GENeric programming Interface for CAMeras

Dr. Friedrich Dierks, Basler AG

Secretary of the GenICam Standard Group

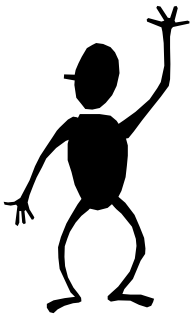
Basler Head of Software Development Components

Version 2.8

Questions Answered in this Presentation

emva
european machine vision association

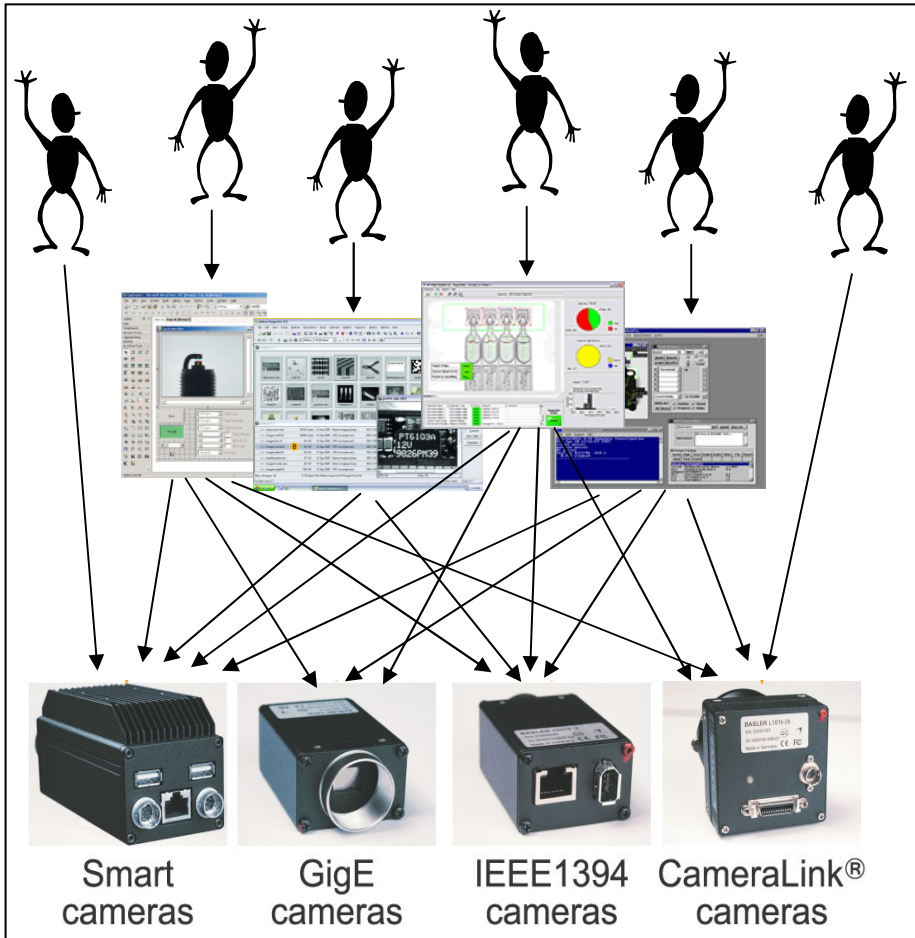
- Why GenICam Standard?
- How does it work?
- How is the standard committee organized?
- Who is driving GenICam?
- What is the status and the roadmap?
- How can you become part of GenICam?
- What are your benefits?



GEN<i>CAM

BASLER
VISION TECHNOLOGIES

Situation Yesterday



Customers want to use…
- …any image processing library
- …any camera
- …any smart feature in the camera

Camera Vendors want to..
- …sell to every customer
- …supply their smart features

