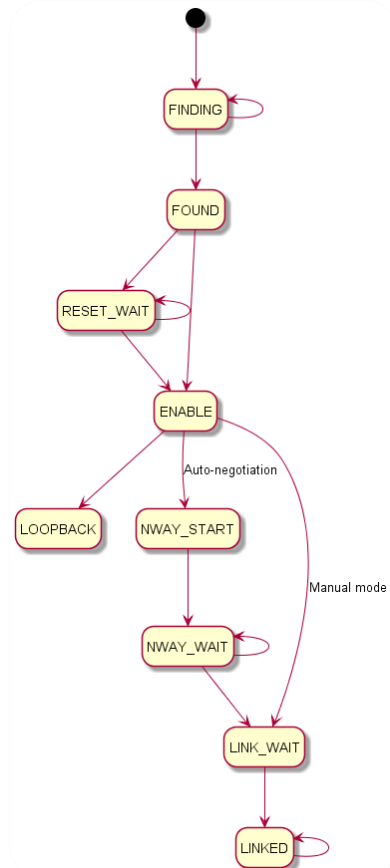
# Ethernet Switch – PHY Integration

- PHY abstraction layer as part of CPSW LLD
  - State machine based
  - PHY-specific drivers can implement their own callbacks or reuse generic implementation



```
CpswPhy_Drv MyPhy_phyDrv =  
{  
    .name = "My PHY driver",  
    .isPhyDevSupported = MyPhy_isPhyDevSupported,  
    .isMacModeSupported = MyPhy_isMacModeSupported,  
    .config = MyPhy_config,  
    .reset = MyPhy_reset,  
    .isResetComplete = MyPhy_isResetComplete,  
    .runComplianceTest = NULL,  
    .readExtReg = GenericPhy_readExtReg,  
    .writeExtReg = GenericPhy_writeExtReg,  
    .dumpRegs = MyPhy_dumpRegs,  
};
```

- Runtime device to driver mapping based on PHY ID
- Auto-negotiation and manual mode
- Clause-22 support
- “PHY Integration Guide” document provided as part of CPSW docs





Switch Configuration and Debugging

# ETHERNET SWITCH

# Ethernet Switch – Switch Configuration

- Default configuration loaded during EthFw init – part of
- Run-time switch configuration supported  
Switch Config via Network (config GUI) & UART

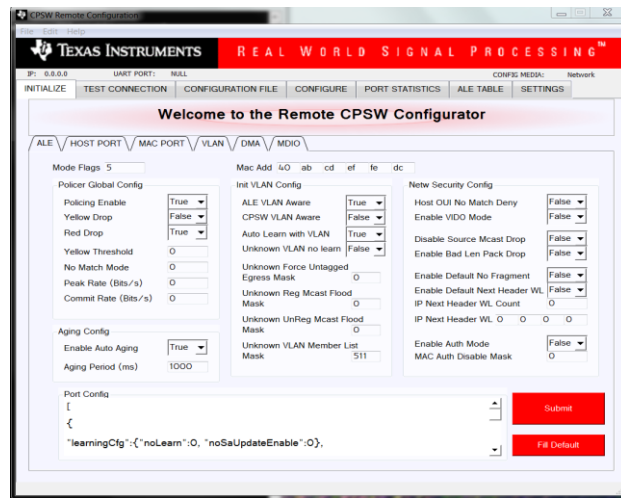


Fig – Switch Config GUI

```
=====
                        EthFw L2 Switching APP
=====
Host MAC address: 04:01:02:03:04:05
CPSW_9G Test on MAIN NAVSS
PHY 0 is alive
PHY 3 is alive
PHY 12 is alive
PHY 15 is alive
PHY 23 is alive
[NIMU_NDK] CPSW has been started successfully

CPSW NIMU application, IP address I/F 1: 192.168.1.108

=====
                        Switch Options
=====
1. Enable/Disable VLAN
2. Enable/Disable Multicast
3. Enable/Disable Rate Limiting
4. Enable/Disable InterVLAN
5. Print ALE & Policer Table
Enter your choice:
root@navss:~#
```

Fig – UART Based Config

# Ethernet Switch – Debugging

- Debugging Infrastructure
  - GEL Scripts
    - HW error stats report.
    - Dump Switch forwarding table
    - PHY reg read/write configuration
  - Debug and diagnostic APIs/IOCTLs
    - DMA statistics
    - Switch config dump IOCTLs
  - Runtime configurable trace support
  - GUI support to run stress tests, get switch statistics.

