

5 Architecture

This chapter addresses the architecture of the ATLAS HLT/DAQ system. It starts with a description of how the HLT/DAQ system is positioned with respect to both the other systems of the experiment and external systems such as the LHC machine and CERN technical services. This is followed by a description of the organisation of the HLT/DAQ system in terms of the functions it provides and how they are organised in terms of sub-systems.

The system architecture, represented in terms of implementation independent components, each associated to a specific function, and their relationships is then presented, together with the mapping of the HLT/DAQ sub-systems onto the architecture. This generic architecture has also been used to evaluate the overall system cost, discussed in Chapter 16. Finally, the concrete implementation (baseline) of the architecture is described.

5.1 TDAQ context

The context diagram of the HLT/DAQ is shown in Figure 5-1. It illustrates the inter-relations between the HLT/DAQ system seen as a whole and elements external to it which are directly related to data acquisition and triggering. It also illustrates the type of data which is exchanged in each case. The associated interfaces are introduced in Section 5.2.3, and discussed in more detail in Part 2 of the present document.

The LVL1 trigger provides LVL2 with RoI data needed to guide the LVL2 selection and processing; this interface is discussed in detail in Part 2. The TTC system [5-1] provides signals associated with events that are selected by the LVL1 trigger. RODs associated with the detectors provide event fragments for all events that are selected by the LVL1 trigger. In addition, the LVL1 system contains RODs which provide LVL1 trigger-decision data to be read out for the selected bunch crossings. The LVL1 trigger system, the TTC system and all the ROD systems need to be configured by the DAQ system, e.g. at the start of each run. These components are shown in Figure 5-1.

Interfaces to external systems are also illustrated in Figure 5-1. These connect to the LHC machine, e.g. to exchange information on beam parameters, to the detectors, e.g. to control voltages, to the experimental infrastructure, e.g. to monitor temperatures of racks, and to the CERN technical infrastructure (such as the rack cooling water supply). The TDAQ interface for all these external systems is the DCS. Also shown are the interfaces relating to: long-term storage of event data retained by the HLT prior to offline analysis, and non-event data which have to be stored: alignment and calibration constants, configuration parameters, etc.

5.2 HLT/DAQ functional analysis

This section analyses the HLT/DAQ system in terms of the basic required functions, the building blocks and sub-systems which provide these functions and their internal and external interfaces.

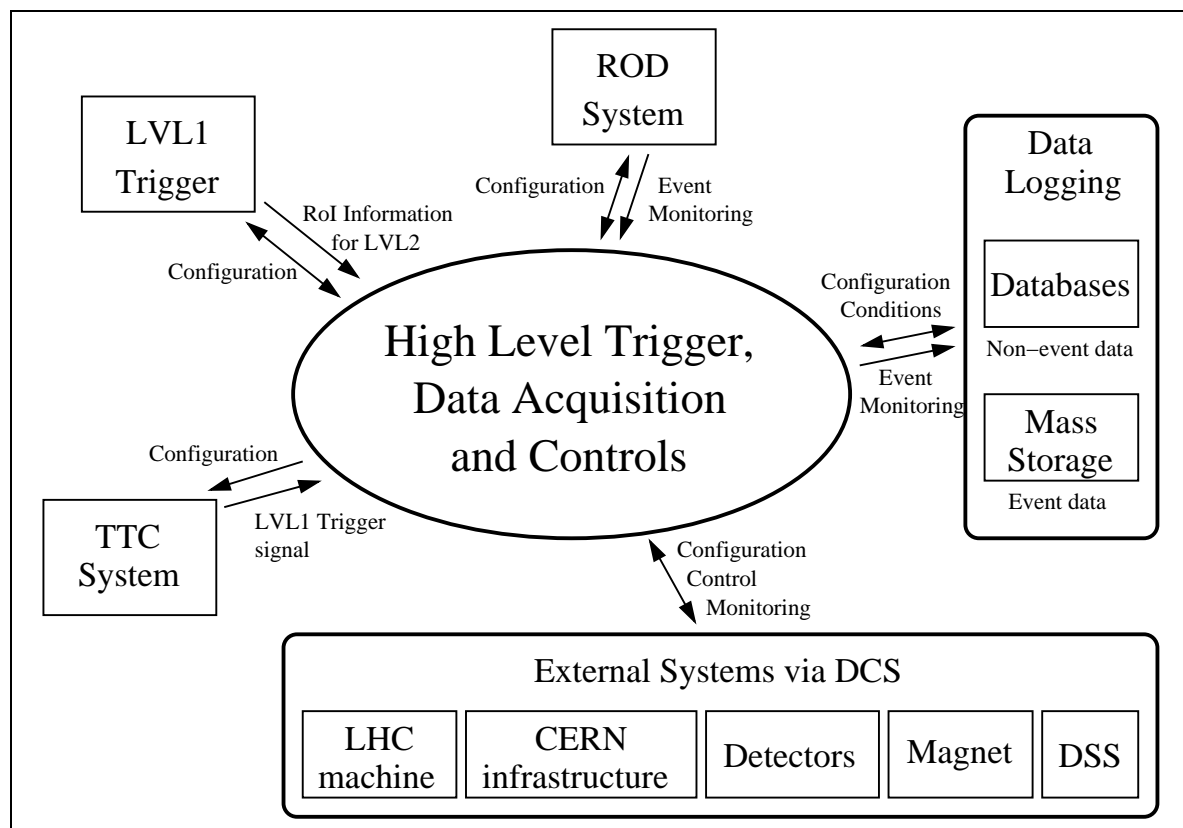


Figure 5-1 Context diagram

5.2.1 Functional decomposition

The HLT/DAQ system provides the ATLAS experiment with the capability of: *moving* the detector data, e.g. physics events, from the detector to mass storage; *selecting* those events which are of interest for physics studies; and *controlling* and *monitoring* the whole experiment.

The following functions are identified:

- **Detector readout**: for events, i.e. bunch crossings, selected by the LVL1 trigger, the data associated with the relevant bunch crossing, are passed through detector-specific modules (RODs) which arrange them in formatted event fragments before sending them on to the HLT/DAQ system. An event arriving at the input to the HLT/DAQ system, i.e. at the ROB, is therefore split in a number of fragments. Quantitatively, there are ~1600 RODs each of which sends one event fragment per event i.e. at the LVL1 rate of up to 100 kHz.
- **Movement of event data**: event fragments buffered in the ROB have to be moved to the HLT and, for selected events, to mass storage. This is a complex process which involves both moving small amounts (typically ~2% of the full event) of data per event at the LVL1 trigger rate (the region-of-interest data for the LVL2 trigger) and the full event, i.e. ~1.5 Mbyte, at the rate of the LVL2 trigger (few kHz).
- **Event selection and storage**: the HLT system is responsible for reducing the rate of events, and selecting those potentially of interest for offline analysis. Events selected by the HLT system are written to permanent storage for offline reconstruction and analysis. The data rate for selected events should not exceed a manageable level of a few hundred Mbyte/s.

