



# Simplifying Portable Computer Vision with OpenVX 2.0

**Kiriti Nagesh Gowda**  
**OpenVX Chair | The Khronos Group**  
**Sr. Staff Engineer | AMD**



# Agenda

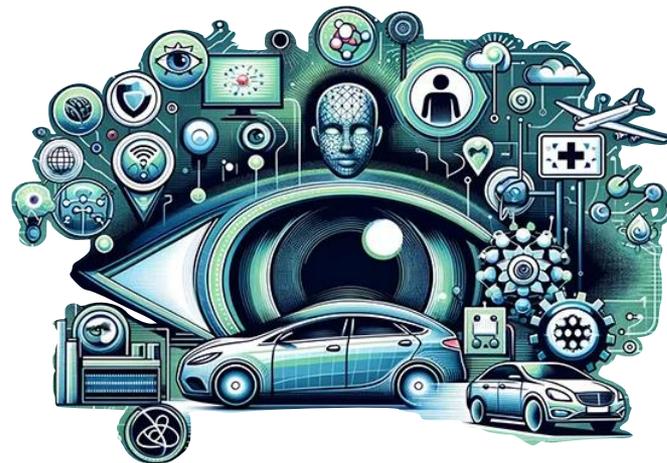
- Computer vision - it's increasingly relevant
- Computer vision API - OpenVX overview
- OpenVX 2.0 - what's new?
- Advantages of version 2.0
- Work group member use-cases
- Beyond vision
- Summary

# Computer vision - it's increasingly relevant?

# Computer vision - it's increasingly relevant

The idea that computer vision might be irrelevant likely stems from a few misconceptions, but here's why it's not just relevant—it's critical to the future of AI and automation.

- **CV is More Than Just "Classic" Image Recognition:**
  - Vision Transformers
  - Multimodal AI
- **Real-World Applications Are Exploding**
  - Autonomous Vehicles
  - Robotics
- **New Challenges = New Opportunities**
  - Efficient Edge AI
  - Video Understanding



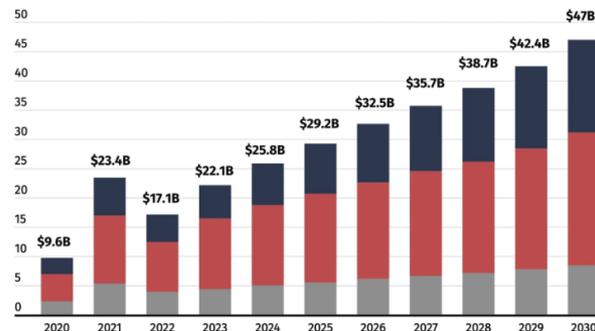
# Computer vision - it's increasingly relevant

- Computer vision is far from just a trend
- It's a foundational technology enabling AI to interact with the visual world.
- As long as cameras and visual data exist, CV will remain critical in shaping the future of automation, AI, and human-computer interaction.

## Computer Vision Market Growth

Market size of the computer vision market from 2020 to 2030, by segment (in billion U.S. dollars)

Source: Statista Market Insights



Source: <https://www.edge-ai-vision.com/2024/05/computer-vision-market-to-grow-by-81-and-hit-a-47-billion-value-by-2030/>

# Open, royalty-free standard for computer vision



OpenVX™ is an **open, royalty-free API** standard for a **cross-platform** acceleration of computer vision applications

