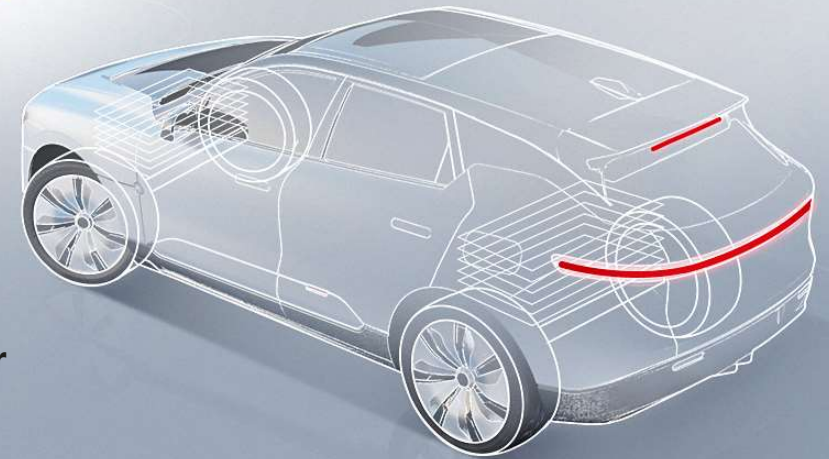


TUM ACADEMIC SALON 2022

The paradigm shift in automotive zonal gatewaying

How to elevate the E/E architecture from good to great: elastic network SoC propelled by HW innovation



Francesc Fons, Angela Gonzalez Mariño, Abdoul Aziz Kane
Automotive Engineering Lab, Huawei Munich Research Center

TUM Garching (Munich), 30.09.2022



Contents

1. [WHY] Automotive Electronic Industry

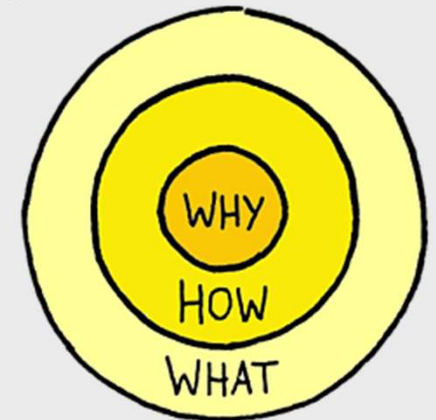
- The Innovation Race
- Pain Points & Challenges

2. [HOW] IVN, Zonal GW Controller & Networking Chipset

- Huawei Automotive IVN Research Group
- Zonal Gateway Controller
- Network SoC Prototyping

3. [WHAT] Elastic Gateway SoC (eGW) Concept

- Related Art on HW/SW Codesign in Automotive
- The 12 HW Innovations on NW & APP Processing of eGW SoC



Contents

1. [WHY] Automotive Electronic Industry

- The Innovation Race
- Pain Points & Challenges

2. [HOW] IVN, Zonal GW Controller & Networking Chipset

- Huawei Automotive IVN Research Group
- Zonal Gateway Controller
- Network SoC Prototyping

3. [WHAT] Elastic Gateway SoC (eGW) Concept

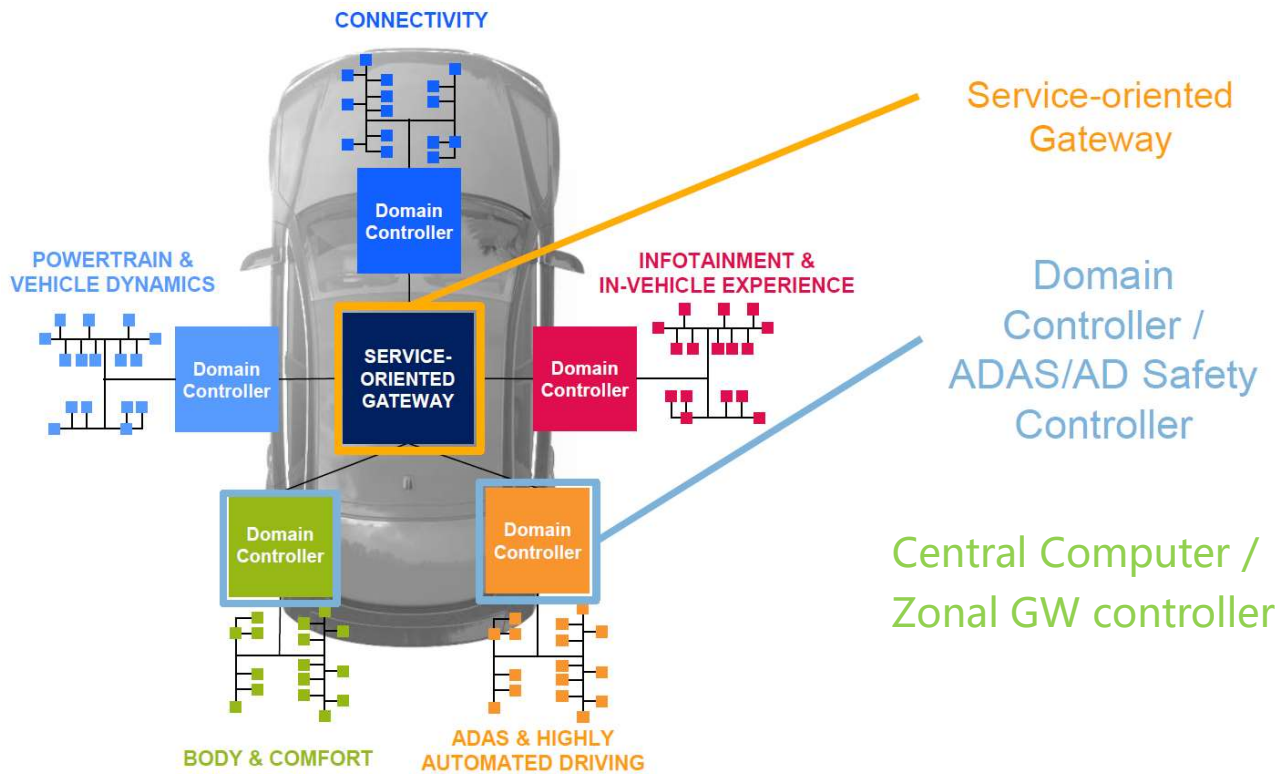
- Related Art on HW/SW Codesign in Automotive
- The 12 HW Innovations on NW & APP Processing of eGW SoC



1.1 Automotive Electronics Industry

Nowadays, the **Automotive E/E Architecture** is being profoundly reshaped, in deep transformation -affecting especially the IVN infrastructure- in order to enable thus the **Autonomous-Connected-Electric-Shared (ACES) vehicle of tomorrow**

From **DOMAIN-BASED E/E IVN ARCHITECTURE...**



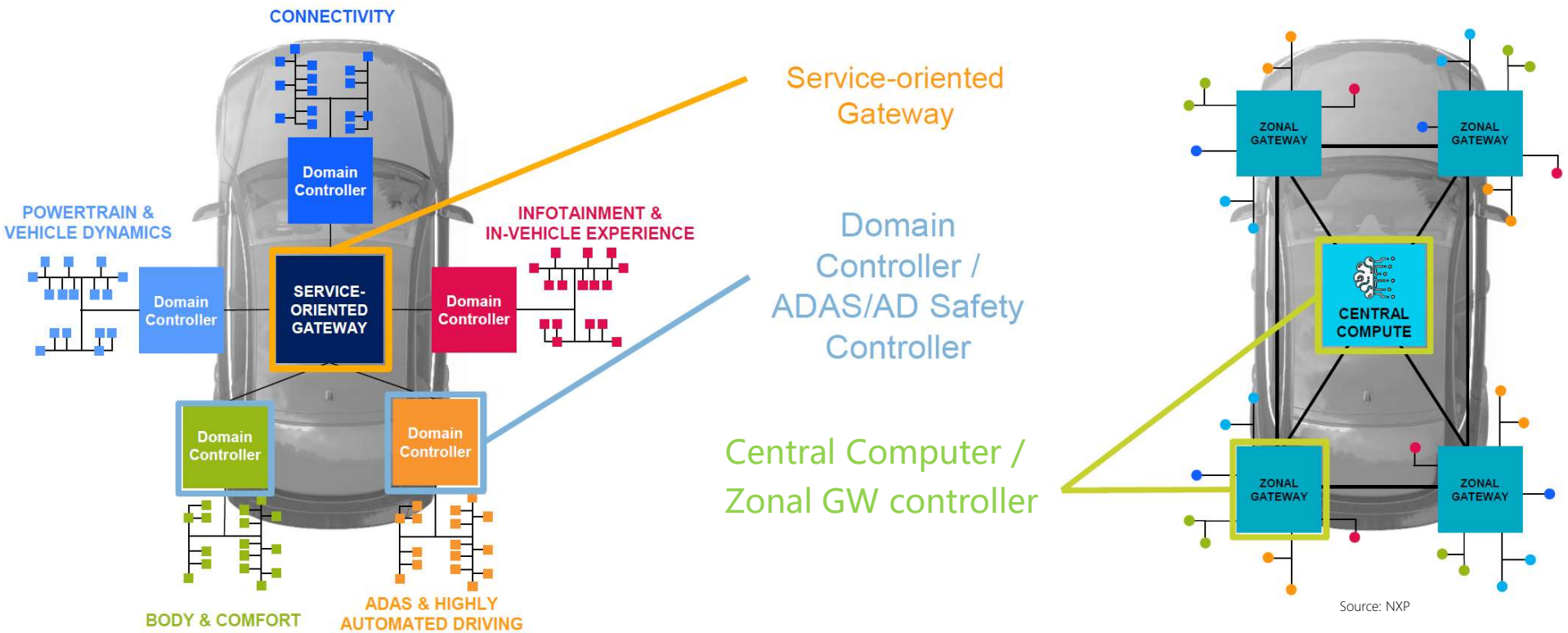
From **LOGICAL** distribution of functions (**Domain Controllers, ZGW & ECUs**)...

1.1 Automotive Electronics Industry

Nowadays, the **Automotive E/E Architecture** is being profoundly reshaped, in deep transformation -affecting especially the IVN infrastructure- in order to enable thus the **Autonomous-Connected-Electric-Shared (ACES) vehicle of tomorrow**

From **DOMAIN-BASED E/E IVN ARCHITECTURE...**

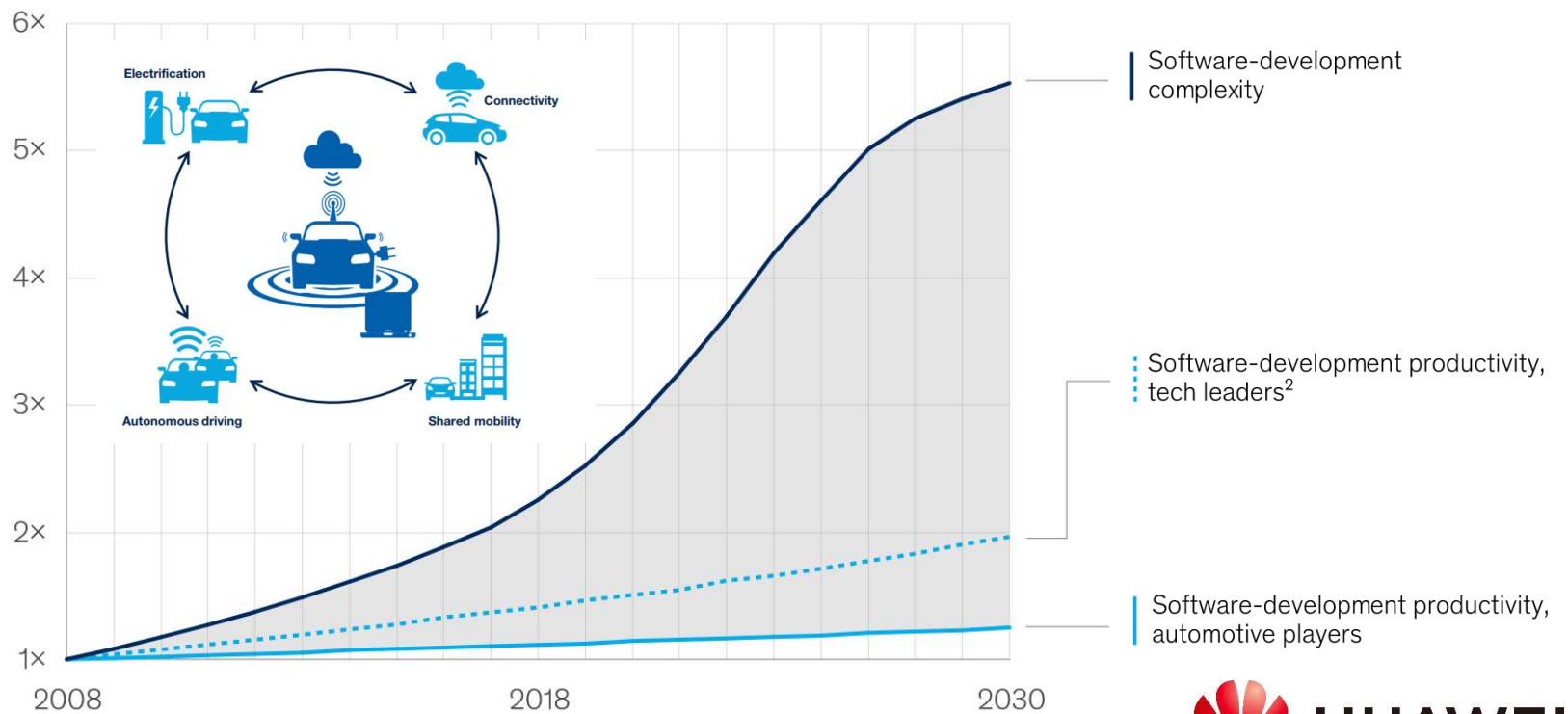
... to **ZONE-BASED E/E IVN ARCHITECTURE**



From LOGICAL distribution of functions (**Domain Controllers, ZGW & ECUs**)... ..to PHYSICAL (i.e. cross-domain) distribution (**HPCs and Zonal GW Controllers**)

1.1 Automotive Electronics Industry

MOTIVATION (WHY?): The automotive industry is confronting a **widening and unsustainable gap** between SW complexity and productivity levels. The complexity of automotive software to develop the **Software Defined Vehicle (SDV)** is escalating on both functional and architectural levels, but development productivity is not rising at the same pace



¹Analysis of >200 software-development projects from OEMs and from tier-1 and tier-2 suppliers.

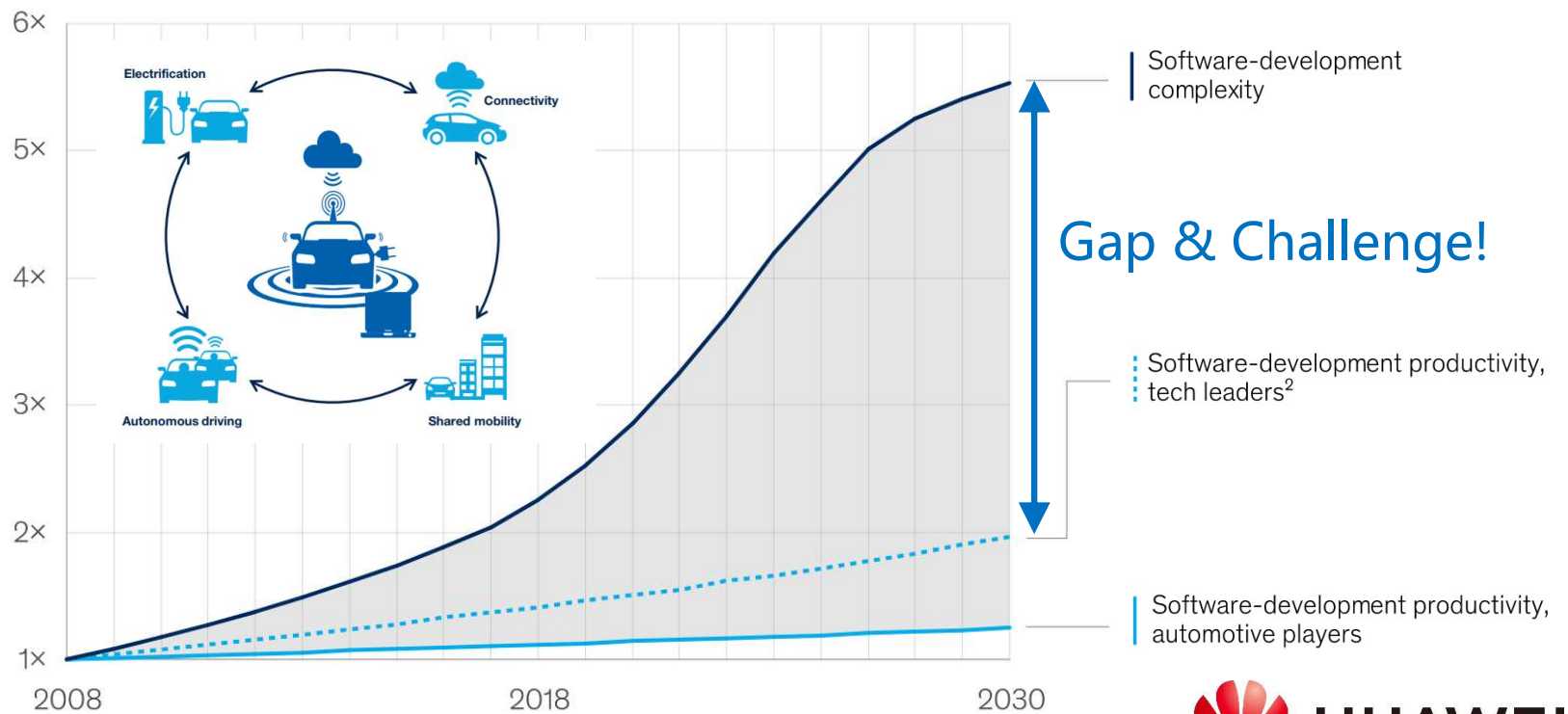
²Top-performing quartile of technology companies.

Source: McKinsey&Co.



1.1 Automotive Electronics Industry

MOTIVATION (WHY?): The automotive industry is confronting a **widening and unsustainable gap** between SW complexity and productivity levels. The complexity of automotive software to develop the **Software Defined Vehicle (SDV)** is escalating on both functional and architectural levels, but development productivity is not rising at the same pace



¹Analysis of >200 software-development projects from OEMs and from tier-1 and tier-2 suppliers.

²Top-performing quartile of technology companies.

1.1 Automotive Electronics Industry

MOTIVATION (WHY?): The automotive industry is facing a **widening and unsustainable gap** between SW complexity and productivity. To develop the **Software Defined Vehicle (SDV)** is escalating on both sides, but productivity is not rising at the same pace

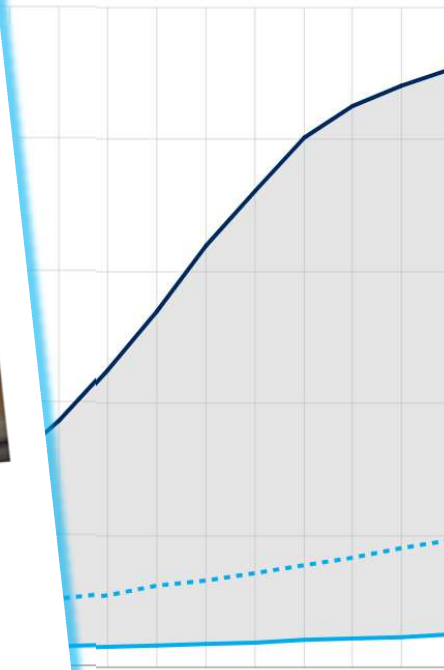
Elon Musk says the difference between Tesla being 'worth a lot of money or worth basically zero' all comes down to solving self-driving technology

Matthew Loh Jun 16, 2022, 6:24 AM



Elon Musk's vision for fully autonomous cars involves them being able to find their drivers and pick them up. Maja Hitaj/Getty Images

- Elon Musk says the true value of Tesla comes down to solving the question of self-driving.
- Without the revolutionary tech, the automaker would be "worth basically zero," Musk said.
- He's promised self-driving cars since 2015, but Tesla has since faced setbacks with its software.



Software-development complexity

Gap & Challenge!

Software-development productivity, tech leaders²

Software-development productivity, automotive players

2030
Source: McKinsey&Co.



²Top-performing quartile of technology companies.

1.1 Automotive Electronics Industry

MOTIVATION (WHY?): The automotive industry is facing a sustainable gap between SW complexity and

Elon Musk says the difference between Tesla being 'worth a lot of money or worth basically zero' all comes down to solving self-driving technology

Matthew Loh Jun 16, 2022, 6:24 AM



Elon Musk's vision for fully autonomous cars involves them being able to find their drivers and pick them up. Maja Hitaj/Getty Images

- Elon Musk says the true value of Tesla comes down to solving the question of self-driving.
- Without the revolutionary tech, the automaker would be "worth basically zero," Musk said.
- He's promised self-driving cars since 2015, but Tesla has since faced setbacks with its software.

Top-performing quartile of technology companies.

VW CEO lost his job over buggy software that delayed new models

Struggles at Cariad unit have pushed back projects like the electric Porsche Macan

BLOOMBERG Jul 25th 2022 at 2:28PM



It says a lot about the state of the auto industry and where it's going that software problems have cost the CEO of a carmaker his job. Volkswagen ousted Herbert Diess as chief executive officer after severe software-development delays set back the scheduled launch of new Porsches, Audis and Bentleys. This was untenable considering buggy software postponed the debut of VW's initial rollout of ID models, and customers are still having to drop off their cars at the dealer for updates the company has struggled to make over the air.

1.1 Automotive Electronics Industry

MOTIVATION (WHY?): The automotive industry is facing a sustainable gap between SW complexity and

Elon Musk says the difference between Tesla being 'worth a lot of money or worth basically zero' all comes down to solving self-driving technology

Matthew Loh Jun 16, 2022, 6:24 AM



Elon Musk's vision for fully autonomous cars involves them being able to find their drivers and pick them up. Maja Hitaj/Getty Images

- Elon Musk says the true value of Tesla comes down to solving the question of self-driving.
- Without the revolutionary tech, the automaker would be "worth basically zero," Musk said.
- He's promised self-driving cars since 2015, but Tesla has since faced setbacks with its software.

"Crisis" in Chinese (2 characters)

危機

Danger Opportunity

VW CEO lost his job over buggy software that delayed new models

Struggles at Cariad unit have pushed back projects like the electric Porsche Macan



It says a lot about the state of the auto industry and where it's going that software problems have cost the CEO of a carmaker his job. Volkswagen ousted Herbert Diess as chief executive officer after severe software-development delays set back the scheduled launch of new Porsches, Audis and Bentleys. This was untenable considering buggy software postponed the debut of VW's initial rollout of ID models, and customers are still having to drop off their cars at the dealer for updates the company has struggled to make over the air.

Top-performing quartile of technology companies.

1.1 Automotive Electronics Industry

MOTIVATION (WHY?): The automotive industry is facing a sustainable gap between SW complexity and

Elon Musk says the difference between Tesla being 'worth a lot of money or worth basically zero' all comes down to solving self-driving technology

Matthew Loh Jun 16, 2022, 6:24 AM



Elon Musk's vision for fully autonomous cars involves them being able to find their drivers and pick them up. Maja Hitz/Getty Images

- Elon Musk says the true value of Tesla comes down to solving the question of self-driving.
- Without the revolutionary tech, the automaker would be "worth basically zero," Musk said.
- He's promised self-driving cars since 2015, but Tesla has since faced setbacks with its software.

"Crisis" in Chinese (2 characters)

危機

Danger Opportunity

VW CEO lost his job over buggy software that delayed new models

Struggles at Cariad unit have pushed back projects like the electric Porsche Macan



It says a lot about the state of the auto industry and where it's going that software problems have cost the CEO of a carmaker his job. Volkswagen ousted Herbert Diess as chief executive officer after severe software-development delays set back the scheduled launch of new Porsches, Audis and Bentleys. This was untenable considering buggy software postponed the debut of VW's initial rollout of ID models, and customers are still having to drop off their cars at the dealer for updates the company has struggled to make over the air.

Top-performing quartile of technology companies.

Contents

1. [WHY] Automotive Electronic Industry

- The Innovation Race
- Pain Points & Challenges

2. [HOW] IVN, Zonal GW Controller & Networking Chipset

- Huawei Automotive IVN Research Group
- Zonal Gateway Controller
- Network SoC Prototyping

3. [WHAT] Elastic Gateway SoC (eGW) Concept

- Related Art on HW/SW Codesign in Automotive
- The 12 HW Innovations on NW & APP Processing of eGW SoC

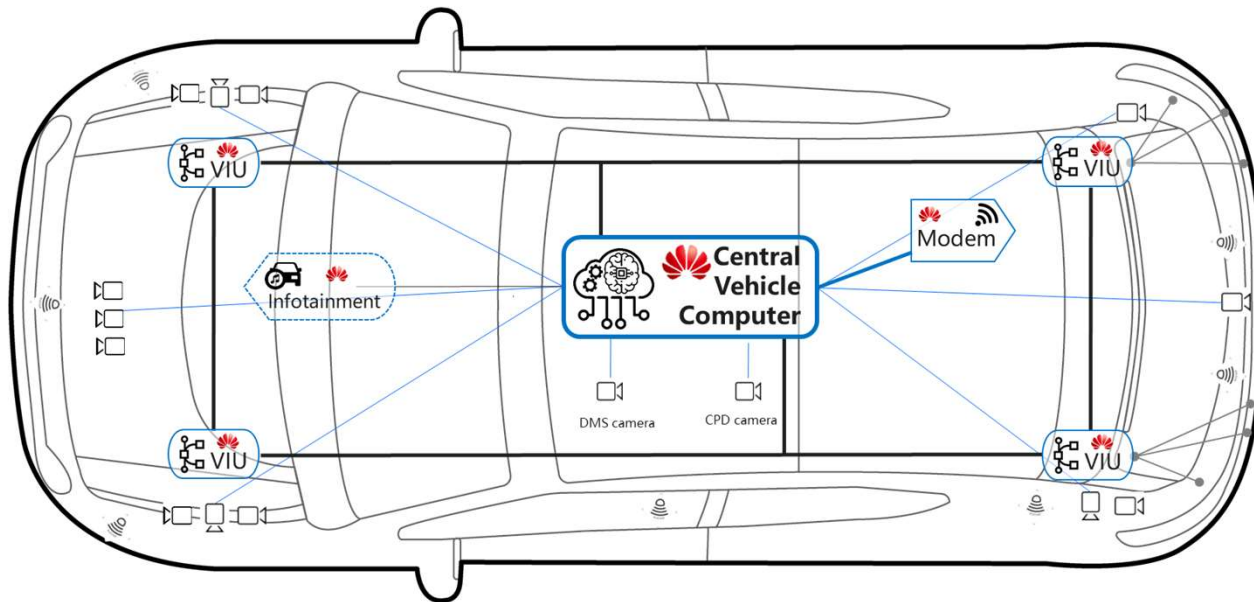


2.1 Huawei Automotive In-Vehicle Network Research Group – Goals & Scope

Our goal is to perform research, generate new knowledge and develop innovative technological solutions in the areas of:

- **Automotive E/E IVN Architectures (System/Vehicle Level: NW)**
- **Zonal GW Controller Development (Component Level: HW/SW ECU)**
- **Network SoC and Hardware Accelerators Design (Chipset Level: SoC/HWA)**

while validating them through hands-on deployment of **Proof-of-Concepts (PoC)** on rapid prototyping platforms (SoC/MCU/FPGA)

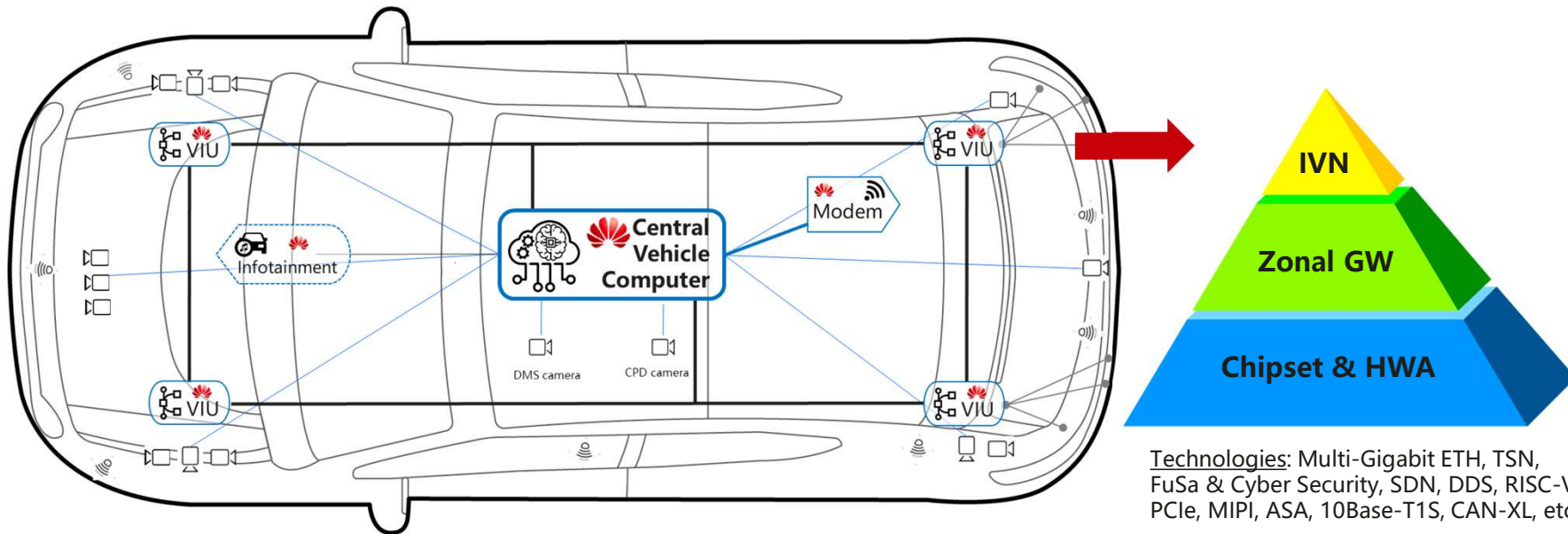


2.1 Huawei Automotive In-Vehicle Network Research Group – Goals & Scope

Our goal is to perform research, generate new knowledge and develop innovative technological solutions in the areas of:

- **Automotive E/E IVN Architectures (System/Vehicle Level: NW)**
- **Zonal GW Controller Development (Component Level: HW/SW ECU)**
- **Network SoC and Hardware Accelerators Design (Chipset Level: SoC/HWA)**

while validating them through hands-on deployment of **Proof-of-Concepts (PoC)** on rapid prototyping platforms (SoC/MCU/FPGA)

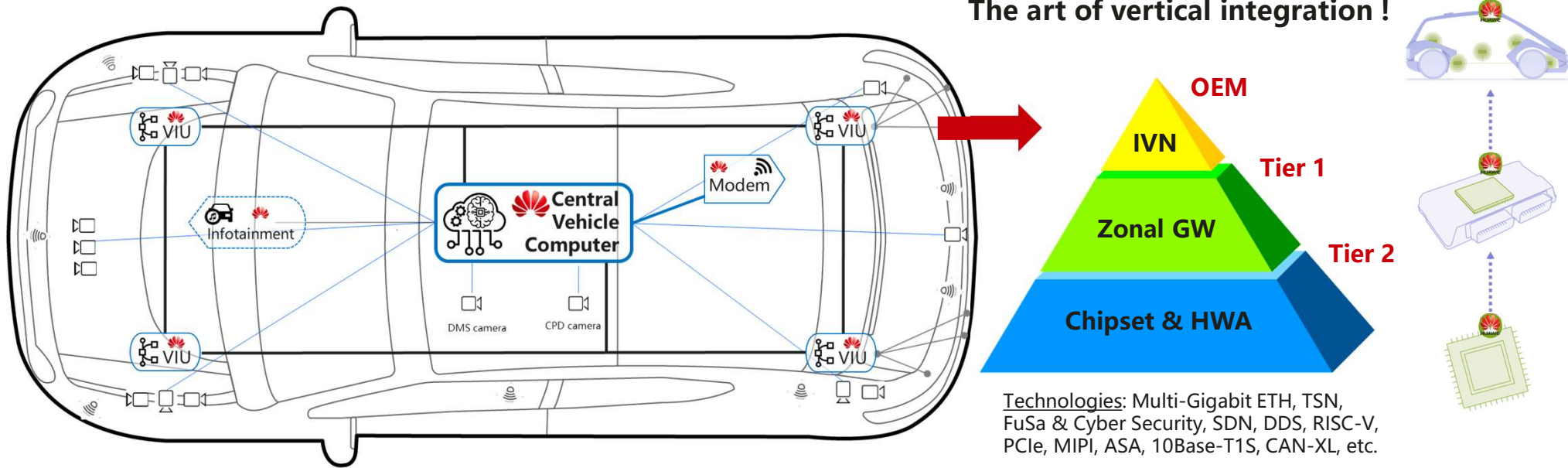


2.1 Huawei Automotive In-Vehicle Network Research Group – Goals & Scope

Our goal is to perform research, generate new knowledge and develop innovative technological solutions in the areas of:

- **Automotive E/E IVN Architectures (System/Vehicle Level: NW)**
- **Zonal GW Controller Development (Component Level: HW/SW ECU)**
- **Network SoC and Hardware Accelerators Design (Chipset Level: SoC/HWA)**

while validating them through hands-on deployment of **Proof-of-Concepts (PoC)** on rapid prototyping platforms (SoC/MCU/FPGA)

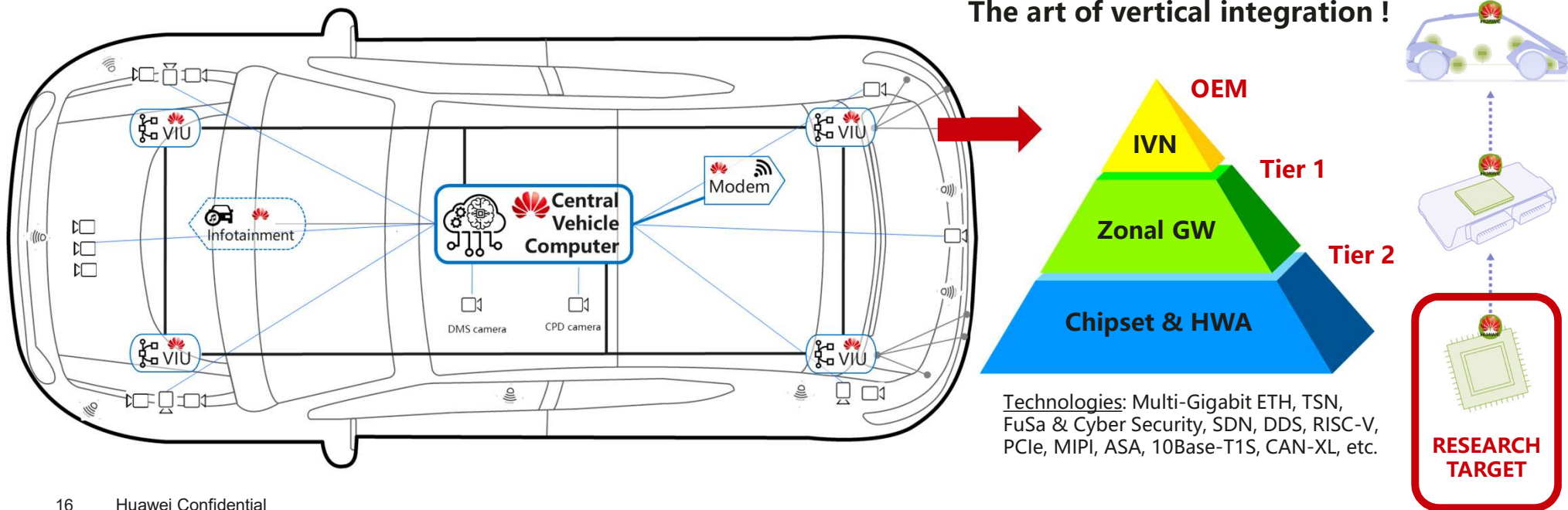


2.1 Huawei Automotive In-Vehicle Network Research Group – Goals & Scope

Our goal is to perform research, generate new knowledge and develop innovative technological solutions in the areas of:

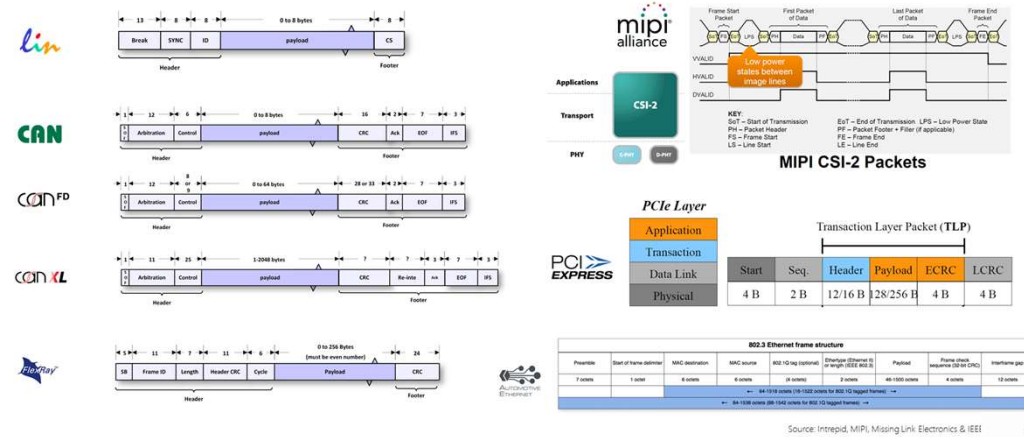
- **Automotive E/E IVN Architectures (System/Vehicle Level: NW)**
- **Zonal GW Controller Development (Component Level: HW/SW ECU)**
- **Network SoC and Hardware Accelerators Design (Chipset Level: SoC/HWA)**

while validating them through hands-on deployment of **Proof-of-Concepts (PoC)** on rapid prototyping platforms (SoC/MCU/FPGA)

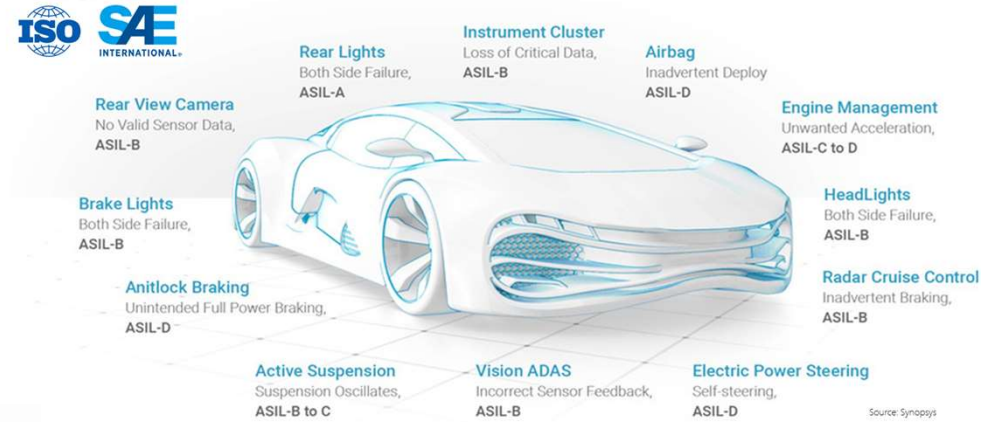


2.2 Huawei Automotive In-Vehicle Network Research Group – Research Technologies

T1. High performance in **Gatewaying/Tunneling/Routering/Forwarding** processing is crucial (E2E latency, jitter, bandwidth, etc.)

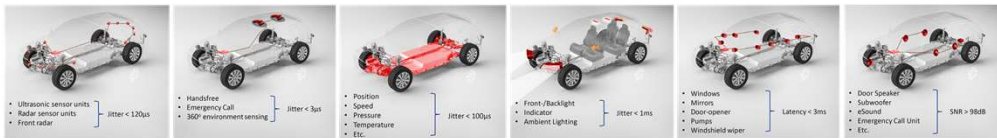


T2. **Functional Safety** is getting more and more attention in the ACES vehicle of tomorrow

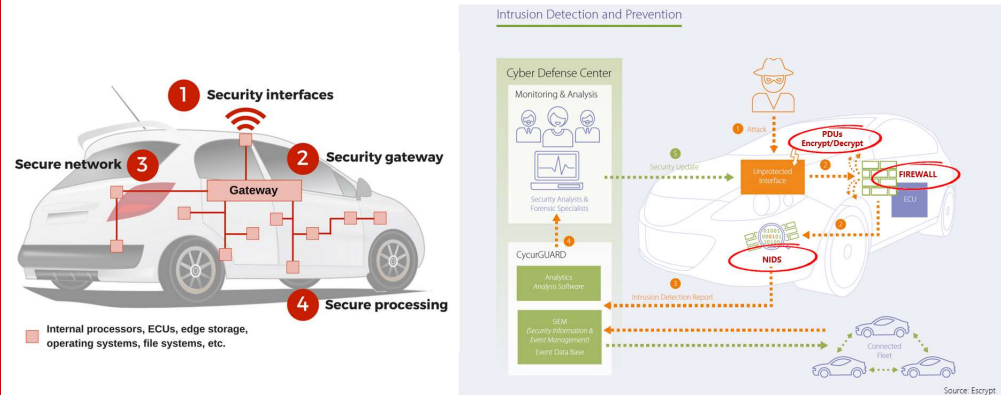


T4. The deployment of **Application Layer functionality** in zonal GW controllers, **on top of the IVN infrastructure**, needs to guarantee right level of reliability, performance and cost-effectiveness

Applications	Powertrain/Chassis	Body/Comfort	Cockpit/Infotainment	ADAS/AD	Connectivity
Bandwidth	Low	Low/Mid	High	High	High
E2E Latency	< 1ms	<100ms	~10ms	< 1ms	100ms
Cost	Low	Low	High	High	Medium
Network Technology	CAN/LIN FlexRay/10Base-T1S	CAN/LIN/10Base-T1S 100Base-T1	1000Base-T1	1000Base-T1	100Base-T1 1000Base-T1

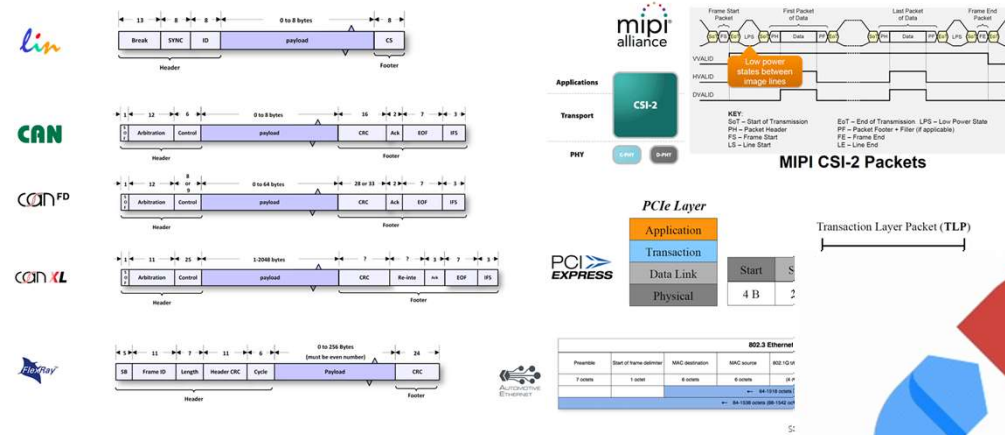


T3. **Cyber Security**, together with FuSa, deployed in the **Zonal Gateway Controller** is key in order to achieve a holistic security solution of the vehicle infrastructure

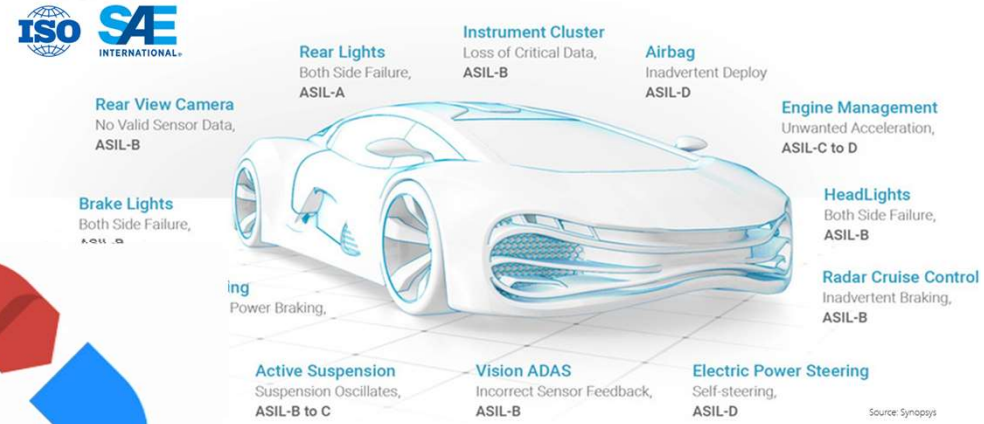


2.2 Huawei Automotive In-Vehicle Network Research Group – Research Technologies

T1. High performance in **Gatewaying/Tunneling/Routering/Forwarding** processing is crucial (E2E latency, jitter, bandwidth, etc.)

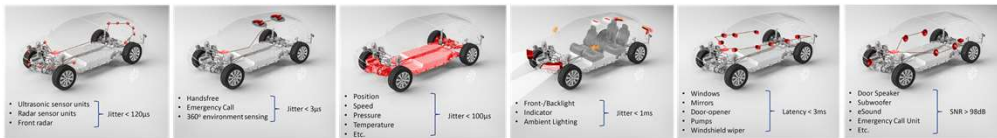


T2. **Functional Safety** is getting more and more attention in the ACES vehicle of tomorrow

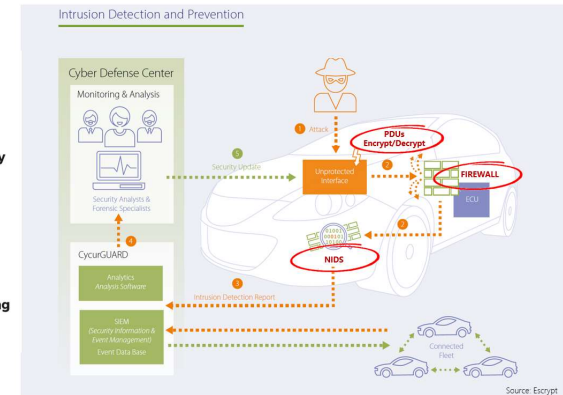
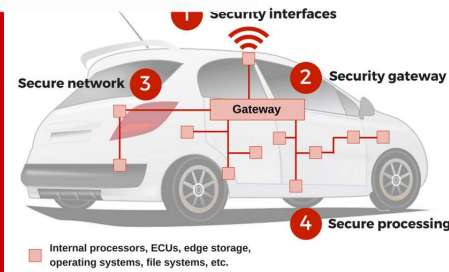


T4. The deployment of **Application Layer functionality** in zonal GW controllers, on top of the IVN infrastructure, right level of reliability, performance and cost-effectiveness

Applications	Powertrain/Chassis	Body/Comfort	Cockpit/Infotainment	ADAS/AD
Bandwidth	Low	Low/Mid	High	High
E2E Latency	< 1ms	<100ms	~10ms	< 1ms
Cost	Low	Low	High	High
Network Technology	CAN/LIN FlexRay/10Base-T1S	CAN/LIN/10Base-T1S 100Base-T1	1000Base-T1	1000Base-T1



ther with FuSa, deployed in the **Zonal Gateway Controller** is key in order to achieve a holistic security infrastructure



2.3 Zonal Gateway Controller – Conceptualization & Requeriments

All in all, the **Zonal Gateway Controller** is probably one of the most complex networking devices ever, at least in terms of mixing heterogeneous application functionality and networking technologies



T1. High Performance Networking



T2. Functional Safety



T4. Zone Applications



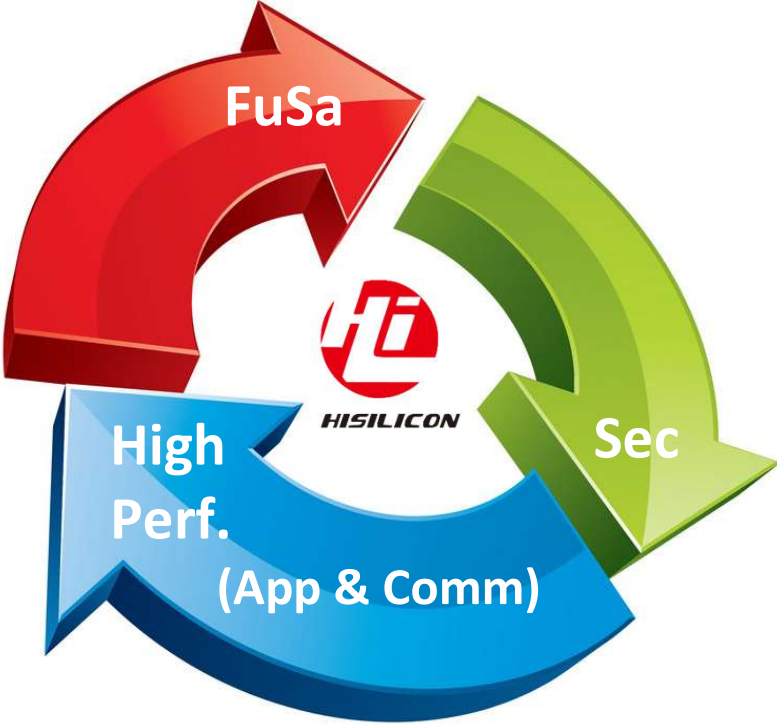
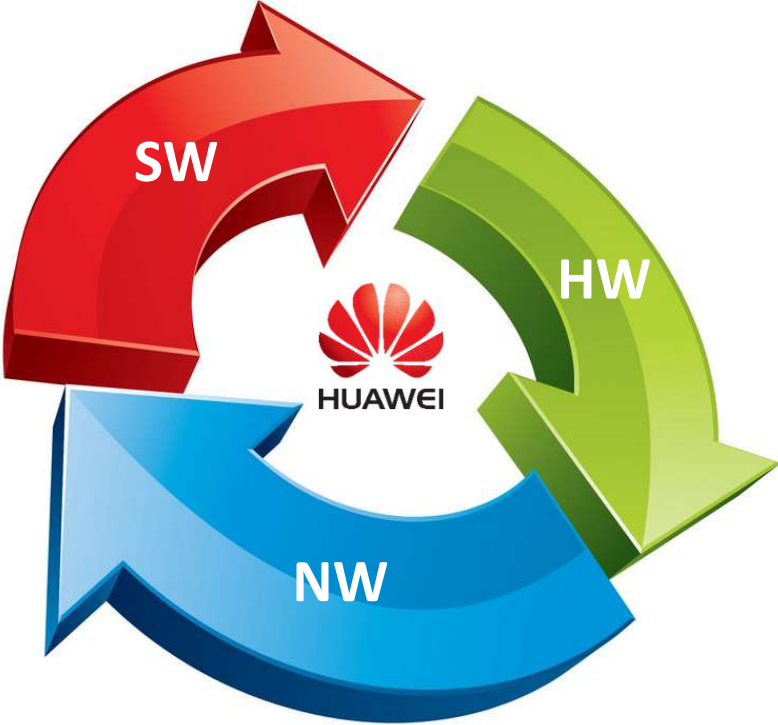
T3. Cyber Security



Zonal GW Controller = Application (Cross-Domain) Functions + Networking & Power Distribution + Safety & Security

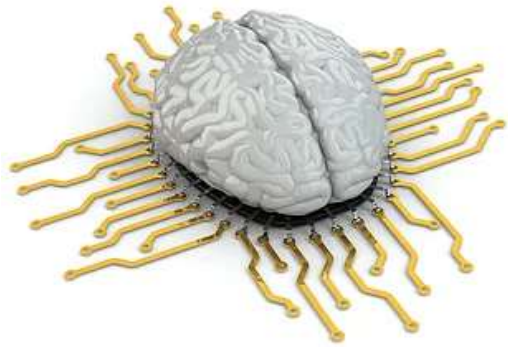
2.4 Our HW-centric Solution: Elastic Gateway SoC (eGW)

CHALLENGE: In search of the right **HW/SW/NW** codesign in order to deliver a well balanced **high performing, safe and secure** **Zonal Gateway Controller/SoC** product architecture able to solve the current engineering pain points and meet all the demanded requirements

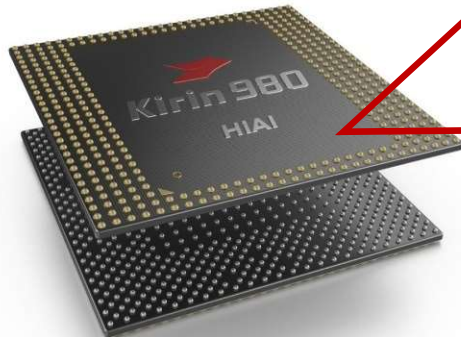


2.4 Our HW-centric Solution: Elastic Gateway SoC (eGW)

HOW? Our target is to design and develop a new **SoC architecture** provided with innovative **hardware accelerators** (i.e. **standardizable peripherals or coprocessors**) to accomplish the **whole set of cyber physical system** (i.e. **SDV**) requirements, while **minimizing complexity and maximizing performance at affordable cost**



HUAWEI



RESEARCH GOALS

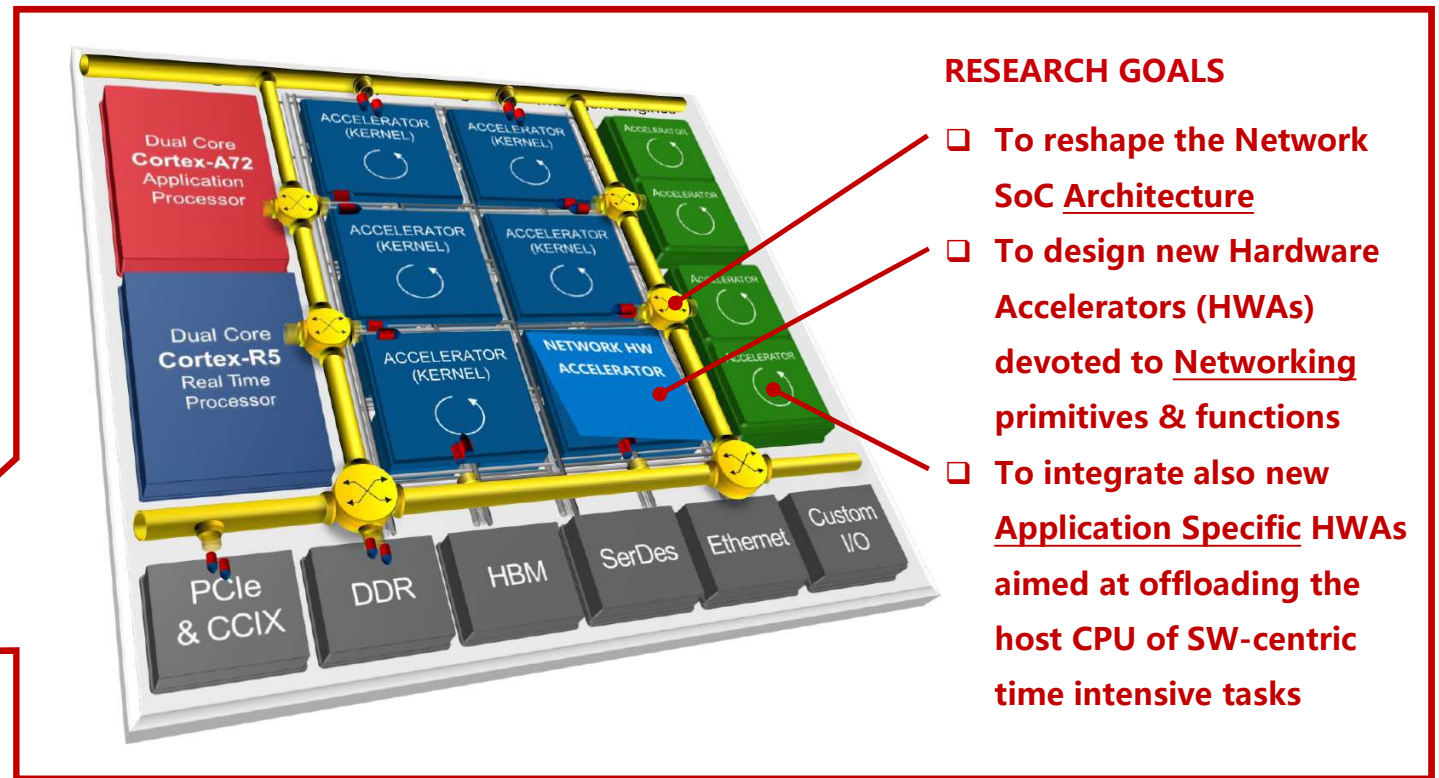
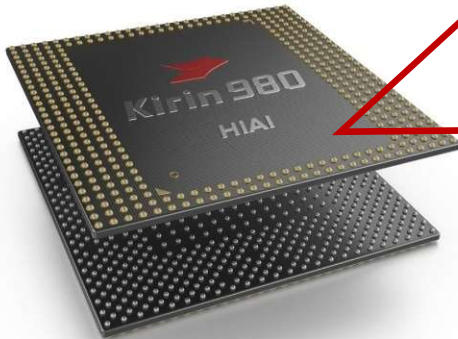
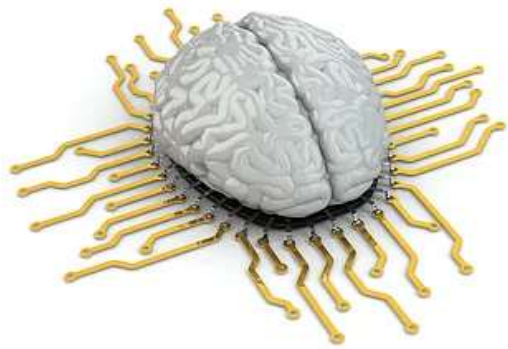
- ❑ To reshape the Network SoC Architecture
- ❑ To design new Hardware Accelerators (HWAs) devoted to Networking primitives & functions
- ❑ To integrate also new Application Specific HWAs aimed at offloading the host CPU of SW-centric time intensive tasks



HUAWEI

2.4 Our HW-centric Solution: Elastic Gateway SoC (eGW)

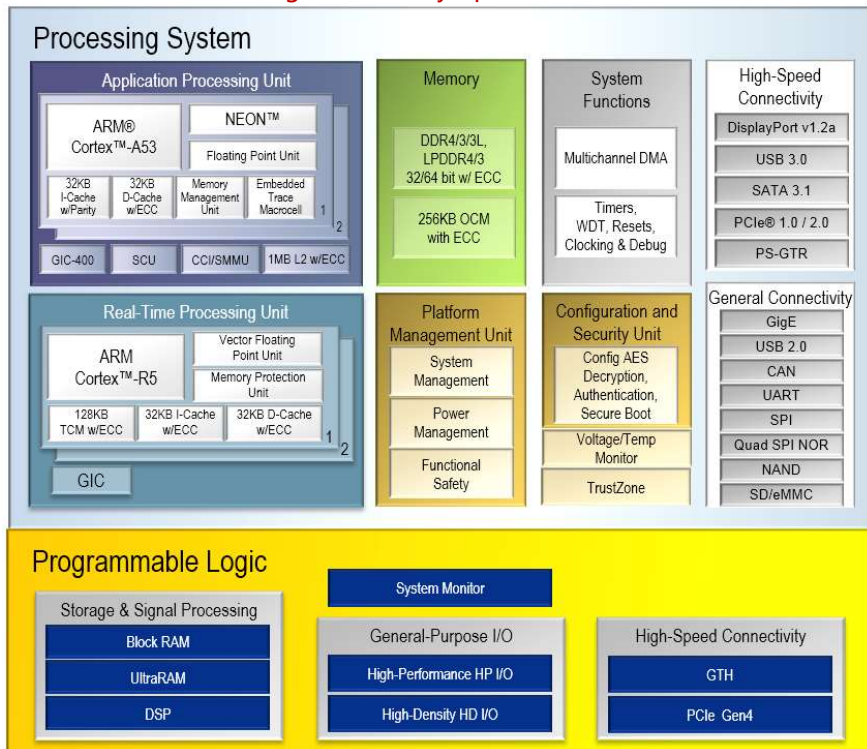
HOW? Our target is to design and develop a new **SoC architecture** provided with innovative **hardware accelerators** (i.e. **standardizable peripherals or coprocessors**) to accomplish the **whole set of cyber physical system** (i.e. **SDV**) requirements, while **minimizing complexity and maximizing performance at affordable cost**



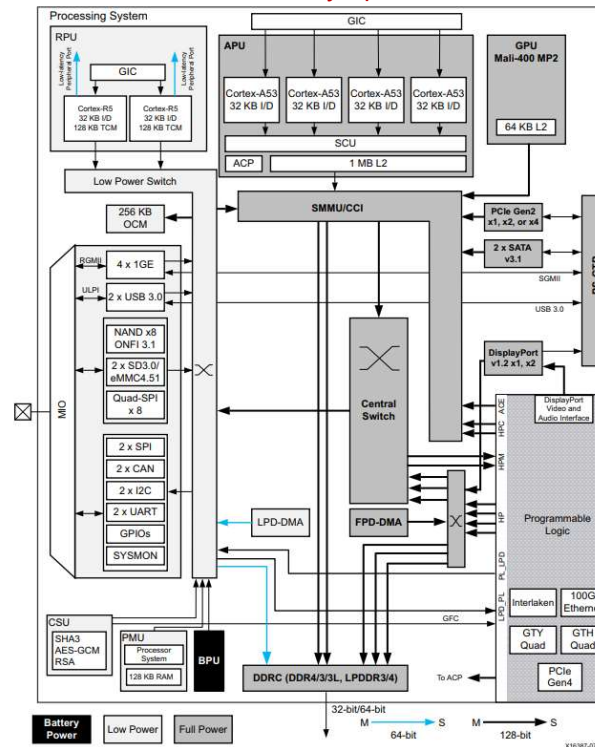
2.5 eGW SoC Prototyping

HOW? Xilinx Zynq UltraScale+ MPSoC is the prototyping platform chosen to deploy our HW-centric eGW SoC Concept

Block Diagram Xilinx Zynq UltraScale+ MPSoC



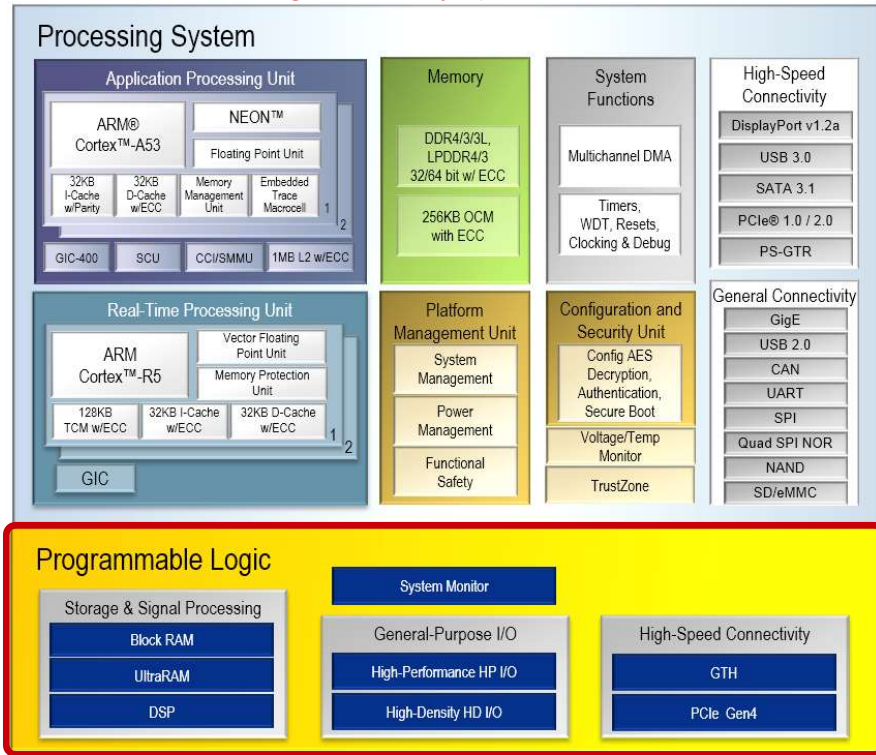
Interconnections Xilinx Zynq UltraScale+ MPSoC



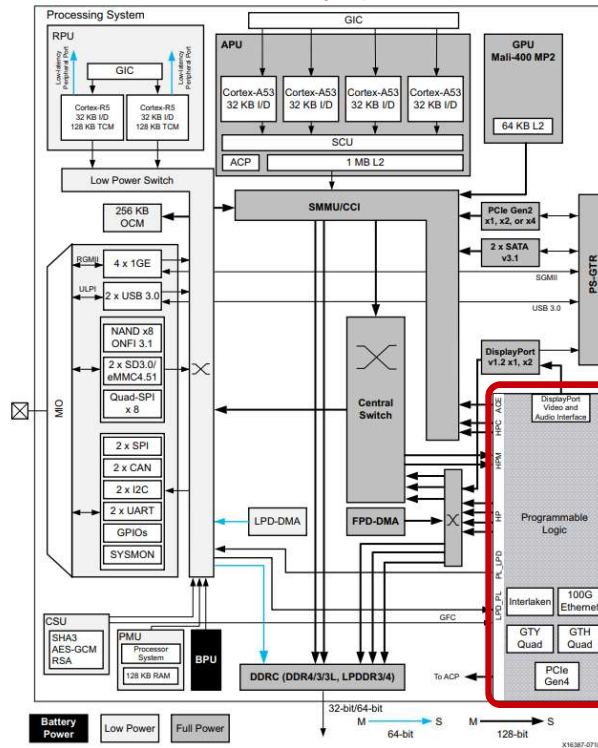
2.5 eGW SoC Prototyping

HOW? Xilinx Zynq UltraScale+ MPSoC is the prototyping platform chosen to deploy our HW-centric eGW SoC Concept

Block Diagram Xilinx Zynq UltraScale+ MPSoC



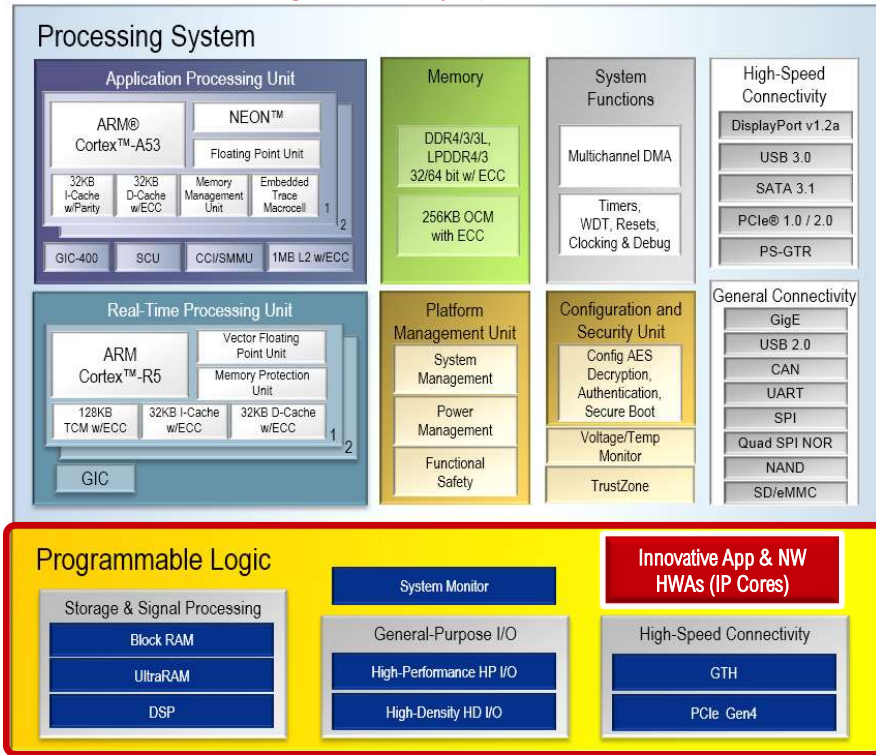
Interconnections Xilinx Zynq UltraScale+ MPSoC



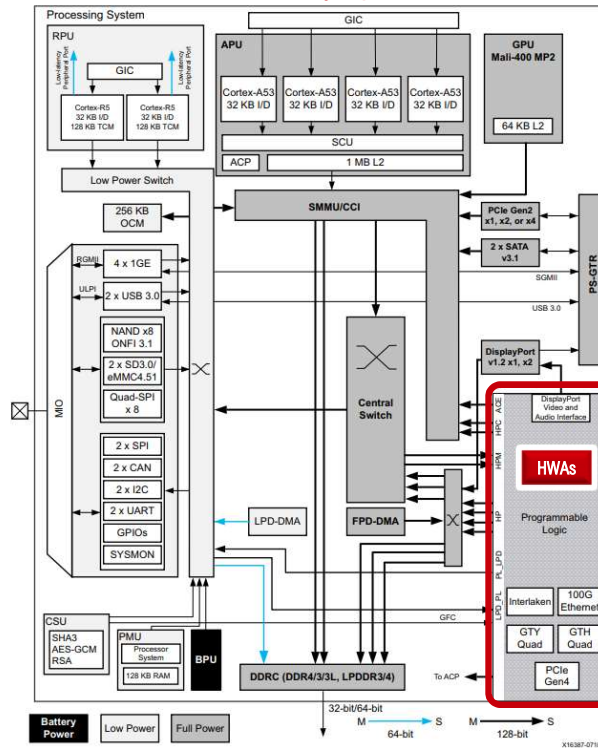
2.5 eGW SoC Prototyping

HOW? Xilinx Zynq UltraScale+ MPSoC is the prototyping platform chosen to deploy our HW-centric eGW SoC Concept

Block Diagram Xilinx Zynq UltraScale+ MPSoC



Interconnections Xilinx Zynq UltraScale+ MPSoC



Contents

1. [WHY] Automotive Electronic Industry

- The Innovation Race
- Pain Points & Challenges

2. [HOW] IVN, Zonal GW Controller & Networking Chipset

- Huawei Automotive IVN Research Group
- Zonal Gateway Controller
- Network SoC Prototyping

