

The logo for the 2024 Embedded Vision Summit is centered within a white octagonal shape. The octagon is surrounded by a colorful, multi-layered border of overlapping geometric shapes in shades of purple, blue, green, yellow, and orange. The text inside the octagon reads "2024 embedded VISION SUMMIT" in a sans-serif font. "2024" is in a smaller font size above "embedded". "VISION" is in a large, bold, dark blue font with a gradient effect. "SUMMIT" is in a smaller font size below "VISION".

2024
embedded
VISION
SUMMIT®

Advancing Embedded Vision Systems: Harnessing Hardware Acceleration and Open Standards

Neil Trevett
President
Khronos Group

KHRONOS
GROUP

- The need for parallel offload of vision and inferencing processing
- How open standard APIs add value for embedded vision
- Introduction to Khronos and its family of open API standards
- Khronos and Safety Critical APIs
- Introduction to the new Kamaros Embedded Camera API in development
- How to get involved!

Khronos Connects Software to Silicon



KHRONOS[®] GROUP

Open, royalty-free interoperability standards to harness the power of GPU, XR and multiprocessor hardware

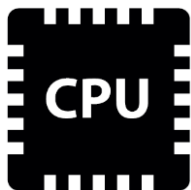
3D graphics, augmented and virtual reality, parallel programming, inferencing and vision acceleration

Non-profit, member-driven standards organization, open to any company

Proven multi-company governance and Intellectual Property Framework

Founded in 2000

~ 200 Members | ~ 40% US, 30% Europe, 30% Asia

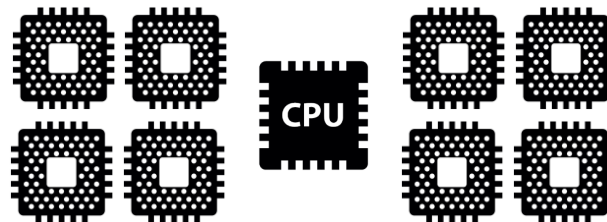


Single Processor

Simple to program *but may not*
provide enough performance

especially

As Moore's Law frequency/power
scaling is slowing



Multi-Processor

Additional processors can process expanded workloads
but adds complexity to system design and programming:

- (i) Distribute workload across processors
- (ii) Synchronize use of compute and memory resources
- (iii) Communicate intermediate data and results

Can open standards help solve this complexity?

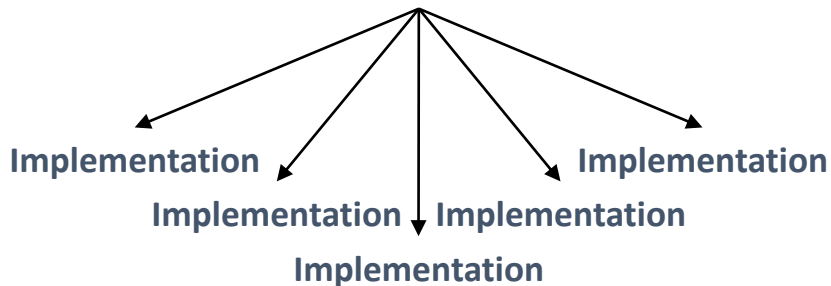
What is an Open Interoperability Standard?

INTEROPERABILITY standards define precise **COMMUNICATION**

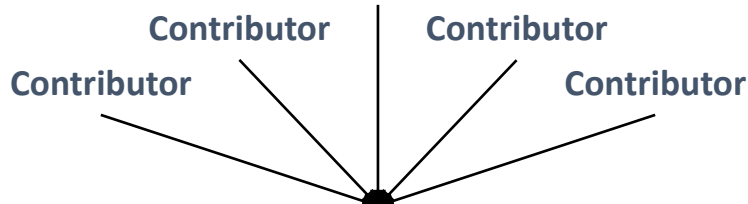
E.g., software to hardware, client to server

OPEN standards are created through multi-company cooperation

**Open Standard =
Shared Specification**



Contributor



**Open Source =
Shared Implementation**

Open standards enable **MULTIPLE IMPLEMENTATIONS** that can meet the needs of diverse markets, price points, and use cases - with rigorous conformance testing to ensure cross-vendor consistency

Open standards often use open source to share the implementation effort for sample implementations, tools, samples, conformance tests, validators, etc.