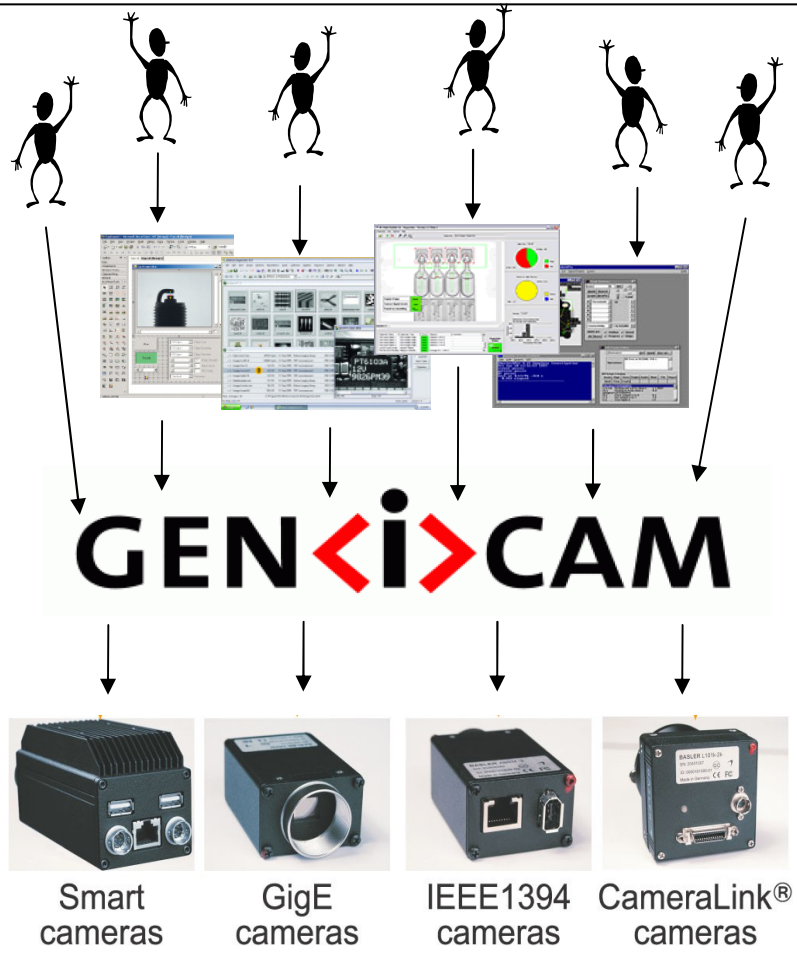
Library Vendors have to…
- …support all cameras
- …support all smart features

➔ **This is expensive**

➔ **This reduces time-to-market**

➔ **This prevents market growth**

Situation Today



GenICam can connect the Customer..

- …to all cameras
- …through all libraries
- …giving access to all smart features

GenICam can support…

- …any interface technology
- …products from any vendor
- …products with different register layout

