

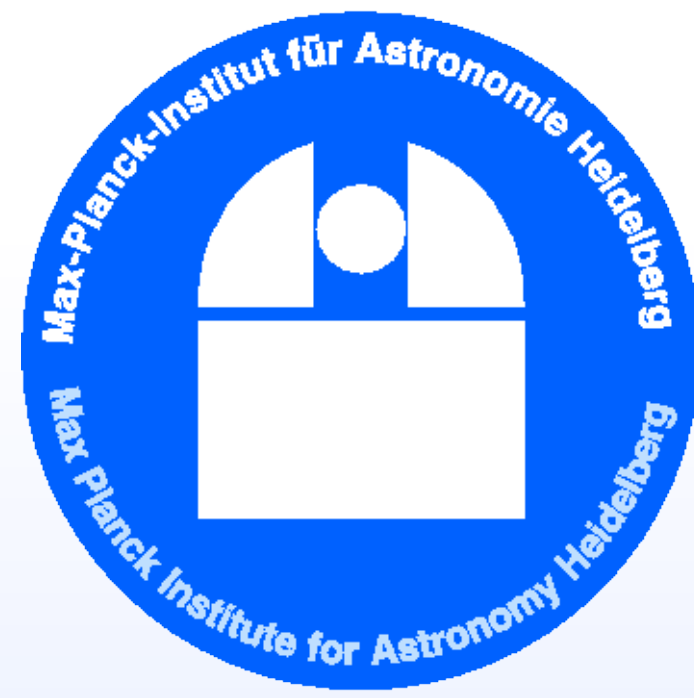
# Software-centric view on OVMS for LBT



J. Trowitzsch, J.L. Borelli, J.-U. Pott, and M. Kürster

Max-Planck-Institute, Königstuhl 17, 69117 Heidelberg, Germany

trowitzsch@mpia.de



## 1 Introduction

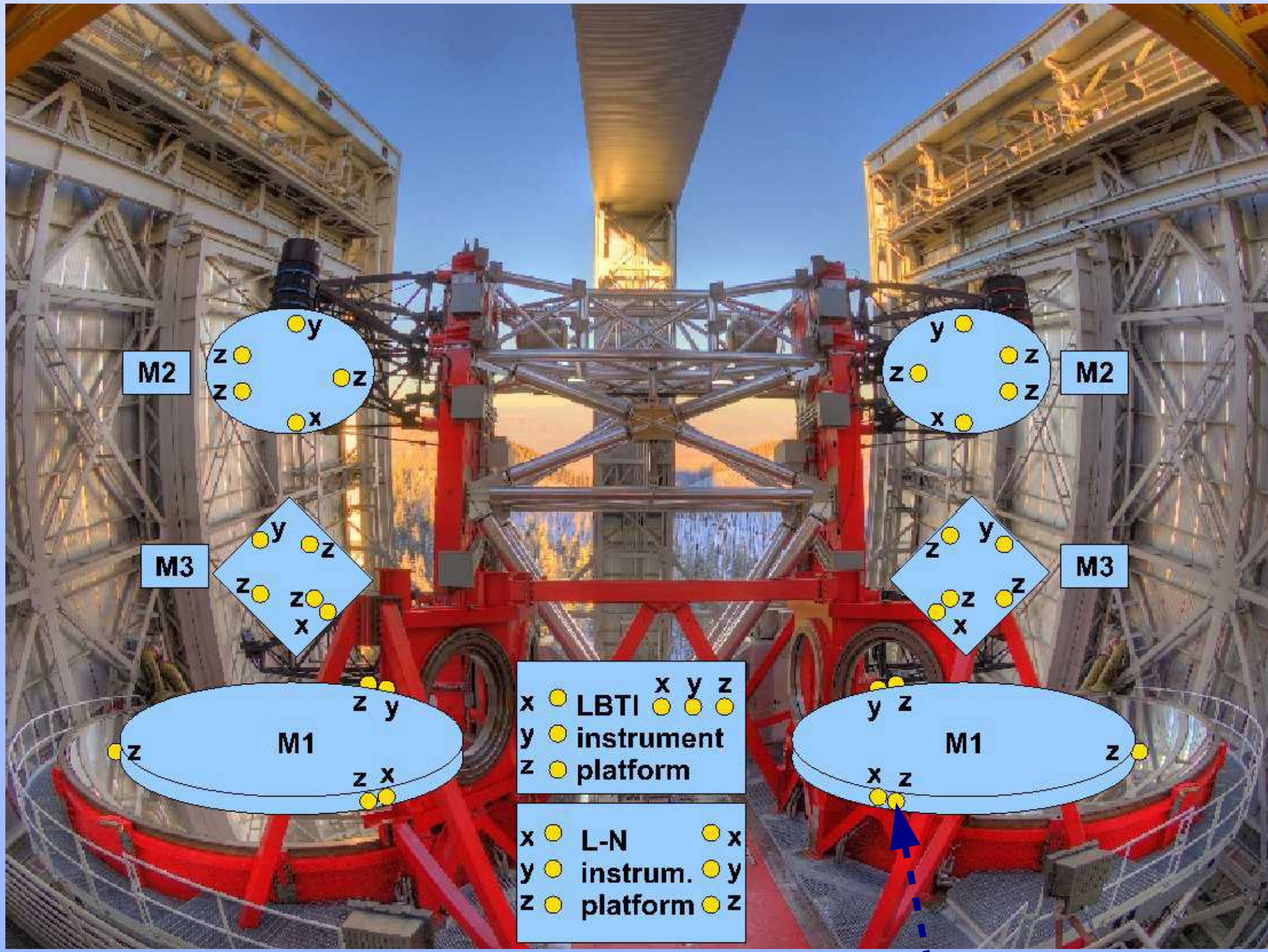
The performance of infrared interferometry (IF) and adaptive optics (AO) strongly depends on the mitigation and correction of telescope vibrations. Therefore, at the Large Binocular Telescope (LBT) the OVMS, the Optical Path Difference and Vibration Monitoring System, is being installed. It is meant to ensure suitable conditions for adaptive optics and interferometry. The vibration information is collected from accelerometers that are distributed over the optical elements of the LBT. The collected vibration measurements are converted into tip-tilt and optical path difference data. That data is utilized in the control strategies of the LBT adaptive secondary mirrors and the beam combining interferometers, LINC-NIRVANA and LBTI.

