sondrel

digital turnkey services

Complexity delivered simply

A Scalable Framework for Fast Design Space Exploration of AI Workloads in Automotive SoCs

Carlos Román
Piyush Singh

ARC Processor Summit, 8 September 2022

Agenda

sondrel

- Sondrel - Who we are? - What we do?
- Introduction to Sondrel's Scalable Architecture Framework (SAF)
- Methodology for Architecture Modelling & Design Space Exploration
- Modelling a ViT (Vision Transformer) Automotive Application Comprising of:
  - ARC® NPX6 NPU and VPX5 DSP
  - Arteris FlexNoC® Interconnect
  - DesignWare® LPDDR5 Memory Controller (SystemC™ TLM Model)
- Conclusions

Be sure with Sondrel

sondrel

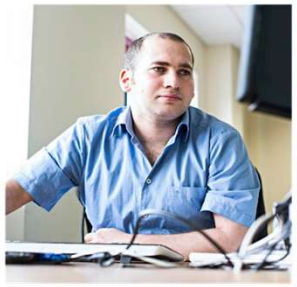
UK
Private company
Est. 2002

140+ Engineers
Access to scale, & expertise

100's of designs
180nm to 5nm

Approved by
Global Foundries,
Samsung & TSMC

Quality & security audited
ISO 9001, 26262, 27001



Design Centres

UK
- Theale, UK (HQ)
- Bristol

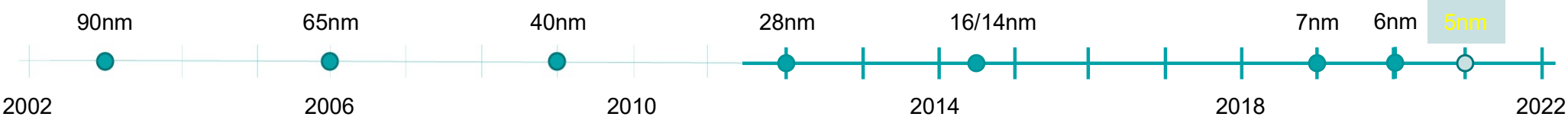
China
- Xi'an

India
- Hyderabad

Morocco
- Rabat

Sales

Santa Clara, USA
Theale, UK
Xi'an, CN



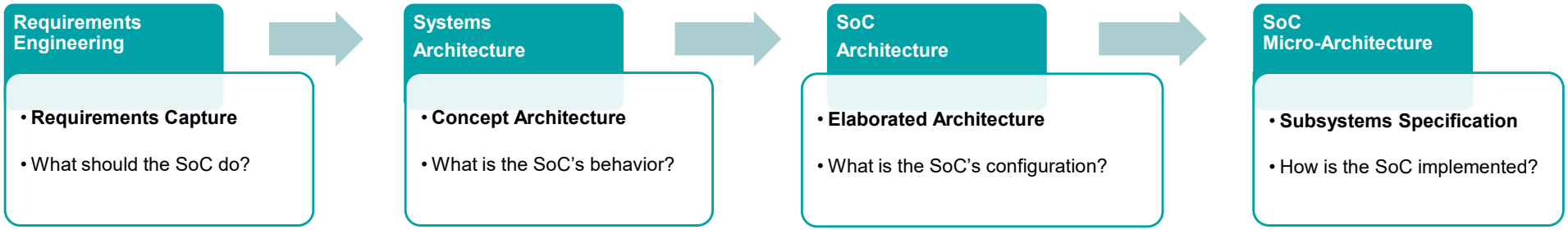
ASIC Services Engagement Models

sondrel

FULL TURNKEY	From product requirements to volume production ASIC design including integration of proprietary IP Supply Chain for shipment of tested devices	**TRUSTED** **SECURE**
SCM	Supply Chain Management as a standalone service	**EFFICIENT**
PROJECT	Engagements defined by a SoW (Statement-of-Work) Complete projects undertaken within Sondrel's secure design centers Project managed and tool chain defined and maintained by Sondrel	**EFFECTIVE** **EXPERT** **QUALITY**

Problem Statement

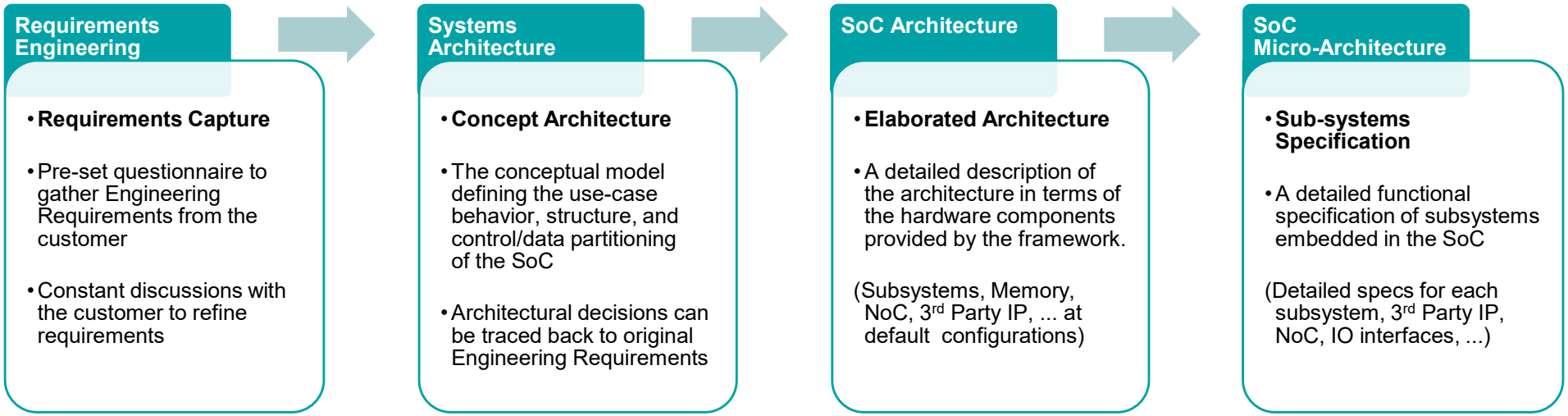
sondrel



SoC Architecture Challenge: **"What is the configuration of the SoC?"**

Answer: It depends on several factors (next slide).

Phases Defined in the Scalable Architectural Framework



Partitioning the framework into phases allows for targeted change management as requirements converge when customizations are needed

Motivation for Architecture Exploration and Analysis

sondrel

Challenges which the SAF framework overcomes:

1. Product Complexity

   More functionality integrated into one or more SoCs/Chiplets

2. SoC Architecture Challenges

   Complex SoC infrastructure due to wide range of Use-case requirements

3. SoC Design Challenges

   SoC infrastructure specialization given a large design space

Steps to Architecture Exploration

sondrel

1. Define Use-case model
   - Capture KPIs

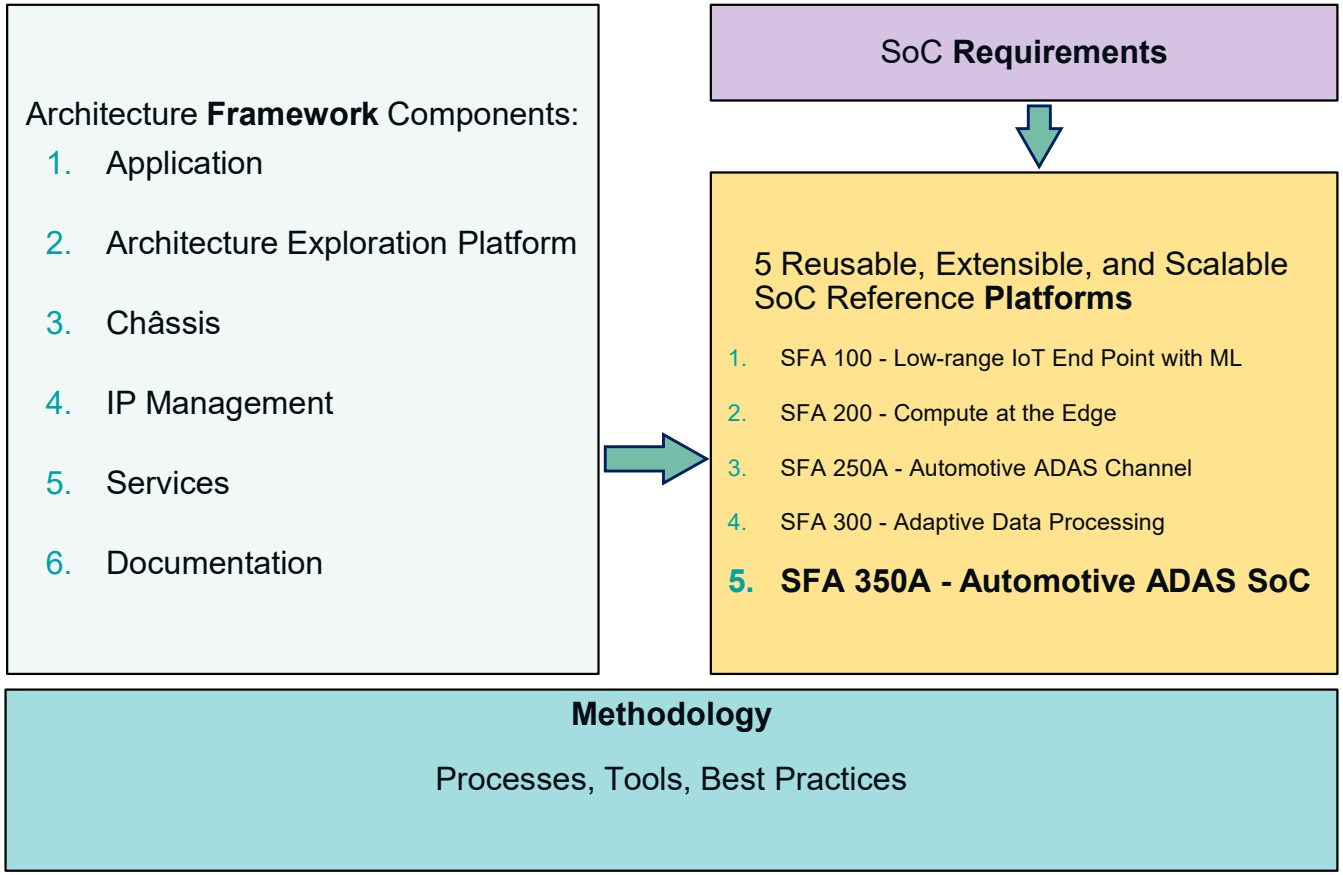
2. Choose relevant SoC Platform model
   Application Specific (IoT, Edge Computing, Automotive, ...)