Open Standard Why's, When's and How's

| | | |
|--------------|--|--|
| Why? | <p>Grow Markets Reduced consumer confusions and increased capabilities and usability</p> | <p>Reduce Costs Shared development between many companies and driving volume</p> |
| | <p>Speed Time to Market Well-proven functionality, testing and interoperability</p> | <p>Enable Innovation Free resources to compete on quality, performance, power etc.</p> |
| When? | <p>When Technologies are Proven Avoid research and development by standards committee</p> | <p>Consensus Need The downsides of not having a standard are obvious</p> |
| How? | <p>Multi-company governance Avoid single-company control or dependency</p> | <p>Well-defined IPR Policy Royalty-free standards have clearer path to wide adoption</p> |
| | <p>With Extensibility Enable a responsive innovation pipeline to meet customer and market needs</p> | <p>Thoughtful Abstraction Encourage innovation through freedom to innovate implementation details</p> |

- **Cross-platform software reusability**
- **Decoupled software and hardware for easier development and integration**
 - **Cross-generation reusability and field upgradability**
 - **Developer training and familiarity**
- **Shared development of an industry-wide ecosystem of tools and libraries**

Open standards can provide proven and accessible languages and APIs to enable effective parallel processing and hardware acceleration

Khronos Active Standards

3D Graphics
Desktop, Mobile, Web



3D Asset
Authoring/Delivery



Portable XR
Augmented and
Virtual Reality



Parallel Computation
Vision, Camera, Inferencing,
Machine Learning



Safety Critical APIs

Khronos standards most relevant to the embedded vision industry

Khronos Compute Acceleration Standards

**Higher-level
Languages and APIs**
Streamlined development
and performance portability



Accelerated Single
source C++
programming



Neural Network
Exchange Format
Trained Networks



Graph-based vision
and inferencing
acceleration



Third party vision,
streaming and
inferencing libraries

**Lower-level
Languages and APIs**

Explicit hardware control



GPU rendering +
compute acceleration



GPU

Shaders



Intermediate Representation
(IR) compiler target supporting
parallel execution and graphics

Kernels



Heterogeneous
compute acceleration



FPGA

CPU

GPU

DSP

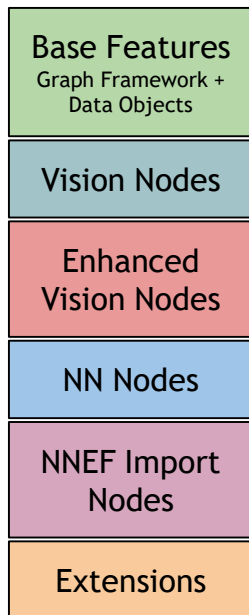
Custom HW

AI/Tensor HW

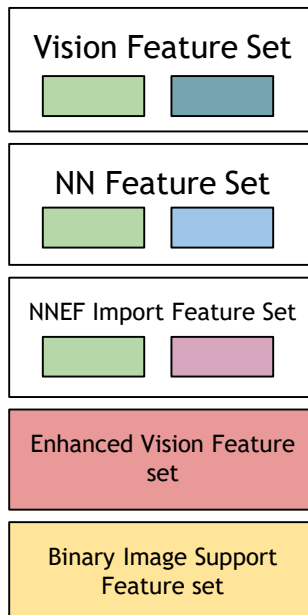
Multiple programming abstractions to meet the needs
of diverse software stack architectures