High-level graph-based abstraction for **portable, efficient** vision processing

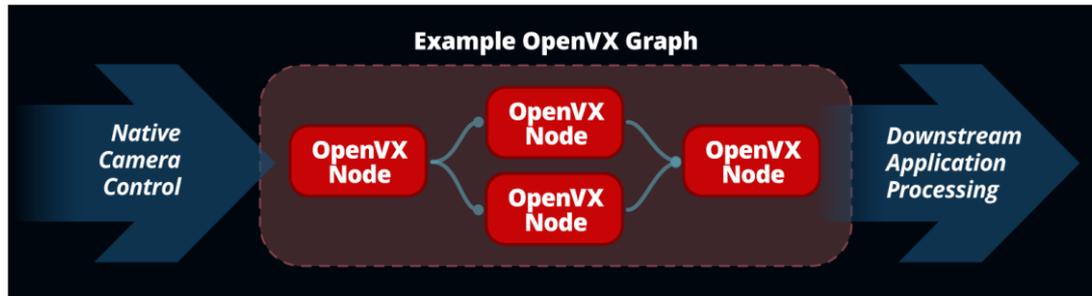
# Computer vision API - OpenVX overview

# Computer vision API - OpenVX current version



# Computer vision API - OpenVX current version

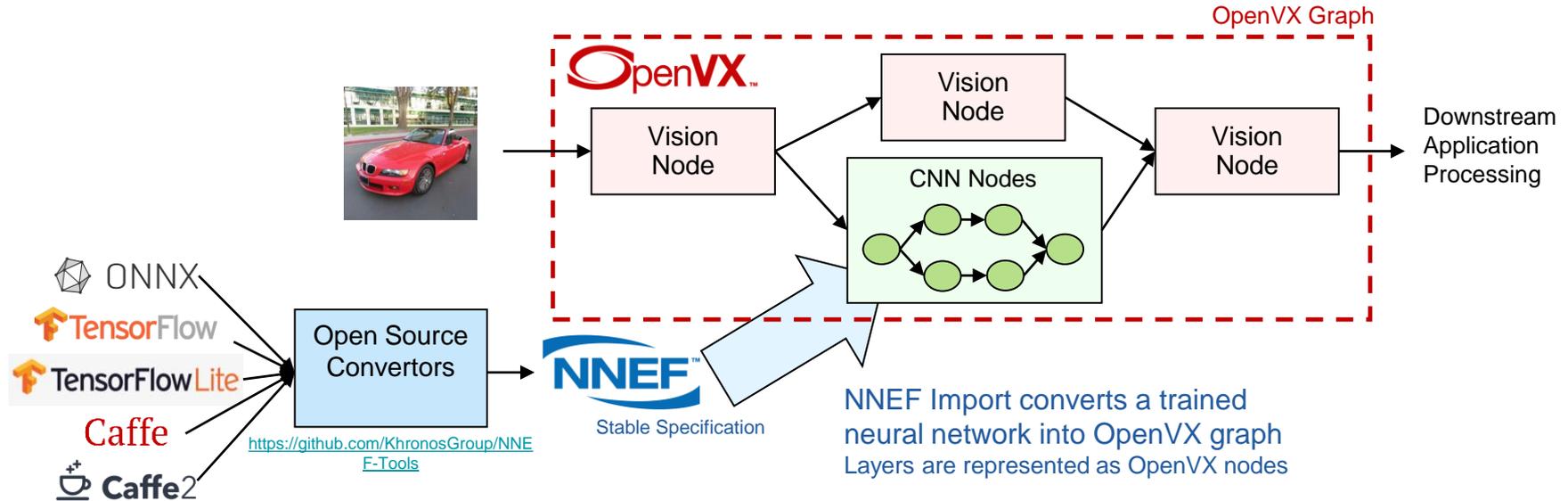
## OpenVX™ Graph Framework



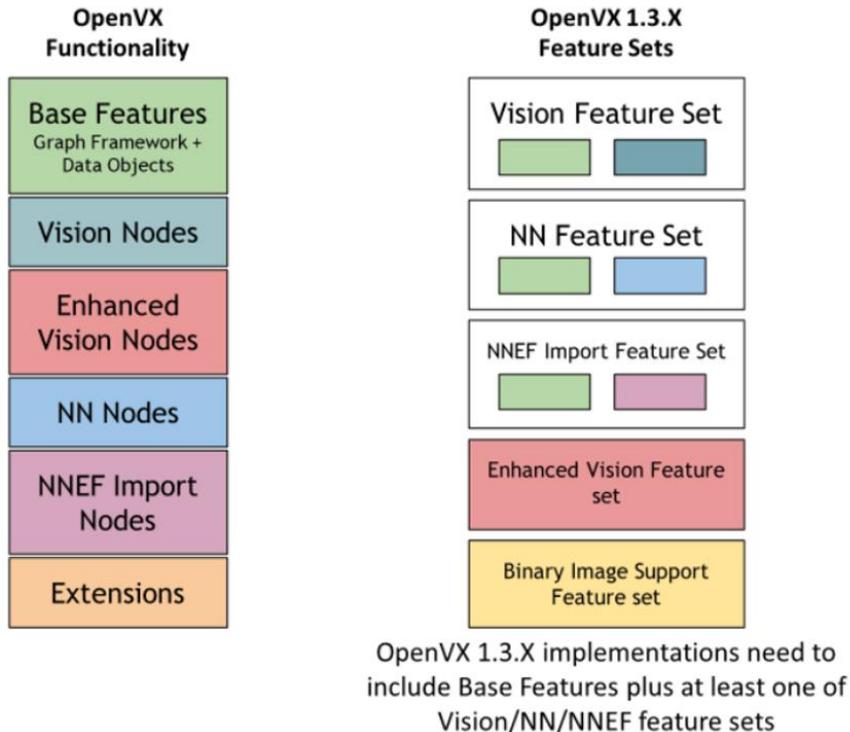
- Fuse nodes, when possible, to achieve better overall performance
- Auto graph-level memory optimizations to achieve a low memory footprint
- Deployed on a wide range of computer hardware, including small embedded CPUs, ASICs, APUs, discrete GPUs, and heterogeneous servers