3. Map Use-case onto SoC Platform
   - Perform process-to-resource mapping
     - Tune default configuration parameters of resources

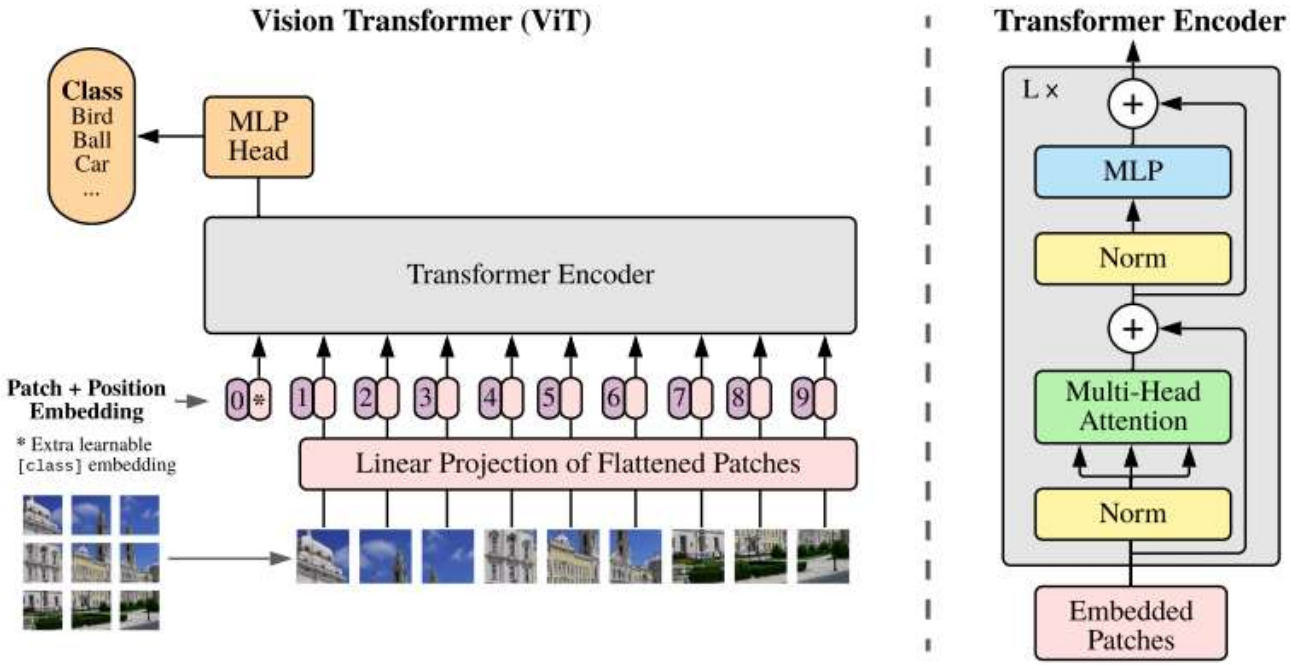
4. Perform full system exploration
   - Fine-tune SoC's configuration parameters to meet KPIs

Scalable Architecture Framework™

sondrel

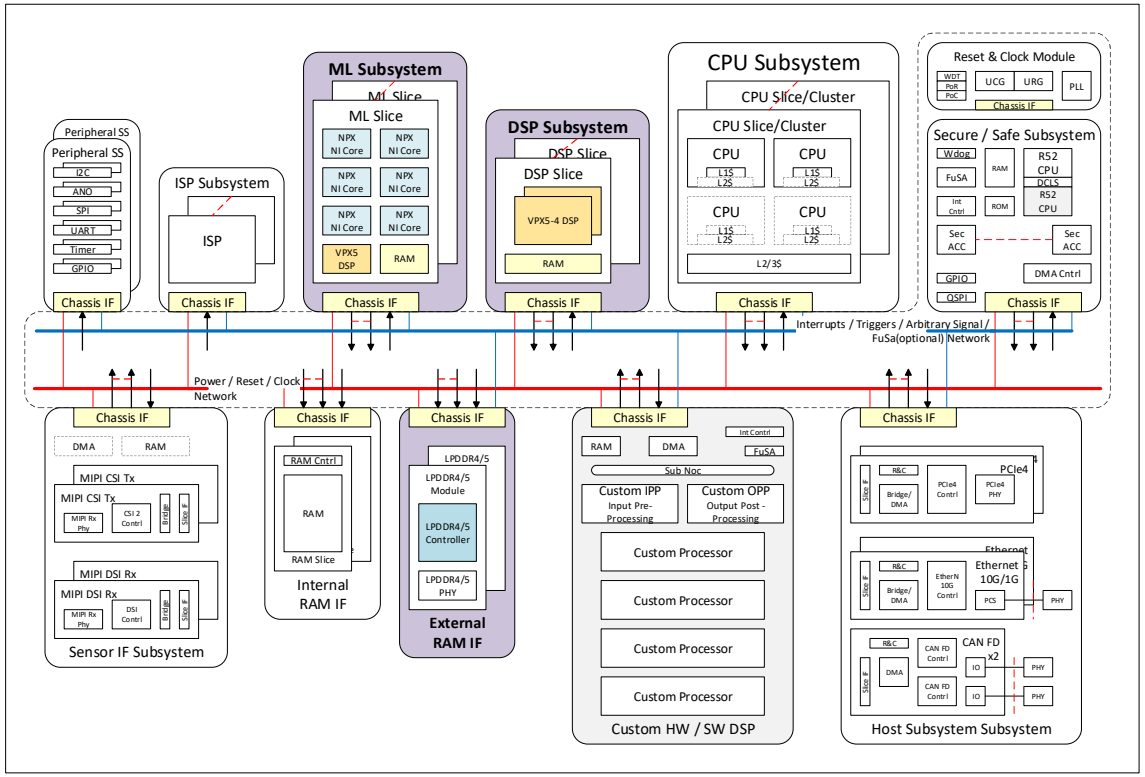


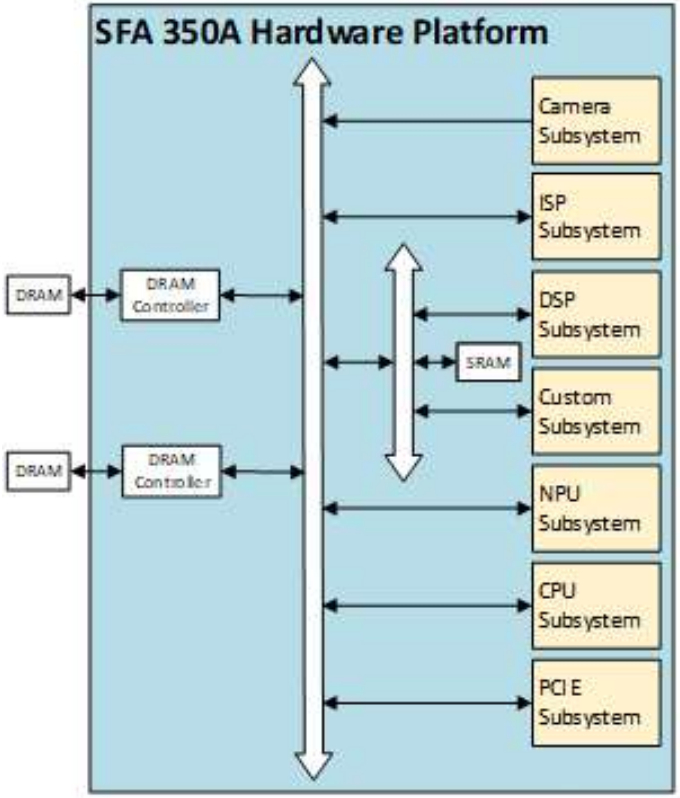
What is a Vision Transformer (ViT)?