OpenVX Partitioning

OpenVX Functionality



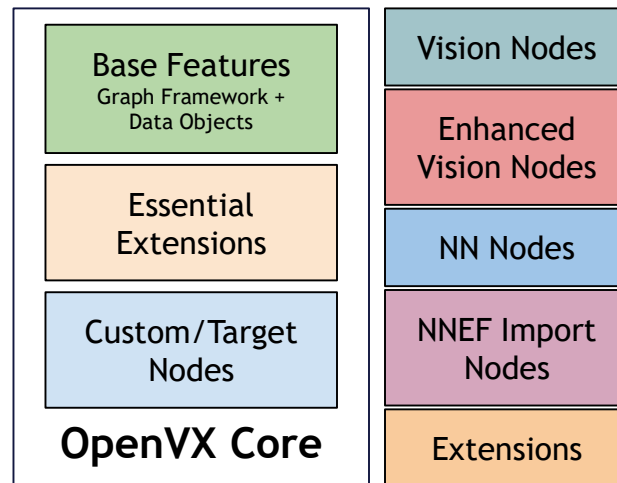
OpenVX 1.3.X Feature Sets



OpenVX 1.3.X implementations need to include Base Features plus at least one of Vision/NN/NNEF feature sets

OpenVX 2.0 Core + Optional Extensions

Focus on the OpenVX Graph Framework
Flexible pipelined data flow through target hardware
Seamless custom kernel support
Extensions e.g., for vision/radar/lidar processing



OpenVX 2.0 implementations need to include just OpenVX Core

Growing Need for APIs for Functional Safety

Safety-critical APIs reduce system-level certification effort where functional safety is paramount

- 1) Streamlined to reduce documentation and testing surface area
- 2) Deterministic behavior to simplify system design and testing
- 3) Unambiguous and comprehensive fault handling



1990s - Avionics

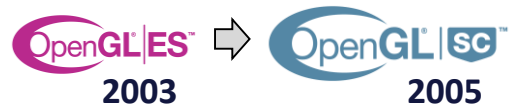


2010s - Automotive



2020s - Autonomous vehicles and devices, avionics, medical, industrial, and energy

Khronos has 20 years experience in adapting safety-critical API from proven mainstream APIs



The Need for a Camera System API Standard

Increasing Sensor Diversity

Including camera arrays and depth sensors such as Lidar



Multiple Sensors Per System

Synchronization and coordination become essential



The cost and time to integrate and utilize sensors in embedded systems has become a major constraint on innovation and efficiency in the embedded vision market

Increasing Sensor Processing Demands

Including inferencing. Sensor outputs need to be flexibly and efficiently generated and streamed into acceleration processors

Proprietary APIs Hinder Innovation

Vendor-specific APIs to control cameras, sensors and close-to-sensor ISPs prevent access of full camera capabilities

What is Kamaros?

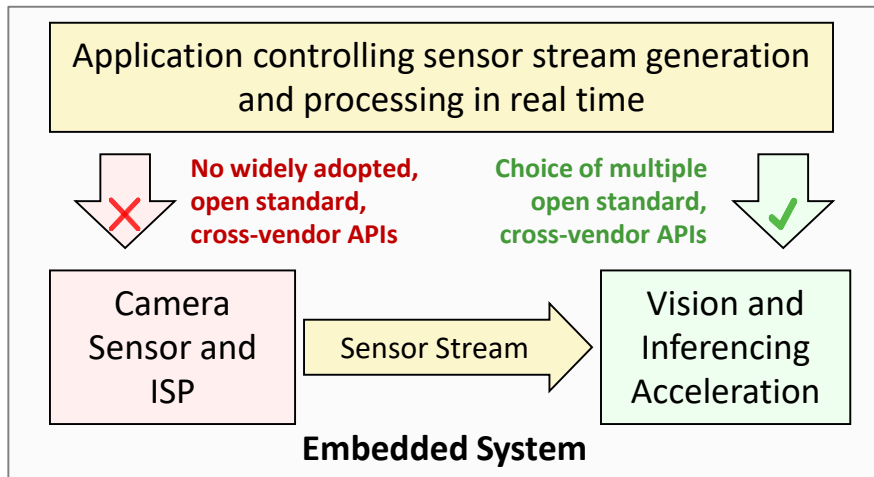


EMBEDDED CAMERA SYSTEM API

Jointly promoted by Khronos and the European Machine Vision Association (EMVA), the Kamaros API Working Group is developing an open, royalty-free standard for controlling camera system runtimes in embedded, mobile, industrial, XR, automotive, and scientific markets