# Computer vision API - OpenVX current version

**OS Support:** Linux, Windows, Android, iOS, Raspbian, Embedded OS, no OS

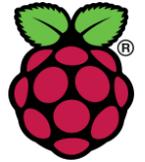


# Computer vision API - OpenVX current version



# Computer vision API - OpenVX current version

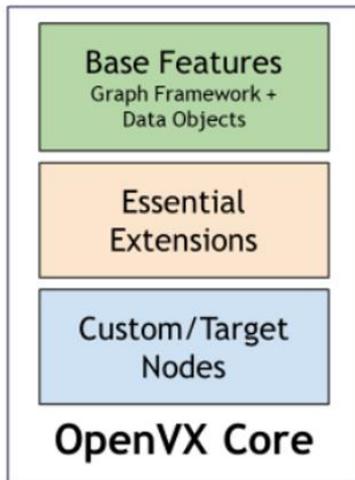
Conformant implementations of OpenVX from the following vendors



# OpenVX 2.0 - What's new?

# OpenVX 2.0 - What's new?

By simplifying and reducing the conformance process 2.0 focusses on the framework feature **ONLY**

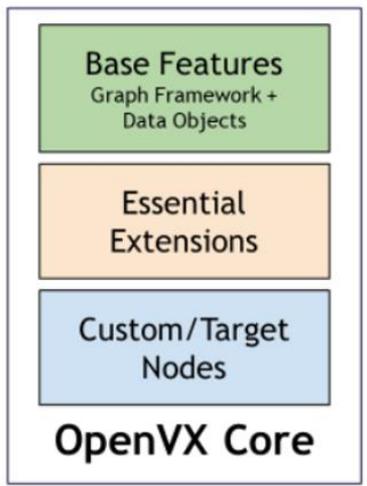


- **Core Framework Improvements**
- **Focus on User-Defined and Generic Workloads**
- **Low Barrier to Entry**
- **Extensive Conformance Testing**

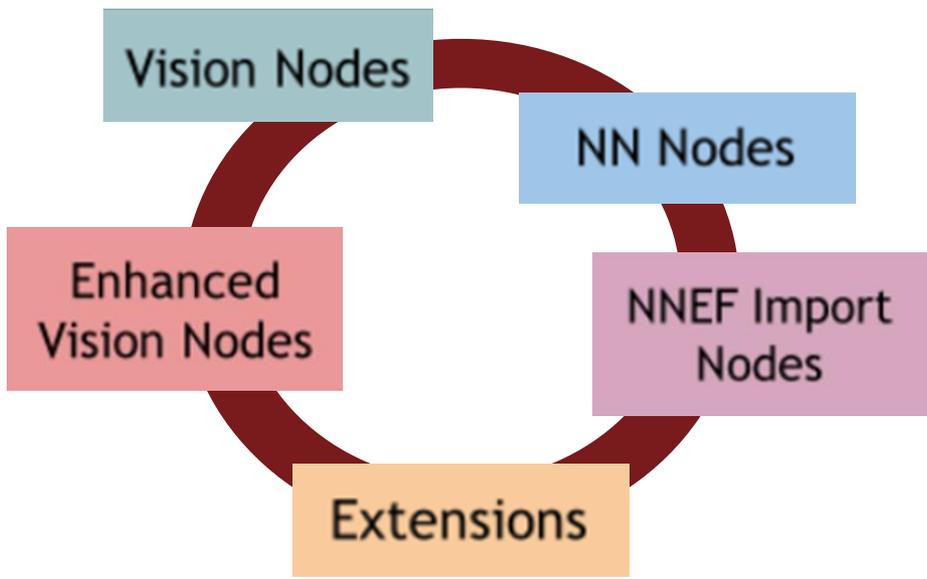
# OpenVX 2.0 - What's new?



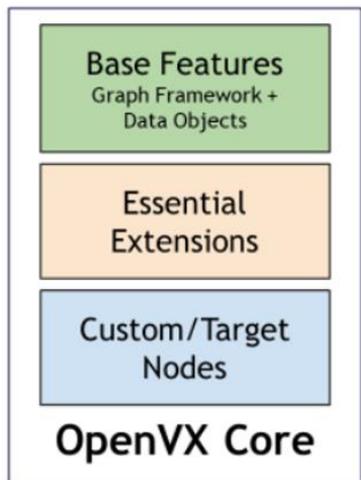
Conformance reached by fulfilling the core API **only**



Vendor can choose (or not) the extension of its choice and submit conformance test results (if existing)