- Similar to a CNN (Convolutional Neural Network):
  - **Input:** Image or video frames
  - **Output:** Information about what is contained in the input image or video frames

Scalable Architecture Framework™ – Chassis Schematic





- **Chassis**: Generic starting point framework to arrive at SoC's concept architecture
- **SFA350A**: Reference scalable platform for architecture modelling featured in this presentation

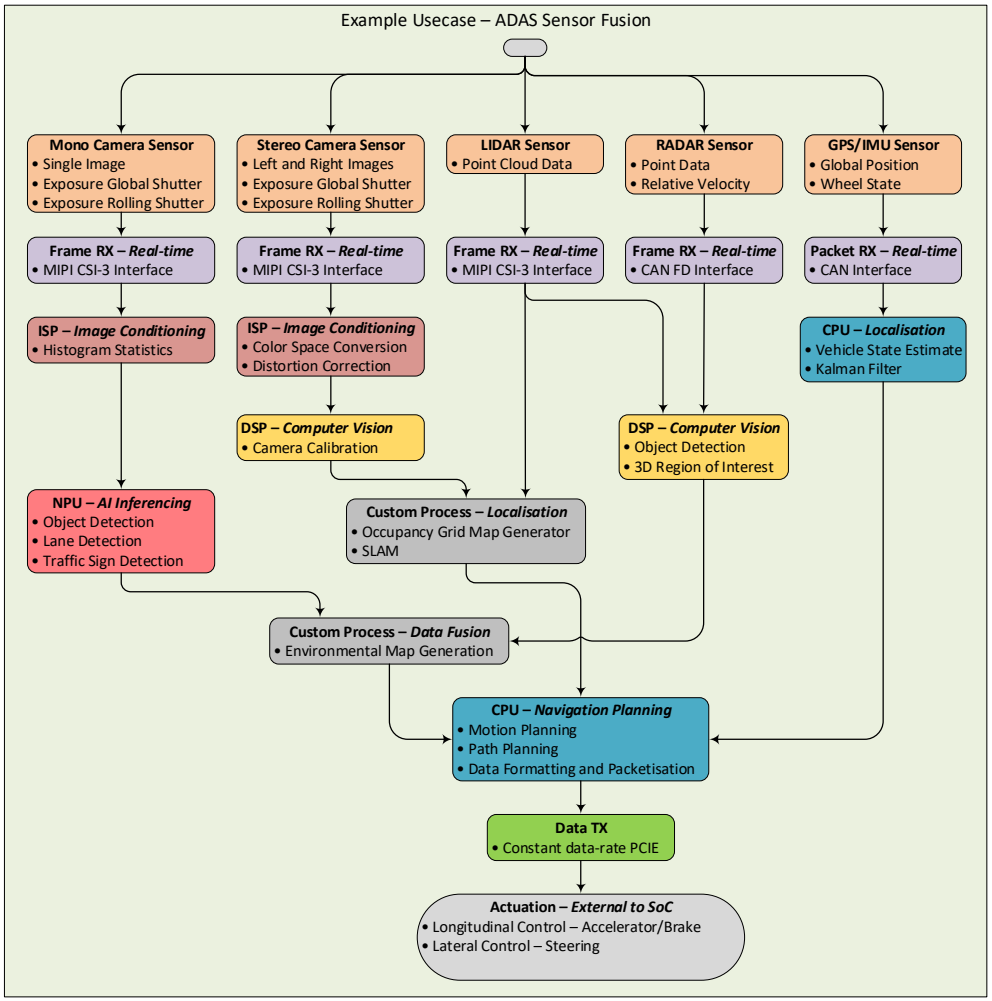
Modelling the Application Use-case

sondrel

- The “application Use-case” is a representation of the software that an SoC needs to support. Also known as “workload models”.
- Workload models replace actual models of the IP subsystems, by generating bus traffic equivalent to the actual IP blocks

Model Style	Modelling Target	Description
Stochastic	System	Generate random bus traffic based on functional parameters controlling the probability distribution of events and their sequence
Application Driven	Software	Generate bus traffic based on a task graph performance model the application
Trace Driven	Hardware	Generate bus traffic based on transactions

Workload Model – ADAS Sensor Fusion



Workload model capture means generating a graph

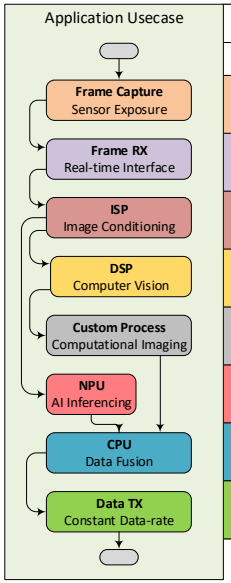
- Nodes tasks or processes
- Edges dataflows between nodes

Notes:

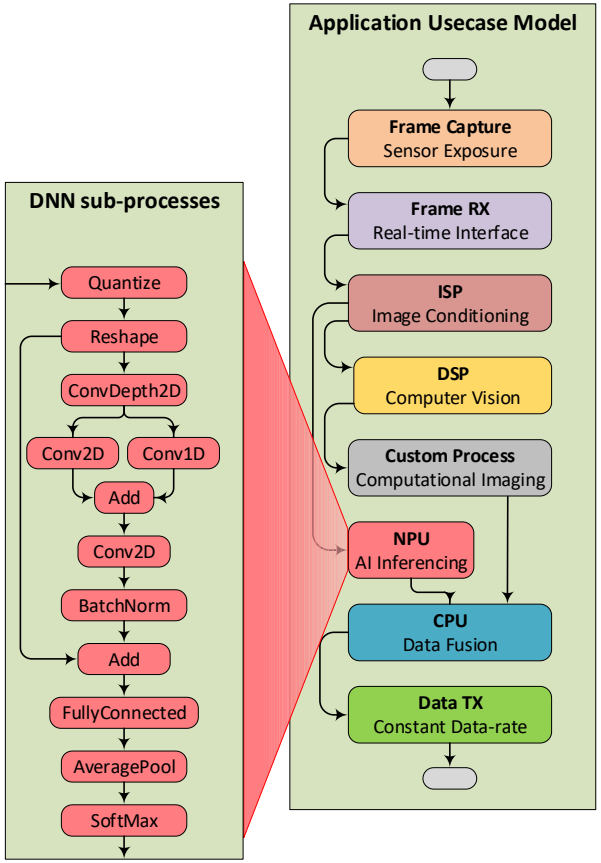
1. Task decomposition helps conquer complexity
2. Algorithmic details of a compute task are **not** required
3. Capture of the compute's task duration is sufficient