- **Controls**: TDAQ and detector online control includes the capability to configure, operate and control the experiment (detector, infrastructure, TDAQ) during data-taking, in testing and calibration runs and also to control certain vital service systems which remain active in periods when most of the detector and the LHC are shut down.
- **Monitoring**: online monitoring includes the capability to monitor the state and behaviour (operational monitoring), and the performance (detector and physics monitoring) of all parts of ATLAS both during physics data-taking and when calibration and testing operations are in progress.

5.2.2 HLT/DAQ building blocks and sub-systems

The HLT/DAQ system is designed to provide the above functions using the following building blocks:

- **ROD crate DAQ**. This is the building block which provides the DAQ for a ROD crate. During detector commissioning, debugging or testing it allows data-taking to be performed on a single ROD crate. It is itself built from the building blocks described in the following paragraphs and is thus an integral part of the overall DAQ system, i.e. during normal experiment operations ROD crate DAQ is operated as an integral part of the overall DAQ.
- **Readout**. The Readout is a building block associated to functions of detector readout and the movement of event data. It provides for the receiving and buffering of event fragments coming from the RODs. The depth of the required buffers is determined by the duration of the LVL2 processing, plus the duration of event building (for events accepted by LVL2) and the time taken to remove¹ the events from the system. In addition, the Readout provides a sub-set of the buffered event fragments to the LVL2 trigger and all² event fragments to the event building.

It also provides the first stage in the HLT/DAQ where event fragments from more than a single ROD can be coherently sampled (with respect to the EL1ID) for the provision of Monitoring functionality.

- **LVL2 processing**. As outlined in Chapter 1, the LVL2 trigger uses the RoI mechanism [5-1] to selectively read out only the parts of each event that it needs to make the selection. Starting from RoI information supplied by the LVL1 trigger, appropriate event fragments are requested from the Readout and used to decide on the acceptance or rejection of the event. Event fragments are requested on the basis of the LVL1 event identifier and the η - ϕ region. The selection process may make several requests for data, but events are rejected immediately one of the algorithmic criteria is not met.
- **RoI Collection**: This is the building block which, in conjunction with the Readout, provides the movement of a sub-set of event data to the LVL2 processing. It maps the η - ϕ region into Readout identifiers, collects the event fragments from the Readout and provides the LVL2 processing with the resulting single data structure representing the RoI data.

1. An event is removed from the ROS if it is rejected by the LVL2 and then following the completion of the event building process for those events which were accepted by the LVL2.
2. It is also foreseen that the quantity of event fragments sent to the event building may depend on the type of event.

- **Event Builder**: This building block provides, for those events passing the LVL2 processing selection criteria, for the movement of the event fragments out of the ROS. It builds, buffers and formats the complete event as a single data structure. Following the building of the event the EB subsequently provides the events to the Event Filter.

It is also the first stage in the HLT/DAQ where complete event fragments may be sampled for the provision of Monitoring functionality.

- **Event Filter**: This final level of event rate and data volume reduction. With the LVL2 processing they provide the Event selection functionality described in Section 5.2.1. It executes complex selection algorithms on complete events which are provided to it by the Event Building. It also provides an additional point in the HLT/DAQ where complete events may be used for the purposes of monitoring.
- **Controls**: This is the building block that provides the configuration, control and monitoring of the ATLAS experiment for and during the process of data taking. In conjunction with the Detector control and monitoring (see below) it provides the control functionality.
- **Detector control and monitoring**: The detector is brought to and maintained in an operational state by this building block. It also performs the monitoring of the state of the detector using the information provided by detector sensors, e.g. flow meters and temperature monitors. In addition it receives and monitors information provided to ATLAS by external systems, e.g. the LHC machine, CERN technical infrastructure.
- **Information services**: This building block provides for the exchange of information between the Controls building block and all other building blocks for the purposes of configuration, control and monitoring. It provides information management, error handling and with making available event fragments, sampled by the Readout and Event Building, for the purpose of event-data-based operational monitoring of the detector.
- **Data bases**: Similarly to the Information services this is a building block that supports the Controls and Detector control and monitoring building blocks with respect to providing the functionality of configuration. It provides for experiment's configuration description and its access by all building blocks other building. In addition, it allows the recording and accessing of information pertaining to the experiments operation during data-taking.

5.2.3 HLT/DAQ Interfaces

The HLT/DAQ system interfaces to a variety of other subsystems inside ATLAS, as well as external systems which are not under the experiment's control. The following sub-sections describe these interfaces with particular reference to the systems being connected, the interface responsibilities and the data exchanged across the interface. References to more detailed documentation on the interfaces, including data formats are also given.

The interfaces can be split into two classes, those to other systems in ATLAS, and those to external systems. Table 5-1 summarizes the characteristics on these interfaces.

Table 5-1 Overview of interfaces between HLT/DAQ and other ATLAS or external systems

Interface	Data Rate	Data Volume	Data Type
LVL1–LVL2	100 kHz	~1 kB/event	RoI data
Detector specific trigger	few kHz	few words	Trigger signals
LVL1 & Detector Front-ends	100 kHz	~150 Gbyte/s	Raw data
Detector Monitoring	few Hz	few Mbyte/s	Raw, processed and control data
Conditions Database	Intermittent	~100 Mbyte/run	System status
Mass Storage Interface	~300 Hz	~450 Mbyte/s	Raw data + LVL2 and EF results

5.2.3.1 Interfaces to other parts of ATLAS

5.2.3.1.1 LVL1–LVL2 trigger interface

Although part of the TDAQ system, the LVL1 trigger is an external subsystem from the point of view of the DAQ and the HLT components.

A direct interface from the LVL1 trigger is provided through the RoIB which receives data from all of the LVL1 subsystems, the Calorimeter trigger, the Muon trigger and the CTP for events retained by LVL1. This information is combined on a per-event basis inside the RoIB, at the LVL1 rate, and passed to the supervisor of the LVL2 system, the L2SV. The interface has to run at full LVL1 speed, i.e. up to 100 kHz. The detailed specification of the LVL1–LVL2 interface is described in Ref. [5-2].

As mentioned above, there are RODs associated with the LVL1 trigger that receive detailed information on the LVL1 processing, e.g. intermediate results, for the events that are selected. The data from the corresponding ROB are included in the event that is built following a LVL2-accept decision and is therefore accessible for the Event Filter processing.

Trigger control signals generated by the LVL1 trigger are interfaced to the rest of the TDAQ system, as well as the detector readout systems, by the TTC system which is documented in detail in [5-1].

5.2.3.1.2 Detector specific triggers

For test beams, during integration, installation and commissioning, and also for calibration runs, it will be necessary to trigger the DAQ for a sub-set of the full system, i.e. a partition, independent of the LVL1 CTP and LVL2, in parallel with other ongoing activities. Detectors are provided with Local Trigger processors (LTP) [5-3] which offer necessary common functions for such running independently of the LVL1 CTP. Triggers provided independently of the CTP are referred to as detector-specific triggers. A DFM is needed for each partition to initiate the building of partial events in that partition.

Any detector-specific trigger will communicate via the TTC system with its corresponding DFM. The DFM component therefore requires a TTC input and a mode where it will work independently from LVL2. A back-pressure mechanism throttles the detector specific trigger via the ROD-busy tree.