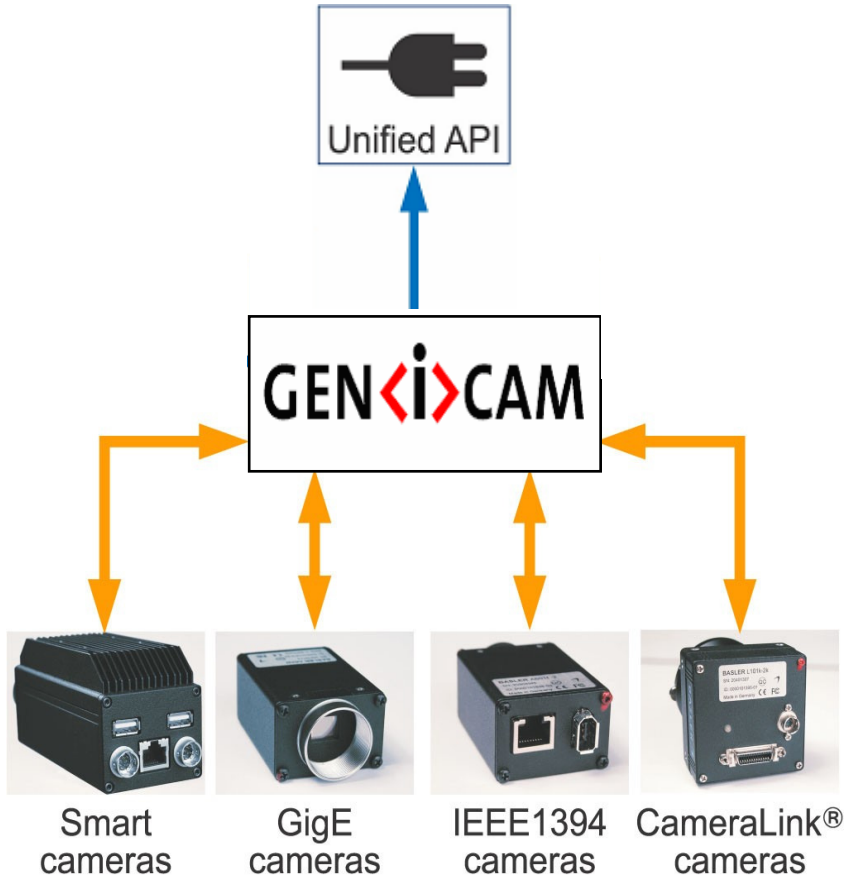
GenICam is easy to integrate for…

- …customers
- …camera vendors
- …software library vendors
- …frame grabber / driver vendors

GenICam in an NutShell

emva
european machine vision association

➔ GenICam provides a Unified Programming Interface for machine vision cameras



GEN<i>CAM

BASLER
VISION TECHNOLOGIES

GenICam Use Cases

- Configuring the Camera
- Grabbing Images
- Providing a Graphical User Interface
- Delivering Events
- Transmitting Extra Image Data

Customer Viewpoint

Configuring the Camera

Use Case

emva
european machine vision association

User API

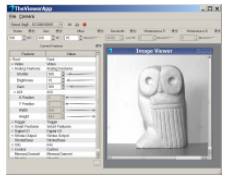
- C++ programming interface

```
if( IsWritable(Camera.Gain) )
      Gamera.Gain = 42;
```

- Provided by freely available GenICam reference implementation
- Other programming languages can be supported, e.g., .NET

Transport Layer API

- Read / Write Register
- Provided by driver vendors (small adapter required)
- Send / Receive ASCII Command extension under planning



User API

Set Gain = 42

GEN<i>CAM

TL API

Write Register
Address, Len, Data

Transport Layer

Interface

1394, GigE Vision,
Camera Link, etc.