# OpenVX 2.0 - What's new?



## Essential Extensions

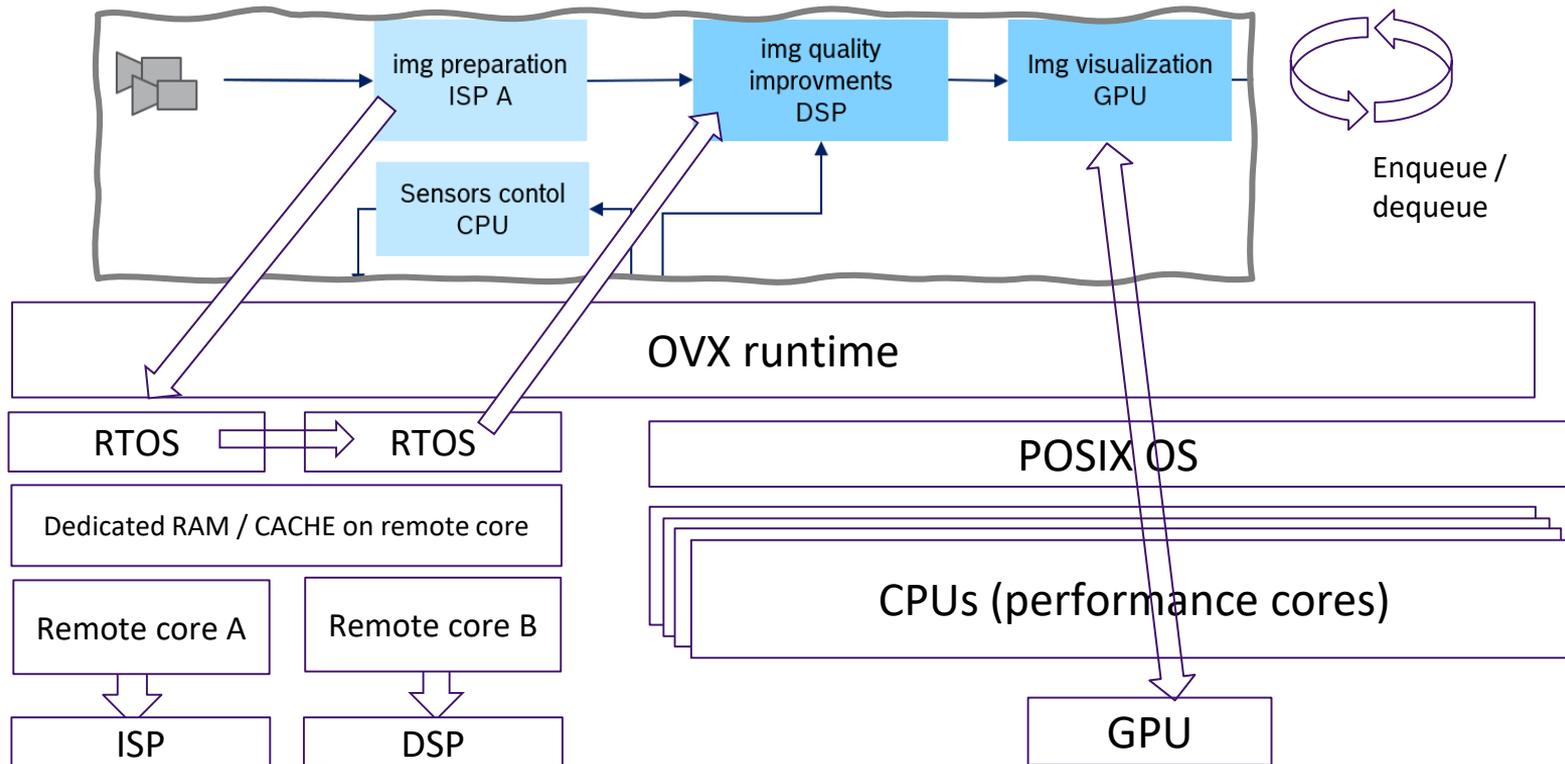
- OpenVX KHR Supplementary Data
- OpenVX KHR Safe Cast
- OpenVX KHR Bidirectional Parameters Extension
- OpenVX KHR Swap And Move kernel
- OpenVX KHR Pipelining version 2.0