5.2.3.1.3 Interface to the detector front-ends

The detector front-end systems provide the raw data for each event that LVL1 accepts. The detector side of the interface is the ROD, while the TDAQ side is the ROB. The connection between the two is the ROL.

From the point of view of the detector, i.e. ROD, the interface follows the S-LINK specification ([5-4]). Implementation details of the ROL can change as long as this specification is followed. All RODs support a standard mezzanine card to hold the actual interface, either directly or on a rear-transition module. Data flow from the RODs to the ROB, while only the link flow control is available in the reverse direction.

This interface has to work at the maximum possible LVL1 accept rate, i.e. up to 100 kHz and at 160 MByte/s.

5.2.3.1.4 TDAQ access to the Conditions Databases

The conditions databases are expected to come from an LHC Computing Grid applications area project, with any ATLAS-specific implementation supported by the Offline Software group. The Online Software system will provide interfaces to the conditions databases for all TDAQ systems and detector applications which require them online. It remains to be studied how accessing the conditions database will affect the HLT performance itself and how frequently such access will need to occur. This is an area that will be addressed in more detail in the offline computing TDR which is currently planned to be published in 2005.

The conditions database will store all time-varying information of the ATLAS detector that is required both for reconstructing and analysing the data offline and for analysing and monitoring the time variation of detector parameters and performance. Components of the HLT/DAQ system will write information into the database and read from it. In most cases read access will occur at configuration time, but the HLT may need to access the database more often.

5.2.3.1.5 Mass Storage

Events that have passed the EF will be written to mass storage. The design and support of the mass storage service will be centrally provided at CERN. However, in the first step of data storage, the Sub Farm Output (SFO) component will stream data directly to files on local disks. These will be large enough to accommodate typically a day of autonomous ATLAS data-taking in the event of failures in the network connecting ATLAS to the CERN mass storage system, or possible failures in the offline prompt reconstruction system. The event-data files are stored in a standard format (see [5-5] and [5-6]) and libraries are provided to read these files in offline applications. In normal running, data will be continuously sent from the local disk storage to the CERN mass storage system.

5.2.3.2 External interfaces

5.2.3.2.1 Interface to the LHC machine, safety systems and the CERN technical infrastructure

All communications between: the LHC machine; the general safety systems; the CERN technical infrastructure; the detector safety system; the magnets and the TDAQ are done via DCS, with the exception of fast signals (such as the LHC 40 MHz clock and orbit signal) that are handled in the LVL1 trigger. These communication mechanisms and interfaces are described in Chapter 11.

5.3 Architecture of the HLT/DAQ system

This section presents the global architecture of the ATLAS HLT/DAQ system. The architecture builds upon the work presented in previous documents that we have submitted to the LHCC: the ATLAS Technical Proposal [5-7], the DAQ, EF, LVL2 and DCS Technical Progress Report [5-8] and the DAQ/DCS/HLT Technical Proposal (TP) [5-9], and further developed by the requirements and performance studies, design, and development of prototypes and their exploitation in test beams, which has been done since the TP. Table 5-2 summarizes the performance required from the different TDAQ system functions.

Table 5-2 Required system performance capabilities for 100 kHz LVL1 accept rate

Function	Input requirements	Output requirements
Detector readout	~1600 event fragments of size typically 1 Kbyte at 100 kHz	Few per cent of input event fragments to LVL2 at 100 kHz; ~1600 event fragments at a few kHz to EB
Level-2	Few per cent of event fragments at 100 kHz	100 kHz decision rate (~3 kHz accept rate)
Event Builder	~1600 event fragments at ~3 kHz	~3 kHz at ~4.5 Gbyte/s
Event Filter	~3 kHz / ~4.5 Gbyte/s	~300 Hz / ~450 Mbyte/s

The numbers presented in this table are given based on a 100 kHz LVL1 accept rate (we recall that the ATLAS baseline assumes a LVL1 accept rate of 75 kHz, upgradeable to 100 kHz), and therefore represent an upper limit in the required design and performance capabilities of the system. Detailed simulation of the LVL1 trigger (see Section 13.5) conclude that its accept rate at LHC start-up luminosity ($2 \times 10^{33} \text{ cm}^{-2} \text{ s}^{-1}$) will be ~20 kHz. Simulation of the LVL2 trigger at this luminosity indicate a rejection rate of a factor of ~30. This rejection has been linearly extrapolated to 100 kHz to give the numbers in the table. This is probably a pessimistic assumption (*see SECTION 14.x*). The rate at the output of the EF is, however, quoted as that expected from the above simulations. The output capacity of the EF could be increased, but will be limited finally by offline data processing and storage capabilities

The architecture is presented following the functional breakdown given in the previous sections. The architectural components are described from the functional point of view, without reference to possible implementations at this stage. The baseline implementation of this architecture is presented in Section 5.5.

5.3.1 Architectural components

Figure 5-2 depicts the architectural components defined in the following text.

