Camera System API Working Group Proposed Scope of Work

December 2021

Approved by Exploratory Group vote on 8th December 2021

This document has been produced and agreed by the Embedded Camera Exploratory Group jointly hosted by the EMVA and The Khronos Group which met over the period of March to December 2021.

Created under the Khronos New Initiative Process, this document identifies the industry need, and outlines the scope, requirements and design methodology for a new open standard Camera System API. Deliverables and proposed timeline for the standard's design and deployment by a working group are discussed.

Linked Diagrams: Camera API SOW Graphics

Linked Glossary, Members and Requirements: Camera API Exploratory Group Materials

1. Background and Motivation

The use of cameras and imaging sensors is increasingly critical to diverse markets including embedded, mobile, industrial automation, XR, automotive, medical, life sciences, defense, aerospace and more. This has motivated significant developments in high resolution image sensors operating at high speeds with low noise in small packages, inexpensive high-performance processors and memory with low power consumption, as well as small and complex adaptive optic systems, all of which can be combined to create many types of camera systems for a tremendously broad spectrum of applications.

However, the lack of cross-vendor interoperability API standards for software control of camera systems places developers in a position of working with low-level interfaces to physical components, or through vendor or domain specific libraries and application interfaces. This increases development time and maintenance costs while reducing portability and opportunity for code reuse, resulting in unnecessarily high integration costs and long software development times.

The lack of camera API standards is becoming more acute as systems integrate multiple sensors of increasing sophistication and diversity, and require generation of sophisticated image streams to increase the effectiveness of downstream processing by a diverse range of accelerators, often using modern machine learning approaches, such as neural networks.

2. Exploratory Group Methodology and Conclusion

During 2021 the EMVA (European Machine Vision Association) and the Khronos Group, coordinated a series of meetings of an Embedded Camera Exploratory Group under the Khronos New Initiative Process, with the intent to understand and verify if there was sufficient industry need and interest in a new camera system open API standard. The Exploratory Group was publicly promoted, open to all interested parties without cost or discrimination, and represented companies and individuals from many different verticals and sectors. The group sought to build mutual understanding and consensus through the presentation of existing approaches, collation and examination of use cases, detailed capture of requirements, and establishing of a consistent glossary used to describe camera systems.

The 73 companies that have participated in the Exploratory Group are listed in Attachment A.

Particular existing standards, libraries, and frameworks which were considered and influenced the discussions of the Exploratory Group included the Android Camera2 framework API, GenICam (a generic programming interface for all kinds of devices, although predominantly cameras, hosted by the EMVA), libArgus (an API for acquiring images and associated metadata from cameras, developed at NVIDIA), and libcamera (an open source camera stack and framework for Linux, Android, and ChromeOS).

The work of the Exploratory Group concluded that a) there is sufficient need to develop a cross-market camera system API, b) existing standards and interfaces did not provide all the necessary features required and c) that there was sufficient interest from across industry to participate in the development of a new API.

3. Camera System API Technical Scope and Requirements

In this and following sections, capitalized technical terms are defined in the Glossary in Attachment B.

The Camera System API (API) broadly corresponds to the concept of a 'software defined camera' which inputs some physically sensed representation of a scene and outputs a flow of digital data and information relating to the scene suitable for consumption by further downstream applications and clients.

The API shall provide third party Applications, Libraries and Frameworks low-level, explicit control over Physical Devices through a Camera System Runtime, with the lowest level of abstraction that still provides Application portability with effective, performant control of a wide variety of Camera Systems.