Use-case Capture for Architecture Exploration

sondrel

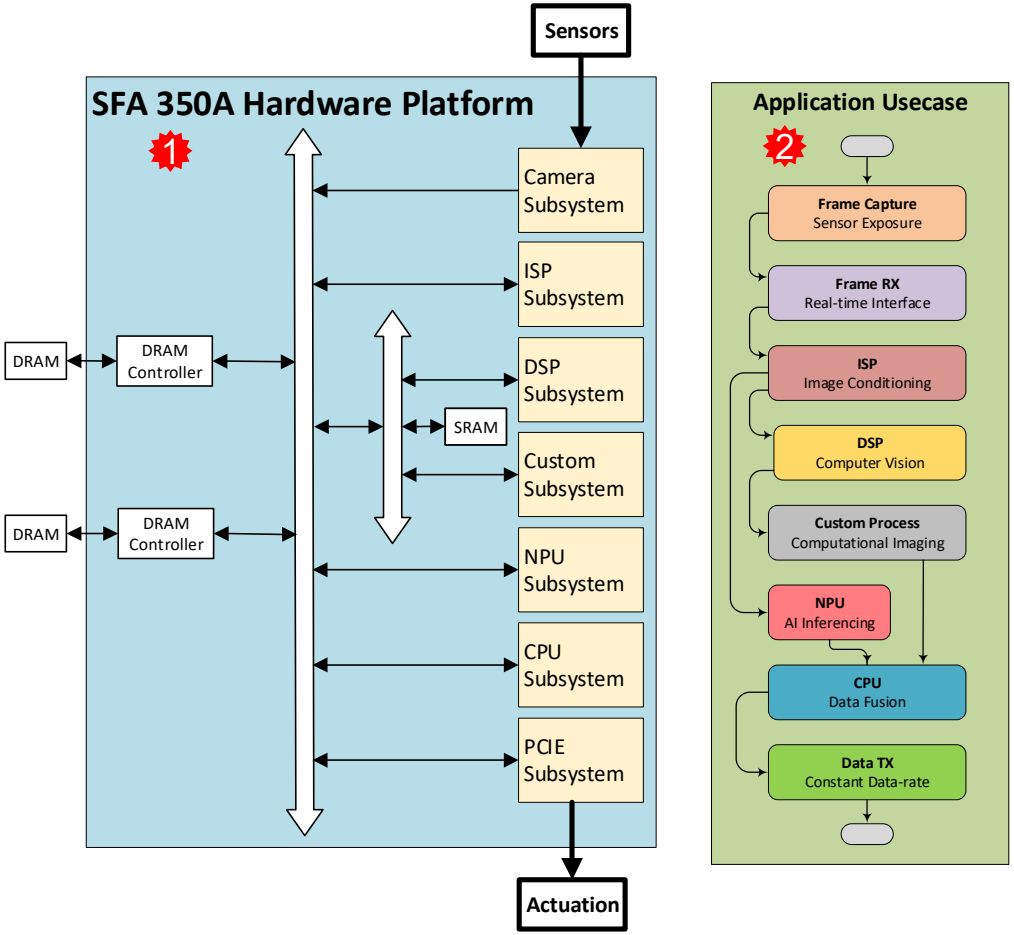


	Read Blocks					Processing	Write Blocks				
	Size	Total	Pace	Mem Address	Addr Pattern	Duration	Size	Total	Pace	Mem Address	Addr Pattern
Frame Capture Sensor Exposure	n.a.	n.a.	n.a.	n.a.	n.a.	5ms	n.a.	n.a.	n.a.	n.a.	n.a.
Frame RX Real-time Interface	n.a.	n.a.	n.a.	n.a.	n.a.	28ms	8KB	3096	3.5us	0x000	sequential
ISP Image Conditioning	8KB	3096	3.5us	0x000	sequential	14ms	4KB	1548	7us	0x4000	sequential
DSP Computer Vision	4KB	1548	10us	0x4000	tiled	10ms	4KB	1548	10us	0x8000	sequential
Custom Process Computational Imaging	4KB	1548	30us	0x8000	tiled	10ms	4KB	1548	30us	0xc400	sequential
NPU AI Inferencing	4KB	1548	40us	0x4000	sequential	15ms	4KB	1548	40us	0xc000	sequential
CPU Data Fusion	8KB	1548	5us	0xc000	sequential	5ms	8KB	1548	5us	0xc000	sequential
Data TX Constant Data-rate	8KB	1548	33ms	0xc000	sequential	33ms	n.a.	n.a.	n.a.	n.a.	n.a.



- Tabular format describes the use-case graph to the simulator
  - Each Row represents a process in the Use-case graph
  - Each Column captures attributes for each process (i.e., SRAM access patterns)
- Who provides this information?
  - Usually Product Manager working with Systems Architect

Architecture Exploration Models

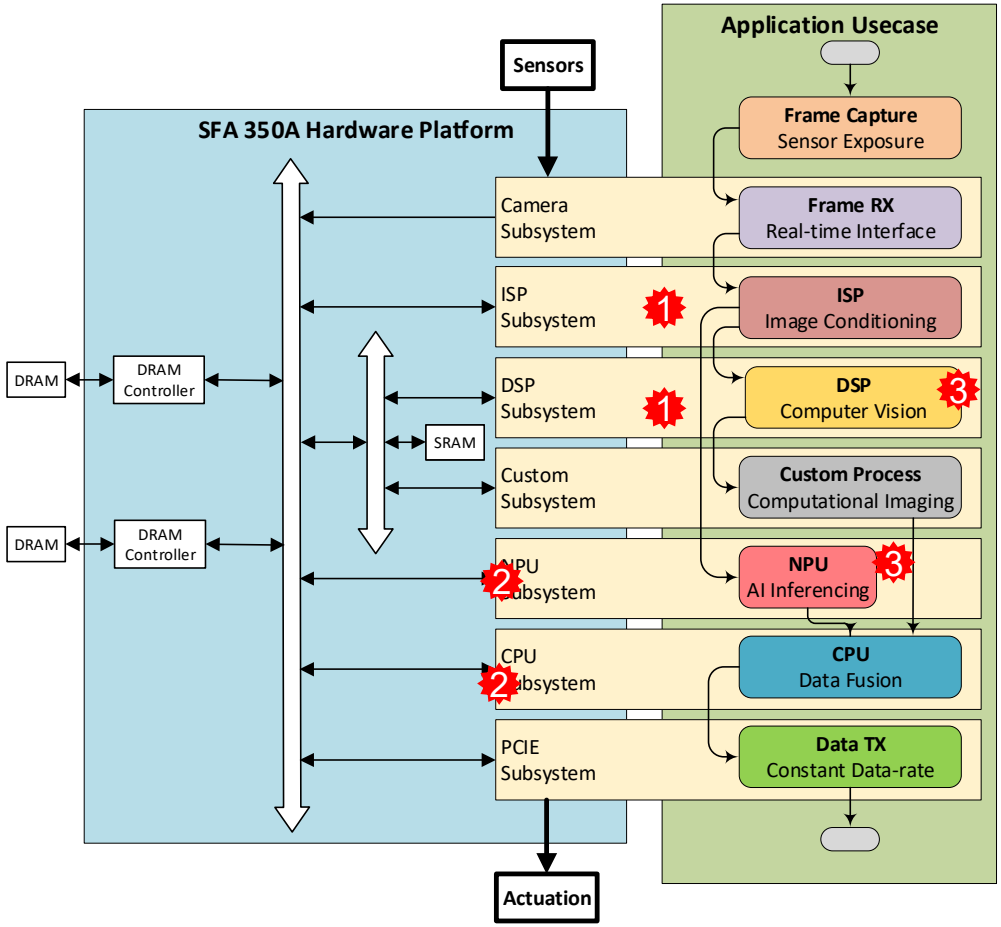


Hardware platform & Application use-case are the two fundamental models required to construct the SoC's workload model for architecture performance exploration

1. Hardware platform:
   - Cycle-accurate (fast-timed model) representation of the SoC as a **Concept Architecture** for exploration

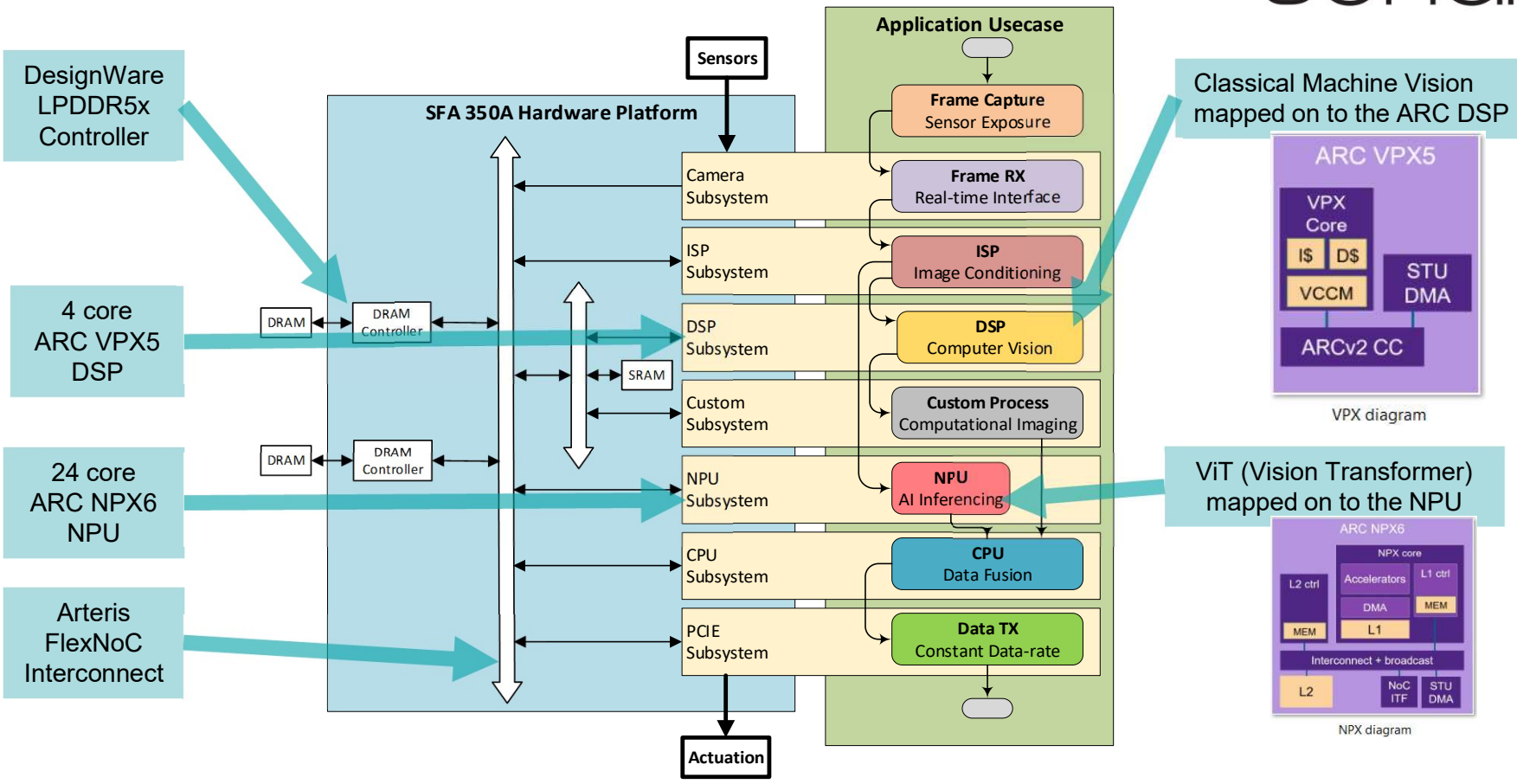
2. Application use-case:
   - Defines memory access patterns and computational behaviours needed to satisfy the SoC's functional requirements