Need for Embedded Camera API Standard

An effective open, cross-vendor open standard for camera, sensor and ISP control is a major ecosystem gap compared to processing APIs



Benefits of a Cross-vendor Embedded Camera API

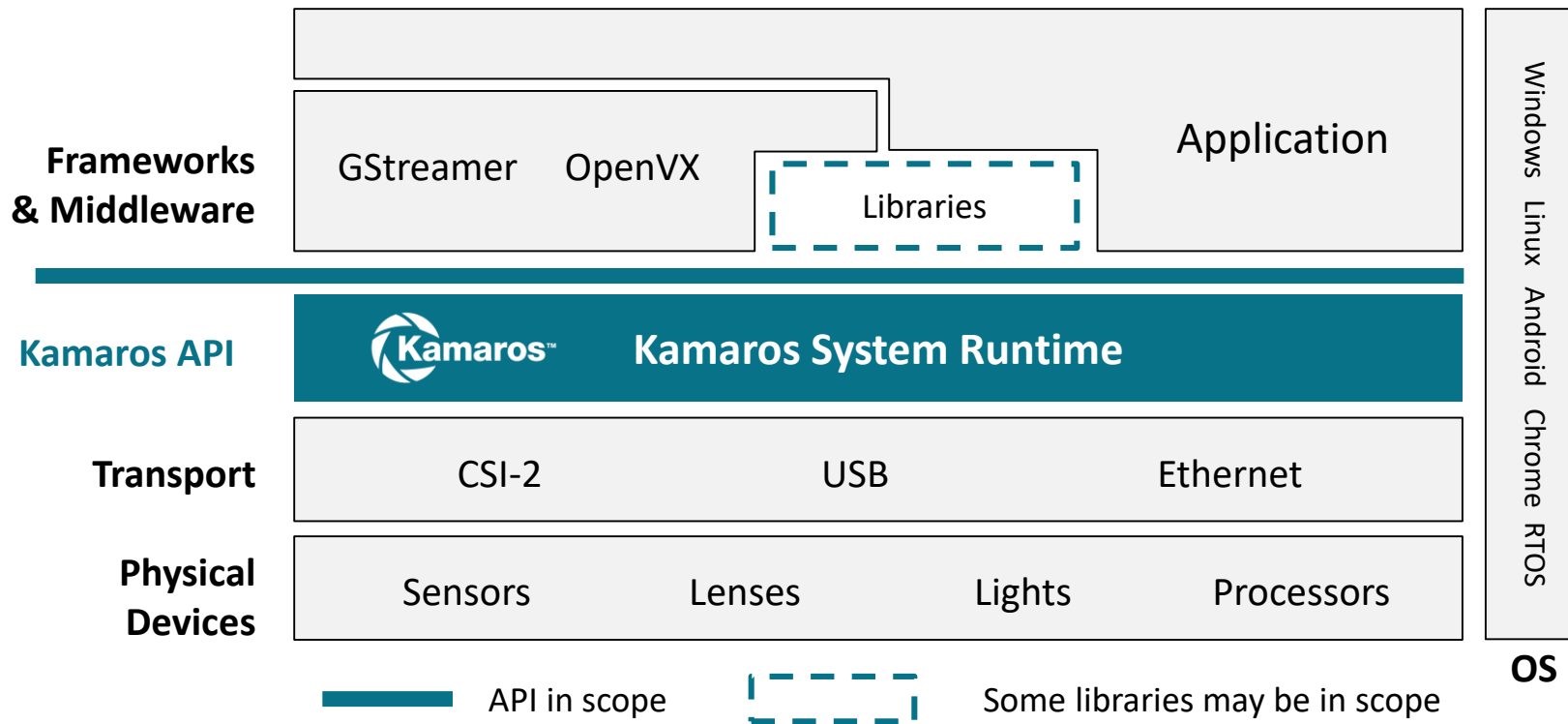
Portability of camera/sensor code for easier system integration of new sensors

Preservation of application code across multiple generations of cameras and sensors

Sophisticated control over sensor stream generation increases effectiveness of downstream accelerated processes

An effective camera API abstraction will enable sensor vendors to expose fuller hardware capabilities without exposing proprietary implementation details while while gaining access to a larger ecosystem of libraries and applications

Typical Kamaros Software Stack



Names of transport layers, frameworks and operating systems are illustrative examples

Structure of a Kamaros Application



1. Discover Camera

Select Camera via Kamaros Loader
Data provided on physical or geographic location, sensor and lens etc.