GEN<i>CAM

BASLER
VISION TECHNOLOGIES

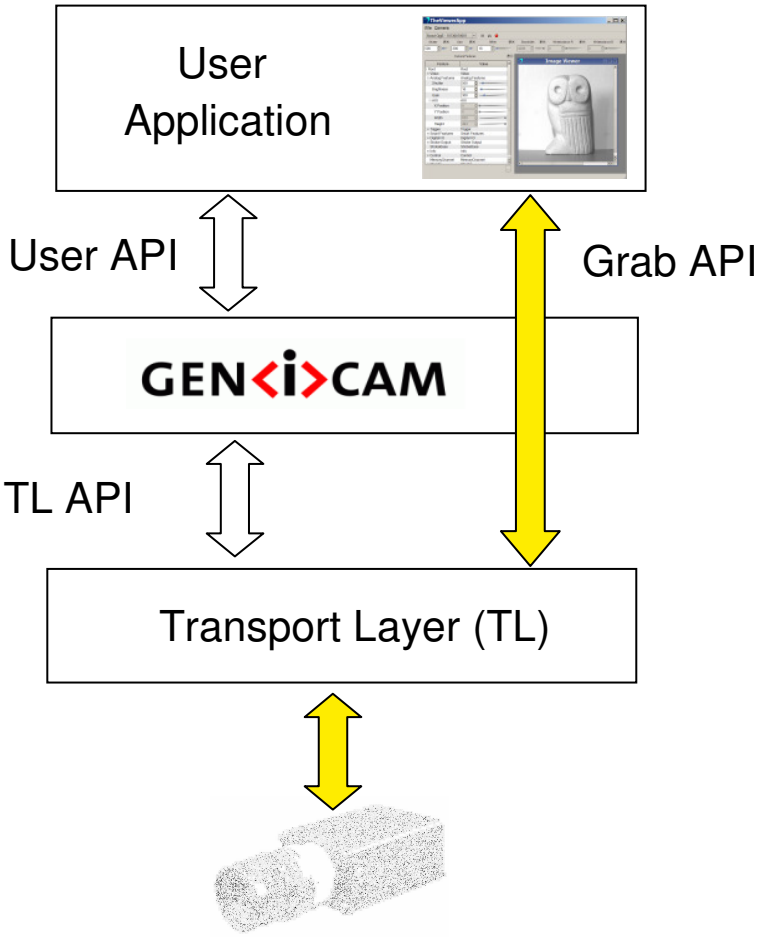
Grabbing Images

Use Case

emva
european machine vision association

Grab API

- Abstract C++ programming interface
  - `Get device names`
  - `Create camera access object`
  - `Configure camera`
  - `Queue buffers`
  - `Start acquisition`
  - `Wait for buffers`
- Implemented by transport layer DLLs
- Provided by driver vendors (adapter required)
- GenICam provides services to
  - register transport layer DLLs
  - enumerate devices and
  - instantiate camera access objects



GEN<i>CAM

BASLER
VISION TECHNOLOGIES

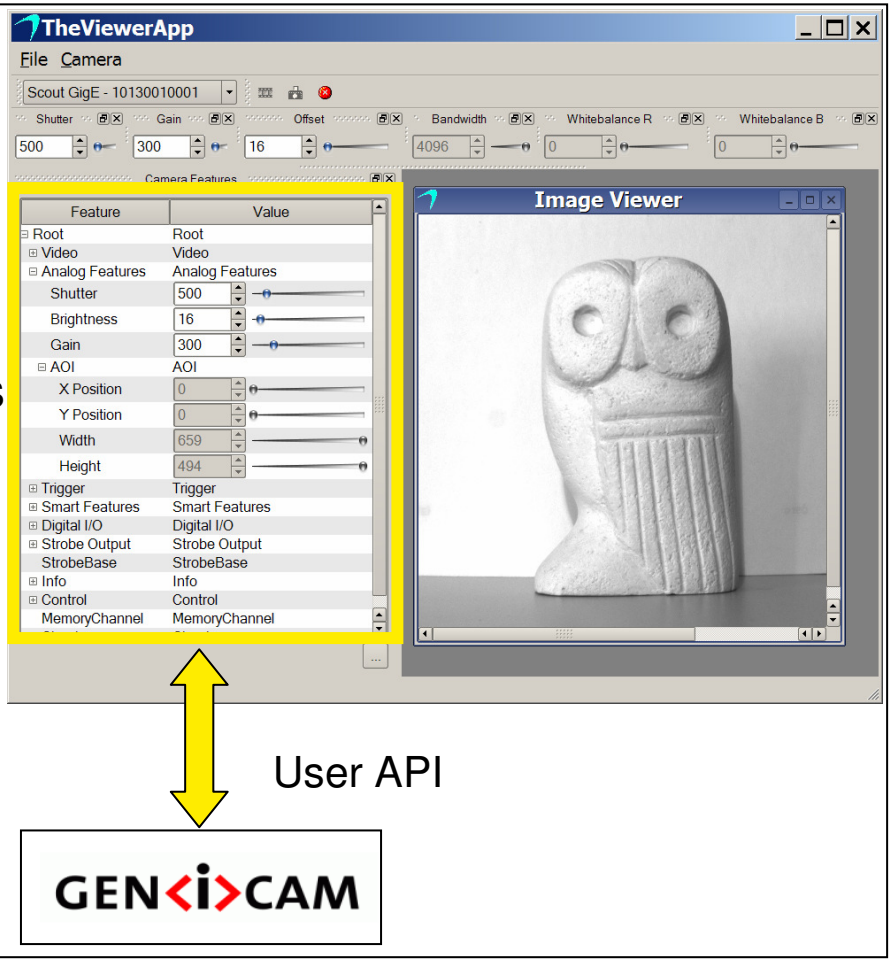
Providing a Graphical User Interface

Use Case

emva
european machine vision association

GUI support

- Feature tree
- Widgets support
  - **Slider** → value, min, max
  - **Drop-Down Box** → list of values
  - **Edit Control** → From/ToString
  - etc.
- Access mode information
  → RW, RO, WO, …
- Full model / view support
  → callback if a feature changes