3. [WHAT] Elastic Gateway SoC (eGW) Concept

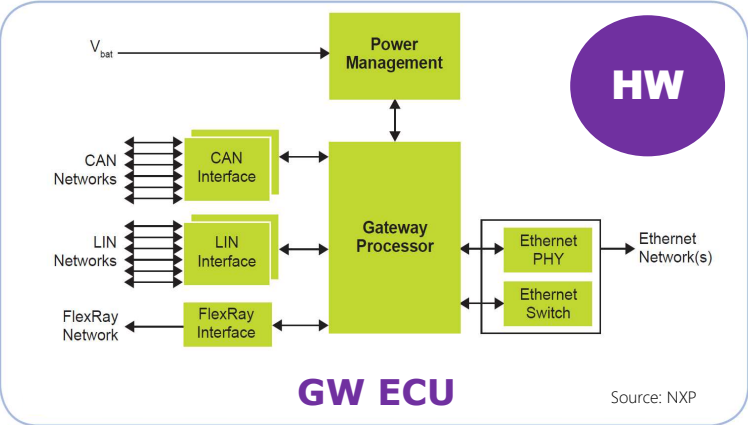
- Related Art on HW/SW Codesign in Automotive
- The 12 HW Innovations on NW & APP Processing of eGW SoC

3.1 State of the Art of Gateway Controllers

Nowadays, the solution for **Automotive GW Controllers** is a SW-centric implementation on **MCU/SoC** devices

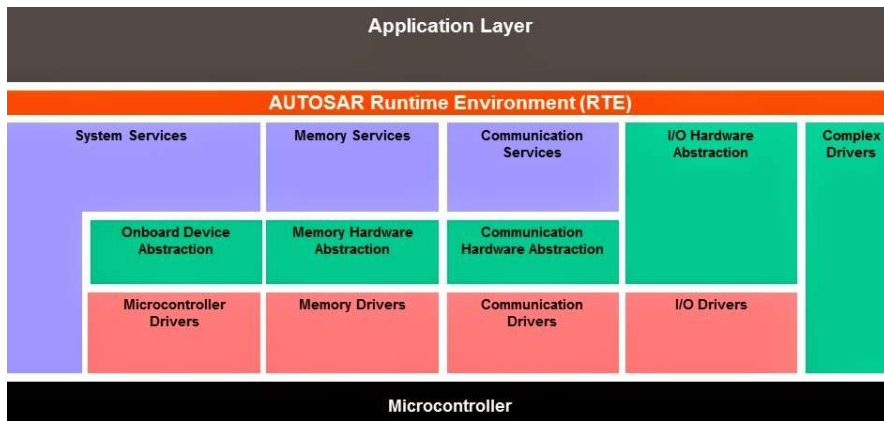


BMW Central GW

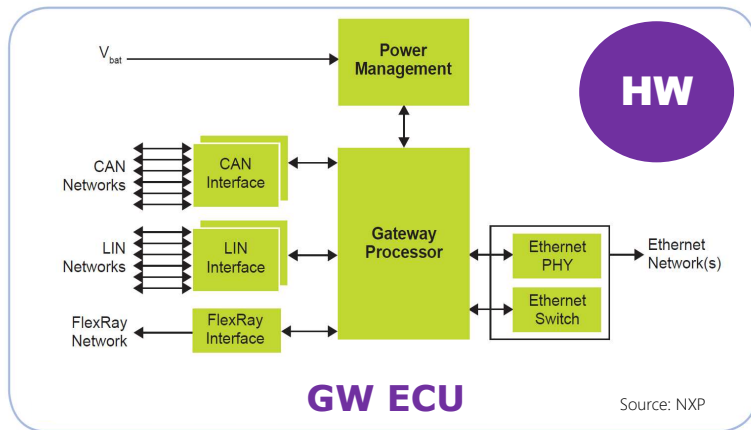


3.1 State of the Art of Gateway Controllers

Nowadays, the solution for **Automotive GW Controllers** is a SW-centric implementation on **MCU/SoC** devices



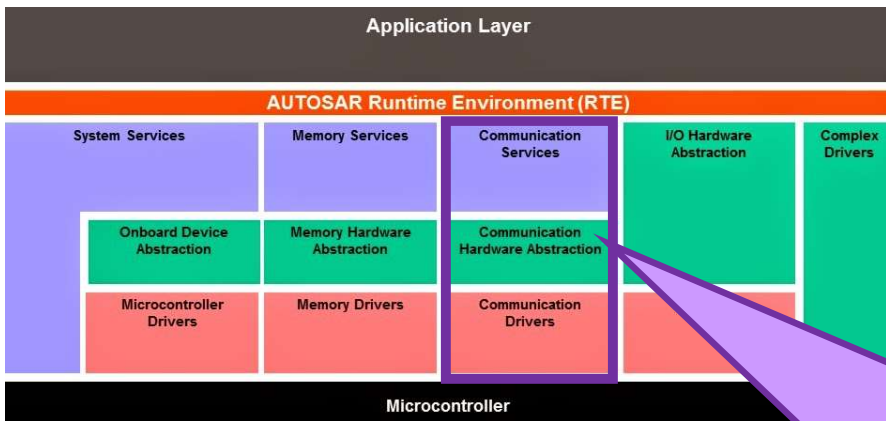
Source: AUTOSAR



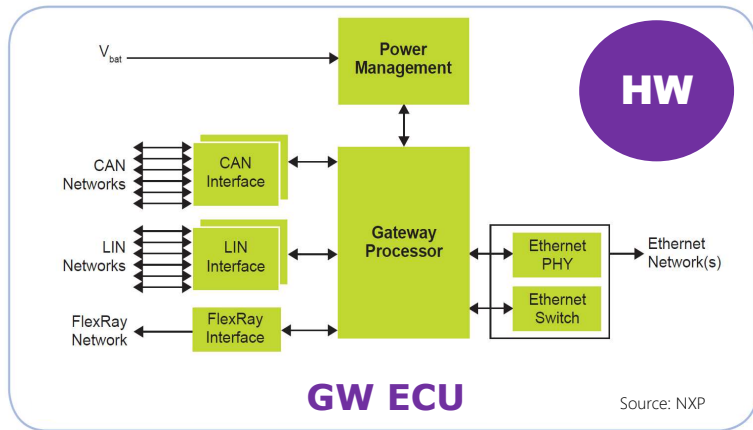
Source: NXP

3.1 State of the Art of Gateway Controllers

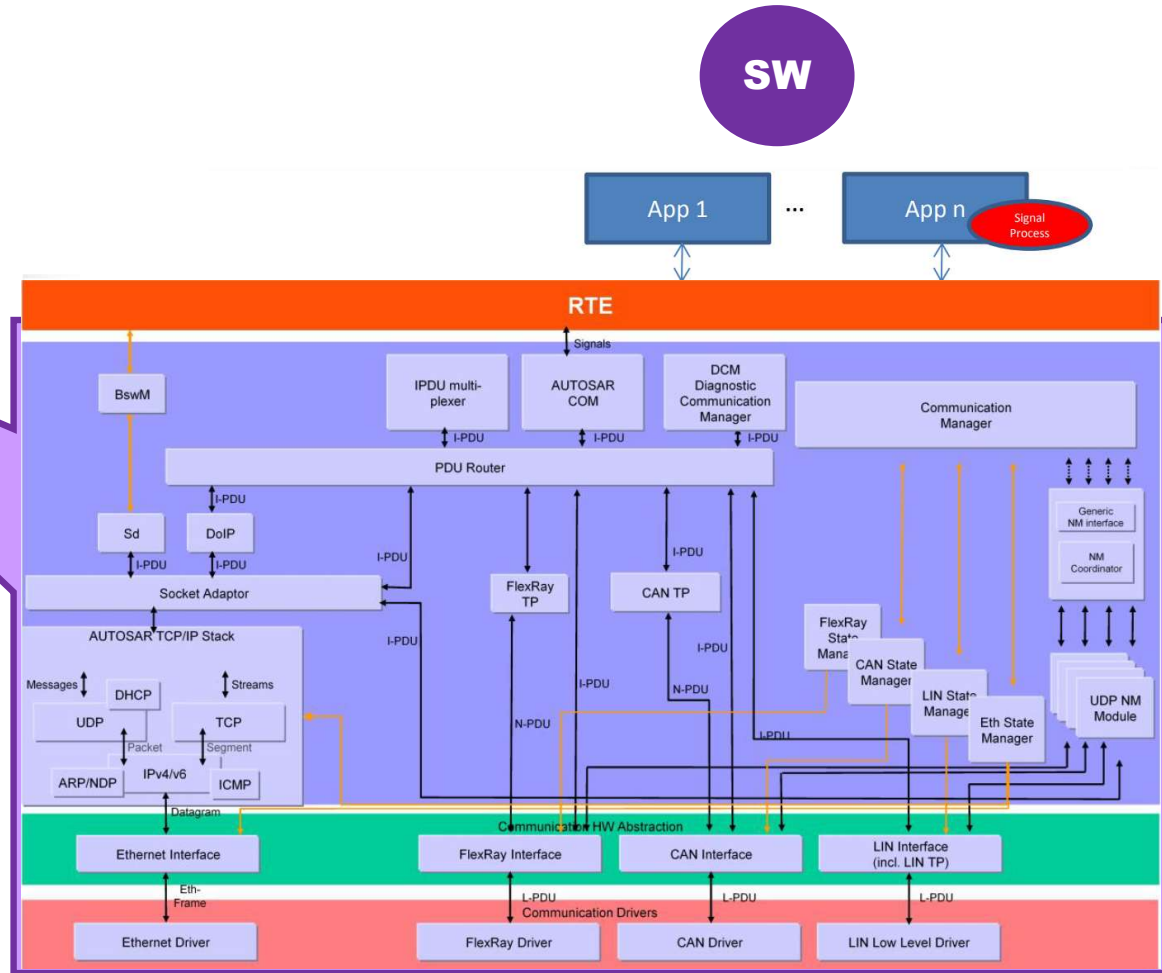
Nowadays, the solution for **Automotive GW Controllers** is a SW-centric implementation on **MCU/SoC** devices



Source: [unintelligible]

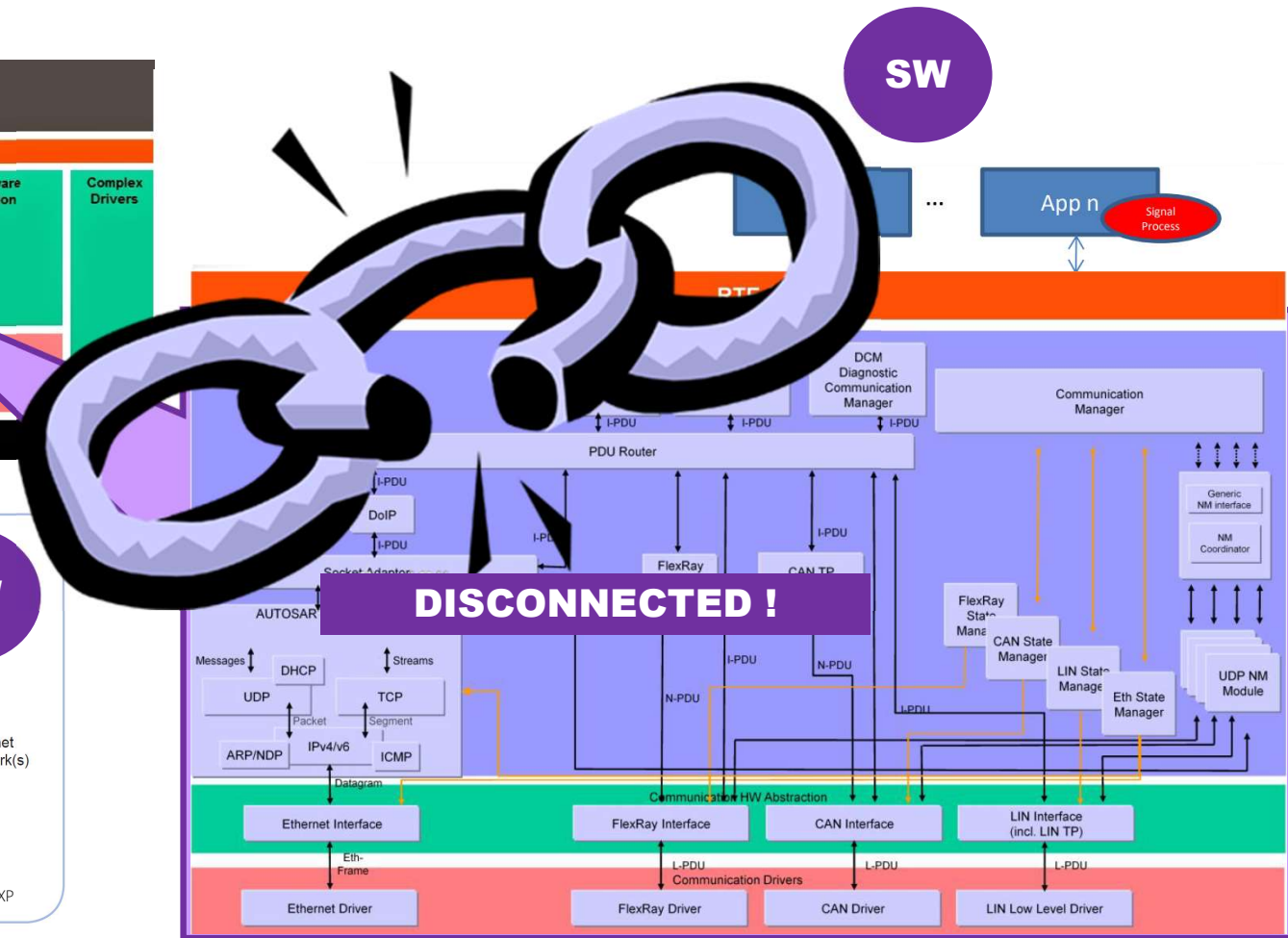
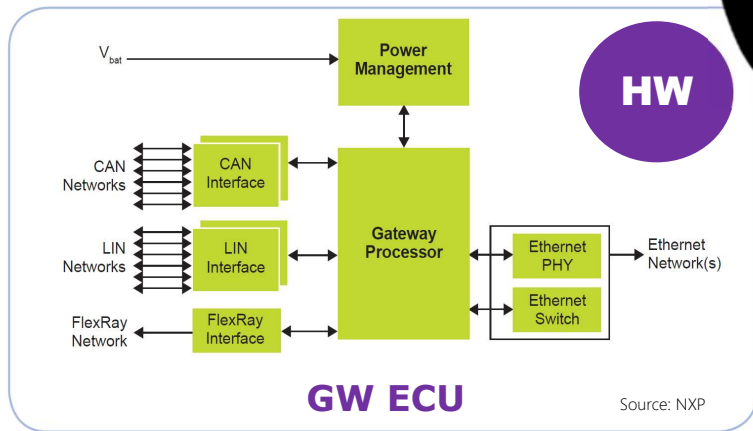
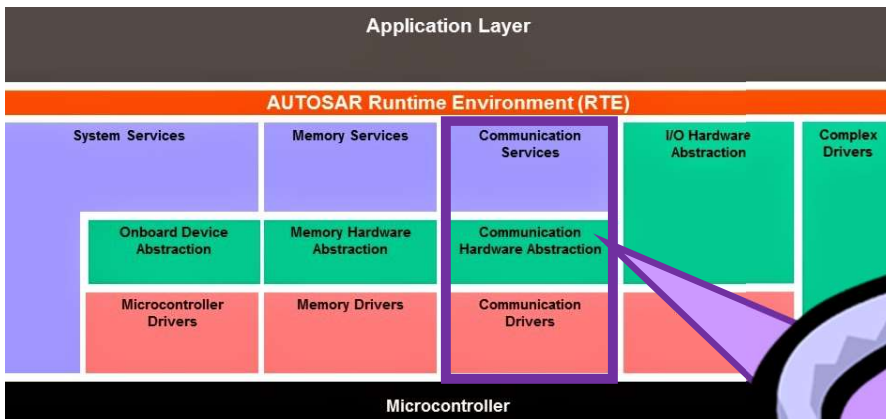


Source: NXP



3.1 State of the Art of Gateway Controllers

Nowadays, the solution for **Automotive GW Controllers** is a SW-centric implementation on **MCU/SoC** devices



3.1 State of the Art of Gateway Controllers

Nowadays, the solution for Automotive GW Control is based on

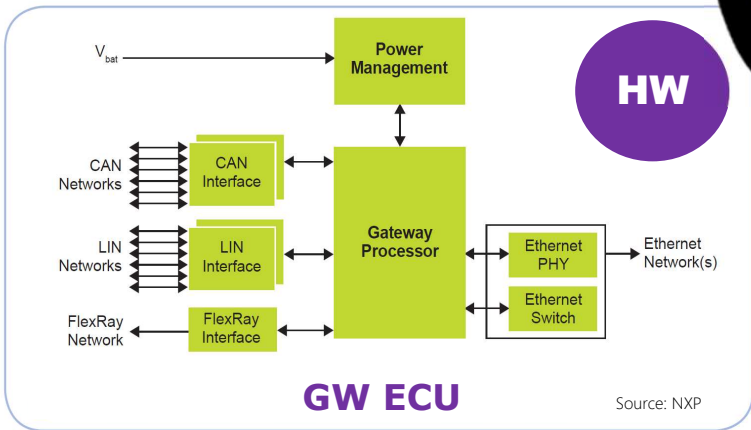
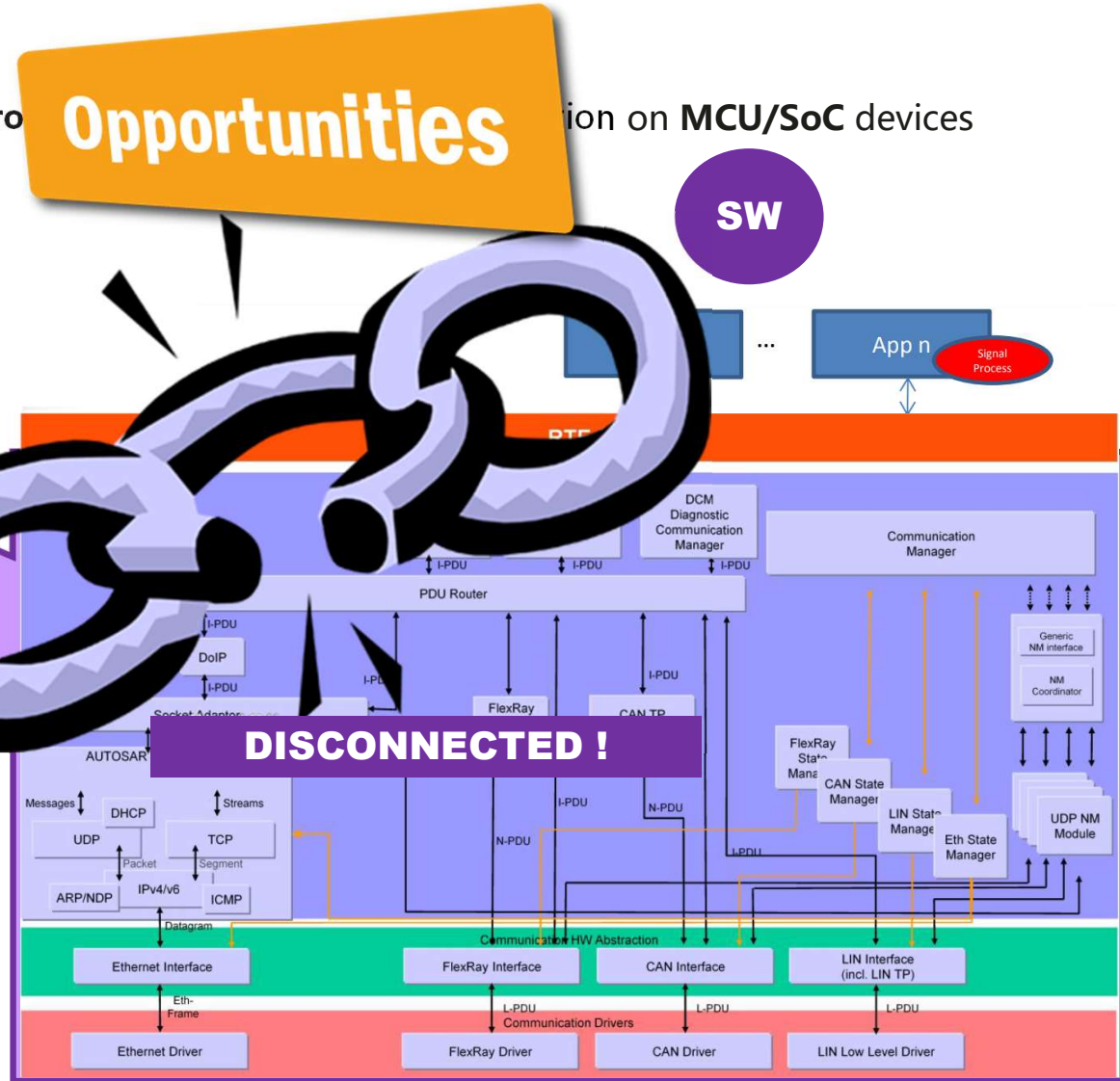


Opportunities

... on MCU/SoC devices

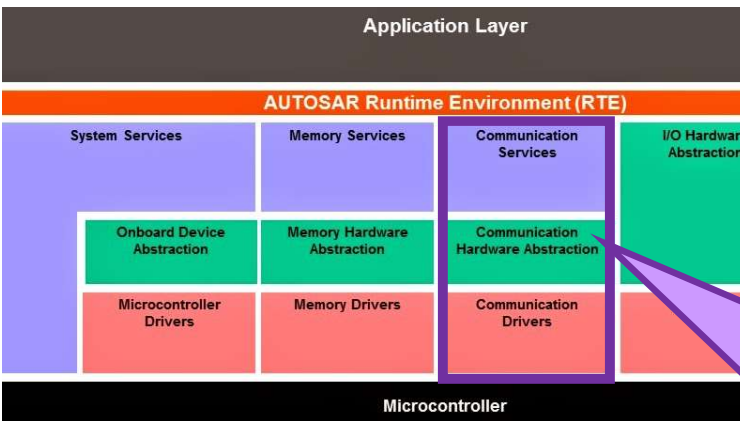
SW

HW

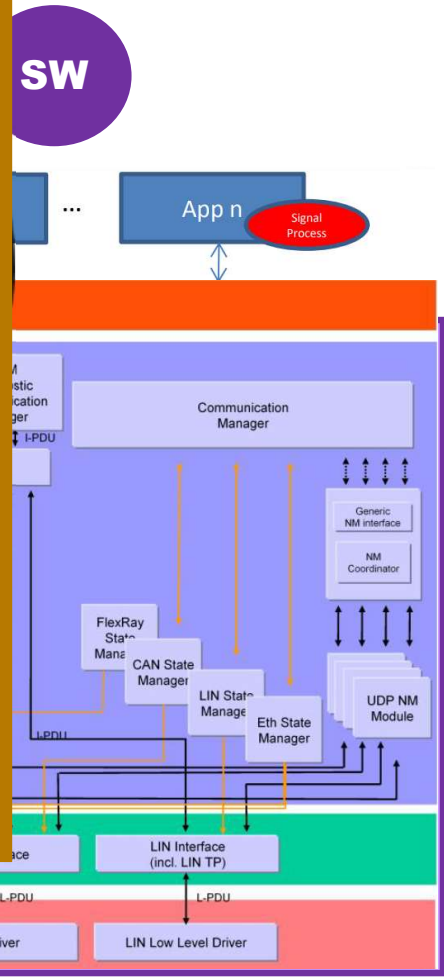
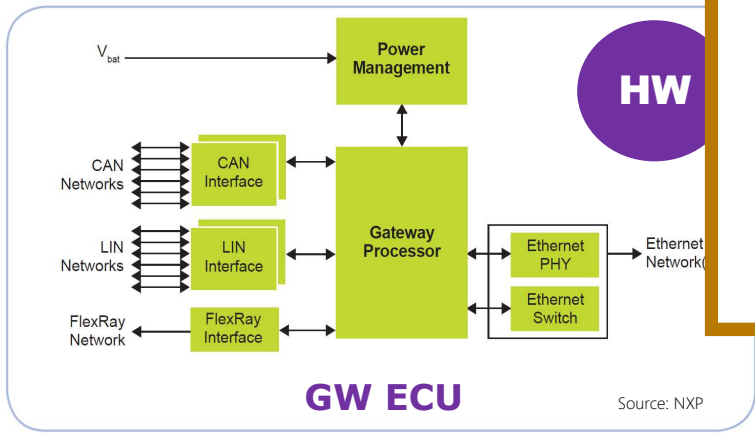
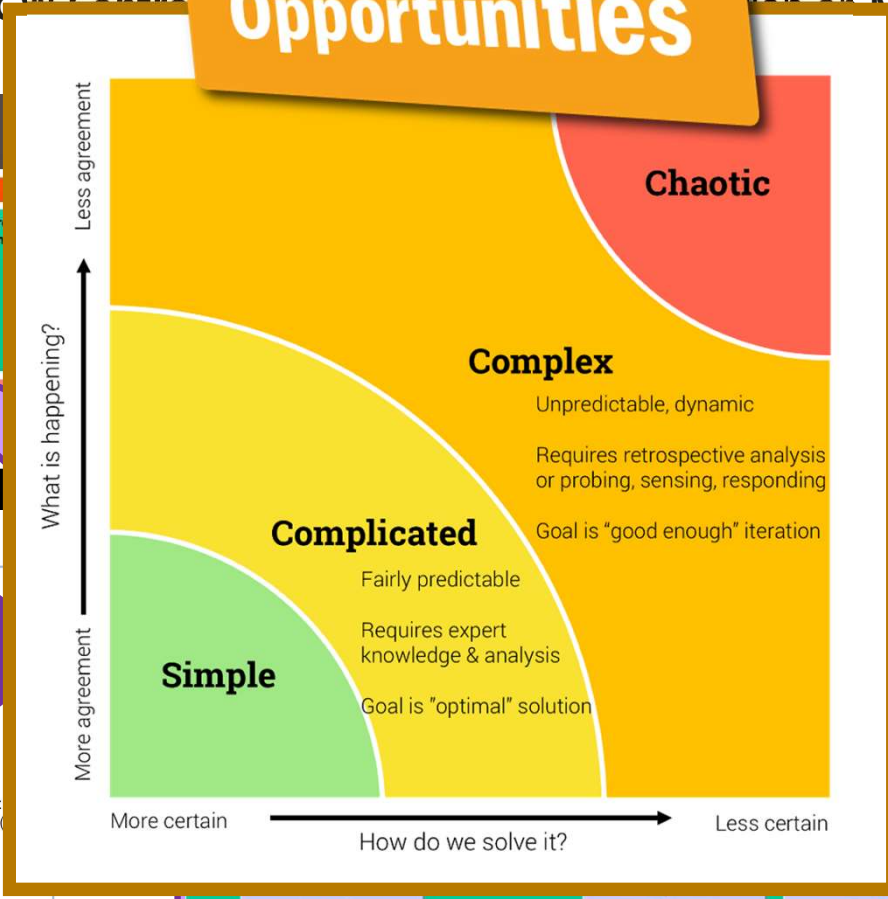


3.1 State of the Art of Gateway Controllers

Nowadays, the solution for Automotive GW Controller is based on MCU/SoC devices



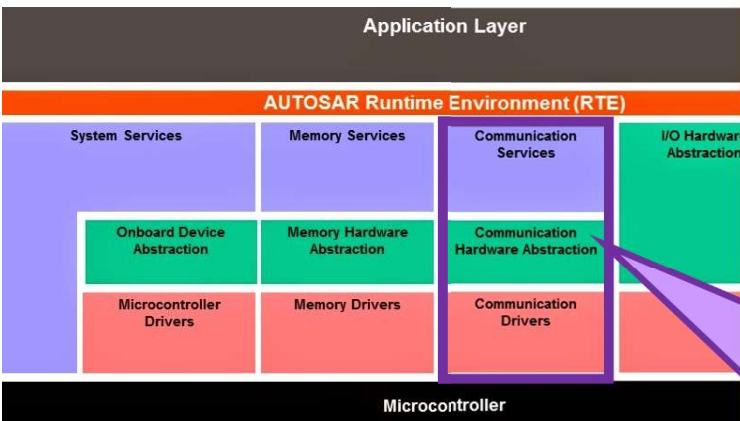
Opportunities



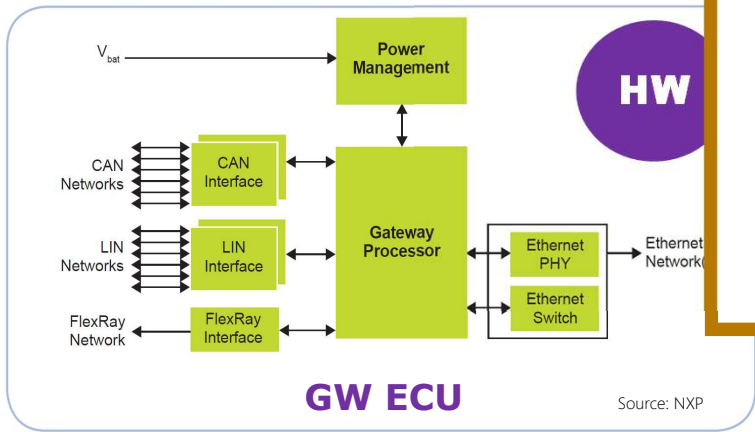
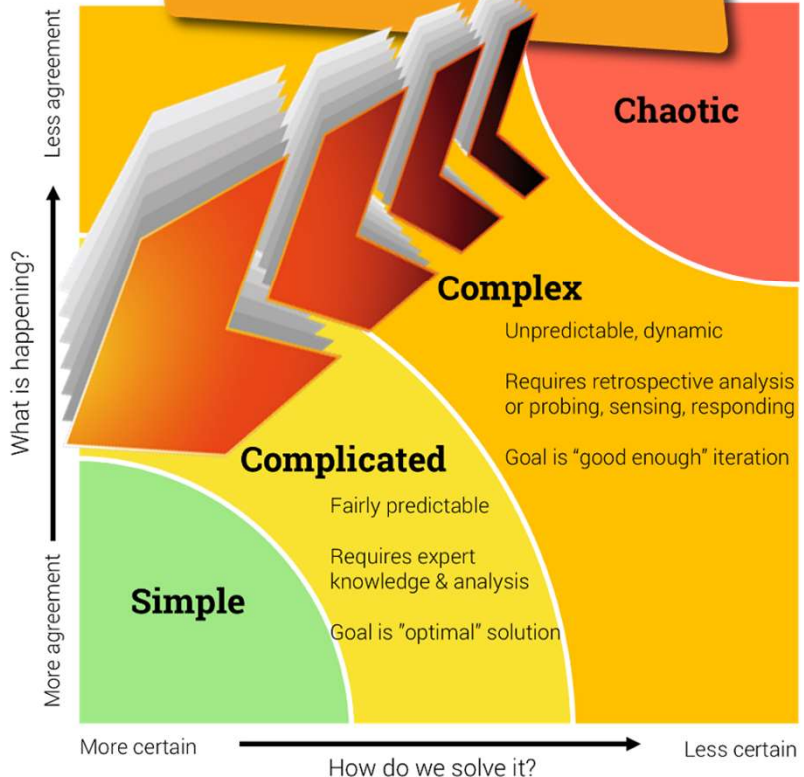
Source: NXP

3.1 State of the Art of Gateway Controllers

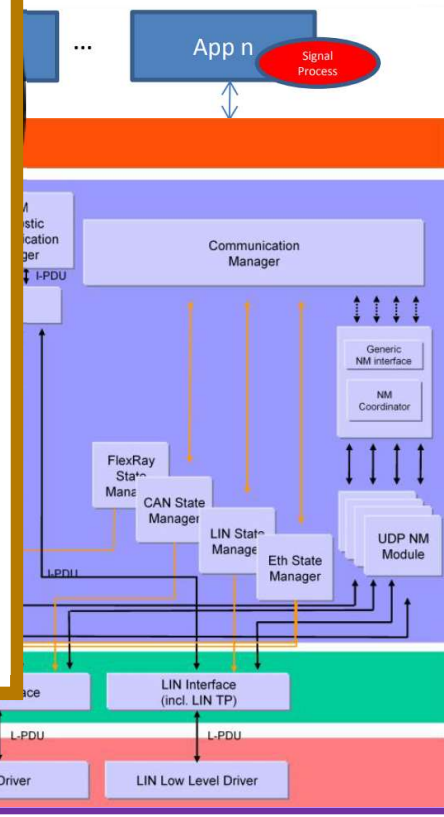
Nowadays, the solution for Automotive GW Controller is based on MCU/SoC devices



Opportunities

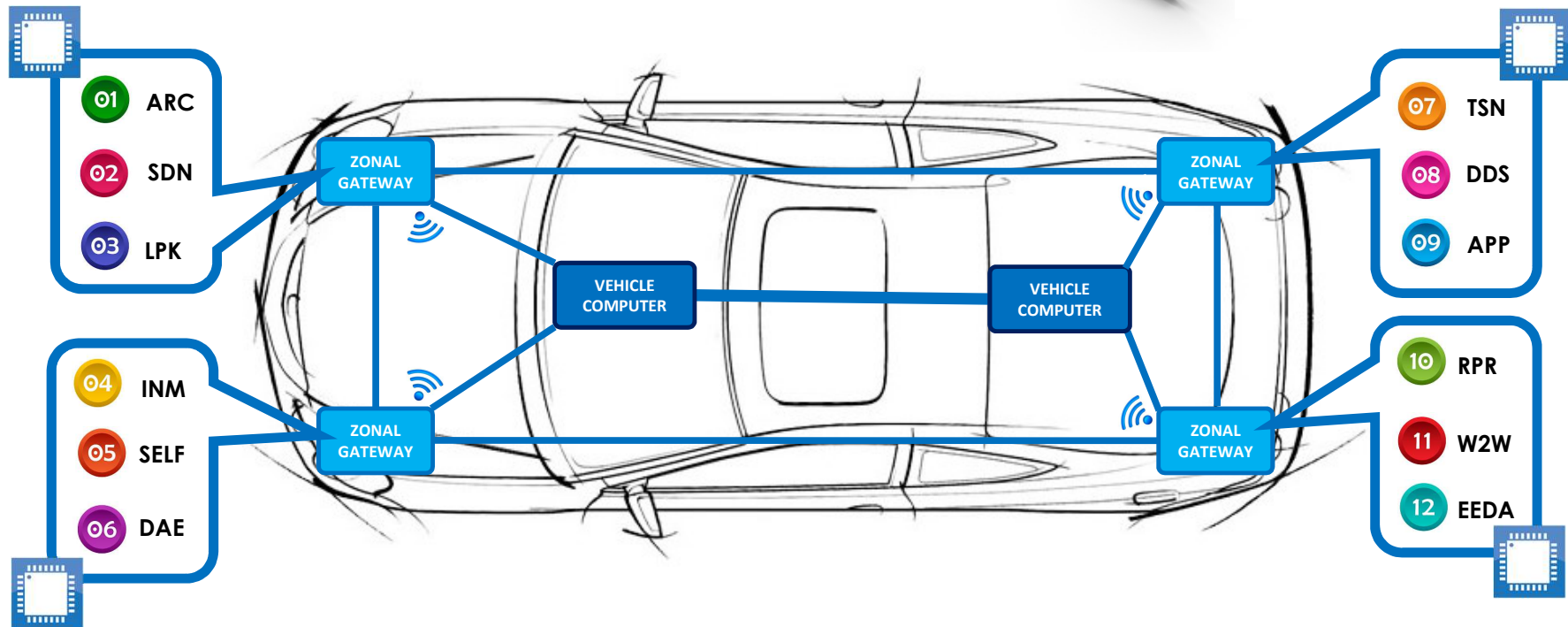


SW



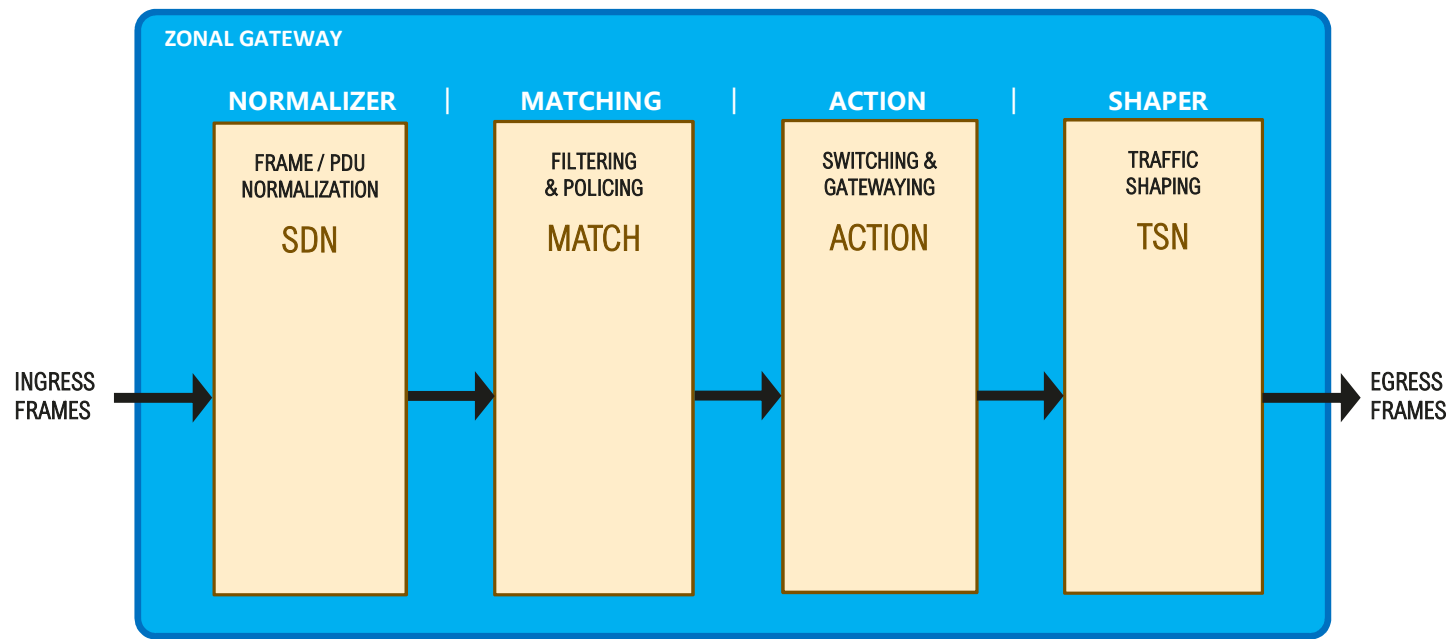
3.2 The 12 HW-centric Innovations for Networking & App Processing deployed in eGW SoC

The **Game Changer** is a new SoC Architecture carefully crafted for networking and accelerating the execution of applications in Zonal GWs aimed at reducing SW complexity and offloading the CPU. The exploration of innovative HW Accelerators (HWAs) is in the **DNA of our Research**. Up to 12 HW innovations have been developed to be integrated in NG Zonal GW SoCs.



3.3 INNOVATION 01 : ARC – Our Elastic Network SoC Reference Architecture (eGW)

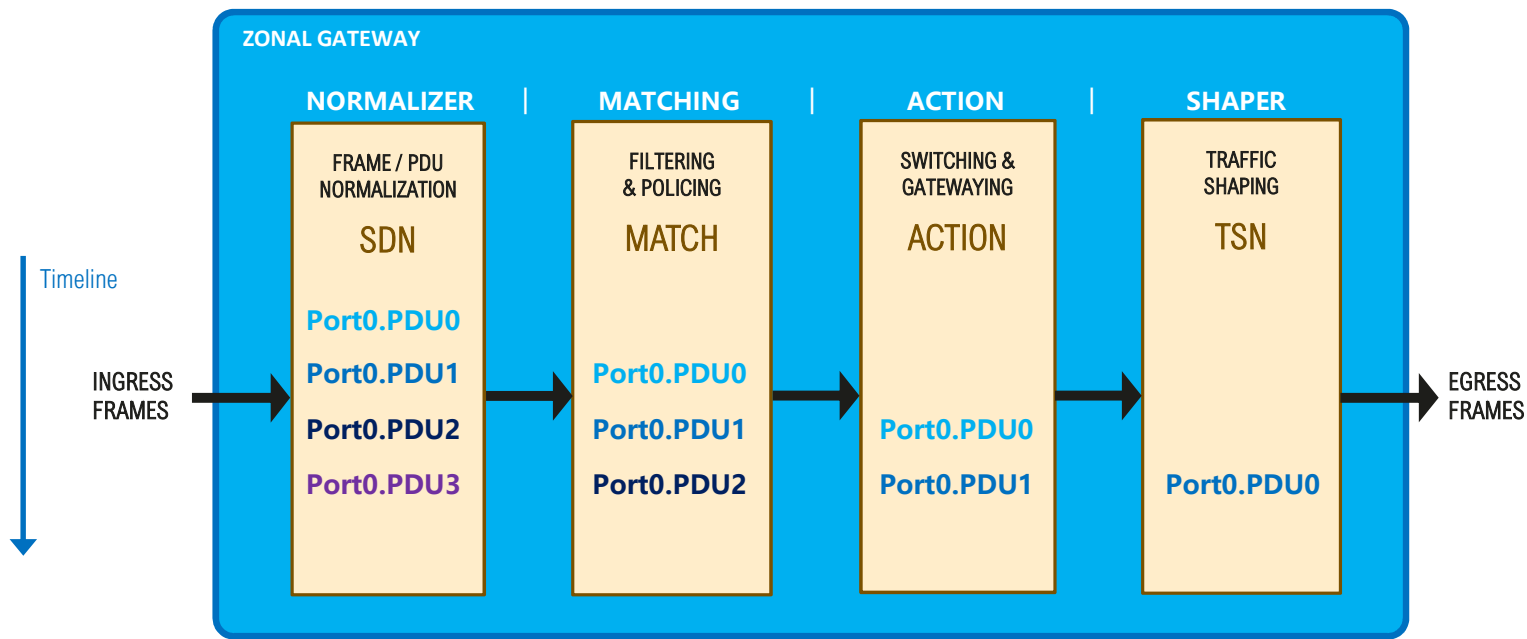
- Modeling of a four-stage **pseudo-pipeline** as regular data path pioneering a **HW-centric inline ingress-to-egress** processing through HWAs by **exploiting parallelism**
- Innovative **PDU normalization** (SDN parsing) as first stage and **traffic scheduling/shaping** as last stage (ns/ μ s time handling) in between the typical frame processing stages of Matching & Action (M&A)



Rationale: Effective balance of high-performance, flexibility/versatility, modularity/reusability and scalability

3.3 INNOVATION 01 : ARC – Our Elastic Network SoC Reference Architecture (eGW)

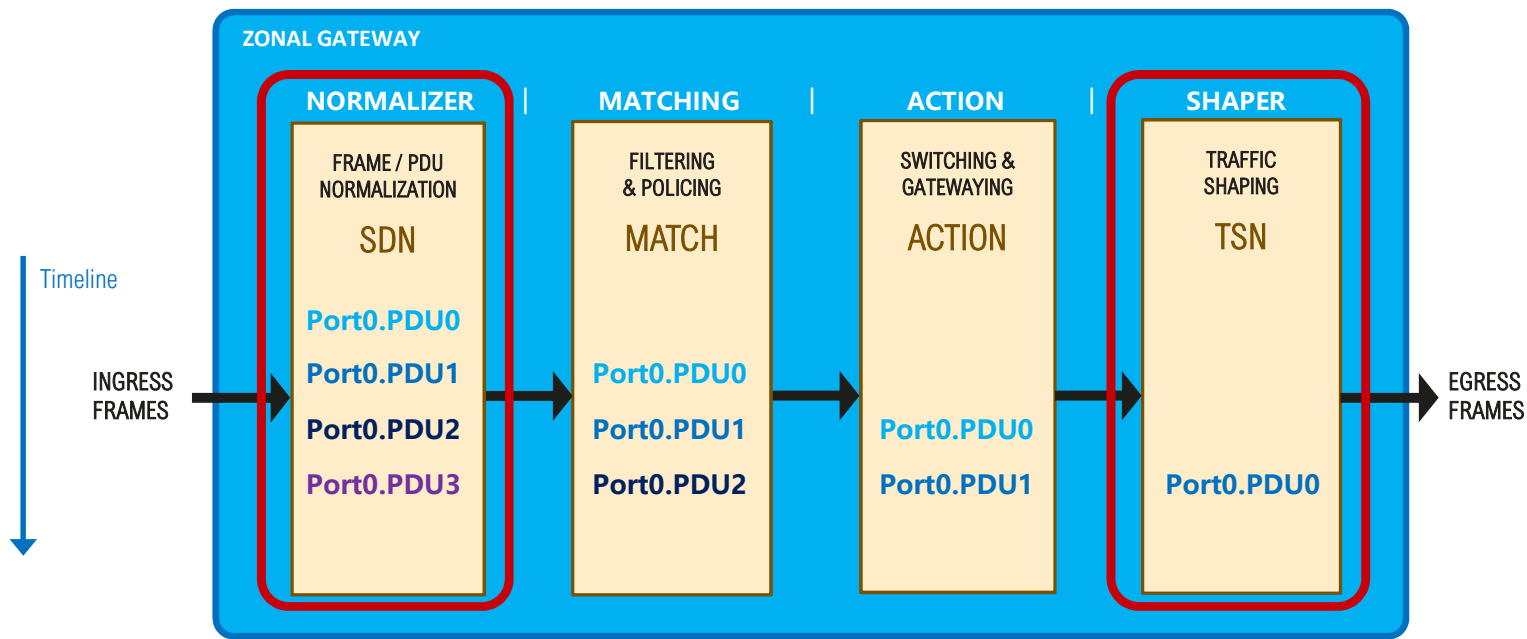
- Modeling of a four-stage **pseudo-pipeline** as regular data path pioneering a **HW-centric inline ingress-to-egress** processing through HWAs by **exploiting parallelism**
- Innovative **PDU normalization** (SDN parsing) as first stage and **traffic scheduling/shaping** as last stage (ns/ μ s time handling) in between the typical frame processing stages of Matching & Action (M&A)



Rationale: Effective balance of high-performance, flexibility/versatility, modularity/reusability and scalability

3.3 INNOVATION 01 : ARC – Our Elastic Network SoC Reference Architecture (eGW)

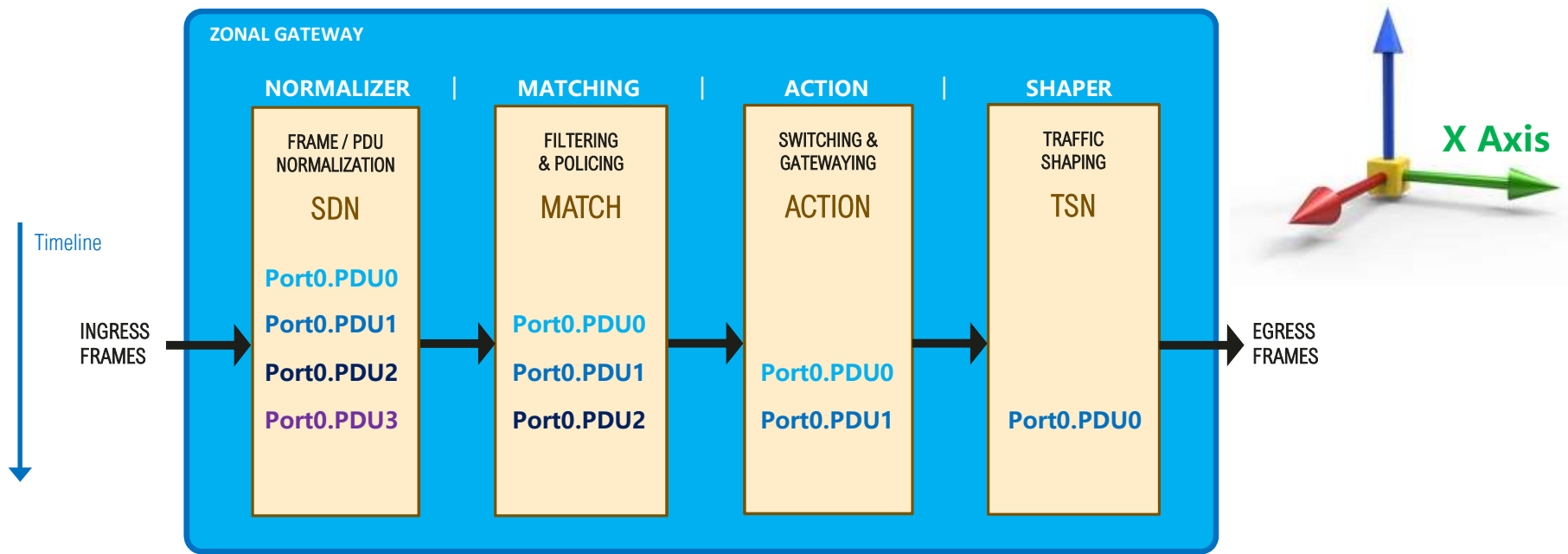
- Modeling of a four-stage **pseudo-pipeline** as regular data path pioneering a **HW-centric inline ingress-to-egress** processing through HWAs by **exploiting parallelism**
- Innovative **PDU normalization** (SDN parsing) as first stage and **traffic scheduling/shaping** as last stage (ns/ μ s time handling) in between the typical frame processing stages of Matching & Action (M&A)



Rationale: Effective balance of high-performance, flexibility/versatility, modularity/reusability and scalability

3.3 INNOVATION 01 : ARC – Our Elastic Network SoC Reference Architecture (eGW)

- Modeling of a four-stage **pseudo-pipeline** as regular data path pioneering a **HW-centric inline ingress-to-egress** processing through HWAs by **exploiting parallelism**
- Innovative **PDU normalization** (SDN parsing) as first stage and **traffic scheduling/shaping** as last stage (ns/ μ s time handling) in between the typical frame processing stages of Matching & Action (M&A)

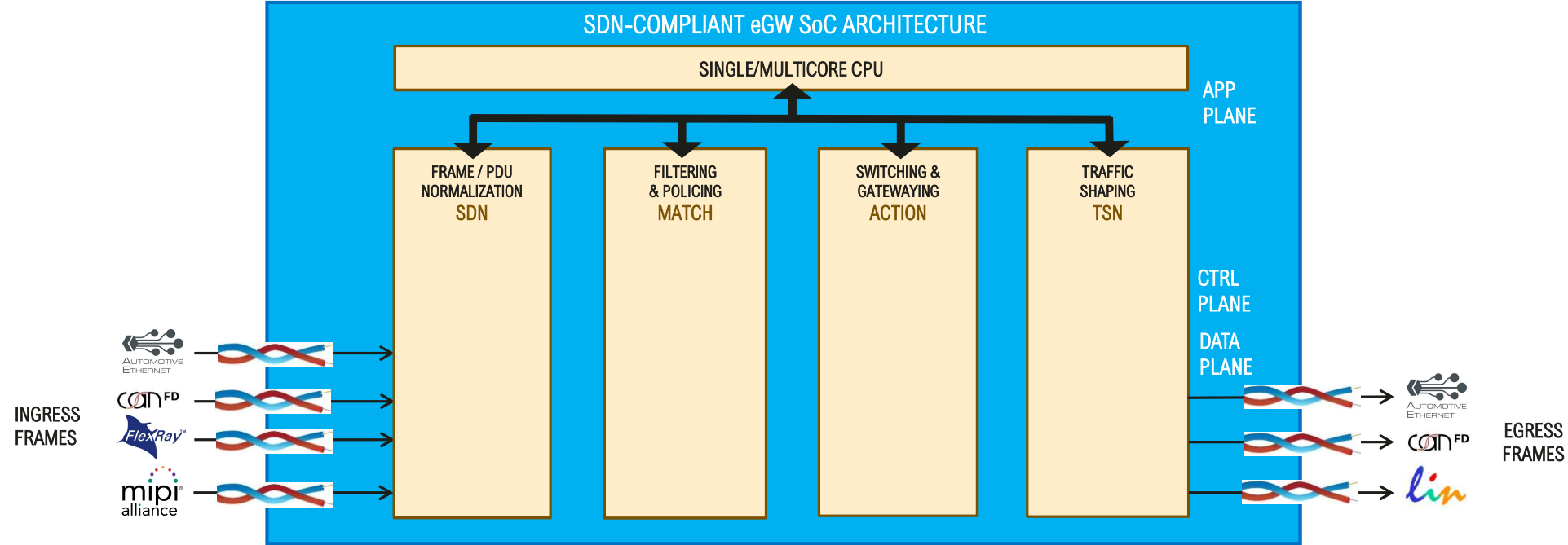


Rationale: Effective balance of high-performance, flexibility/versatility, modularity/reusability and scalability



3.3 INNOVATION 02 : SDN – Software Defined Networking paradigm moved onchip

- SDN paradigm **adopted at IVN level within Network SoC** aligned with Software-Defined Vehicle (SDV) concept
- SDN planes abstracted and deployed **onchip through PDU normalization**
- Frames Normalization abstracted as a much **broader processing concept** than parsing/deparsing (**beyond P4/PISA !**)



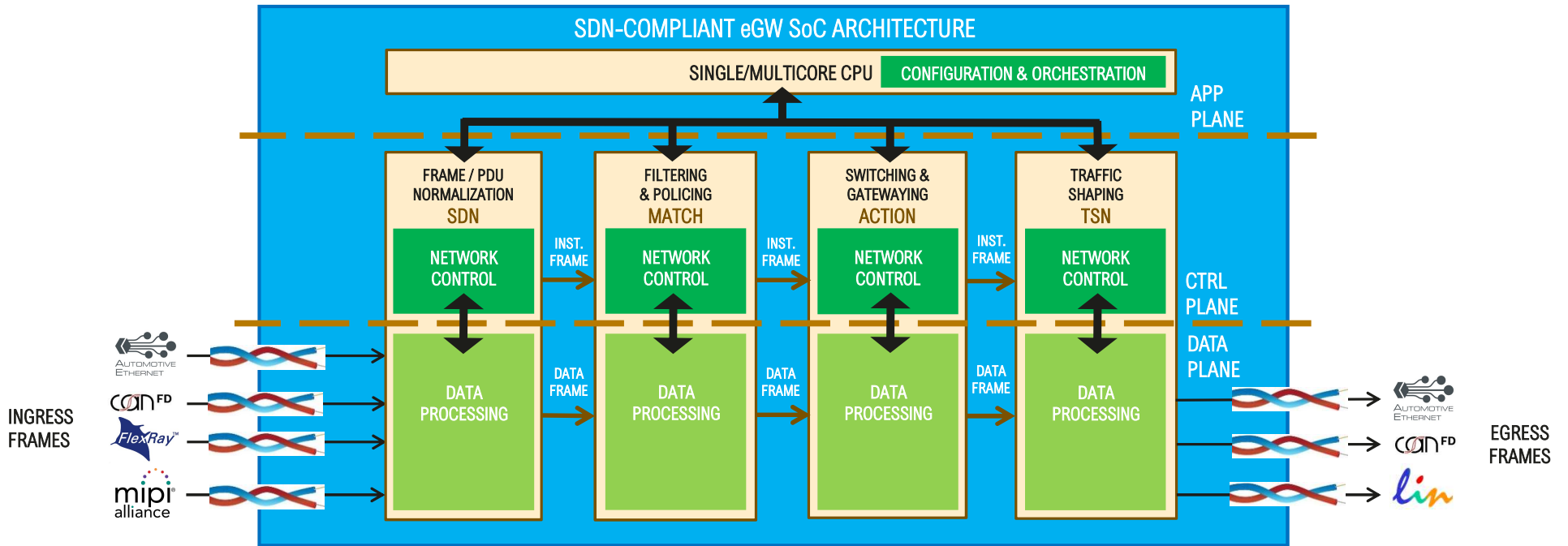
Rationale: Effective decoupling of data (data forwarding from data plane) and metadata (network orchestration from control plane) in **heterogeneous PDUs** in order to make them **protocol agnostic**.





3.3 INNOVATION 02 : SDN – Software Defined Networking paradigm moved onchip

- SDN paradigm **adopted at IVN level within Network SoC** aligned with Software-Defined Vehicle (SDV) concept
- SDN planes abstracted and deployed **onchip through PDU normalization**
- Frames Normalization abstracted as a much **broader processing concept** than parsing/deparsing (**beyond P4/PISA !**)



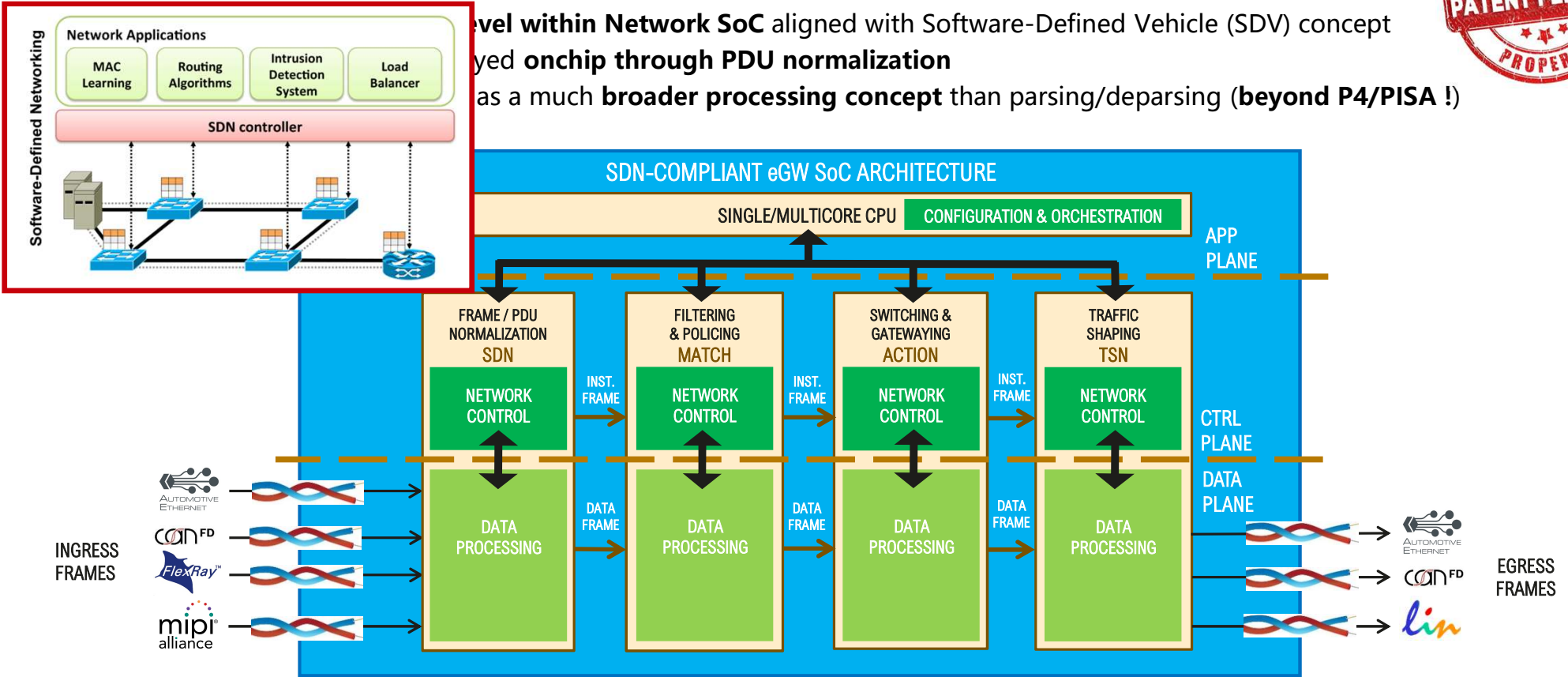
Rationale: Effective decoupling of data (data forwarding from data plane) and metadata (network orchestration from control plane) in **heterogeneous PDUs** in order to make them **protocol agnostic**.



3.3 INNOVATION 02 : SDN – Software Defined Networking paradigm moved onchip

level within Network SoC aligned with Software-Defined Vehicle (SDV) concept
implemented onchip through PDU normalization

as a much broader processing concept than parsing/deparsing (beyond P4/PISA !)

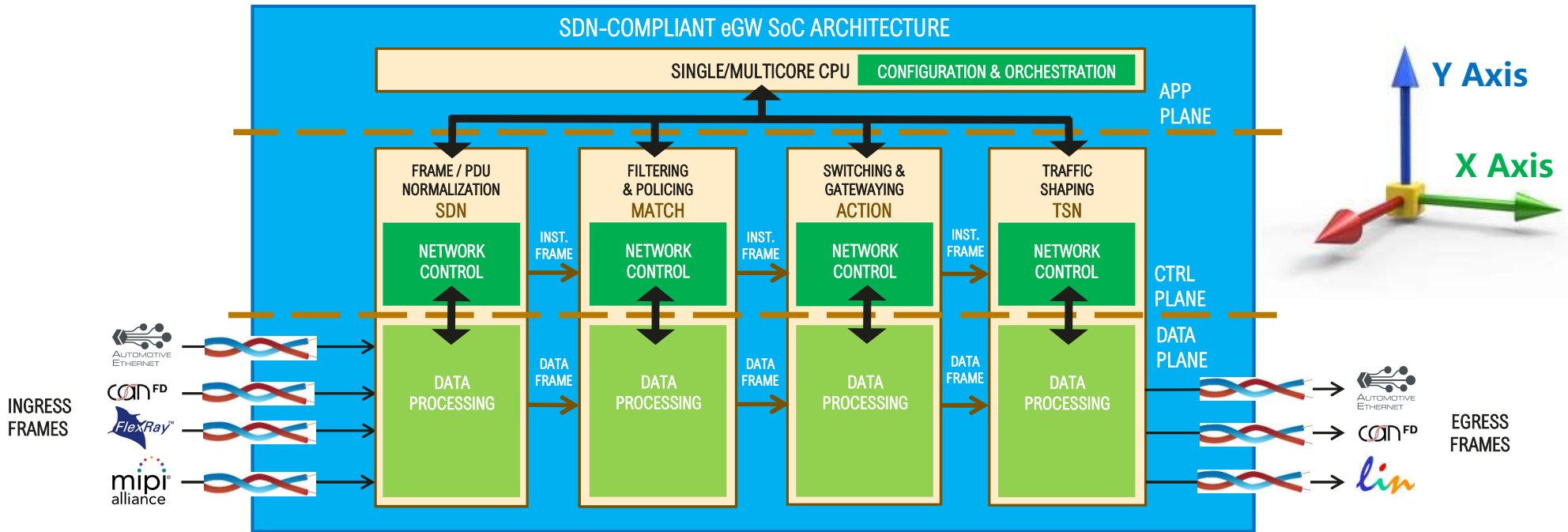


Rationale: Effective decoupling of data (data forwarding from data plane) and metadata (network orchestration from control plane) in **heterogeneous PDUs** in order to make them **protocol agnostic**.



3.3 INNOVATION 02 : SDN – Software Defined Networking paradigm moved onchip

- SDN paradigm **adopted at IVN level within Network SoC** aligned with Software-Defined Vehicle (SDV) concept
- SDN planes abstracted and deployed **onchip through PDU normalization**
- Frames Normalization abstracted as a much **broader processing concept** than parsing/deparsing (**beyond P4/PISA !**)



Rationale: Effective decoupling of data (data forwarding from data plane) and metadata (network orchestration from control plane) in **heterogeneous PDUs** in order to make them **protocol agnostic**.



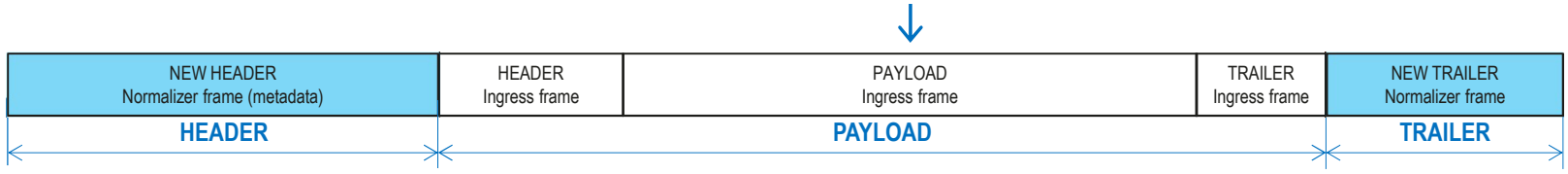
3.3 INNOVATION 02 : SDN – Software Defined Networking paradigm moved onchip

- SDN paradigm **adopted at IVN level within Network SoC** aligned with Software-Defined Vehicle (SDV) concept
- SDN planes abstracted and deployed **onchip through PDU normalization**
- Frames Normalization abstracted as a much **broader processing concept** than parsing/deparsing (**beyond P4/PISA !**)

(1) ORIGINAL INGRESS DATA FRAME (CAN, LIN, ETH, MIPI, etc.):



(2) A FIRST IDEA ABOUT NORMALIZATION FRAME (NW OSI MODEL INSIGHT):



Rationale: Effective decoupling of data (data forwarding from data plane) and metadata (network orchestration from control plane) in **heterogeneous PDUs** in order to make them **protocol agnostic**.



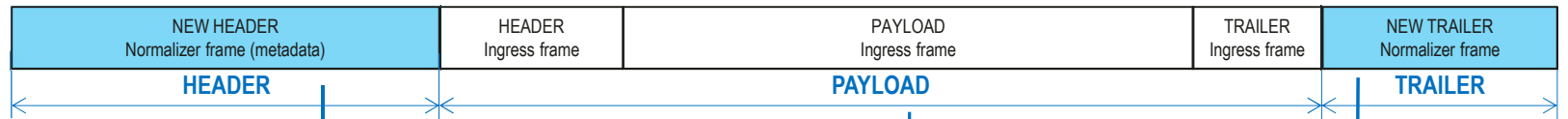
3.3 INNOVATION 02: SDN – Software Defined Networking paradigm moved onchip

- SDN paradigm **adopted at IVN level within Network SoC** aligned with Software-Defined Vehicle (SDV) concept
- SDN planes abstracted and deployed **onchip through PDU normalization**
- Frames Normalization abstracted as a much **broader processing concept** than parsing/deparsing (**beyond P4/PISA !**)

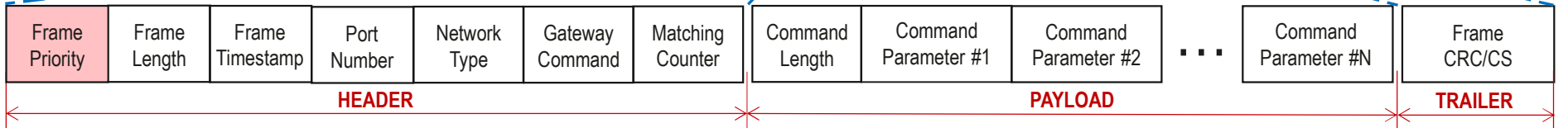
(1) ORIGINAL INGRESS DATA FRAME (CAN, LIN, ETH, MIPI, etc.):



(2) A FIRST IDEA ABOUT NORMALIZATION FRAME (NW OSI MODEL INSIGHT):



(3) NORMALIZATION FRAME OPTIMIZED THROUGH SDN (DATA & CTRL PLANES)



Rationale: Effective decoupling of data (data forwarding from data plane) and metadata (network orchestration from control plane) in **heterogeneous PDUs** in order to make them **protocol agnostic**.

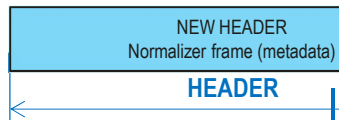


3.3 INNOVATION 02 : SDN – Software Defined Networking paradigm moved onchip

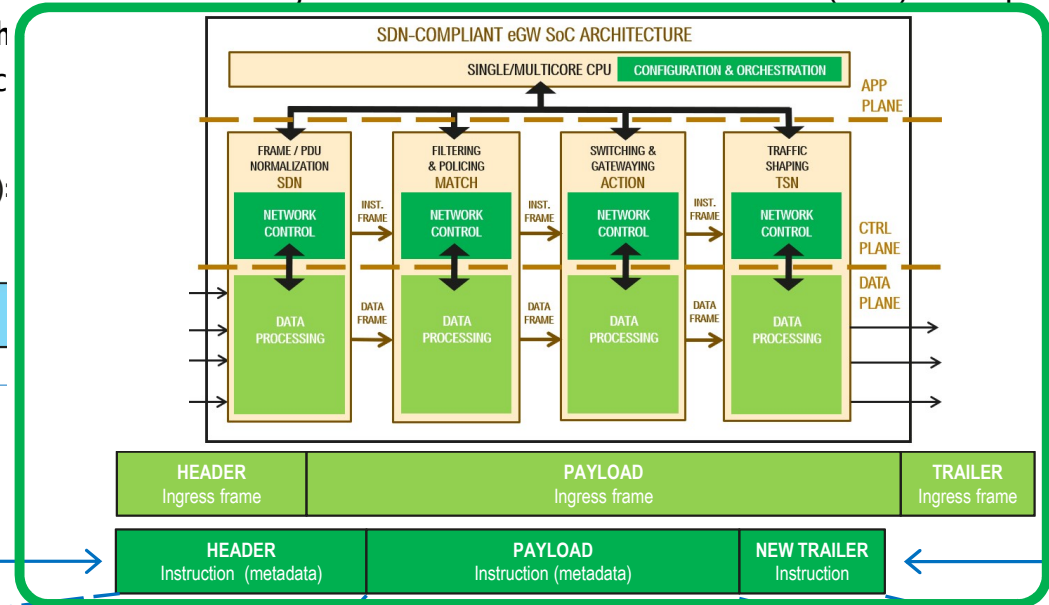
- SDN paradigm adopted at IVN level within Network SoC aligned with Software-Defined Vehicle (SDV) concept
- SDN planes abstracted and deployed onchip
- Frames Normalization abstracted as a muc

(1) ORIGINAL INGRESS DATA FRAME (CAN, LIN, ETH, MIPI, etc.):

(2) A FIRST IDEA ABOUT NORMALIZATION FRAME (NW OSI MODEL INSIGHT):



(3) NORMALIZATION FRAME OPTIMIZED THROUGH SDN (DATA & CTRL PLANES)

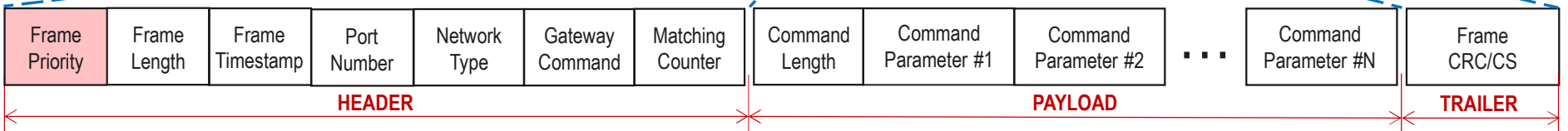


!A !)