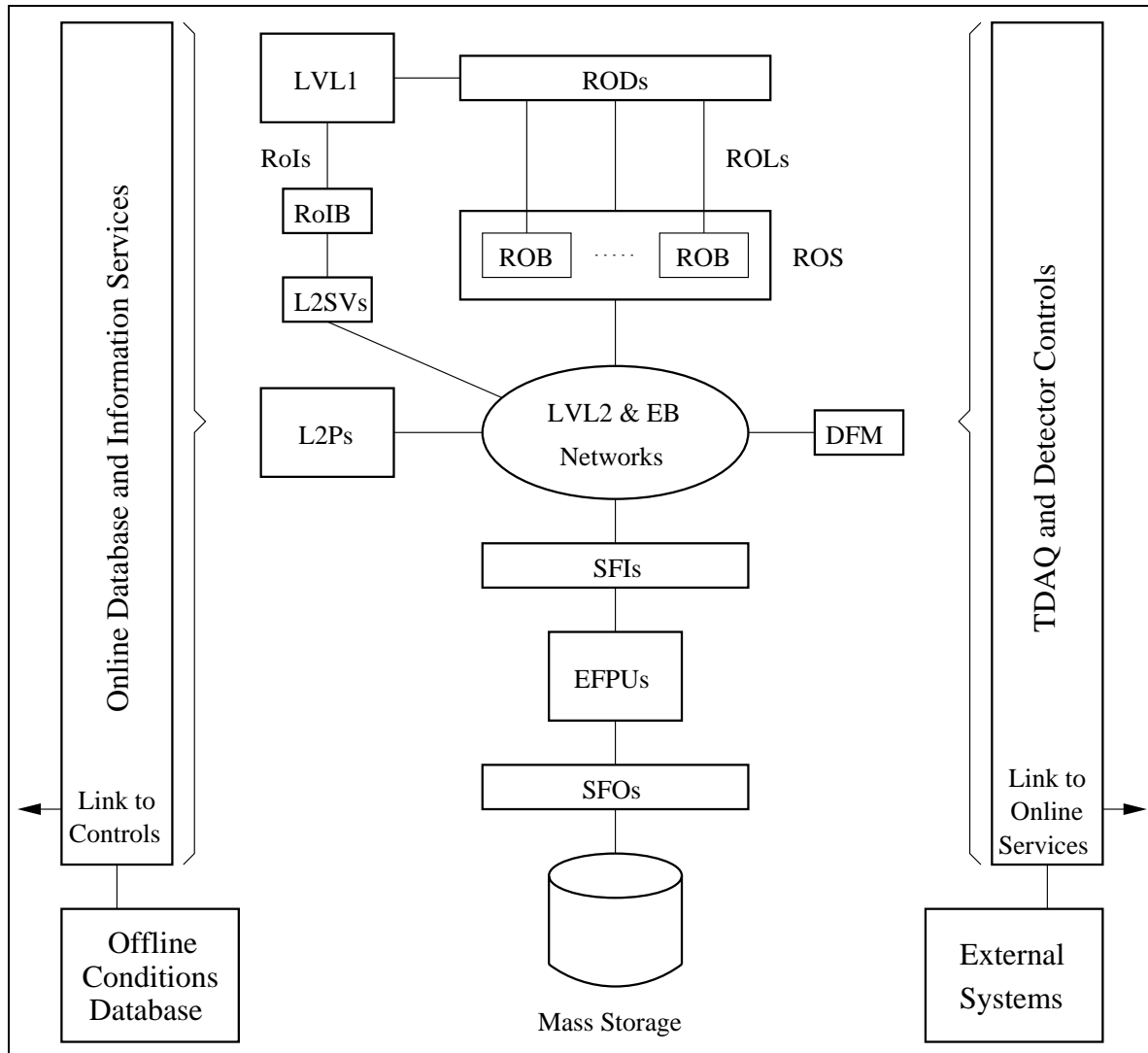


Figure 5-2 General architectural components and their relations

5.3.1.1 Detector readout

The communication link from the detector readout units (RODs), to the ROBs is the ROL. Each ROL carries one ROD fragment per event. The ROL must be able to transport data at a rate equal to the average event fragment size times the maximum LVL1 rate (possibly up to 160 Mbyte/s) — in the calculation, the average event fragment size should be taken as the average over LVL1 triggers for the ROD that sees the largest event fragments on average.

ROBs are the memory buffers located at the receiving end of the ROLs, there being one ROL associated to one ROB. Several (typically two or four) ROBs are physically implemented on a ROBin card and several ROBins can be located in a ROS — the number depending on the implementation option (see Section 5.5.4). The ROS and its component ROBs provide two functions: buffering event data and serving requests for event data from the HLT.

The ROBins are of a unique design used by all the ATLAS sub-detectors¹. The detector event fragments are sent from the RODs and stored in the ROBs. One or more event fragments may be stored in a single ROBin for the same event. The ROBs buffer the event fragment for the time needed by LVL2 to reject or select the event.

The ROS is the component for serving data from the ROBs to LVL2 and the EB. In order to reduce the number of connections into the LVL2 and EB networks it funnels a number of ROBs into a single port for each network. This ROB multiplexing capability is known as the ROB-ROS merging (RRM) function. The ROS also includes a software local controller of the online software (see Section 10.5), for the purpose of controlling data taking and local monitoring.

5.3.1.2 LVL2

The LVL2 trigger uses RoI information, provided by LVL1 via the RoIB, to request relevant event fragments from the ROS's (at the granularity of individual ROBs). Using these data, it produces a decision on the event and delivers the decision together with data it produced during its algorithmic processing back to the Data Flow components.

The RoIB receives information from LVL1 following each LVL1 trigger, allocates a LVL2 Supervisor (L2SV - see below) for the event, and transmits aggregate LVL1 information, after formatting, to the allocated supervisor. The RoIB will run at the rate of the LVL1 trigger. The L2SV then assigns, according to a load-balancing algorithm, a LVL2 Processor (L2P) from a pool under its control to process the event, and forwards the LVL1 information provided by the RoIB to an event handling process, the LVL2 processing unit (L2PU), running on the L2P. The L2PU, using this LVL1 information, requests event fragments from the ROS, processes the RoI data, and makes a decision to accept or reject the event. In the course of processing the event data, additional requests for event fragments may be issued in several steps. The L2PU may, according to a pre-defined algorithm, decide to flag for accept, events which have in fact been rejected by the algorithm processing. This would be done in order to monitor the LVL2 selection process either in the EF or offline. The final accept/reject decision is sent back to the L2SV. If an event is rejected, the decision is passed to the ROS via the DFM in order to remove the event from the ROBs. If an event is accepted, the decision is forwarded to the DFM, which then initiates the event building operation for the event.

A switching network, the L2N (LVL2 Network) component, links the ROS's, the L2SV and the L2Ps. The L2Ps will be organized into sub-farms in a manner which optimizes the use of the L2N.

Online software local controllers are associated to the RoIB, the L2SVs and L2P sub-farms, for the purpose of controlling and monitoring the LVL2 trigger.

5.3.1.3 Event Builder

Events accepted by LVL2 are fully assembled and formatted in the EB's destination nodes, the SFIs. The DFM component is informed by the L2SV of the LVL2 decision. For each accepted event, the DFM, according to a load-balancing algorithm and other selective allocation consid-

1. Contrary to the ROD, which is a specialised detector-specific module, there is a single ROBin implementation for the whole experiment. RODs for the different detectors implement detector-dependent functions. For example, digital signal processing is implemented in the case of the LAr sub-detector.

erations (e.g. special triggers and monitoring - see below), allocates an SFI. For rejected events and for events which have been successfully built, the DFM initiates a procedure to clear these events in the ROSs.

When the LVL2 system is not in use, for example during a detector calibration run, the DFM provides a LVL2-bypass mechanism. It is informed when new events are available for building, directly by the LVL1 CTP system or the Local Trigger Processor (see Section 5.2.3.1.2) of the detector being calibrated, via the TTC network. In the event of several TDAQ partitions being run in parallel, each running partition has a dedicated DFM for event building initiation within that partition.

The DFM has the additional capability to assign SFIs on the basis of pre-defined criteria, e.g. forced-accepted LVL2 events, special LVL1/LVL2 trigger types (as defined by the detector systems), the LVL1 or the LVL2 triggers etc.

The SFI allocated by the DFM requests and receives event data from the ROSs, builds and formats the event in its memory. It notifies the DFM when a complete event has been built, correctly or otherwise, e.g. when expected event fragments are missing. In the latter case, the SFI attempts corrective action, e.g. re-initiating the build. Built events are buffered in the SFIs in order to be served to the EF. For efficiency reasons, the SFI can build more than one event in parallel.

A switching network, the EBN (Event Builder Network) component¹, links ROSs, SFIs and the DFM. The network enables the building of events concurrently into all the SFIs at an overall rate of a few kHz.

Online software local controllers are associated to the DFM and SFIs for the purpose of controlling and monitoring the event building.

5.3.1.4 Event Filter

The EF comprises a large farm of processors, the EFPU components. Each EFPU deals with complete events served by the SFIs, as opposed to the selected event data used by the LVL2 trigger. From the architectural point of view the EF is a general computing tool for analysis of complete events — either events produced by the full ATLAS detector or the set of event data associated to a detector partition. Indeed, it is also envisaged to use all or part of the EF for off-line computing purposes, outside of ATLAS data taking periods.