GEN<i>CAM

BASLER
VISION TECHNOLOGIES

Delivering Events

Use Case

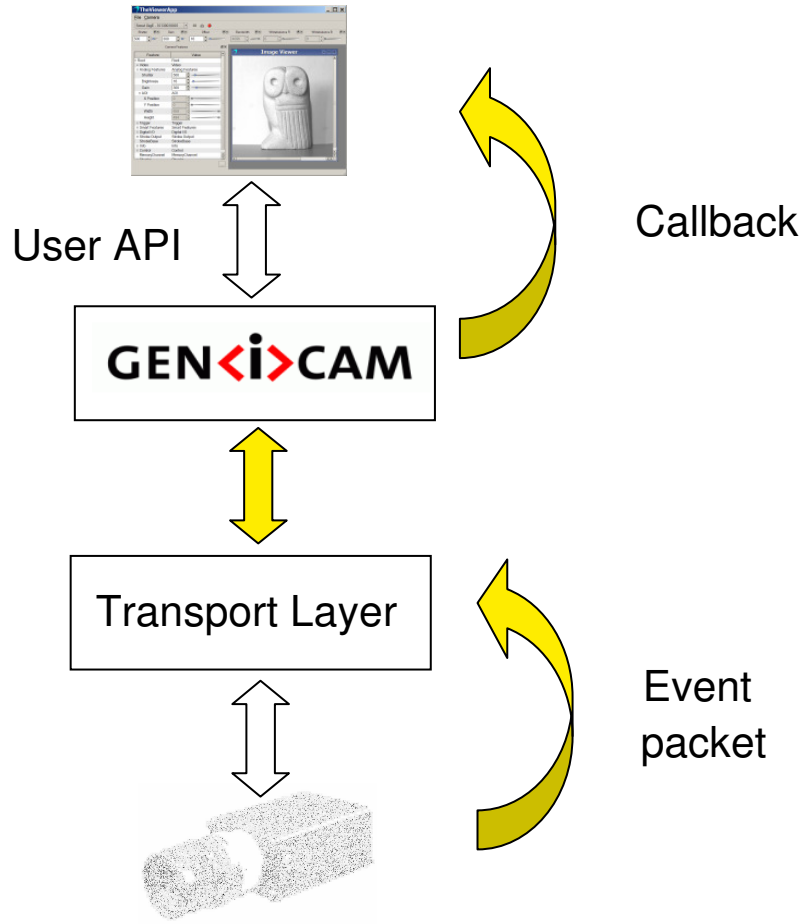
Asynchronous Callbacks

- Cameras can deliver event packets, e.g. when the exposure has finished
- Users can register a callback

```
void Callback( INode* pNode )
{ printf("Hi!"); }

Register( Camera.ExposureEnd,
      &Callback );
```

- Events are identified by an EventID
- If an event packet arrives GenICam fires a callback on all nodes with matching EventID
- Data coming with events is also delivered.