Within the OVMS the software part is responsibility of the LINC-NIRVANA team at MPIA Heidelberg. It comprises the software for the real-time data acquisition from the accelerometers as well as the related telemetry interface and the vibration monitoring quick look tools. The basic design ideas, implementation details and special features are explained here.

## 2 Features

The LBT optical path difference (OPD) and vibration monitoring system (OVMS) is designed to measure vibrations on the different components of the telescope at a level sufficient to calculate the displacements and variations of its optical path. It is a **distributed system** that is composed of a **real-time data acquisition device** from United Electronics Industries (UEI), 45 PCB 393B05 **accelerometers** distributed over the optical elements of the telescope, and a **set of software tools** to monitor, setup, and control it.

The OVMS software carries out several tasks. The acquired vibration data is provided to possible clients via a multicast node. The raw data can be monitored and processed so that vibration thresholds can be checked. Users will be informed if such a threshold is exceeded. Furthermore, the vibration data is written into the LBT telemetry system for later post-processing.



## 4 Acquisition

The data acquisition is carried out by a **real-time service** that runs on the UEI Rackangle device which is basically a *PowerPC* running a *Xenomai* Linux. The service is in charge of **collecting** and **distributing** the raw vibration data through the network via a multicasting protocol. Several **configuration parameters** which can be selected at command-line start-up are supported. These parameters include the used ICP current in Ampere, the device's operation rate (default 5 kHz), the used Gain, the high-pass filtering type for the input cards, and the details for the multicast connection.

A command line acquisition tool can be used to check the **connectivity** with the UEI data acquisition system. It prints out the time stamp and the raw vibration value of each accelerometer separated by space characters. This data can also be redirected into a file, and later on, be analyzed using GNUPlot, Matlab, or similar tools.