Mapping Application Use-case to Hardware Platform



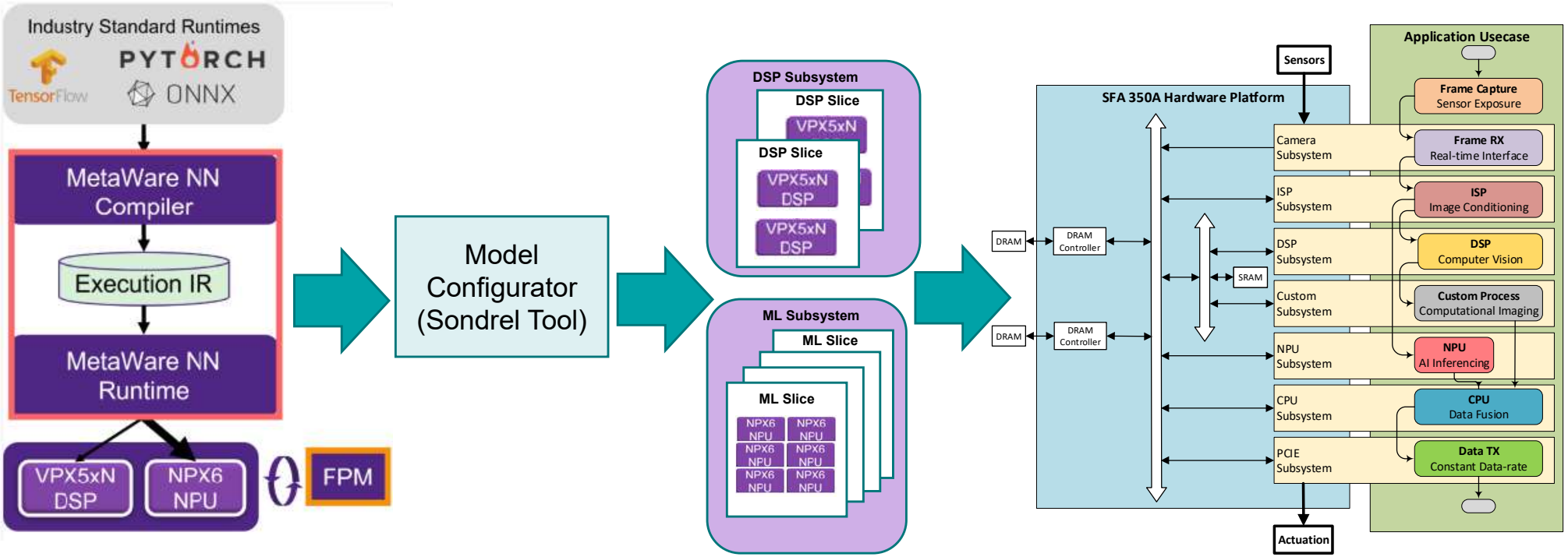
- 1 Process-to-resource mapping (i.e. hardware to one or more Use-cases)
- 2 Transactors between interconnect & memory subsystems
- 3 NPU & DSP generate burst transfers
  - ViT running on ARC NPX6 NPU
  - Computer Vision running on ARC VPX5 DSP

Sample Elaborated Architecture



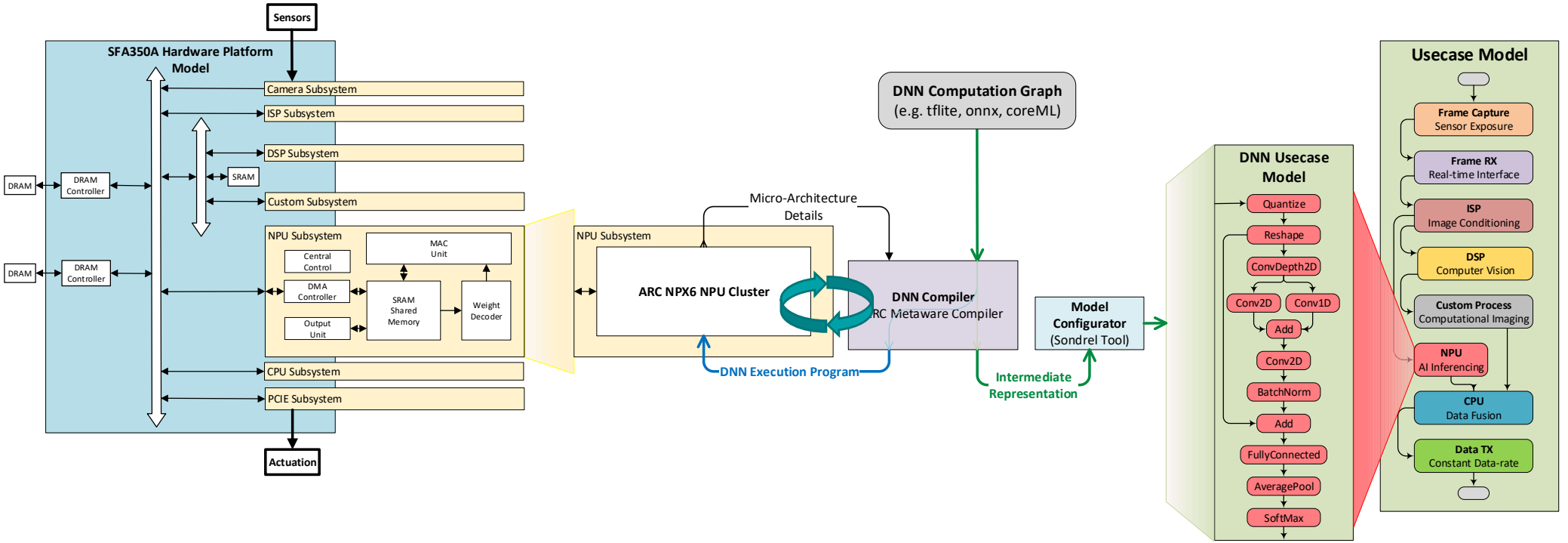
Tuning the workload model compute engines

- Mapping of the NN model onto the NPX6 NPU and VPX5 DSP is deduced from the MetaWare compiler's output