Transmitting Extra Image Data

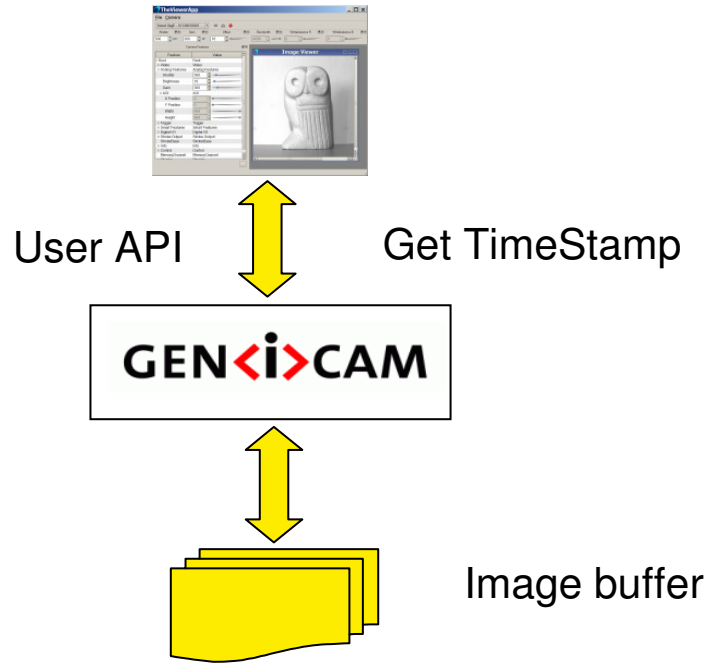
Use Case

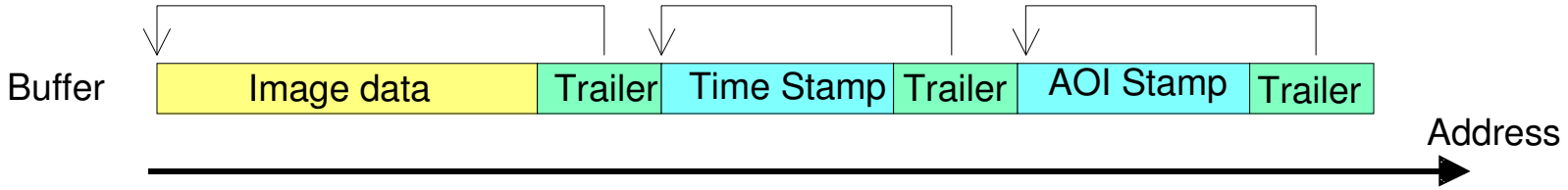
Chunked Data Stream

- Images can have chunks of additional data appended, e.g. a time stamp.
- GenICam makes this data accessible

```
if( IsReadable( Camera.TimeStamp ) )
      cout << Camera.TimeStamp();
```

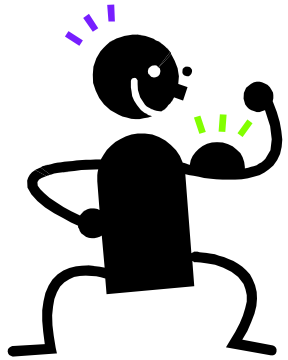
- The transport layer “shows” each buffer to GenICam.
- GenICam interprets the chunks as read only registers identified by a ChunkID





Making GenICam Compatible Products

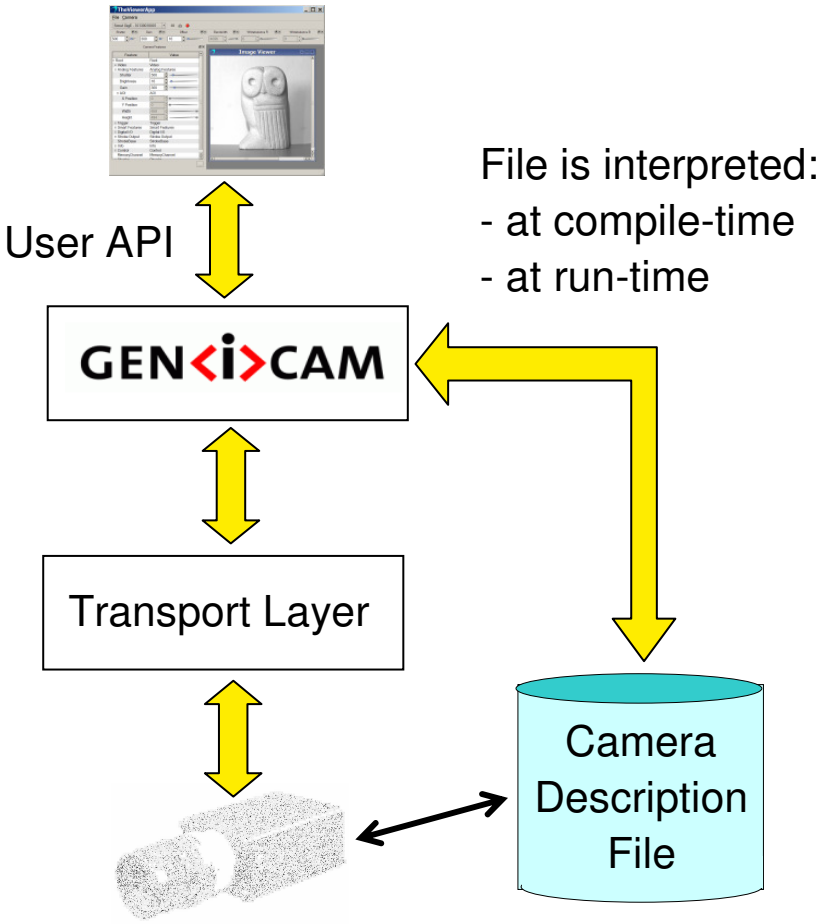
- Features
- Making Cameras Interchangeable
- Reference Implementation
- License Issues



