



A new communication interface for the European Southern Observatory (ESO)'s Very Large Telescope Technical Detector Control System using Aravis, an open-source library for GenICam cameras



K. F. Mulholland¹, Observatory Sciences Ltd., Kirkcaldy, U.K.

J. Knudstrup, F. Pellegrin, European Southern Observatory, Garching, Germany

¹ kfm@observatorysciences.co.uk

Introduction

The European Southern Observatory's Very Large Telescope (VLT) provides support for high-performance industrial cameras with the Technical Detector Control System (TDCS).

The VLT's Core control System (CCS) is a software framework providing general operational tools, e.g. messaging system, Real-Time Display (RTD), and interfaces for instrument control.

TDCS uses a "communication interface", based on the *Adapter Pattern*, to handle communication with the camera. Until now, the communication interface has used an API from Allied Vision Technologies (AVT), which only supports AVT cameras.

As part of the VLT 2019 release, a new communication interface has been developed for TDCS using Aravis, the open-source library for GenICam cameras.

GenICam and GigE Vision

Generic Interface for Cameras (GenICam) standard specifies that camera properties are stored in a self-describing XML file.

GigE Vision cameras use a gigabit-ethernet connection to directly connect to a camera and comply with the GenICam standard.

Usually, vendors provide an API and APIs from different vendors are not compatible. The vendor APIs are proprietary.