# Advantages of version 2.0

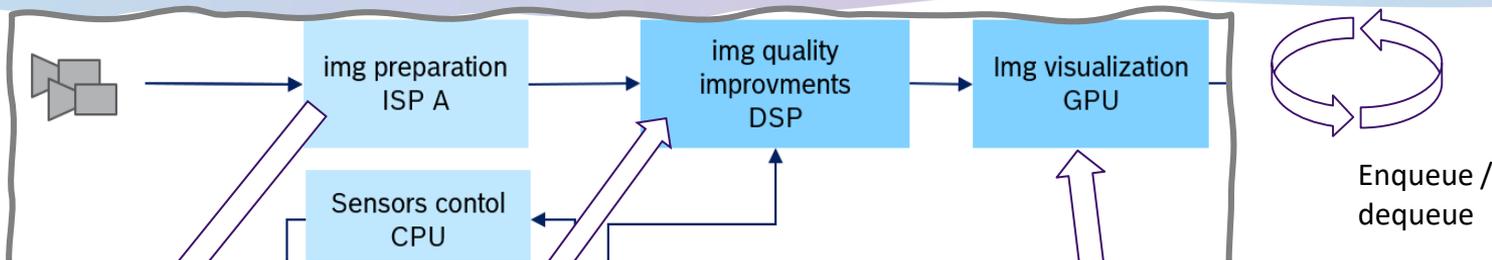
# OpenVX 2.0 - Advantage

- **No Soc lock-in**
  - Ability to (fast) transition to a new hardware platform based on requirements.
  - Flexibility to switch (fast) to a new SoC, enabling cost savings and dual sourcing.
  - Provides a platform to showcase hardware capabilities within the same ecosystem.
- **Customer and third-party software integration**
  - Facilitates integration by providing a standardized interface.
  - Library Integration: Enables validation and testing processes for library integrations.
  - Reduced Complexity: Mitigates and reduces integration complexity
- **Cross-Generation Compatibility**
  - Builds and capitalizes across system generations by providing mechanisms and standards to avoid reinventing the wheel when switching to another SoC

# OpenVX 2.0 - Advantage



# OpenVX 2.0 - Advantage



- **Decomposing data pipelines into nodes allows for:**
  - Supplier Flexibility: Nodes can be delivered by different suppliers.
  - Independent Validation: Each node can be validated separately, outside of the entire graph.
  - IP Protection: IP can be protected by using binaries if necessary.
- **No Soc lock-in:**
  - Pipelines (graphs) remain the same across different SoC (same “functional” need)
- Only kernel implementation differs (if needed)

# OpenVX 2.0 - Advantage

- Every applications (and beyond) should
  - Meet safety and security goals
  - Be portable
  - Be robust and reliable
  - Offer mechanisms for error detection and recovery
- Requirements are complex and time consuming to develop!
- Reusing these concepts across SoC and system generations is a smart approach.

You can have the best AI algorithm but if the data is not available, it does not help

**OpenVX provides the API and a stable platform to achieve this**

# OpenVX 2.0 - Advantage

- OpenVX features that help achieve:
- **Safety and security goals**
  - Lightweight Implementation: Suitable for ASIL validation.
  - Conformance Tests: Ensure the robustness and correctness of the framework.
  - Clear Separation: Clear split between the framework and the application, ensuring freedom from interference.
- **Reliability**
  - Standardized API: Stable foundation, enabling code improvement across product generations and SoCs.
  - Error Detection and Recovery: Various mechanisms for error detection and recovery, including node errors, graph timeouts, latency checks, and node diagnostics.
  - Custom Extensions: You can create your own extensions and propose them to the Working Group!

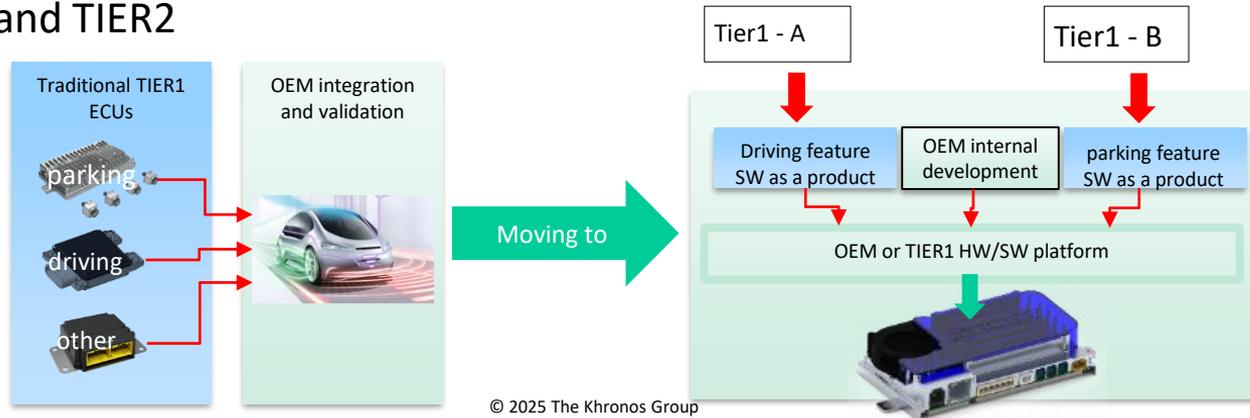
# Work group member use-cases

# Work group member use-cases



## A New business Model for the Automotive Industry

- Over the past few years and the emergence of the **SW defined vehicle**, the market has shifted from multiple dedicated (black-box) ECUs for parking, driving, and other functions to a centralized ECU that consolidates these functionalities.
- This transition has led to a restructuring of the development and integration models for the OEM / TIER1 and TIER2





