### ­High-level

###### Components (from bottom up)

* Hardware
* Thor DLLs (ThorConfocal and ThorPMT)
* NI-DAQmx DLL
* ThorLSM module (C++)
* MATLAB Dabs.Mex (CFAE)
* MATLAB Dabs
* MATLAB ScanImage

Quirk in the stack is that for NI, MATLAB Dabs talks directly to NI-DAQmx, and for a subset of Thor, MATLAB Dabs talks directly to Thor DLLs (bypassing ThorLSM).

ThorLSM is listed here as a module but ‘ThorLSM’ is also the name of a class available in that module.

Note that CFAE is not the same as ThorLSM. CFAE is a mexFunction gateway to ThorLSM and is very thin. CFAE is capable of holding/accessing multiple ThorLSM instances, although this capability is not used at the moment.

MATLAB has an execution thread (and other threads we do not worry about). The Thor DLL (ThorConfocal) has its own thread(s). ThorLSM has multiple threads.

### ThorLSM

###### What ThorLSM does

ThorLSM exists because a) MATLAB is single-threaded, and b) possibly for performance reasons. It is a C++ module accessed by Dabs through MEX. ThorLSM has the following responsibilities:

* Buffer acquired data to prevent dropping frames
* Process frames, eg averaging, warping, wrapping
* Make processed frames available to services/customers:
  + MATLAB, for display
  + Logging, for streaming to disk (Logging is included in ThorLSM)
  + USER MATLAB, for custom analysis
* Coordinate acquisitions as appropriate/necessary to achieve the above goals

###### How ThorLSM does it

ThorLSM can be thought of as a (pretty short) frame pipeline. The pipeline starts at the interface to ThorConfocal. ThorLSM is notified via a Windows event when ThorConfocal has a new frame. ThorLSM pushes this frame into the front of the pipeline. The pipeline processes the frame through one or more stages. Each stage potentially has its own thread for processing. When one stage is done with a frame, it pushes it onto the queue(s) for the next stage(s). The services/customers noted above (MATLAB, Logging, USER Matlab) are the final stages of the pipeline.

ThorLSM is vanilla C++ with STL and uses the Windows API for threads and synchronization. It also links against ThorConfocal and MATLAB libs (mex, mat, etc).

### Focus: ThorDLL, ThorLSM, dabs.thorlabs.LSM, CFAE

###### Configuring an acquisition

Before an acquisition can be run, setup and configuration must be performed. A large portion of this configuration occurs without involving ThorLSM. Instead, dabs.thorlabs.LSM (from now on: dabs.LSM) talks directly to ThorConfocal through callib, eg to set/get various acquisition parameters using ThorConfocal::SetParam and GetParam.

Some initialization/configuration is done through ThorLSM as well. Naturally ThorLSM has its own initialization needs (eg allocating its various queues) that must be handled at the start of an acquisition. Since ThorLSM handles logging as one of the end-of-pipeline-services, logging configuration is also done with ThorLSM. More abstractly, any stage of the ThorLSM pipeline may require pre-acquisition configuration.

###### Running an acquisition

Running an acquisition occurs (I believe) mostly through ThorLSM. That is, dabs.LSM talks mostly to ThorLSM (through CFAE) when starting/stopping/pausing an acquisition. This is not 100% true at the moment, as CFAE does permit some direct access to ThorConfocal, eg calls to StartAcquisition or PostFlightAcquisition. dabs.LSM does currently take advantage of this direct access.

***Veej: this is a bad idea and we should remove direct access to ThorConfocal for controlling an acquisition.***

The main reason this is a bad idea is that the integrity of an acquisition cannot be guaranteed if acquisition-running-related calls to ThorConfocal are made without ThorLSM’s knowledge. Starting an acquisition directly without calling an appropriate method on ThorLSM would be nonsensical for example. Even it were apparently functional, it could leave things in an unspecified and probably dangerous state.

###### Modifying an acquisition on the fly

It is very important to be clear about what configuration parameters can change during a running acquisition. At the moment, the only such parameters relate to file logging:

* Name of logging file
* Heading string for images (not implemented)

*Note: Logging-parameter changes should ultimately be implemented as an instance of FrameActor-info-arrival during acq.*

### ThorLSM Details

###### Components

* ThorFrameCopier
* Queues
* Logger
* Processor
* MATLAB-notifier