Vendor Viewpoint

Camera Description File

- Describes how features (“Gain”) map to registers (or commands)
- XML format with a syntax defined in the GenICam standard
- Static use case : a code generator creates a camera specific C++ class at compile-time
- Dynamic use case : the program interprets the XML file at run-time
- Camera description files are provided by the camera vendor



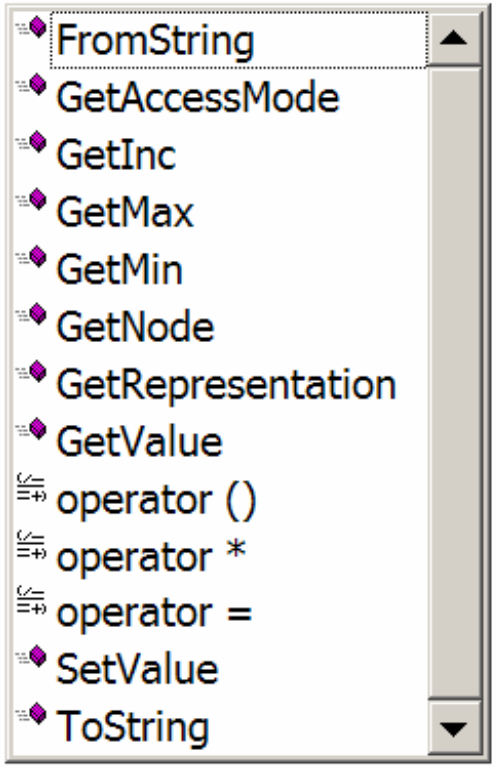
Feature Types

- Each feature has a type that is defined by an abstract interface
- Common types with associated controls are:
  - Integer, Float ⇔ slider
  - String ⇔ edit control
  - Enumeration ⇔ drop down box
  - Boolean ⇔ check box
- With GenICam camera vendors can use whatever feature names, types and behavior they like.
- As a consequence GenICam alone does not make cameras interchangeable!

  ➔ **Standard Feature List** is required

Example: Integer interface

```
Camera.Gain.
```



Standard Feature List

For **GigE Vision** cameras a list of ~**180** standard features is provided.

- This list is organized along use cases:
  - Image size control
  - Acquisition and trigger controls
  - Digital IO
  - Analog Controls
  - …
- Only 7 features are mandatory, the others are just recommended

- The GigE Vision standard says

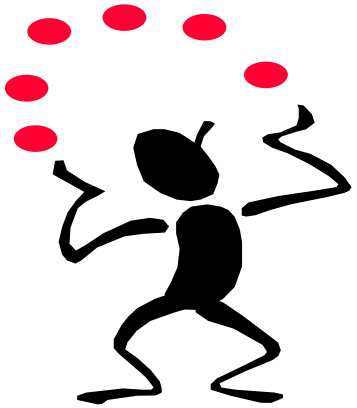
  *…any GigE Vision device **MUST** provide an XML device description file compliant to the syntax of the GenApi module of GenICam™.*

For **1394 IIDC** cameras the same list of features can be used with only a few adaptations.