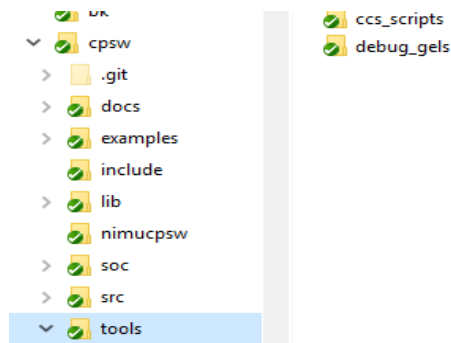
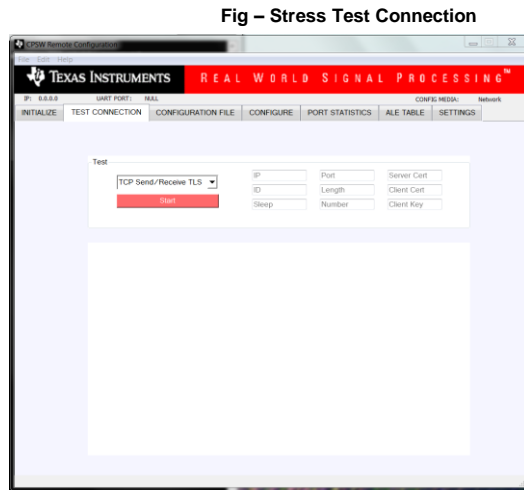


Fig – UART Based Config

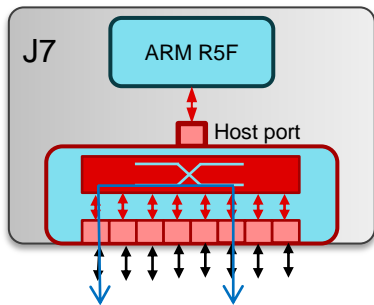
# Q & A

# BACKUP

TI Confidential – NDA Restrictions

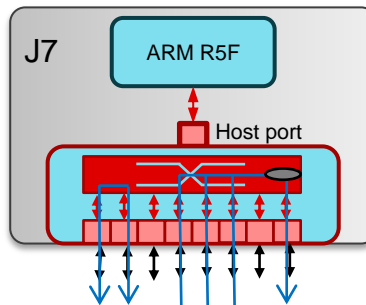
# Ethernet Switch – Key Usecases

## L2 Switching Usecase



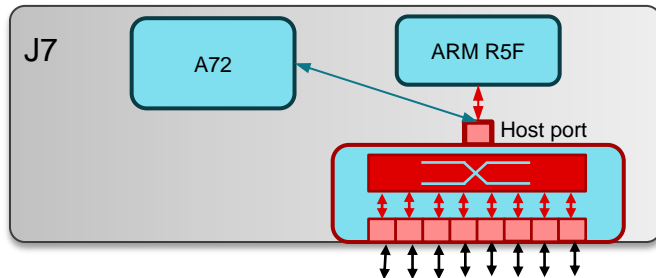
Traffic from any external port to any external port

## L2 Aggregator Usecase



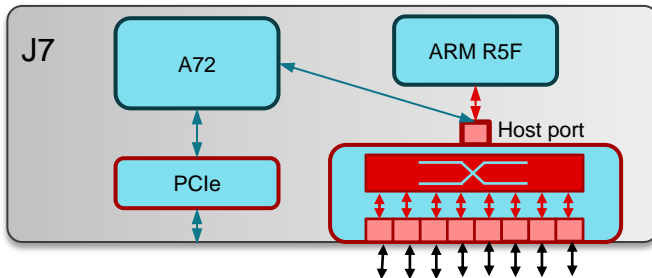
Traffic from multiple external ports aggregated and sent over external port