Query Pipeline Templates for selected Camera
Select Pipeline that has Controls and capability limits that satisfy use case

2. Discover Pipeline

Request Pipeline Configuration
Controls set in priority order

Best Attempt Pipeline Configuration
Achievable Control Values, Defaults and Constraints

3. Create Pipeline

Create Pipeline with initial Controls
Allocate any needed resources e.g., buffers

4. Capture Frames

Sets Controls per Frame
Synchronize for Frame availability

Process Returned Frame
Per frame metadata provides applied settings and errors

Stop Pipeline
Free resources

Application

Pipeline Templates

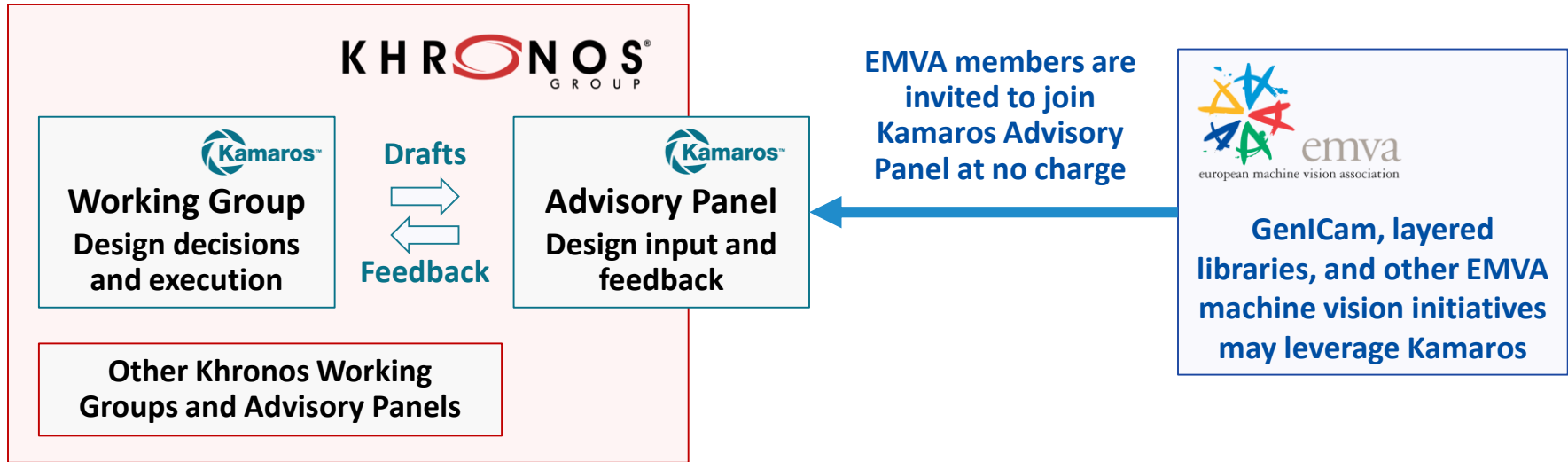
Define controls and capabilities of available streams through camera sensor, ISP and processors

Enable flexible and scalable discovery of camera capabilities

Simpler Templates provide can application portability

Customized Templates can provide detailed hardware control and may be vendor-supplied

Kamaros, Khronos and EMVA Cooperation



Khronos / EMVA have a Liaison Agreement for ongoing coordination and joint membership privileges for designated liaisons

Khronos is developing a growing family of open, royalty-free API standards relevant to embedded and safety-critical markets

Any company is welcome to join Khronos to influence standards development
<https://www.khronos.org/members/> or email memberservices@khronosgroup.org

More information on any Khronos API

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

Khronos members can participate in the Kamaros Camera Working Group

EMVA Members can join the Kamaros Advisory panel

<https://www.khronos.org/kamaros>