@ DATA PLANE

@ CTRL PLANE

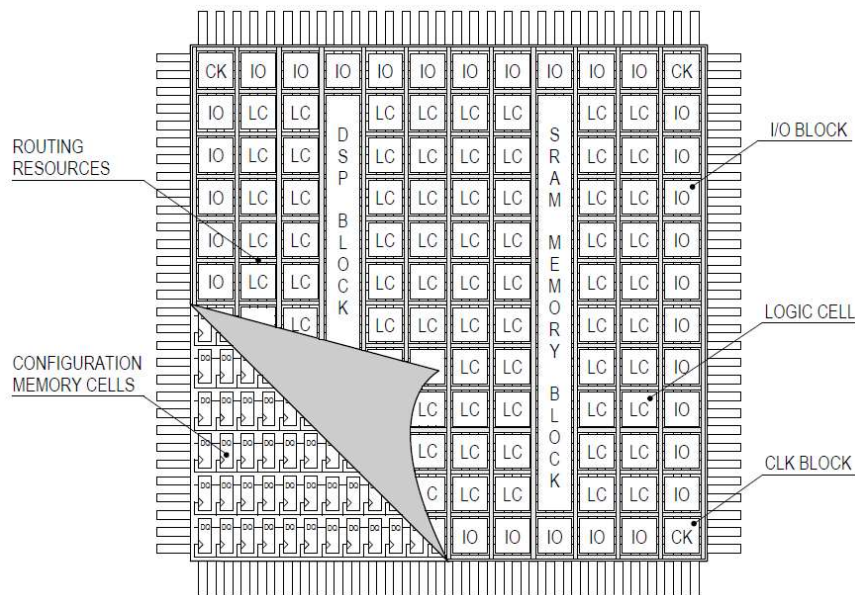


Rationale: Effective decoupling of data (data forwarding from data plane) and metadata (network orchestration from control plane) in heterogeneous PDUs in order to make them protocol agnostic.



3.3 INNOVATION 03 : LPK – Loopback to optimize iterative/recursive processes and HW reuse

- Loopback brings **flexibility** (insight from FPGA-based combinational/sequential digital circuits based on LUT+FF+Interconnect)
- **Interconnects** of network processing stages through **queues and crossbars**
- Still **effective in latency** when combined with **prioritization and arbitration strategy** (refer to innovation 06 : DAE)
- Loopback as a much **deeper concept than frames recirculation** (used in Intel Tofino SoC family)

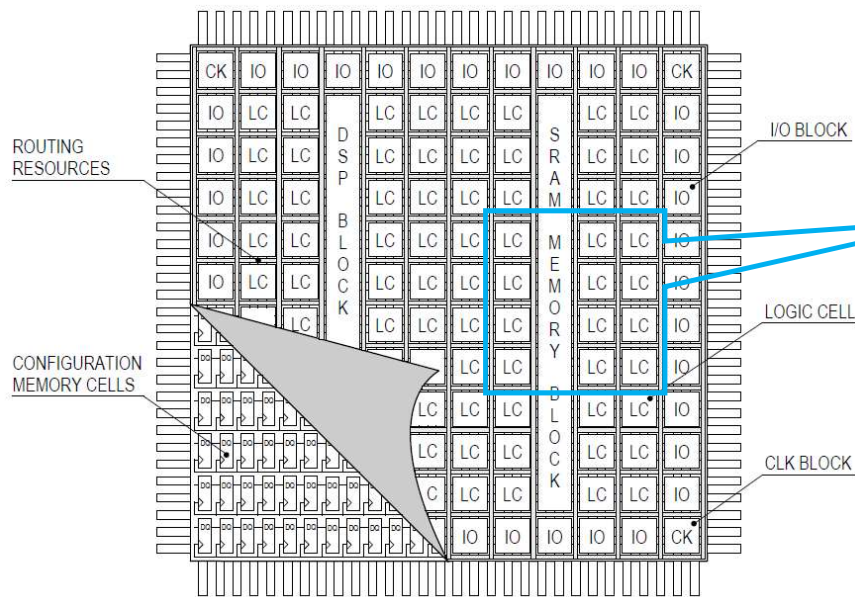


Generic SRAM-based FPGA Architecture

Rationale: Elastic processing path for frames, **back and forth**, not only fast forward but also fast feedback (with prioritization and arbitration strategy) and well balanced with other KPIs (high-performance, latency, jitter, bandwidth, throughput)

3.3 INNOVATION 03 : LPK – Loopback to optimize iterative/recursive processes and HW reuse

- Loopback brings **flexibility** (insight from FPGA-based combinational/sequential digital circuits based on LUT+FF+Interconnect)
- **Interconnects** of network processing stages through **queues and crossbars**
- Still **effective in latency** when combined with **prioritization and arbitration strategy** (refer to innovation 06 : DAE)
- Loopback as a much **deeper concept than frames recirculation** (used in Intel Tofino SoC family)



Generic SRAM-based FPGA Architecture

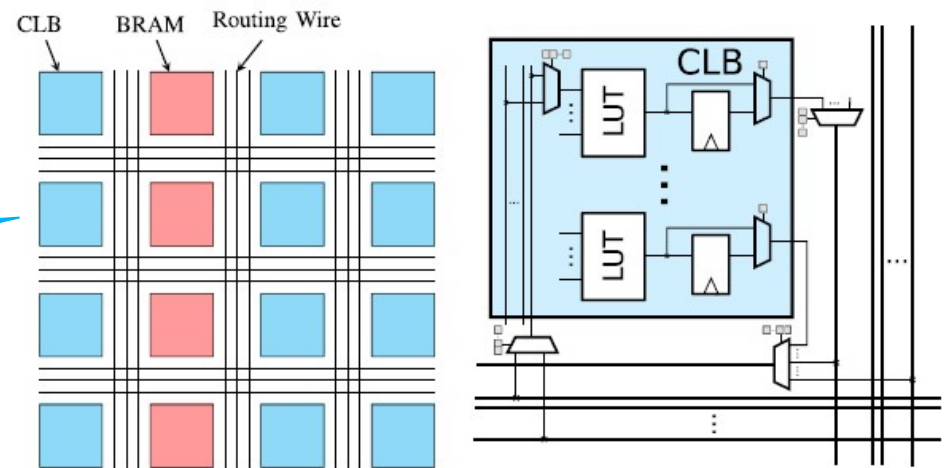
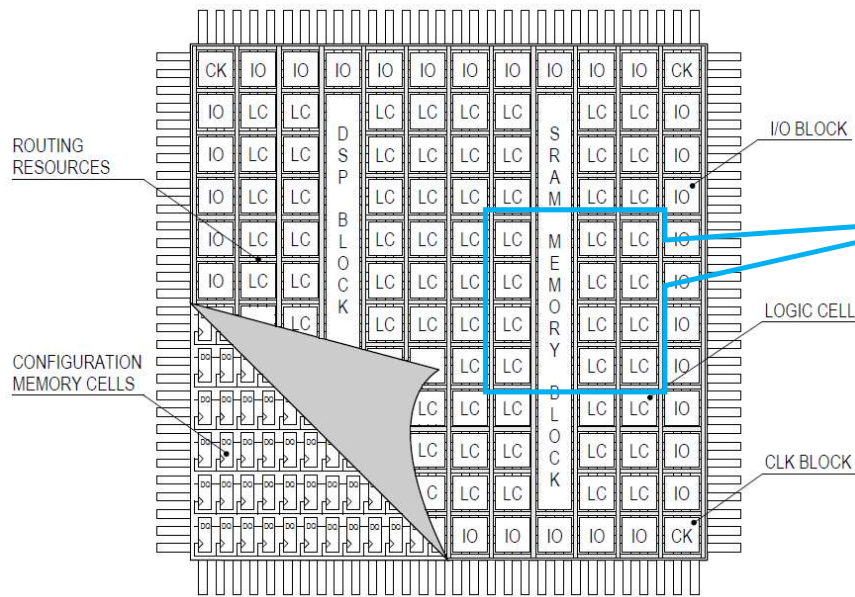


Figure. (Left) An FPGA consisting of logic cells (LC) / configurable logic blocks (CLB) and Block RAM (BRAM), along with programmable routing to interconnect them. (Right) Each CLB contains Look Up Tables (LUT) and flip-flops (FF). Routing Muxes are configured to interconnect elements with each block or interblock routing wires.

Rationale: Elastic processing path for frames, **back and forth**, not only fast forward but also fast feedback (with prioritization and arbitration strategy) and well balanced with other KPIs (high-performance, latency, jitter, bandwidth, throughput)

3.3 INNOVATION 03 : LPK – Loopback to optimize iterative/recursive processes and HW reuse

- Loopback brings **flexibility** (insight from FPGA-based combinational/sequential digital circuits based on LUT+FF+Interconnect)
- **Interconnects** of network processing stages through **queues and crossbars**
- Still **effective in latency** when combined with **prioritization and arbitration strategy** (refer to innovation 06 : DAE)
- Loopback as a much **deeper concept than frames recirculation** (used in Intel Tofino SoC family)



Generic SRAM-based FPGA Architecture

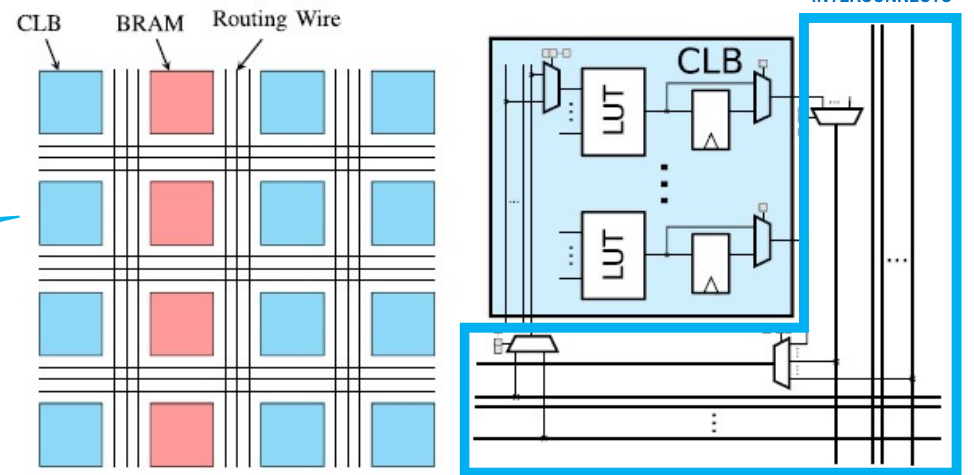
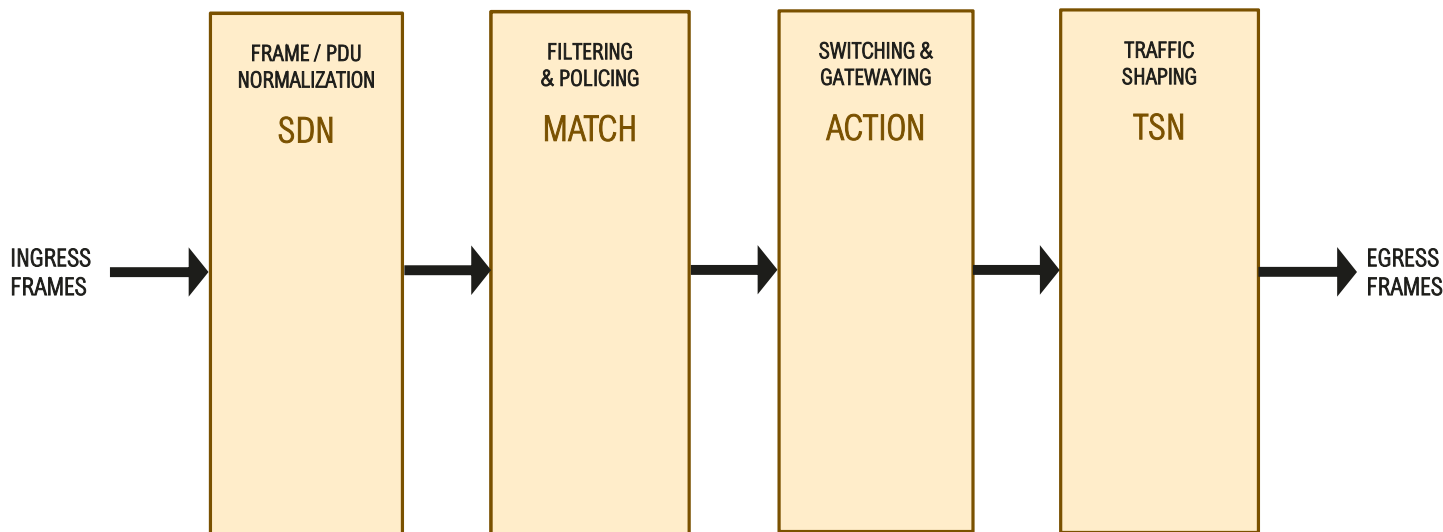


Figure. (Left) An FPGA consisting of logic cells (LC) / configurable logic blocks (CLB) and Block RAM (BRAM), along with programmable routing to interconnect them. (Right) Each CLB contains Look Up Tables (LUT) and flip-flops (FF). Routing Muxes are configured to interconnect elements with each block or interblock routing wires.

Rationale: Elastic processing path for frames, **back and forth**, not only fast forward but also fast feedback (with prioritization and arbitration strategy) and well balanced with other KPIs (high-performance, latency, jitter, bandwidth, throughput)

3.3 INNOVATION 03 : LPK – Loopback to optimize iterative/recursive processes and HW reuse

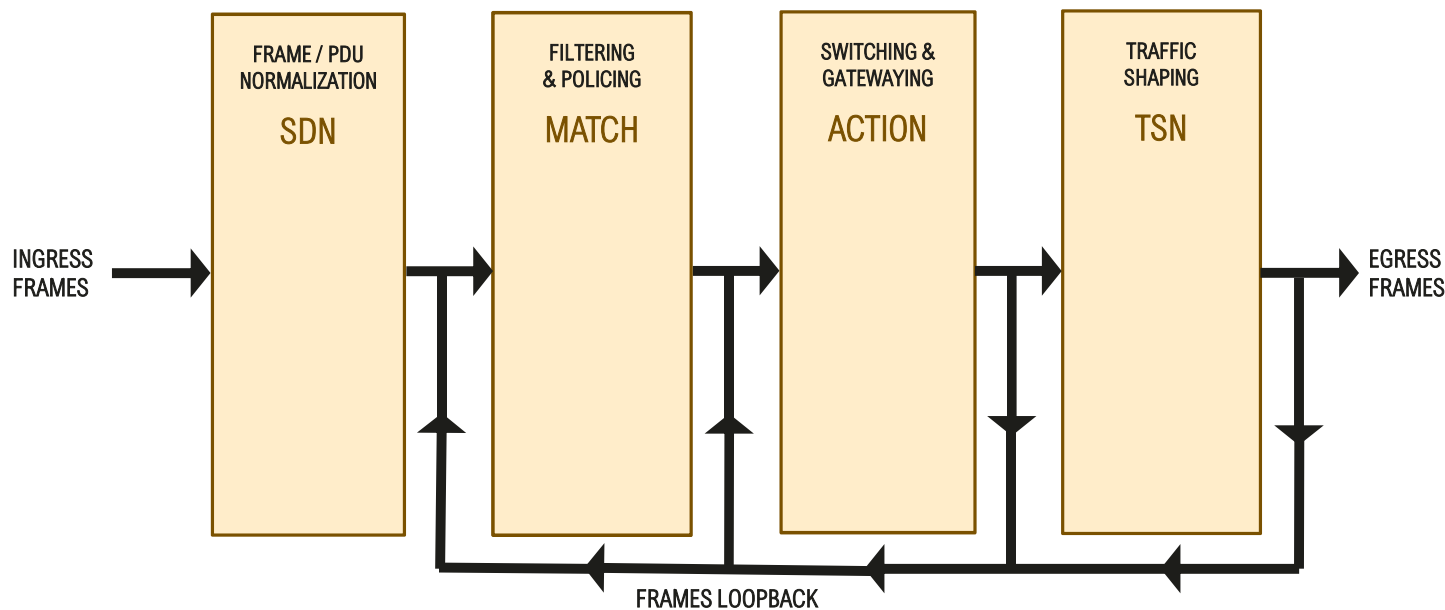
- Loopback brings **flexibility** (insight from FPGA-based combinational/sequential digital circuits based on LUT+FF+Interconnect)
- **Interconnects** of network processing stages through **queues and crossbars**
- Still **effective in latency** when combined with **prioritization and arbitration strategy** (refer to innovation 06 : DAE)
- Loopback as a much **deeper concept than frames recirculation** (used in Intel Tofino SoC family)



Rationale: Elastic processing path for frames, **back and forth**, not only fast forward but also fast feedback (with prioritization and arbitration strategy) and well balanced with other KPIs (high-performance, latency, jitter, bandwidth, throughput)

3.3 INNOVATION 03 : LPK – Loopback to optimize iterative/recursive processes and HW reuse

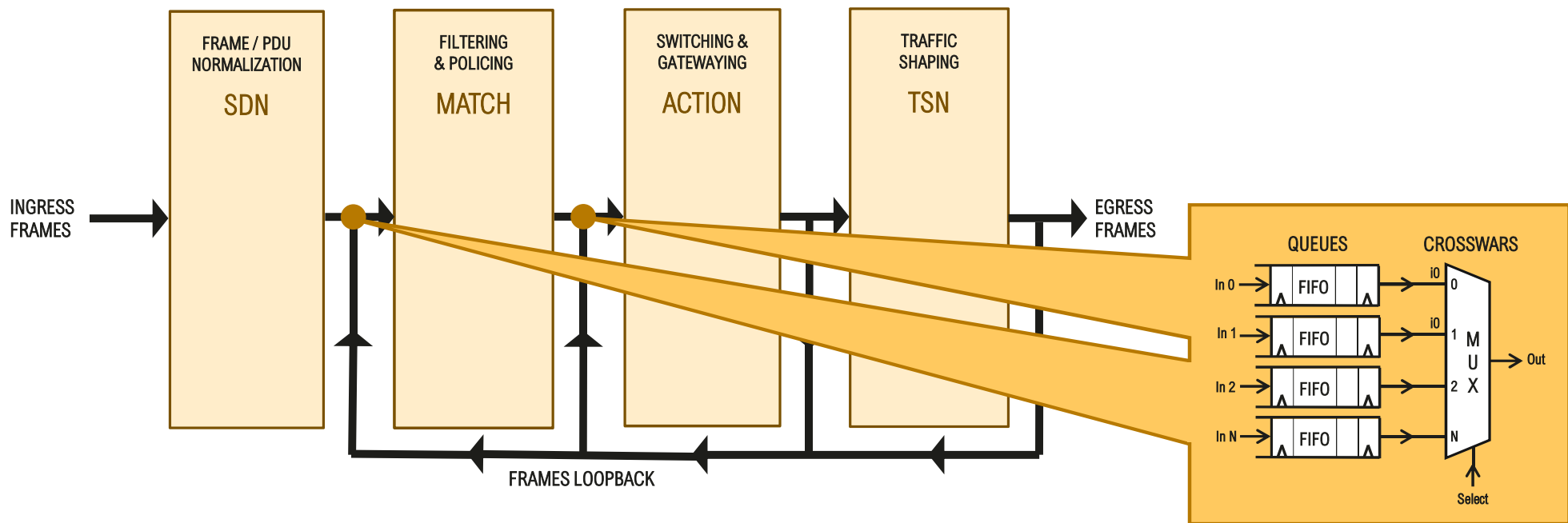
- Loopback brings **flexibility** (insight from FPGA-based combinational/sequential digital circuits based on LUT+FF+Interconnect)
- **Interconnects** of network processing stages through **queues and crossbars**
- Still **effective in latency** when combined with **prioritization and arbitration strategy** (refer to innovation 06 : DAE)
- Loopback as a much **deeper concept than frames recirculation** (used in Intel Tofino SoC family)



Rationale: Elastic processing path for frames, **back and forth**, not only fast forward but also fast feedback (with prioritization and arbitration strategy) and well balanced with other KPIs (high-performance, latency, jitter, bandwidth, throughput)

3.3 INNOVATION 03 : LPK – Loopback to optimize iterative/recursive processes and HW reuse

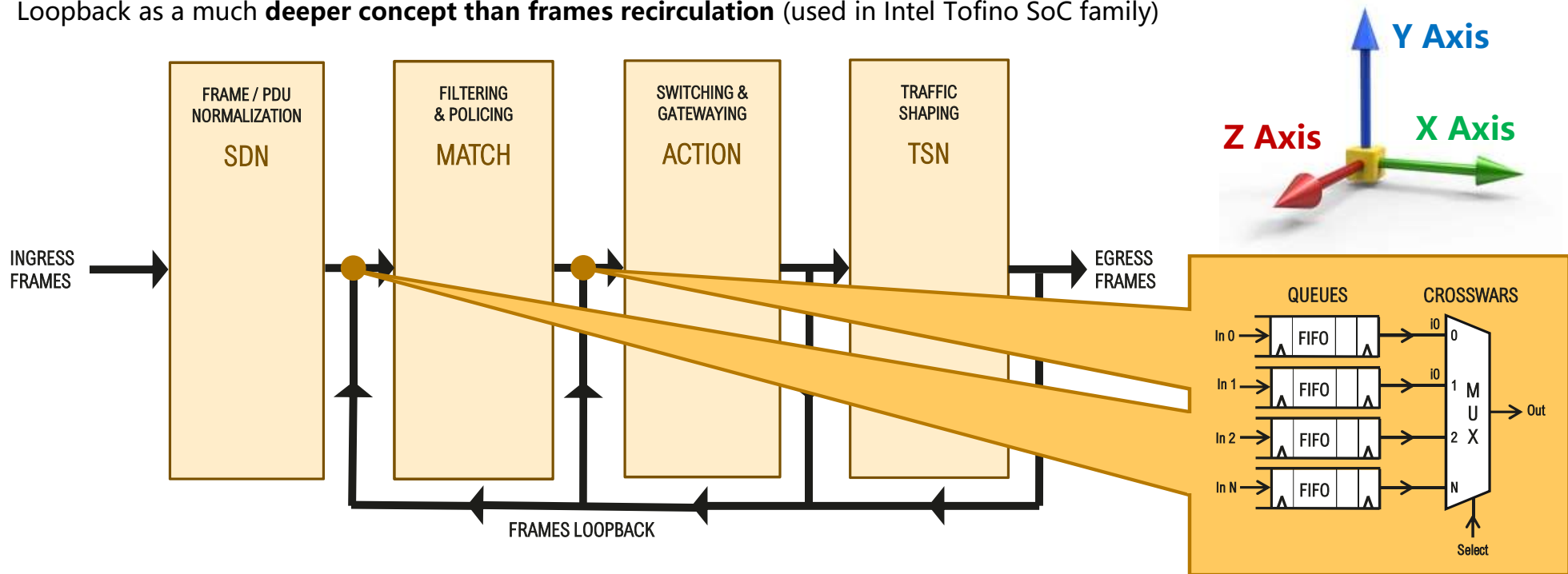
- Loopback brings **flexibility** (insight from FPGA-based combinational/sequential digital circuits based on LUT+FF+Interconnect)
- **Interconnects** of network processing stages through **queues and crossbars**
- Still **effective in latency** when combined with **prioritization and arbitration strategy** (refer to innovation 06 : DAE)
- Loopback as a much **deeper concept than frames recirculation** (used in Intel Tofino SoC family)



Rationale: Elastic processing path for frames, **back and forth**, not only fast forward but also fast feedback (with prioritization and arbitration strategy) and well balanced with other KPIs (high-performance, latency, jitter, bandwidth, throughput)

3.3 INNOVATION 03 : LPK – Loopback to optimize iterative/recursive processes and HW reuse

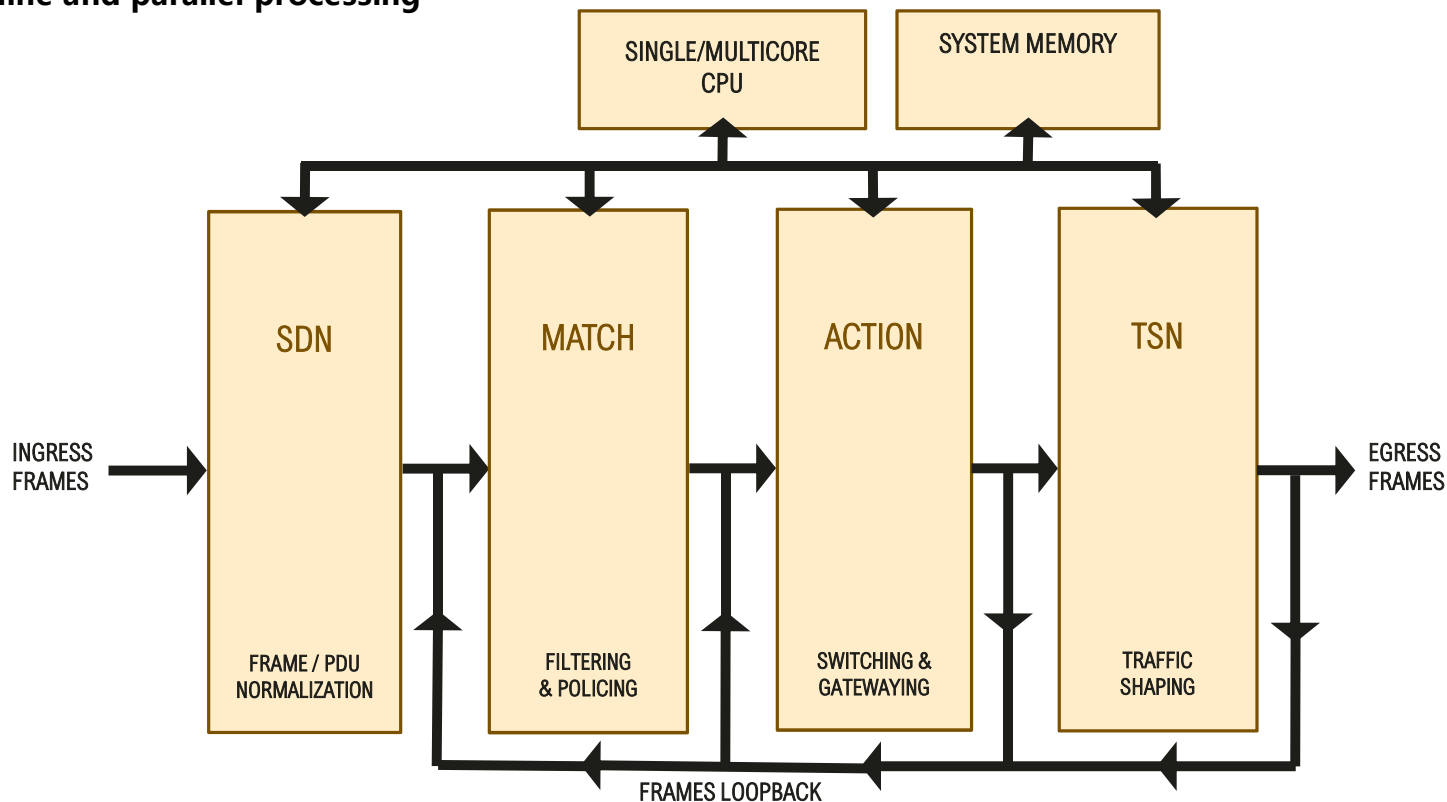
- Loopback brings **flexibility** (insight from FPGA-based combinational/sequential digital circuits based on LUT+FF+Interconnect)
- **Interconnects** of network processing stages through **queues and crossbars**
- Still **effective in latency** when combined with **prioritization and arbitration strategy** (refer to innovation 06 : DAE)
- Loopback as a much **deeper concept than frames recirculation** (used in Intel Tofino SoC family)



Rationale: Elastic processing path for frames, **back and forth**, not only fast forward but also fast feedback (with prioritization and arbitration strategy) and well balanced with other KPIs (high-performance, latency, jitter, bandwidth, throughput)

3.3 INNOVATION 04 : INM – In-memory local processing is the real enabler for offloading CPUs

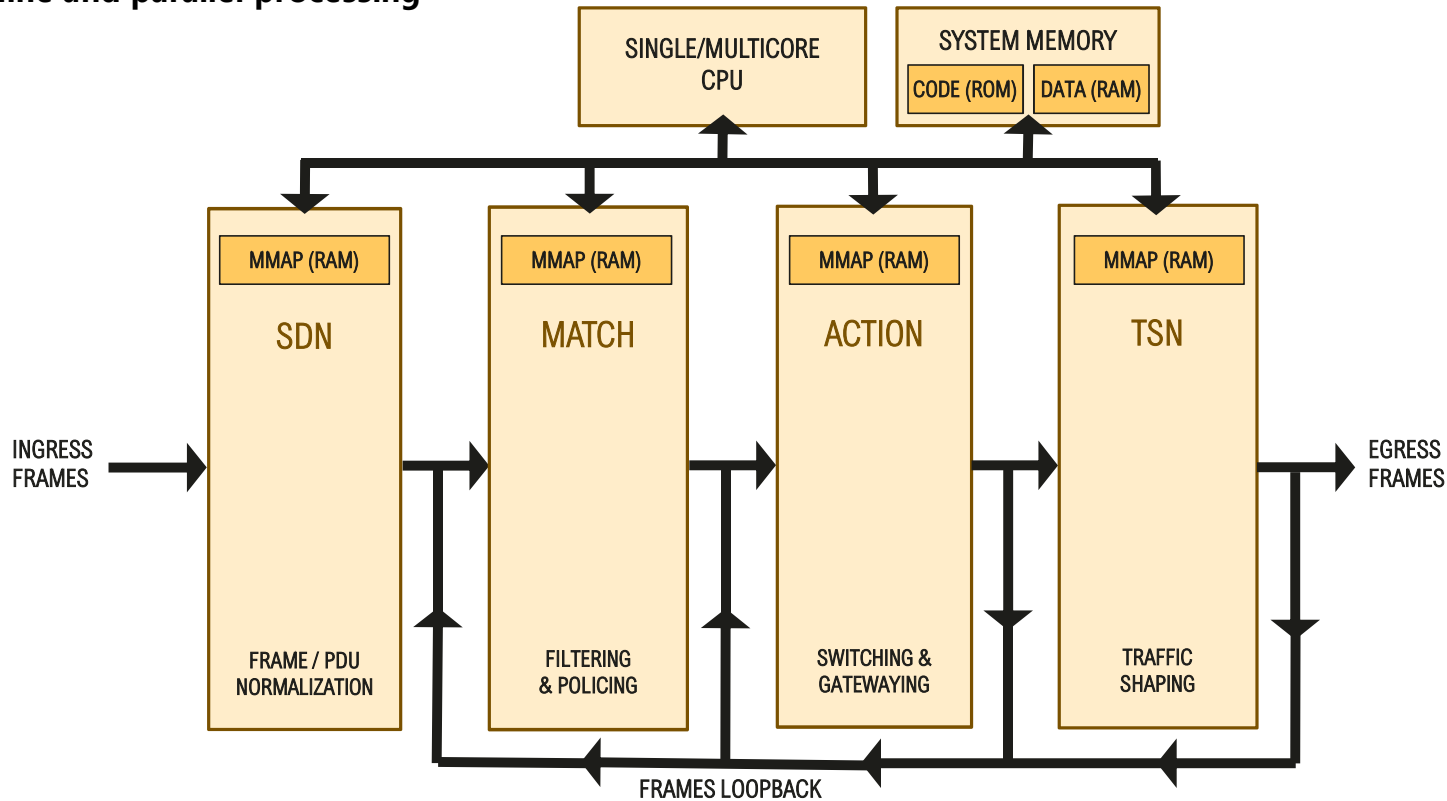
- Our HWAs synthesized through onchip **centralized Block RAM** (true dual-port RAM, FIFO) and **distributed local RAM** (LUTs, FFs) optimize **inline and parallel processing**



Rationale: Overall, much more **onchip memory** is required in a Network SoC. Onchip local memory is the best option and it pays off because data movement is expensive in terms of **energy consumption**! That is, external DDRx memory to be skipped if possible.

3.3 INNOVATION 04 : INM – In-memory local processing is the real enabler for offloading CPUs

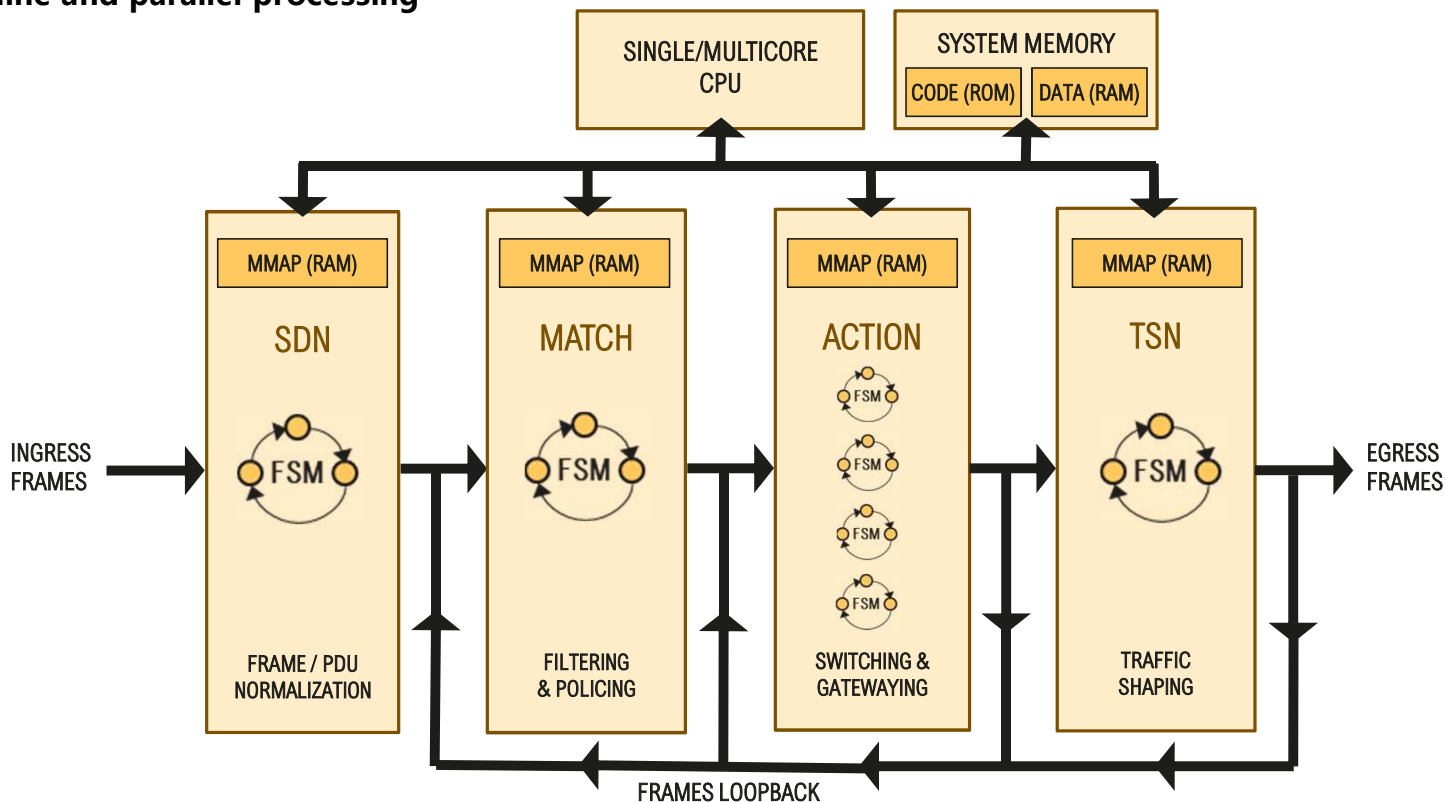
- Our HWAs synthesized through onchip **centralized Block RAM** (true dual-port RAM, FIFO) and **distributed local RAM** (LUTs, FFs) optimize **inline and parallel processing**



Rationale: Overall, much more **onchip memory** is required in a Network SoC. Onchip local memory is the best option and it pays off because data movement is expensive in terms of **energy consumption**! That is, external DDRx memory to be skipped if possible.

3.3 INNOVATION 04 : INM – In-memory local processing is the real enabler for offloading CPUs

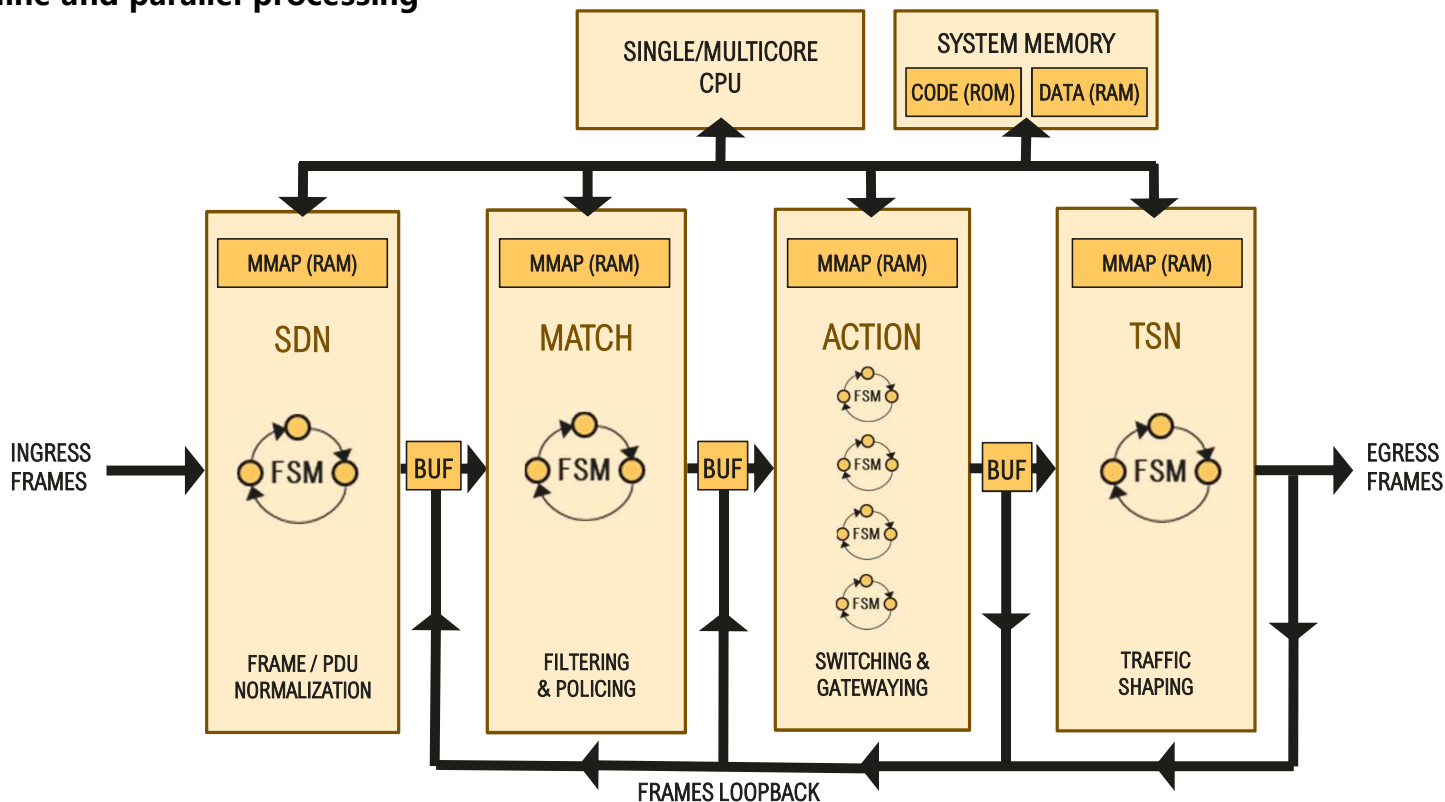
- Our HWAs synthesized through onchip **centralized Block RAM** (true dual-port RAM, FIFO) and **distributed local RAM** (LUTs, FFs) optimize **inline and parallel processing**



Rationale: Overall, much more **onchip memory** is required in a Network SoC. Onchip local memory is the best option and it pays off because data movement is expensive in terms of **energy consumption**! That is, external DDRx memory to be skipped if possible.

3.3 INNOVATION 04 : INM – In-memory local processing is the real enabler for offloading CPUs

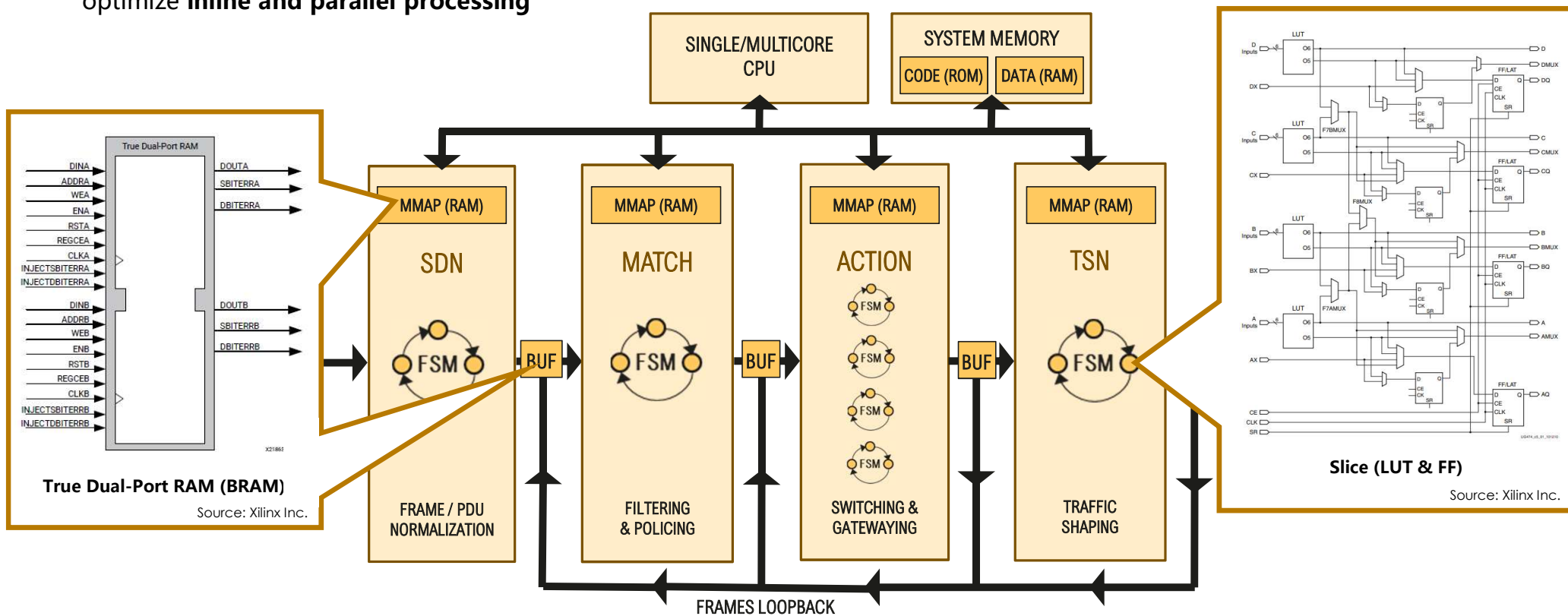
- Our HWAs synthesized through onchip **centralized Block RAM** (true dual-port RAM, FIFO) and **distributed local RAM** (LUTs, FFs) optimize **inline and parallel processing**



Rationale: Overall, much more **onchip memory** is required in a Network SoC. Onchip local memory is the best option and it pays off because data movement is expensive in terms of **energy consumption**! That is, external DDRx memory to be skipped if possible.

3.3 INNOVATION 04 : INM – In-memory local processing is the real enabler for offloading CPUs

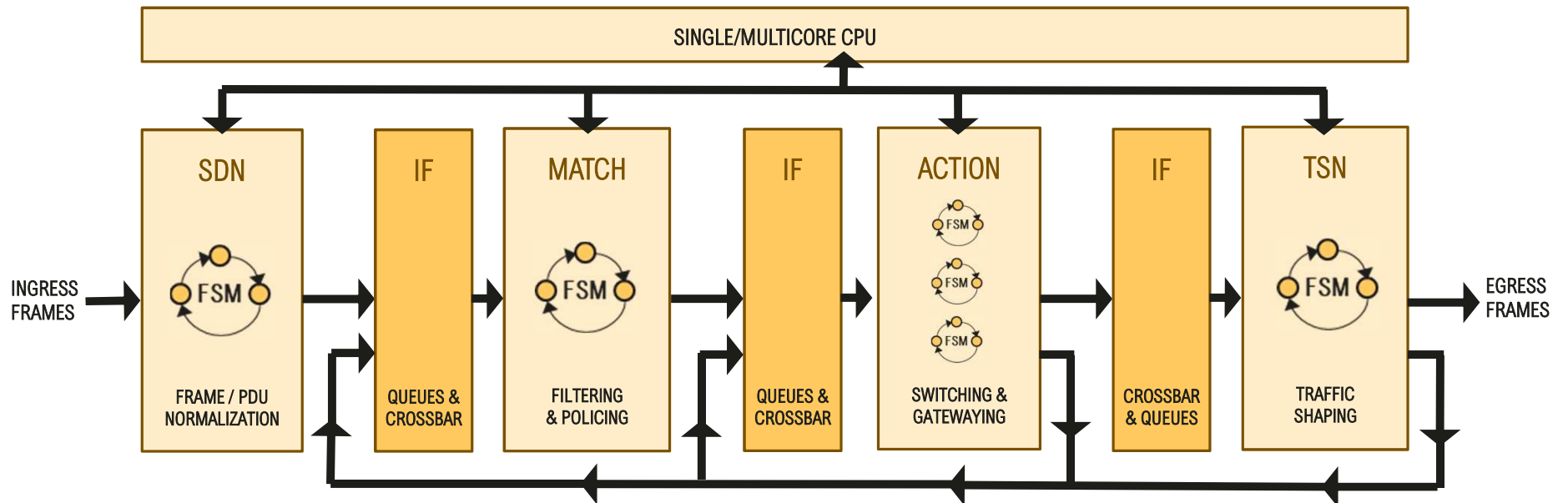
- Our HWAs synthesized through onchip **centralized Block RAM** (true dual-port RAM, FIFO) and **distributed local RAM** (LUTs, FFs) optimize **inline and parallel processing**



Rationale: Overall, much more **onchip memory** is required in a Network SoC. Onchip local memory is the best option and it pays off because data movement is expensive in terms of **energy consumption!** That is, external DDRx memory to be skipped if possible.

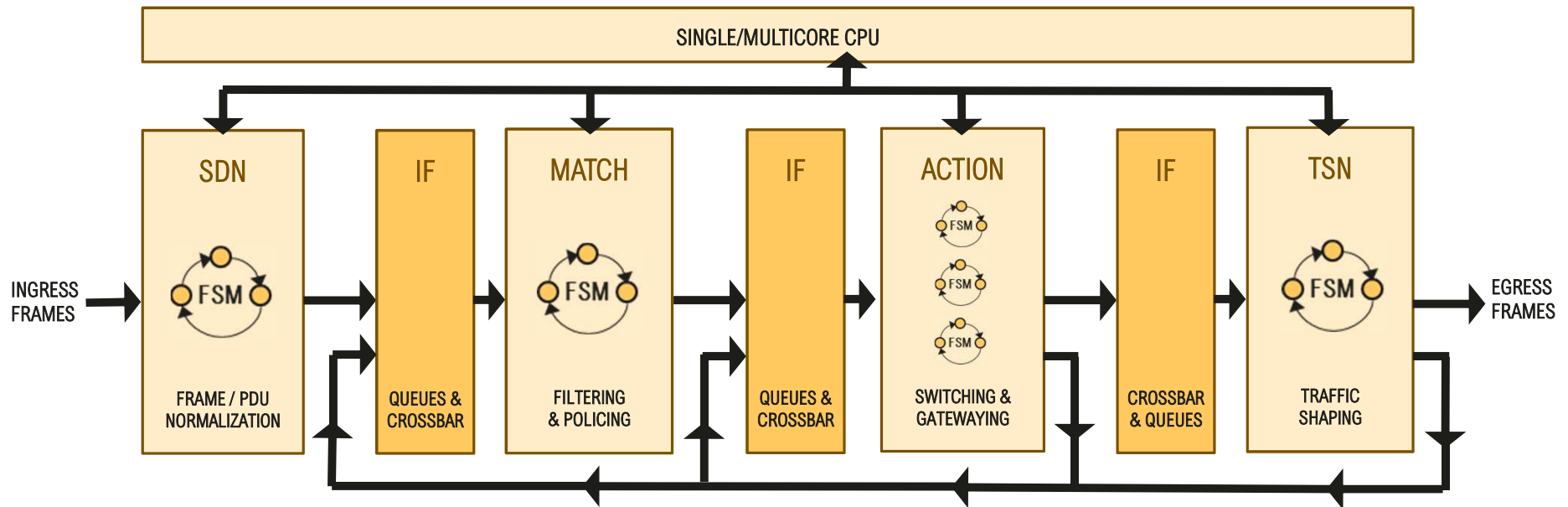
3.3 INNOVATION 05: SELF – Self-contained intermediate data allocated in the instruction

- Avoidance of buffering intermediate computation data (e.g. counters & flags) inside HWAs by inserting queues between stages and using the instruction itself as storage, reaching thus the **right decoupling between data and processing** (scaling solution)

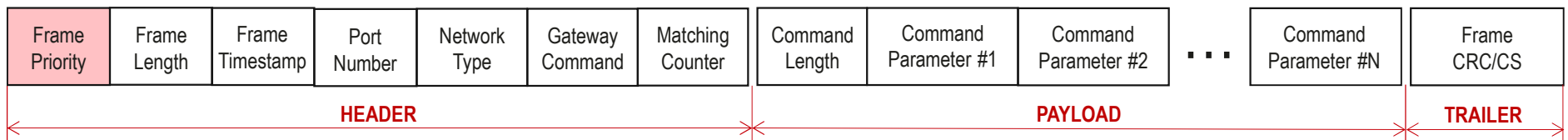


3.3 INNOVATION 05: SELF – Self-contained intermediate data allocated in the instruction

- Avoidance of buffering intermediate computation data (e.g. counters & flags) inside HWAs by inserting queues between stages and using the instruction itself as storage, reaching thus the **right decoupling between data and processing** (scaling solution)

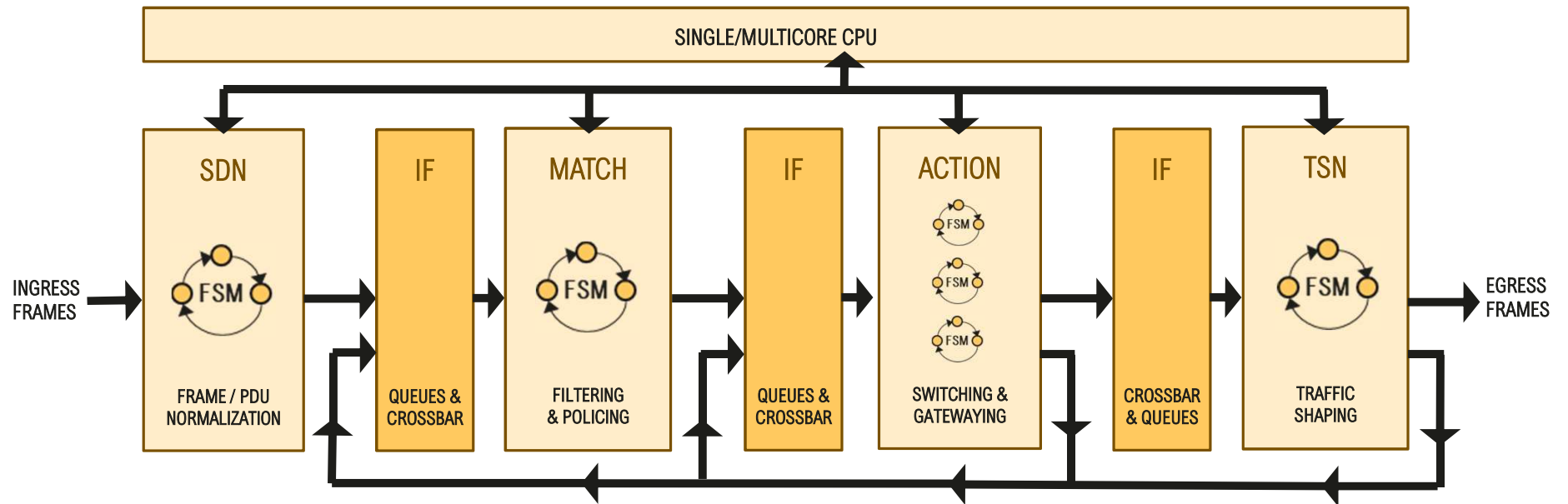


INSTRUCTION FRAME (NORMALIZATION):

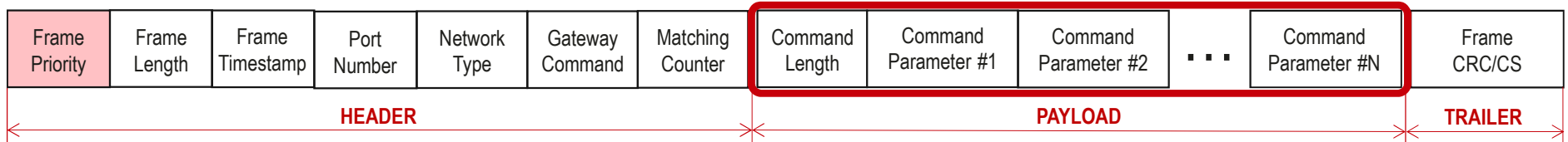


3.3 INNOVATION 05: SELF – Self-contained intermediate data allocated in the instruction

- Avoidance of buffering intermediate computation data (e.g. counters & flags) inside HWAs by inserting queues between stages and using the instruction itself as storage, reaching thus the **right decoupling between data and processing** (scaling solution)

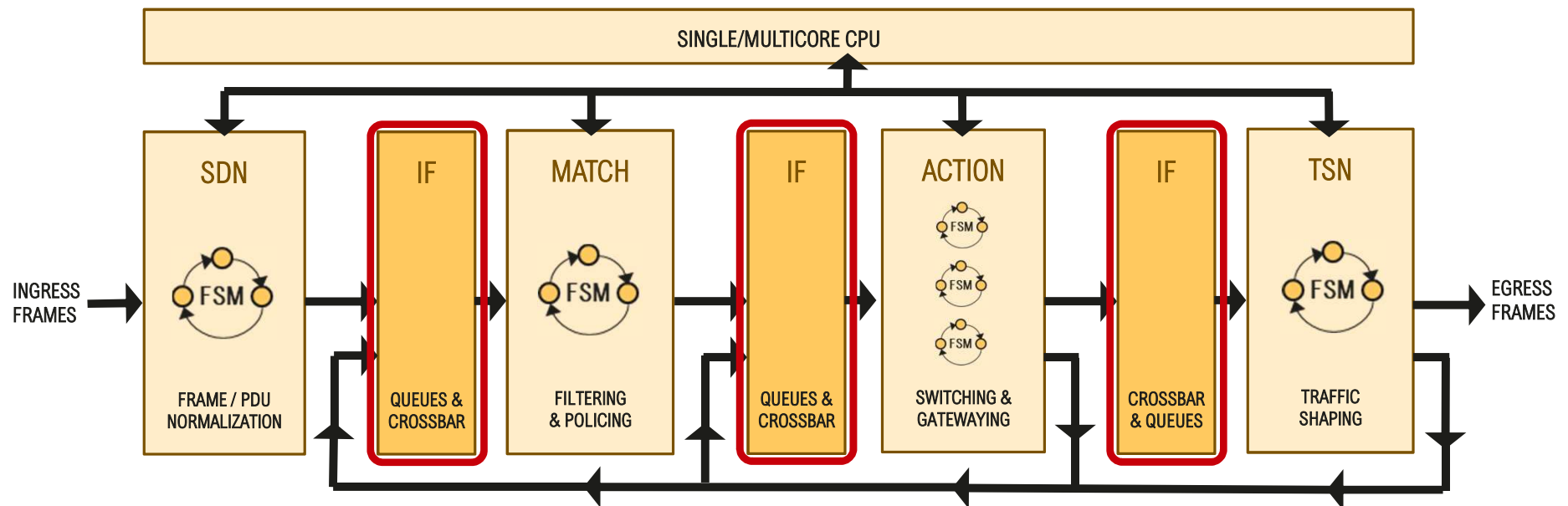


INSTRUCTION FRAME (NORMALIZATION):

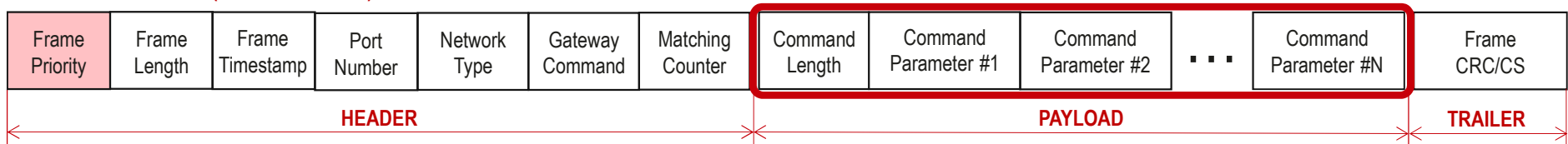


3.3 INNOVATION 05: SELF – Self-contained intermediate data allocated in the instruction

- Avoidance of buffering intermediate computation data (e.g. counters & flags) inside HWAs by inserting queues between stages and using the instruction itself as storage, reaching thus the **right decoupling between data and processing** (scaling solution)



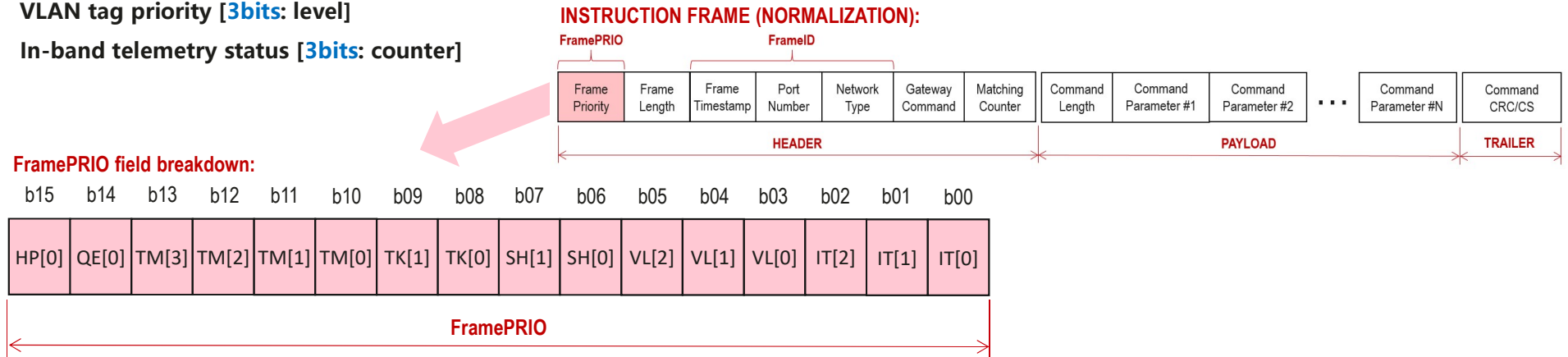
INSTRUCTION FRAME (NORMALIZATION):





3.3 INNOVATION 06: DAE – Distributed Arbitration Engine to handle priorities

- As arbitration strategy, the first and foremost relevant field of the instruction frame (**FramePRIO**) defines the priority (insight from CAN carrier sense multiple access / Collision Detection CSMA/CD). Many factors take part in the inline priority assignment per frame, for instance compacted in a **16-bits word**:
 - Highest priority [1bit: interrupt]
 - Queue status [1bit: (nearly) full, (nearly) empty]
 - Timeout/timestamp [4bits: time factor]
 - Tasks/HWAs status [2bits: free, free2taken, taken2free, taken]
 - Shaper status [2bits: free, free2use, use2free, in use]
 - VLAN tag priority [3bits: level]
 - In-band telemetry status [3bits: counter]

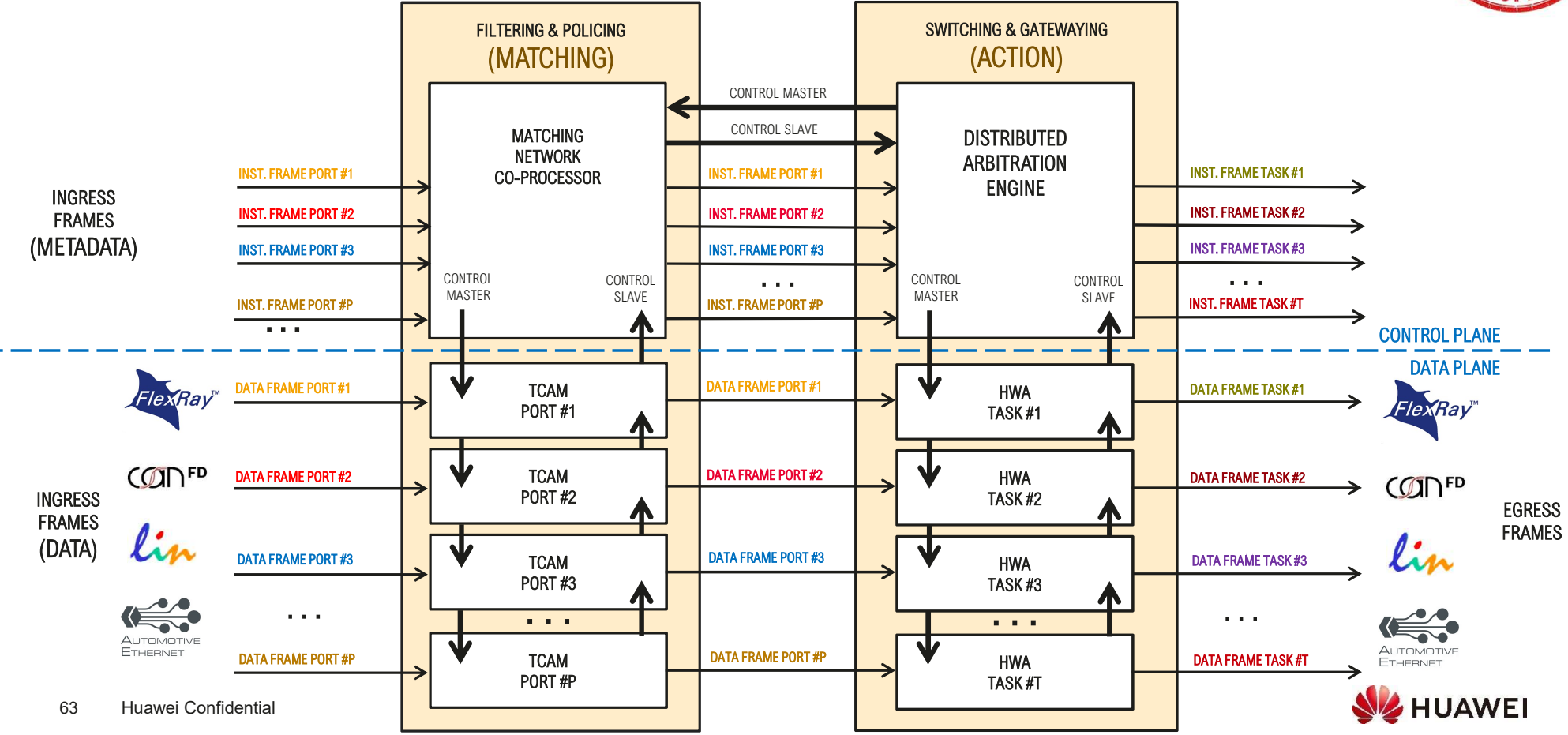


Rationale: The priority of each frame is **updated on the fly** in order to adapt its priority to each moment (**time-dependent solution**, i.e, taking into account timing factors like expiration timeouts or DDS timing requirements like latency budget, delivery time, etc.)