## L2 Switching with Traffic to/from Compute Cores



Traffic from any external port to Jacinto 7 compute cores

## PCIe / Ethernet Bridge



# Ethernet Switch – Integrated Switch Key Value Propositions

Use-case	Key Features	Software Enablement
<b>Gateway – Network Security</b>	<ul style="list-style-type: none"><li>• IP address whitelisting</li><li>• Protocol whitelisting</li><li>• Rate limiting</li><li>• Traffic shaping</li><li>• MAC Authentication</li></ul>	<ul style="list-style-type: none"><li>• ALE driver with full IP entitlement</li><li>• Support for switch configuration via AUTOSAR/Linux</li></ul>
<b>IO Hub</b>	<ul style="list-style-type: none"><li>• Support for 2-port driver &amp; 9-port driver to connect vehicle network to switch</li></ul>	<ul style="list-style-type: none"><li>• Support for multicore data planes using packet based classifications (flows)</li><li>• Single driver supporting all CPSW IP variants on Jacinto 7, Maxwell, Pascal.</li><li>• Scatter-gather</li></ul>
<b>Central Gateway</b>	<ul style="list-style-type: none"><li>• Programmable L3 routing</li><li>• InterVLAN routing (4 hardware routes)</li><li>• 200ms Ethernet response</li></ul>	<ul style="list-style-type: none"><li>• AUTOSAR MCAL switch dev</li><li>• LCPD Virtual netdev driver</li><li>• TCP/IP stack (NDK integration)</li><li>• Performance optimizations</li></ul>
<b>Ethernet Surround View</b>	<ul style="list-style-type: none"><li>• IET/EST</li><li>• 802.1AS (PTP)</li><li>• IEEE 802.1Qav (Forwarding)</li></ul>	<ul style="list-style-type: none"><li>• Time sync support</li><li>• TSN support</li></ul>

# Ethernet Switch – Release Plan

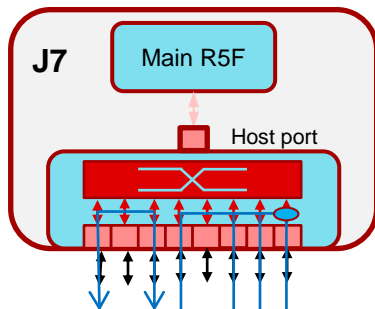
Milestone	CPSW_LLD	EthFw	Demos	Date
<b>SDK 0.8</b>	<ul style="list-style-type: none"><li>• Basic Switching</li><li>• VLAN</li><li>• RGMII</li><li>• 1G / 100M</li><li>• Host TX / RX</li></ul>	<ul style="list-style-type: none"><li>• Switch config via UART</li></ul>		30 March 2019
<b>SDK 0.9 Bring-up</b>	<ul style="list-style-type: none"><li>• Multicast switching</li><li>• InterVLAN routing</li><li>• Ethernet PHY</li></ul>	<ul style="list-style-type: none"><li>• TCP/IP stack (NDK integration)</li><li>• Switch config via telnet</li><li>• A72 remote core support</li><li>• Main R5_1 as packet source/sink</li><li>• Bring-up complete on EVM</li></ul>	<ul style="list-style-type: none"><li>• CAN ↔ Ethernet gateway</li><li>• L2 Switching – multicast, VLAN</li></ul>	30 June 2019
<b>SDK 1.0 RTM</b>	<ul style="list-style-type: none"><li>• Network Security</li><li>• Classifier/Policer</li><li>• QoS/Packet Priority regeneration</li><li>• Traffic Shaping</li></ul>	<ul style="list-style-type: none"><li>• Linux A72 Virtual driver integration</li></ul>	<ul style="list-style-type: none"><li>• Firewall Demo</li><li>• InterVLAN routing and Traffic Classification demo</li></ul>	30 Sep 2019



# Ethernet Switch – Usecases Entitlement

- Latency improvement to meet requirements
  - Direct data path from AUTOSAR R5 to Switch – eliminates need of IPC
  - Plan for TSN features to enable deterministic latency like EST, IET and switch FIFO tuning
- Optimized ETH driver for CPU load
  - Receive and transmit interrupt pacing
  - Support for scatter-gather to enable zero copy transmit
  - L3 checksum offload support
  - Support for enabling InterVLAN routing in hardware
- L3 routing
  - Dedicated switch R5 for Ethernet switch (CPSW9G) control – can be controlled via AUTOSAR or Linux
  - Multi-core support – support direct data path for each core (no IPC during data movement)
    - Support for enabling hardware features to classify and route traffic to intended core
  - Linux Virtual netdev driver & AUTOSAR Ethernet switch (CPSW9G) driver enablement