## Bosch Adaptation to this New Situation

### Tailored computing solution

Our approach enables customer tailored computing solutions based on the underlying modular construction kit



### No SoC lock-in

Based on our flexible electronic concept we can integrate SoCs of various vendors to avoid SoC lock-in effect



### ECU and system competence

We as Bosch bring in comprehensive know-how and series-proven experience both on ECU and system level



### Modular electro-mechanic concept

Based on our modular electro-mechanical concept we can offer single ECU solutions as well as multi-ECU rack integration



### Customer and 3rd party SW support

Our flexible integration platform supports Bosch SW as well as customer and 3rd party SW solutions



# Work group member use-cases



## How does OpenVX help TIER1 and OEMS ?



**Tailored computing solution**  
Our approach enables customer tailored computing solutions based on the underlying modular construction kit

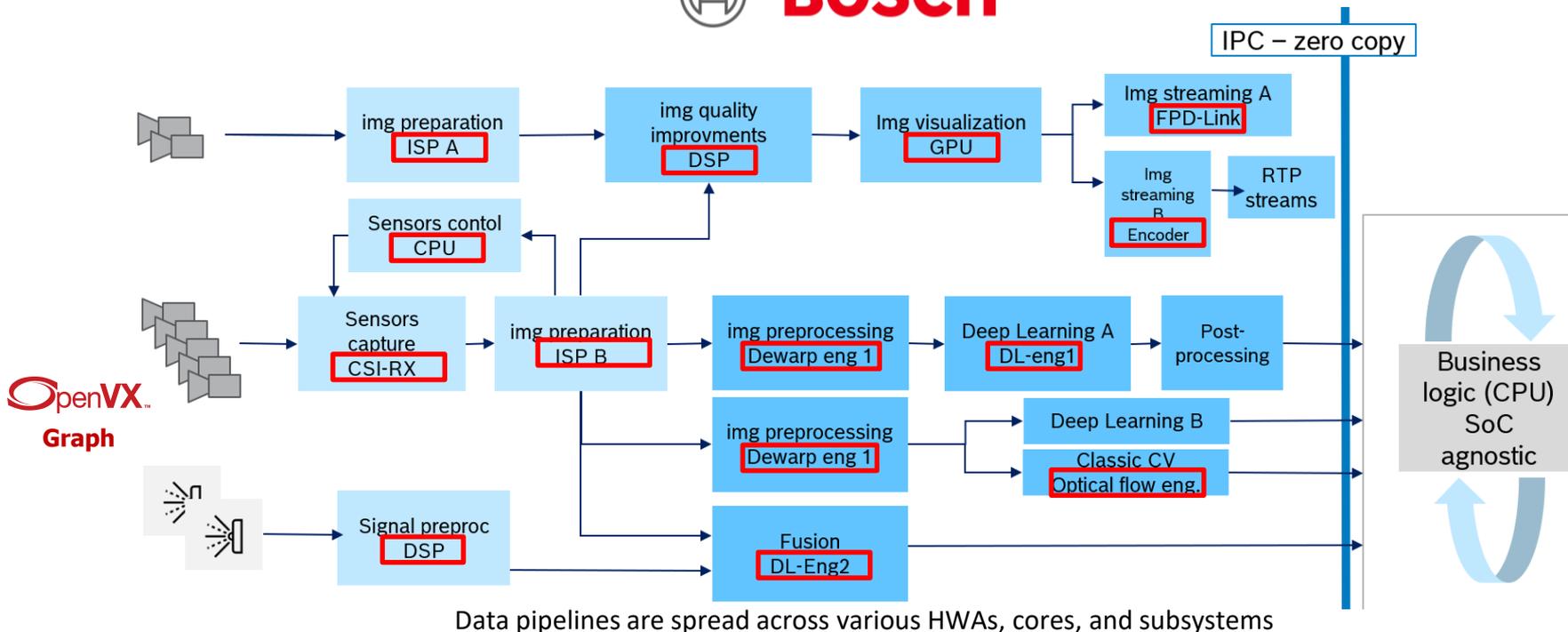
**OpenVX™ for electro-mechanic concept**  
Based on our modular electro-mechanical concept we can offer single ECU solutions as well as multi-ECU rack integration

**2 No SoC lock-in**  
Based on our flexible electronic concept we can integrate SoCs of various vendors to avoid SoC lock-in effect