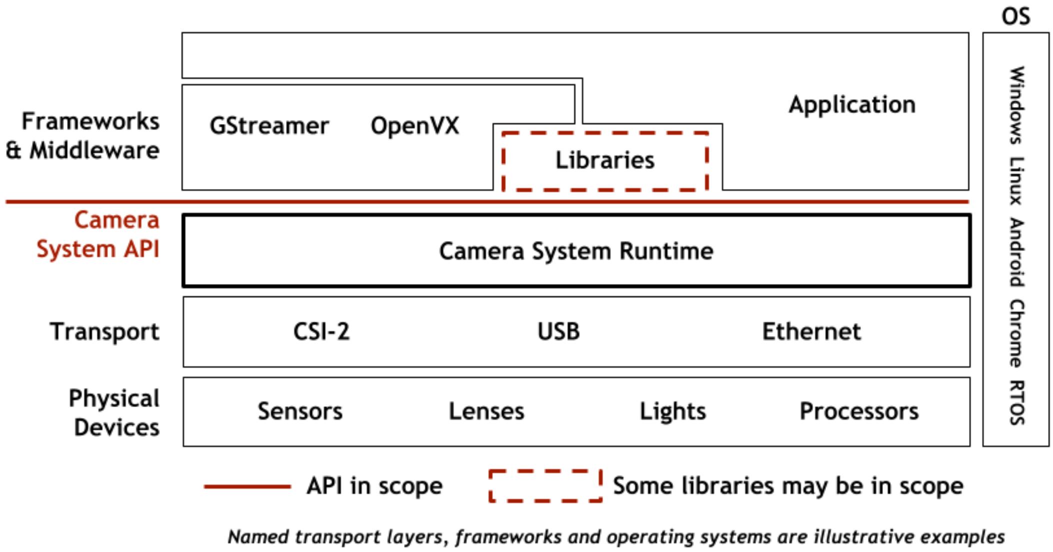

Figure 1. Camera System Runtime and API with optional Libraries

Layered libraries may provide non-core functionality such as synchronized access to the Camera System from multiple Applications, and scenario-driven Camera System control. The definition and open source implementation of some libraries may be in scope of this initiative.

The specification will be defined as an object-oriented API enabling bindings to C, C++ and other languages. The API shall be OS agnostic, including support for minimal embedded OS, and may include platform specific extensions.

The API will primarily focus on typical image data formats such as dense 2D arrays containing various image representations but will also include the capability to control additional sensor types via optional extensions, (depth, lidar/radar, and ultrasound for example), and extend to non-image data formats possibly provided by auxiliary sensing devices (for example inertial sensors, position sensors etc.).

Implementation details such as physical transport layer protocols will be hidden beneath the API. No specific reliance on any underlying technology or algorithm will be required in order to be compliant with the developed specification. For example, Camera Systems that include higher-latency transport mechanisms, such as Ethernet, shall not be precluded, but not all API functionality may be exposed in such systems.

The API shall enable discovery of Camera System capabilities that may include one or more Cameras configured from multiple Devices to process Sensor data and produce one or more Streams of Frames sent to External Runtimes or hardware. The API shall enable discovery and enumeration of all Devices and their Topology within the Camera System. Devices will be assigned Device IDs used for discovering Device Capabilities and setting Device Controls.

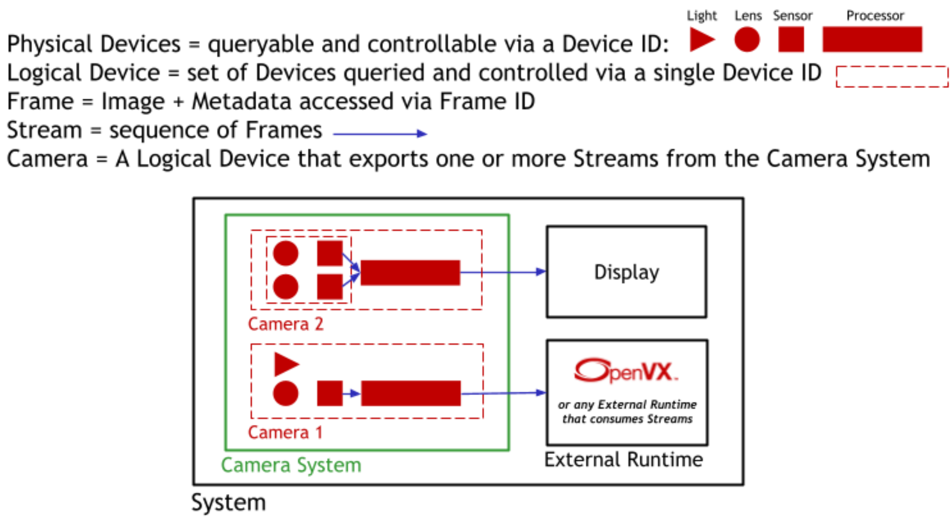

Figure 2. An example Camera System with Devices, Streams and Cameras

A key abstraction in the API is to enable one or more Devices to be gathered into a Logical Device, controllable with a higher level of abstraction via a single Device ID. This capability will enable simplified programming, enhanced software portability, and coordinated control of multiple Devices.