3.3 INNOVATION 06: DAE – Distributed Arbitration Engine to handle priorities

- Simplified MICROARCHITECTURE of the **MATCHING & ACTION** processing



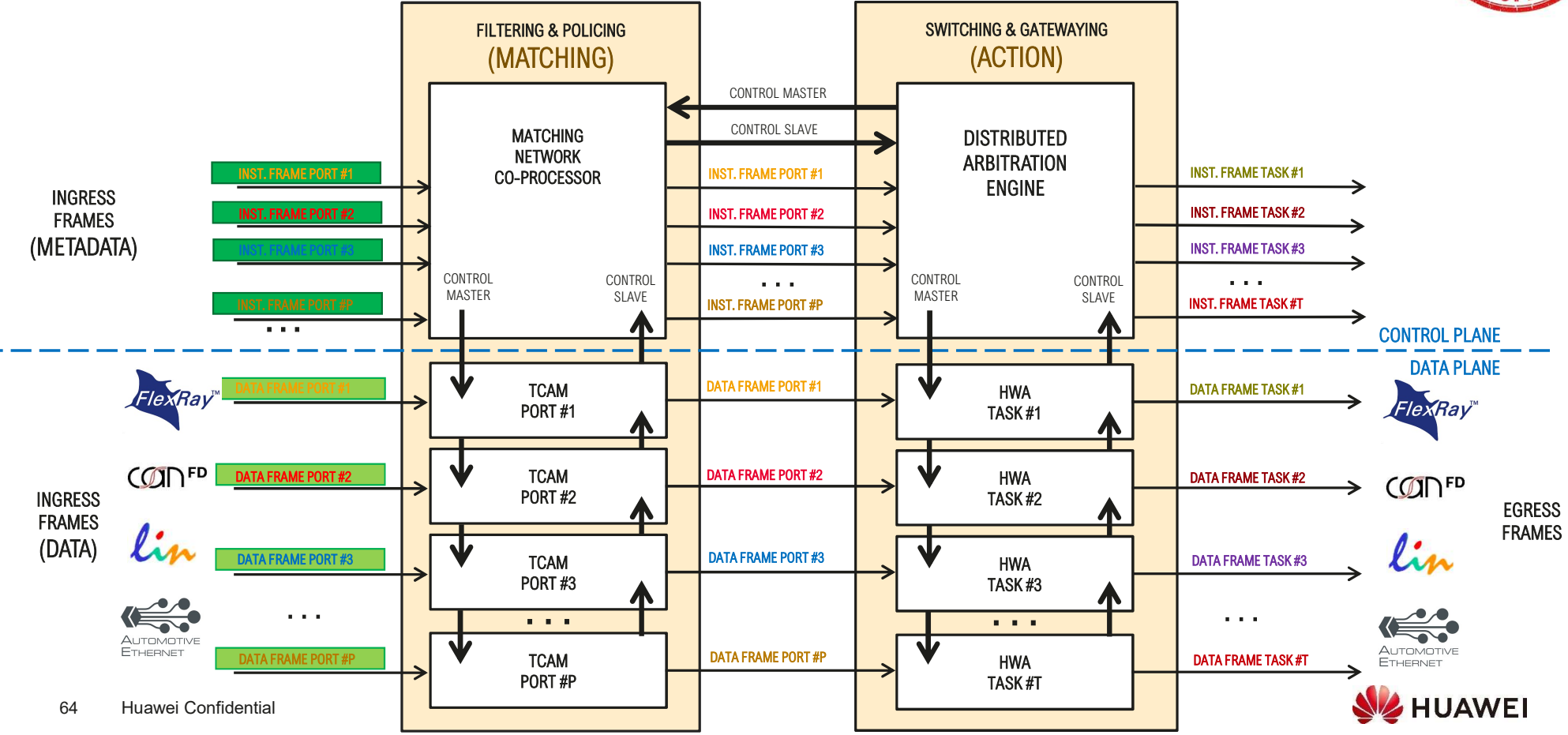
HWA = Hardware Accelerator





3.3 INNOVATION 06: DAE – Distributed Arbitration Engine to handle priorities

- Simplified MICROARCHITECTURE of the **MATCHING & ACTION** processing



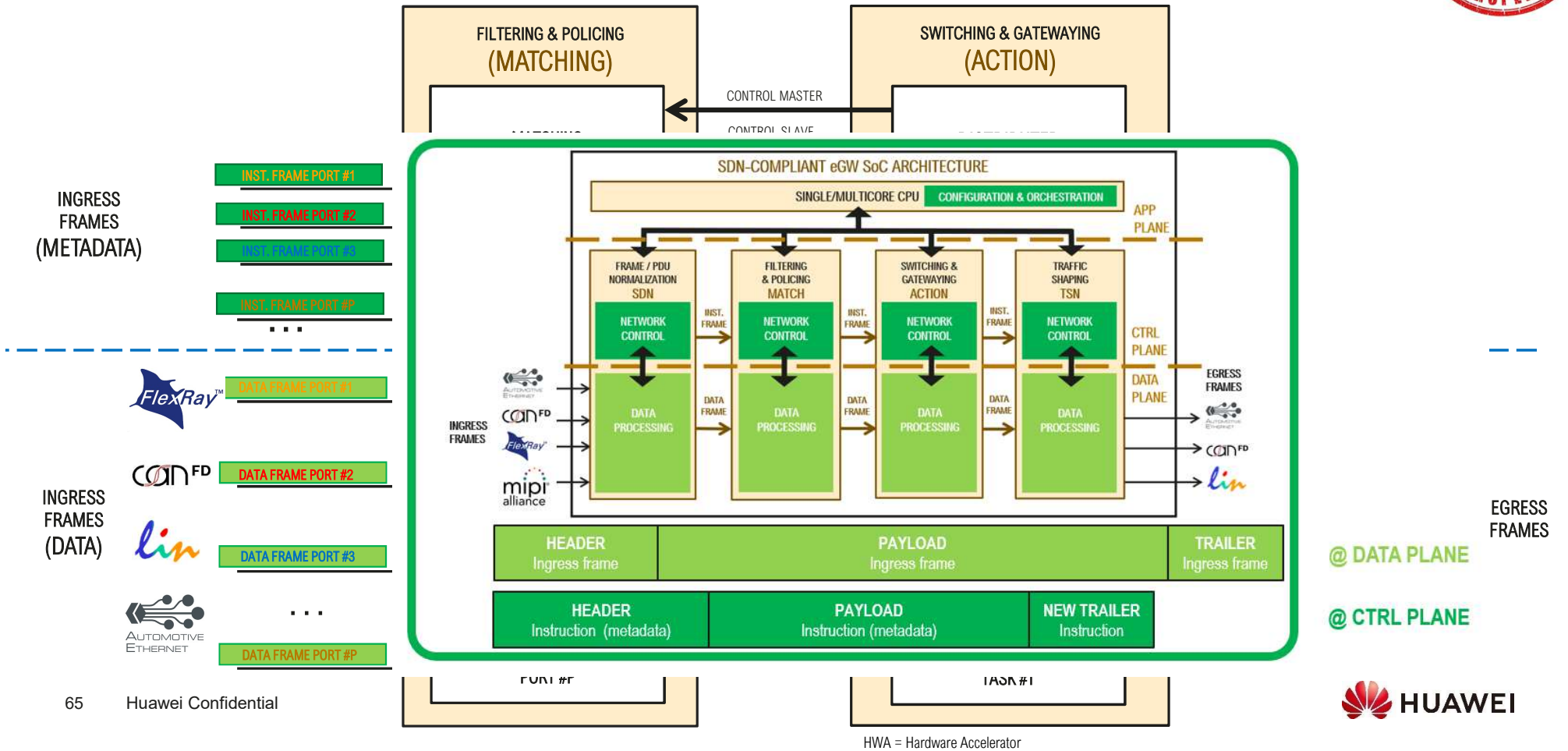
HWA = Hardware Accelerator





3.3 INNOVATION 06: DAE – Distributed Arbitration Engine to handle priorities

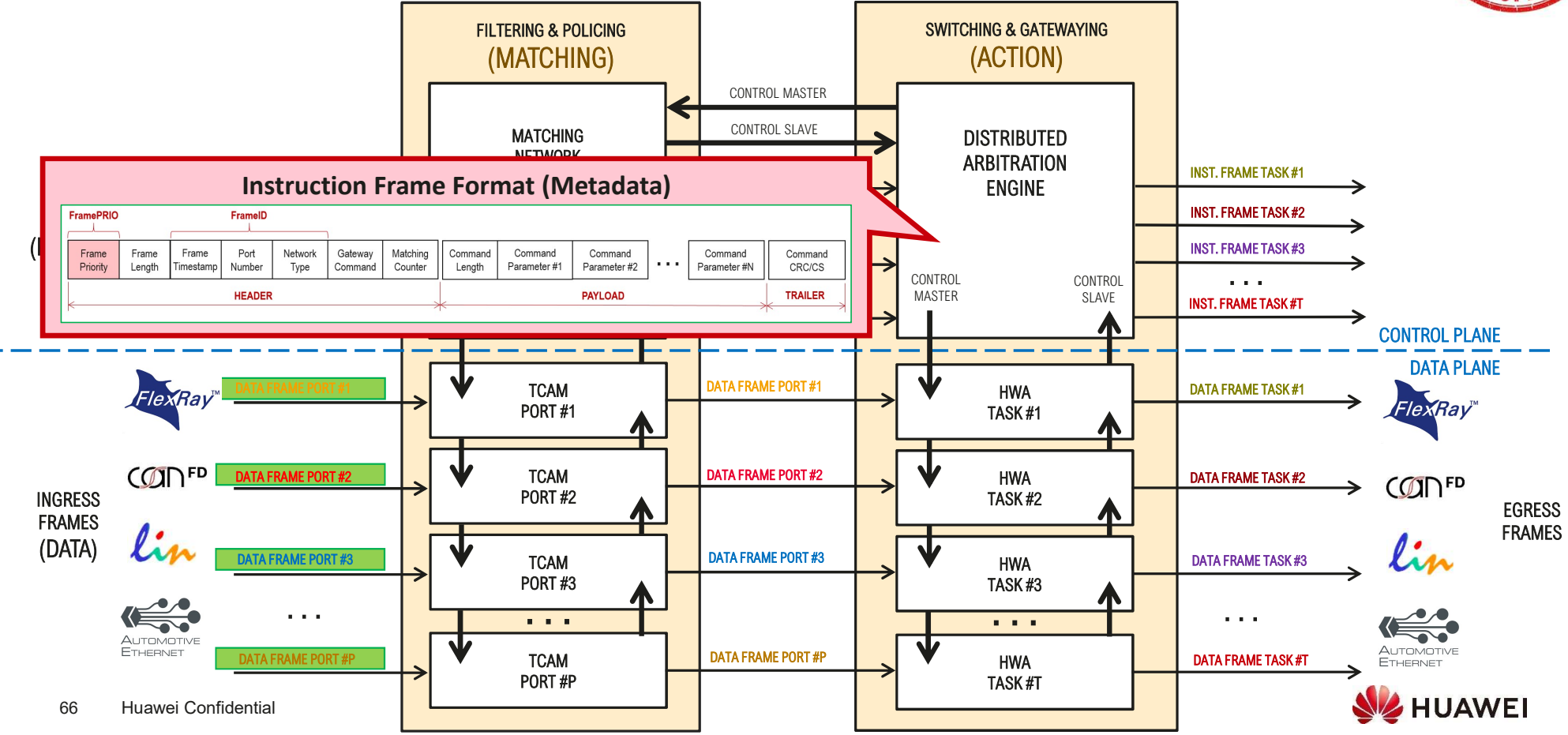
- Simplified MICROARCHITECTURE of the **MATCHING & ACTION** processing





3.3 INNOVATION 06: DAE – Distributed Arbitration Engine to handle priorities

- Simplified MICROARCHITECTURE of the MATCHING & ACTION processing



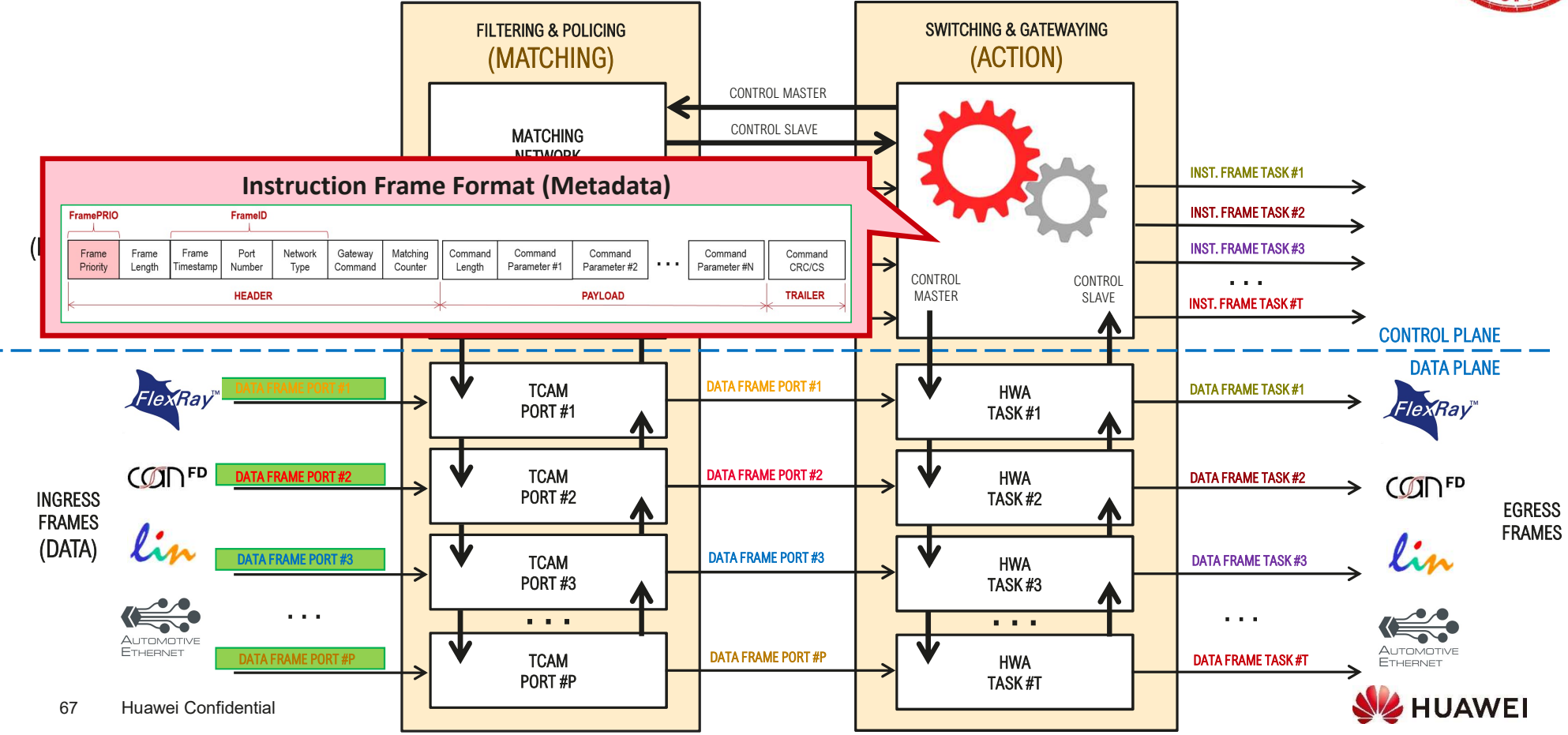
HWA = Hardware Accelerator





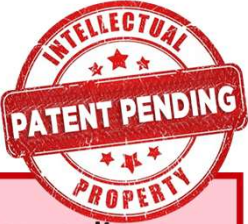
3.3 INNOVATION 06: DAE – Distributed Arbitration Engine to handle priorities

- Simplified MICROARCHITECTURE of the **MATCHING & ACTION** processing



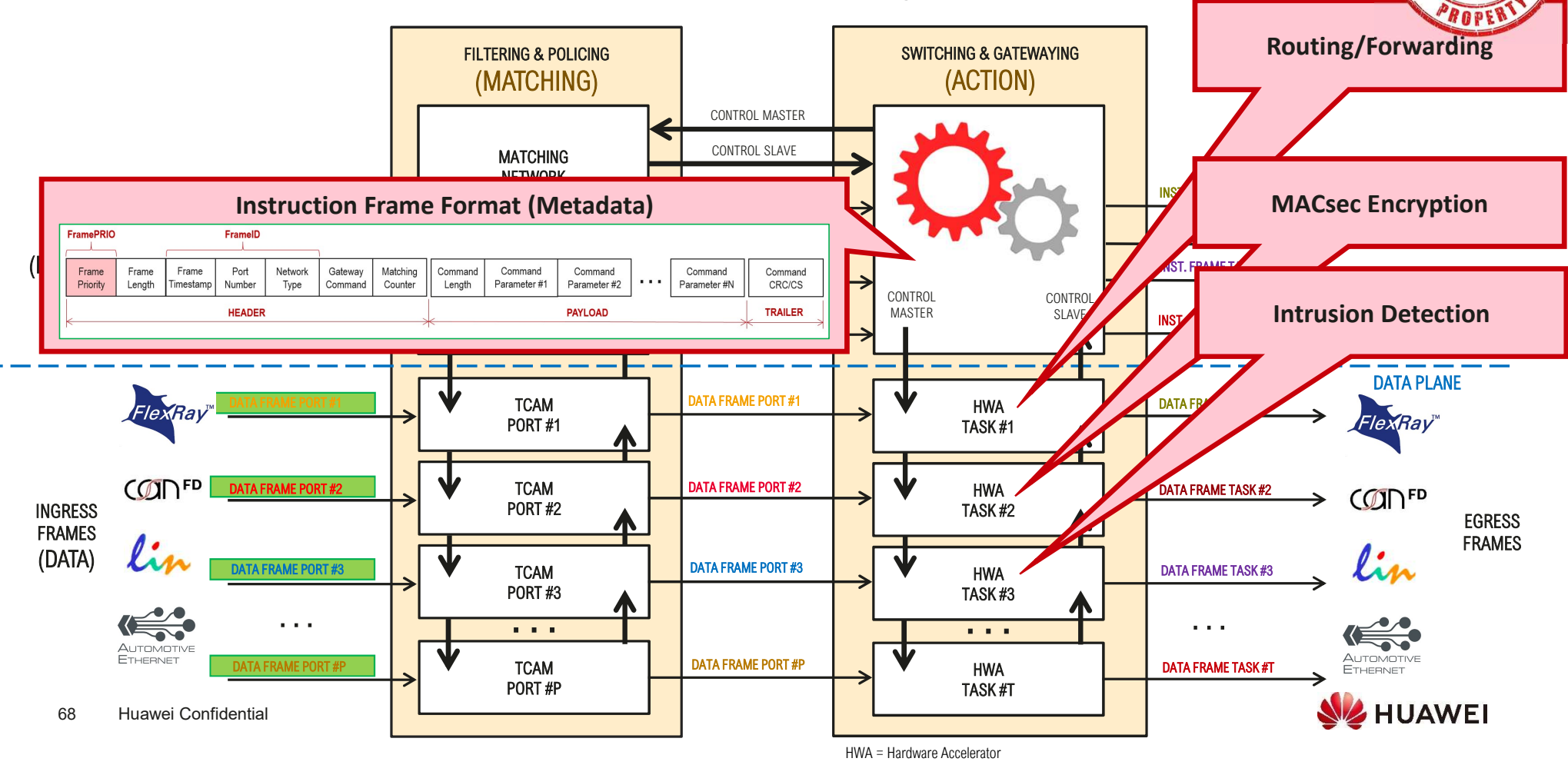
HWA = Hardware Accelerator

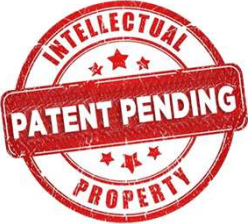




3.3 INNOVATION 06: DAE – Distributed Arbitration Engine to handle priorities

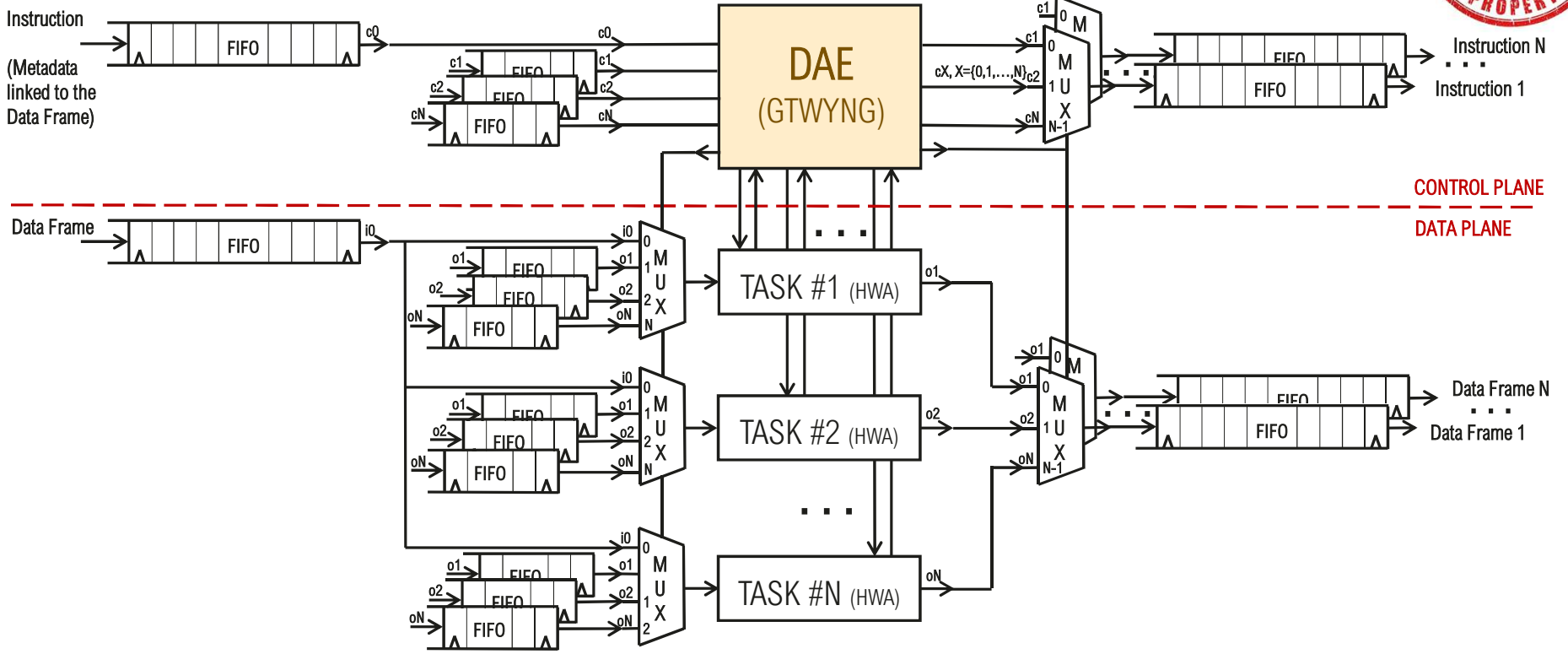
- Simplified MICROARCHITECTURE of the MATCHING & ACTION processing





3.3 INNOVATION 06: DAE – Distributed Arbitration Engine to handle priorities

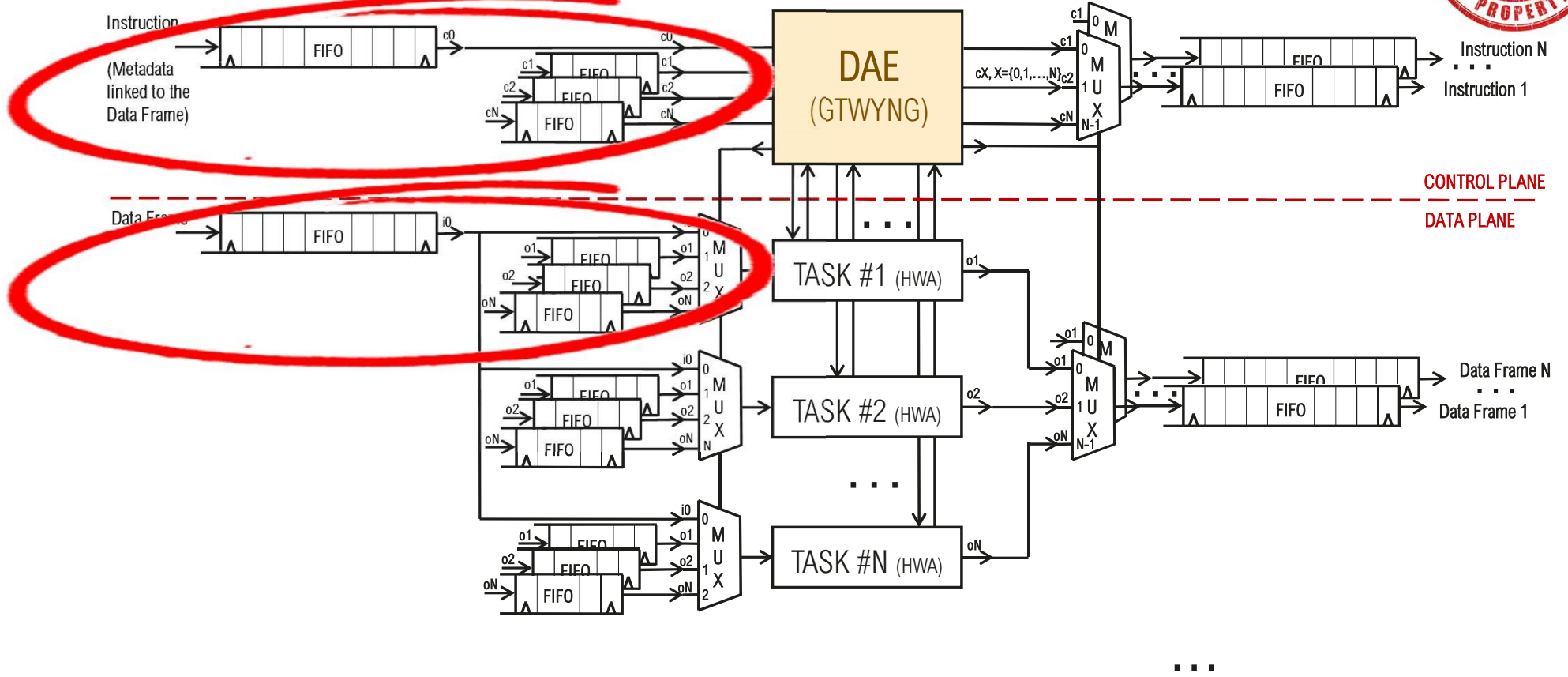
- Simplified MICROARCHITECTURE of the DAE processing (quick sorting in one clock cycle)





3.3 INNOVATION 06 : DAE – Distributed Arbitration Engine to handle priorities

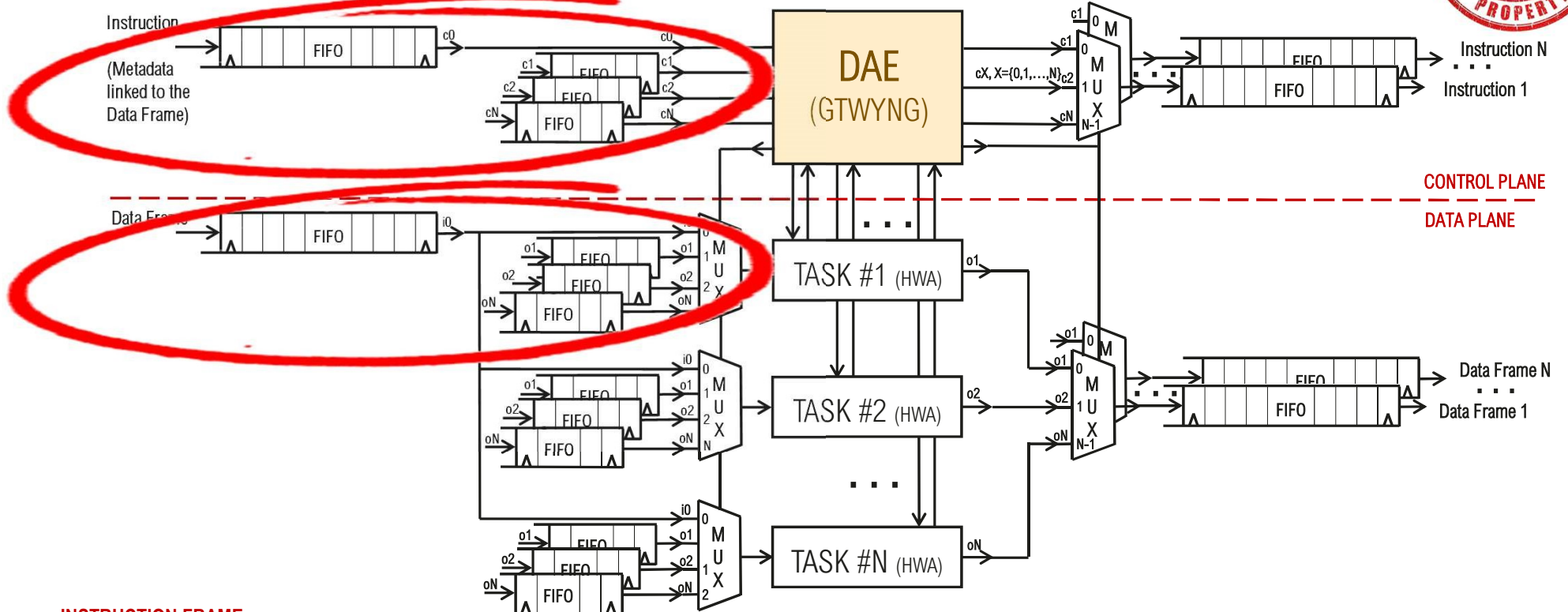
- Simplified MICROARCHITECTURE of the DAE processing (quick sorting in one clock cycle)



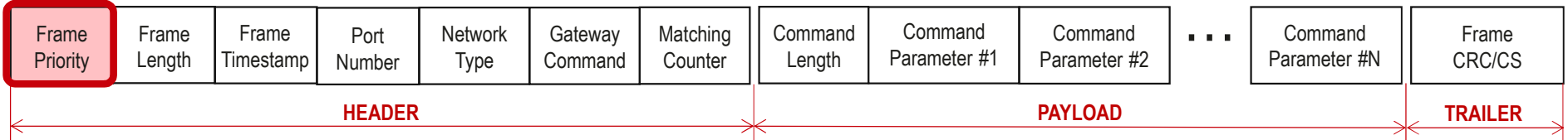


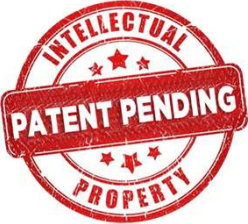
3.3 INNOVATION 06 : DAE – Distributed Arbitration Engine to handle priorities

- Simplified MICROARCHITECTURE of the DAE processing (quick sorting in one clock cycle)



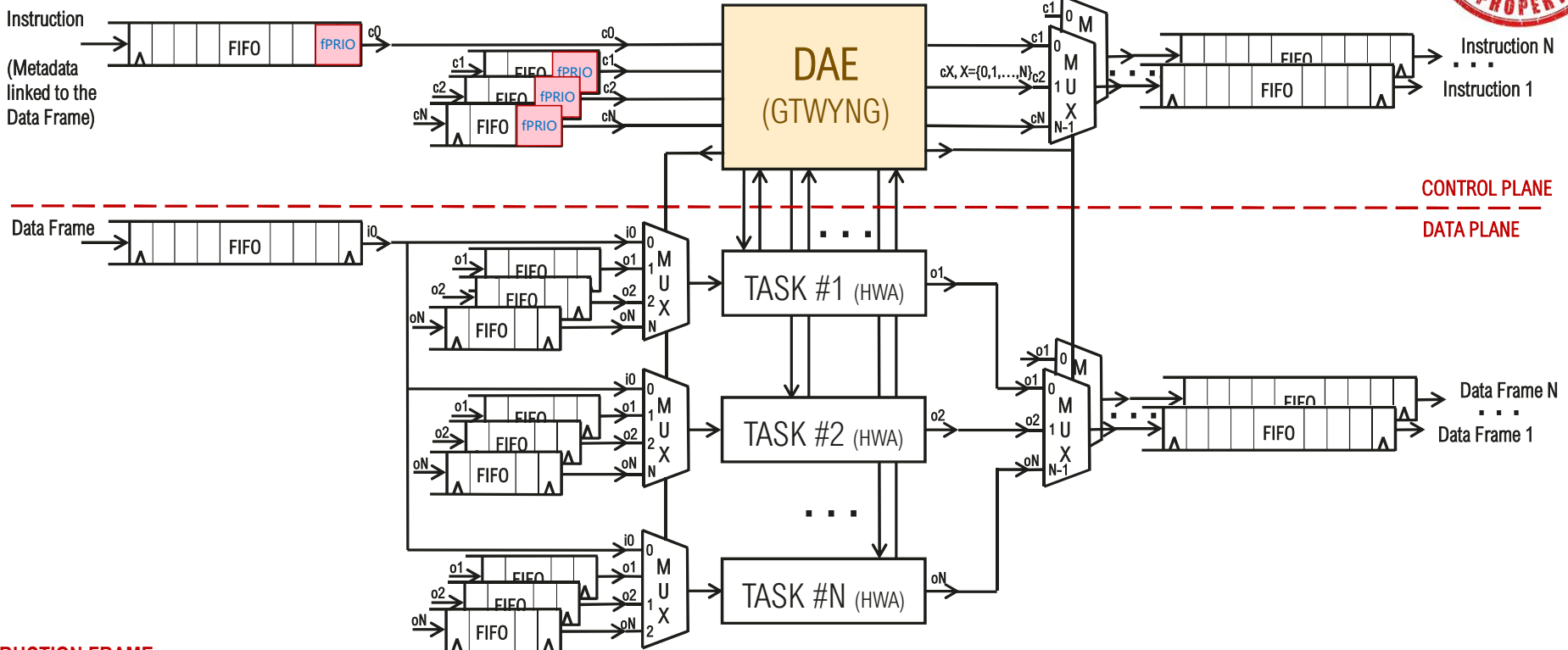
INSTRUCTION FRAME:



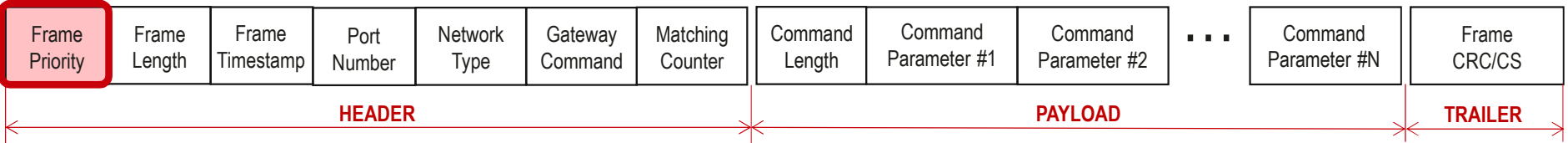


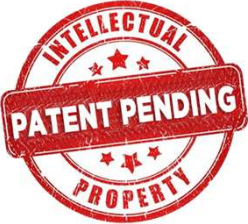
3.3 INNOVATION 06: DAE – Distributed Arbitration Engine to handle priorities

- Simplified MICROARCHITECTURE of the DAE processing (quick sorting in one clock cycle)



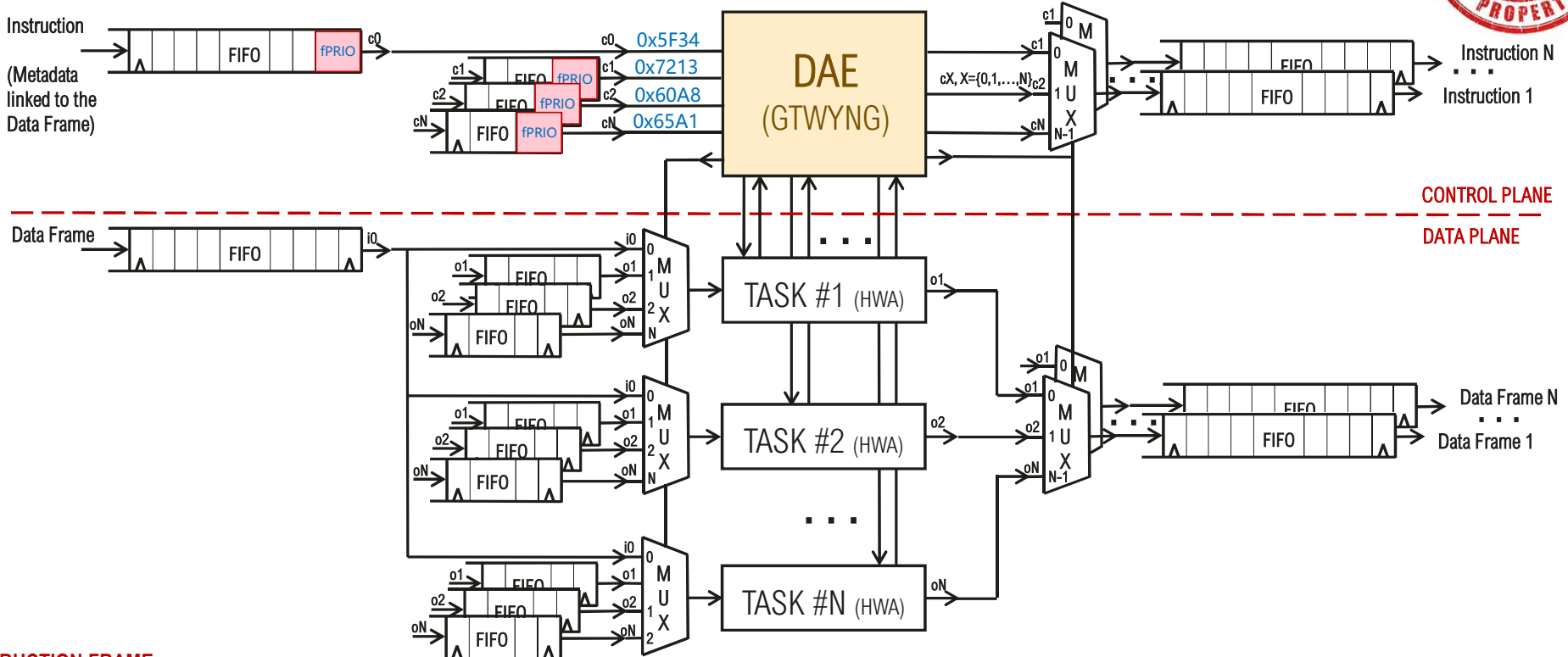
INSTRUCTION FRAME:



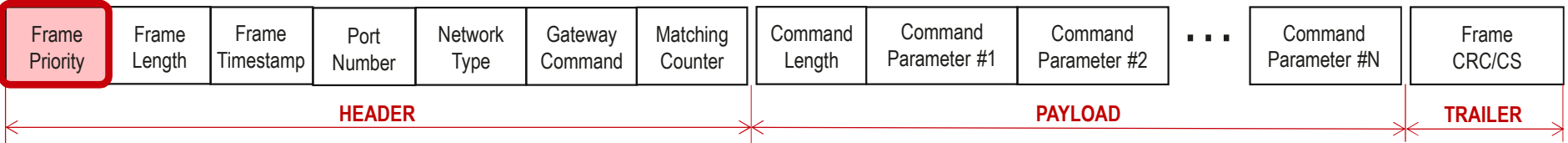


3.3 INNOVATION 06: DAE – Distributed Arbitration Engine to handle priorities

- Simplified MICROARCHITECTURE of the DAE processing (quick sorting in one clock cycle)



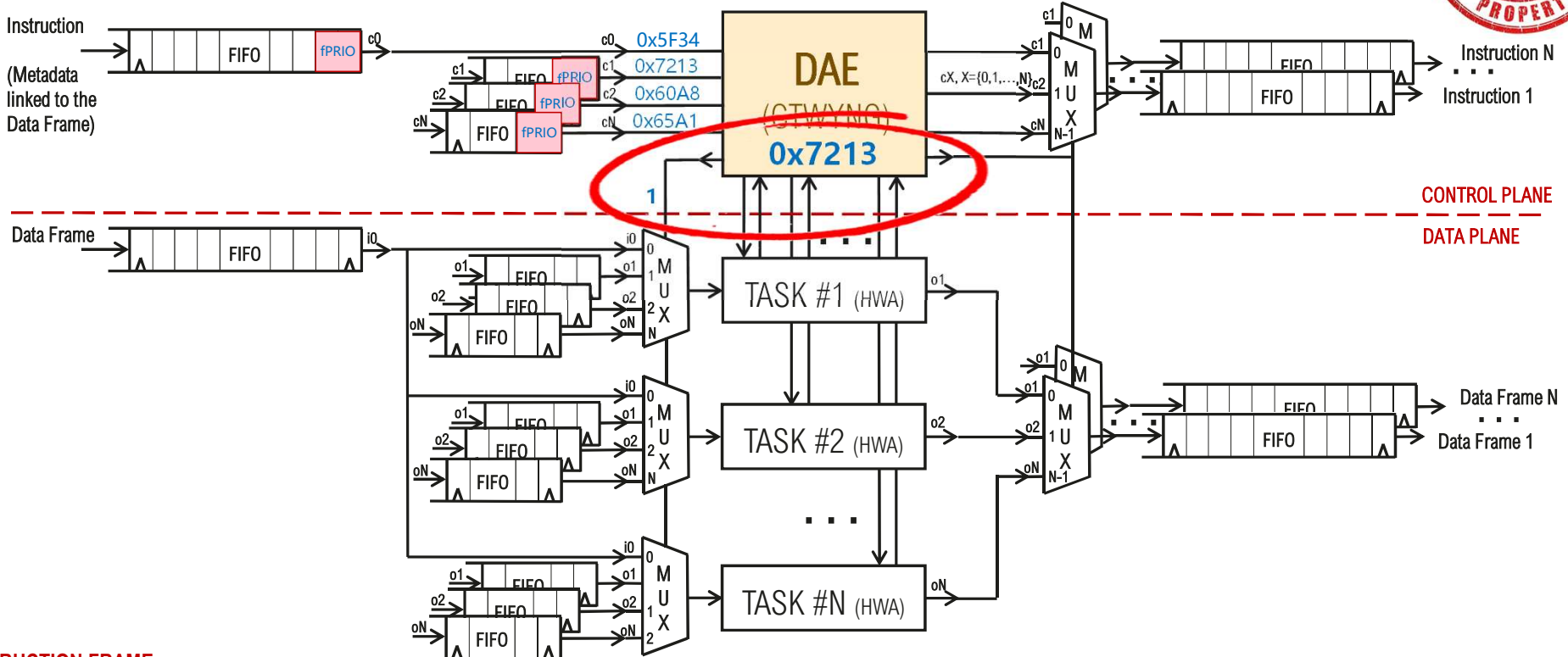
INSTRUCTION FRAME:



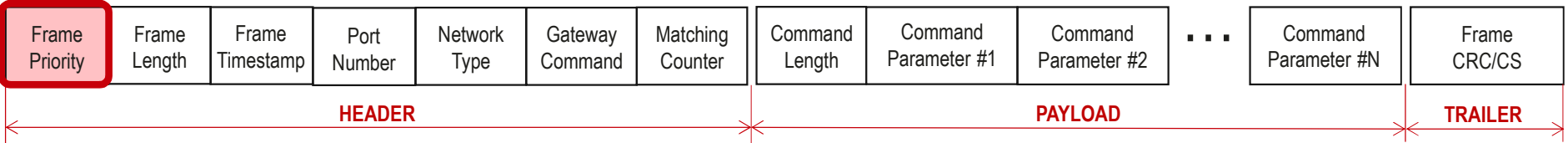


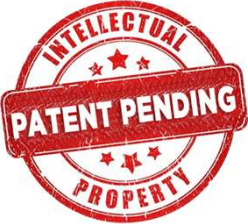
3.3 INNOVATION 06: DAE – Distributed Arbitration Engine to handle priorities

- Simplified MICROARCHITECTURE of the DAE processing (quick sorting in one clock cycle)



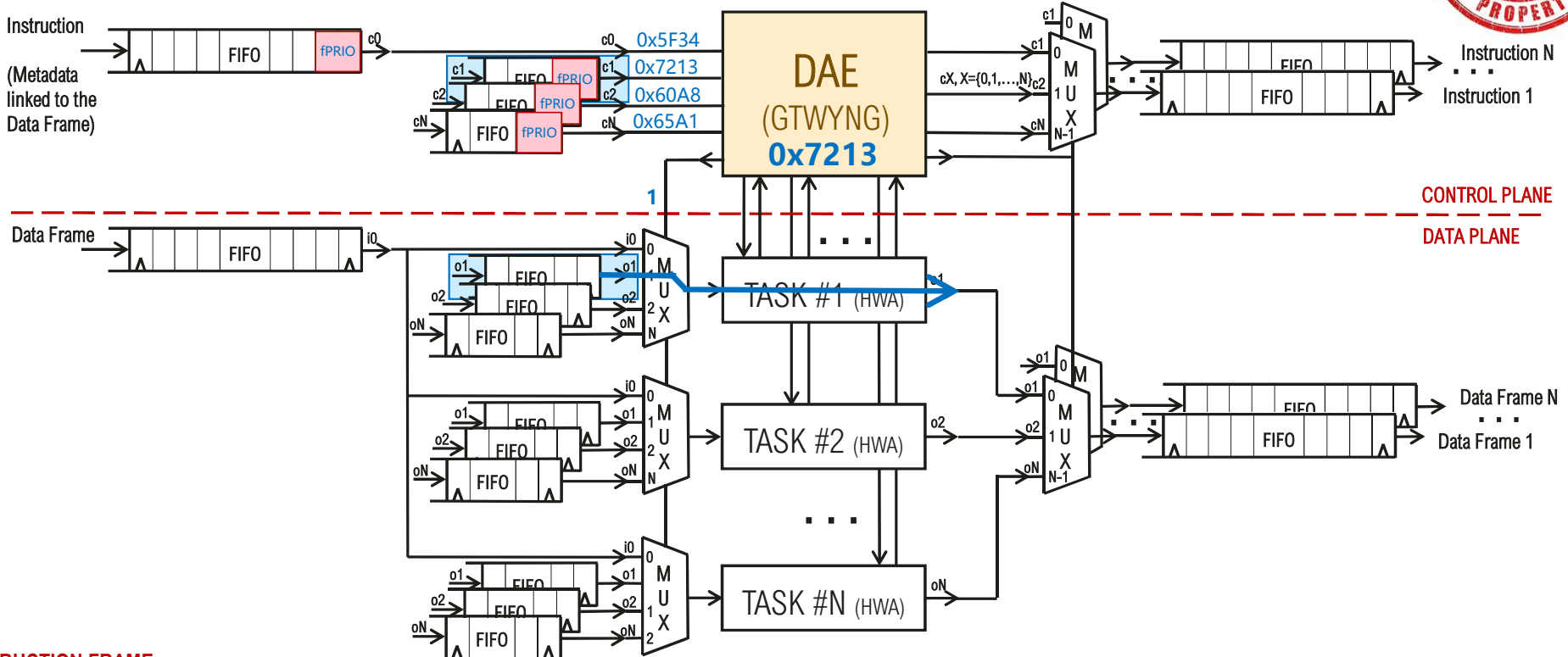
INSTRUCTION FRAME:



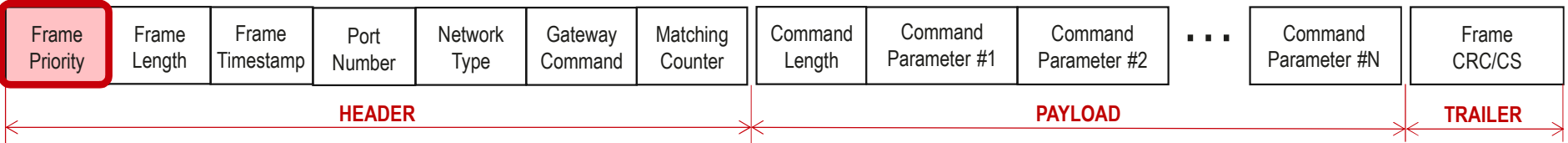


3.3 INNOVATION 06: DAE – Distributed Arbitration Engine to handle priorities

- Simplified MICROARCHITECTURE of the DAE processing (quick sorting in one clock cycle)



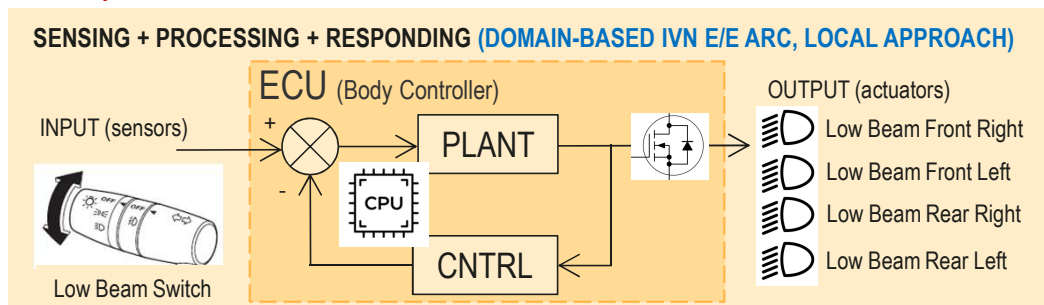
INSTRUCTION FRAME:



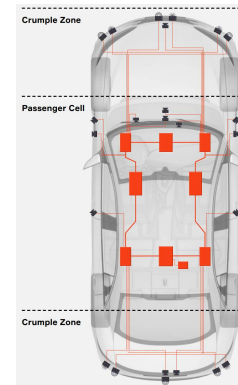
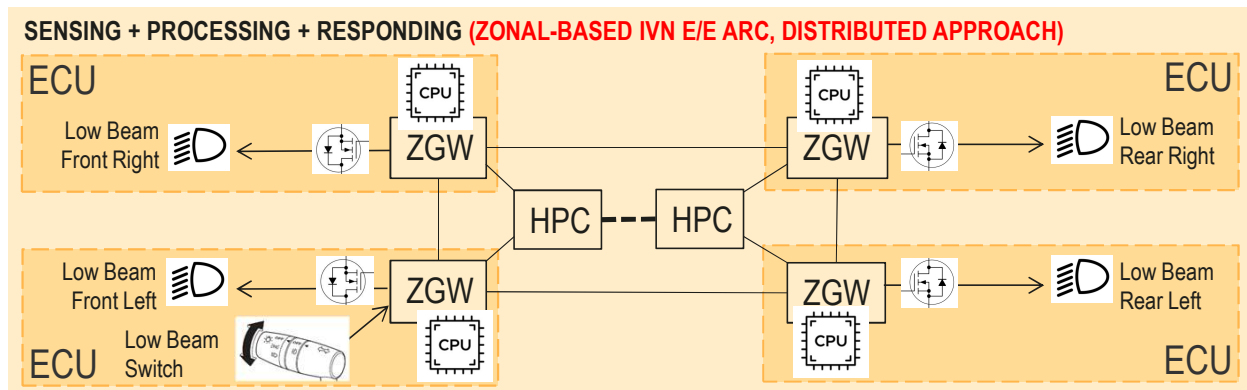
3.3 INNOVATION 07 : TSN – Time Sensitive Networking

- Many of the TSN standards can be performed directly in HW through dedicated engines, not only traffic shaping (e.g. TAS, CBS, preemption, etc.) but also other features like 802.1CB or 802.1AS

Example: Servo system



RENAULT MEGANE

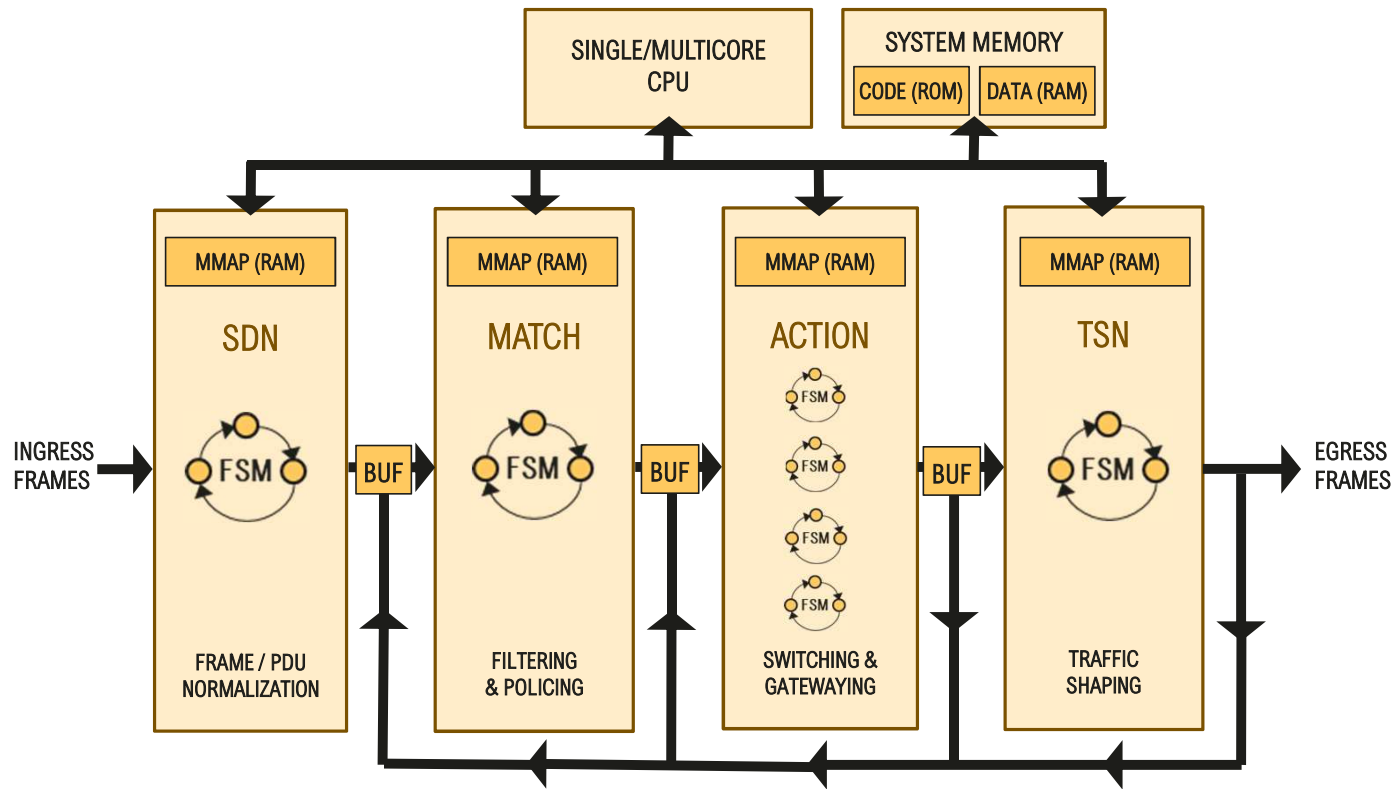


All the ECUs in the IVN need to share a common notion of time (gPTP, TSN 802.1AS)

Rationale: Time determinism guaranteed by design in hardware

3.3 INNOVATION 07 : TSN – Time Sensitive Networking

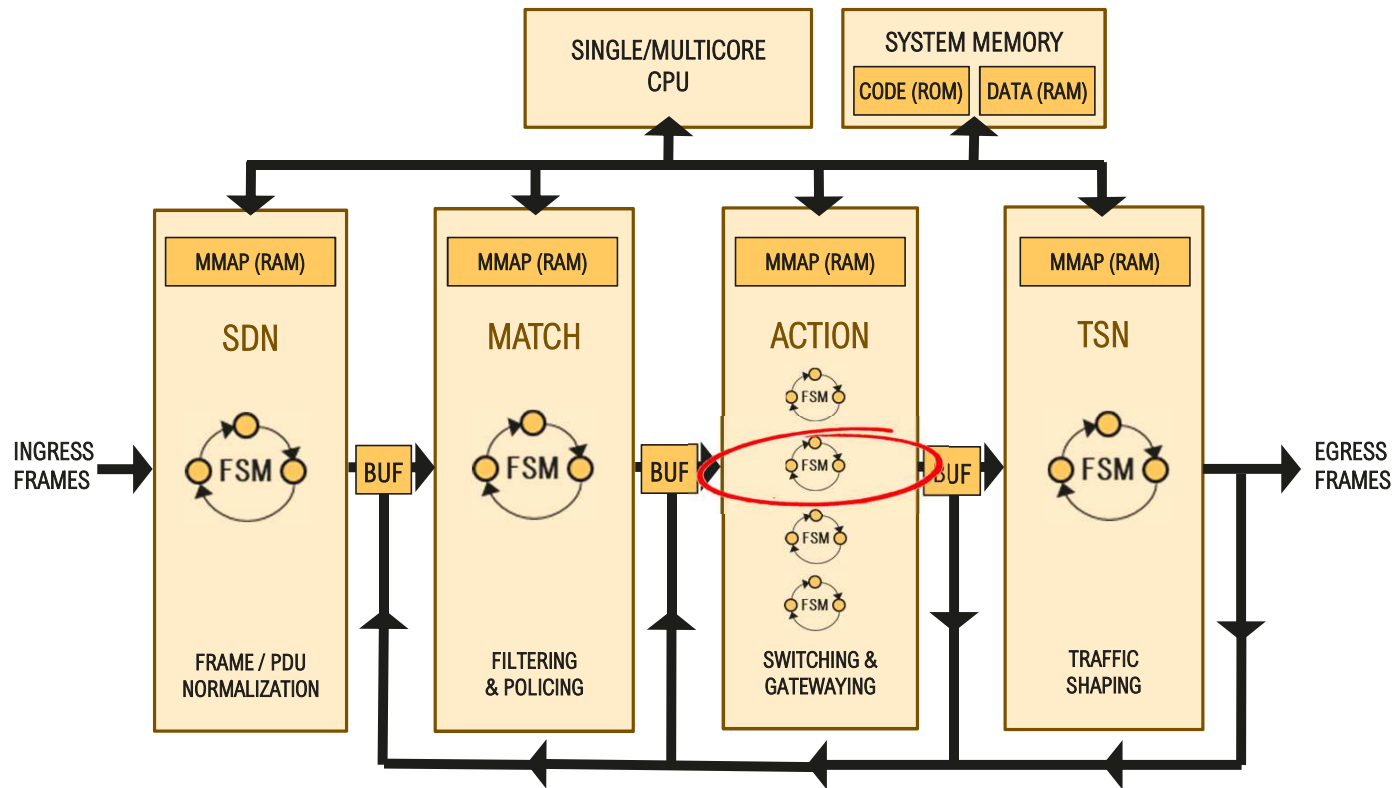
- Many of the TSN standards can be performed directly in HW through dedicated engines, not only traffic shaping (e.g. TAS, CBS, preemption, etc.) but also other features like 802.1CB or 802.1AS



Rationale: Time determinism guaranteed by design in hardware

3.3 INNOVATION 07 : TSN – Time Sensitive Networking

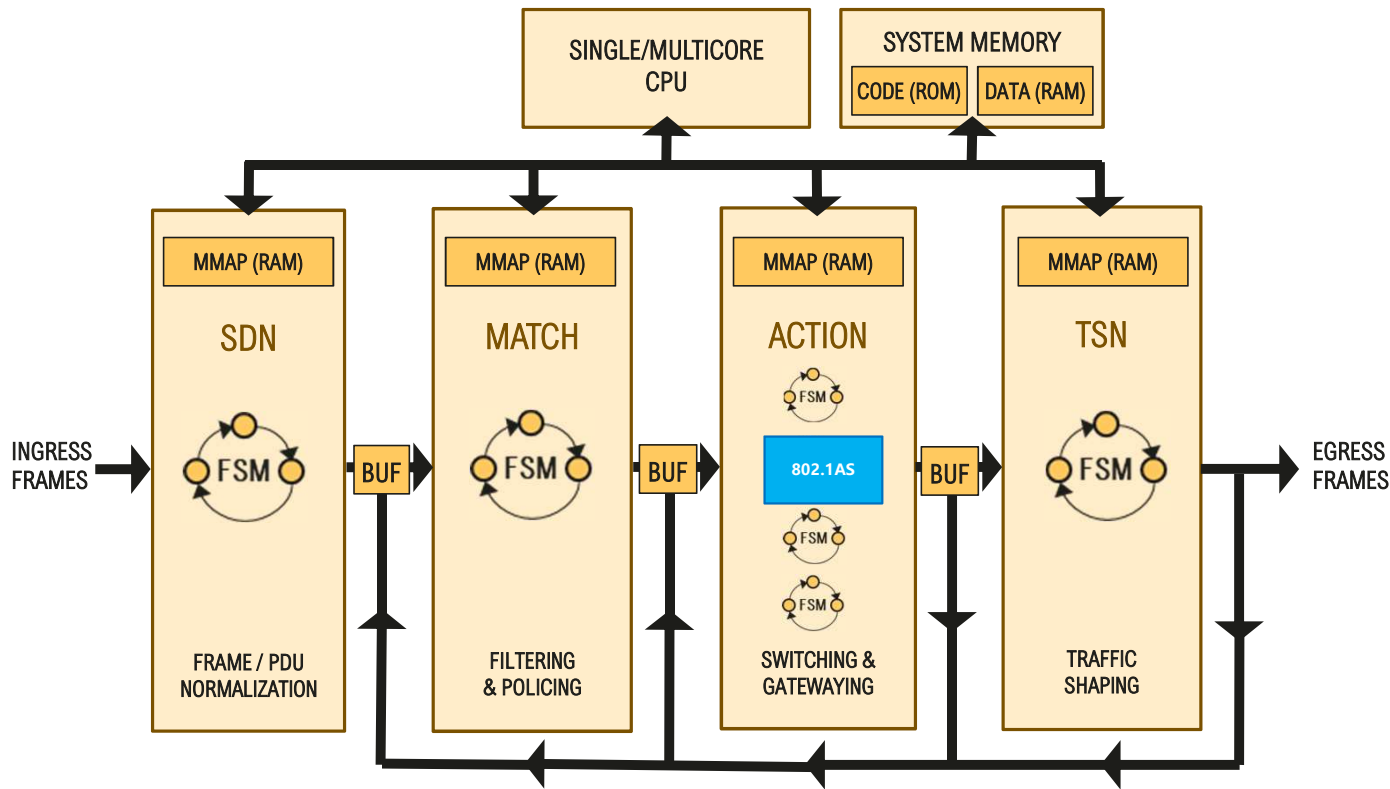
- Many of the TSN standards can be performed directly in HW through dedicated engines, not only traffic shaping (e.g. TAS, CBS, preemption, etc.) but also other features like 802.1CB or 802.1AS



Rationale: Time determinism guaranteed by design in hardware

3.3 INNOVATION 07 : TSN – Time Sensitive Networking

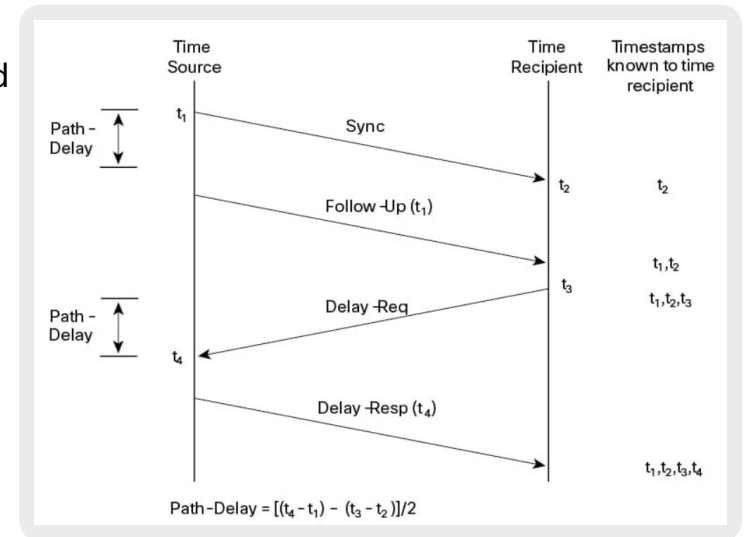
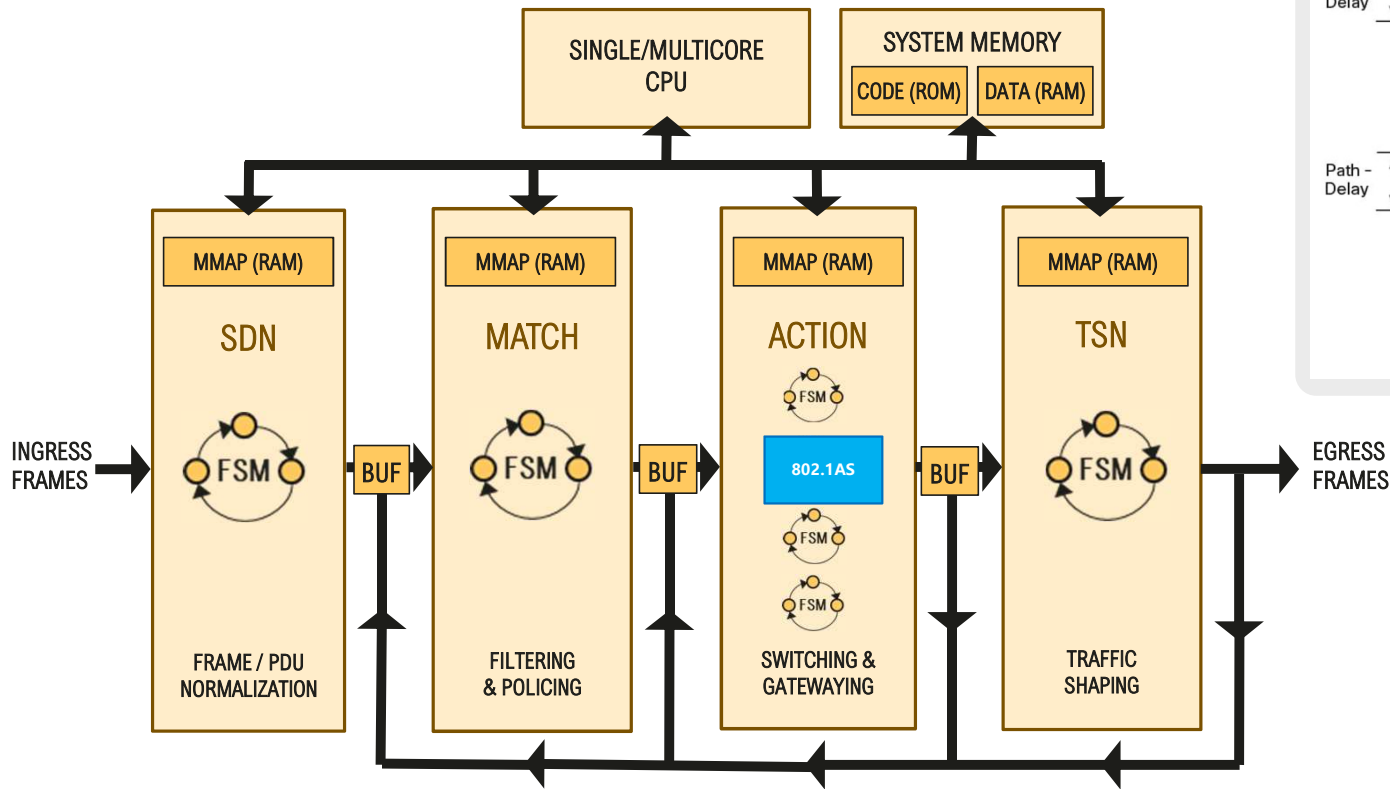
- Many of the TSN standards can be performed directly in HW through dedicated engines, not only traffic shaping (e.g. TAS, CBS, preemption, etc.) but also other features like 802.1CB or 802.1AS



Rationale: Time determinism guaranteed by design in hardware

3.3 INNOVATION 07 : TSN – Time Sensitive Networking

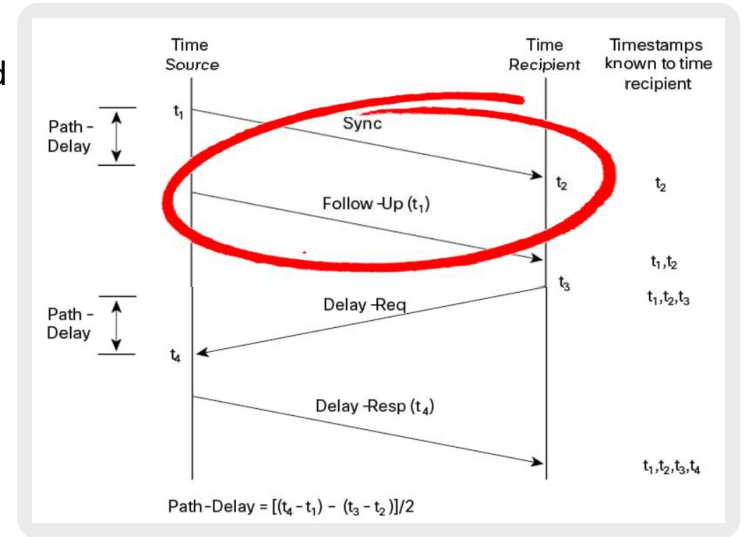
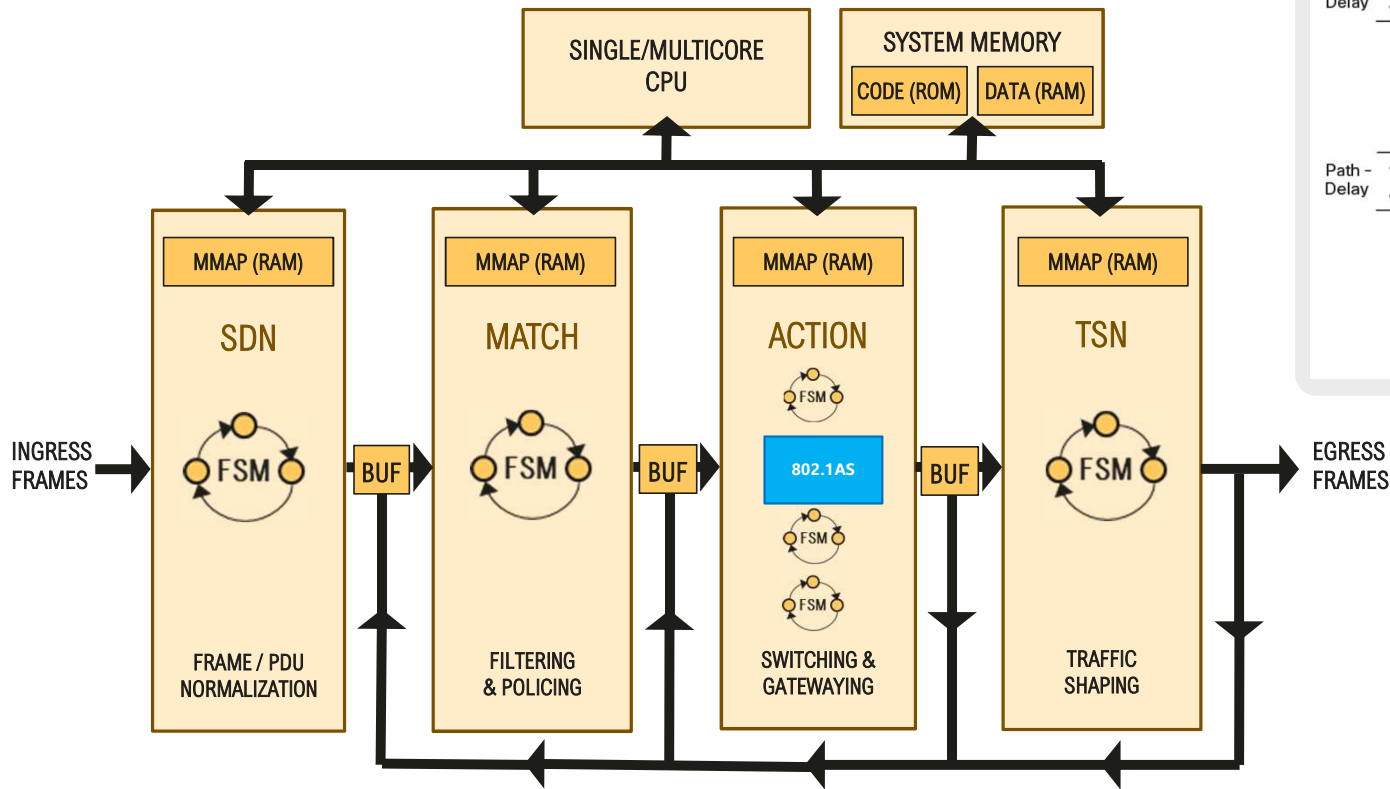
- Many of the TSN standards can be performed directly in HW through dedicated preemption, etc.) but also other features like 802.1CB or 802.1AS



Rationale: Time determinism guaranteed by design in hardware

3.3 INNOVATION 07 : TSN – Time Sensitive Networking

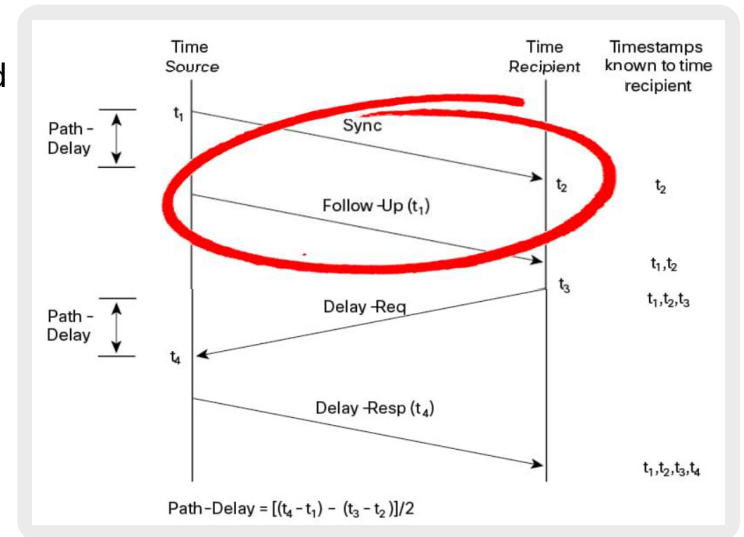
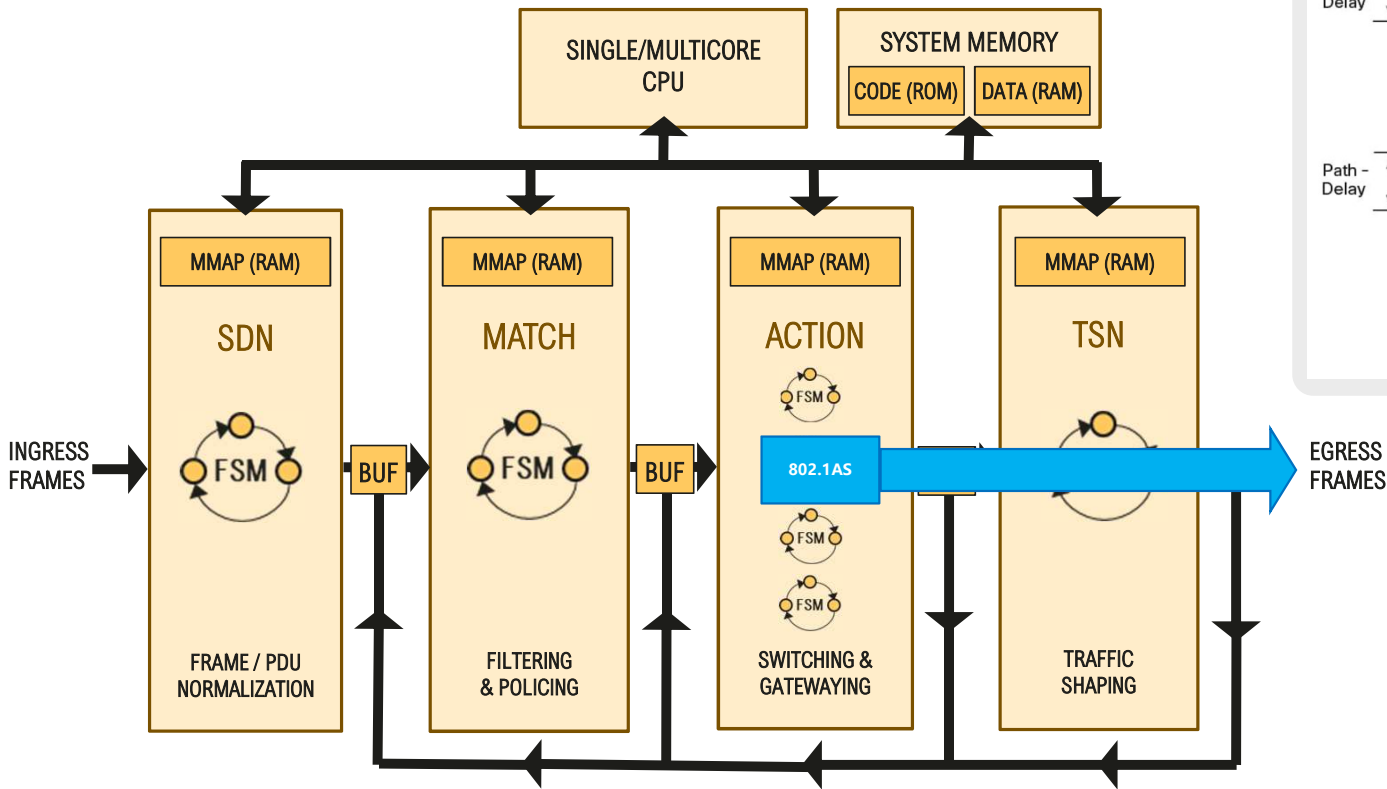
- Many of the TSN standards can be performed directly in HW through dedicated preemption, etc.) but also other features like 802.1CB or 802.1AS