Each EFPU runs an EF data flow control program (EFD) which receives built events from the SFIs. Several independent Processing Tasks (PTs) continuously process events which are allocated to them on demand by the EFD. Using offline-like event reconstruction and selection algorithms, the PT processes the event and produces a final trigger decision. When a given PT has completed processing an event, it requests a new one to be transferred from an SFI via the EFD. Data generated by the PTs during processing are appended to the complete raw event, if accepted, by the EFD. Accepted events are classified and moved to respective Sub-Farm Output buffers (SFOs), where they are written into local disk files. Completed files are accessed by a mass storage system for permanent storage (Section 5.2.3.1.5). Note that EFDs may send events to one

1. The logically separate LVL2 and EB networks (which fulfil two different functions) could be implemented on a single physical network; in particular, this might be the case in the early phase of the experiment when the full performance is not required.

of several parallel SFO output stream for further dedicated analysis, e.g. express analysis, calibration, debugging.

The EFPUs are organized in terms of clusters or sub-farms, with each sub-farm being associated to one or more SFI and SFO. The minimum granularity of EFPUs as seen from partitions is the sub-farm. SFIs, SFOs and EFPUs are interconnected via a switching network, the EFN (EF Network) component.

The EF sub-farms may also be used for purposes other than event triggering and classification. A sub-farm may be allocated to running detector calibration and monitoring procedures which require events possessing specific characteristics. As discussed above, the DFM can assign events to dedicated SFIs from which EF sub-farms can request those events for dedicated analyses.

Monitoring and control of the EF itself is performed by a number of local controllers which interface to the online system.

5.3.1.5 Online software system

The Online Software system encompasses the software to configure, control and monitor the TDAQ system (known as the TDAQ control system), but excludes the management, processing and transportation of physics data. Examples of the online software services are: local control processes which interface to other TDAQ components, as noted above, general process management, run control, configuration data storage and access, monitoring and histogramming facilities. A number of services are also provided to support the various types of information exchange between TDAQ software applications. Information can be shared between applications in the same sub-system, across different sub-systems, and between TDAQ systems and detectors.

TDAQ and detectors use the configuration databases to store the parameters and dependencies which describe their system topology and the parameters which are used for data-taking. The offline conditions databases are used to read and to store data which reflect the conditions under which the event data were taken.

The software infrastructure integrates the online services with the rest of TDAQ into a coherent software system. The hardware infrastructure comprises the Online Software Farm (OSF), the computers on which the services run, and a Online Software switching network (OSN) which interconnects the OSF with other TDAQ components including the DCS for control and information exchange as well as detector components which require access to online software facilities. The OSF will include machines dedicated to monitoring event data and database servers which hold the software and firmware for all of the TDAQ system — there will be servers local to clusters of computers which perform a common function, as well as central, backup servers.

Online software, services and infrastructure, are described in detail in Chapter 10.

5.3.1.6 Detector Control System

The DCS has a high degree of independence from the rest of the HLT/DAQ system, being required to be able to run at all times, even if HLT/DAQ is not available. At the level of detail suitable for this chapter, the DCS is a single component, without internal structure. It is inter-

faced to the rest of the system via the Online Software system. The interplay between DCS and TDAQ control is discussed in Section 5.3.2.

DCS is the only ATLAS interface with the LHC machine, with the exception of that for fast signals (which is handled directly by the LVL1 system). A detailed definition and description of the DCS architecture, interfaces and implementation is given in Chapter 11.

5.3.2 Overall experimental control

The overall control of ATLAS includes the monitoring and control of the operational parameters of the detector and of the experiment infrastructure, as well as the supervision of all processes involved in the event readout and selection. This control function is provided by a complementary and coherent interaction between the TDAQ control system and the DCS. Whilst the TDAQ control is only required when taking data or during calibration and testing periods, the DCS has to operate continuously to ensure the safe operation of the detector, and a reliable coordination with the LHC control system and essential external services. DCS control services are therefore available for the sub-detectors at all times. The TDAQ control has the overall control during data-taking operations.

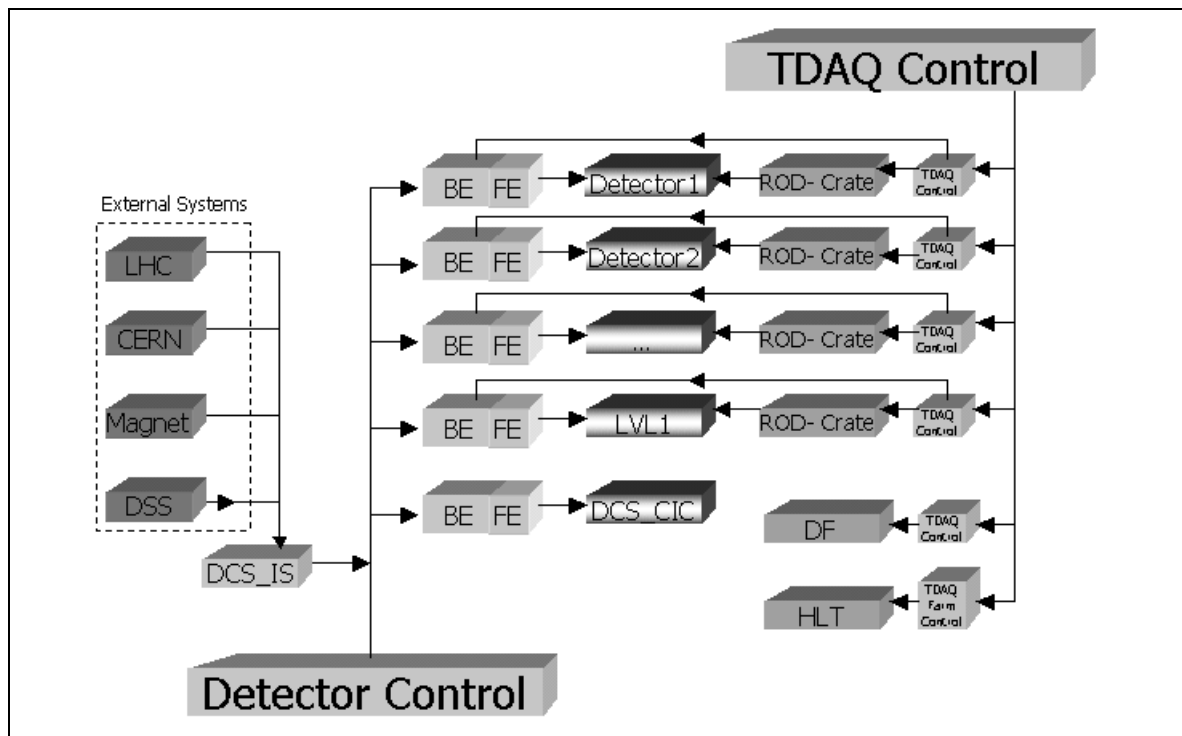


Figure 5-3 Logical experiment controls flow

There is a two-way exchange of information between DCS and the TDAQ control via mechanisms defined and provided by the Online Software (see Chapter 10). DCS will report information about the status and readiness of various components which it controls and monitors to the Online system. The Online system will provide both configuration information and issue commands related to run control to DCS. Figure 5-3 presents the logical experiment control flow. In the figure, the lines represent the bi-directional information exchange between the systems, while the arrows on the lines indicate the direction of the command flow. The TDAQ control

and DCS actions are coordinated at the sub-detector level. Each sub-detector, as well as the HLT and Data Flow systems, have TDAQ control elements responsible for the data-taking control. DCS control elements, called the Back-End (BE), control the sub-detector Front-End (FE) and LVL1 trigger equipment. DCS also controls the hardware infrastructure which is common to all the detectors, the DCS-Common Infrastructure Controls (DCS-CIC). The interaction with the LHC machine, which is explained in Chapter 11, and other external services is handled by DCS via the Information Service (DCS-IS). Experiment control is addressed in more detail in Chapter 12.