The API shall provide flexible control over Sensor Captures, including the ability to mix persistent and optional per Frame Device Control updates.

The API shall provide thread safety and document all threading assumptions, including thread synchronization for multi-threaded access to Devices. The Camera System Runtime will support a single client Application, but shall enable multiple client sharing through the use of a layered Library.

More detailed requirements and an analysis which requirements are satisfied by current APIs is here.

4. Design Methodology

Primary target markets shall include embedded, mobile, industrial automation, XR, automotive, and scientific. The API design shall remain general purpose, to not preclude other markets such as medical, life sciences, defense, aerospace and others. Aspects of the API to ease system-level functional safety certification shall be considered during API design.

It is recognized that the Camera System API scope and requirements outlined above may be too ambitious for an initial design. Additional insights gained through the initial design work may lead to prioritization of requirements and a reduced scope to ensure the timely delivery of a V1.0 of the specification that delivers critical industry value and drives initial industry adoption, followed by incremental expansion of capabilities.

The API will be designed with extensibility including working group designed extensions for optional or experimental capabilities, and vendor extensions to enable any implementor to extend the API however they wish to meet market and customer needs, including exposing proprietary functionality. Proprietary vendor extensions are not included in testing for conformance. Optional extensions will be conformance tested if present.

5. Organization and Deliverables

The EMVA and Khronos have agreed that the Camera System API specification and ecosystem will be developed, evolved and managed by a dedicated Khronos Working Group, open to any Khronos member and under standard Khronos processes, as a joint initiative of the EMVA and Khronos. On the formation of the Working Group an industry-wide call for participation shall be issued.

The API shall be made openly available to the industry under royalty-free licensing terms as defined by the Khronos Intellectual Property (IP) Framework.

Working group deliverables shall include:

- A Camera System API specification for use by implementers of the API, and developers using the API.
- A central extension namespace registry for Working Group and vendor extensions.
- An open source conformance test suite, including a precise definition of conformance.
- A maintained Adopters Program, so that test results may be submitted to enable implementations to become officially conformant.
- An API trademark and logo for promotion, and use on conformant implementations, that will be registered if possible.
- A conformant portable open-source sample implementation of the API.
- Open source samples and documentation.
- Open source SDK, tools and Libraries.

A call for design contributions will be made on formation of the Working Group as adapting a shipping API can significantly reduce development time by providing a proven design starting point. If one or more starting design contributions are received, a provisional specification would typically be available approximately 18 months after the Working Group is established.

The Working Group will not share detailed design contributions with any other Khronos Working Group and so can be placed in its own IP Zone.

It is expected that the Working Group will probably meet weekly, and will participate at occasional Khronos face to face meetings. The Working Group may delegate tasks to subgroups and elect Working Group Officers as necessary using standard Khronos processes.

6. Forecast Participation and Adoption

Industry interest in the Exploratory Group has been high, with over 70 companies joining the group. The weekly calls have a sustained attendance of around 30 participants on each call from over 20 companies.

It is estimated that any attrition in active members in the transition from Exploratory Group to Working Group will be more than compensated for by increased participation from passive Exploratory Group members, Khronos Members, and the wider industry once the Working Group call for participation is announced. Therefore sustaining the current size of 30 participants from over 20 companies is probably conservative.

Approximately 50% of the current active Exploratory Group members are not Khronos members, and so with a conservative total of 20 participating companies, Khronos could expect approximately 10 new members.

Both the EMVA and Khronos view the continued participation of representatives from many industry sectors as critical to the success of the Camera System API. Both organizations are committed to support this new industry development, and so will negotiate and execute a liaison agreement based on the Khronos liaison template, with joint membership privileges for designated liaison individuals to enable effective ongoing cooperation and coordination over the Camera API and related initiatives.