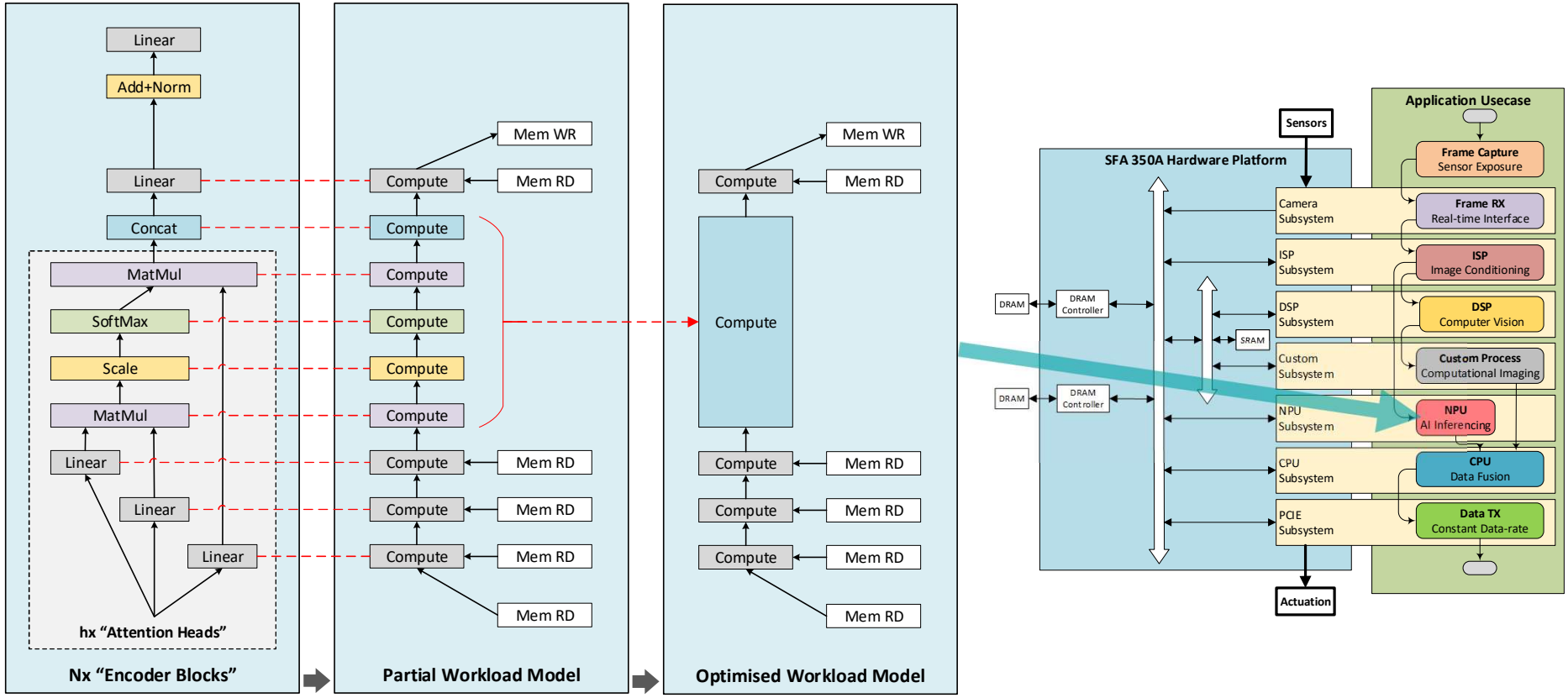


Optimizing Number of IP Compute Engine Cores (Trade-off: Die Area vs Performance)

sondrel



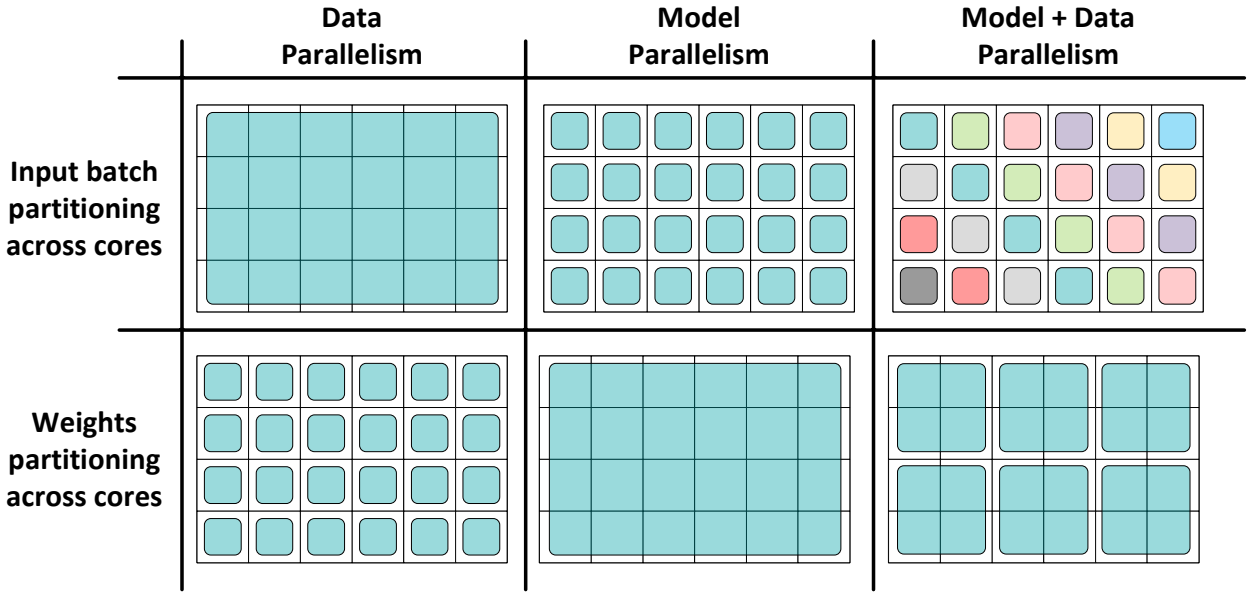
Workload Model for Vision Transformers



- MetaWare compiler simplifies the computation graph to arrive at an optimal workload structure

Workload Model Partitioning across NPX cores

sondrel

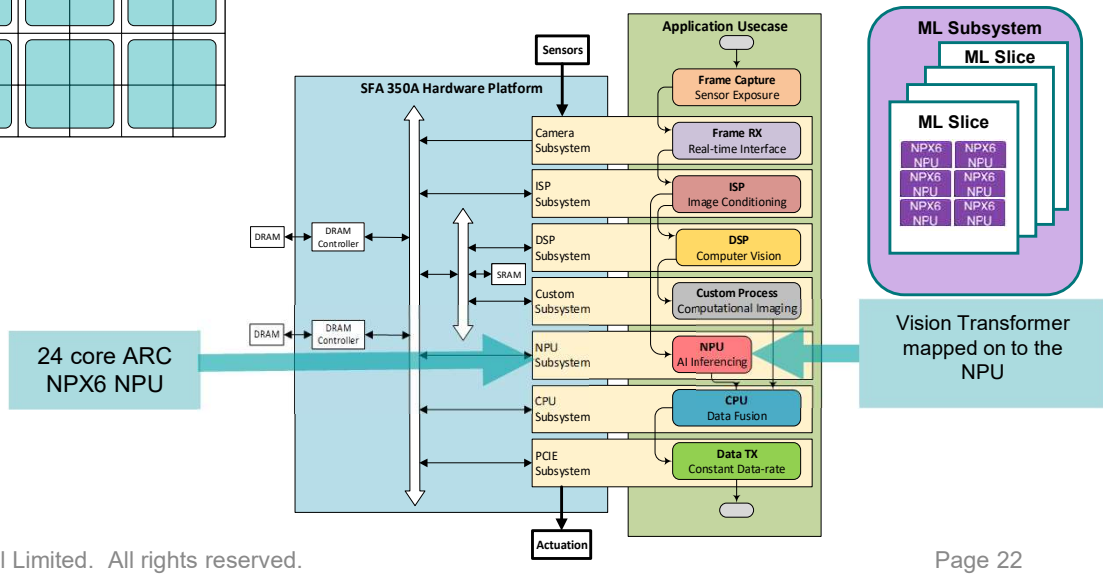


- Each tile == 1 NPX6 core (Total of 24 cores)
- Data partitioning varies depending on data type (feature-map or weights) and parallelism modality

Key:

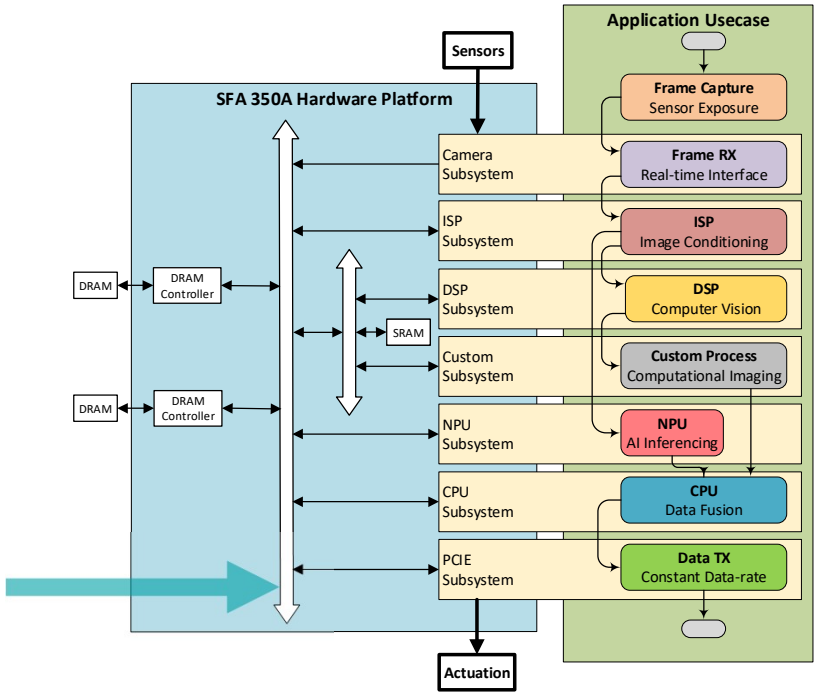
Data-set split 4 ways

4 copies of same dataset



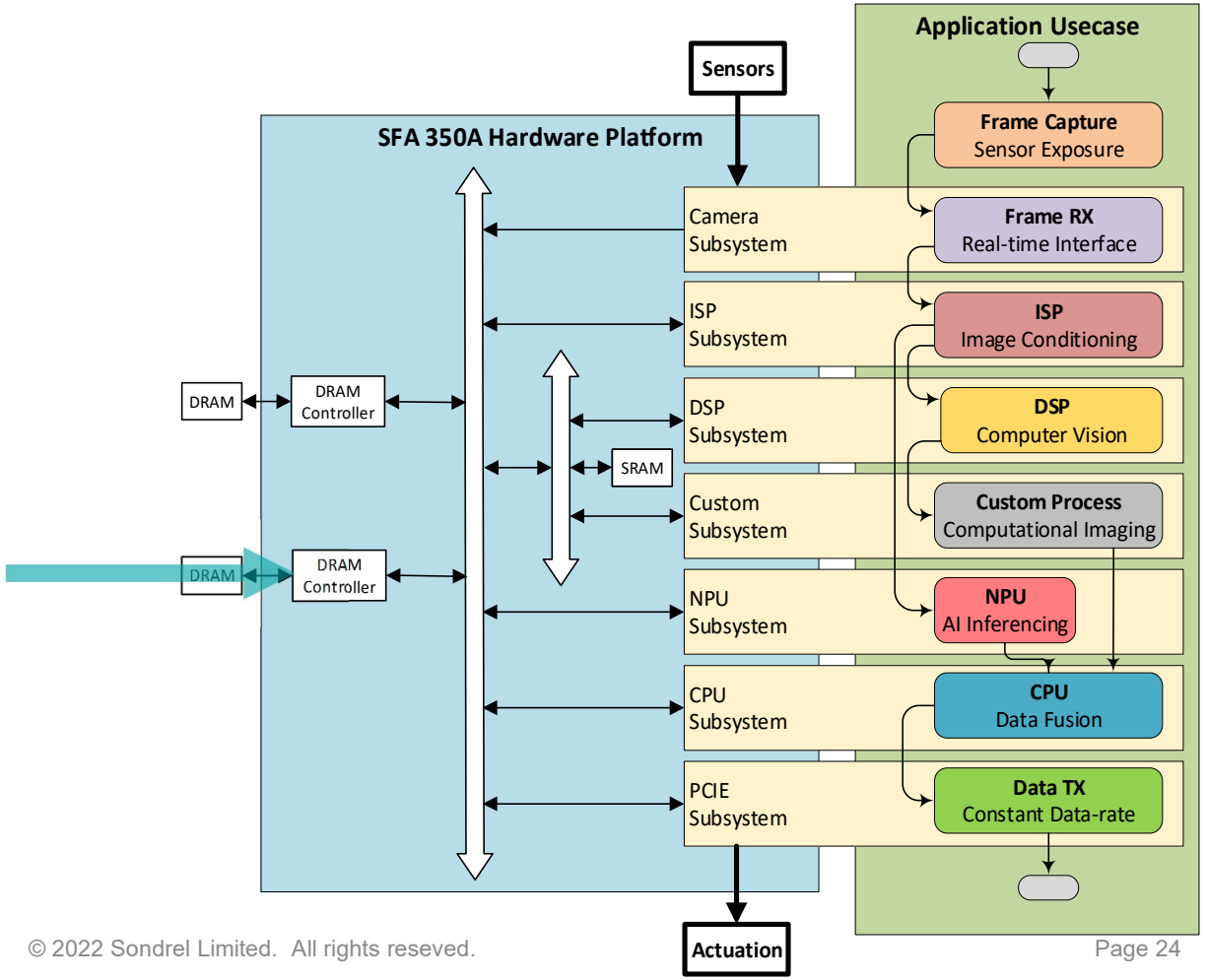
Tuning the FlexNoC Interconnect Performance

- Interface
  - Bus widths
  - Clock frequency
- Transport
  - Multi-layer Connectivity matrix
  - Switch topology
  - Outstanding transaction buffers
  - Wire serialisation
  - Pipeline stages
  - QoS limiter/regulator



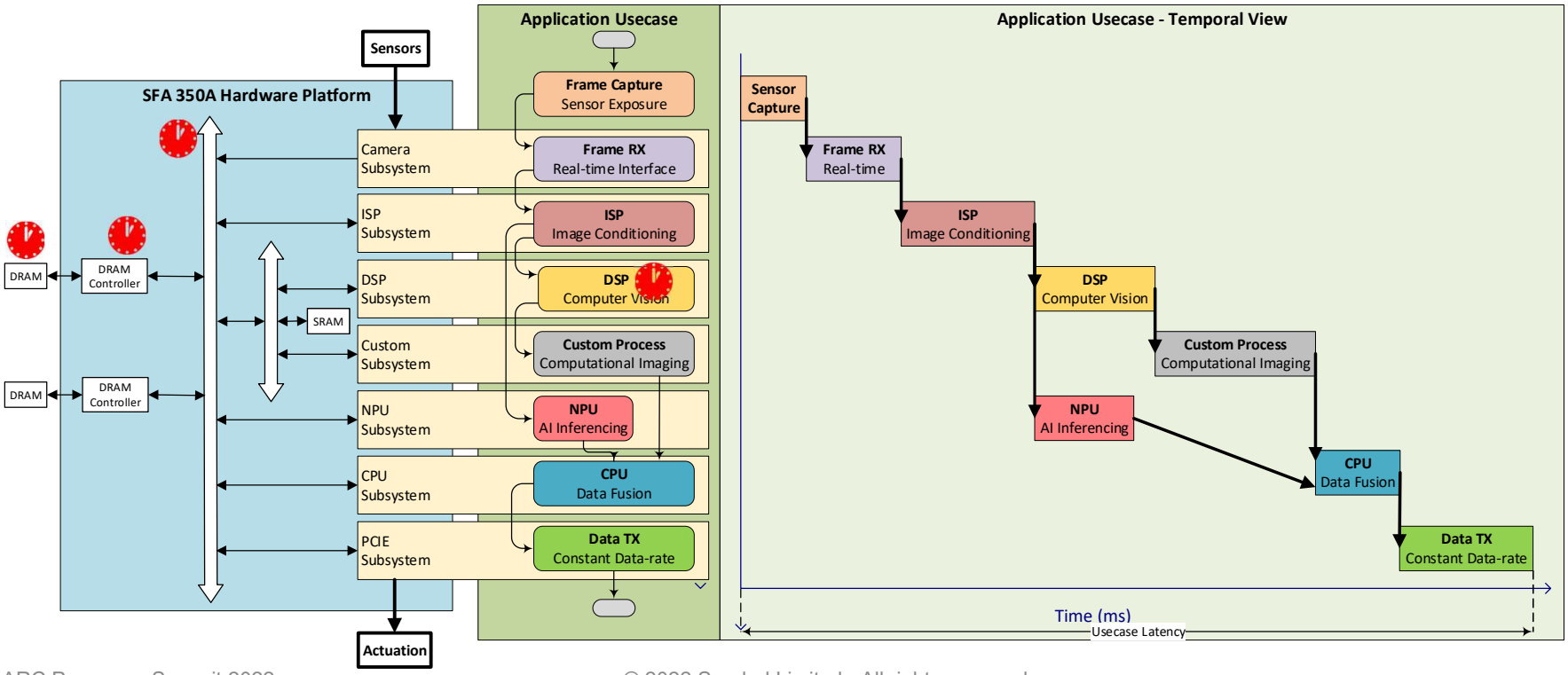
Tuning the LPDDR5x Controller's Performance

- Scheduler
  - CAM memory depth
  - Read/write priority
- Memory
  - Controller frequency
  - Burst length
  - Data width
  - Refresh/pre-charge
  - Row/bank/column mapping
  - Interleaving
- Interface
  - Memory bus width
  - Number of AXI slave interfaces
  - Queue depth
- Arbiter
  - Arbitration policy & delay