5.4 Partitioning

As already discussed in Chapter 3, partitioning refers to the capability of providing the functionality of the complete TDAQ to a sub-set of the ATLAS detector. For example, the ability to read out all or part of a specific detector with a dedicated local trigger in one partition, while the other detectors are data-taking in one or more independent partitions.

The definition of the detector sub-set defines which ROBs belong to the partition (because of the connectivity between RODs and ROBs). For example, if a partition of the muon MDT chambers is required, the ROBs associated to the MDT RODs will be contained in the partition. Downstream of the ROBs a partition is realised by assigning part of TDAQ (EBN, SFI, EF, OSF and networking) to the partition — this is a resource-management issue. Some examples of TDAQ components which may have to be assigned to a TDAQ partition are: a sub-set of the ROBs, as mentioned above, and a sub-set of the SFIs. The precise resource allocation in a particular case will depend on what the partition is required to do. For example, in order to calibrate an entire detector, all that detector's allocated ROBs, a fraction of the event building bandwidth and several EF sub-farms as well as online software control and monitoring functions may be required.

As regards the transport of the data to the allocated resources, the DFM allocates SFIs responsible for event building from a sub-set of ROSs. In order for this to happen, in the case of partitions associated to non-physics runs (i.e. when there is no LVL2) but which require functionality beyond the ROS, the DFM must receive, via the TTC, the triggering information for the relevant partition. Hence the need for full connectivity between the DFMs and the TTC partitions.

5.5 Implementation of the architecture

5.5.1 Overview

This section defines a concrete implementation for each of the architectural components described in the previous sections. The choices for what we call the baseline implementation have been guided by the following criteria:

- The existence of working prototypes.
- Performance measurements which either fulfil the final ATLAS specifications today or can be safely extrapolated to the required performance on the relevant timescale (e.g. extrapolating CPU speed of commodity PCs according to Moore's law).

- The availability of a clear evolution and staging path from a small initial system for use in test beams, to the full ATLAS system for high-luminosity running.
- The overall cost-effectiveness of the implementation and the existence of a cost-effective staging scenario.
- The possibility to take advantage of future technological changes over the lifetime of the experiment.

The proposed baseline implementation is a system that could be built with today's technology and achieves the desired performance. It is expected that technological advances in the areas of networking and computing will continue with the current pace over the next few years; these will simplify various aspects of the proposed architecture and its implementation. Optimization in the area of the ROB I/O (see Section 5.5.4) remains to be performed prior to freezing details of the implementation.

By making use of commercial off-the-shelf (COTS) components based on widely-supported industrial standards wherever possible, the architecture will be able to take advantage of any future improvements from industry in a straightforward way. Only four custom components are foreseen in the final HLT/DAQ/DCS system as such¹: the DCS ELMB module; the RoIB, of which only a single instance is needed; the ROL implementation; and the ROBin, which implements the ROL destination and the ROB functionality. Components of the TTC system are also custom elements but are common across the entire experiment (and indeed used in other LHC experiments).

The component performance and overall system performance figures which help to justify the proposed implementation can be found in Part 3 of this document.

Figure 5-4 depicts the baseline implementation. Table 5-3 lists the principal assumptions from which the size of the implementation has been determined. Table 5-4 presents more details on the size of the system; it lists the components that make up the baseline implementation and, for each one, gives the number of units and the associated technology.

5.5.2 Categories of components

The implementation calls for the use of a small number of categories of components, which are either custom or commercial, as follows:

- **Buffers**: These are used to decouple the different parts of the system: detector readout, LVL2, EB and EF. Because of the parallelism designed into the system, buffers fulfilling the same function, e.g. ROBs, operate concurrently and independently.
- **Processors**: These run event selection algorithms, monitor and control the system. They are typically organized in farms, i.e. groups of processors performing the same function.
- **Supervisors**: These are generally processors dedicated to coordinating concurrent activities, in terms of assigning events to processors and buffers at the different levels: the LVL2 trigger (supervised by the RoIB and L2SV units), the EB (supervised by the DFM) and the EF (supervised internally by its data flow control).

1. The ROL source cards and the ELMB (see Chapter 11) are custom components which are specified and produced under the responsibility of the TDAQ project. However, these are integrated in the detector systems.