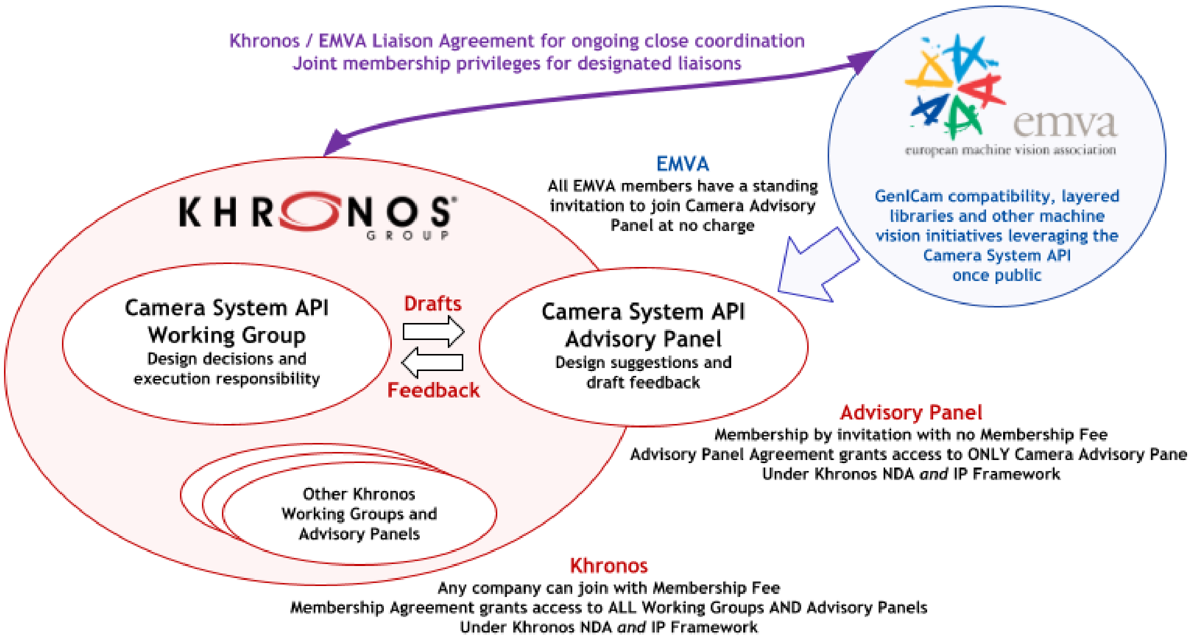

Figure 3. Proposed Camera System API coordination between EMVA and Khronos

To encourage wide industry participation, the Working Group will immediately establish an Advisory Panel under standard Khronos processes, with free-of-charge participation by invitation, and operating under the Khronos NDA and IP Framework. Any current EMVA member shall be offered a standing invitation to join at no charge for broad ongoing engagement with the machine vision industry. Any Advisory Panel member is welcome to join Khronos at any time for full access to ALL Working Groups AND Advisory Panels.

A typical Khronos Working Group would expect approximately 50% of participating companies to become Adopters, however, the Machine Vision industry has traditionally seen significantly wider adoption, for example the EMVA's GenICam standard has over 200 Adopters. It is recognized that eventual adoption patterns will depend on a number factors including API implementation complexity, layered partitioning into core API and Libraries, complementary value with existing standards, and level of Adoption fees. Consequently the Working Group will design an Adopter Program to best meet the needs of the industry when the API design and the character of interest in industry adoption becomes clearer.

Attachment A
Exploratory Group Participating Companies

The 73 companies that have participated in the Exploratory Group are listed below.

Adimec

AImotive

Alexey Kynazev

ALL3D, Inc.

Allied Vision Technologies GmbH

Almalence Inc.

AMD

Analog Devices Inc.

Apertus

AREA

Arm Limited

AVAL DATA CORPORATION

Axis Communications AB

Basemark Oy

Basler AG

Baumer Optronic GmbH

Cadence Design Systems, Inc.

Caster Communications, Inc.