- A common XML file is still under construction

GenICam Organization

- Standard Committee
- Supporting Companies
- Status & Roadmap
- Benefits



Industry Viewpoint

GenICam Standard Committee

emva
european machine vision association

- GenICam is hosted by the European Machine Vision Association (EMVA)
- **Contributing members** are working(!) on the standard and the reference implementation. Only contributing members can **vote**.
- **Associated members** agree to the GenICam rules. They get full access to the source code and are placed on the mailing list but **cannot vote**.
- **Interested outsiders** get the GenICam run-time and the released standard documents
- You can **register** at www.genicam.org

Contributing Members
- currently 8 companies -

Associated Members
- currently 20 companies -

Interested Outsiders

no fees!

*) as of b/o May 2006

GEN<i>CAM

BASLER VISION TECHNOLOGIES

GenICam Members

emva
european machine vision association

GEN<i>CAM

BASLER VISION TECHNOLOGIES

Status*) and Roadmap

emva
european machine vision association

GenApi Module

- Standard and reference implementation v1.0 are released and are available on www.genicam.org.
- The number of GenICam aware products is constantly growing. Among them are:
  - All GigE Vision compliant cameras
  - Many of the image procession software libraries
  - Some 1394 cameras

GenTL Module

- Defined interfaces and working adapters for GigE Vision, 1394, and Camera Link
- Draft standard expected Q1 2007

Standard Feature List

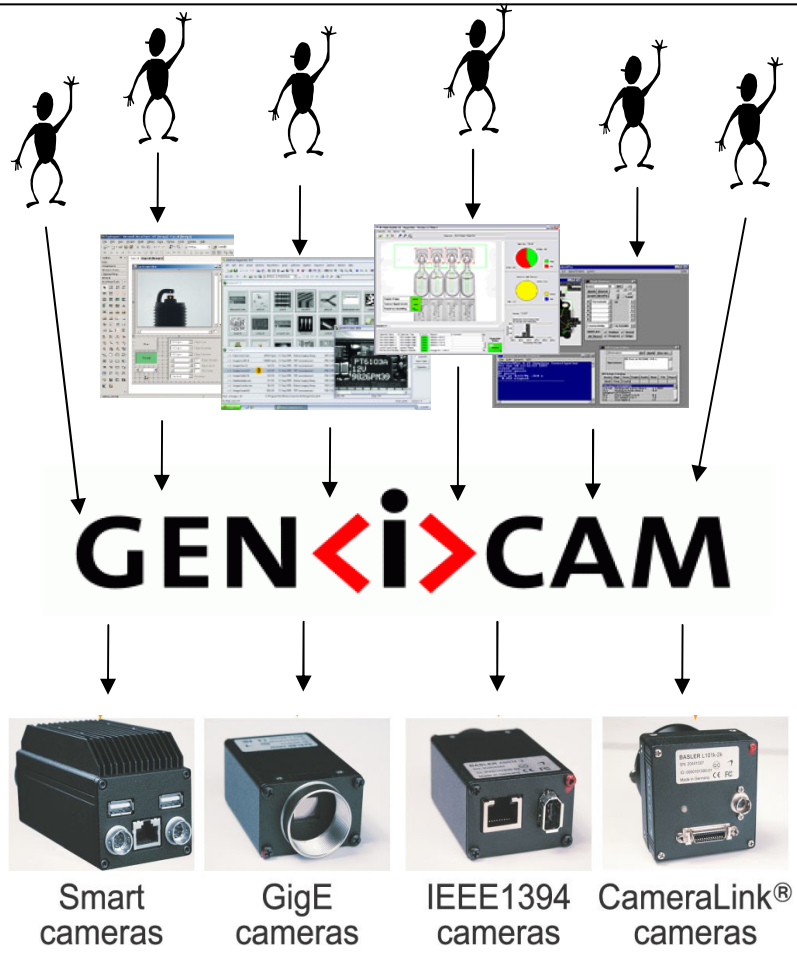
- GigE Vision : v1.0 is released
- 1394 IIDC : under construction

*) cw36 / 2006

GEN<i>CAM

BASLER VISION TECHNOLOGIES

Benefits



Customers

- Combine
  → any camera with
  → any smart feature with
  → any software library
- Mix interface technologies and cameras from different vendors

Vendors

- Enlarge your market
- Reduce your cost
- Speed up time-to-market

Thank you for your attention!

Contact me ➔ friedrich.dierks@baslerweb.com

Get information ➔ www.genicam.org