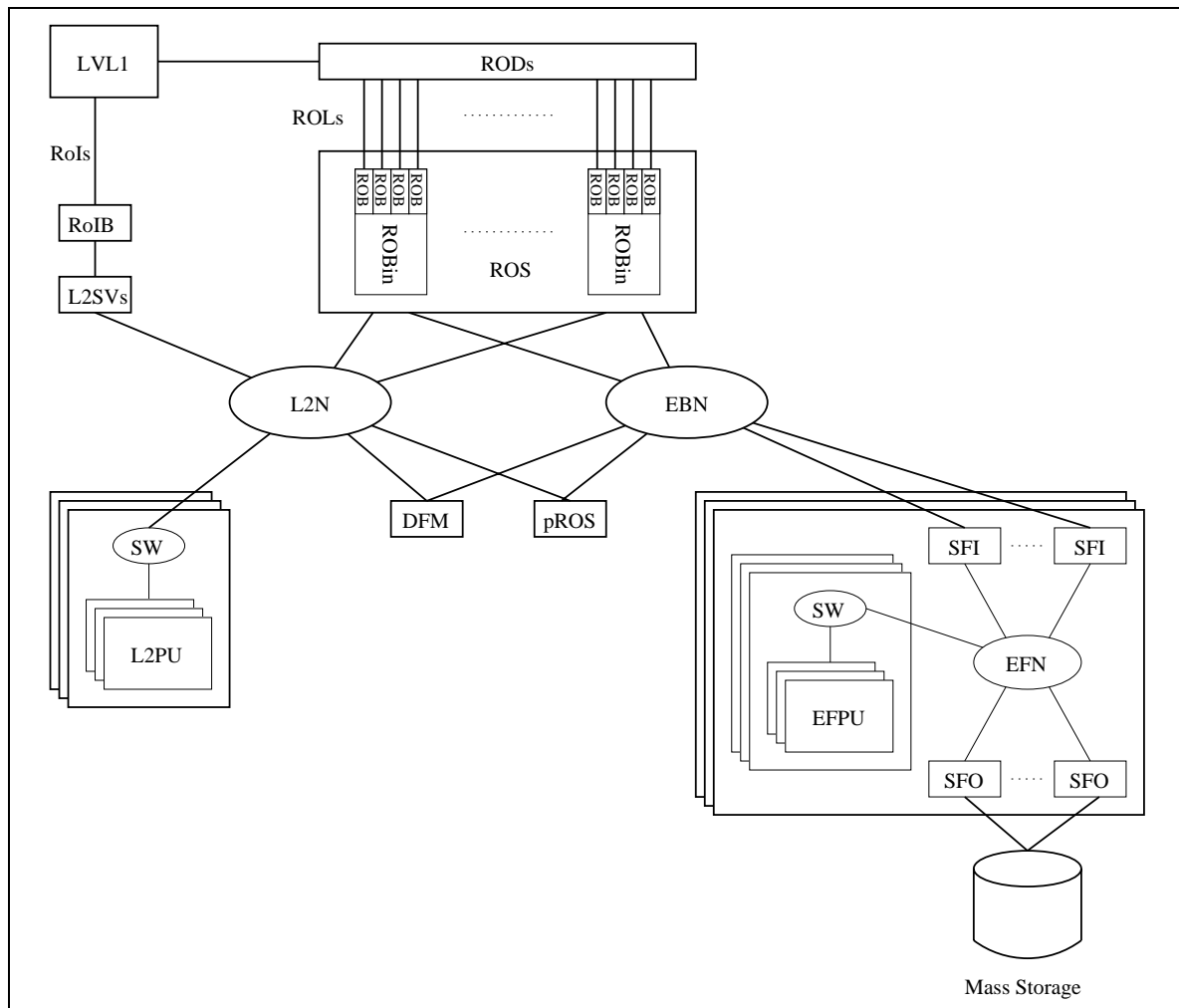


Figure 5-4 Baseline implementation

Table 5-3 Assumptions

Parameter	Assumed value	Comments
LVL1 rate	100 kHz	HLT/DAQ designed at this LVL1 rate
Event size (maximum)	2 Mbyte	Assumes maximum fragment size for all sub-detectors during normal high-luminosity running
Event size (average)	1.5 Mbyte	Assumed average value
LVL2 rejection	~30	EB rate of 3.3 kHz
RoI data volume	2% of total event size	Detector data within an RoI - average value
L2P rate (events/second)	~100	LVL2 decisions/sec/processor Assume dual-CPU (8 GHz) machines
EFPU rate (events/second)	~1	EF decisions/sec/processor Assumed dual-CPU (8 GHz) machines

- **Communication systems:** These connect buffers and processors to provide a path for transporting event data or a path to control and operate the overall system. Communica-

Table 5-4 Baseline implementation and size at 100 kHz Level-1 rate

Component	Number of elements	Technology	Comments
ROL	~1600 Links	Custom	Follows S-Link specification.
ROB	~400 ROB's	Custom module	One ROB multiplexes and buffers data from 4 ROL's.
ROS ^a	~60 switches ~60 PCs	Gigabit Ethernet Industrial PC	10(ROB's)x4(uplinks) concentrating factor. PC houses ~10 ROL's. ROB's are addressed individually via the concentrator switch.
	~150 PCs	Industrial PC	Multiplexes 3 ROB's. Relays Rol requests to appropriate ROBIn. Builds super-fragment for the Event Builder.
LVL2 Network	~650 ports	Gigabit Ethernet	Connects ROS, L2P, L2SV, DFM.
EB Network	~250 ports	Gigabit Ethernet	Connects ROS ,SFI, DFM.
RoIB	1 unit	Custom Module	
L2SV	~10	Dual-CPU PC ^b	One L2SV every 10 kHz of Level-1 trigger rate.
DFM	~35	Dual-CPU PC ^b .	One per TTC partition.
L2P	~500	Dual-CPU PC ^b .	Run LVL2 selection algorithms.
SFI	~90	Dual-CPU PC ^b .	Build (and buffer) complete events.
EFPU	~1600	Dual-CPU PC ^b .	Run EF selection algorithms.
EF Network	~1700 ports	Gigabit Ethernet	Connects SFI, EFPU and SFO.
SFO	~30	Dual-CPU PC ^b . and ~1 Tbyte disk storage	Buffers events accepted by EF and stores them on permanent storage.
File Servers	~100	Dual-CPU PC ^b . with ~1 Tbyte of disk space	Holds copy of databases and software. Local to a group of functionally homogeneous elements (e.g. a group of EFP).
Data base servers	~2	RAID based file server	Hold master copy of databases, initialisation data and down-loadable software.
Online farm	~50	Standard PCs	Operate and monitor the experiment. Runs online services.
Local control switch	~100	Gigabit Ethernet	Connects a group of ~30 elements (e.g. a LVL2 sub-farm) to the central control switch.
Central control switch	250 ports	Gigabit Ethernet	Connects online farm, local control switches, and other system elements.

- a. The ROS row indicates the number of components for both the switch-based (top numbers in the row) and bus-based ROS (bottom numbers).
- b. Assume 8 GHz CPU clock rate.

tion systems are present at different locations in the system. Some of them provide a multiplexing function, i.e. they concentrate a number of input links into a smaller number of

output links. Depending on how the architecture is physically realised, a multiplexer may have a physical implementation (e.g. a switch, or a bus) or not (viz. a one-to-one connection, without multiplexing).

5.5.3 Custom Components

5.5.3.1 The ELMB

Text from HB

5.5.3.2 The Read Out Link

The ROL will be implemented based on the S-LINK protocol [5-4]. It is discussed in detail in Section 8.1.2. The RODs will host the link source cards, either as mezzanines mounted directly on the RODs or on rear-transition modules. The ROBins, located near the RODs in USA15, will host several link destination cards. About 1600 links will be needed.

5.5.3.3 The ROBin

The ROBin is implemented as a PCI board receiving data from a number of ROLs. Each ROBin has a PCI interface and a Gigabit Ethernet output interface [5-10]. The high-speed input data paths from the RODs are handled by an FPGA. The buffers (associated to each input ROL) will be big enough to deal with the system latency (LVL2 decision time, time to receive the clear command after a LVL2 reject and time to build the complete event following a LVL2 accept). A PowerPC CPU is available on each ROBin. The prototype ROBin is currently implemented on a single PCI board implementing two input links. The final ROBin design is expected to support four ROL channels.

5.5.3.4 The Region-of-Interest Builder

The RoIB design is modular and scalable and is custom designed and built. It is described in detail in [5-11]. The link interface from LVL1 to the RoIB follows the S-LINK specification. There are inputs from eight different sources to the RoIB (Section 5.2.3.1.1). Data flow from the LVL1 system to the RoIB, with the link interface providing only flow control in the reverse direction. Asserting XOFF is the only way for the RoIB to stop the trigger. The output of the RoIB is sent via S-LINK or via Gigabit Ethernet to one of the L2SV processors.

5.5.4 ReadOut System

The ROS is implemented as a rack-mounted PC. Each ROS contains several ROBins, and has connections to the LVL2 and EB networks. Each ROBin multiplexes up to four ROLs into a single physical output channel. The ROS multiplexes the ROBin outputs onto the central LVL2 and EB networks (the RRM function, see Section 5.3.1.1). The RRM multiplexing factor depends on the physical number of links between the ROS and each of the central networks. The maximum factor can be calculated from external parameters, in particular the average RoI size, the peak RoI fragment request rate per ROB, and the LVL2 rejection power.