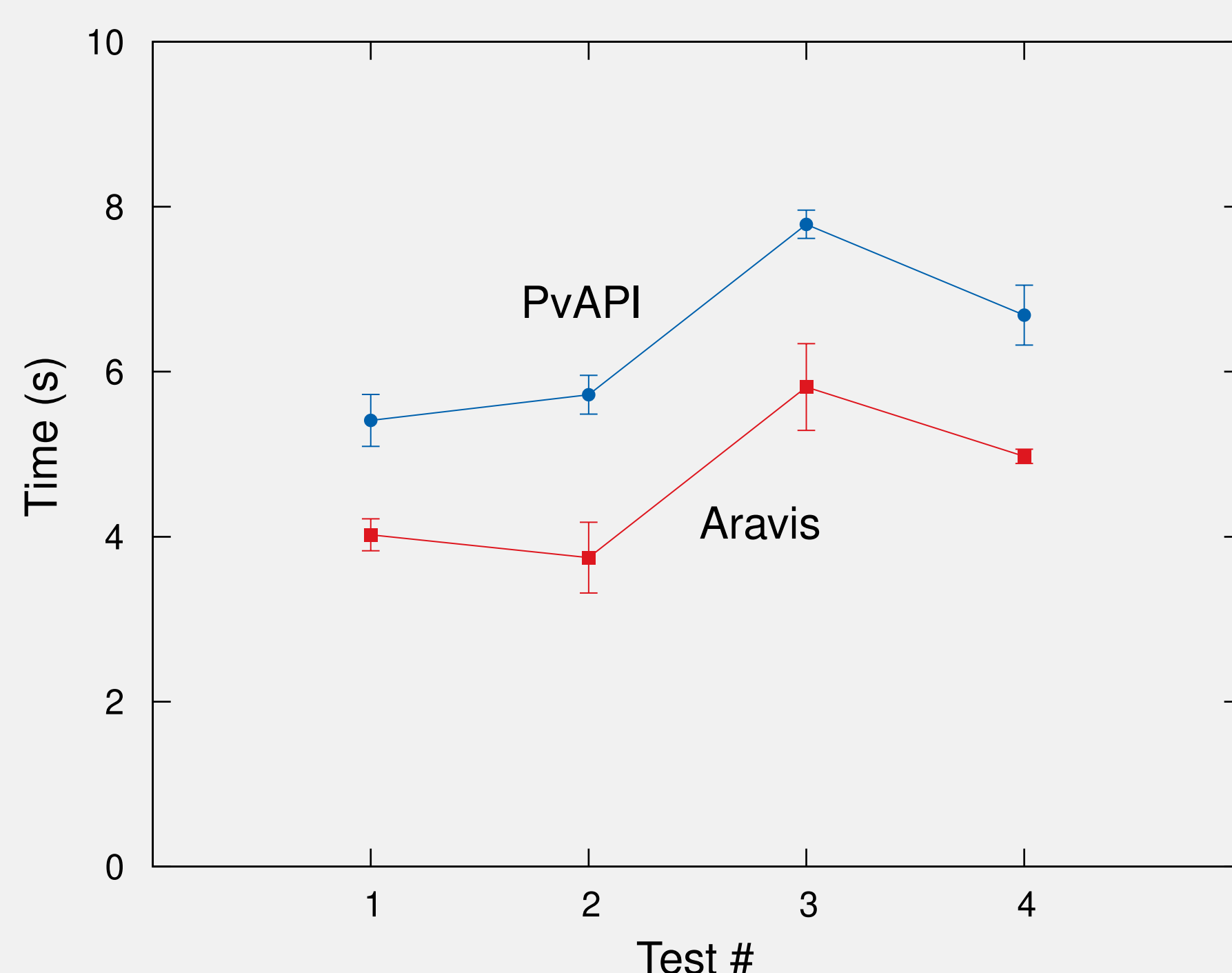
Aravis

- Library for communicating with GenICam cameras
- GLib/GObject based
- Open source and actively maintained
- *Should* be compatible with all GenICam-compliant GigE/USB3 Vision cameras
- Easy to install from tarball
- Includes basic tools, an ethernet camera simulator, and video viewer
- On github: <https://github.com/AravisProject/aravis>
- Forum: <https://aravis-project.discourse.group>
- Documentation is a little patchy!

Benchmarking

Four tests were used to compare the performance of the new and old communication interfaces. The tests were:

- | | |
|--|--|
| <p>Test 1</p> <ul style="list-style-type: none"> • State transition: Off→Standby→Online • Start acquisition • Stop acquisition • State transition: Online→Standby→Off | <p>Test 3</p> <ul style="list-style-type: none"> • State transition: Off→Standby→Online • Start acquisition • Acquire one frame • Stop acquisition • State transition: Online→Standby→Off |
| <p>Test 2</p> <ul style="list-style-type: none"> • State transition: Off→Standby→Online • Start acquisition • Acquire six frames • Stop acquisition • State transition: Online→Standby→Off | <p>Test 4</p> <ul style="list-style-type: none"> • State transition: Off→Standby→Online • Start acquisition • Acquire one frame • Set exposure time: 0.2 s • Acquire one frame • Stop acquisition • State transition: Online→Standby→Off |



All test cases show that the PvAPI communication adaptor has a small overhead which is not seen with the Aravis communication adaptor.