# Ethernet Switch – Customer Enablement & Demos

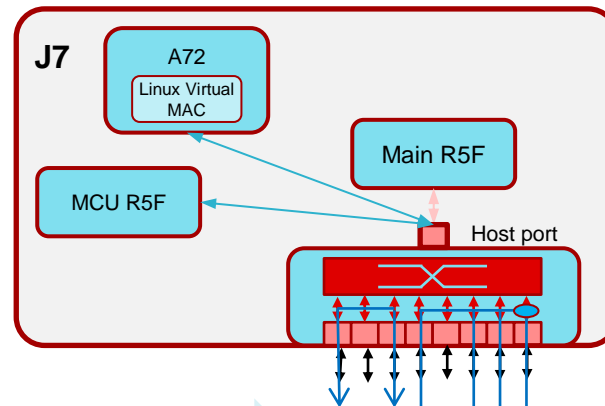


## L2 Switch Usecase

- Traffic from any external port to any external port
- Features:
  - Switch config via UART
  - VLAN
  - Multicast
  - RGMII, 100M / 1G

## L2 Aggregator Usecase

- Traffic from multiple external ports aggregated and sent over external port
- Features:
  - Network security
  - Policer
  - QoS / Packet priority regeneration
  - InterVLAN routing



## L2 Switching with internal cores

- Traffic from multiple external ports aggregated to J7 compute cores
- Features:
  - Traffic shaping
  - Traffic classification
  - TCP/IP (NDK) stack on Switch R5F
  - Packet TX/RX from MCU R5F

Performance

# ETHERNET SWITCH

# Ethernet Switch – Performance

Content to be added

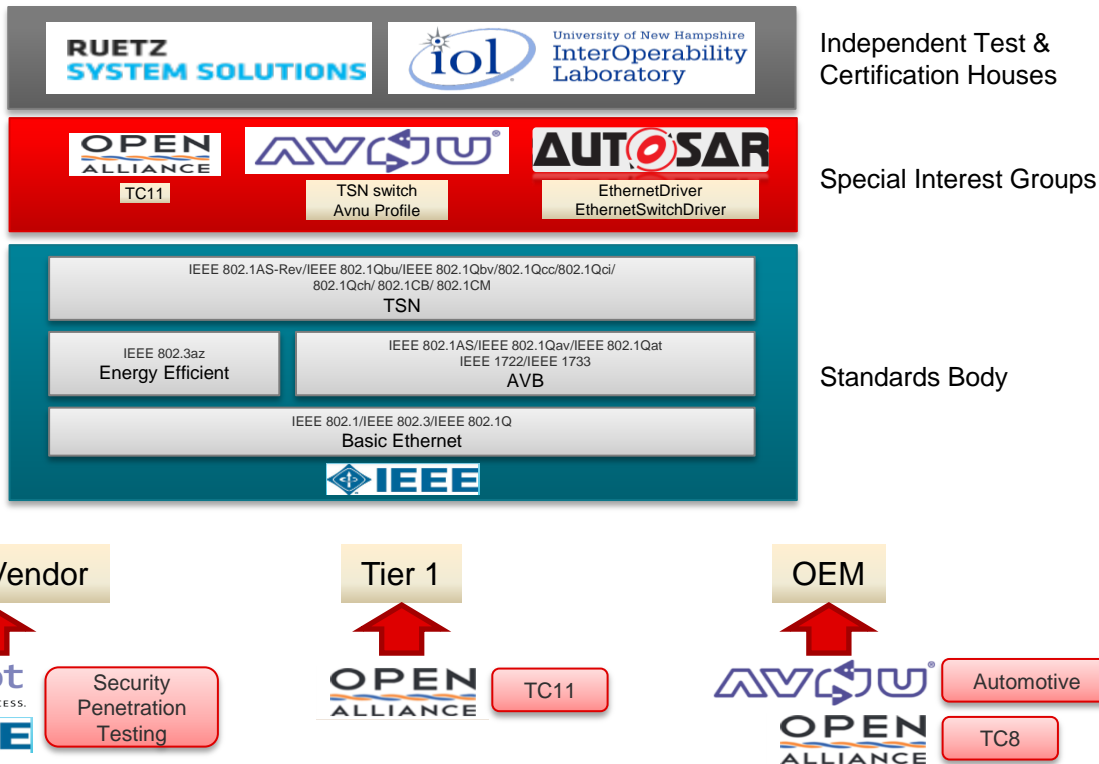
# Ethernet Switch – Performance Measurement