## 3 Conceptual Design

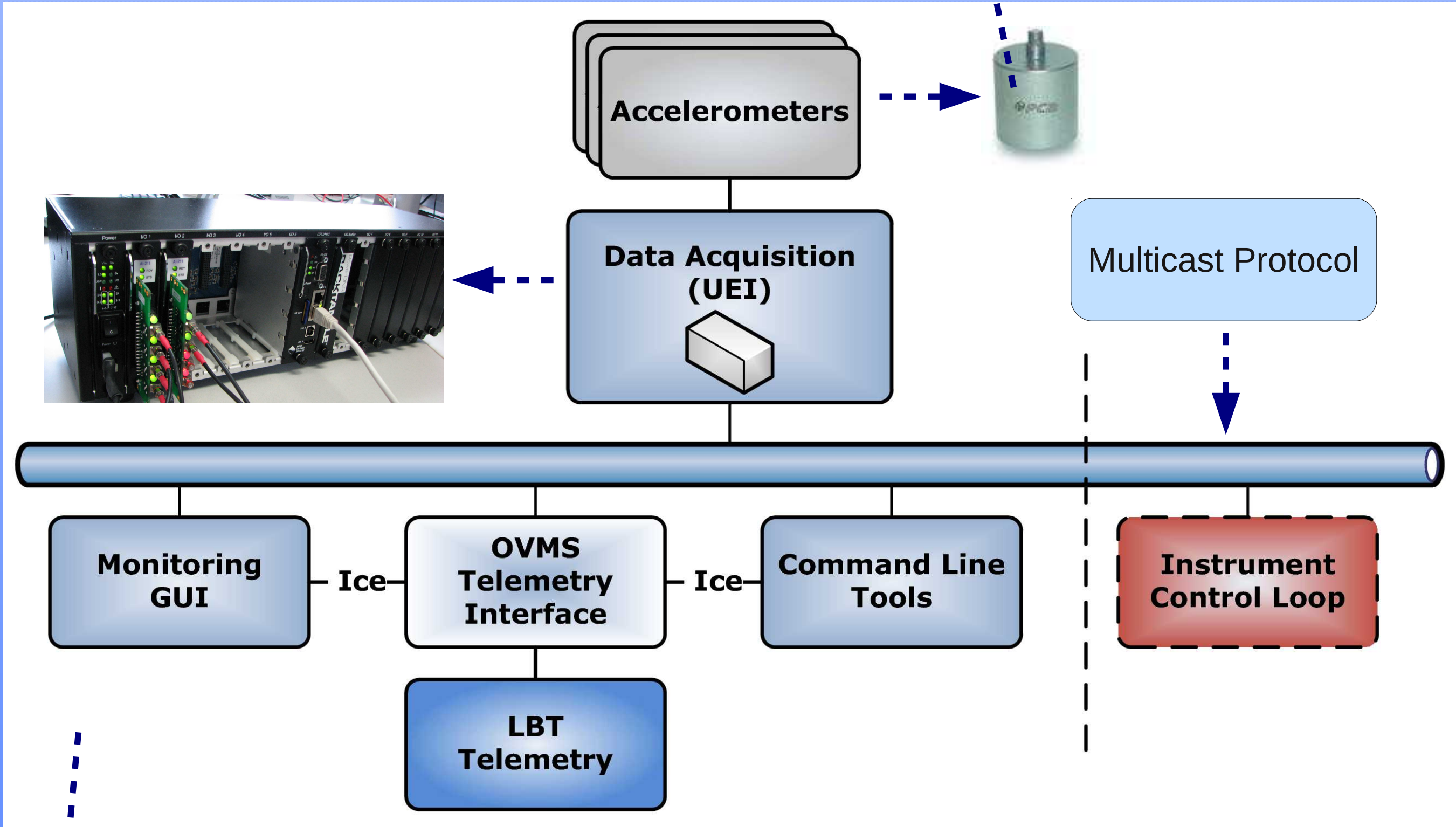
The basic conceptual design of the OVMS software system is shown in the middle. The **real-time data acquisition** service runs on the UEI device and collects the raw vibration data from the different accelerometers. It generates a related **NTP-based time stamp** for that data and finally broadcasts this time-stamped data using a **multicasting protocol**.

Different clients join the related multicast node in order to retrieve the raw vibration data stream. This means several clients receive **exactly the same data**, reducing the network traffic and any possible timing issue.

The OVMS software is implemented using known software packages such as Qt3, Qwt, Ice, or Boost. Additionally special packages like the LBT TCS telemetry and the UEI software development kit are used.

The build-in clients for the OVMS are the **Monitoring GUI**, the **Telemetry Interface**, and some **Command Line Tools**.

Each instrument control loop can connect to the vibration data stream and utilize the data in its **control strategies**.



## 5 Telemetry

The telemetry interface is in charge of collecting the **raw accelerometer measurements** from the multicast node and storing them into the **LBT telemetry subsystem** with a sampling rate of about 1 kHz. It is divided into two parts, the stand-alone telemetry service and the clients. The service is basically a **finite state machine** that works in the background storing data and processing commands from clients.

It provides a simple **set of commands** that can be issued either from the OVMS monitor or from the corresponding telemetry client. This client controls the telemetry service from the command line console. Its purpose is to automatize some procedures, either using scripts or scheduled tasks.

## 6 Monitoring

The OVMS monitors the vibration data. For each accelerometer **individual thresholds** in the spectral domain are specified for separate frequency regions. In order to check these thresholds an **on-the-fly spectral analysis** is done. The data of about one hours is stored in circular buffers for each accelerometer. **Data plots** for both the raw vibration as well as for the spectral data are available.

When a threshold value is exceeded the user is informed by means of an **easy recognizable alarm** as shown below. Details are logged to disk, so that it is easy to trace back what happened. The **status** of the Telemetry Interface Service is monitored as well.