**1 Customer and 3rd party SW support**  
Our flexible integration platform supports Bosch SW as well as customer and 3<sup>rd</sup> party SW solutions

**ECU and system competence**  
We as Bosch bring in comprehensive know-how and series-proven experience both on ECU and system level

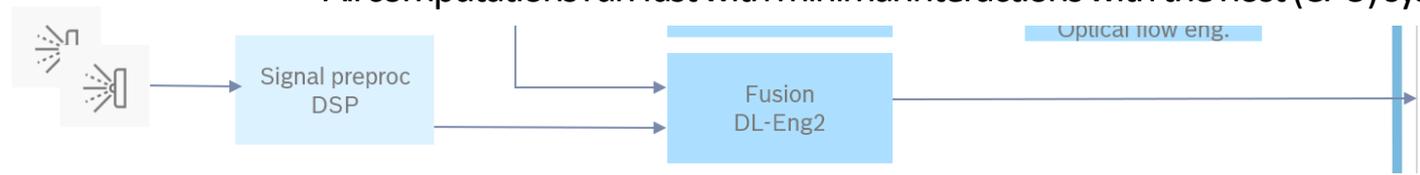
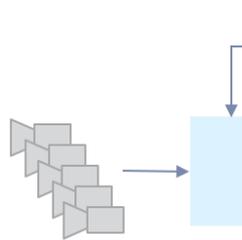
# Work group member use-cases



# Work group member use-cases



- Each block is a compute unit
  - Uses different hardware accelerators
  - Runs on host or different remote systems
- All blocks/nodes may run in parallel and form a graph
- All computations run fast with minimal interactions with the host (CPU) system