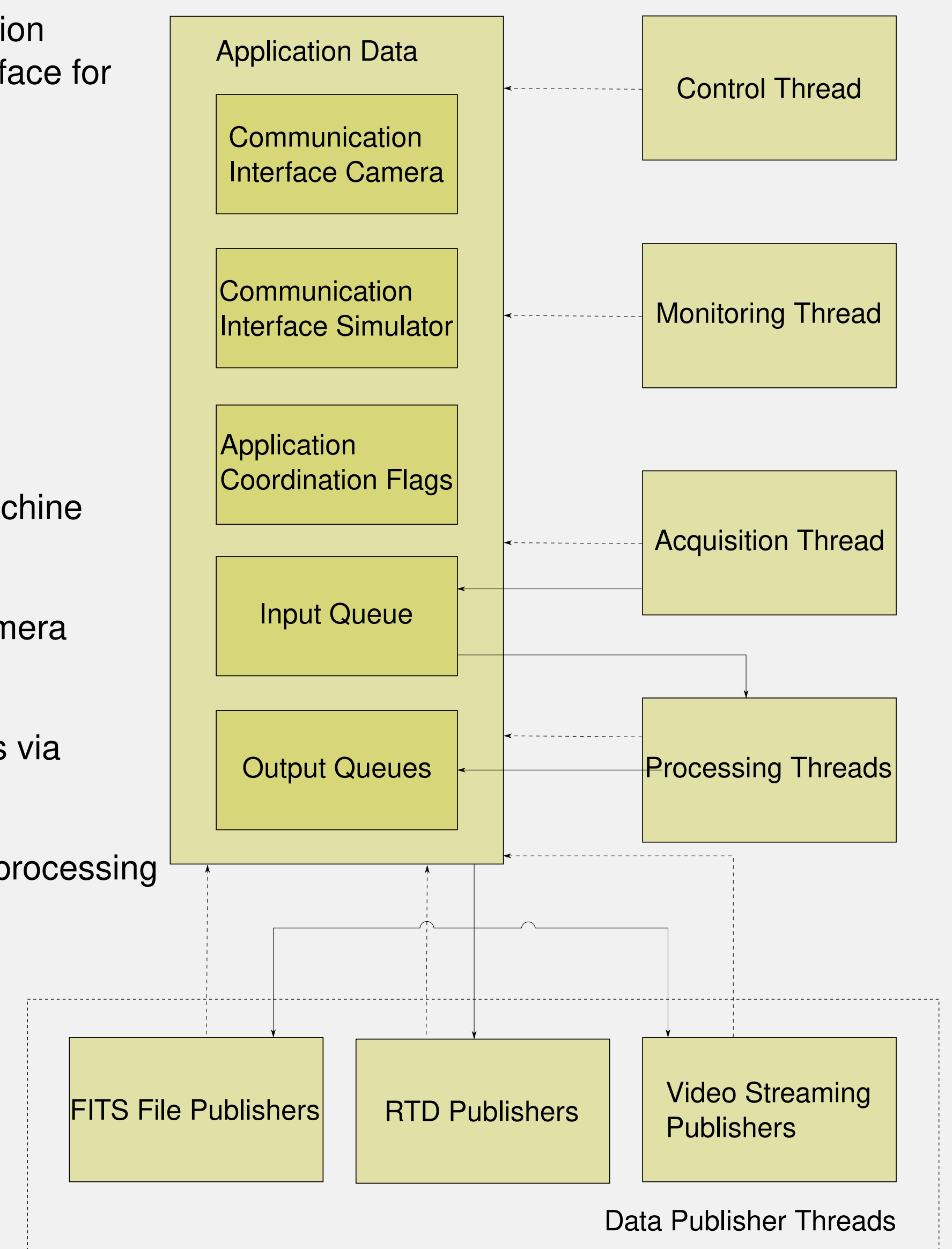
Technical Detector Control System

TDCS is delivered as part of the VLT instrumentation common software package, and provides an interface for high-performance industrial cameras.

- Written in C++
- Multithreaded: data acquisition, processing, and publishing are done in parallel
- Controls camera properties
- Publishes data to FITS files etc
- Implemented around an SCXML-based state-machine engine
- Implemented as a toolkit to integrate specific camera control application solutions
- Decoupled from vendor-specific implementations via communication adaptors
- Provides support for adding user-provided data-processing and publishing adaptors

TDCS is currently in use by ESPRESSO, which is capable of simultaneously using the light from all four of the 8.1-m Unit Telescopes of the VLT, as well as five new instruments under construction.

ESPRESSO uses up to nine TDCS instances to control AVT BigEye technical cameras. These are used for the *field* and *pupil* stabilisations of the light.