Rationale: Time determinism guaranteed by design in hardware

3.3 INNOVATION 07 : TSN – Time Sensitive Networking

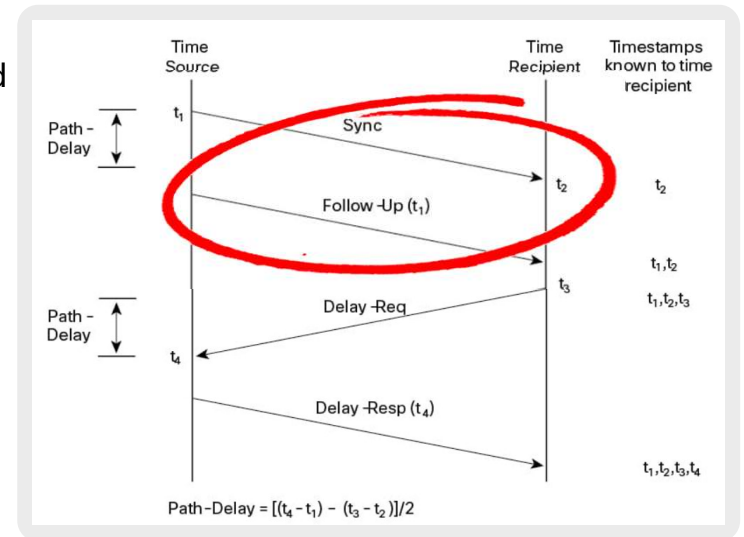
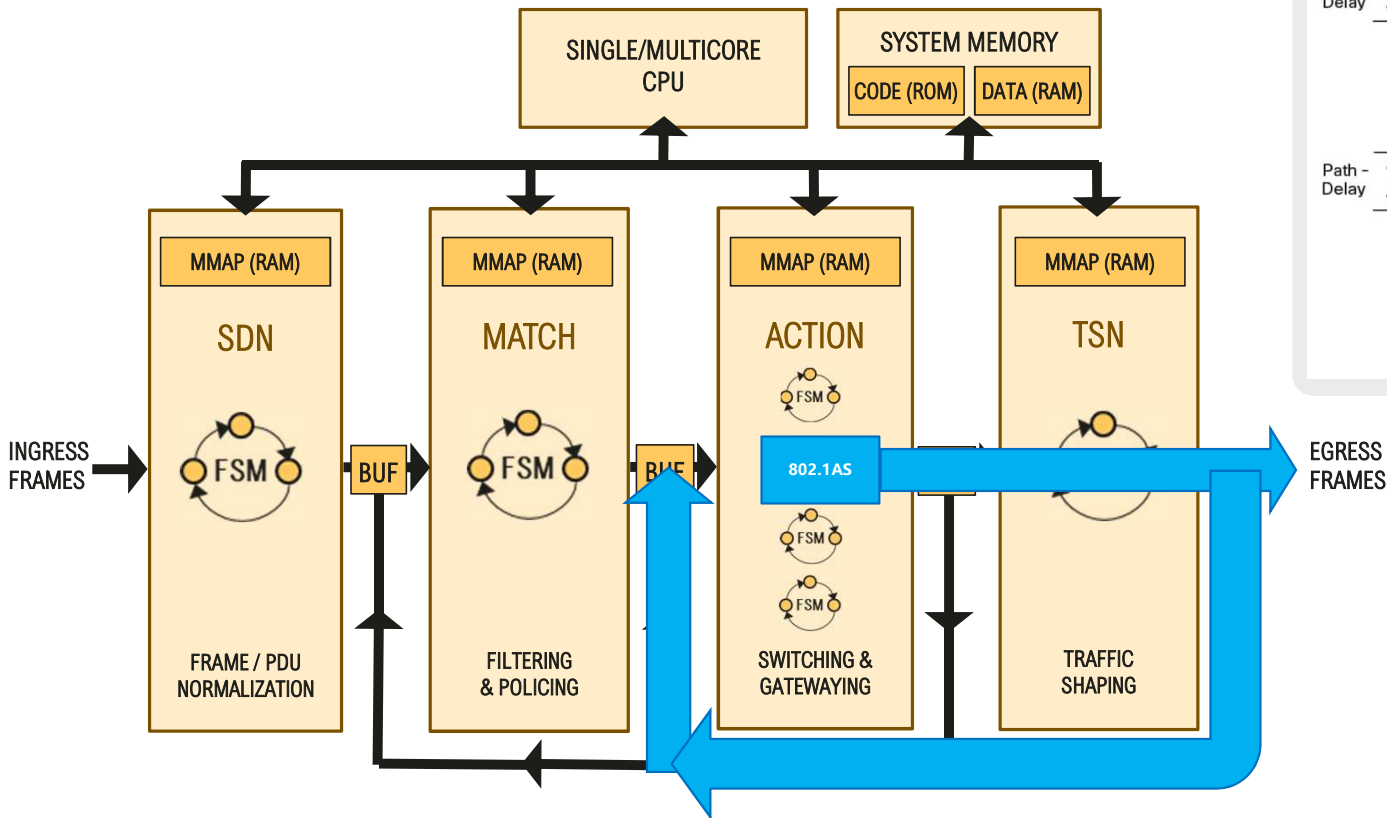
- Many of the TSN standards can be performed directly in HW through dedicated preemption, etc.) but also other features like 802.1CB or 802.1AS



Rationale: Time determinism guaranteed by design in hardware

3.3 INNOVATION 07 : TSN – Time Sensitive Networking

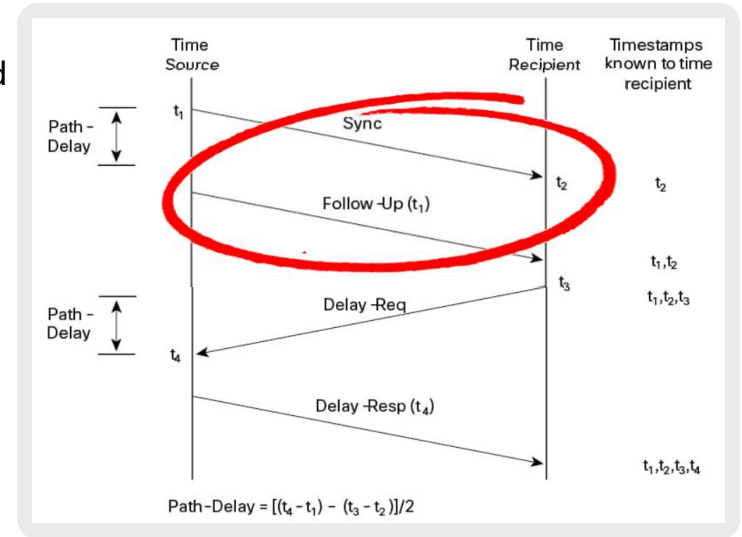
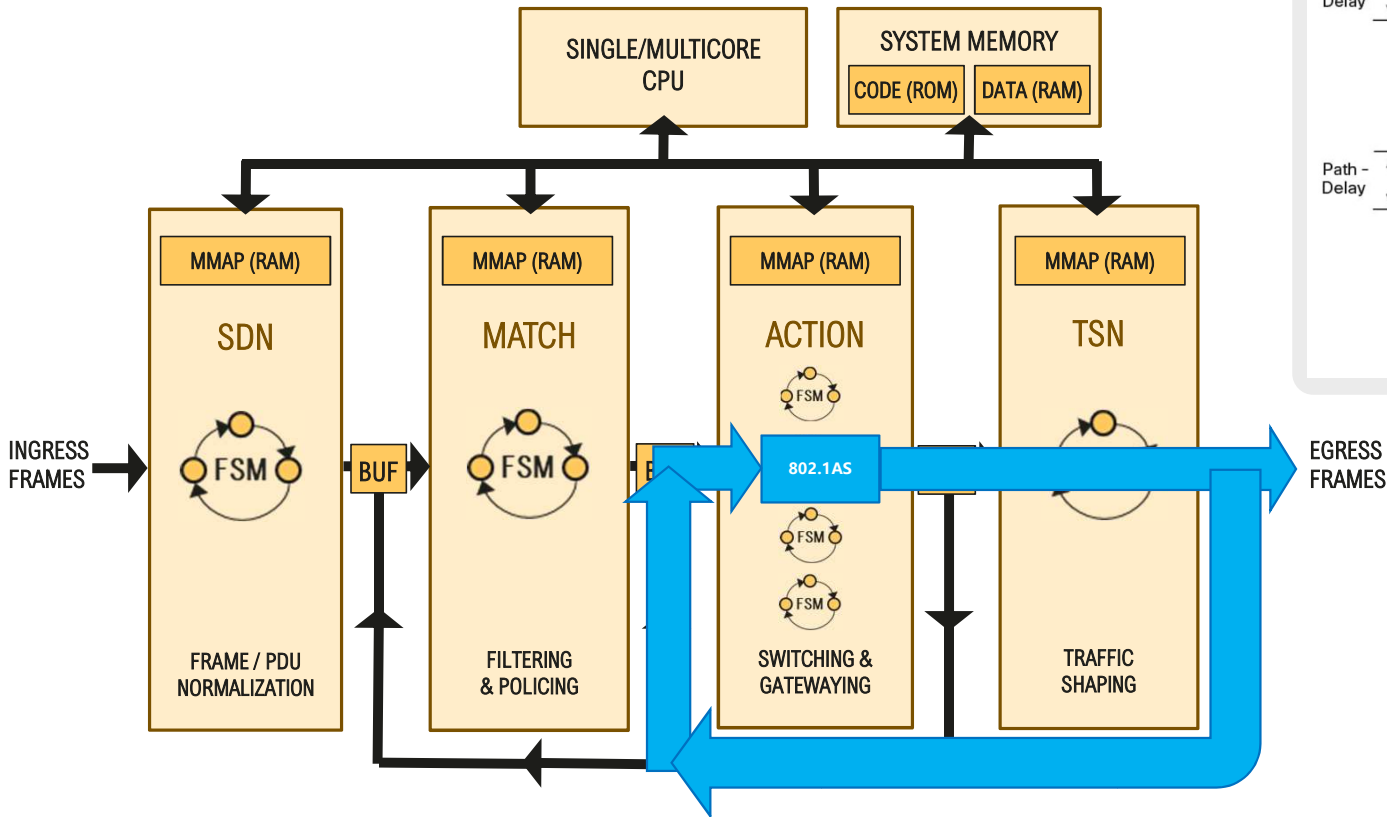
- Many of the TSN standards can be performed directly in HW through dedicated preemption, etc.) but also other features like 802.1CB or 802.1AS



Rationale: Time determinism guaranteed by design in hardware

3.3 INNOVATION 07 : TSN – Time Sensitive Networking

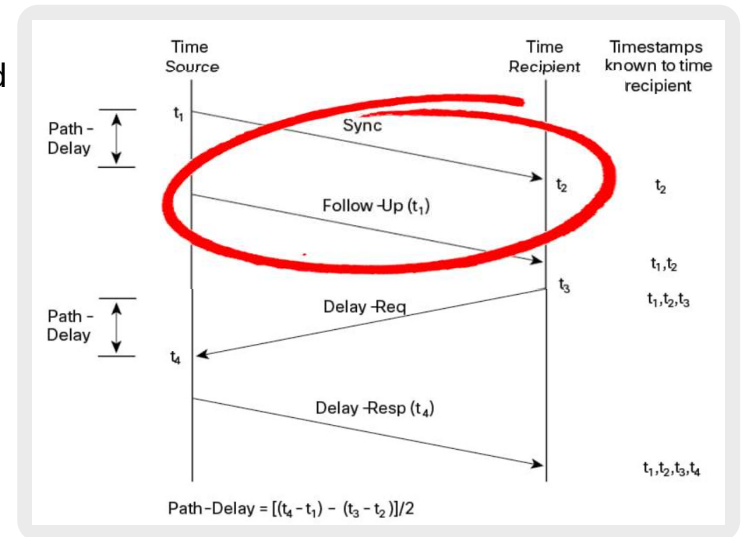
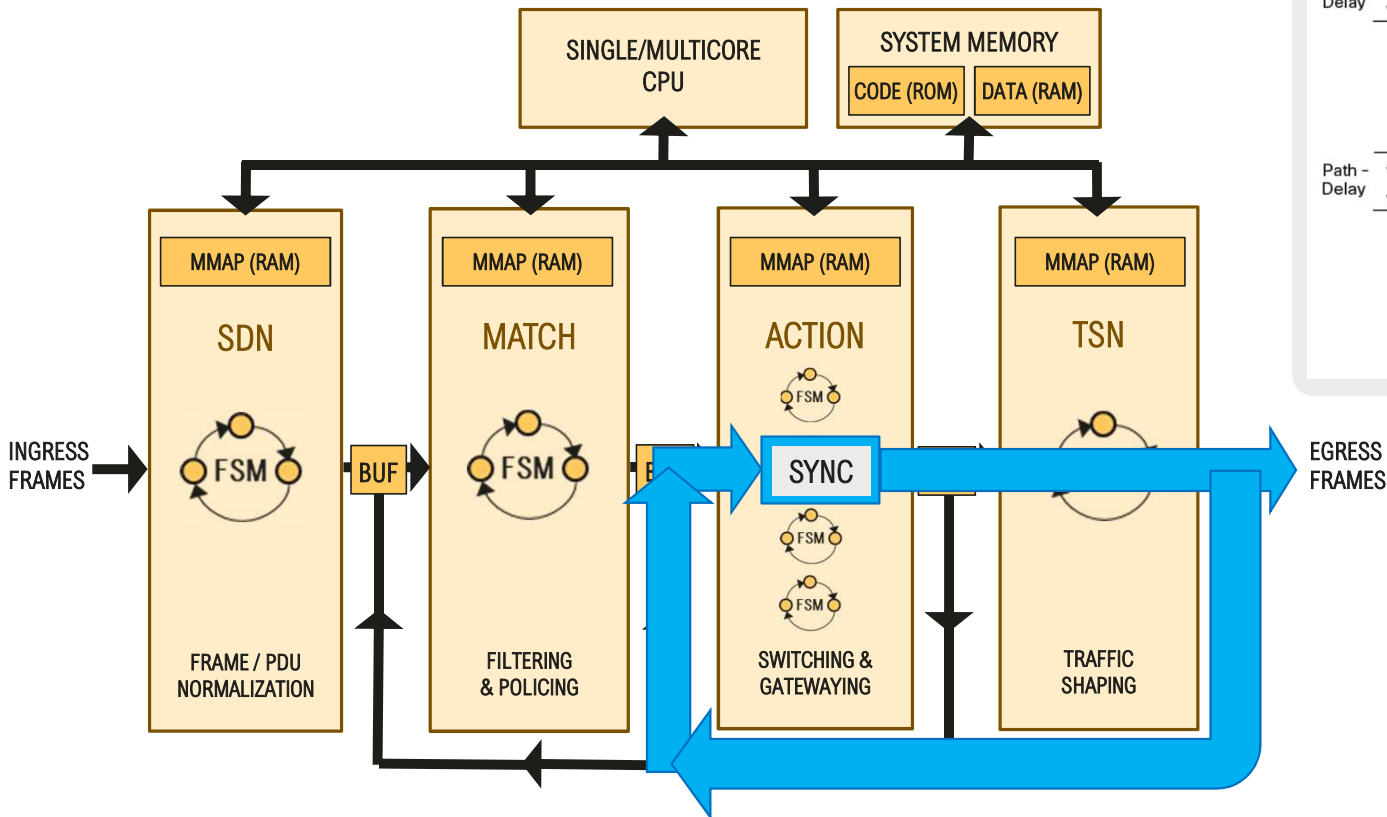
- Many of the TSN standards can be performed directly in HW through dedicated preemption, etc.) but also other features like 802.1CB or 802.1AS



Rationale: Time determinism guaranteed by design in hardware

3.3 INNOVATION 07 : TSN – Time Sensitive Networking

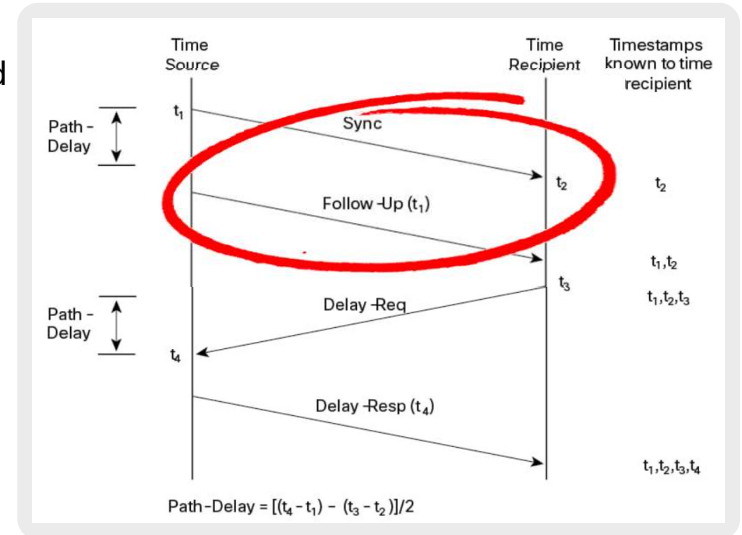
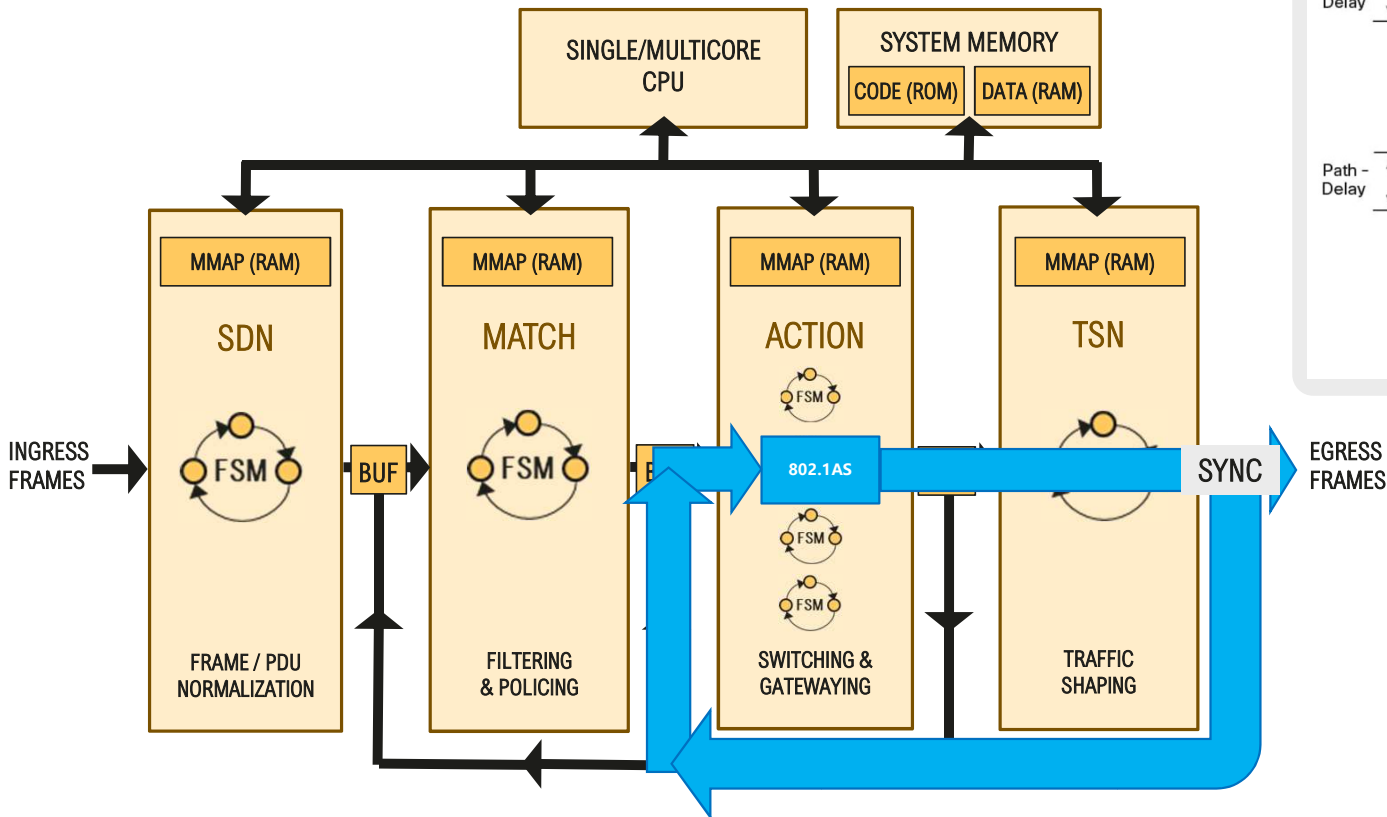
- Many of the TSN standards can be performed directly in HW through dedicated preemption, etc.) but also other features like 802.1CB or 802.1AS



Rationale: Time determinism guaranteed by design in hardware

3.3 INNOVATION 07 : TSN – Time Sensitive Networking

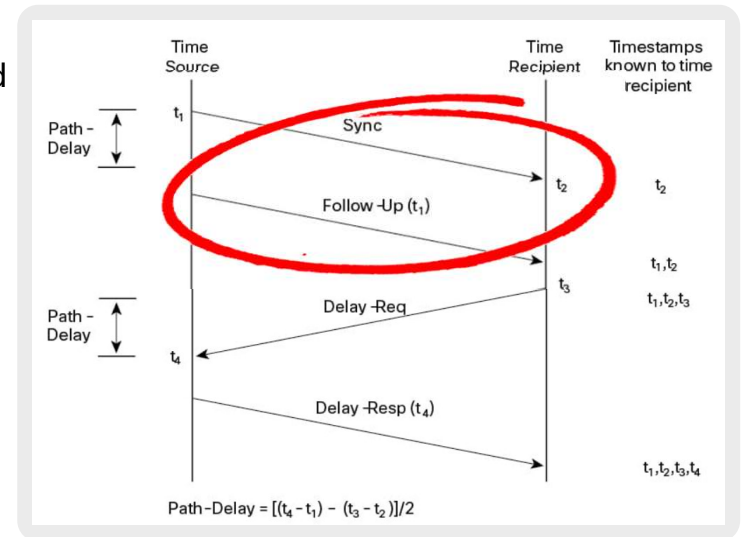
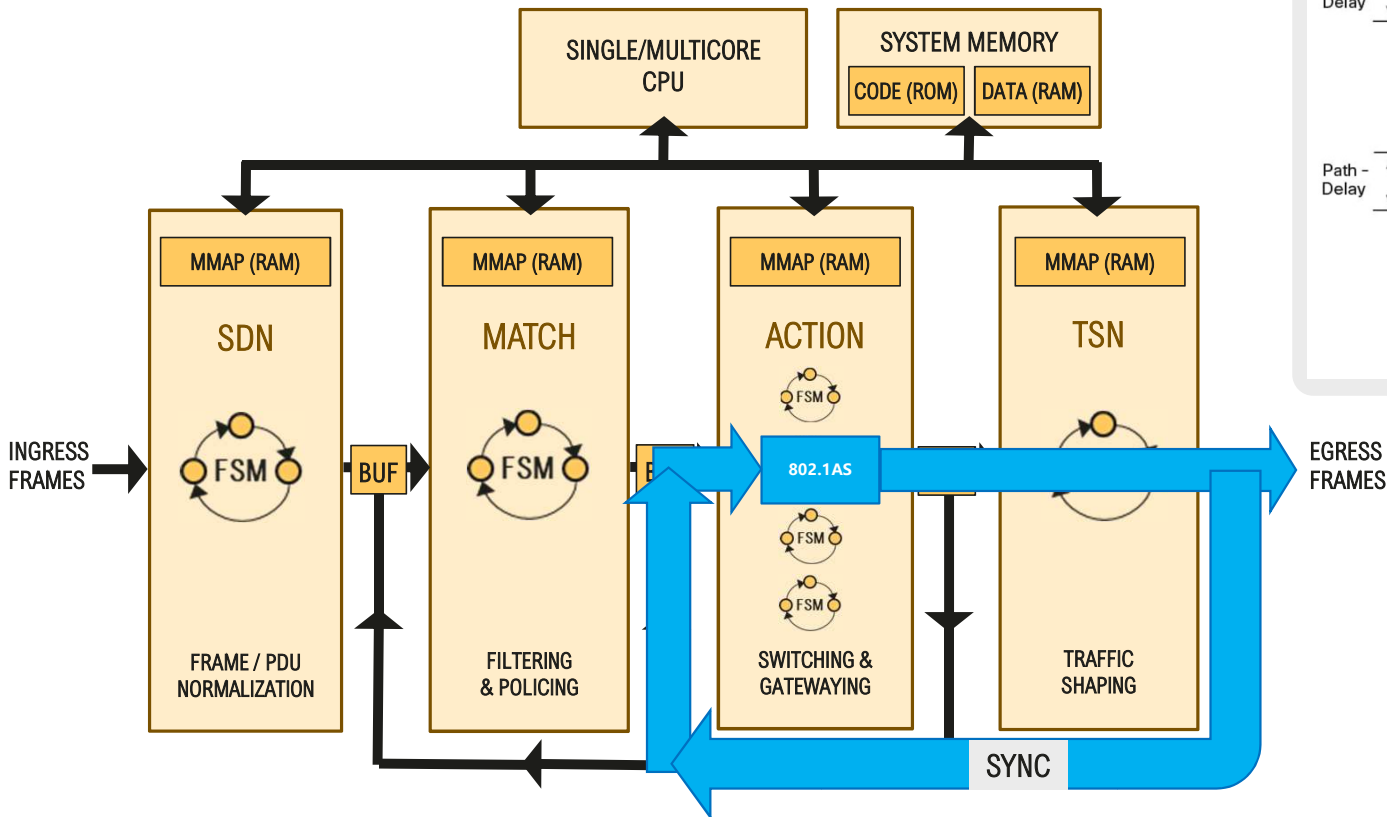
- Many of the TSN standards can be performed directly in HW through dedicated preemption, etc.) but also other features like 802.1CB or 802.1AS



Rationale: Time determinism guaranteed by design in hardware

3.3 INNOVATION 07 : TSN – Time Sensitive Networking

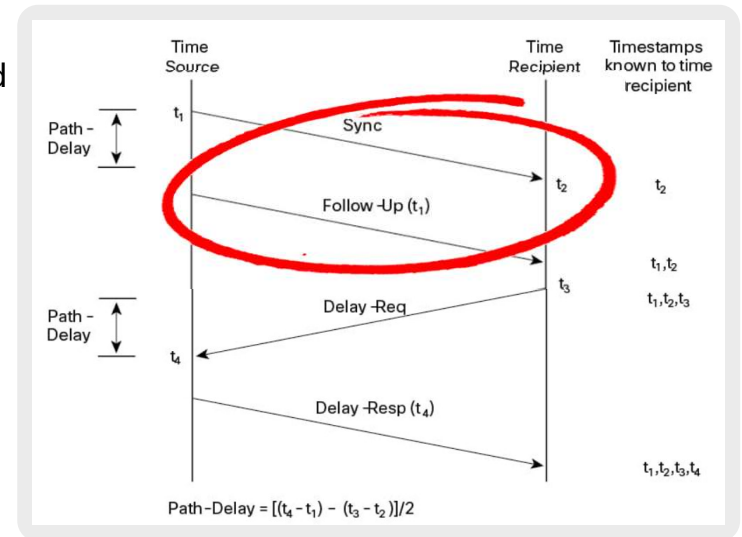
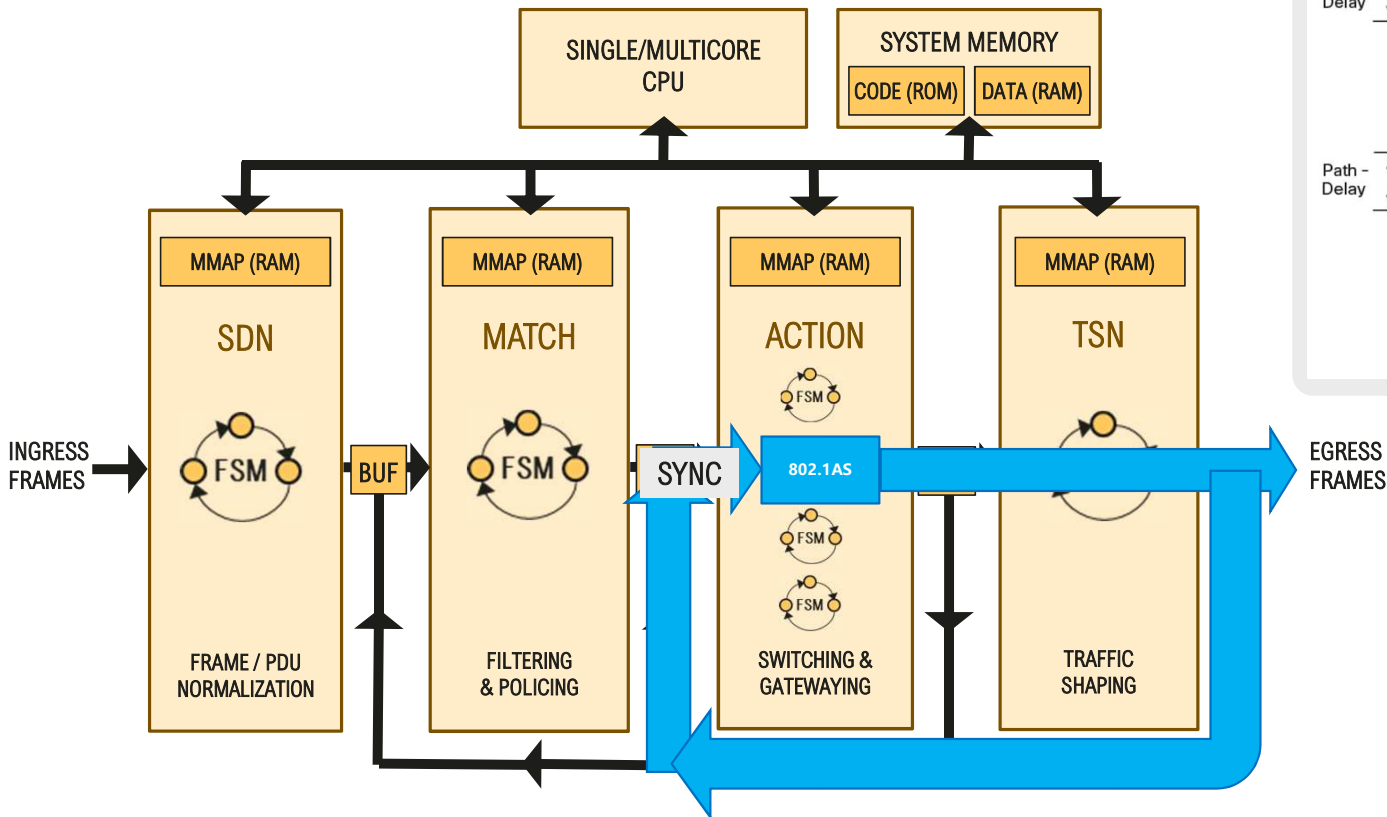
- Many of the TSN standards can be performed directly in HW through dedicated preemption, etc.) but also other features like 802.1CB or 802.1AS



Rationale: Time determinism guaranteed by design in hardware

3.3 INNOVATION 07 : TSN – Time Sensitive Networking

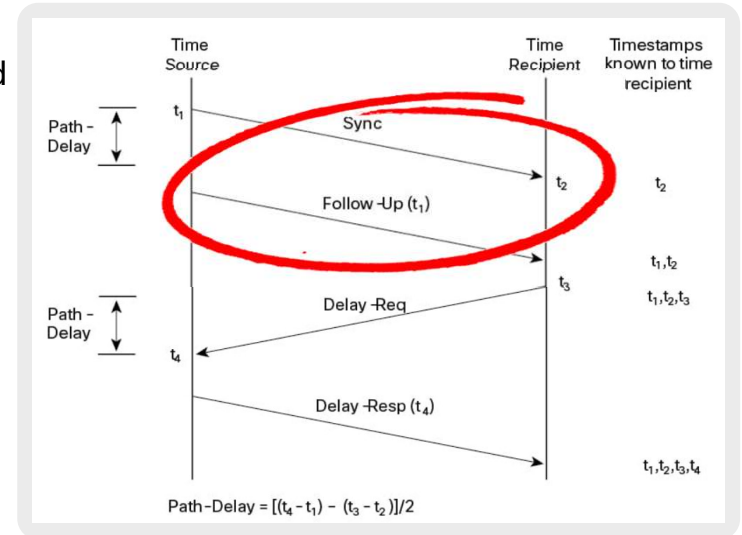
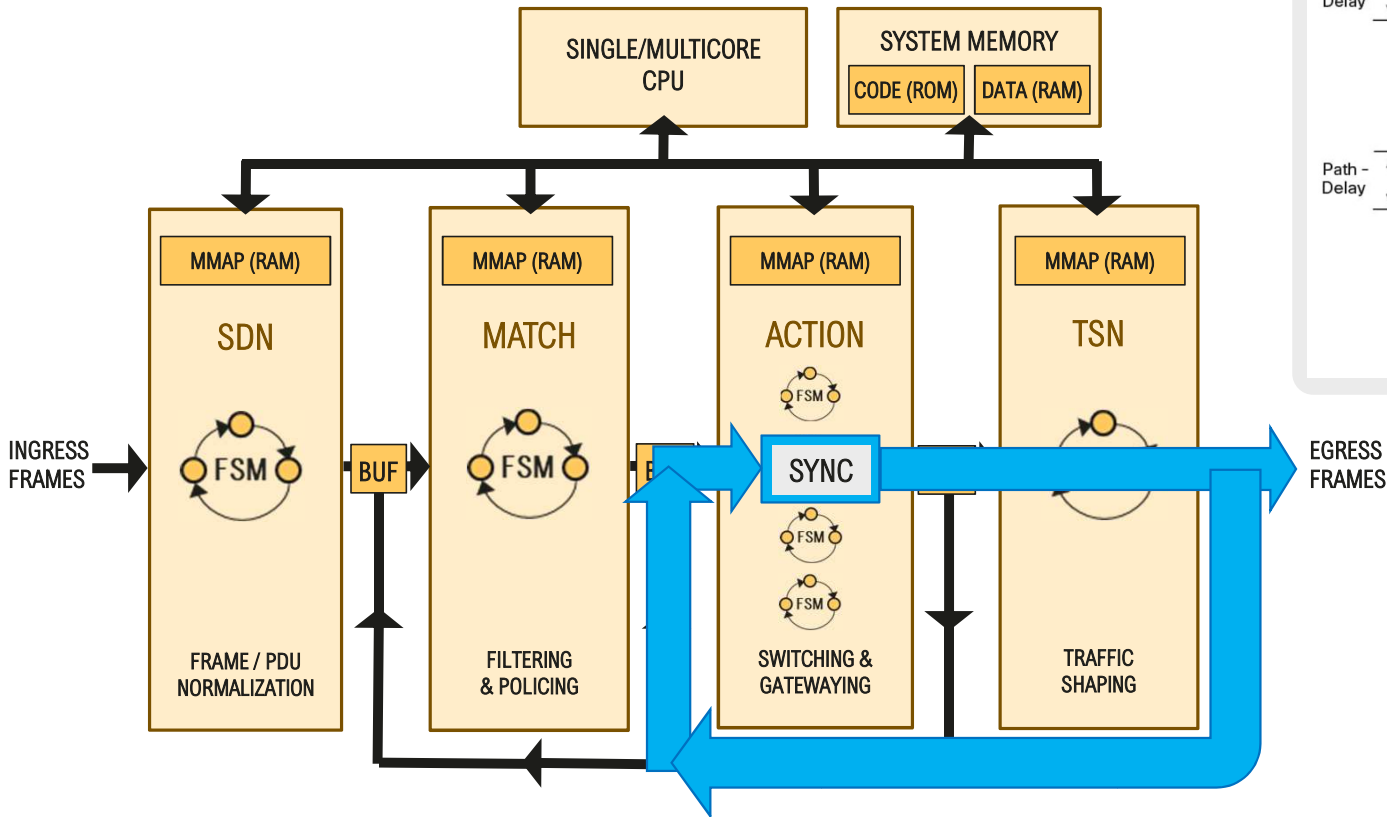
- Many of the TSN standards can be performed directly in HW through dedicated preemption, etc.) but also other features like 802.1CB or 802.1AS



Rationale: Time determinism guaranteed by design in hardware

3.3 INNOVATION 07 : TSN – Time Sensitive Networking

- Many of the TSN standards can be performed directly in HW through dedicated preemption, etc.) but also other features like 802.1CB or 802.1AS

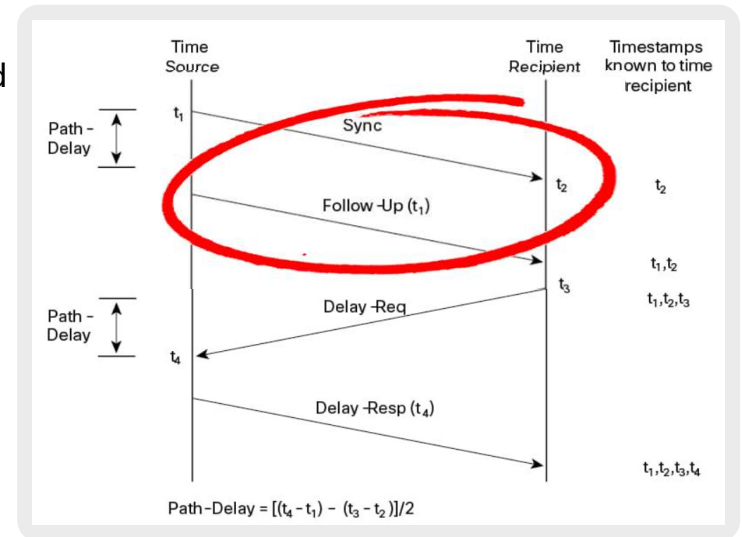
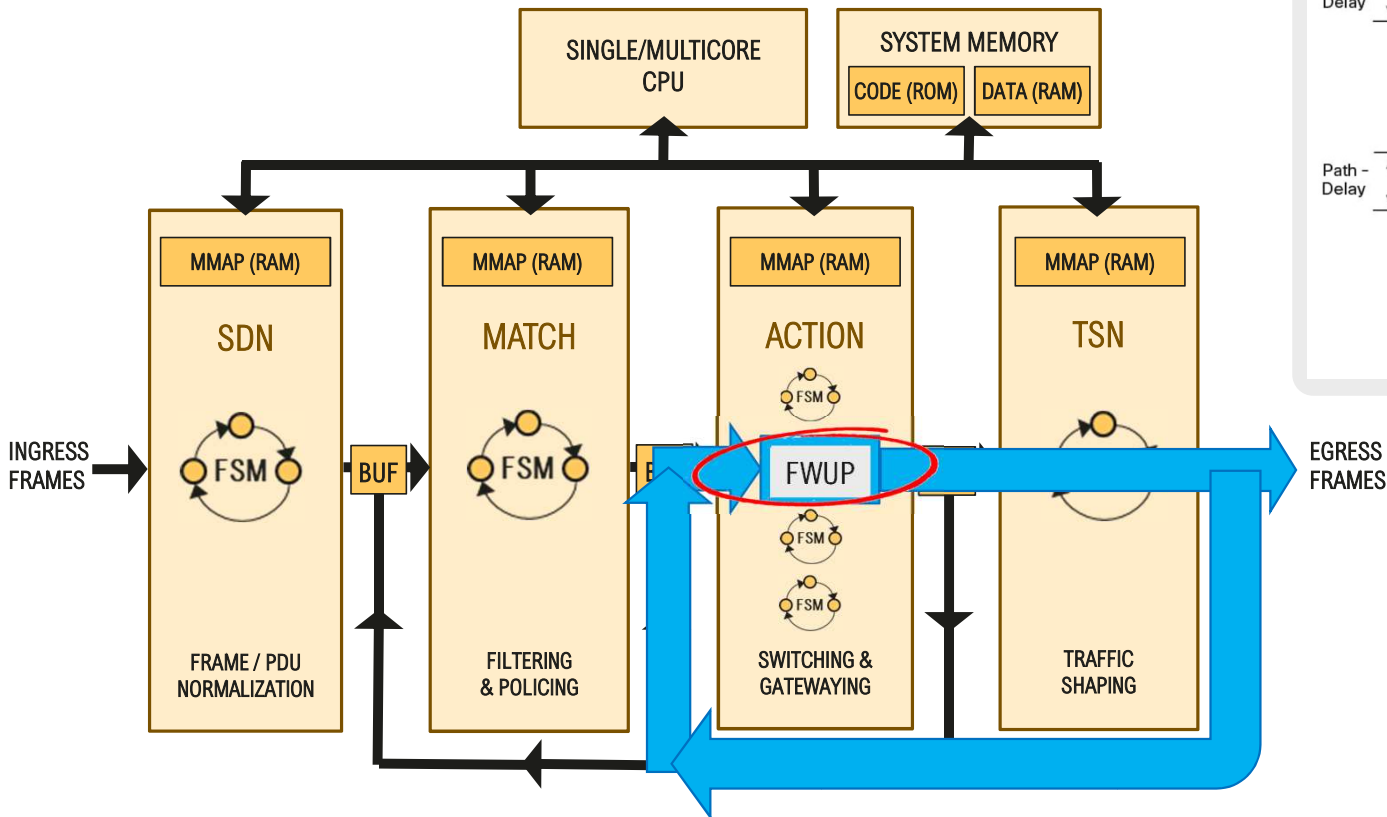


SYNC

Rationale: Time determinism guaranteed by design in hardware

3.3 INNOVATION 07 : TSN – Time Sensitive Networking

- Many of the TSN standards can be performed directly in HW through dedicated preemption, etc.) but also other features like 802.1CB or 802.1AS

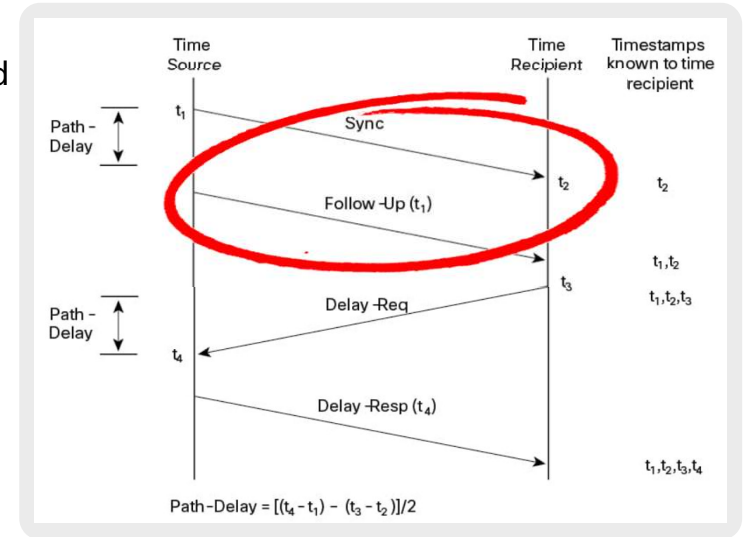
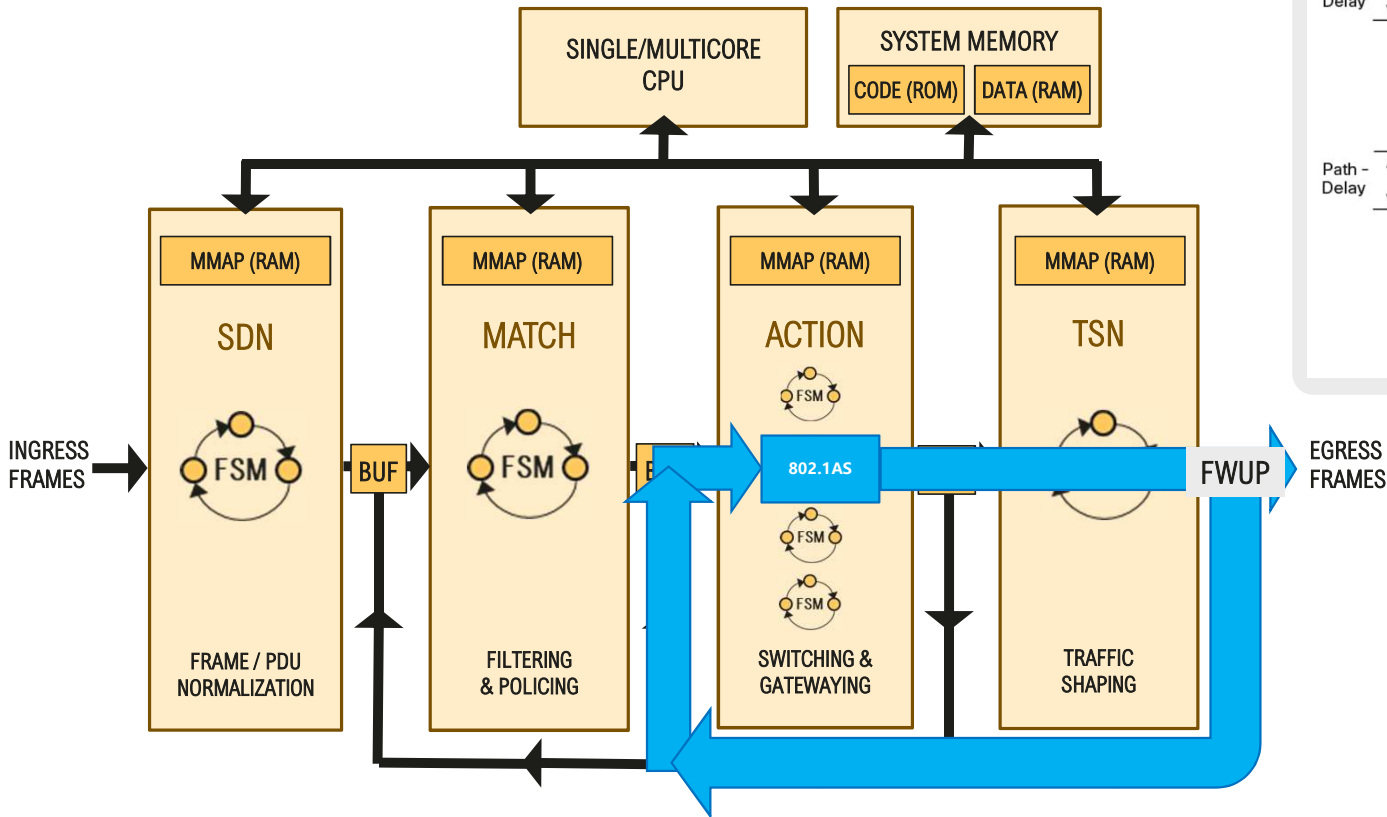


SYNC

Rationale: Time determinism guaranteed by design in hardware

3.3 INNOVATION 07 : TSN – Time Sensitive Networking

- Many of the TSN standards can be performed directly in HW through dedicated preemption, etc.) but also other features like 802.1CB or 802.1AS

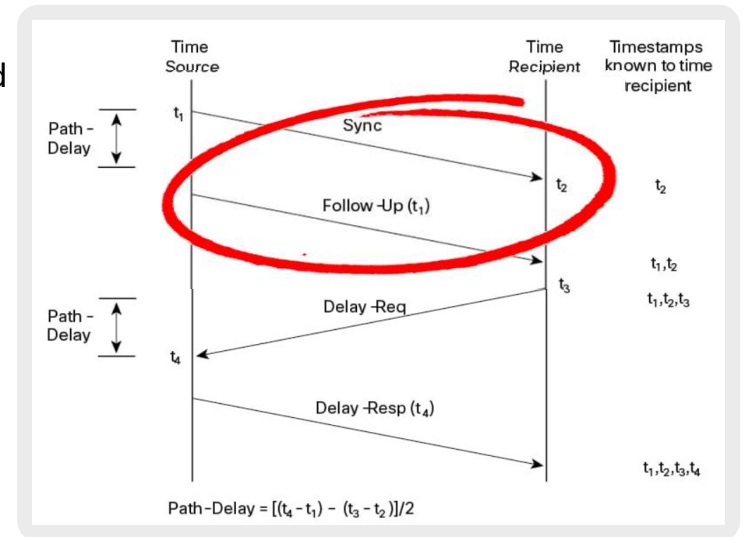
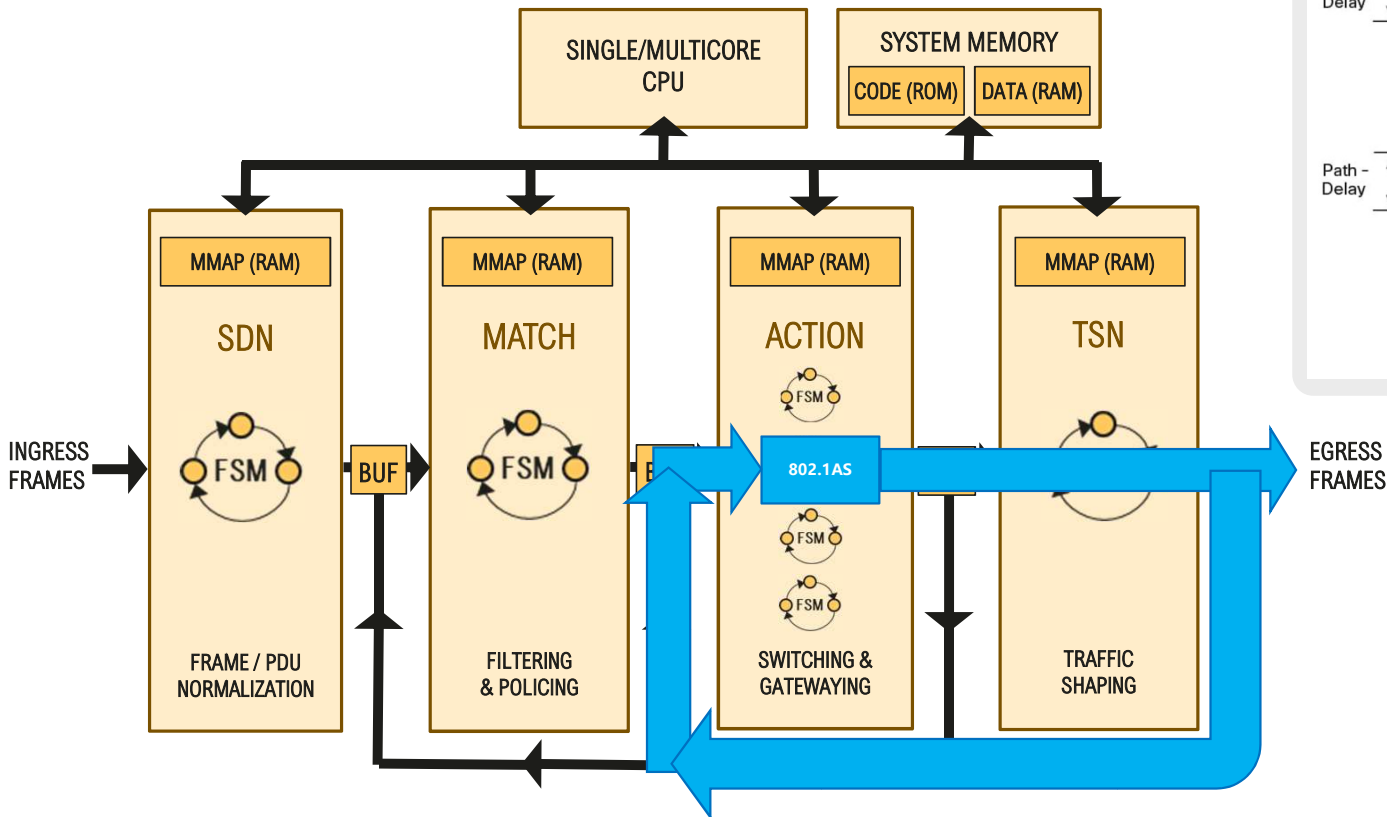


SYNC

Rationale: Time determinism guaranteed by design in hardware

3.3 INNOVATION 07 : TSN – Time Sensitive Networking

- Many of the TSN standards can be performed directly in HW through dedicated preemption, etc.) but also other features like 802.1CB or 802.1AS



FWUP SYNC

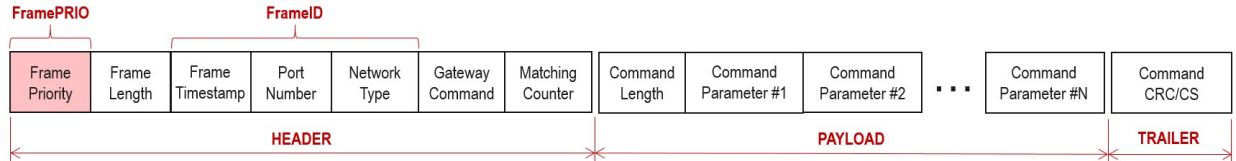
Rationale: Time determinism guaranteed by design in hardware



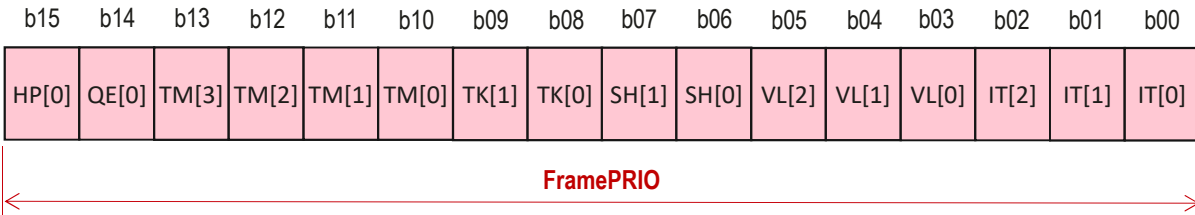
3.3 INNOVATION 07 : DDS – Data Distribution Service deployed in Hardware

- DDS middleware (data-centric Publish-Subscribe model) supported directly in HW, e.g. optimized **time handling** (hard real-time prioritization done on the fly)

INSTRUCTION FRAME (NORMALIZATION):



FramePRIO field:



FramePRIO (16-bits) word:

- Highest priority [**1bit**: interrupt]
- Queue status [**1bit**: (nearly) full, (nearly) empty]
- Timeout/timestamp [**4bits**: time factor]
- Tasks/HWAs status [**2bits**: free, free2taken, taken2free, taken]
- Shaper status [**2bits**: free, free2use, use2free, in use]
- VLAN tag priority [**3bits**: level]
- In-band telemetry status [**3bits**: counter]

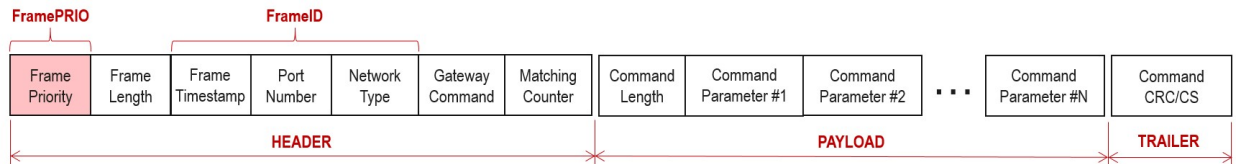
Rationale: Quality-of-Service (QoS) policies guaranteed by design in hardware (e.g. time and priority handling)



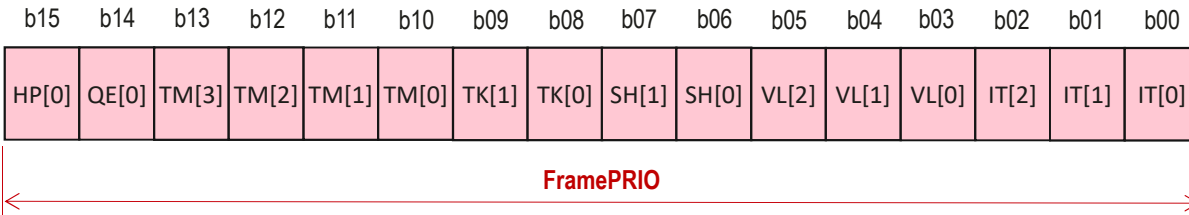
3.3 INNOVATION 07 : DDS – Data Distribution Service deployed in Hardware

- DDS middleware (data-centric Publish-Subscribe model) supported directly in HW, e.g. optimized **time handling** (hard real-time prioritization done on the fly)

INSTRUCTION FRAME (NORMALIZATION):



FramePRIO field:



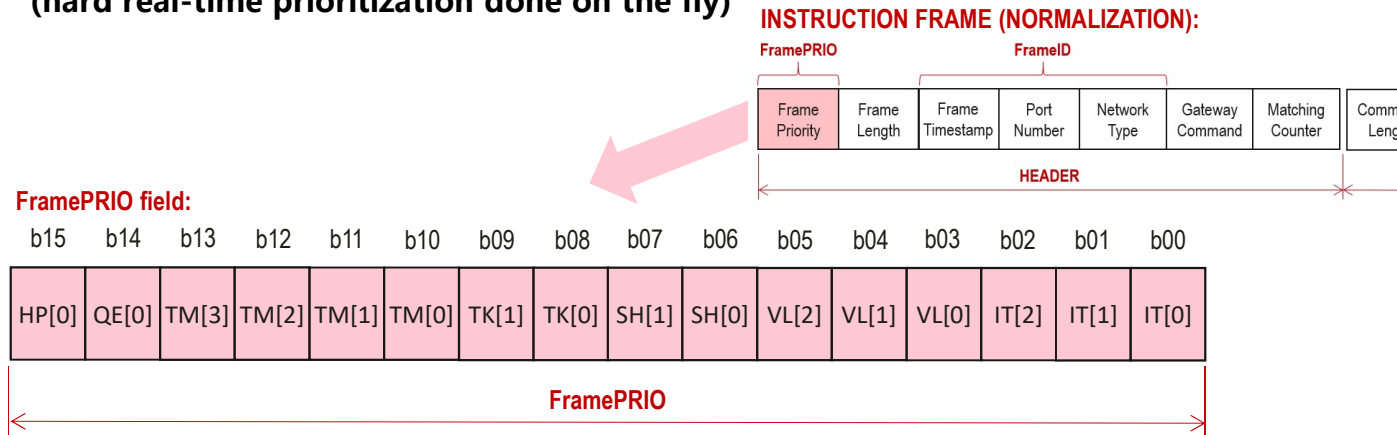
FramePRIO (16-bits) word:

- Highest priority [1bit: interrupt]
- Queue status [1bit: (nearly) full, (nearly) empty]
- Timeout/timestamp [4bits: time factor]**
- Tasks/HWAs status [2bits: free, free2taken, taken2free, taken]
- Shaper status [2bits: free, free2use, use2free, in use]
- VLAN tag priority [3bits: level]
- In-band telemetry status [3bits: counter]

Rationale: Quality-of-Service (QoS) policies guaranteed by design in hardware (e.g. time and priority handling)

3.3 INNOVATION 07: DDS – Data Distribution Service deployed in Hard

- DDS middleware (data-centric Publish-Subscribe model) supported directly in HW, e.g. optim (hard real-time prioritization done on the fly)



FramePRIO (16-bits) word:

- Highest priority [1bit: interrupt]
- Queue status [1bit: (nearly) full, (nearly) empty]
- Timeout/timestamp [4bits: time factor]**
- Tasks/HWAs status [2bits: free, free2taken, taken2free, taken]
- Shaper status [2bits: free, free2use, use2free, in use]
- VLAN tag priority [3bits: level]
- In-band telemetry status [3bits: counter]

Rationale: Quality-of-Service (QoS) policies guaranteed by design in hardware (e.g. time and p

TABLE 1. Supported QoS policies of DDS version 1.4

QoS policy	Description	DataReader	DataWriter	DomainParticipant	Publisher	Subscriber	Topic
USER_DATA	Custom user data	x	x	x			
TOPIC_DATA	Custom user data				x	x	
GROUP_DATA	Custom user data				x	x	
DURABILITY	If data should "outlive" their writing time (e.g. late-joining DataReaders) ^{abc}	x	x				x
DURABILITY_SERVICE	Specifies the service implementing the durability (if any) ^b		x				x
PRESENTATION	How changes to data are presented to subscribing applications ^{abc}				x	x	
DEADLINE	Maximum time after which DataReader expects an update of periodic data ^{ac}	x	x				x
LATENCY_BUDGET	Maximum delay from data write to data reception and notification ^{ac}	x	x				x
OWNERSHIP	If multiple DataWriters can write the same data instance ^{abc}	x	x				x
OWNERSHIP_STRENGTH	Strength of the DataWriter for arbitration in case of exclusive OWNERSHIP ^c		x				
LIVELINESS	Mechanism to determine if an entity is active ("alive") ^{abc}	x	x				x
TIME_BASED_FILTER	Minimum time a DataReader is interested in receiving updates	x					
PARTITION	Logical partition among the topics visible by the Publisher and the Subscriber ^c				x	x	
RELIABILITY	Reliability level of message delivery ^{abc}	x	x				x
TRANSPORT_PRIORITY	Priority to be used on underlying transport ^c		x				x
LIFESPAN	Maximum time of validity of written data, to avoid delivery of "stale" data ^c	x					x
DESTINATION_ORDER	Logical order among changes made by Publishers to the same data instance ^{abc}	x	x				x
HISTORY	Behavior in case a sample changes before being communicated ^b	x	x				x
RESOURCE_LIMITS	Maximum amount of resources consumed by the service ^b	x	x				x
ENTITY_FACTORY	Behavior of an entity when creating other entities				x	x	x
WRITER_DATA_LIFECYCLE	Behavior of DataWriter with respect to the lifecycle of the data-instances		x				
READER_DATA_LIFECYCLE	Behavior of DataReader with respect to the lifecycle of the data-instances	x					

^a Values on the publishing and subscribing sides must be compatible.

^b Not changeable.

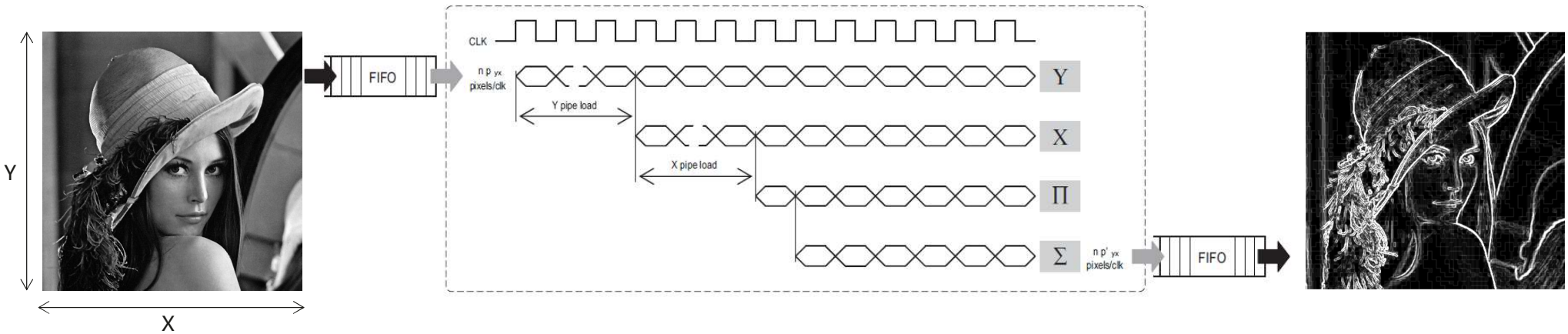
^c May appear as in-line QoS inside RTPS messages.