China Daheng Group, Inc. Beijing Image Vision Technology Branch

Codeplay

Collabora

Continental Automotive GmbH

Digica Solutions

Digital Air Technologies

Digital Media Professionals

Euresys

European Machine Vision Association

FLIR Integrated Imaging Solutions

FOVE

Global Nomad GIS Services

Google

Groget

GuangDong OPPO Mobile Telecommunications Corp., Ltd.

Holochip Corporation

HP

Huawei Technologies Co. Ltd.

Ideas on Board Oy

Imagination Technologies

Institute for Computer Science and Control

Intel

Jon Leech

Khronos Group

Kivisense

LINX

Lucid Vision Labs, Inc.

LunarG, Inc.

m-lighted SRL

MATRIX VISION

MIPI Alliance

MM Solutions

Mobica Ltd

MVTec Software GmbH

Nokia OYJ

NVIDIA

Open AR Cloud Association

PCO AG

Perey Research & Consulting

Pinnacle Imaging Systems

Pleora Technologies

Qtechnology

QUALCOMM

Raspberry PI Ltd

Red Hat, Inc.

Rupert Stelz

Samsung

Sony Corporation

STEMMER IMAGING

Synopsys

Takumi Corporation

Teledyne

Texas Instruments

VeriSilicon

Vision Components

Attachment B
Glossary

This Glossary defines terminology and concepts that are used consistently if capitalized in this document.

Term	Definition
3A	Automatic per Frame control of camera/processing parameters for focus, white balance and exposure
API (Camera System)	Specified set of function calls, with associated parameters, to control the Camera System
Application	Application software using the Camera API, potentially in combination with Middleware and Frameworks
Camera	Device that exports one or more Streams from the Camera System
Camera System	Set of Devices in a system that processes data from one or more Sensors
Capture	Initiate a Sensor or Logical Device to generate a Frame through Device Controls at a nominal point in time
Device	A Physical Device or Logical Device
Device (Physical)	A component controllable via a Device ID: Sensor, Processor, lens, light etc.
Device (Logical)	One or more Devices controllable via a single Device ID
Device Capabilities	Definition of potential Device functionality accessed through a Query, including #IO Streams
Device Controls	An API setting to control operation of a Device using a Device ID
Device Control Set	A group of Device Controls used together to control multiple Devices
Device ID	Unique identifier that can be used to Query and Control a Device
Device Status	Real time status of a Device accessed through a Query
Extension	API functionality that is optional to implement and may be vendor specific
Frame	Image and associated Metadata. Either may be empty (but not both)
Frame Format	Definition of how Image and Metadata can be accessed via a Frame ID
Frame ID	Unique identifier that can be used to access the contents of a Frame
Framework	External Runtime used by the Application that may also use the API (e.g. OpenVX, GStreamer)
GeoPose	OGC's encoding of geospatial position and orientation with six degrees of freedom
Hint	Information that may optionally be used to optimize Camera operation
Host	CPU controlling the Camera System through the Camera API
Image	Collection of Pixels contained within a Frame. May be non-spatially ordered
Image Format	Definition of how Pixels are held within an Image
Metadata	Non-Image data within an Frame
Middleware	Libraries or other software linked into the Application, that may also use the Camera API

Opaque Format	A non-standardized format, often used to optimize processing within an implementation of the API
Operations	A queryable set of Frame processing operations
Pixel	Individual datum contained within an Image
Pixel Format	Definition of how data is organized within a Pixel
Processor	Device that enables Operations on one or more Streams
Query	API call to discover Camera Topology, Device Capabilities or Device Status
Raw Frame	Unprocessed Frame generated directly by a Sensor
Raw Image	Unprocessed Image in a Frame generated directly by a Sensor
Reprocessing	Recirculation of a Frame or Raw Frame for additional processing
ROI	A selected region (of interest) of an Image
Runtime (Camera System)	An implementation of the Camera API
Runtime (External)	An implementation of an API other than the Camera API
Sensor	Device that Captures a Frame from physical phenomenon
Slice	A contiguous set of rows in a Image
Stream	A sequential series of one or more Frames
Topology	Queryable connections between Devices (inc. Hints) and Stream import/export in the Camera System
User	Programmer using the API