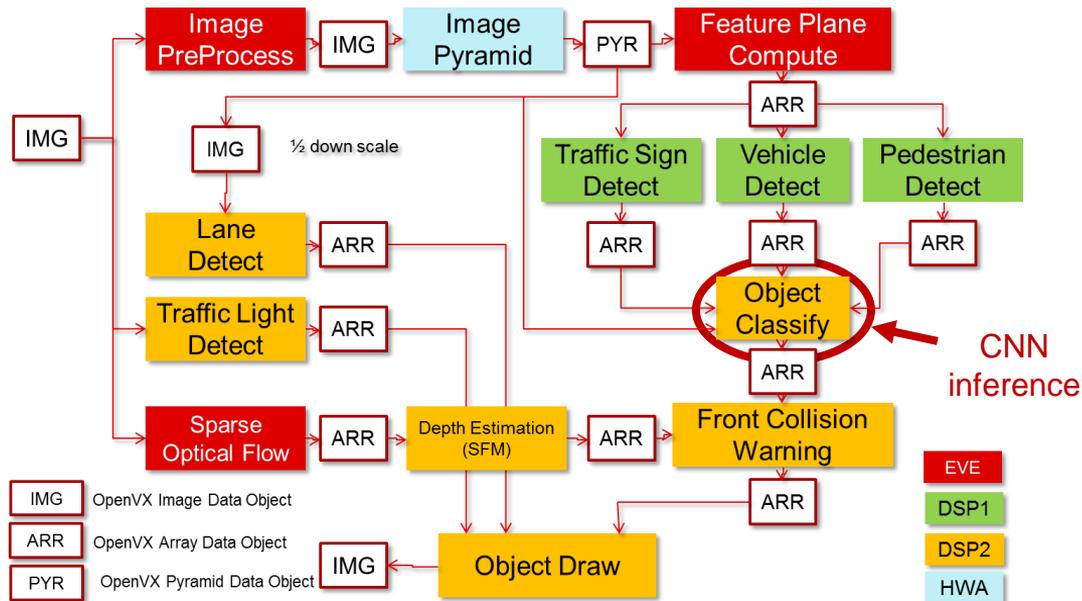


# Work group member use-cases



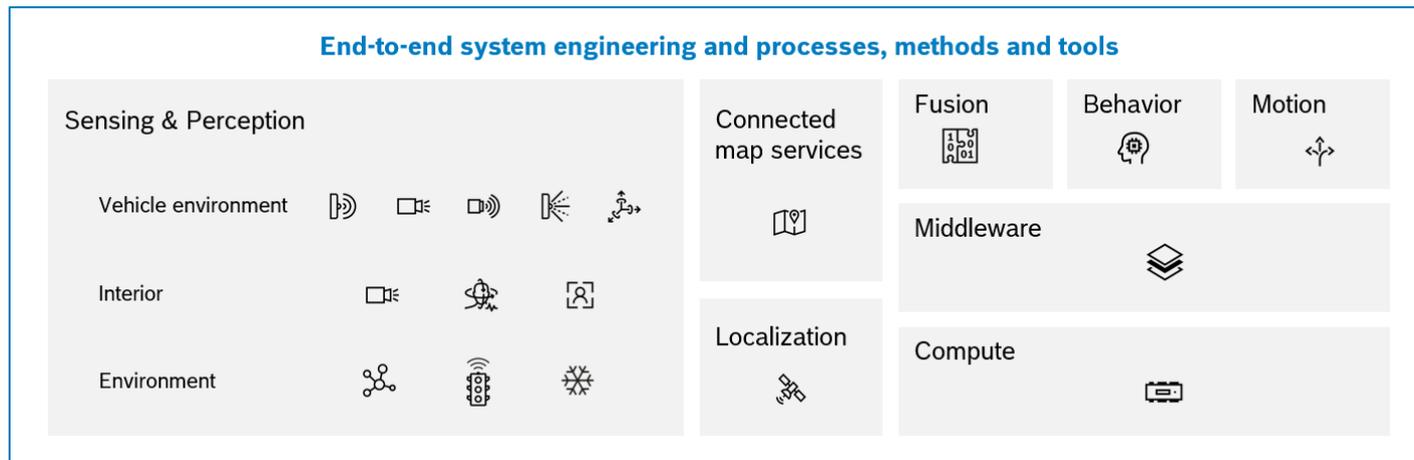
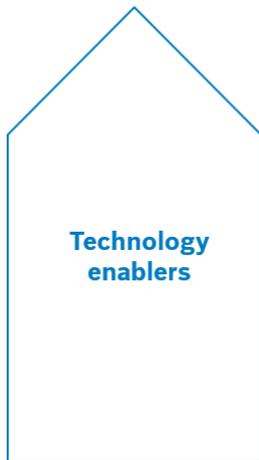
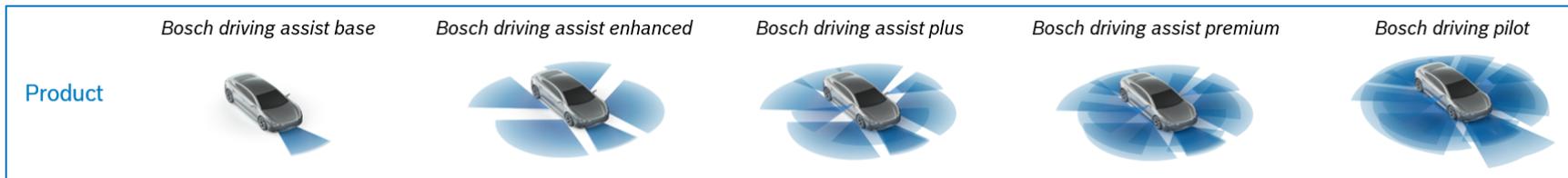
## Application: Front camera ADAS

OpenVX™  
Graph



# Beyond vision

# Beyond vision



Complex sensors combinations and scenarios!

**To summarize**



- OpenVX is unique in being the only vision API shipped as an optimized driver
- OpenVX delivers performance comparable to hand-optimized, non-portable code
- Acceleration on a wide range of vision hardware architectures
- OpenVX provides a high-level graph-based abstraction
  - Enables graph-level optimizations
  - Can be implemented on almost any hardware or processor
- **Portable, Efficient Vision Processing!**

# Acknowledgement

## Thanks To

- Mike Schmit - Director of Software Engineering, AMD
- Neil Trevett - President, The Khronos Group
- AMD's MIVisionX Team
- OpenVX Working Group
- Embedded Vision Summit 2025 Organizers

## OpenVX Resources - Khronos

### Sample Implementation:

<https://github.com/KhronosGroup/OpenVX-sample-impl>

### Sample Applications:

<https://github.com/KhronosGroup/openvx-samples>

### Tutorial Material:

[https://github.com/rgiduthuri/openvx\\_tutorial](https://github.com/rgiduthuri/openvx_tutorial)

## Conformant Implementations

<https://www.khronos.org/conformance/adopters/conformant-products/openvx>

## Khronos OpenVX API Registry

<https://www.khronos.org/registry/OpenVX/>

## OpenVX for Raspberry Pi

<https://www.raspberrypi.org/blog/openvx-api-for-raspberry-pi/>

## AMD ROCm MIVisionX - OpenVX

<https://gpuopen-professionalcompute-libraries.github.io/MIVisionX/>

# Disclaimers and attributions

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

**THIS INFORMATION IS PROVIDED 'AS IS.' AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.**

© 2025 Advanced Micro Devices, Inc. All rights reserved.

AMD, the AMD Arrow logo, EPYC, Radeon, ROCm and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

AMD

The AMD logo symbol is a stylized, geometric shape composed of black and white elements. It features a central white square with a smaller white square inside it, creating a nested effect. The surrounding black shapes are angular and interlocking, forming a square-like overall structure with sharp, pointed corners.