3.3 INNOVATION 09 : APP – Applications embedded in HWAs

- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution (2DCONV) HW engine as part of a convolutional neural network (CNN) HW engine used in image processing or intrusion detection system

$$p'(y,x) = \sum_{j=-n}^n \sum_{i=-m}^m K(j,i) \cdot p(y+j,x+i)$$

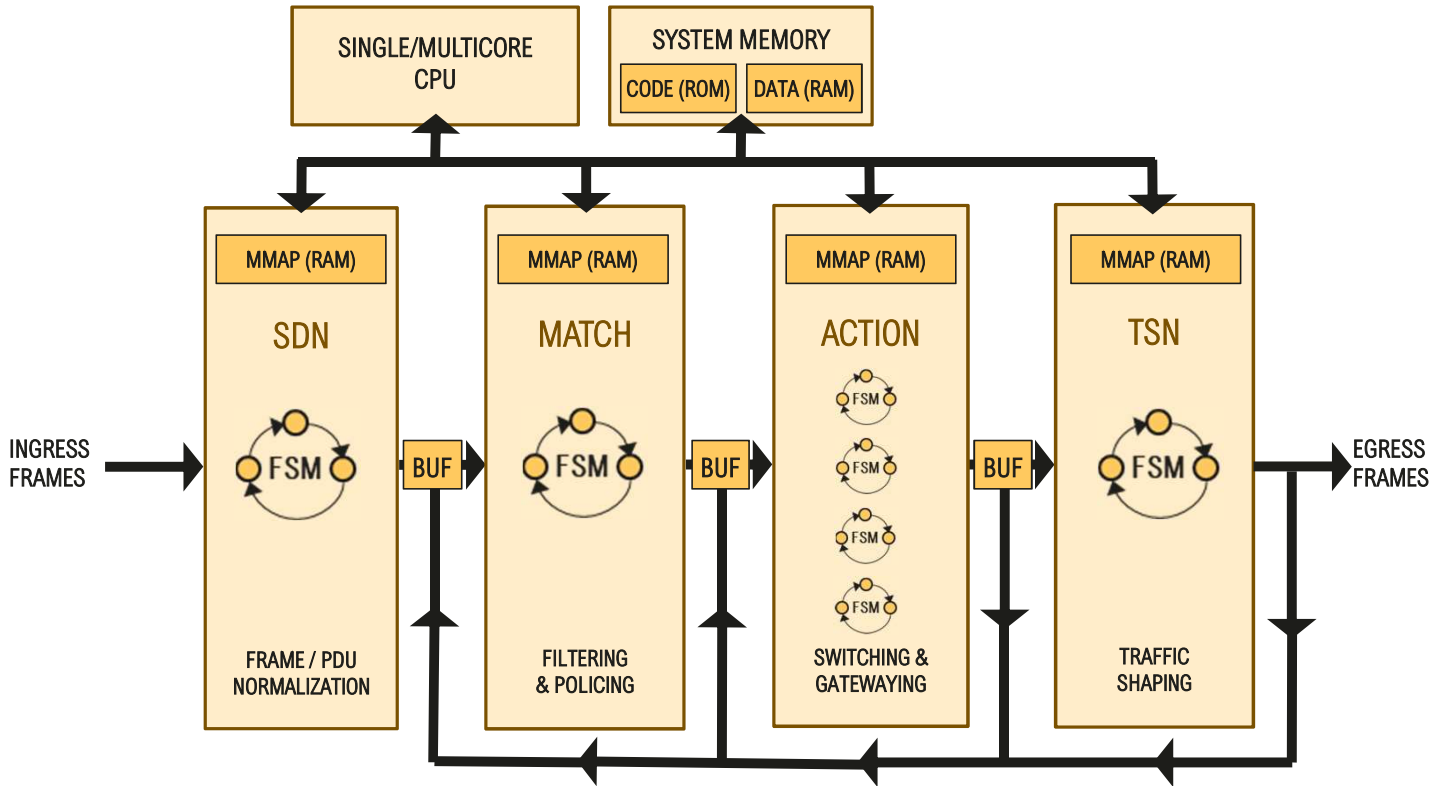


Rationale: Effective spatial processing doing it inline and in-memory



3.3 INNOVATION 09 : APP – Applications embedded in HWAs

- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution (2DCONV) HW engine as part of a convolutional neural network (CNN) HW engine used in image processing or intrusion detection system

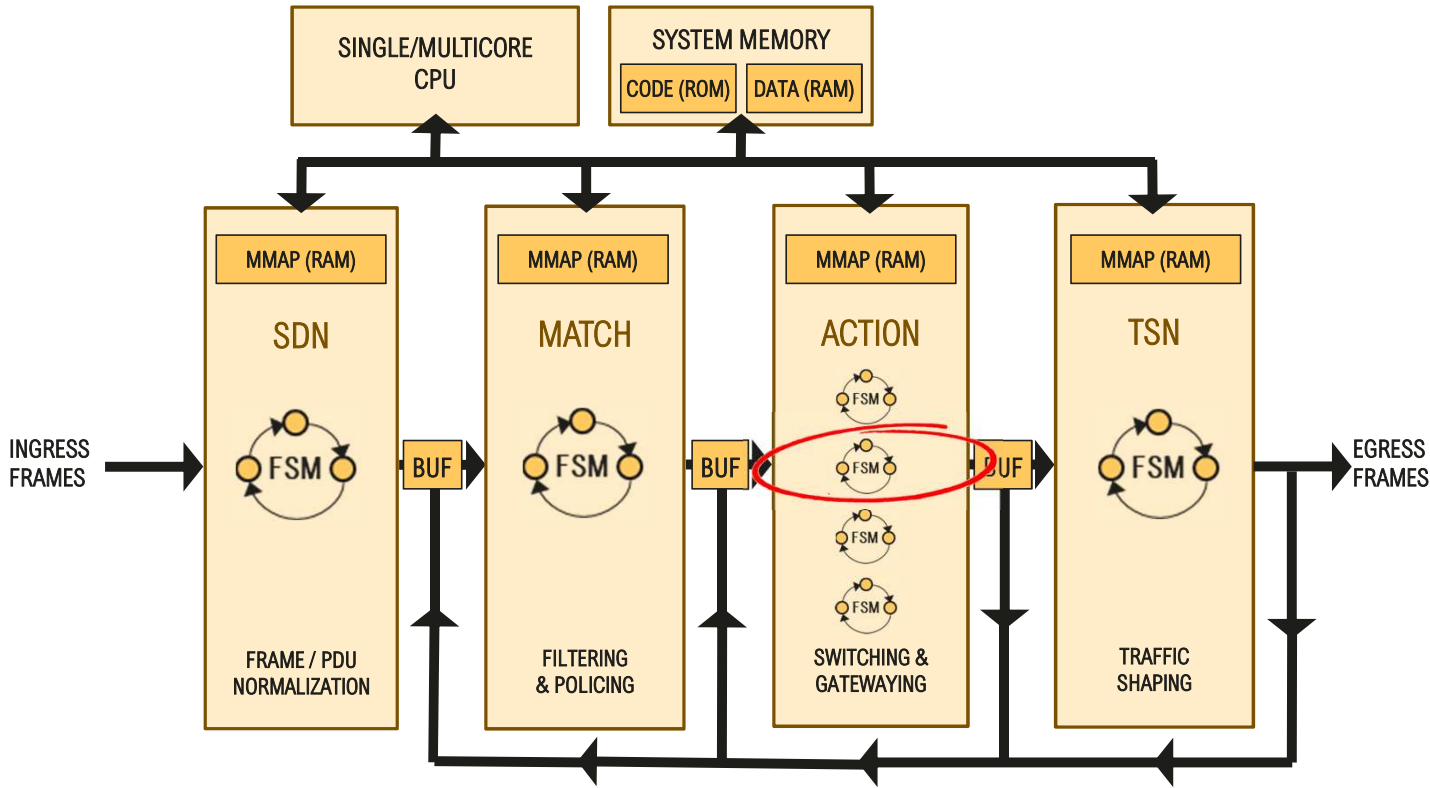


Rationale: Effective spatial processing doing it inline and in-memory



3.3 INNOVATION 09 : APP – Applications embedded in HWAs

- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution (2DCONV) HW engine as part of a convolutional neural network (CNN) HW engine used in image processing or intrusion detection system

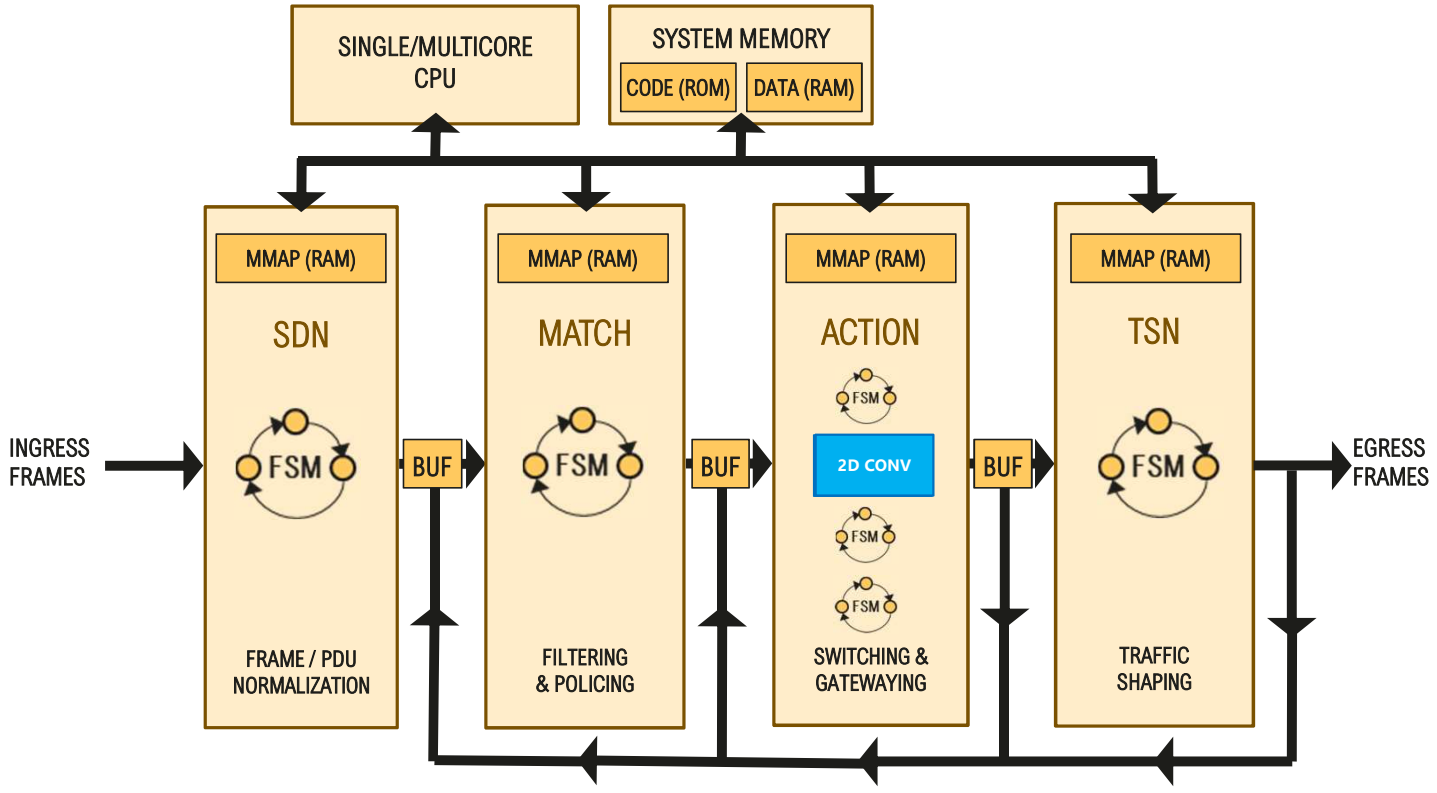


Rationale: Effective spatial processing doing it inline and in-memory



3.3 INNOVATION 09 : APP – Applications embedded in HWAs

- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution (2DCONV) HW engine as part of a convolutional neural network (CNN) HW engine used in image processing or intrusion detection system



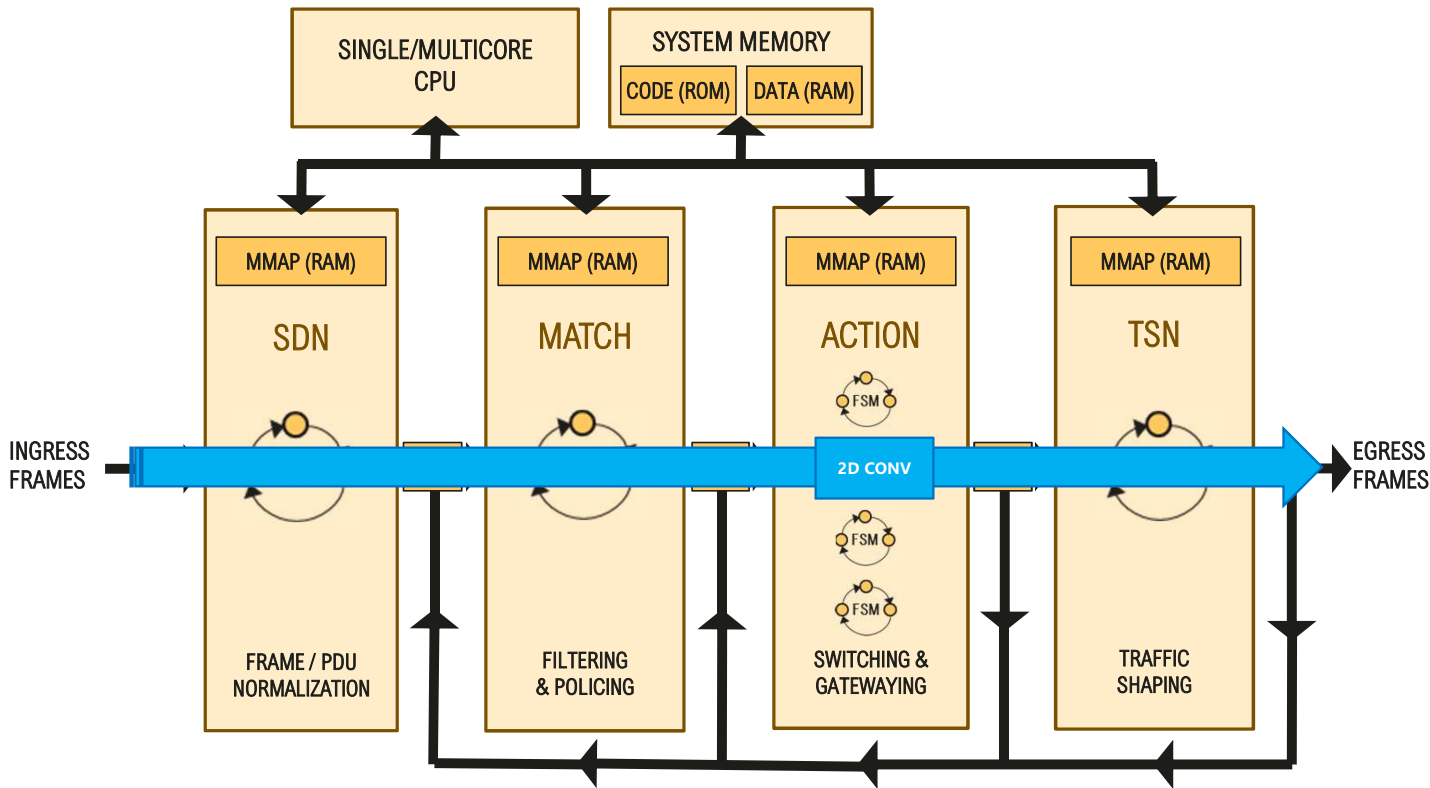
Rationale: Effective spatial processing doing it inline and in-memory





3.3 INNOVATION 09 : APP – Applications embedded in HWAs

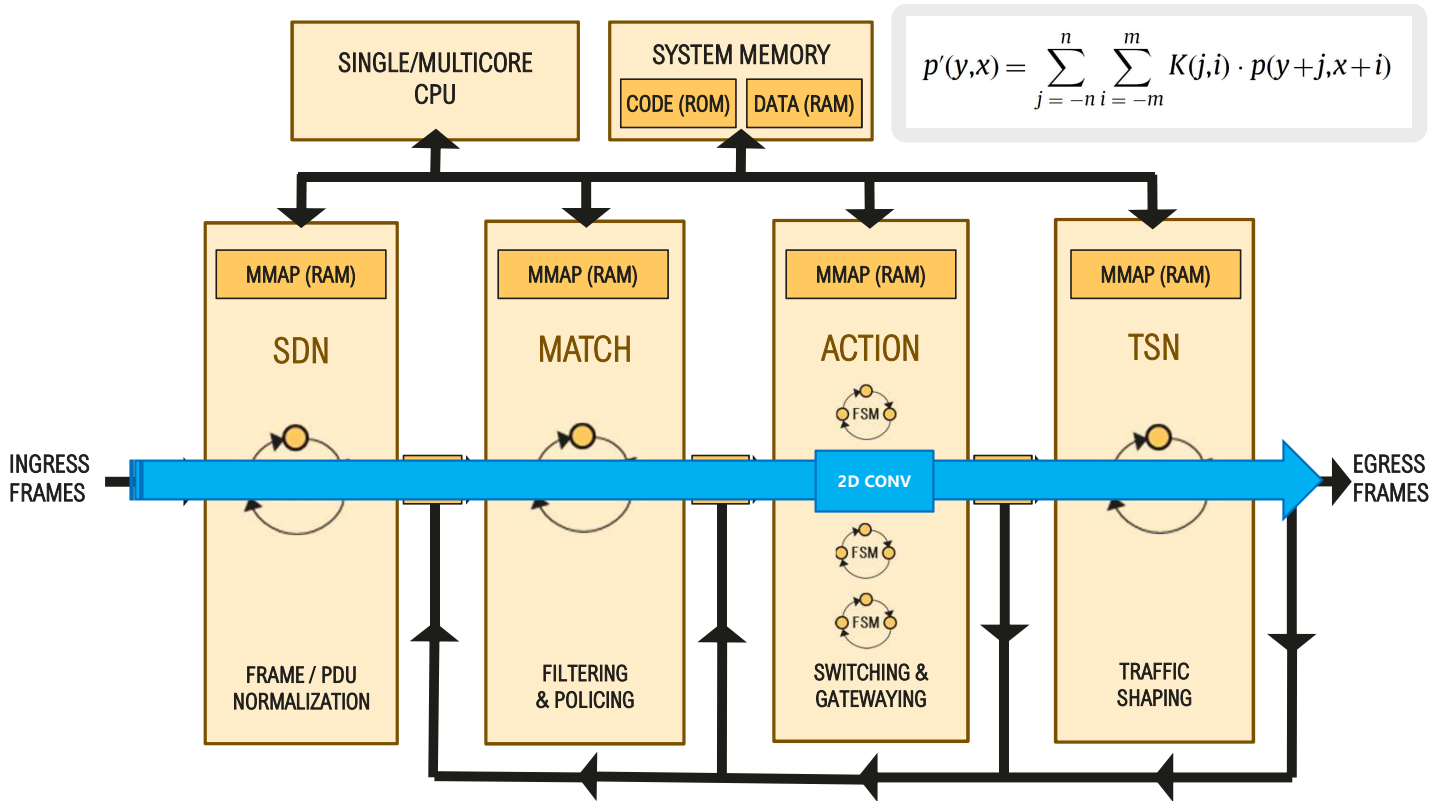
- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution (2D CONV) HW engine as part of a convolutional neural network (CNN) HW engine used in image processing or intrusion detection system



Rationale: Effective spatial processing doing it inline and in-memory

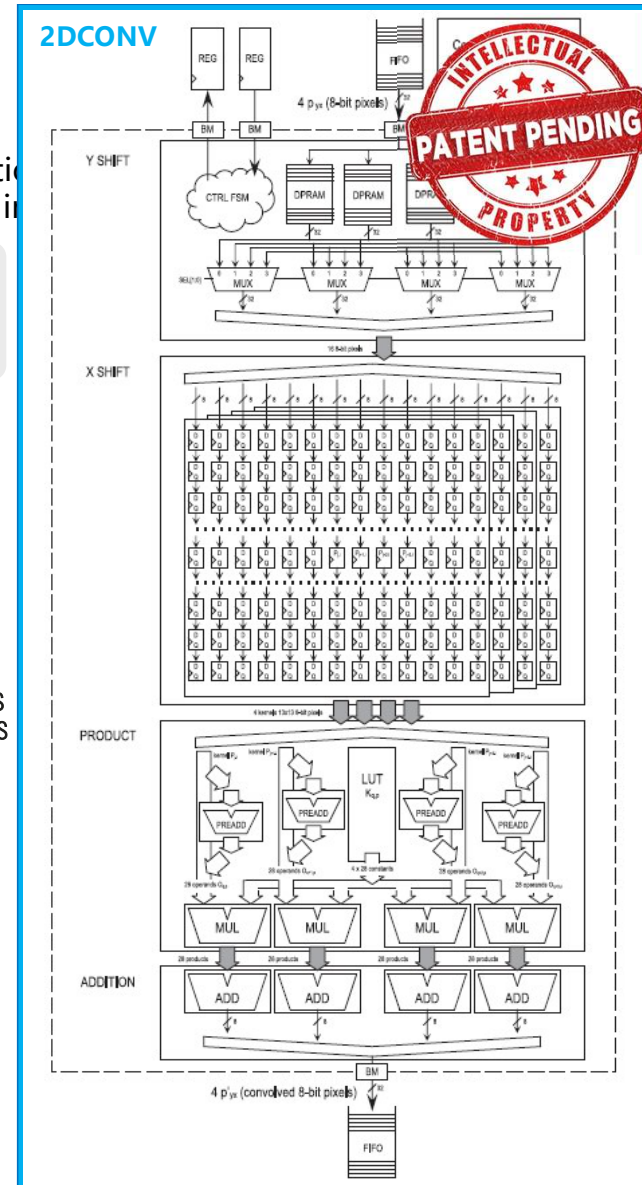
3.3 INNOVATION 09 : APP – Applications embedded in HWAs

- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution as part of a convolutional neural network (CNN) HW engine used in image processing or in



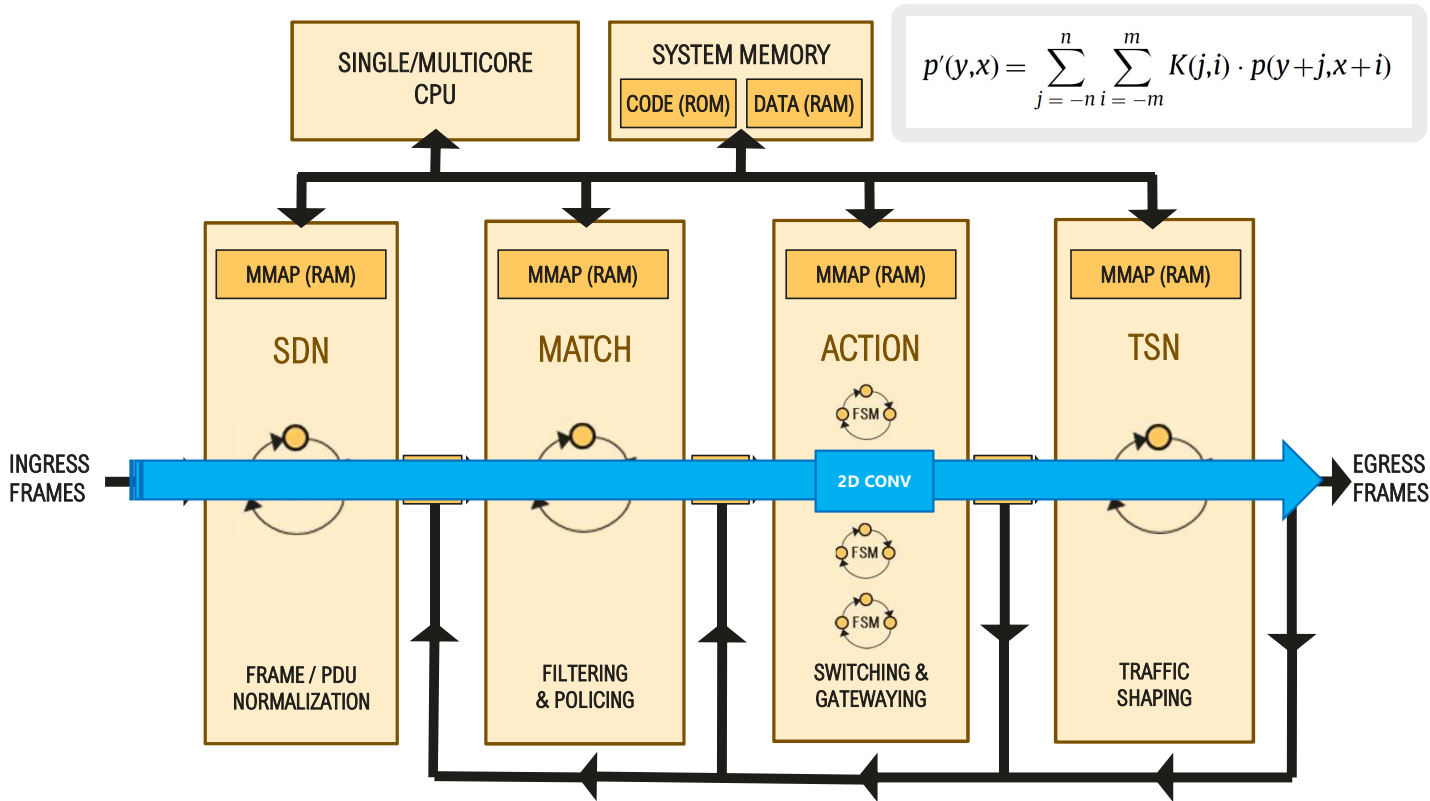
$$p'(y,x) = \sum_{j=-n}^n \sum_{i=-m}^m K(j,i) \cdot p(y+j,x+i)$$

Rationale: Effective spatial processing doing it inline and in-memory



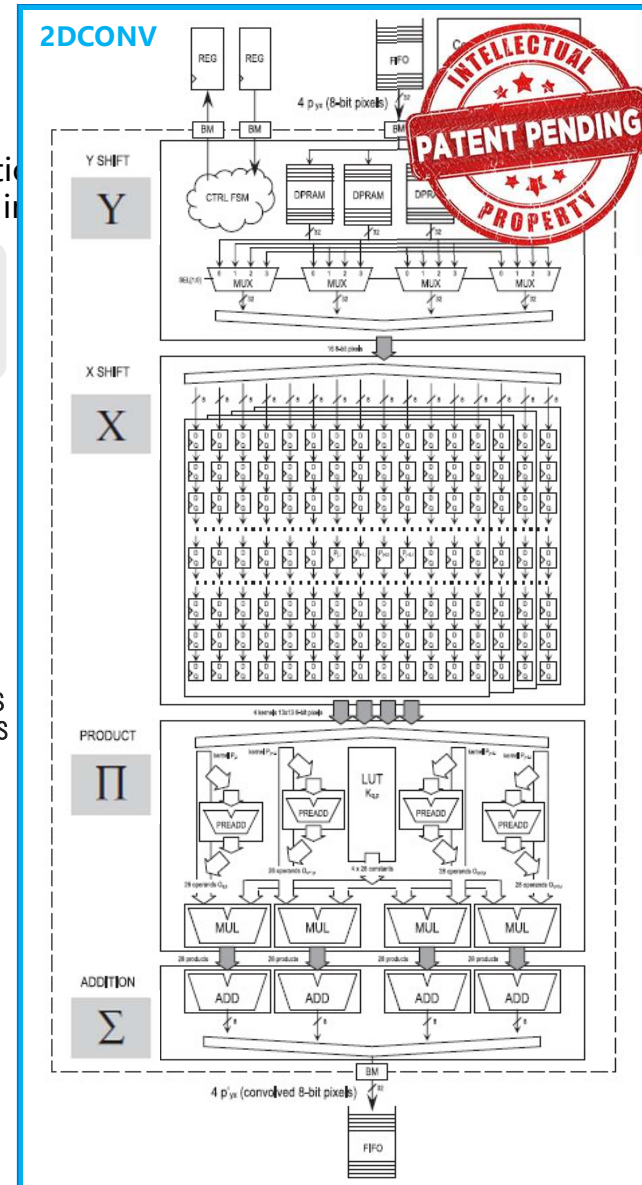
3.3 INNOVATION 09 : APP – Applications embedded in HWAs

- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution as part of a convolutional neural network (CNN) HW engine used in image processing or in



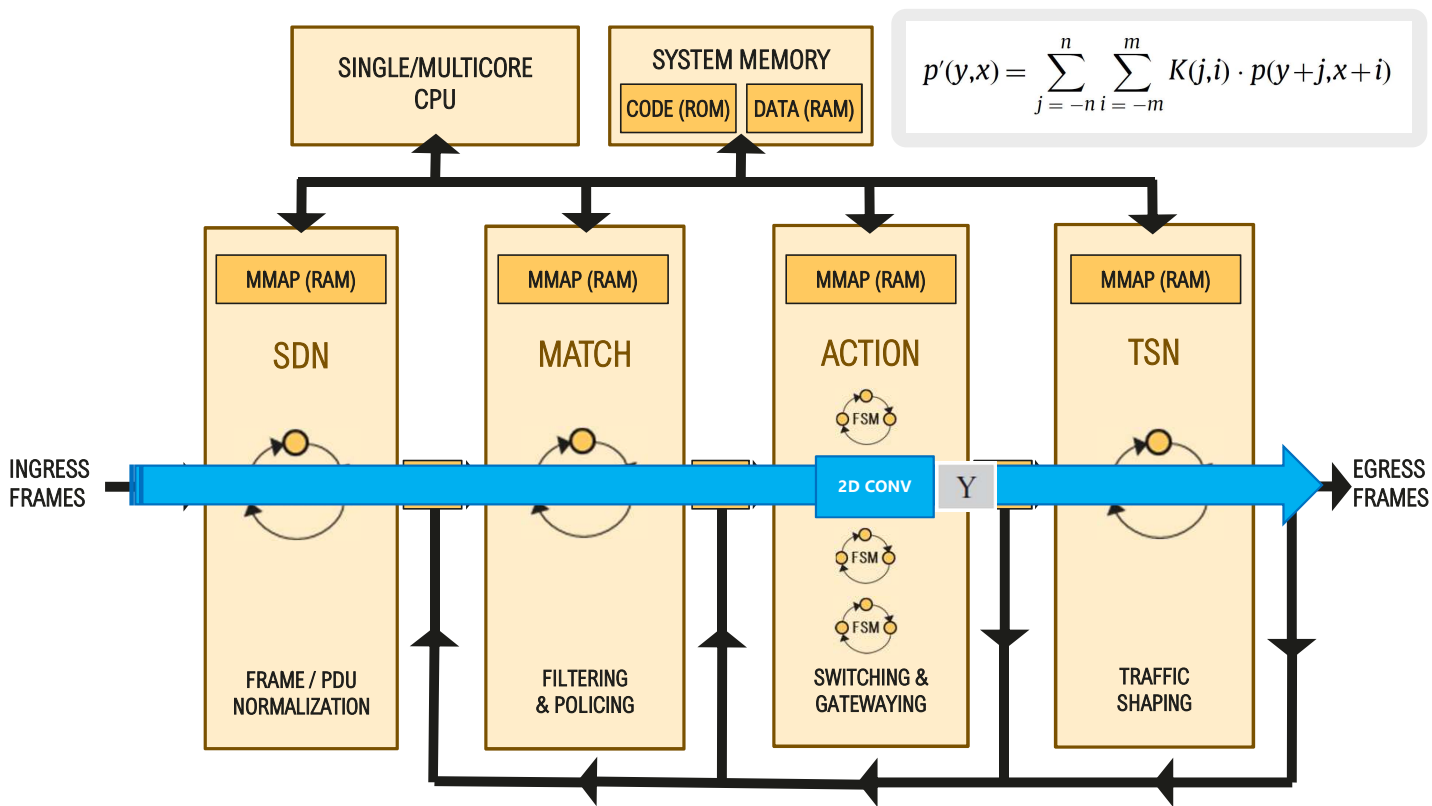
$$p'(y,x) = \sum_{j=-n}^n \sum_{i=-m}^m K(j,i) \cdot p(y+j,x+i)$$

Rationale: Effective spatial processing doing it inline and in-memory



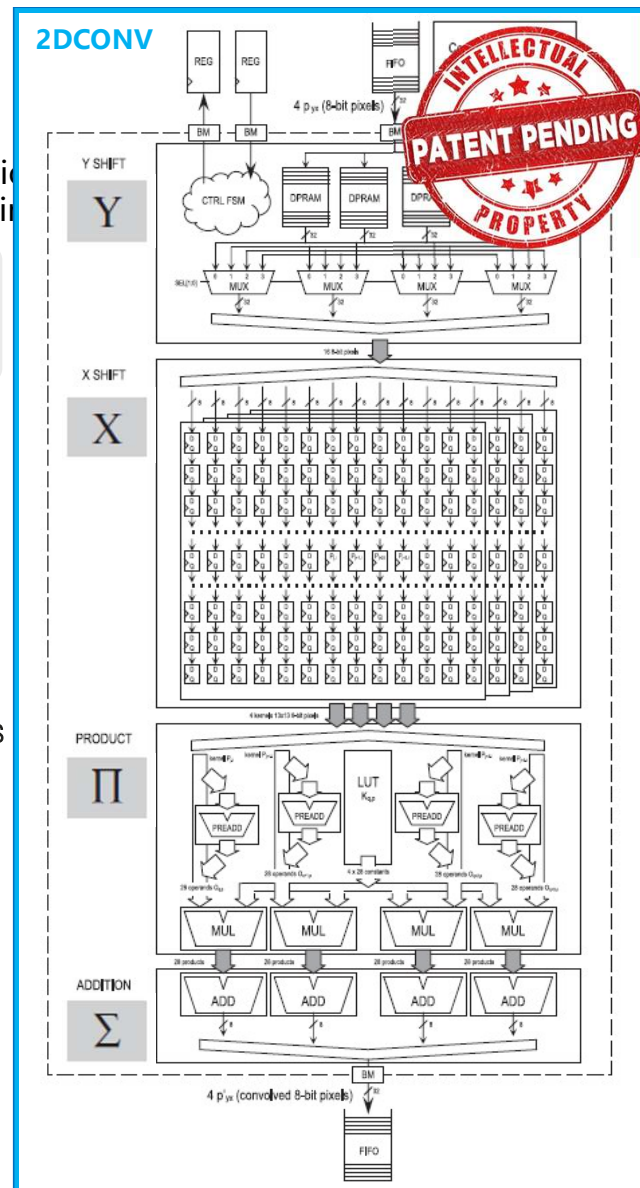
3.3 INNOVATION 09 : APP – Applications embedded in HWAs

- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution as part of a convolutional neural network (CNN) HW engine used in image processing or in



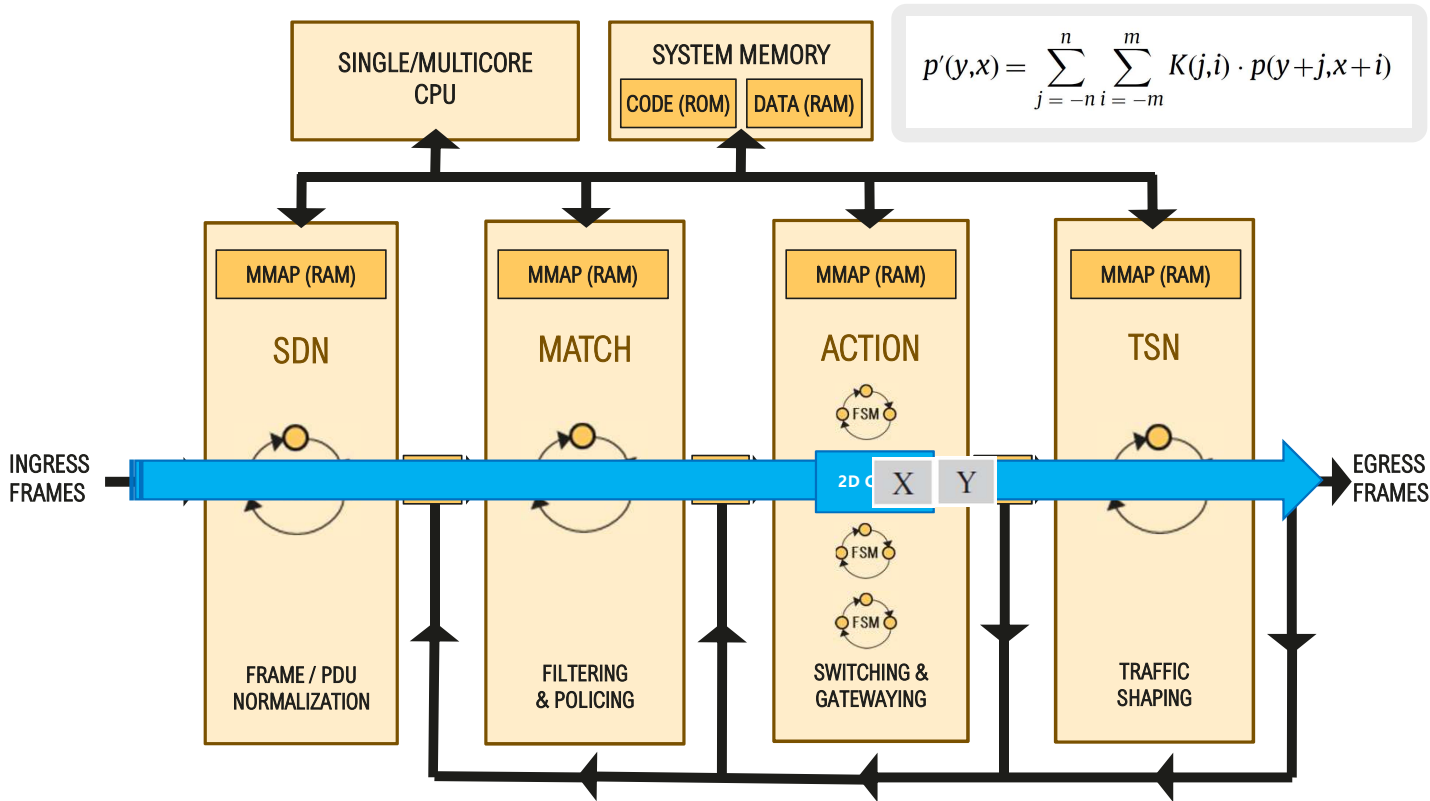
$$p'(y,x) = \sum_{j=-n}^n \sum_{i=-m}^m K(j,i) \cdot p(y+j,x+i)$$

Rationale: Effective spatial processing doing it inline and in-memory



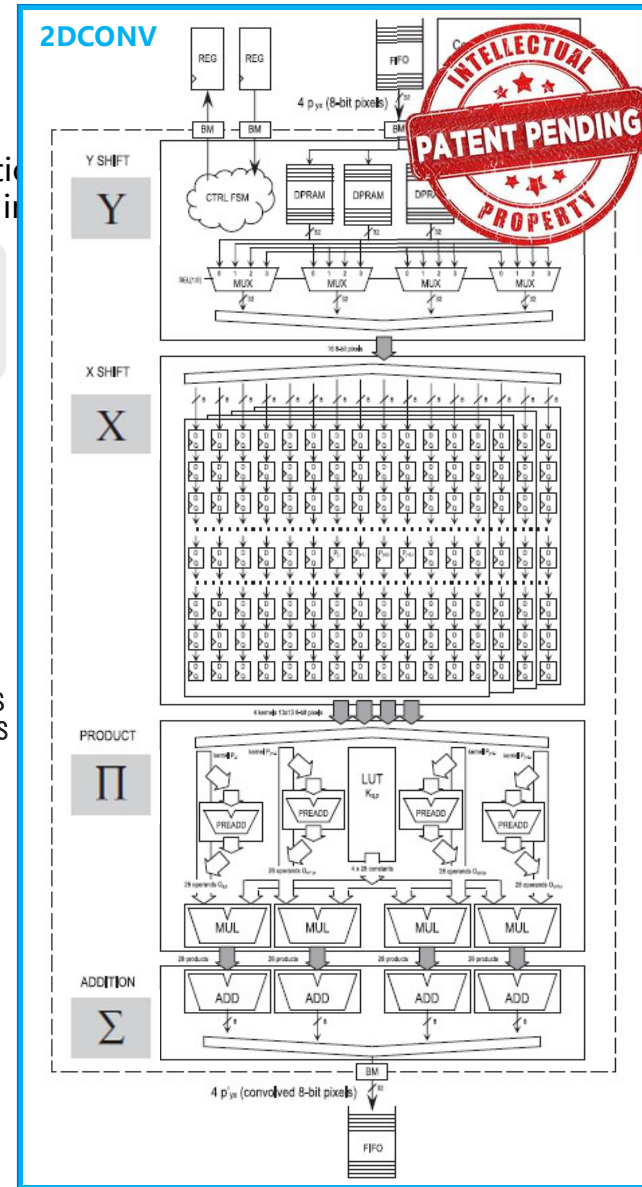
3.3 INNOVATION 09 : APP – Applications embedded in HWAs

- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution as part of a convolutional neural network (CNN) HW engine used in image processing or in



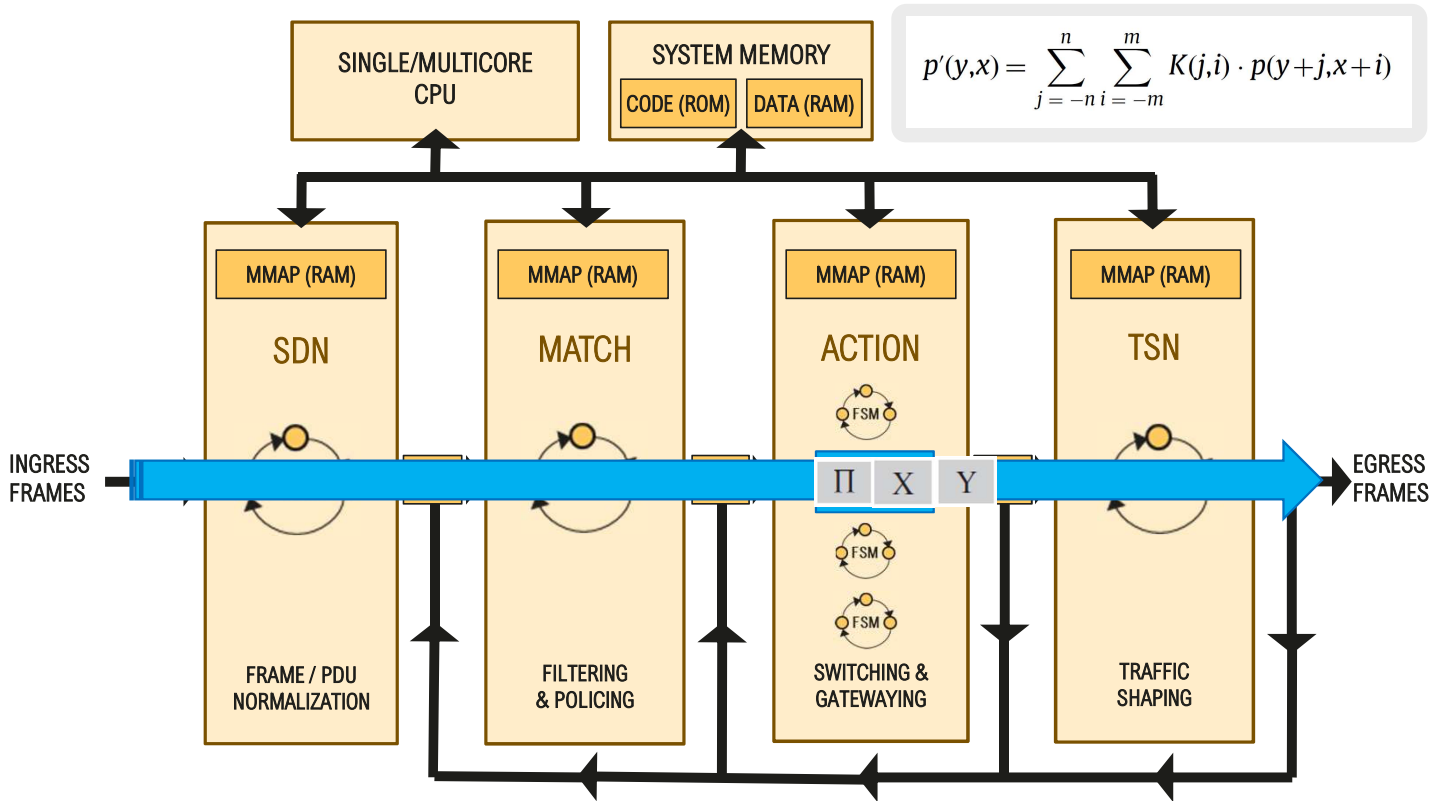
$$p'(y,x) = \sum_{j=-n}^n \sum_{i=-m}^m K(j,i) \cdot p(y+j,x+i)$$

Rationale: Effective spatial processing doing it inline and in-memory



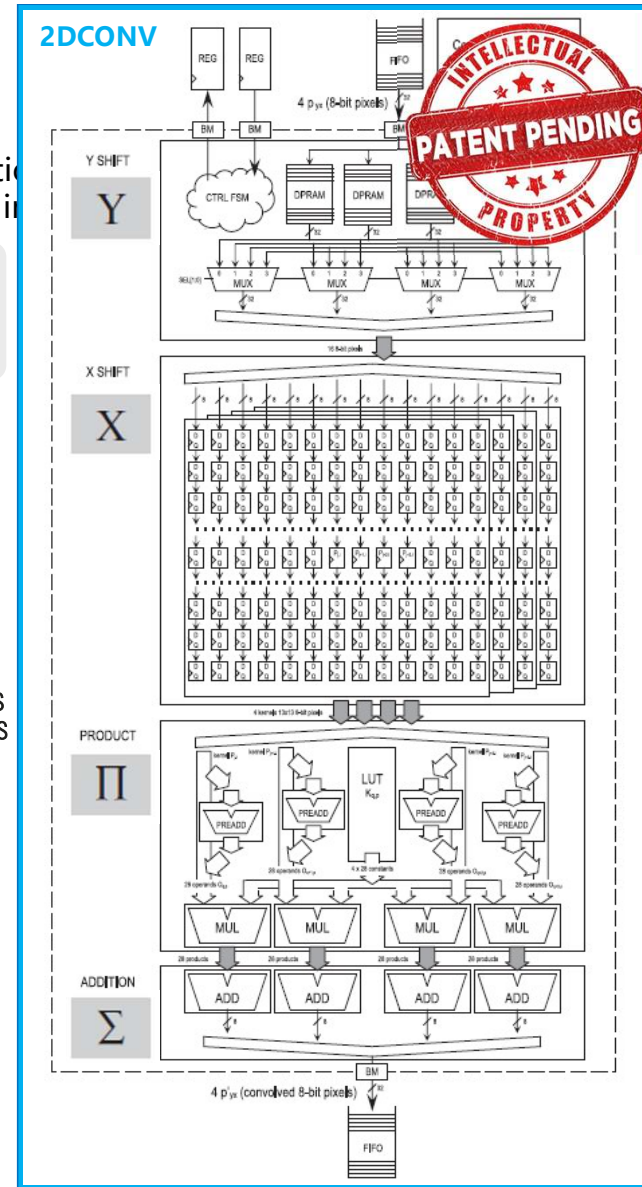
3.3 INNOVATION 09 : APP – Applications embedded in HWAs

- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution as part of a convolutional neural network (CNN) HW engine used in image processing or i



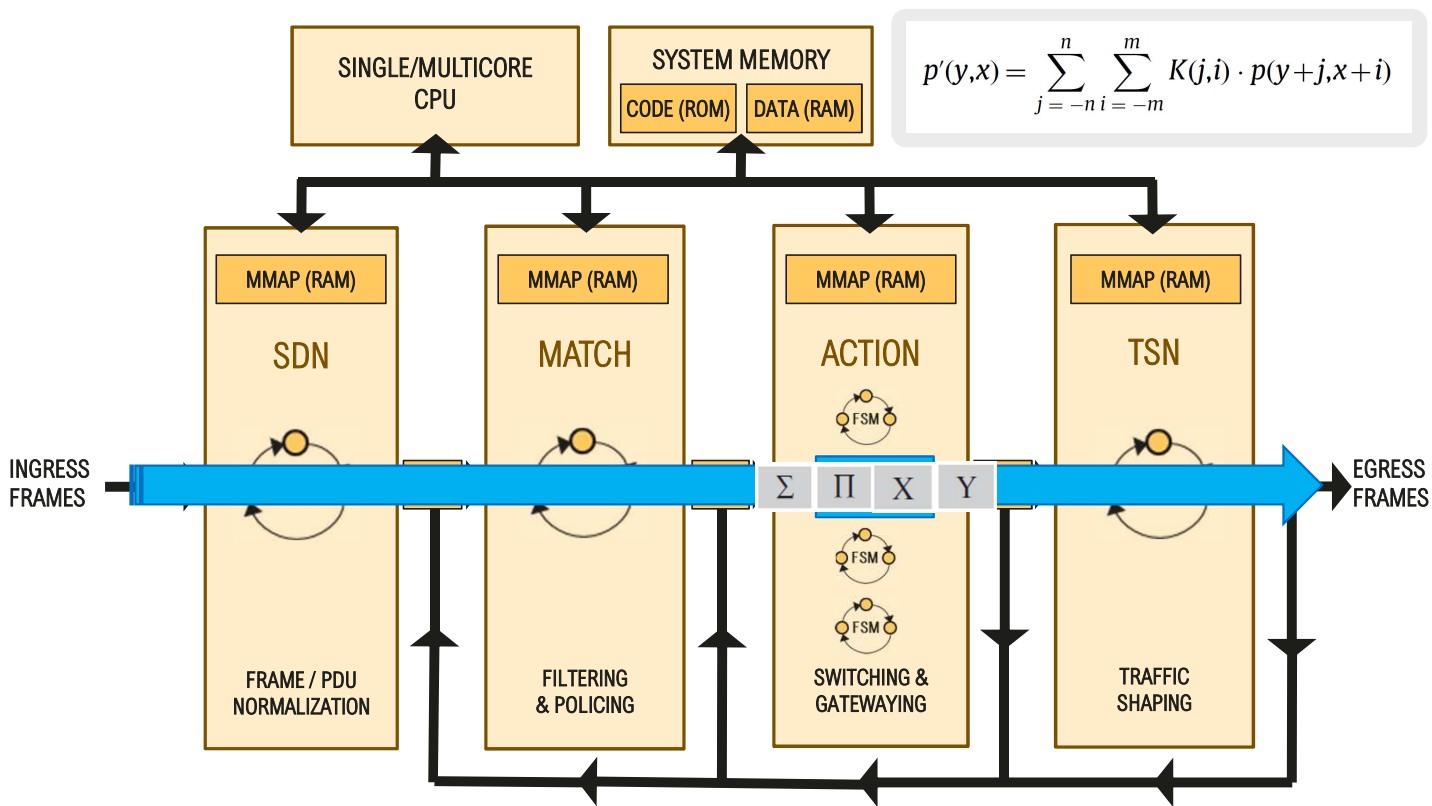
$$p'(y,x) = \sum_{j=-n}^n \sum_{i=-m}^m K(j,i) \cdot p(y+j,x+i)$$

Rationale: Effective spatial processing doing it inline and in-memory

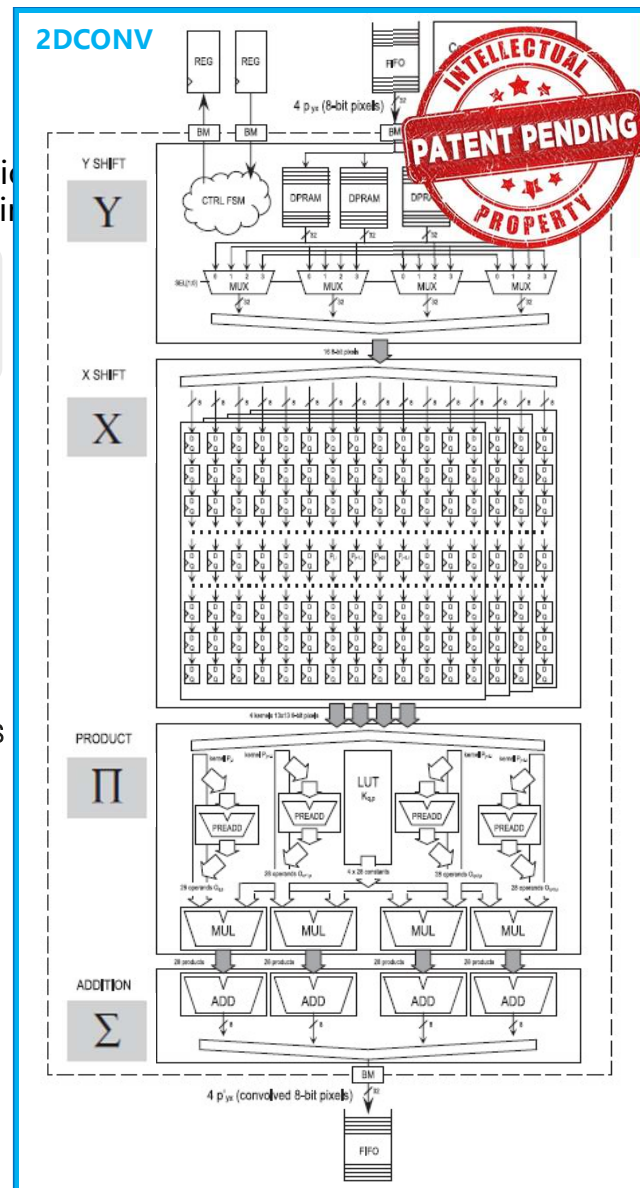


3.3 INNOVATION 09 : APP – Applications embedded in HWAs

- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution as part of a convolutional neural network (CNN) HW engine used in image processing or i



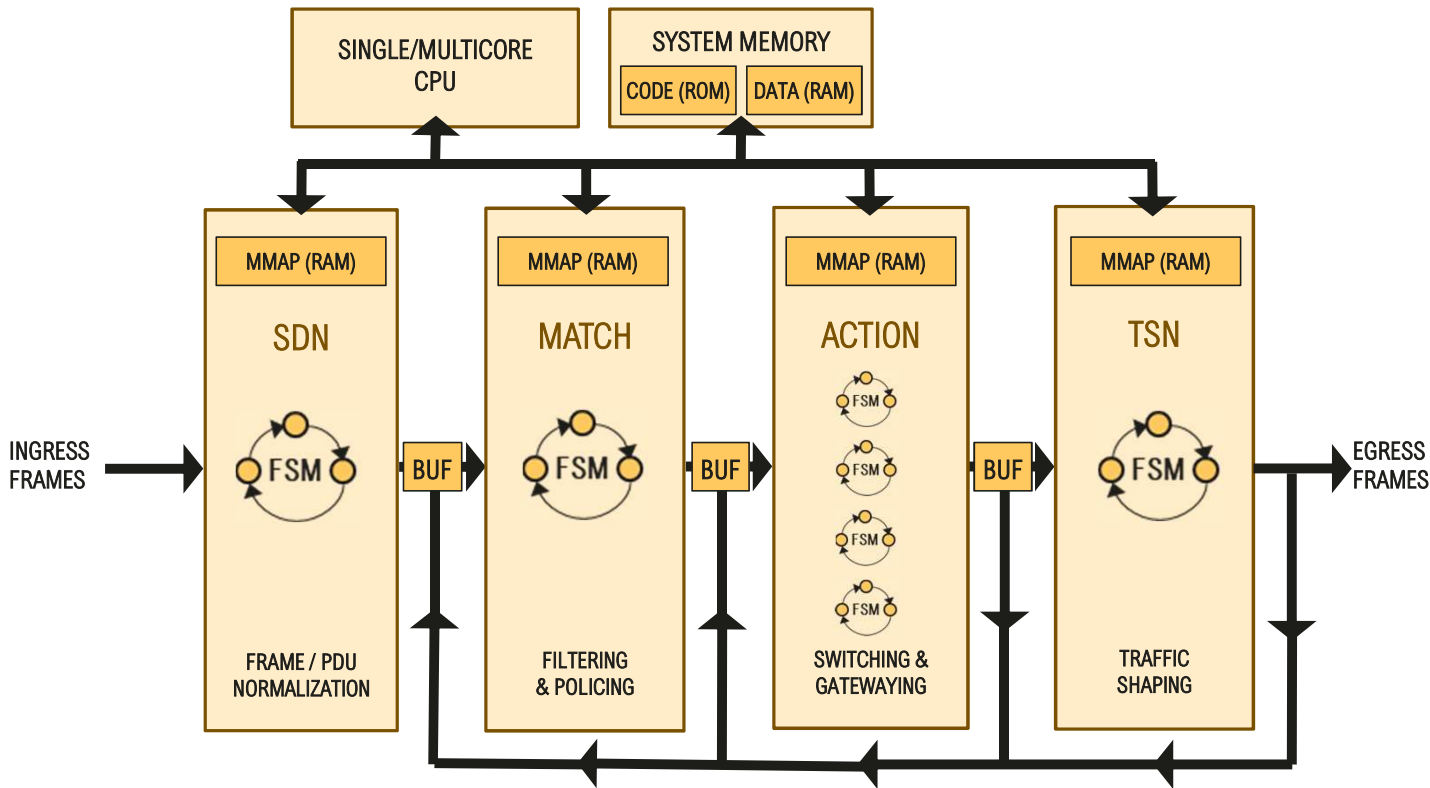
Rationale: Effective spatial processing doing it inline and in-memory





3.3 INNOVATION 09 : APP – Applications embedded in HWAs

- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution (2DCONV) HW engine as part of a convolutional neural network (CNN) HW engine used in image processing or intrusion detection system

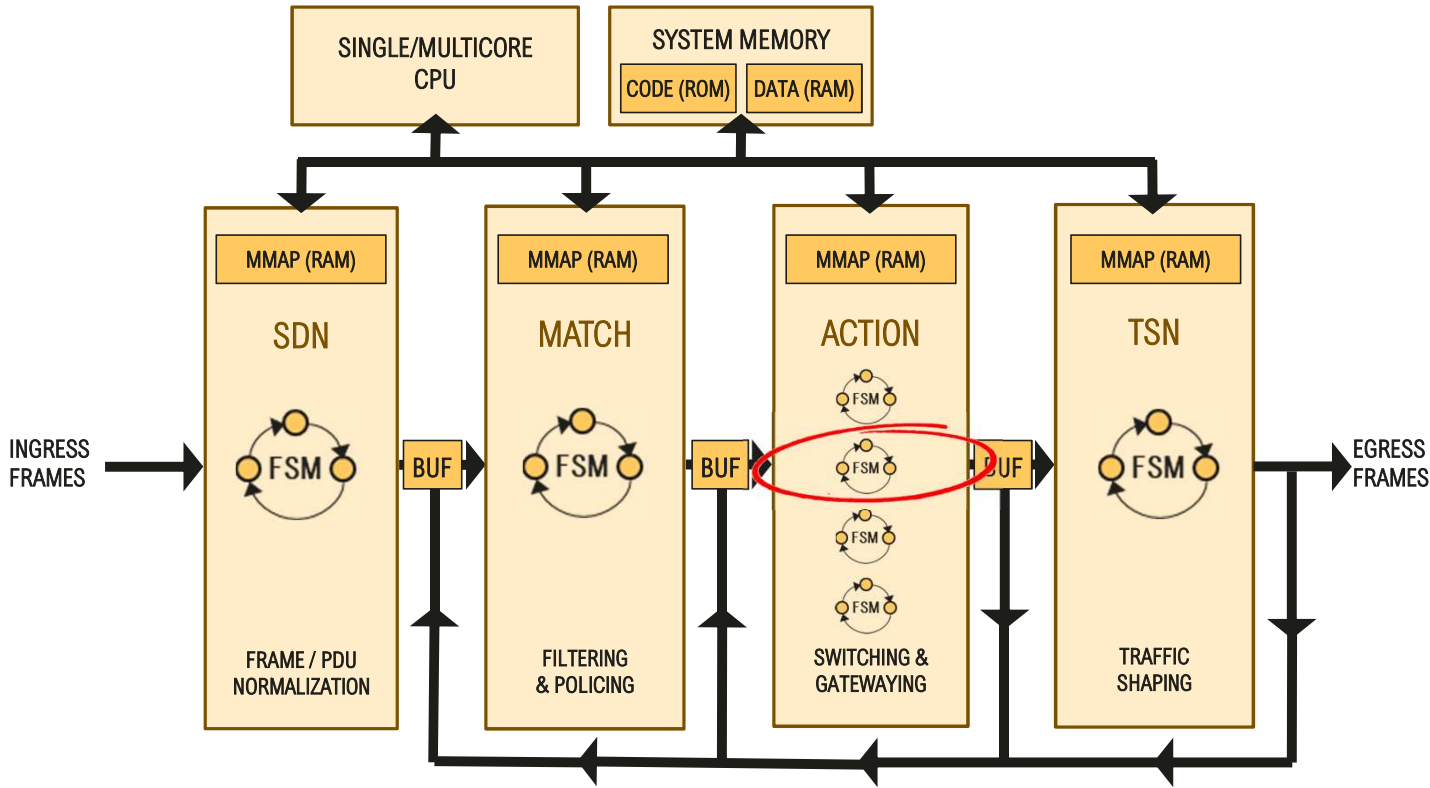


Rationale: Effective spatial processing doing it inline and in-memory



3.3 INNOVATION 09 : APP – Applications embedded in HWAs

- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution (2D CONV) HW engine as part of a convolutional neural network (CNN) HW engine used in image processing or intrusion detection system

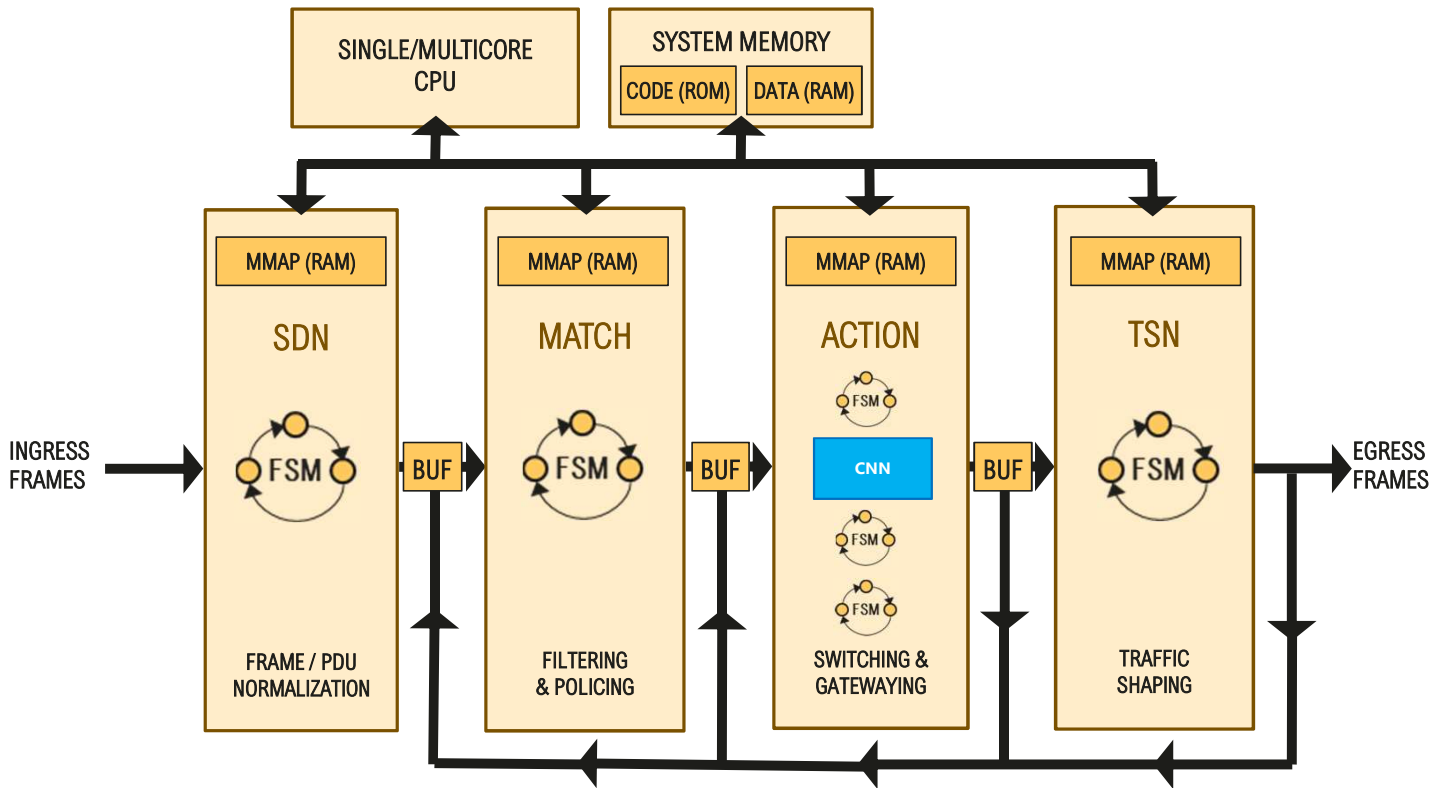


Rationale: Effective spatial processing doing it inline and in-memory



3.3 INNOVATION 09 : APP – Applications embedded in HWAs

- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution (2D CONV) HW engine as part of a convolutional neural network (CNN) HW engine used in image processing or intrusion detection system

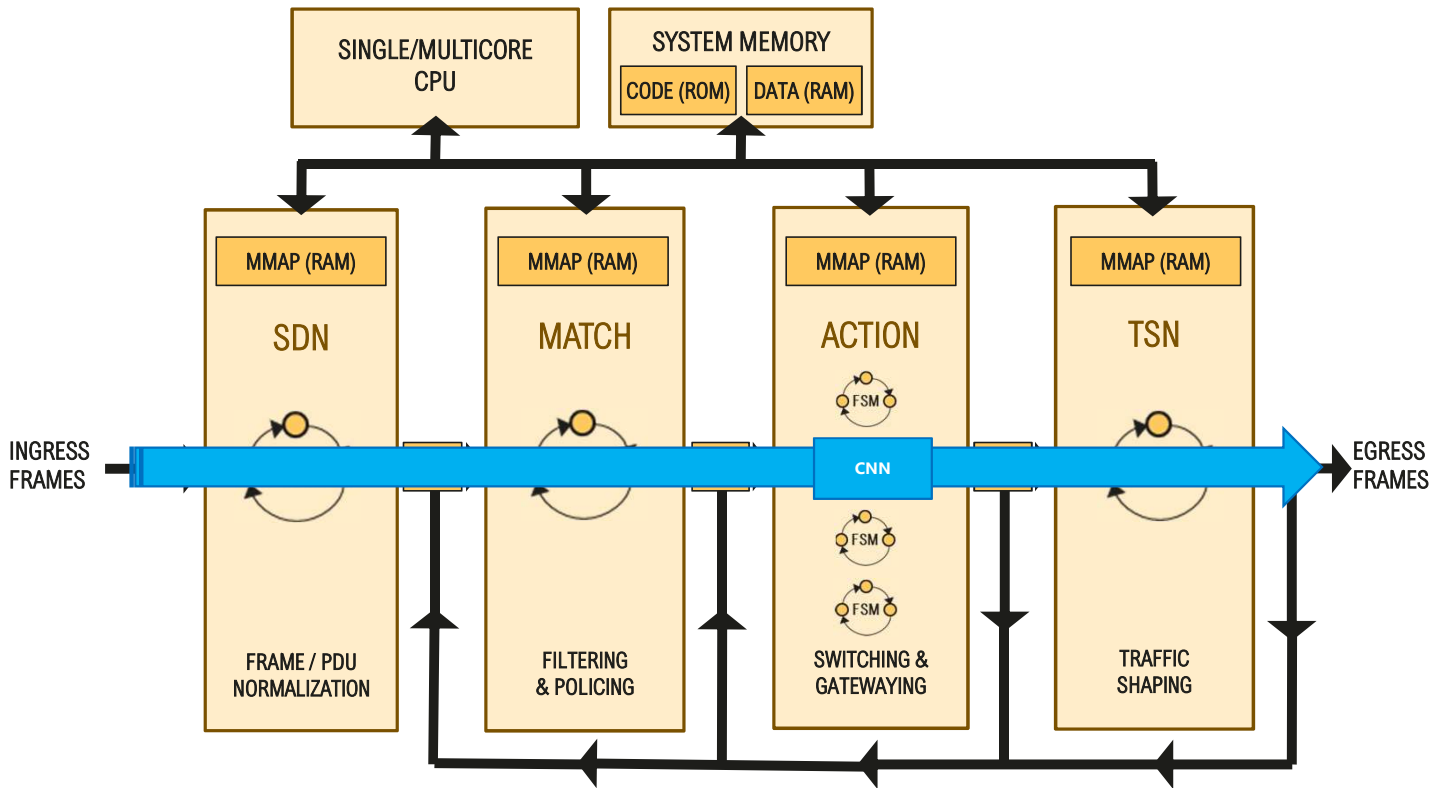


Rationale: Effective spatial processing doing it inline and in-memory



3.3 INNOVATION 09 : APP – Applications embedded in HWAs

- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution (2D CONV) HW engine as part of a convolutional neural network (CNN) HW engine used in image processing or intrusion detection system

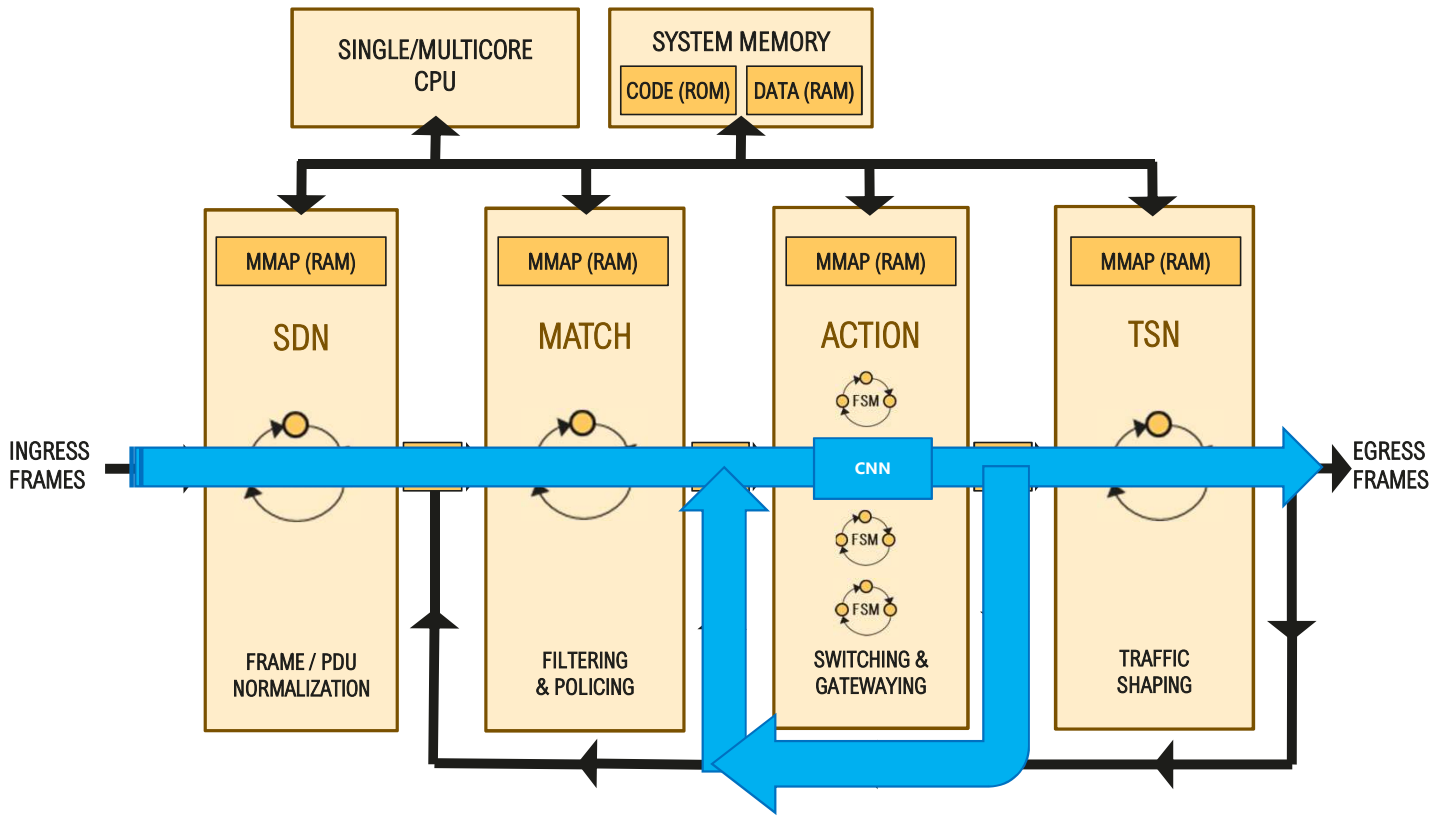


Rationale: Effective spatial processing doing it inline and in-memory



3.3 INNOVATION 09 : APP – Applications embedded in HWAs

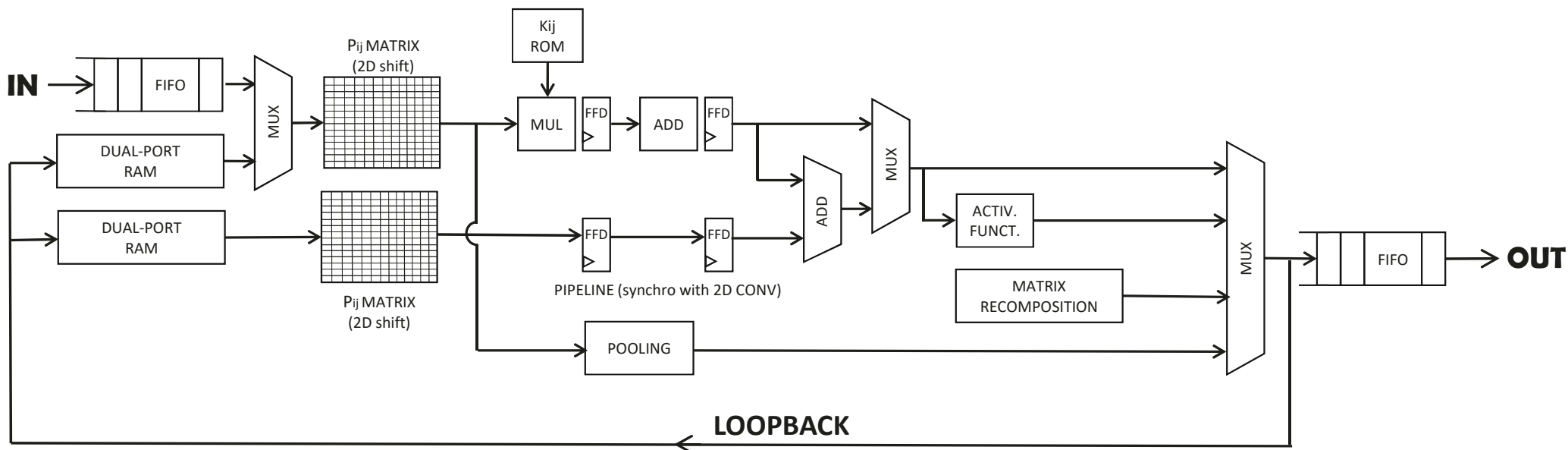
- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution (2D CONV) HW engine as part of a convolutional neural network (CNN) HW engine used in image processing or intrusion detection system



Rationale: Effective spatial processing doing it inline and in-memory

3.3 INNOVATION 09 : APP – Applications embedded in HWAs

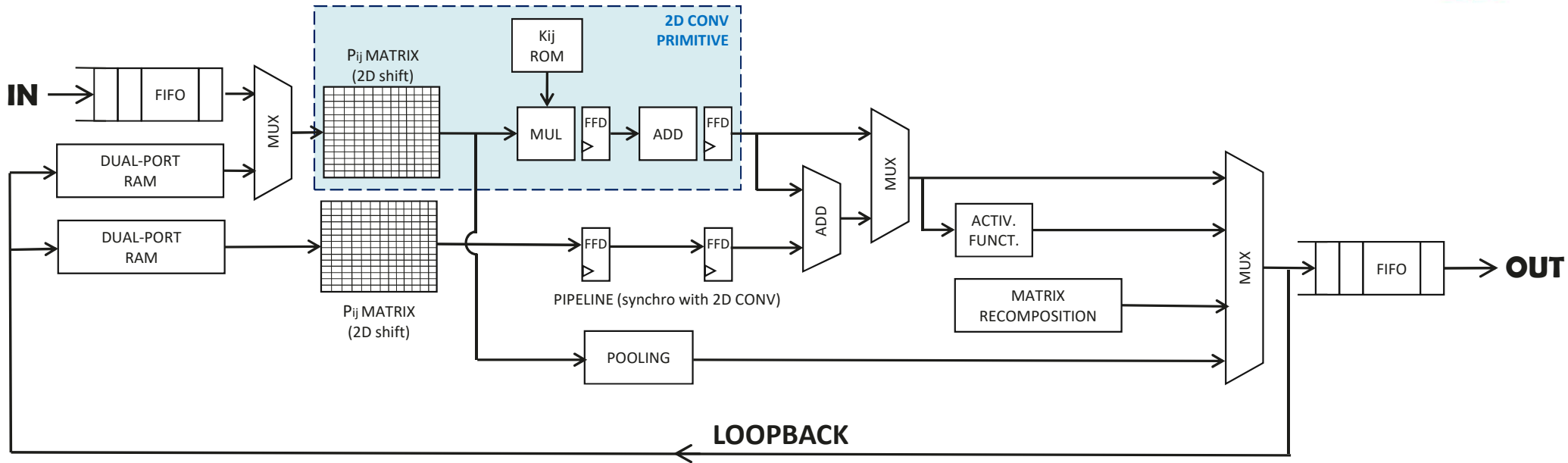
- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution (2DCONV) HW engine as part of a convolutional neural network (CNN) HW engine used in image processing or intrusion detection system