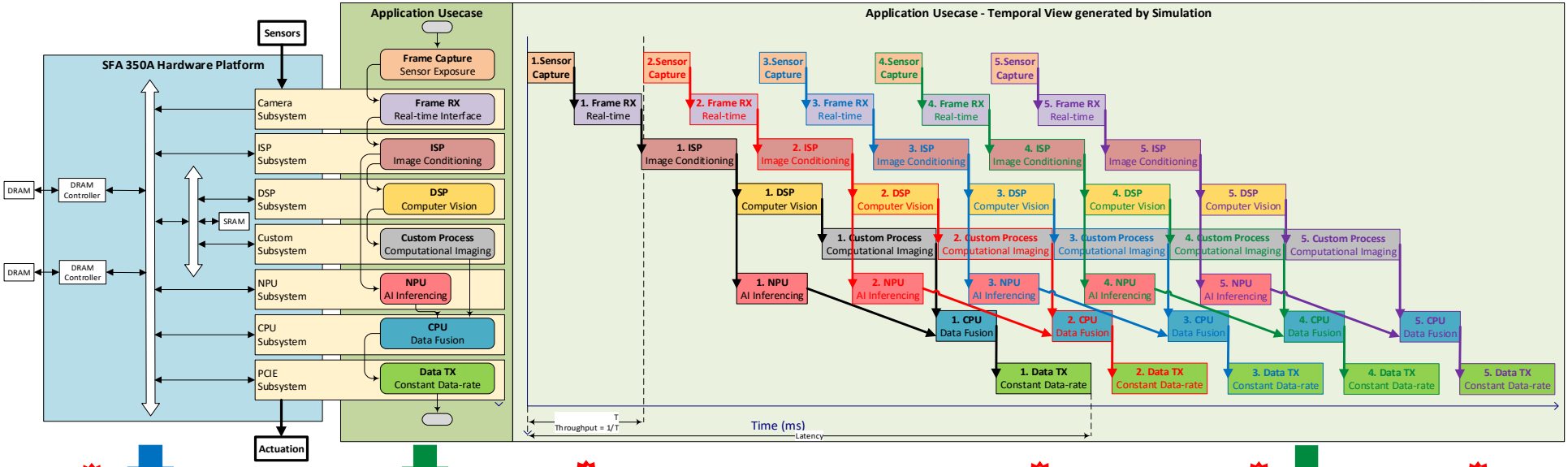
System Level Simulation for Architecture Exploration

- The temporal view in this example shows the time taken one traversal of the Use-case
- Width corresponds to latency of each process in the Use-case model. Depends on end-to-end system latencies
- Sources of latency: interconnect arbitration, memory controller timings, outstanding transaction limits



Overall Methodology for Architecture Exploration

sondrel



Platform Requirements

1. Initiator Requirements
   1.1. CameraSS must tolerate a max write transaction latency of __ ns
   1.2. CameraSS must support a max write throughput of __ MB/s
   1.3. ...
2. Interconnect Requirements
   2.1 Streaming interface must have __MB/s
3. DDR Requirements
   3.1 DDR data channel utilisation must be above 80% for main usecases
   3.2 ...

Usecase Requirements

1. Latency must be less than 4 frames
   1.1. Frame RX must complete in __ µs
   1.2. ISP must complete in __ µs
   1.3. CUSTOM must complete in 1 frame
   1.4. NPU must complete in under __ µs
   1.5. ...
2. Overall frame throughput must be at least __ frames/second
   2.1. FrameRX must achieve a throughput of __ MB/s over active period
      2.1.1. Each FrameRX line must achieve a throughput og __MB/s over a line active period
   2.2. ISP must achieve a throughput of __ MB/s over active period
   2.3. CUSTOM must achieve a throughput of __ MB/s over active period
   2.4. NPU must achieve a throughput of __ MB/s over active period
   2.5. ...

Usecase Key Performance Indicators (KPIs)

1. L ≤ 4 frames
   1.1. ...
   1.2. ...
   1.3. ...
   1.4. ...
   1.5. ...
2. TP ≥ __ frames/second
   2.1. ...
      2.1.1. ...
   2.2. ...
   2.3. ...
   2.4. ...
   2.5. ...

Simulation Metrics

1. ...
   1.1. ...
2. ...
   2.1. ...
      2.1.1. ...

Do simulation metrics staisfy KPIs?

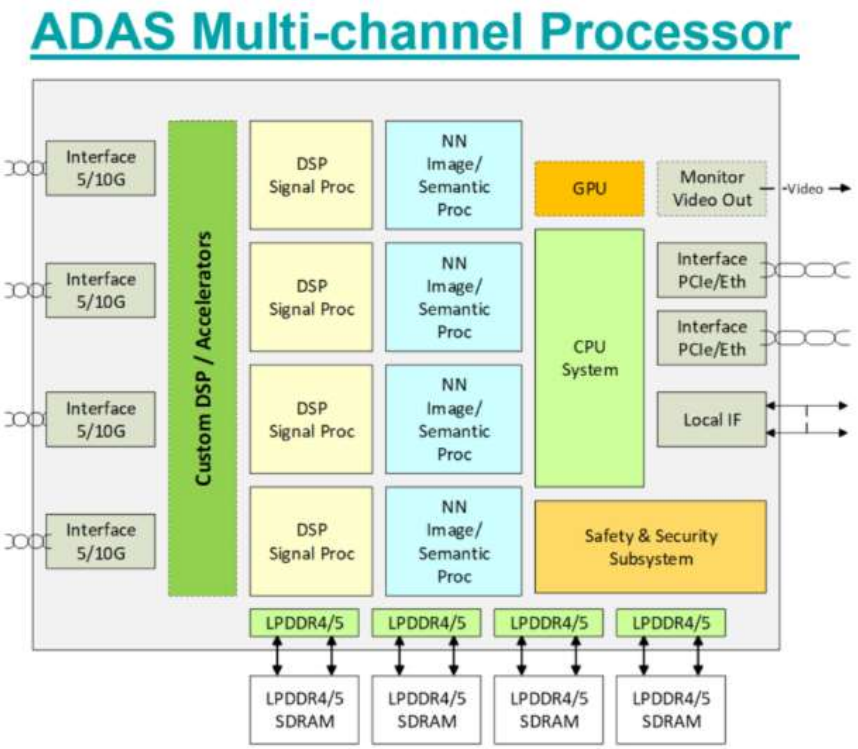
Results Summary Report

1. FAIL
   1.1. PASS
   1.2. PASS
   1.3. FAIL
   1.4. PASS
   1.5. FAIL
2. PASS
   2.1. PASS
      2.1.1. FAIL
   2.2. PASS
   2.3. PASS
   2.4. PASS
   2.5. PASS

Conclusion

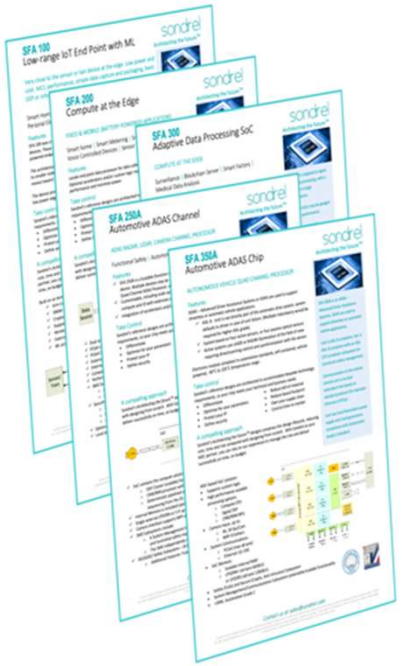
The Scalable Architecture Framework delivers ...

- Efficiently formalized architecture designs for SoCs
- Encapsulates use-case scenarios as “workload” models
- Has delivered 5 adaptable reference platform designs
- Proven methodology for architecture performance exploration for multiple neural net applications
- Aligns with Soc design goals of scalability and reusability together with an IP ecosystem



Scalable Architecture Framework – Customer Benefits

sondrel



Mitigate Risk

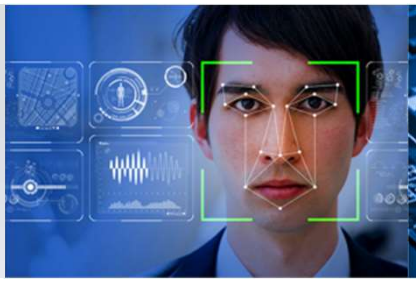
- Creating custom SoCs with subsystems & IP tuned to meet performance

Reduced Time-to-Market

- *Out-of-the-Box* extensible and scalable architectures available now

Lower Cost

- Expertly architected platform designs for today's trending applications



Contact

Carlos Román
Head of US ASIC Solutions
Architecture & Technical Sales

+1 610 844 5534
carlos.roman@sondrel.com

sondrel

Complexity delivered simply

www.sondrel.com