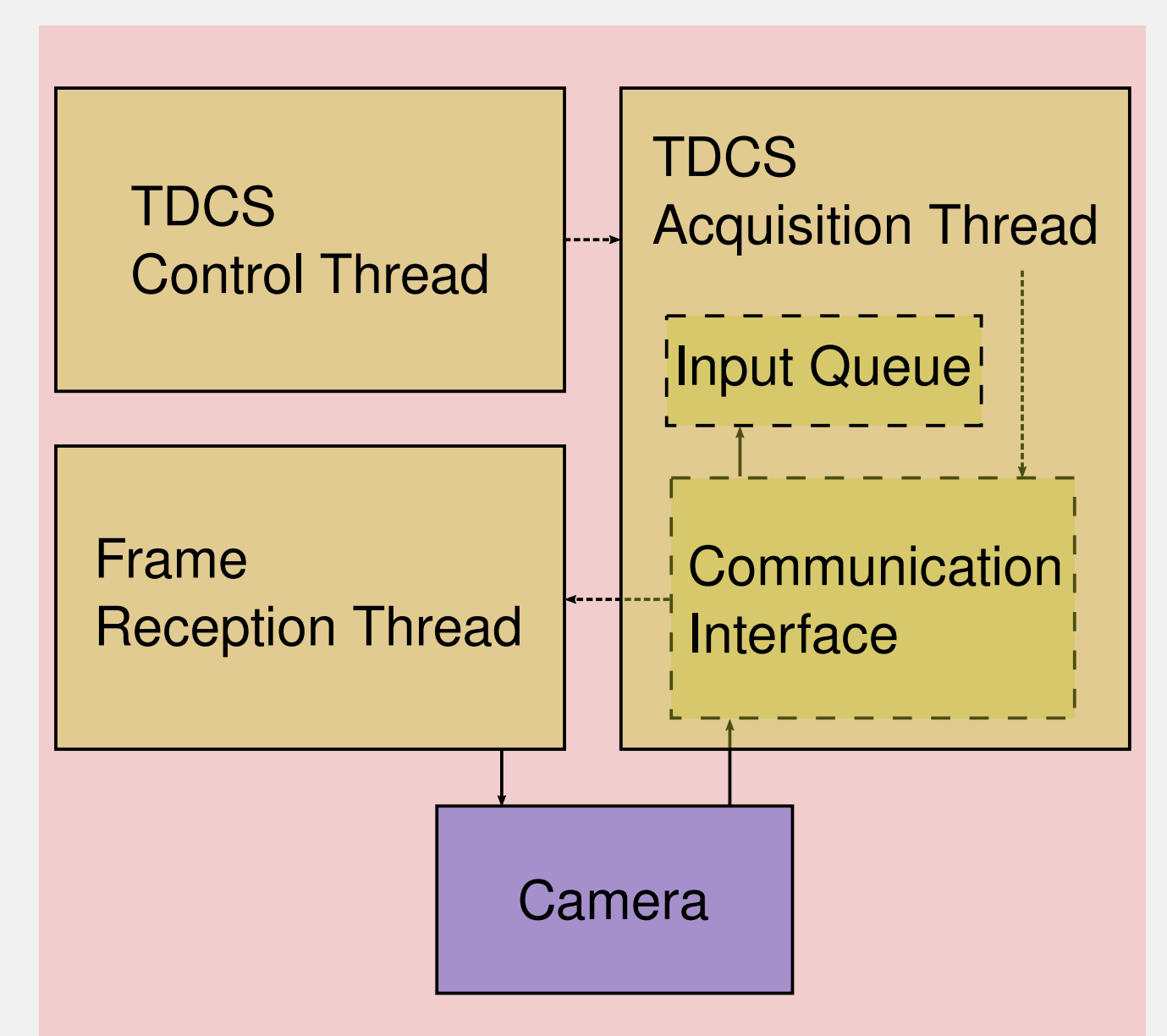


Implementation

- Implemented as a C++ class
- Inherits from the TDCS Base Communication Interface class
- Responsible for implementing defined set of inherited virtual functions.
- Loosely coupled to the rest of TDCS
- Currently supports cameras from AVT and Basler
- Different vendors may provide different names for properties

Data is stored in an object with all relevant information, which is easy to retrieve using Aravis:

```
// We've got a successful image, so fill out its info
frame.ImageBuffer = arv_buffer_get_data(buffer, NULL);
frame.FrameCount = comIntPtr->p_frameCount;
frame.TimestampSecs = static_cast<vltUINT32>(acqTimeInt);
frame.TimestampFrac = static_cast<vltUINT32>(acqTimeFrac);
frame.Width = arv_buffer_get_image_width(buffer);
frame.Height = arv_buffer_get_image_height(buffer);
frame.OffsetX = arv_buffer_get_image_x(buffer);
frame.OffsetY = arv_buffer_get_image_y(buffer);
frame.ImageSize = arv_device_get_integer_feature_value(
    comIntPtr->p_device, comIntPtr->PayloadSizeKey.c_str());
frame.PixelFormat = arv_buffer_get_image_pixel_format(buffer);
frame.BitDepth = ARV_PIXEL_FORMAT_BIT_PER_PIXEL(frame.PixelFormat);
frame.AncillaryBuffer = 0;
frame.AncillaryBufferSize = 0;
frame.FrameID = arv_buffer_get_frame_id(buffer);
```



Vendor	Parameter name map	
	Generic name	Vendor-specific name
AVT	GenName1	AVTName1
	GenName2	AVTName2
Basler	GenName1	BaslerName1
	GenName2	BaslerName2

- Create a C++ parameter name map for each vendor
- Put each parameter name map into a "map of maps" to get the right parameter names
- Easy to add support for more vendors!

Conclusion

A new communication interface for TDCS has successfully been implemented. Aravis has proved a good choice that will not only allow cameras from other vendors to be used, but will improve performance.

Testing of the TDCS Aravis communication adaptor will take place later this month at the VLT, whereby ESPRESSO will be migrated from using the PvAPI interface to the Aravis-based one, for communicating with its AVT BigEye technical cameras.