3.3 INNOVATION 09 : APP – Applications embedded in HWAs

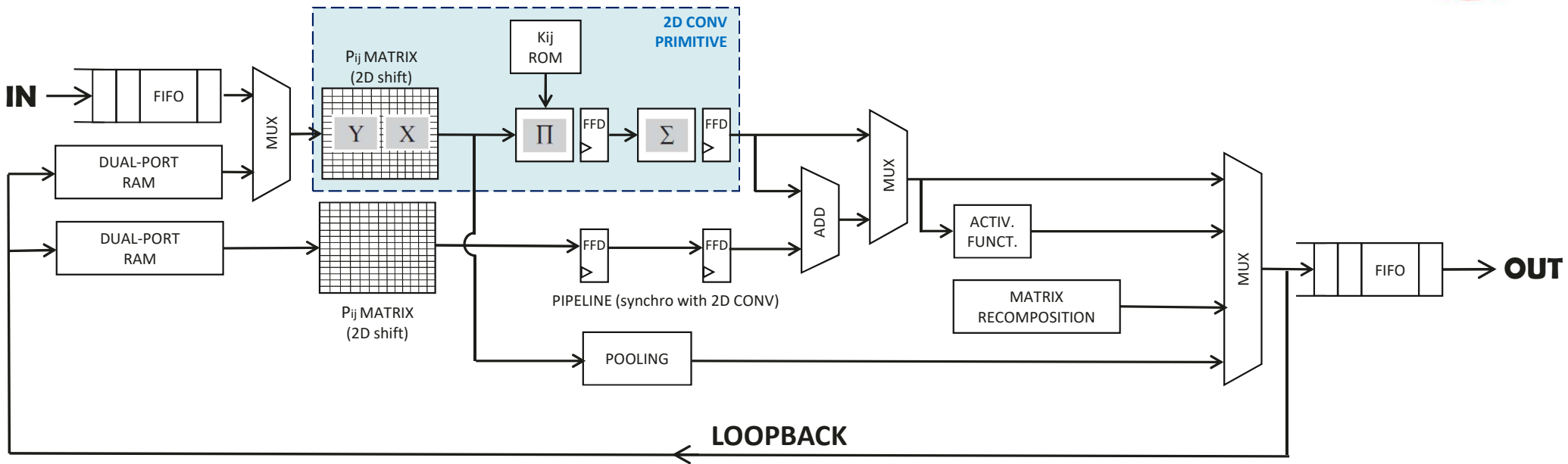
- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution (2DCONV) HW engine as part of a convolutional neural network (CNN) HW engine used in image processing or intrusion detection system





3.3 INNOVATION 09 : APP – Applications embedded in HWAs

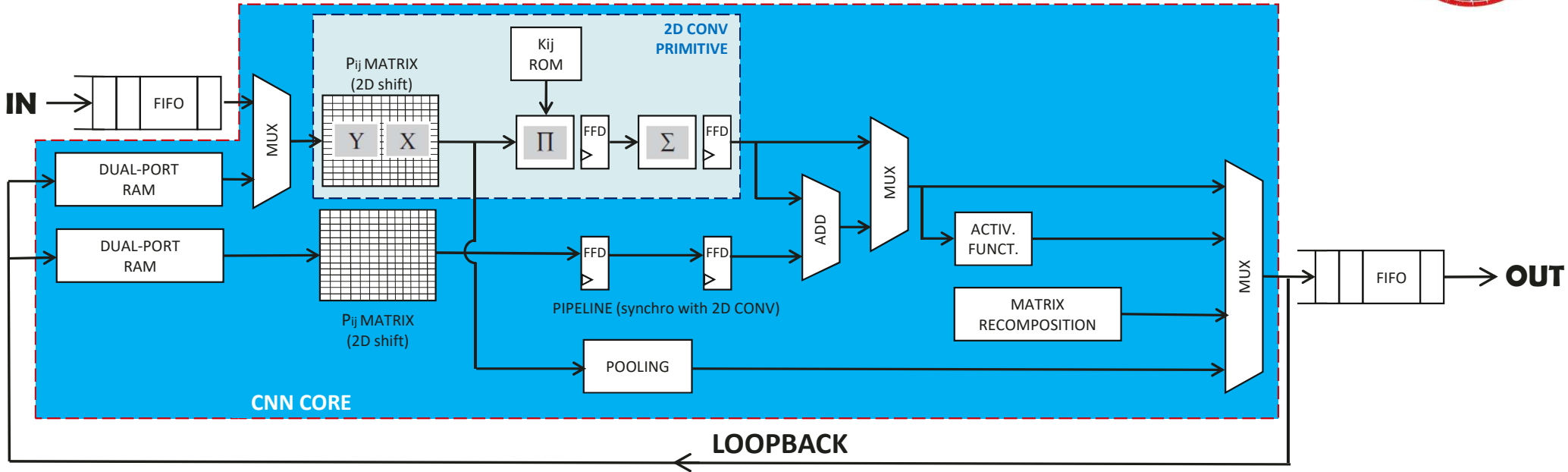
- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution (2DCONV) HW engine as part of a convolutional neural network (CNN) HW engine used in image processing or intrusion detection system





3.3 INNOVATION 09 : APP – Applications embedded in HWAs

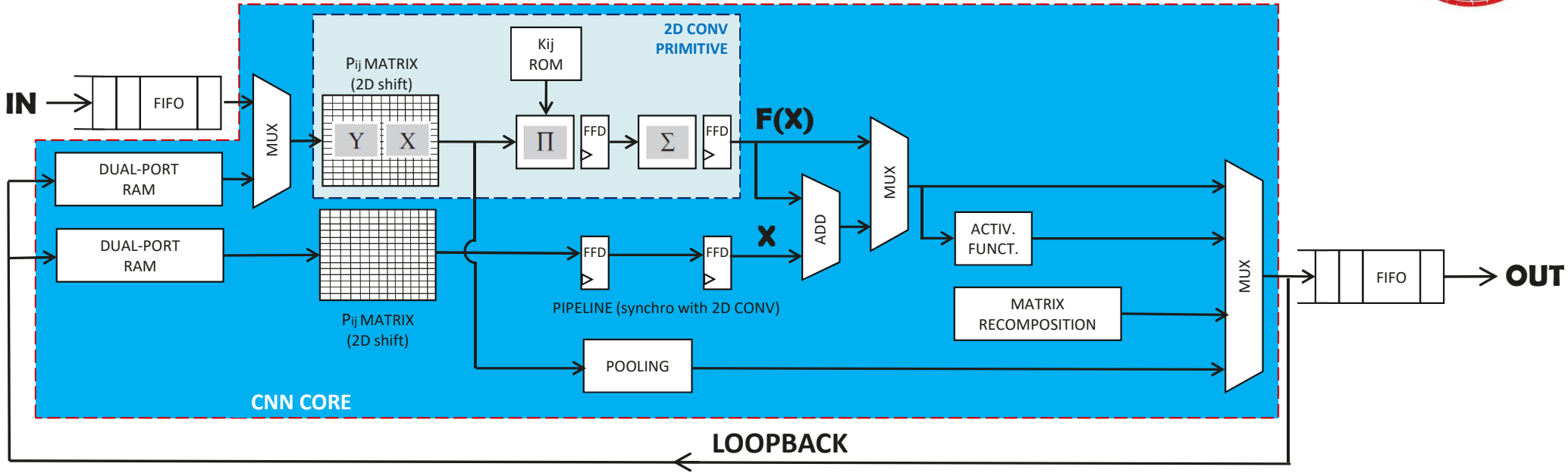
- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution (2DCONV) HW engine as part of a convolutional neural network (CNN) HW engine used in image processing or intrusion detection system



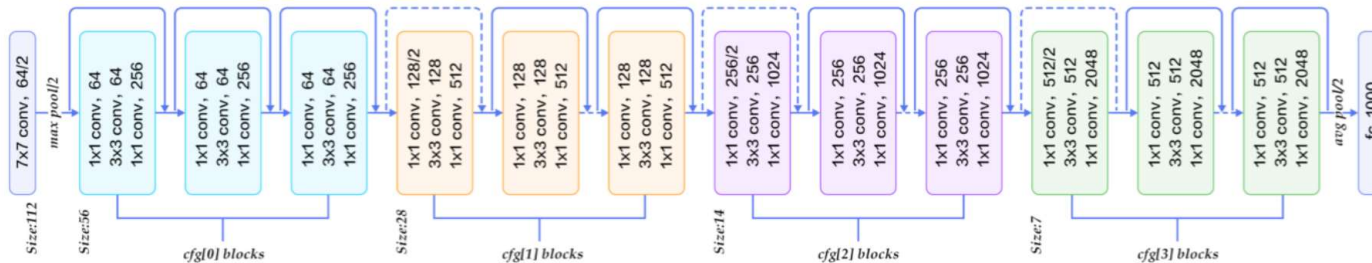


3.3 INNOVATION 09 : APP – Applications embedded in HWAs

- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution (2DCONV) HW engine as part of a convolutional neural network (CNN) HW engine used in image processing or intrusion detection system



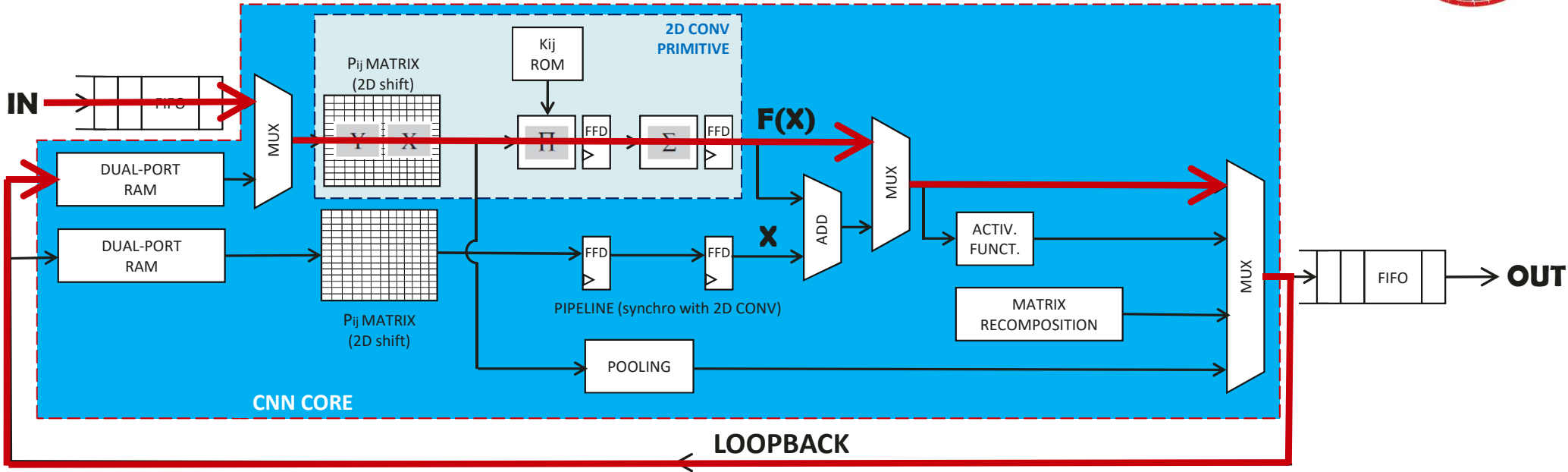
Example: ResNet152 CNN



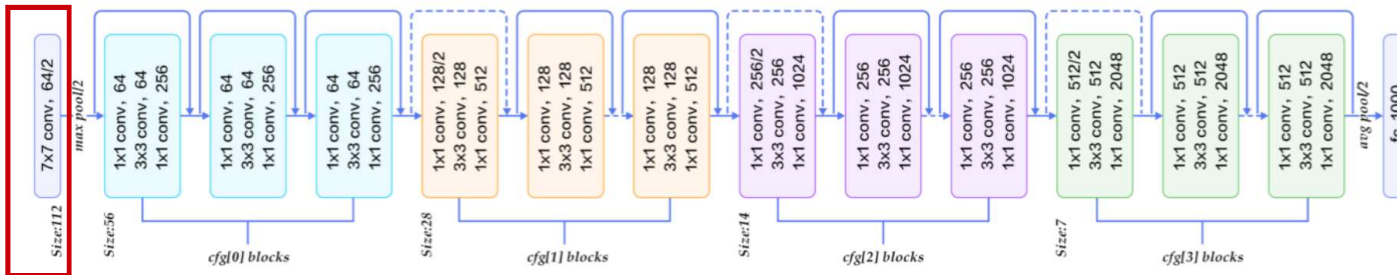


3.3 INNOVATION 09 : APP – Applications embedded in HWAs

- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution (2DCONV) HW engine as part of a convolutional neural network (CNN) HW engine used in image processing or intrusion detection system



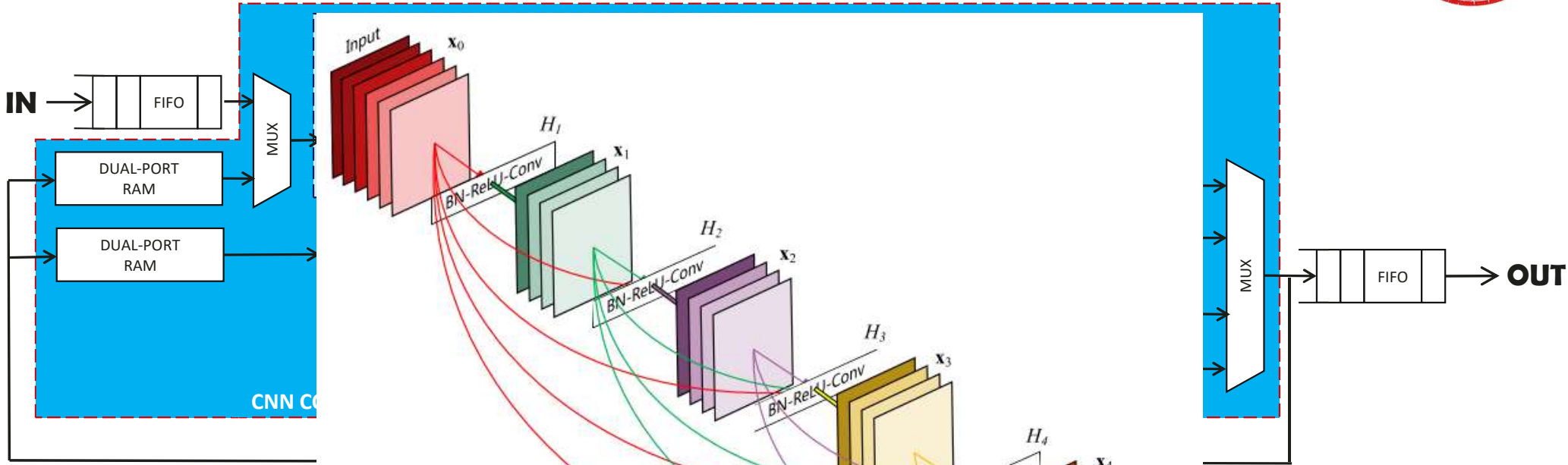
Example: ResNet152 CNN



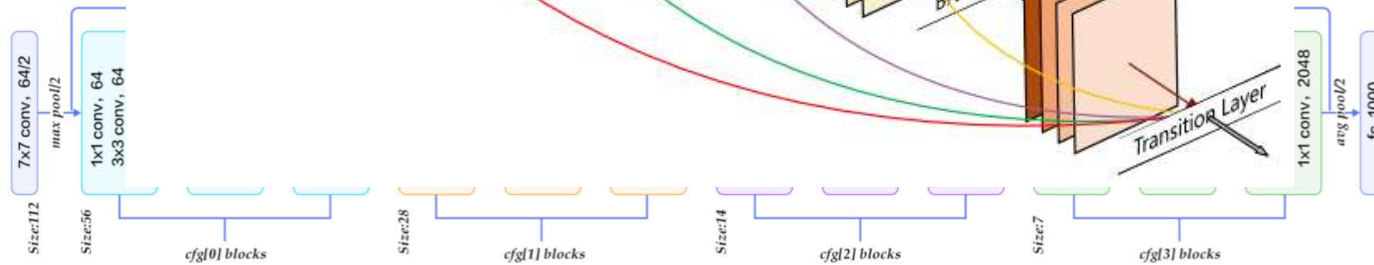


3.3 INNOVATION 09 : APP – Applications embedded in HWAs

- It is possible to design custom HWAs for specific applications, e.g. instance a 2D convolution (2D CONV) HW engine as part of a convolutional neural network (CNN) HW engine used in image processing or intrusion detection system



Example: ResNet152 CNN



3.3 INNOVATION : RPR – Run-Time Partial Reconfiguration

Zonal Gateway Controllers demand certain functionalities that shall be **adapted at run-time** based on specific environmental conditions or events triggered at real-time. Our **SoC modular architecture** enables it. Some use cases are:

- Access Control List (ACL): Add/Remove/Change ACL rules on the fly, e.g. system update at run-time
- Stateful Firewall: Add/Remove/Change firewall rules and state-based FSM at run-time, e.g. response to traffic congestion
- Network Intrusion Detection System (NIDS): Add/Remove/Change Intrusion Detection rules on the fly in response to an attack



Rationale: Self-adaptive response at run-time, i.e., on the fly, without interrupting the operation of the cyber-physical system

3.3 INNOVATION 10 : RPR – Run-Time Partial Reconfiguration

Zonal Gateway Controllers demand certain functionalities that shall be **adapted at run-time** based on specific environmental conditions or events triggered at real-time. Our **SoC modular architecture** enables it. Some use cases are:

- Access Control List (ACL): Add/Remove/Change ACL rules on the fly, e.g. system update at run-time
- Stateful Firewall: Add/Remove/Change firewall rules and state-based FSM at run-time, e.g. response to traffic congestion
- Network Intrusion Detection System (NIDS): Add/Remove/Change Intrusion Detection rules on the fly in response to an attack

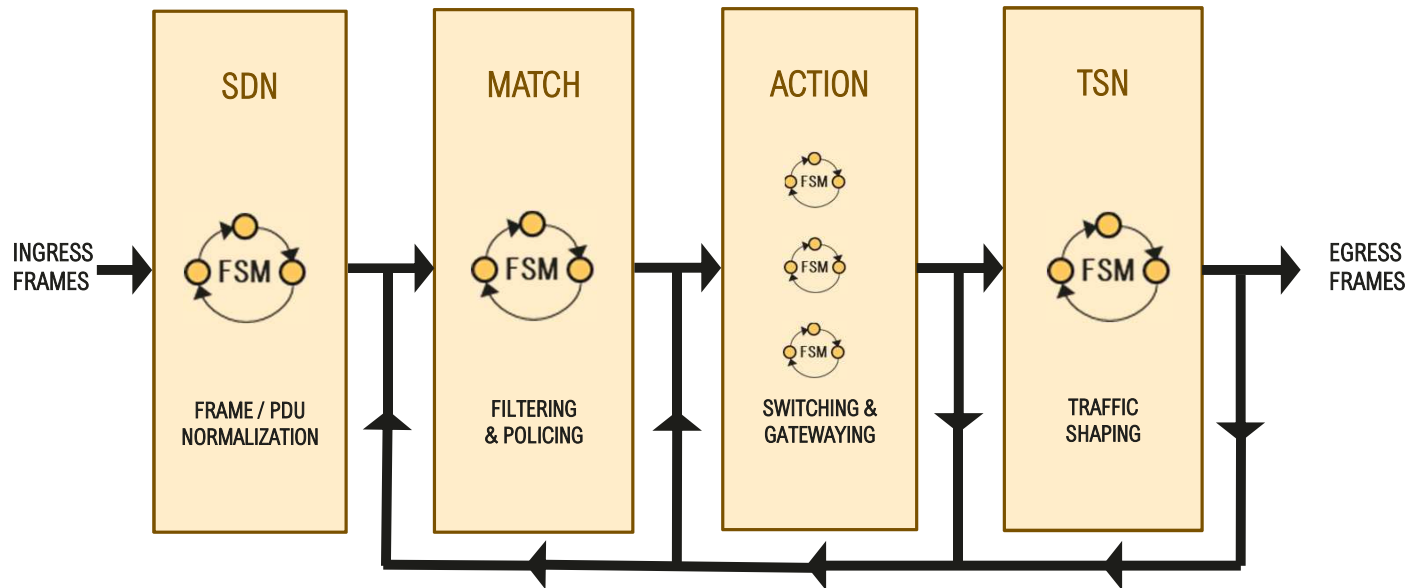


Rationale: Self-adaptive response at run-time, i.e., on the fly, without interrupting the operation of the cyber-physical system

3.3 INNOVATION 10 : RPR – Run-Time Partial Reconfiguration

Zonal Gateway Controllers demand certain functionalities that shall be **adapted at run-time** based on specific environmental conditions or events triggered at real-time. Our **SoC modular architecture** enables it. Some use cases are:

- Access Control List (ACL): Add/Remove/Change ACL rules on the fly, e.g. system update at run-time
- Stateful Firewall: Add/Remove/Change firewall rules and state-based FSM at run-time, e.g. response to traffic congestion
- Network Intrusion Detection System (NIDS): Add/Remove/Change Intrusion Detection rules on the fly in response to an attack

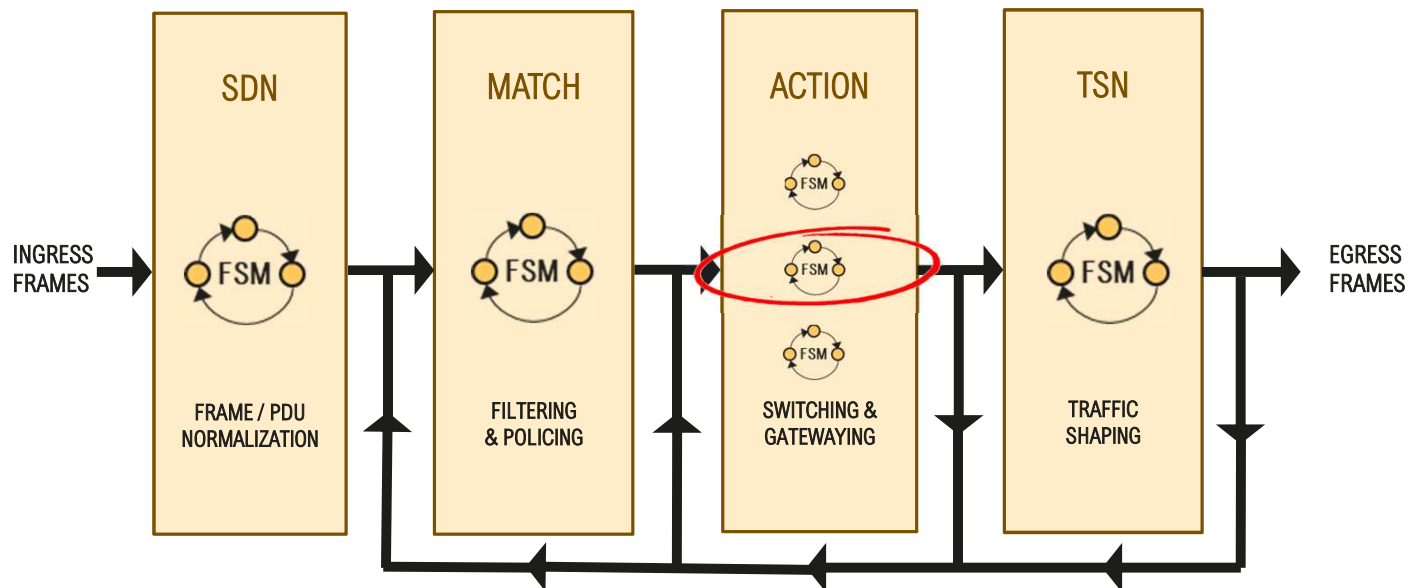


Rationale: Self-adaptive response at run-time, i.e., on the fly, without interrupting the operation of the cyber-physical system

3.3 INNOVATION 10 : RPR – Run-Time Partial Reconfiguration

Zonal Gateway Controllers demand certain functionalities that shall be **adapted at run-time** based on specific environmental conditions or events triggered at real-time. Our **SoC modular architecture** enables it. Some use cases are:

- Access Control List (ACL): Add/Remove/Change ACL rules on the fly, e.g. system update at run-time
- Stateful Firewall: Add/Remove/Change firewall rules and state-based FSM at run-time, e.g. response to traffic congestion
- Network Intrusion Detection System (NIDS): Add/Remove/Change Intrusion Detection rules on the fly in response to an attack

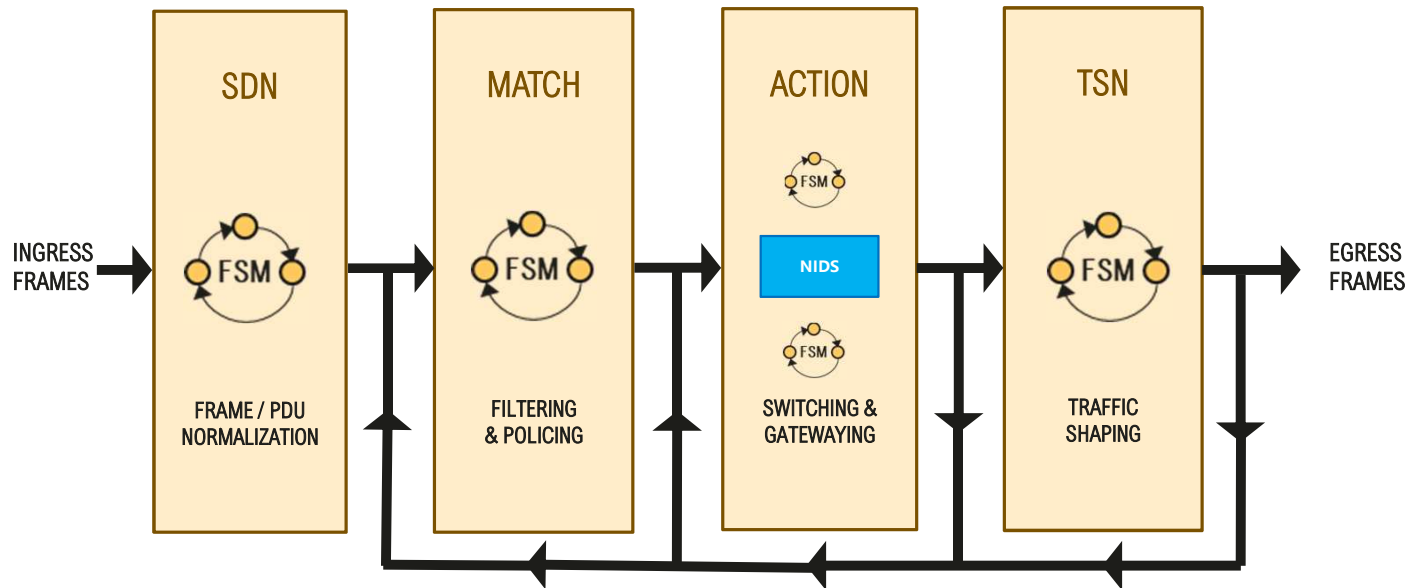


Rationale: Self-adaptive response at run-time, i.e., on the fly, without interrupting the operation of the cyber-physical system

3.3 INNOVATION 10 : RPR – Run-Time Partial Reconfiguration

Zonal Gateway Controllers demand certain functionalities that shall be **adapted at run-time** based on specific environmental conditions or events triggered at real-time. Our **SoC modular architecture** enables it. Some use cases are:

- Access Control List (ACL): Add/Remove/Change ACL rules on the fly, e.g. system update at run-time
- Stateful Firewall: Add/Remove/Change firewall rules and state-based FSM at run-time, e.g. response to traffic congestion
- Network Intrusion Detection System (NIDS): Add/Remove/Change Intrusion Detection rules on the fly in response to an attack

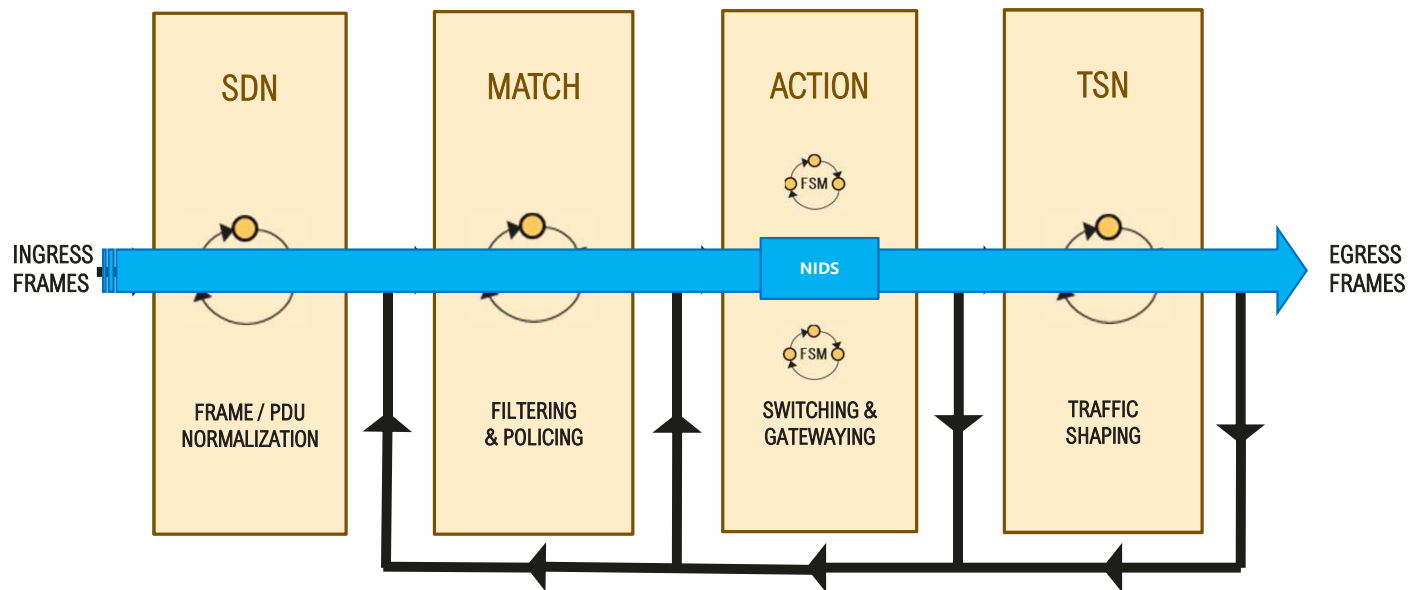


Rationale: Self-adaptive response at run-time, i.e., on the fly, without interrupting the operation of the cyber-physical system

3.3 INNOVATION 10 : RPR – Run-Time Partial Reconfiguration

Zonal Gateway Controllers demand certain functionalities that shall be **adapted at run-time** based on specific environmental conditions or events triggered at real-time. Our **SoC modular architecture** enables it. Some use cases are:

- Access Control List (ACL): Add/Remove/Change ACL rules on the fly, e.g. system update at run-time
- Stateful Firewall: Add/Remove/Change firewall rules and state-based FSM at run-time, e.g. response to traffic congestion
- Network Intrusion Detection System (NIDS): Add/Remove/Change Intrusion Detection rules on the fly in response to an attack

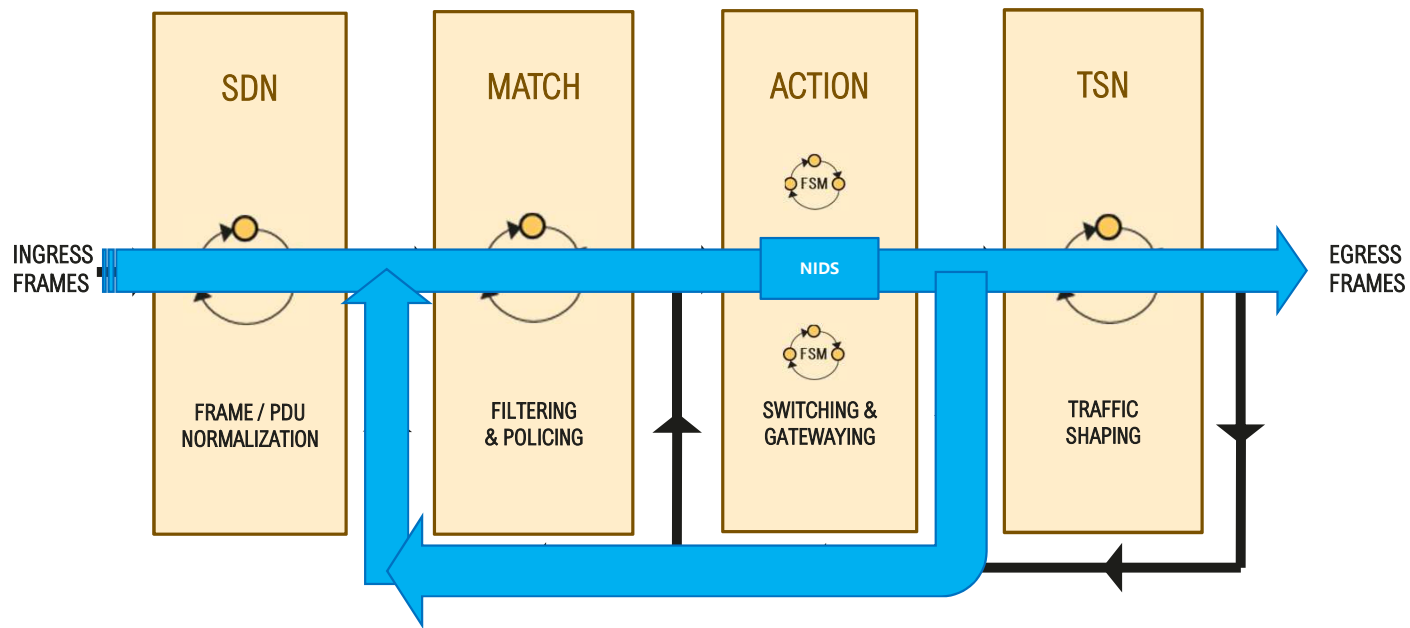


Rationale: Self-adaptive response at run-time, i.e., on the fly, without interrupting the operation of the cyber-physical system

3.3 INNOVATION 10 : RPR – Run-Time Partial Reconfiguration

Zonal Gateway Controllers demand certain functionalities that shall be **adapted at run-time** based on specific environmental conditions or events triggered at real-time. Our **SoC modular architecture** enables it. Some use cases are:

- Access Control List (ACL): Add/Remove/Change ACL rules on the fly, e.g. system update at run-time
- Stateful Firewall: Add/Remove/Change firewall rules and state-based FSM at run-time, e.g. response to traffic congestion
- Network Intrusion Detection System (NIDS): Add/Remove/Change Intrusion Detection rules on the fly in response to an attack

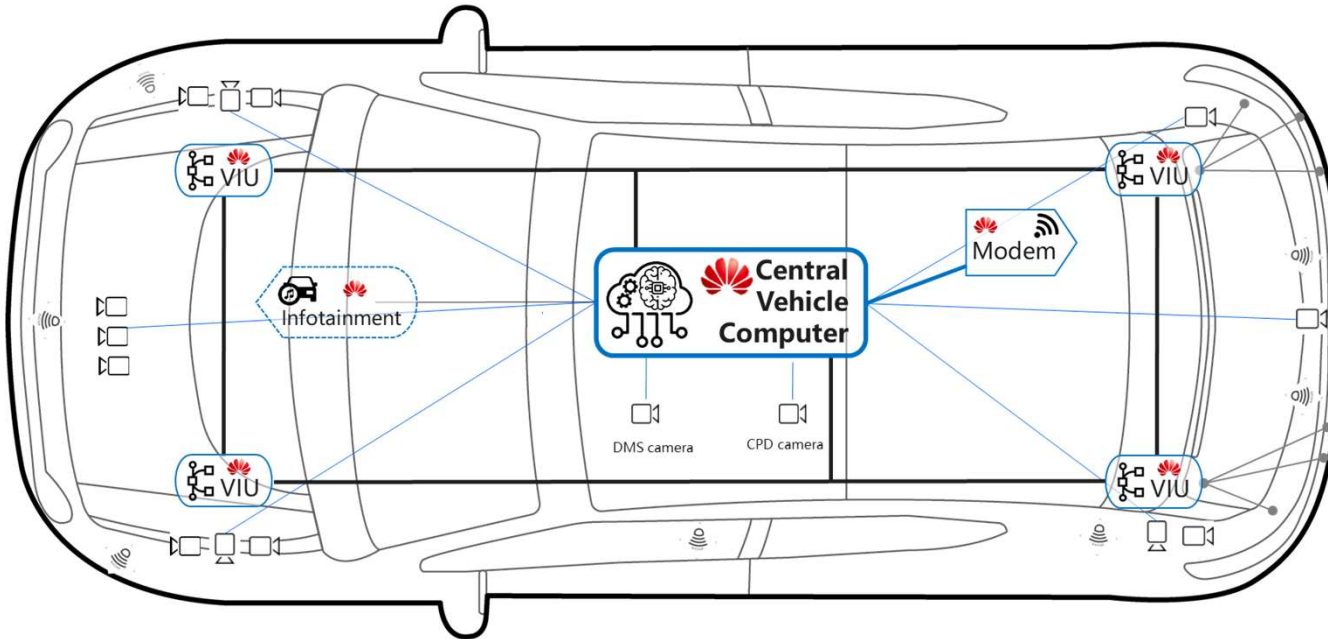


Rationale: Self-adaptive response at run-time, i.e., on the fly, without interrupting the operation of the cyber-physical system



3.3 INNOVATION 11: W2W – IVN Wired-to-Wireless HWA embedded in eGW SoC

- **Wireless technology** applied to IVN backbone infrastructure as a **redundant and diverse secondary link or channel**, complementing the primary link based on wired (either copper- or optical-based) Automotive Ethernet, could be a sensible approach by bringing many advantages to the IVN solution

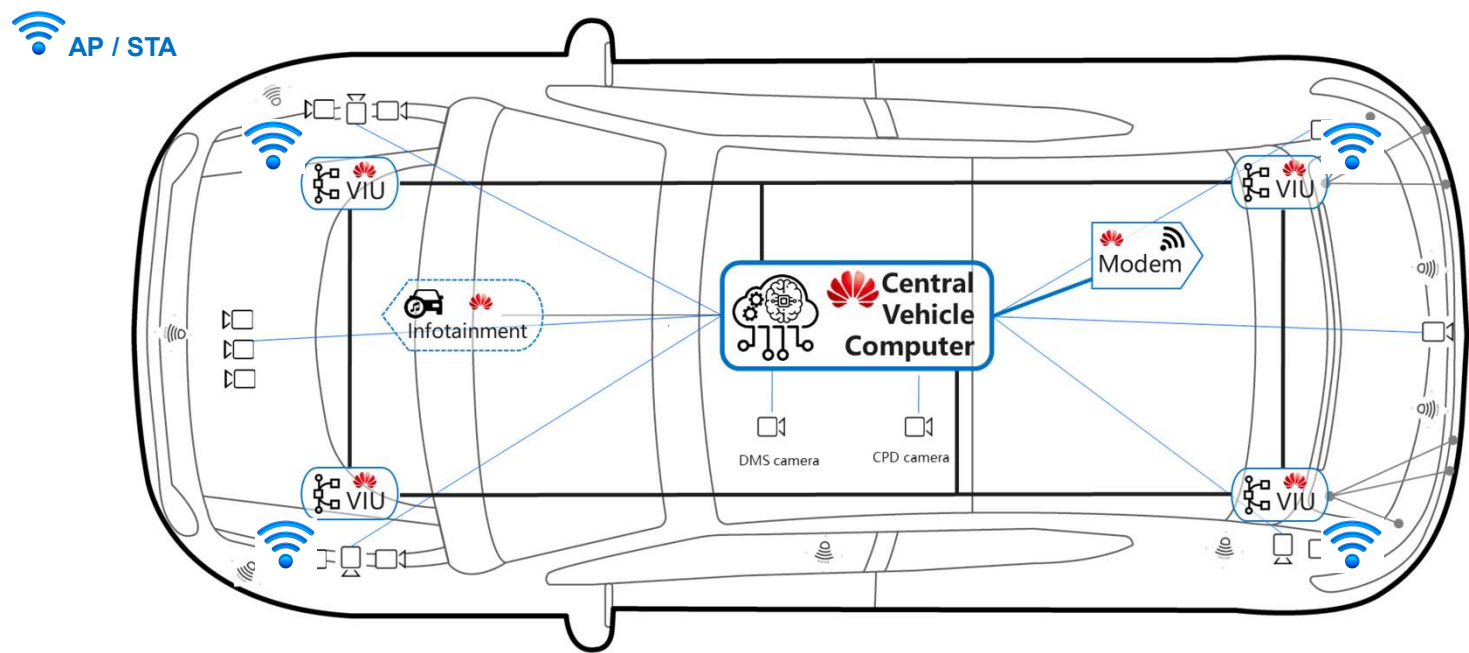


Rationale: Effective balance of high-performance, flexibility/versatility, modularity/reusability and scalability



3.3 INNOVATION 11: W2W – IVN Wired-to-Wireless HWA embedded in eGW SoC

- **Wireless technology** applied to IVN backbone infrastructure as a **redundant and diverse secondary link or channel**, complementing the primary link based on wired (either copper- or optical-based) Automotive Ethernet, could be a sensible approach by bringing many advantages to the IVN solution

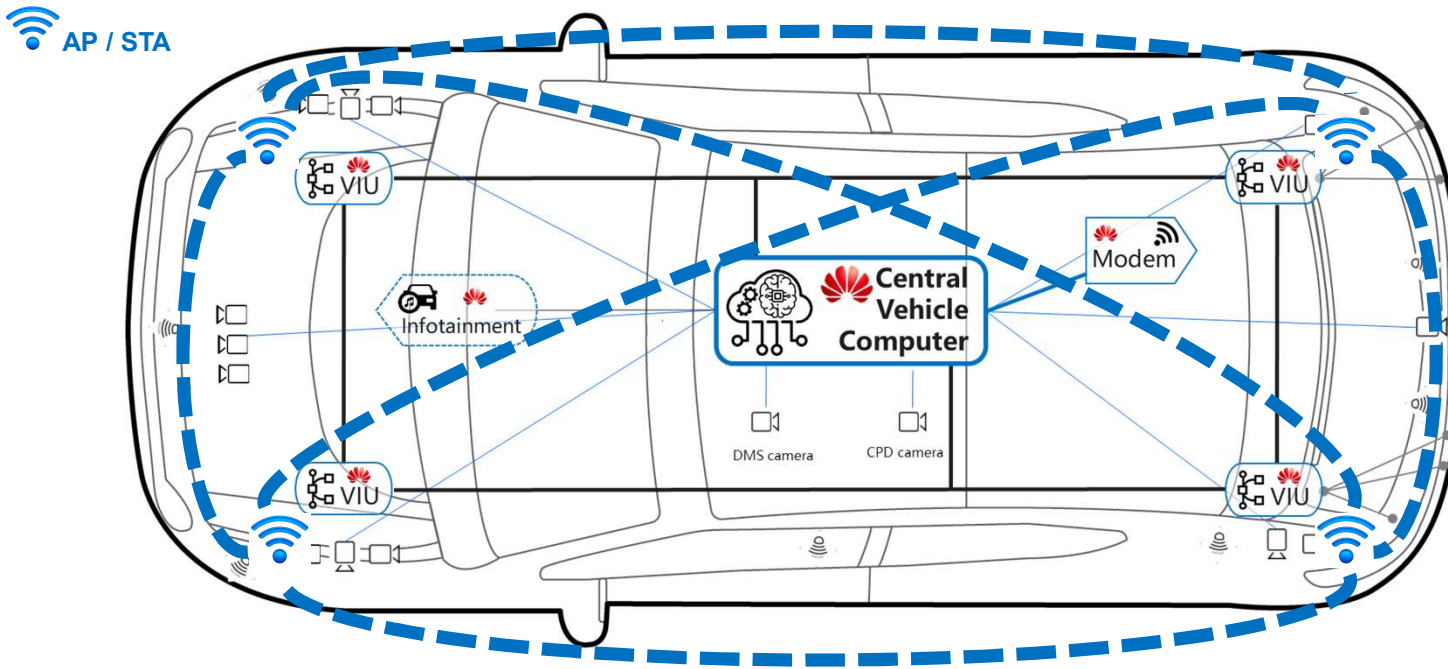


Rationale: Effective balance of high-performance, flexibility/versatility, modularity/reusability and scalability



3.3 INNOVATION 11: W2W – IVN Wired-to-Wireless HWA embedded in eGW SoC

- **Wireless technology** applied to IVN backbone infrastructure as a **redundant and diverse secondary link or channel**, complementing the primary link based on wired (either copper- or optical-based) Automotive Ethernet, could be a sensible approach by bringing many advantages to the IVN solution



Rationale: Effective balance of high-performance, flexibility/versatility, modularity/reusability and scalability

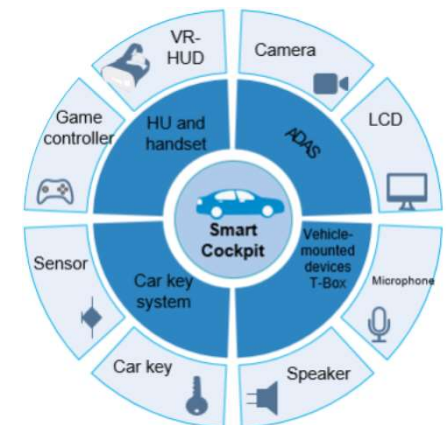
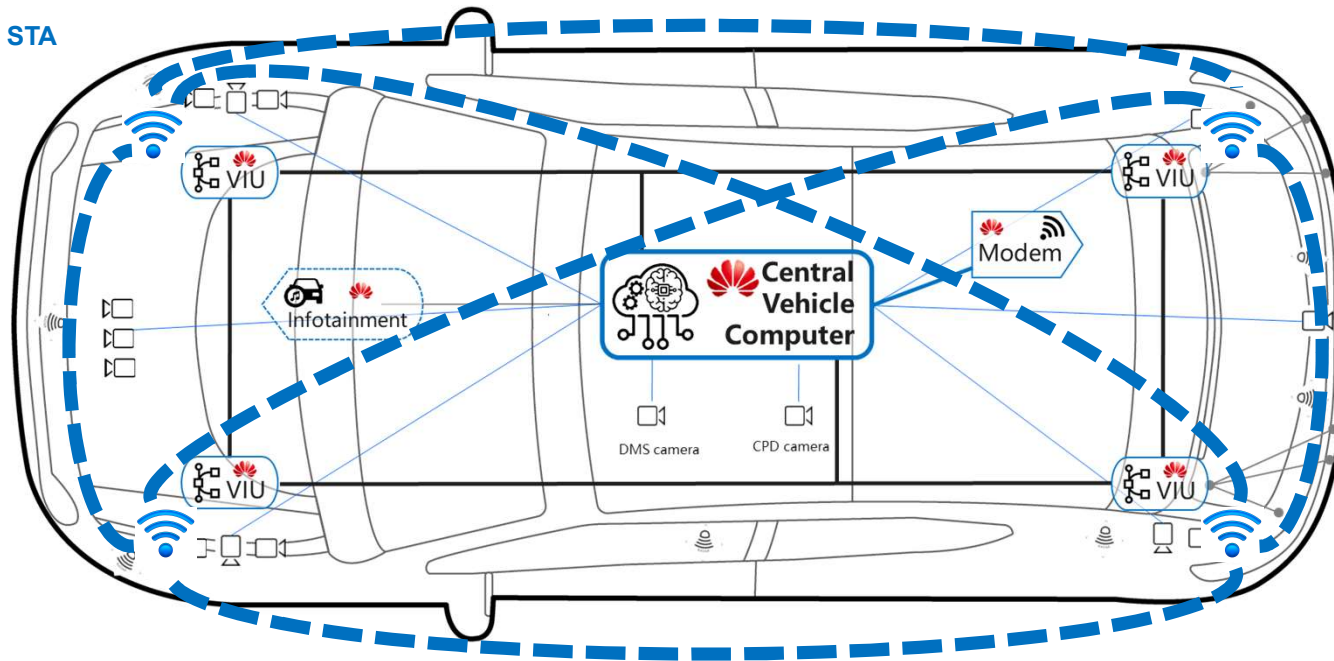
3.3 INNOVATION 11: W2W – IVN Wired-to-Wireless HWA embedded in eGW SoC



- **Wireless technology** applied to IVN backbone infrastructure as a **redundant and diverse secondary link or channel**, complementing the primary link based on wired (either copper- or optical-based) Automotive Ethernet, could be a sensible approach by bringing many advantages to the IVN solution

Trustworthy (Redundant & Diverse) Hybrid “Wired+Wireless” IVN Infrastructure by Design!

AP / STA



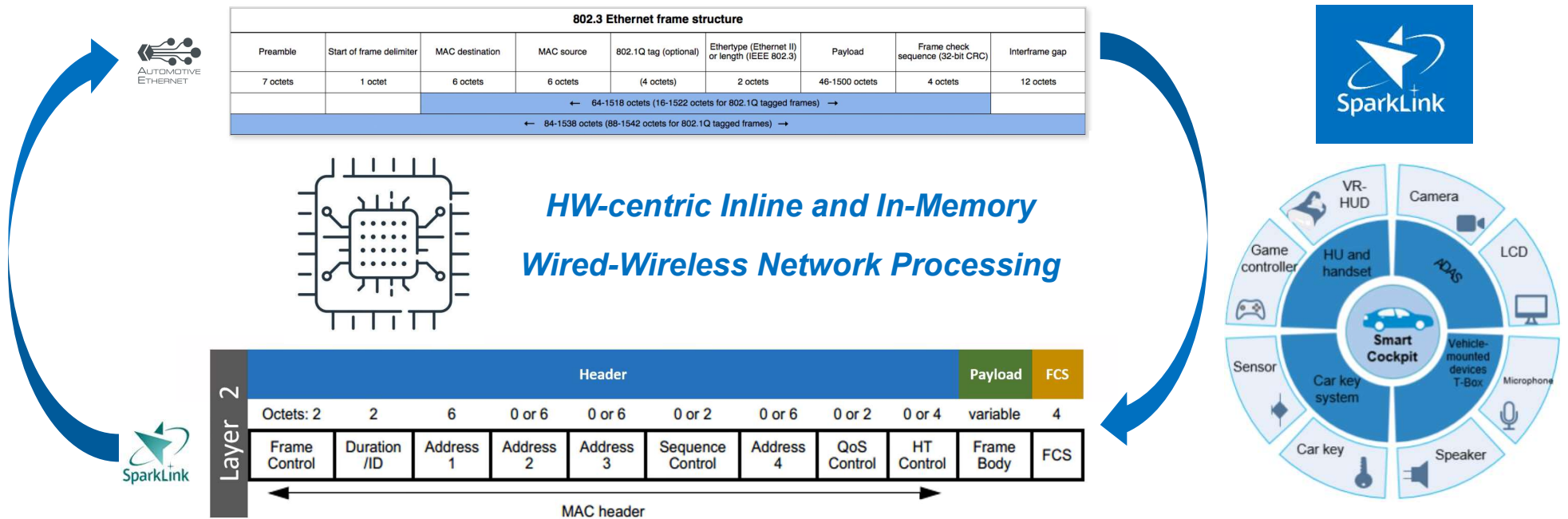
Rationale: Effective balance of high-performance, flexibility/versatility, modularity/reusability and scalability



3.3 INNOVATION 11: W2W – IVN Wired-to-Wireless HWA embedded in eGW SoC

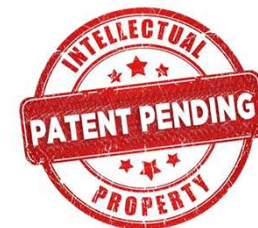
- Wireless technology applied to IVN backbone infrastructure as a **redundant and diverse secondary link or channel**, complementing the primary link based on wired (either copper- or optical-based) Automotive Ethernet, could be a sensible approach by bringing many advantages to the IVN solution

Trustworthy (Redundant & Diverse) Hybrid “Wired+Wireless” IVN Infrastructure by Design!

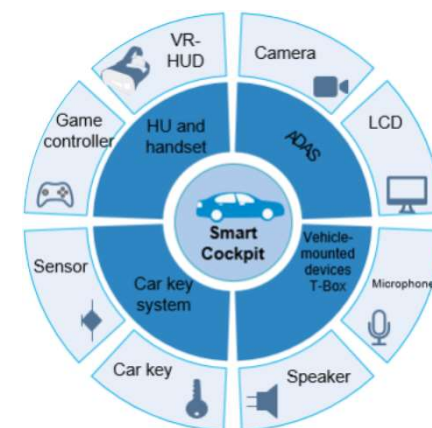
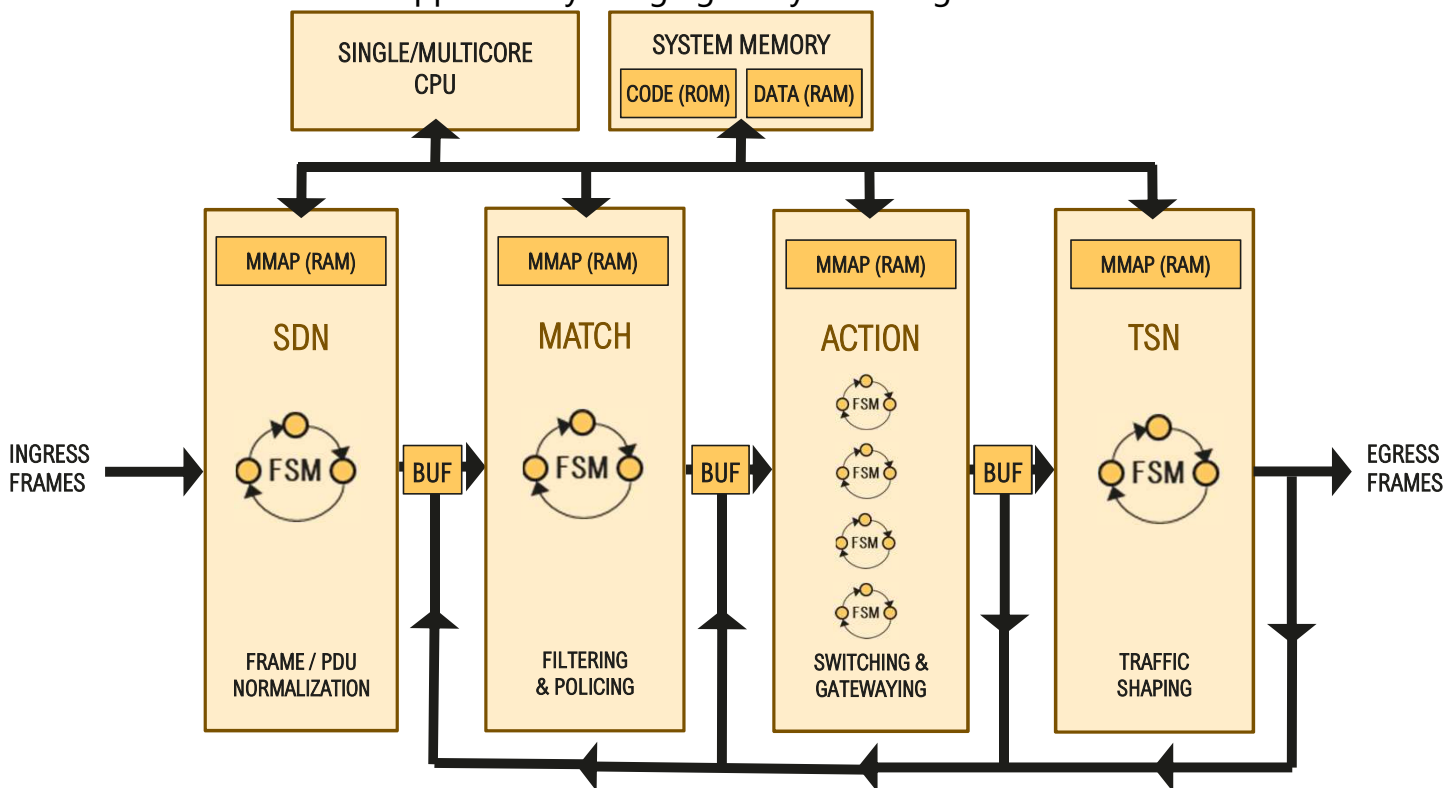


Rationale: Effective balance of high-performance, flexibility/versatility, modularity/reusability and scalability

3.3 INNOVATION 11: W2W – IVN Wired-to-Wireless HWA embedded in eGW SoC



- Wireless technology applied to IVN backbone infrastructure as a **redundant and diverse secondary link or channel**, complementing the primary link based on wired (either copper- or optical-based) Automotive Ethernet, could be a sensible approach by bringing many advantages to the IVN solution

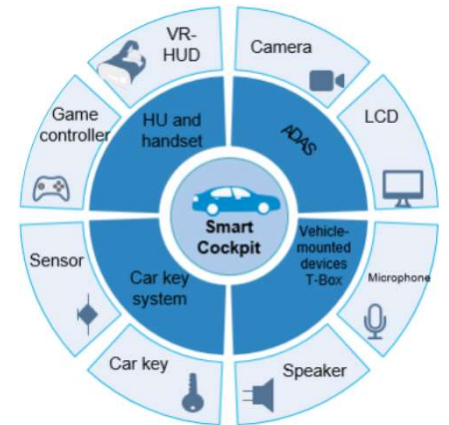
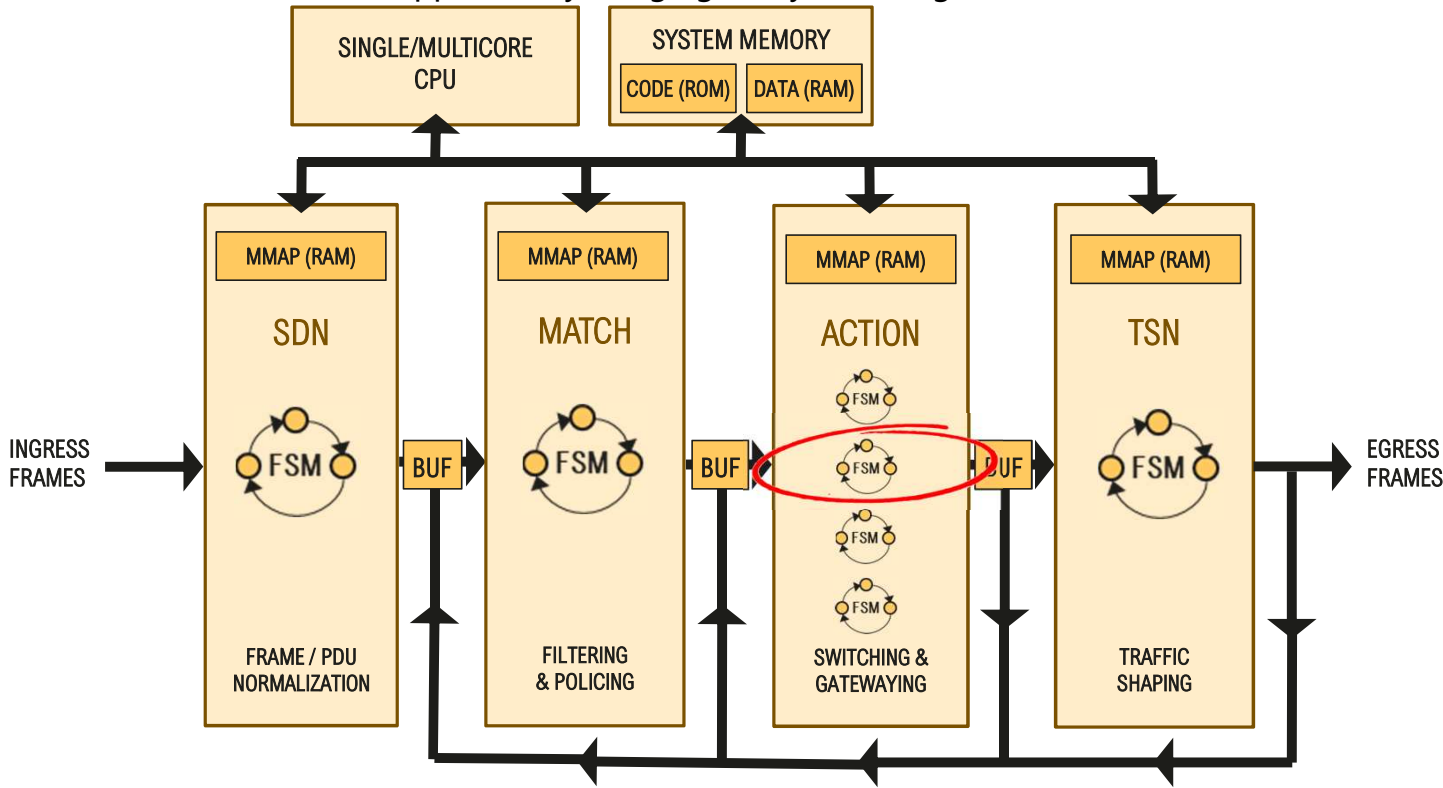


Rationale: Effective balance of high-performance, flexibility/versatility, modularity/reusability and scalability



3.3 INNOVATION 11: W2W – IVN Wired-to-Wireless HWA embedded in eGW SoC

- Wireless technology applied to IVN backbone infrastructure as a **redundant and diverse secondary link or channel**, complementing the primary link based on wired (either copper- or optical-based) Automotive Ethernet, could be a sensible approach by bringing many advantages to the IVN solution

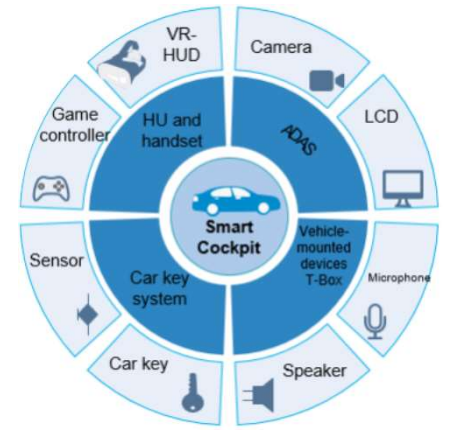
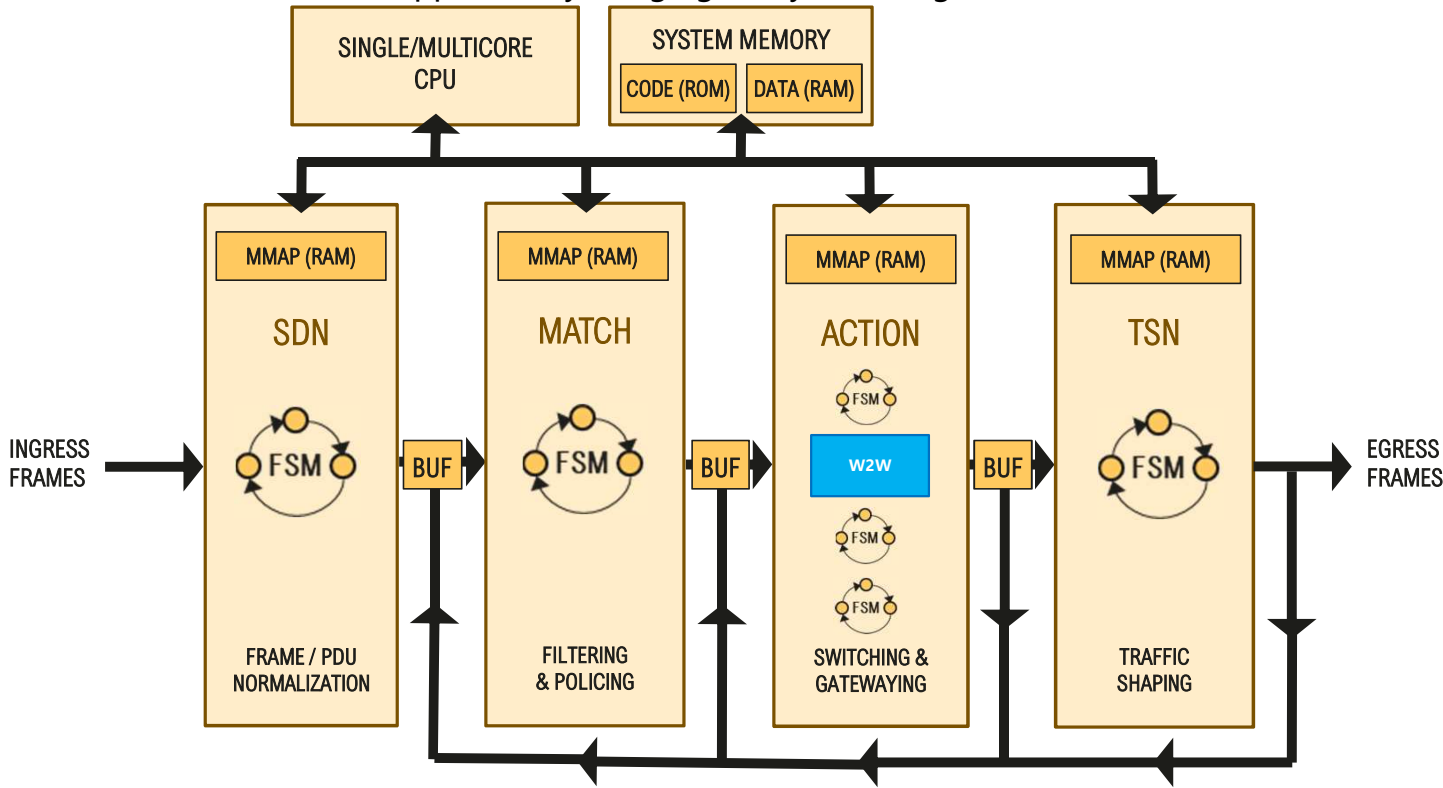


Rationale: Effective balance of high-performance, flexibility/versatility, modularity/reusability and scalability



3.3 INNOVATION 11: W2W – IVN Wired-to-Wireless HWA embedded in eGW SoC

- Wireless technology applied to IVN backbone infrastructure as a **redundant and diverse secondary link or channel**, complementing the primary link based on wired (either copper- or optical-based) Automotive Ethernet, could be a sensible approach by bringing many advantages to the IVN solution



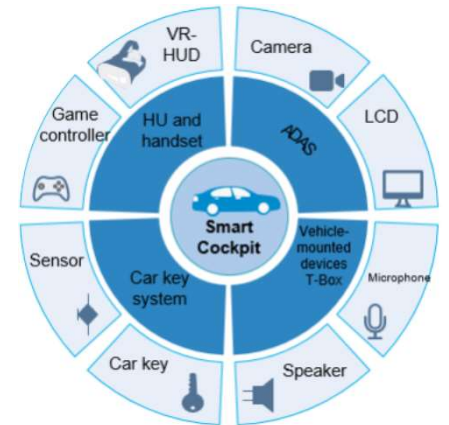
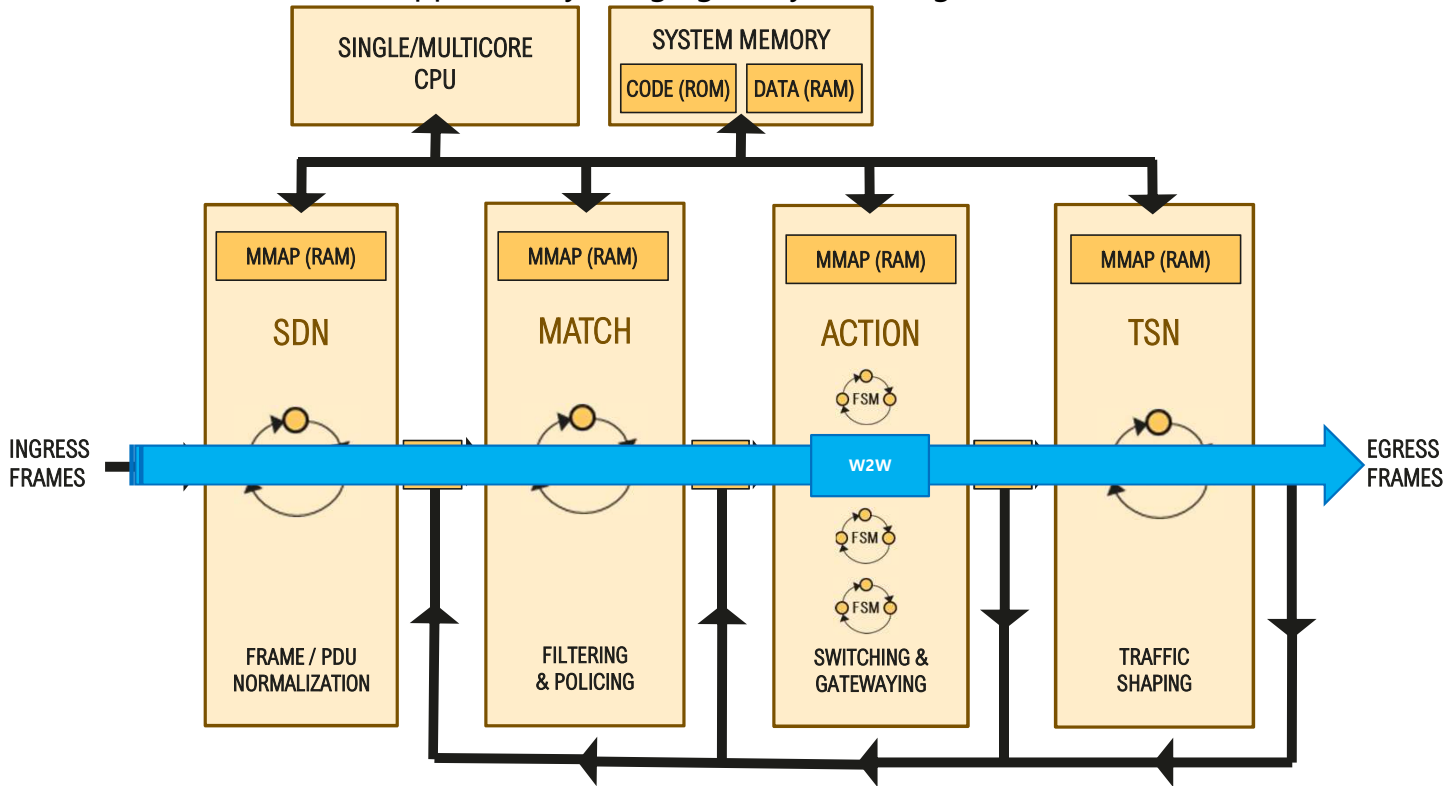
Rationale: Effective balance of high-performance, flexibility/versatility, modularity/reusability and scalability





3.3 INNOVATION 11: W2W – IVN Wired-to-Wireless HWA embedded in eGW SoC

- Wireless technology applied to IVN backbone infrastructure as a **redundant and diverse secondary link or channel**, complementing the primary link based on wired (either copper- or optical-based) Automotive Ethernet, could be a sensible approach by bringing many advantages to the IVN solution