Content to be added – CPU load, Network throughput

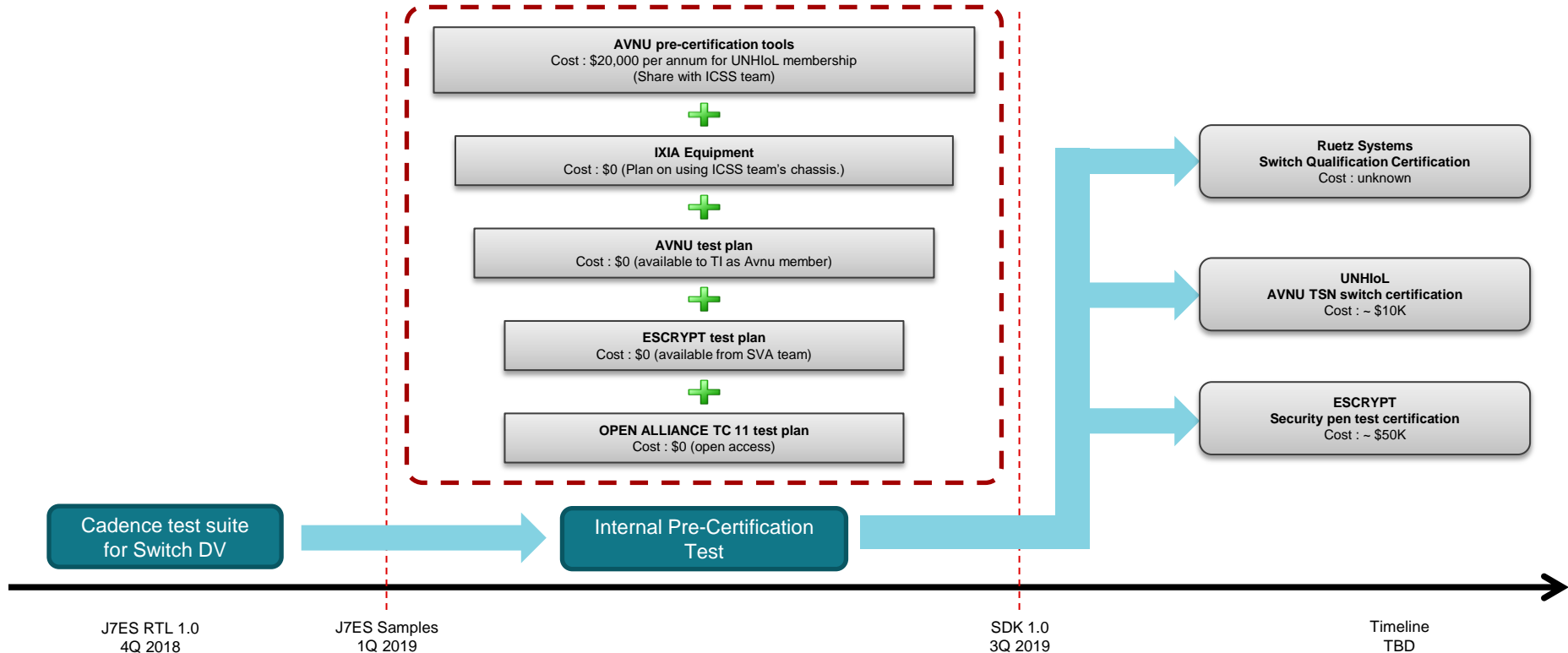
Switch Certification

# ETHERNET SWITCH

# Ethernet Switch – Standards Ecosystem



# Ethernet Switch – Roadmap



TI Confidential – NDA Restrictions



# Ethernet Switch – Linux Virtual Net Driver