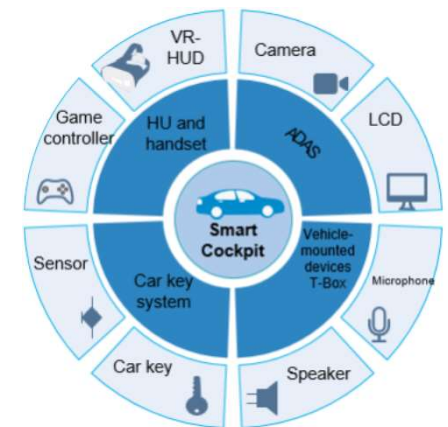
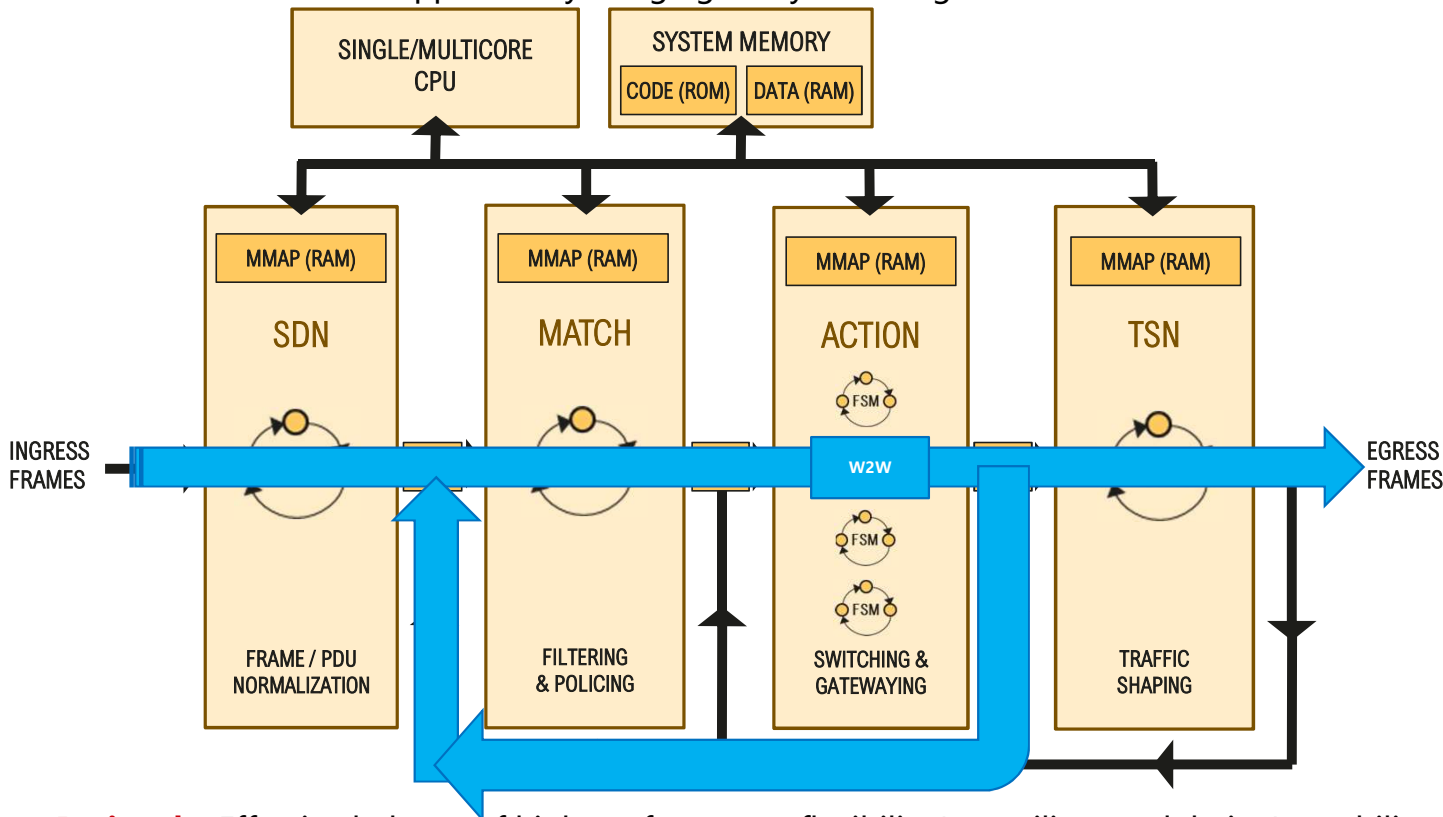


Rationale: Effective balance of high-performance, flexibility/versatility, modularity/reusability and scalability

3.3 INNOVATION 11: W2W – IVN Wired-to-Wireless HWA embedded in eGW SoC



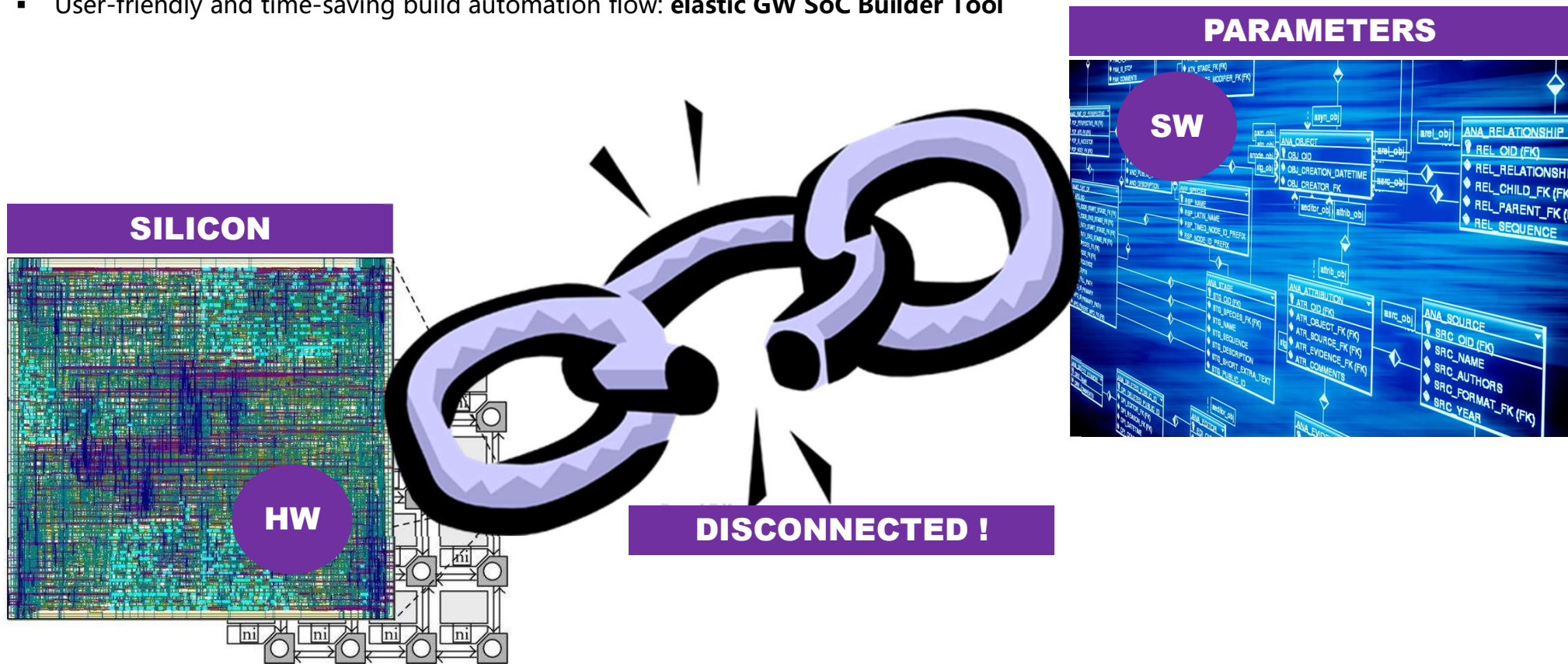
- Wireless technology applied to IVN backbone infrastructure as a **redundant and diverse secondary link or channel**, complementing the primary link based on wired (either copper- or optical-based) Automotive Ethernet, could be a sensible approach by bringing many advantages to the IVN solution



Rationale: Effective balance of high-performance, flexibility/versatility, modularity/reusability and scalability

3.3 INNOVATION 12: EEDA – Elastic Electronic Design Automation (Toolflow for HW Deployment)

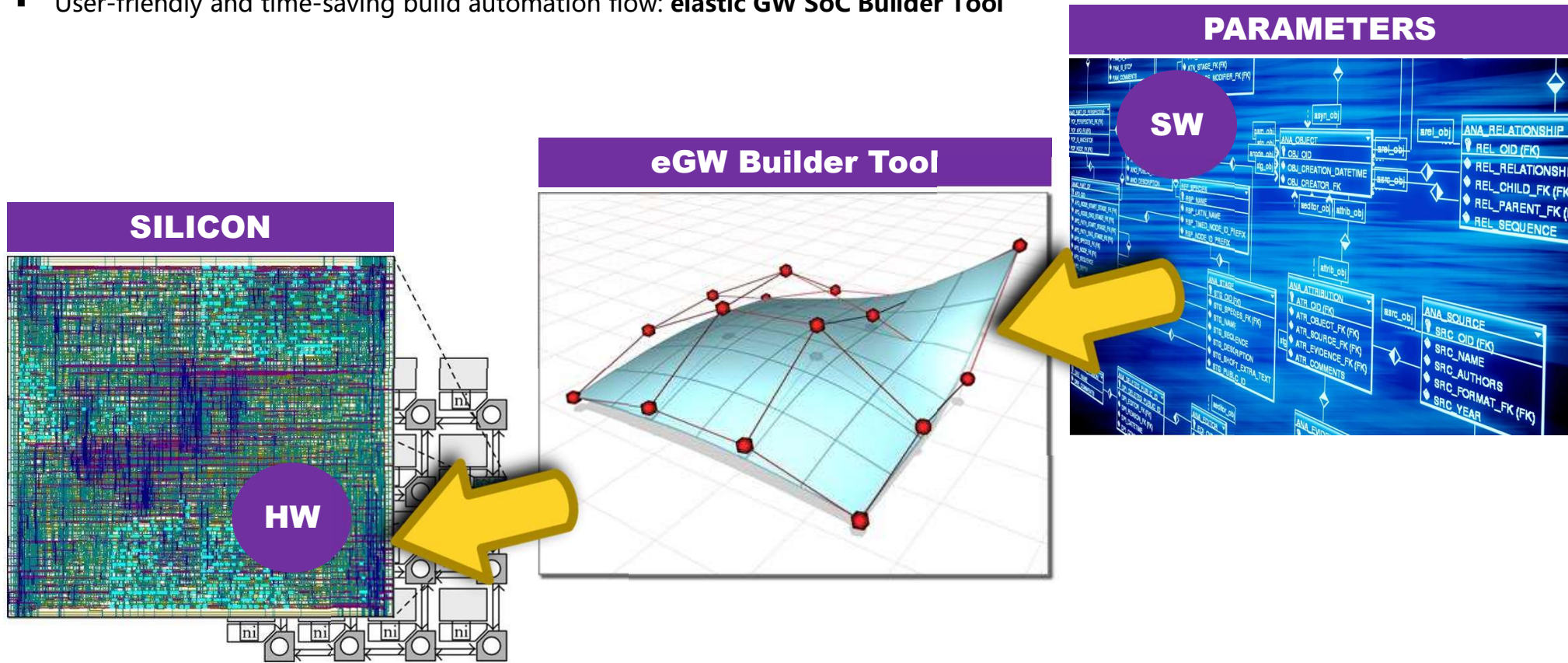
- User-friendly and time-saving build automation flow: elastic GW SoC Builder Tool



Rationale: Automated elastic SoC design methodology based on system parameters exploited at low level

3.3 INNOVATION 12: EEDA – Elastic Electronic Design Automation (Toolflow for HW Deployment)

- User-friendly and time-saving build automation flow: elastic GW SoC Builder Tool



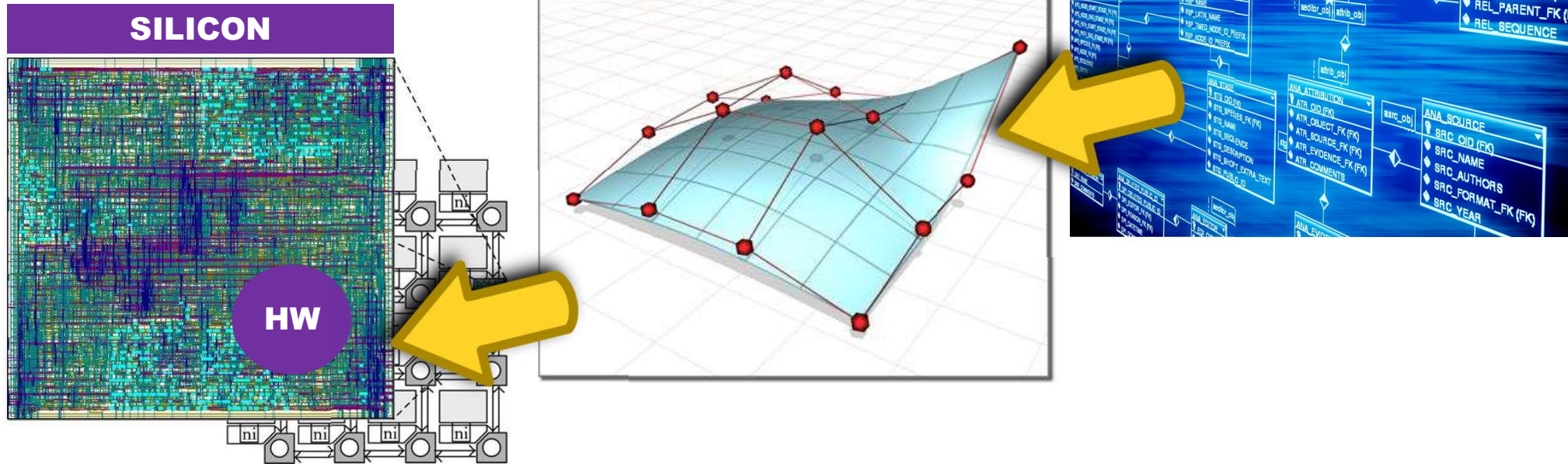
Rationale: Automated elastic SoC design methodology based on system parameters exploited at low level

3.3 INNOVATION 12 : EEDA – Elastic Electronic Design Automation (Toolflow for HW Deployment)

- User-friendly and time-saving build automation flow: **elastic GW SoC Builder Tool**

Our **Core Technology**: **Library of ultra-parameterized IP Cores (HWA)** to enable an extremely flexible low level abstraction able to address SoC design aspects like:

- Geometry (number ingress/egress ports, etc.) and shape (memory, size, wide, etc.)
- IO interface (bus size, wide, etc.)
- Functionality inside (feature #1, #2,... #N)



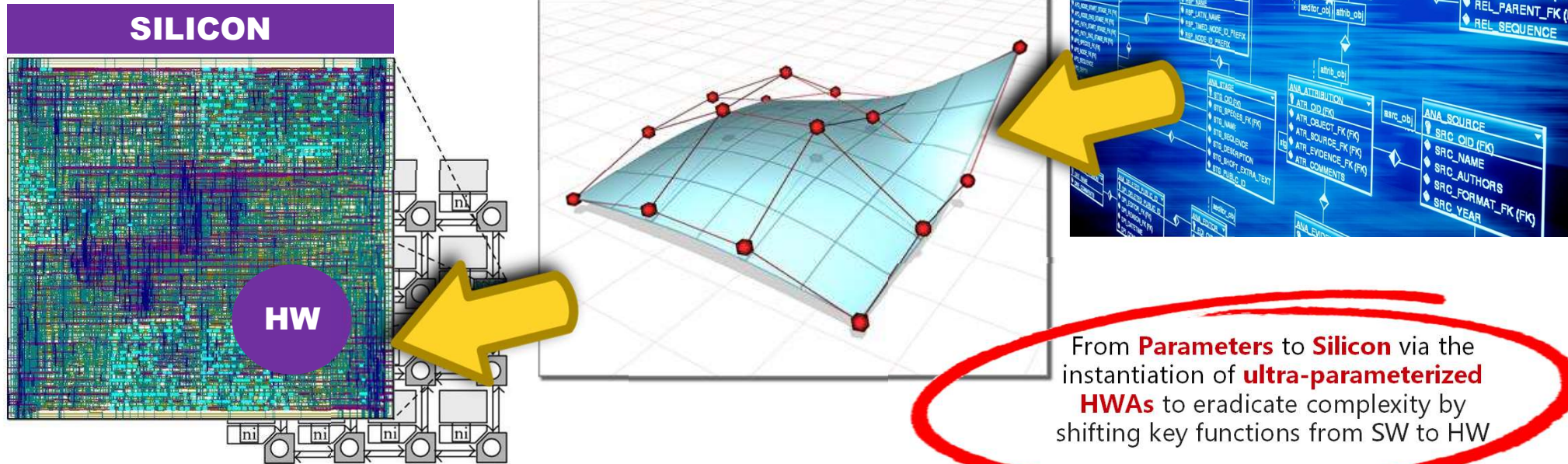
Rationale: Automated elastic SoC design methodology based on system parameters exploited at low level

3.3 INNOVATION 12 : EEDA – Elastic Electronic Design Automation (Toolflow for HW Deployment)

- User-friendly and time-saving build automation flow: **elastic GW SoC Builder Tool**

Our **Core Technology**: **Library of ultra-parameterized IP Cores (HWA)** to enable an extremely flexible low level abstraction able to address SoC design aspects like:

- Geometry (number ingress/egress ports, etc.) and shape (memory, size, wide, etc.)
- IO interface (bus size, wide, etc.)
- Functionality inside (feature #1, #2,... #N)

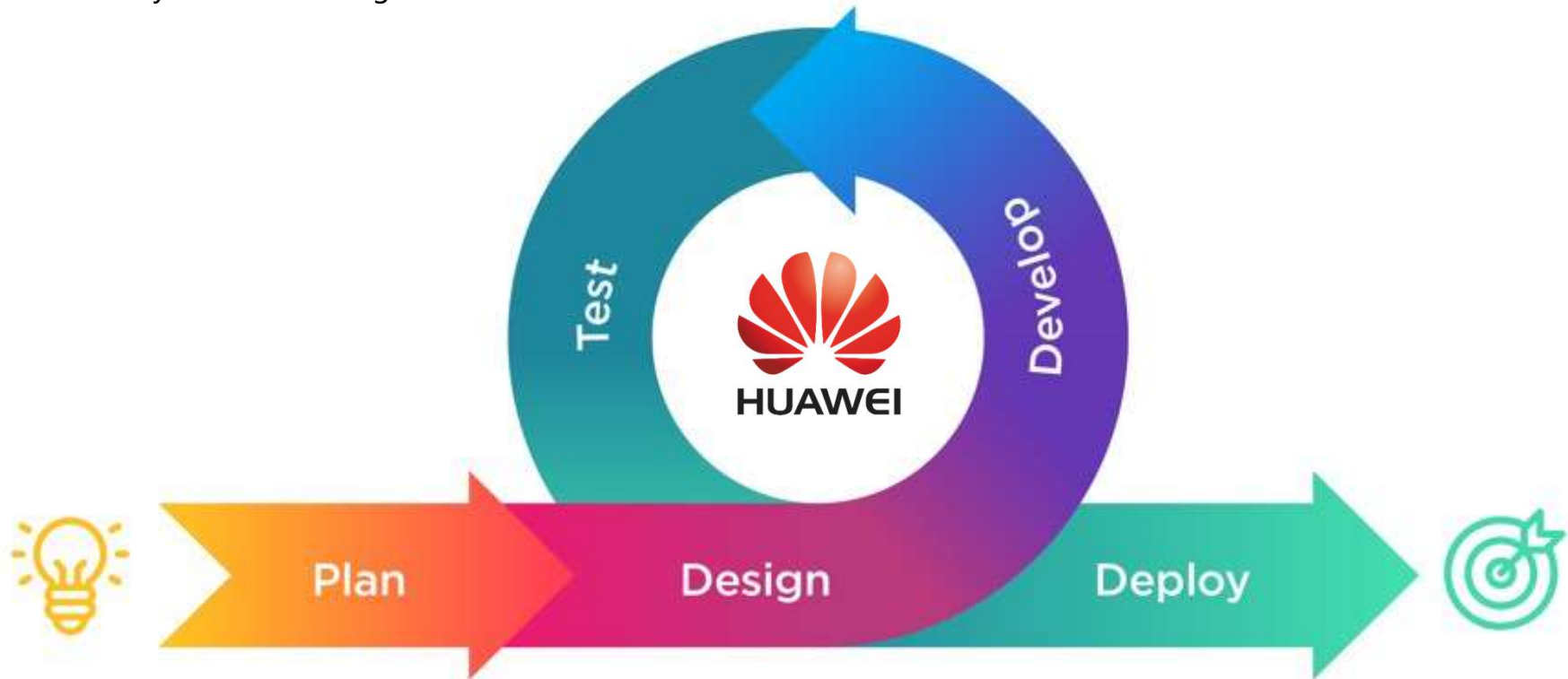


From **Parameters** to **Silicon** via the instantiation of **ultra-parameterized HWAs** to eradicate complexity by shifting key functions from SW to HW

Rationale: Automated elastic SoC design methodology based on system parameters exploited at low level

3.3 INNOVATION ¹²: EEDA – Elastic Electronic Design Automation (Toolflow for HW Deployment)

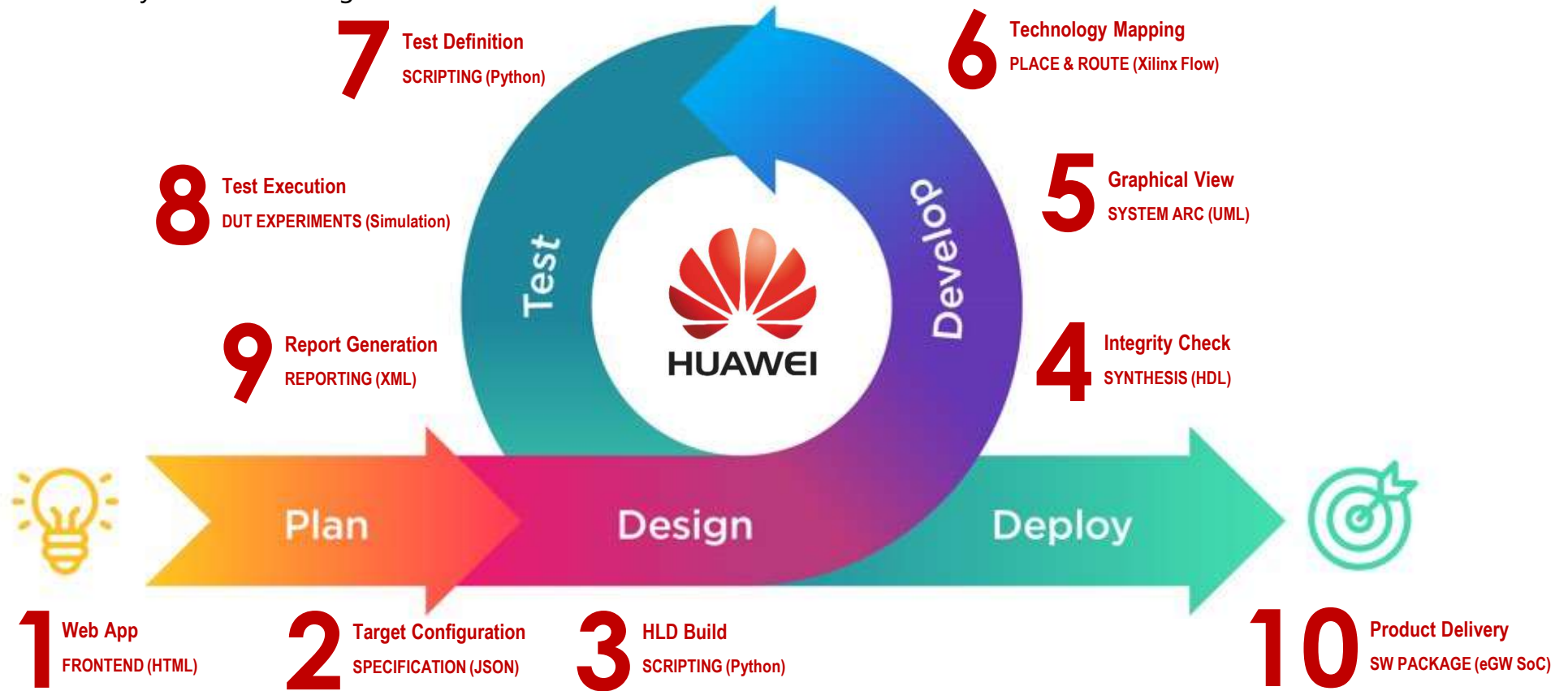
- User-friendly and time-saving build automation flow: **elastic GW SoC Builder Tool**



Rationale: Automated elastic SoC design methodology based on system parameters exploited at low level

3.3 INNOVATION 12: EEDA – Elastic Electronic Design Automation (Toolflow for HW Deployment)

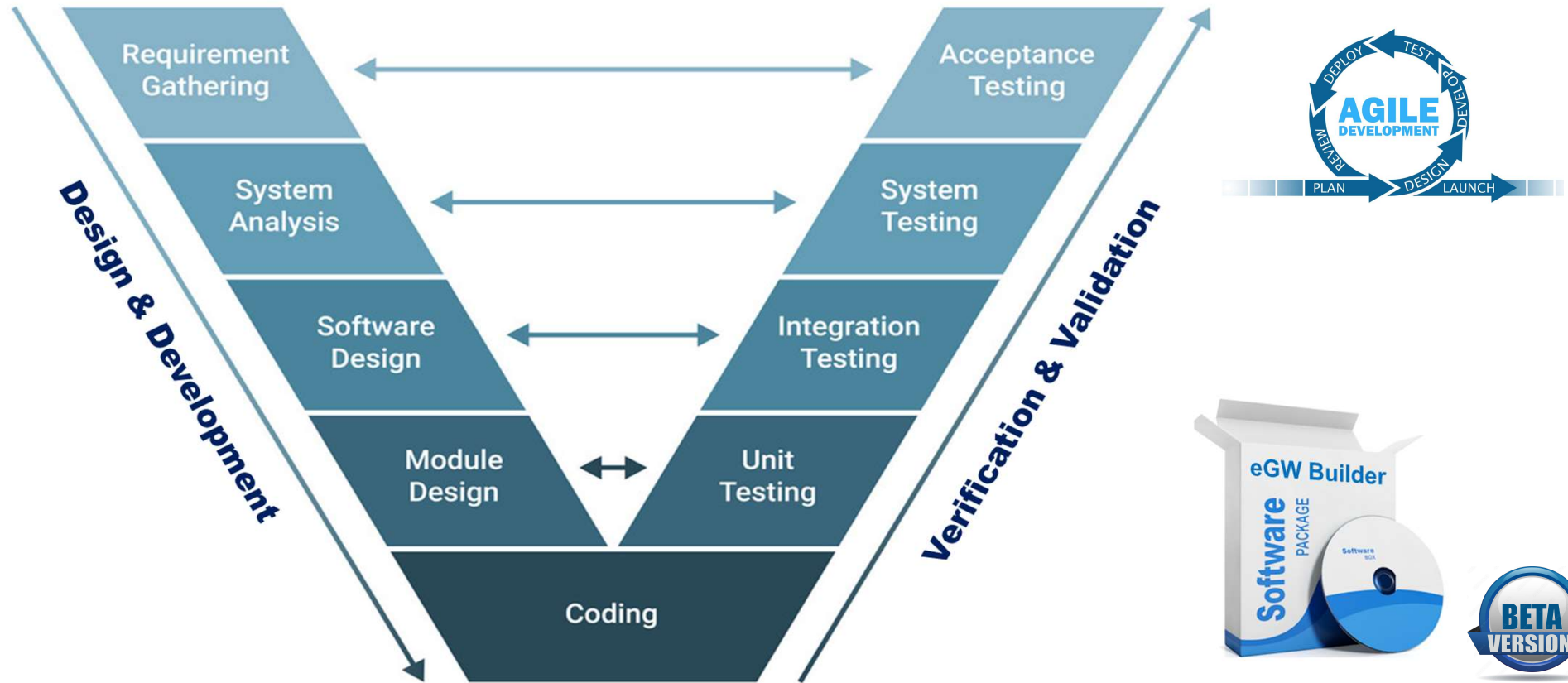
- User-friendly and time-saving build automation flow: **elastic GW SoC Builder Tool**



Rationale: Automated elastic SoC design methodology based on system parameters exploited at low level

3.3 INNOVATION 12: EEDA – Elastic Electronic Design Automation (Toolflow for HW Deployment)

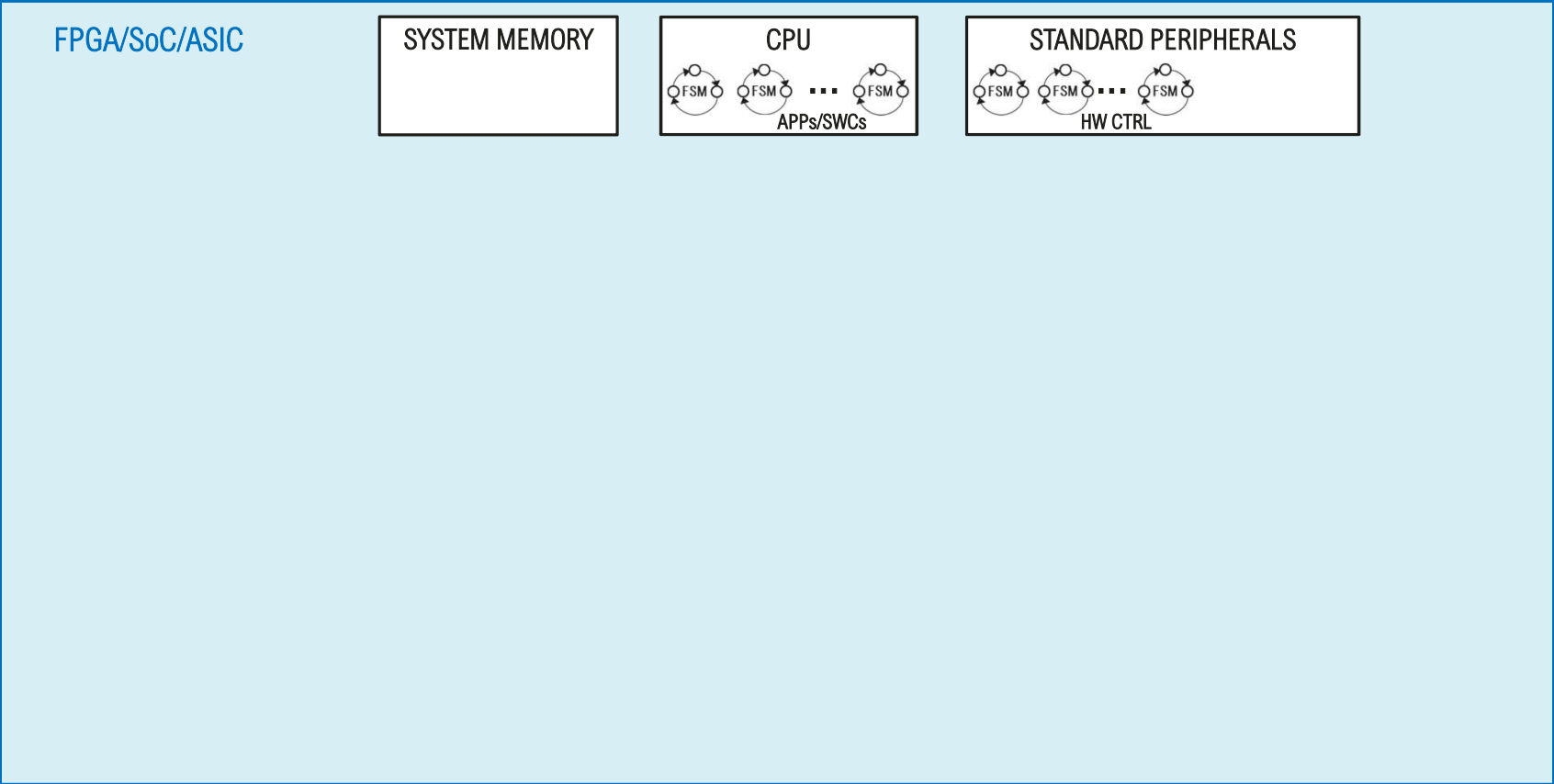
- User-friendly and time-saving build automation flow: elastic GW SoC Builder Tool



Rationale: Automated elastic SoC design methodology based on system parameters exploited at low level

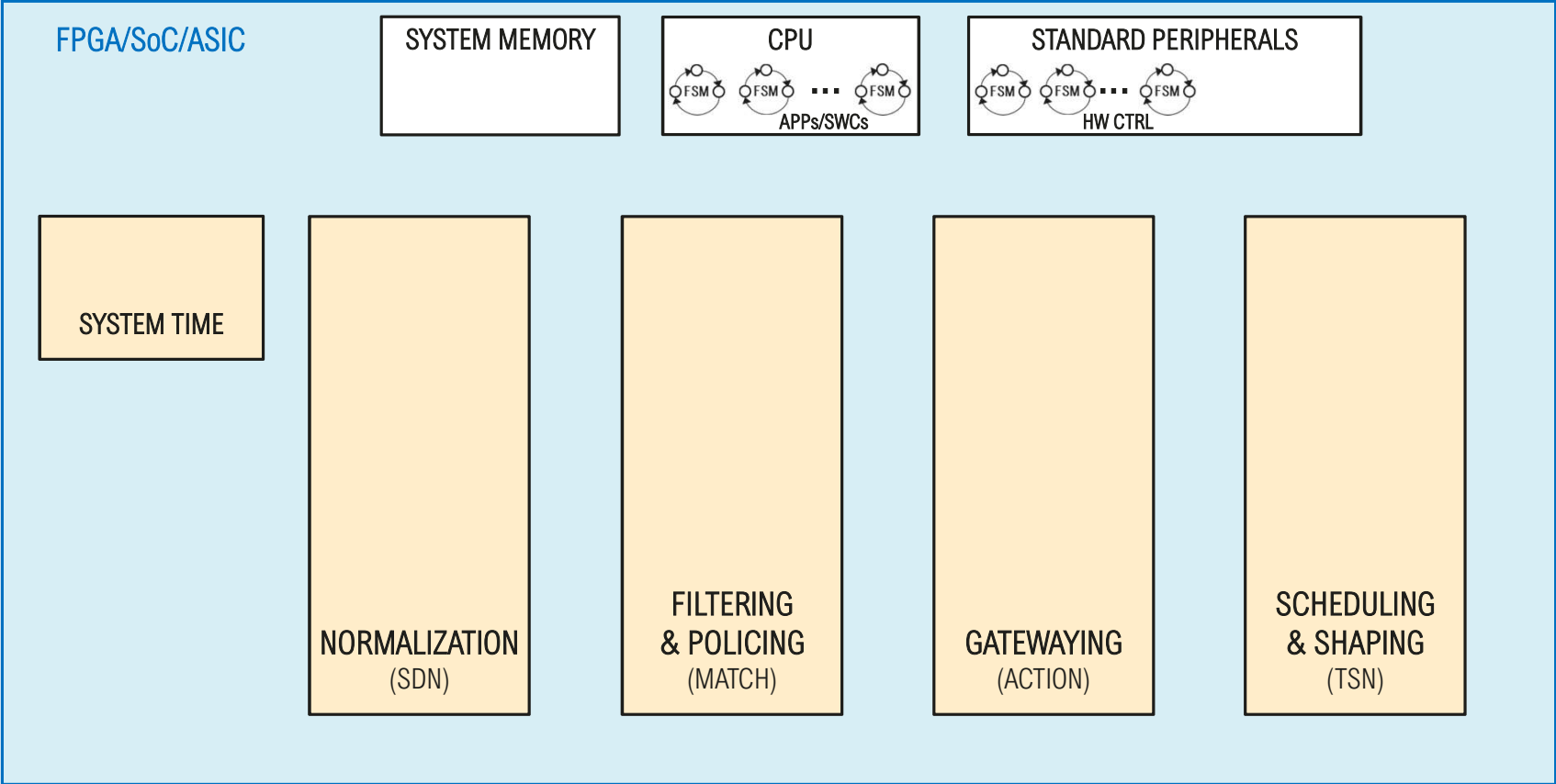
SUMMARY – Automotive Zonal Gatewaying Solution: eGW System & Method

RECAP: eGW SoC targeting Automotive Zonal GWs – 12 HW innovations on Architecture and HW Accelerators (IP cores)



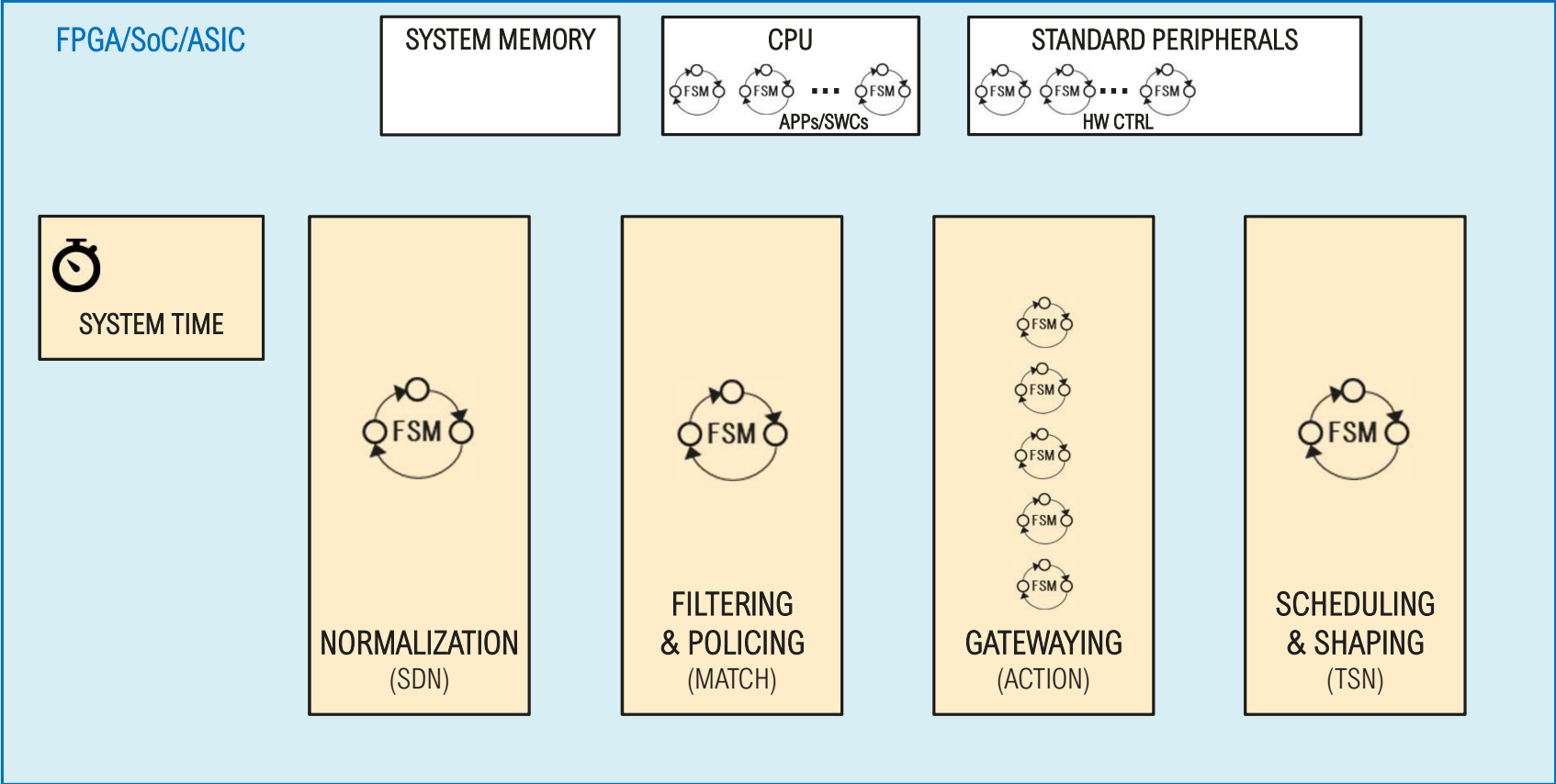
SUMMARY – Automotive Zonal Gatewaying Solution: eGW System & Method

RECAP: eGW SoC targeting Automotive Zonal GWs – 12 HW innovations on Architecture and HW Accelerators (IP cores)



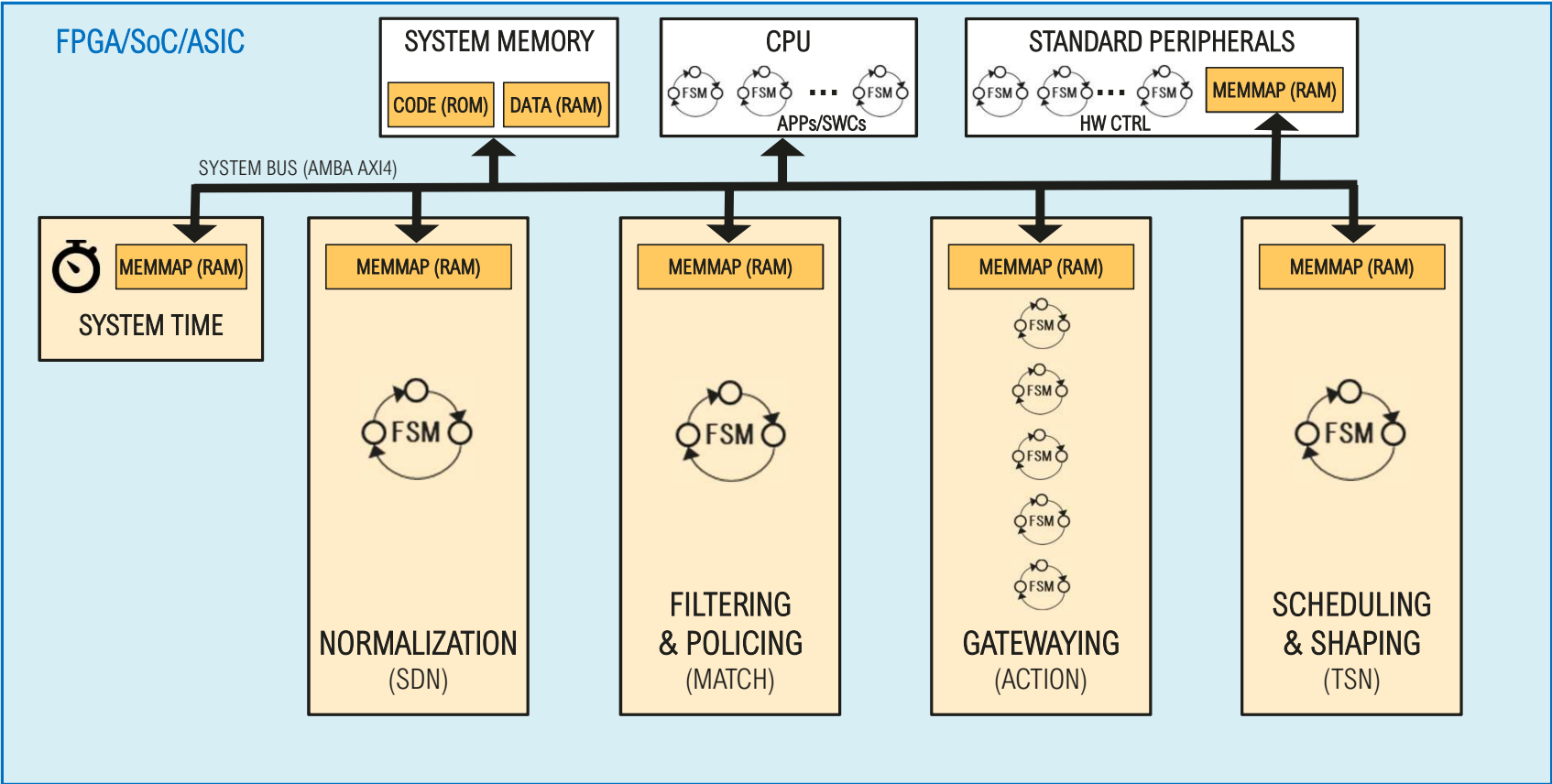
SUMMARY – Automotive Zonal Gatewaying Solution: eGW System & Method

RECAP: eGW SoC targeting Automotive Zonal GWs – 12 HW innovations on Architecture and HW Accelerators (IP cores)



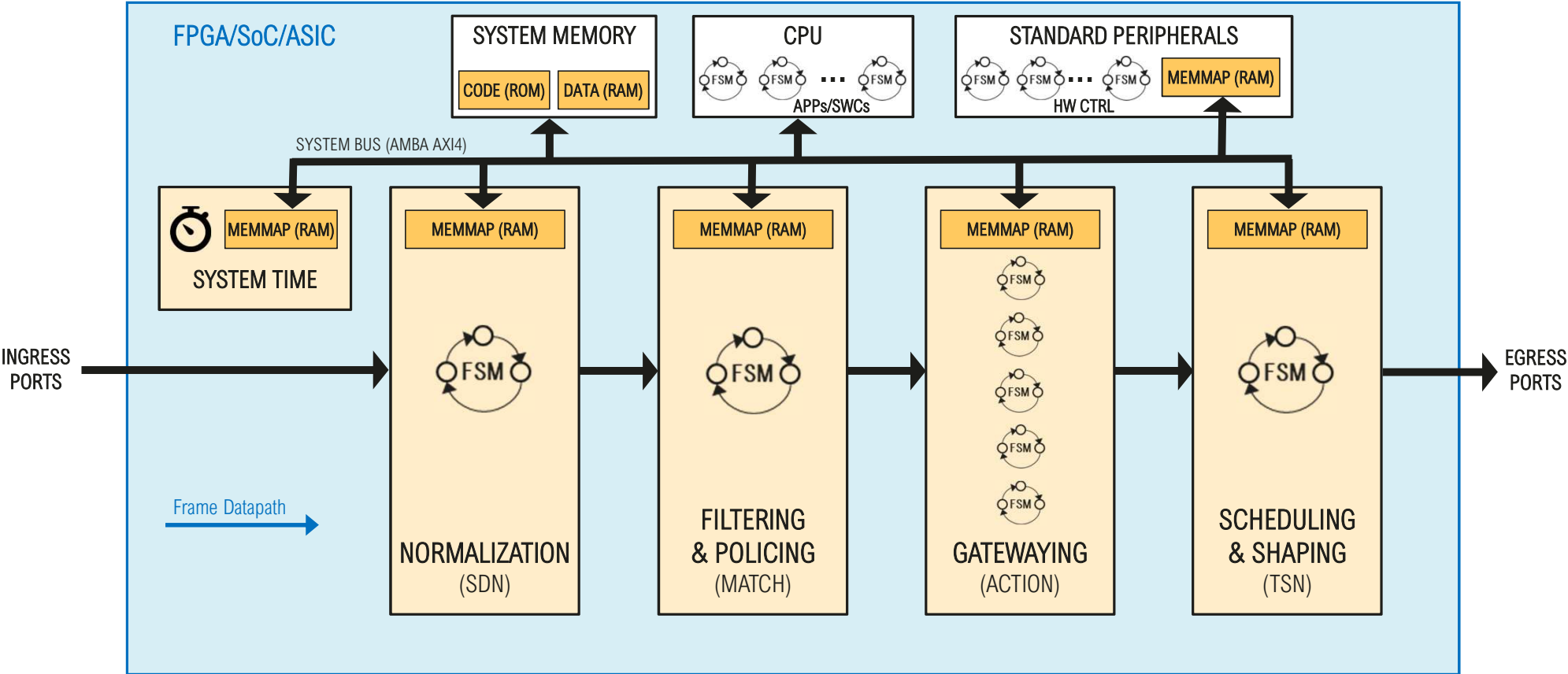
SUMMARY – Automotive Zonal Gatewaying Solution: eGW System & Method

RECAP: eGW SoC targeting Automotive Zonal GWs – 12 HW innovations on Architecture and HW Accelerators (IP cores)



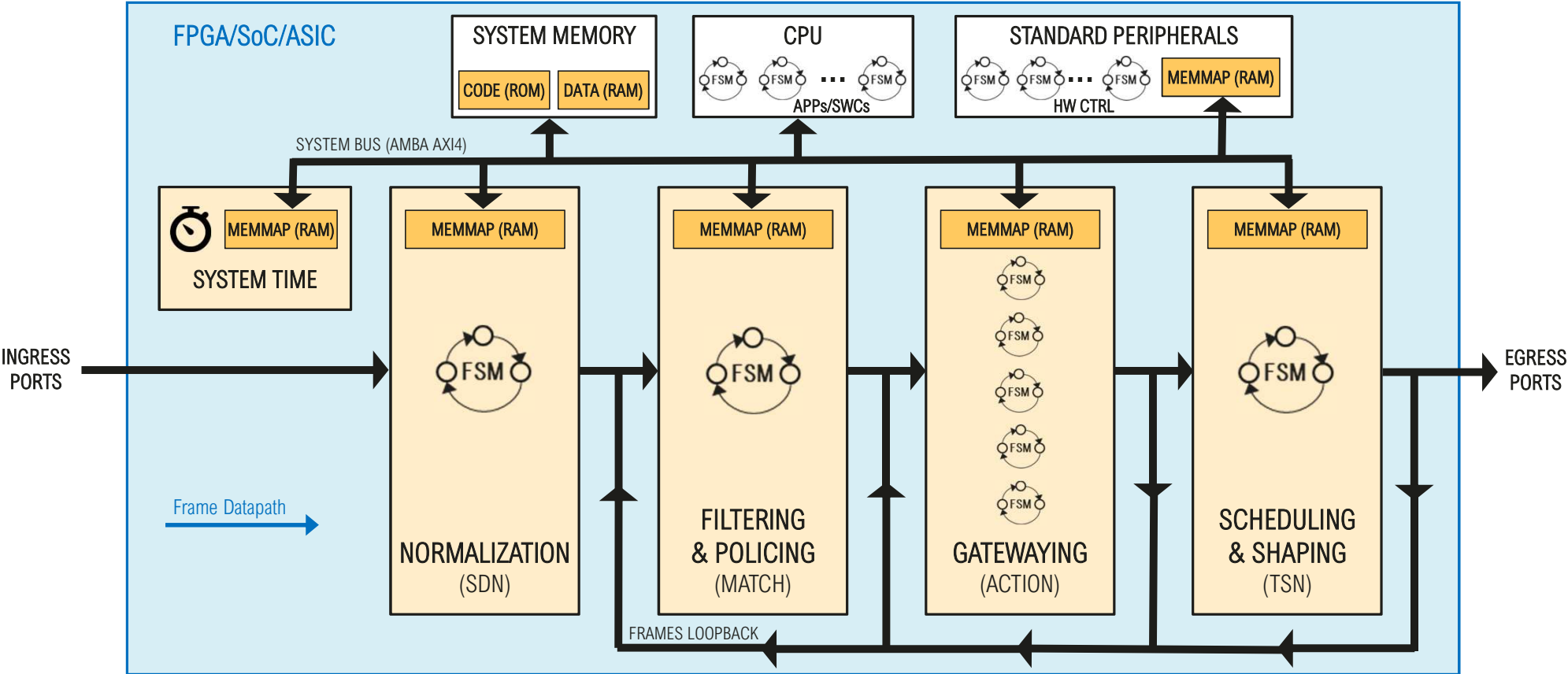
SUMMARY – Automotive Zonal Gatewaying Solution: eGW System & Method

RECAP: eGW SoC targeting Automotive Zonal GWs – 12 HW innovations on Architecture and HW Accelerators (IP cores)



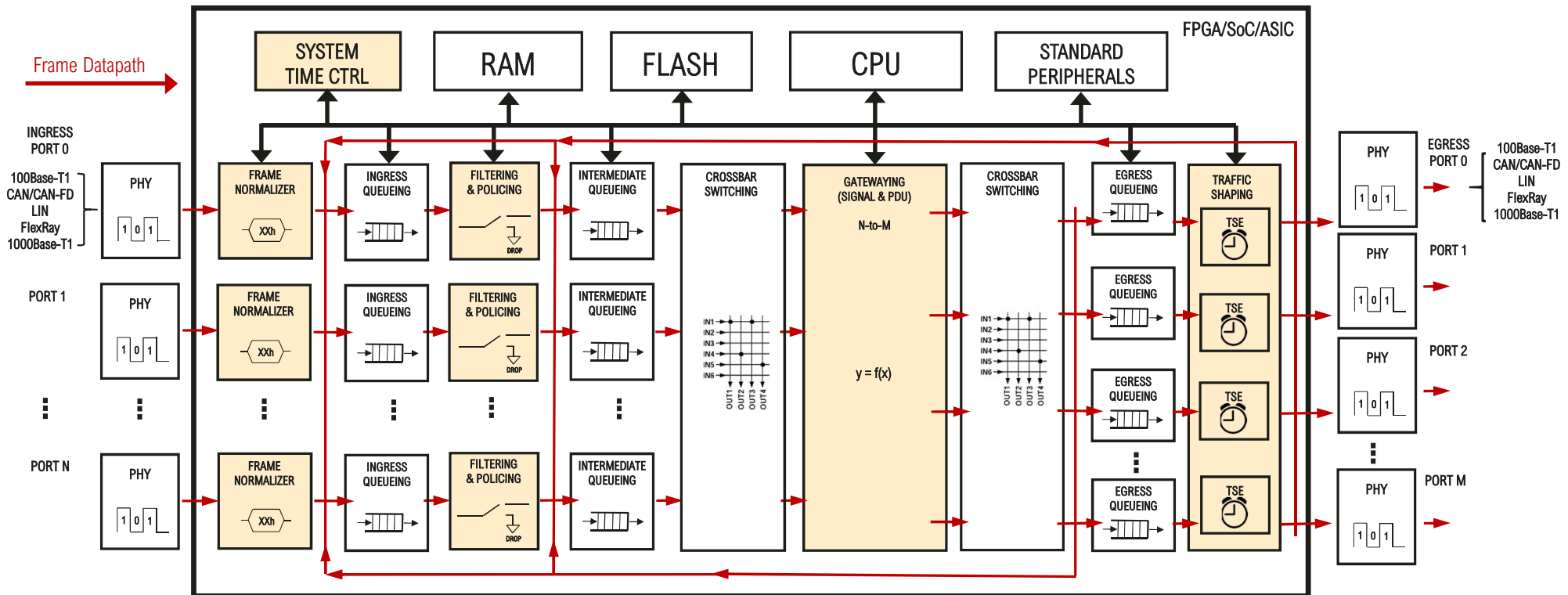
SUMMARY – Automotive Zonal Gatewaying Solution: eGW System & Method

RECAP: eGW SoC targeting Automotive Zonal GWs – 12 HW innovations on Architecture and HW Accelerators (IP cores)



SUMMARY – Automotive Zonal Gatewaying Solution: eGW System & Method

eGW - The pioneer all-in-one gatewaying solution combining high-performance (ultra-low latency, jitter), networking (SDN, TSN, DDS), application processing, security (ACL, NGFW, NIDS) and safety features (redundancy, self-monitoring, self-healing) driven by a full ingress-to-egress data flow deployed through custom HWAs to effectively offload the host CPU and prune SW complexity

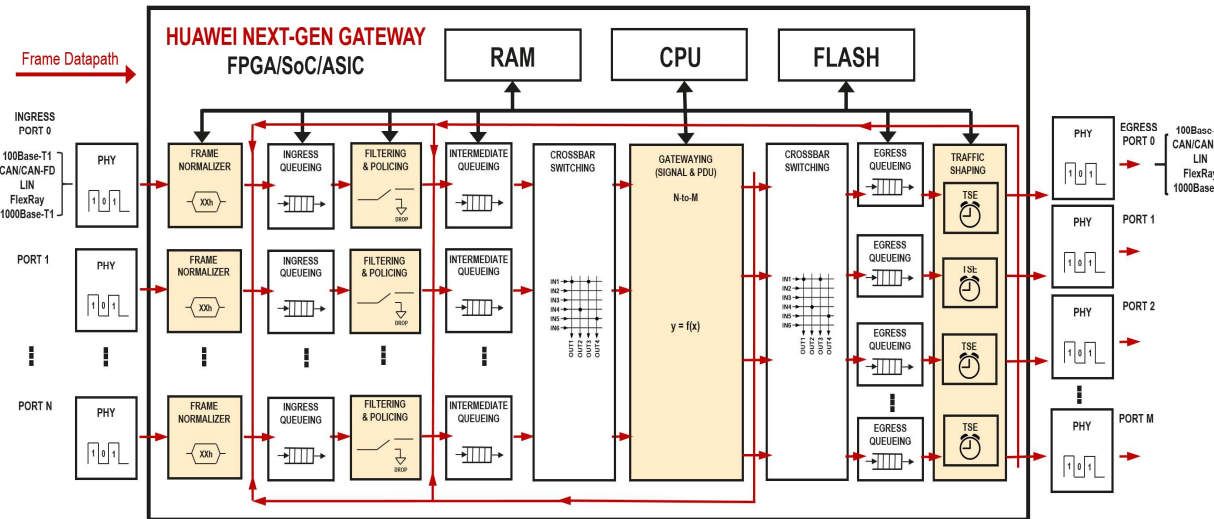
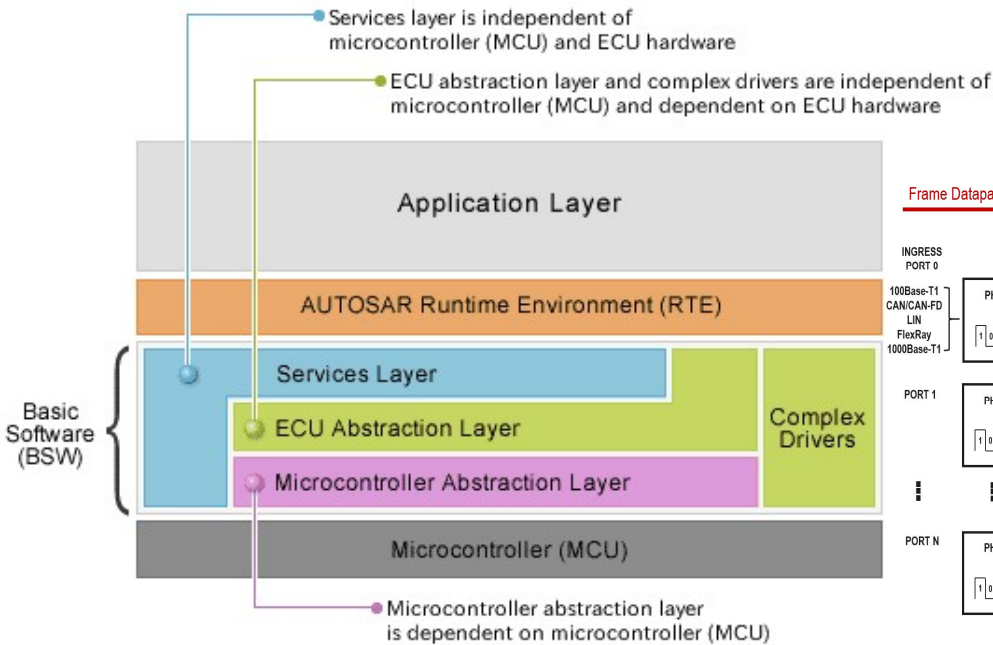


SUMMARY – Automotive Zonal Gatewaying Solution: eGW System & Method

HW/SW Codesign: Synthesis of functionality distributed in SW modules & HW IP cores to shift from SW centricity to HW centricity (our proposed paradigm shift!)

Reshaping of HW/SW stacks with a new functional breakdown:

- **App Layer:** Host CPU (orchestration and management)
- **BSW Layer:** Innovative HWAs (HW-centric APP & NW Functions/Primitives)



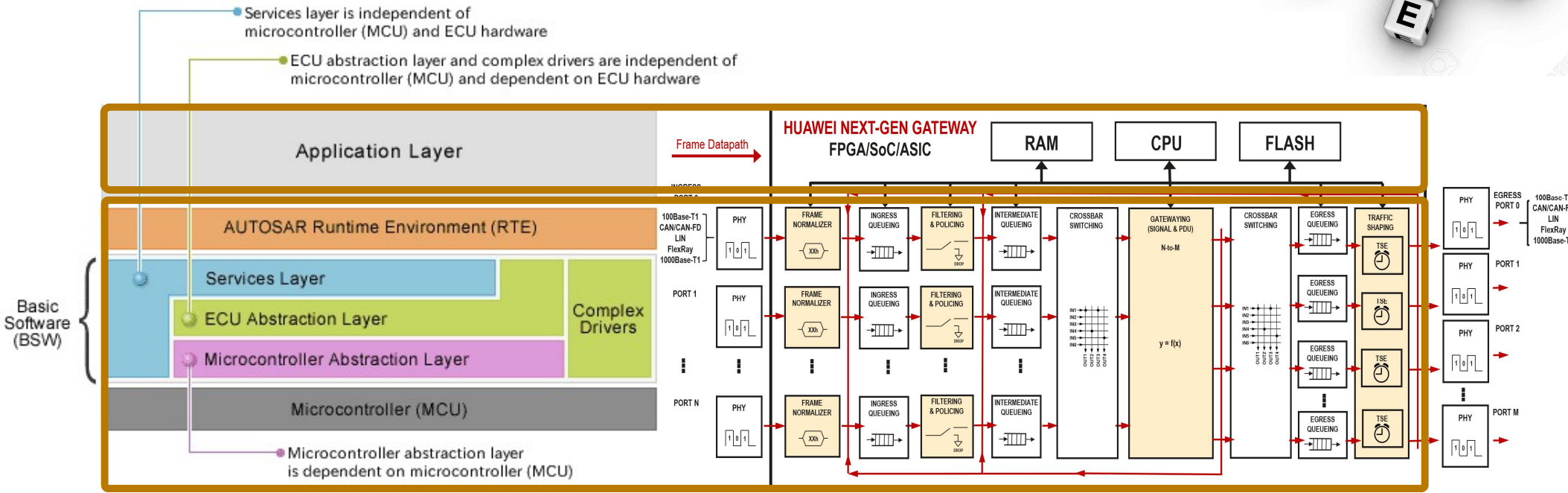
VS eGW-SoC-ARC Paradigm

SUMMARY – Automotive Zonal Gatewaying Solution: eGW System & Method

HW/SW Codesign: Synthesis of functionality distributed in SW modules & HW IP cores **to shift from SW centricity to HW centricity (our proposed paradigm shift!)**

Reshaping of HW/SW stacks with a new functional breakdown:

- **App Layer:** Host CPU (orchestration and management)
- **BSW Layer:** Innovative HWAs (HW-centric APP & NW Functions/Primitives)

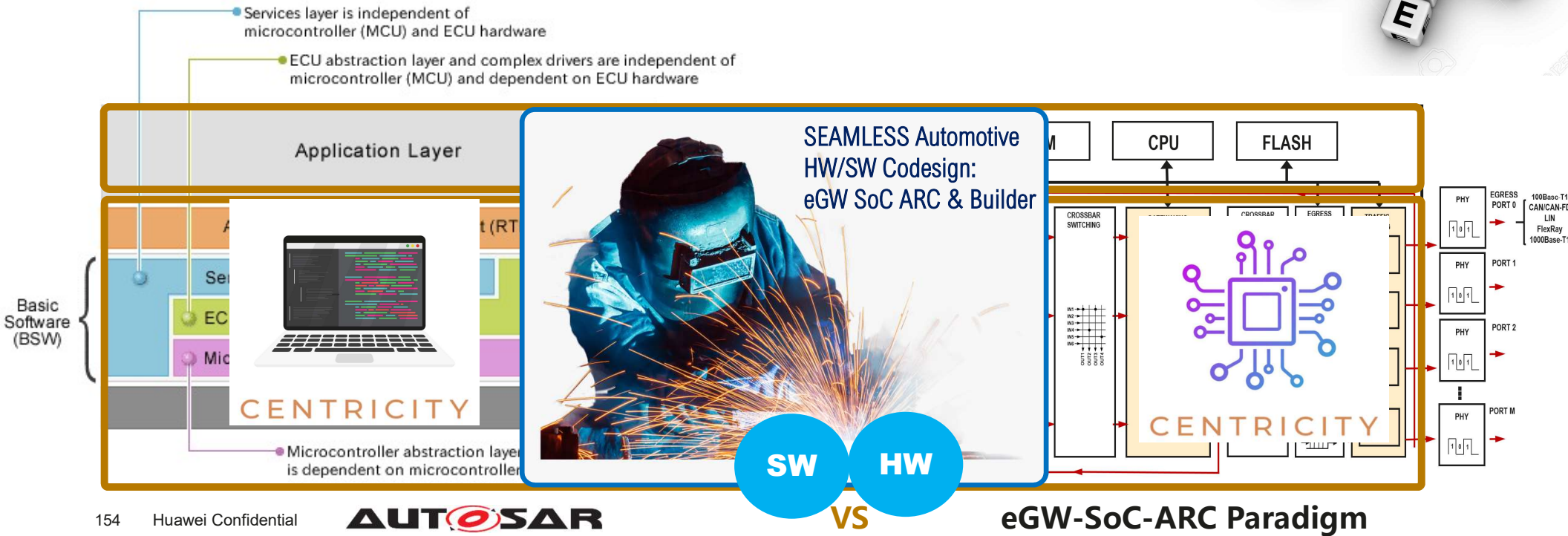


SUMMARY – Automotive Zonal Gatewaying Solution: eGW System & Method

HW/SW Codesign: Synthesis of functionality distributed in SW modules & HW IP cores **to shift from SW centricity to HW centricity (our proposed paradigm shift!)**

Reshaping of HW/SW stacks with a new functional breakdown:

- **App Layer:** Host CPU (orchestration and management)
- **BSW Layer:** Innovative HWAs (HW-centric APP & NW Functions/Primitives)



SUMMARY – Automotive Zonal Gatewaying Solution: eGW System & Method



INNOVATION

SYSTEM (eGW SoC) &

SEAMLESS Automotive HW/SW Codesign: eGW SoC ARC & Builder

SW HW

METHOD (eGW Builder)

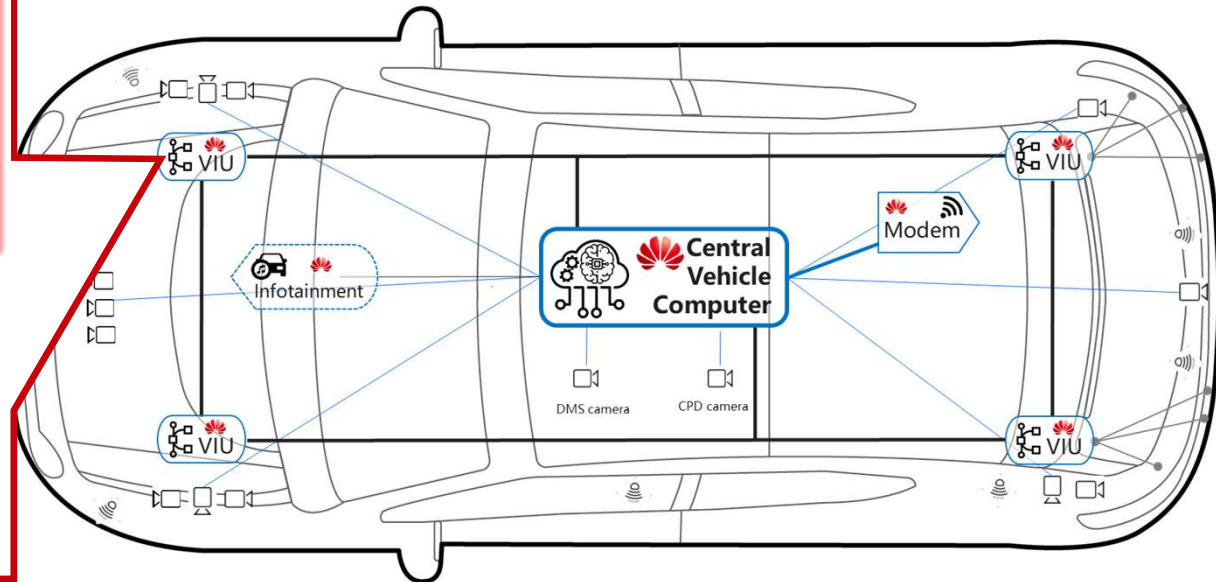
SUMMARY – Automotive Zonal Gatewaying Solution: eGW System & Method

eGW SoC Prototype



Deliverables & Outcomes:

- 14 Patent Applications (EU and CN)
- 8 Conference Papers
- 2 Journals (impact factor)
- 1 Master Thesis eGW Builder (Grade 1.0)
- 1 PhD Thesis eGW SoC (ongoing)



SUMMARY – Automotive Zonal Gatewaying Solution: eGW System & Method

eGW SoC Prototype



Deliverables & Outcomes:

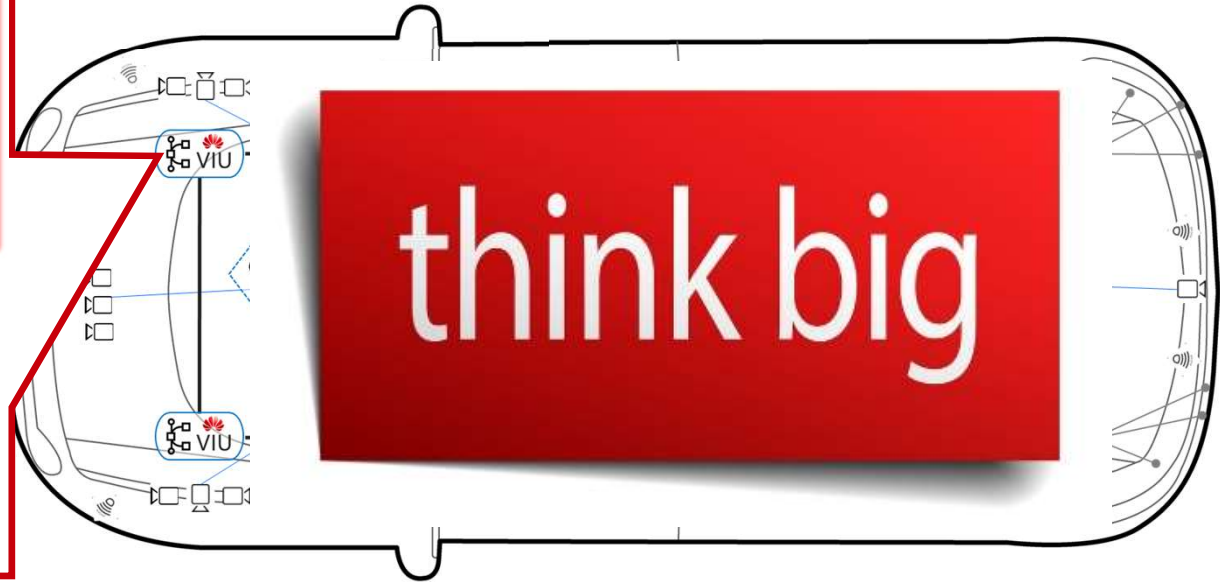
- 14 Patent Applications (EU and CN)
- 8 Conference Papers
- 2 Journals (impact factor)
- 1 Master Thesis eGW Builder (Grade 1.0)
- 1 PhD Thesis eGW SoC (ongoing)

WORK IN PROGRESS...



Our Ultimate Goal !

think big



CONCLUSIONS

"**Software Defined Vehicle (SDV)** should not mean developing **complex SW** but implementing **simple SW** to configure and control reliable & smart **flexible HW** dedicated to **high performance computing and communications.**"

"Because developing **ACES vehicles** implies to efficiently craft **μ s** or **hundreds of ns**, not **ms** any more, **at both computation and communication levels.**"

Francesc Fons, TUM AS 2022

Thank you.



HUAWEI

Dr.-Ing. Francesc Fons

francesc.fons@huawei.com

Automotive IVN Research Group

Automotive Engineering Lab - Huawei Munich Research Center