The RRM function may be implemented in two different ways:

- The bus-based ROS — the ROS PC contains three ROBins, each connected to its own PCI-bus and each having four ROL inputs to four ROBs, and one PCI output. This output is connected to the ROS's PCI-buses, which implement the RRM function. Requests, coming from LVL2 or the EB, for event fragments in the ROB, are handled directly by the ROS PC which aggregates the event fragments over its PCI-buses before dispatching them to the LVL2 or the EB. Gigabit Ethernet interfaces connect the ROS to the LVL2 and EB networks.
- The switch-based ROS — the ROS PC houses typically 10 ROBins, each having four ROL inputs to four ROBs, and one Gigabit Ethernet output. The RRM function is implemented typically using a 10 x 4 Gigabit Ethernet switch, which concentrates the ROBin outputs directly into four Gigabit Ethernet outputs: two for the LVL2 network and two for the EB network¹. The switch-based ROS is understood to include the switch as well as the PC. The ROS PC itself, does not play any role in the process of transferring data between the ROBs, and the LVL2 and EB. It is solely responsible via its PCI-bus for the physical housing of the ROBins, their initialisation and control, and some monitoring functions.

Bus-based and switch-based ROS's are depicted in Figure 5-5. Preliminary studies have already been made on both implementation options, details can be found in Refs. [5-12] and [5-13]. The current ROBin prototype allows access to the ROB data via both the PCI-bus and the Gigabit Ethernet interfaces. The optimization of the ROS architecture, as indicated previously, will be the subject of post-TDR studies using this prototype. The final decision on an optimized design for the ROS architecture will be taken based on the results of these studies on a timescale compatible with the production of the ROBins (see Chapter 18).

As noted above, the ROBins, and therefore the ROSs will be located underground in the USA15 cavern.

5.5.5 Networks

All networks use Gigabit Ethernet technology despite the fact that in some cases, e.g. for the EFN, Fast Ethernet (100 Mbyte/s) would be sufficient to satisfy the requirements today. Considerations of overall network uniformity, technology evolution, and expected cost evolution justify this choice.

Following the first level of ROB concentration in the ROS (see Section 5.5.4), the topology of the EB and LVL2 networks consists of one or two central switches each, connecting sources (i.e. ROS's) and destinations (e.g. SFIs and L2Ps). These central networks will, logically, be monolithic in the sense that each network will connect to all of its sources and destinations. However, they may be physically organised either in terms of large monolithic switches or in terms of combined small switches. The detailed network topology will be fixed at the time of implementation.

Given the number of L2Ps (typically ~500 dual CPUs), small concentrating switches are used around the L2Ps to reduce the number of ports on the central switches. This is possible since the bandwidth per processor is much less than Gigabit Ethernet link capacity.

1. Whether one, two or more outputs are used each, to the EB & LVL2 networks will depend on how many switches are used to implement these networks.

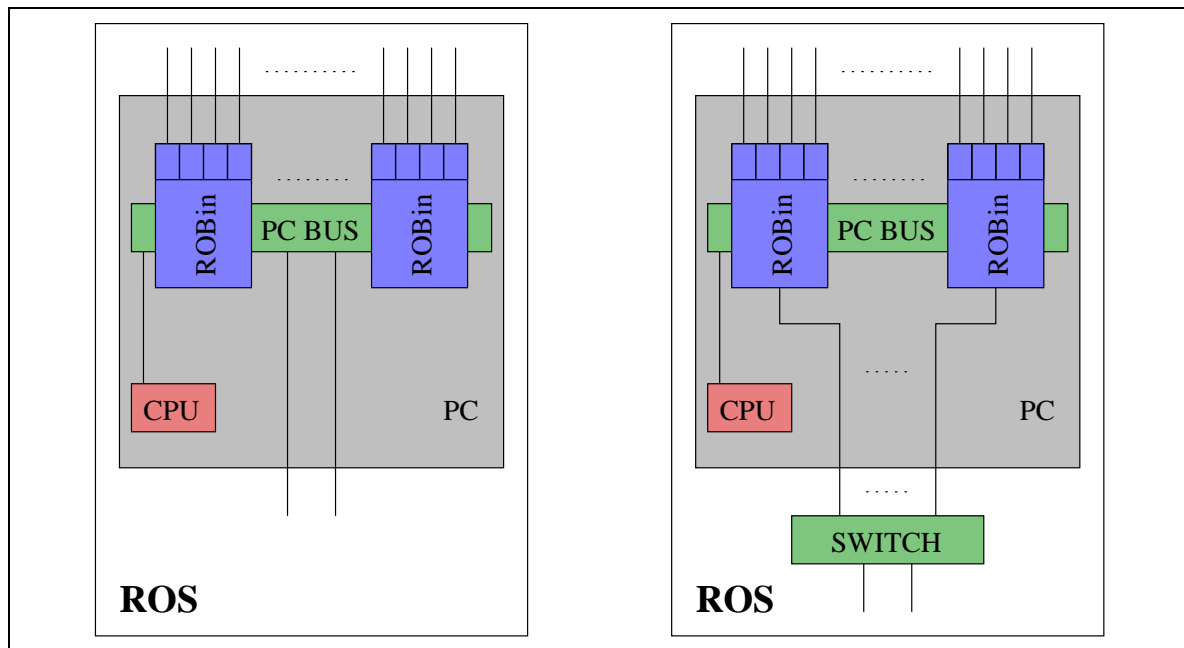


Figure 5-5 Bus and switched based ROS's

The organisation of the OSN, reflects the organization of the online farm. It is hierarchical, with networks local to specific functional elements (e.g. an EF sub-farm) and up-links to a central network providing the full connectivity. Typically, the OSN will be organized in terms of small ($O(40)$ ports) switches, local to groups of components with common functions (e.g. L2Ps), which connect through a large ($O(200)$ ports) central switch.

The current baseline for the overall EF network is a set of small independent networks, each connecting a set of EF nodes with a number of SFI's and SFOs. This scheme allows flexibility in choosing the number of EF nodes for each cluster. The input rate capability can be increased by simply adding more SFI nodes to a given sub-farm. The EF network is organized in two layers. The lower layer consists of EF nodes clustered via small switches ($O(20)$ ports). The higher layer, connects together a number of these EF clusters to one or more SFI's and SFOs via a back-end switch.

One possible network architecture implementation and topology, using the switch-based ROS (see Section 5.5.4) as an example, is given in Ref. [5-13].

5.5.6 Supervisory Components

The LVSV and DFM supervisory components are implemented by high-performance rack mounted dual-CPU PCs running Linux, connected to the EB and LVL2 networks. A custom PCI-to-S-LINK interface or standard Gigabit Ethernet technology may be used to connect the L2SV to the RoIB, and standard Gigabit Ethernet technology to connect the L2SV to the rest of the system. The connection between the DFM and the TTC network has not been defined yet — it could be based, for example, on a network technology or on a dedicated PCI-to-TTC-receiver interface.

Measurements done so far show that at the nominal LVL1 rate (100 kHz), only a small number (~ 10) L2SVs are required. One DFM is required for each active concurrent TDAQ partition (see

Section 5.4) — this includes the main data-taking partition. All DFMs except for the one running the main data-taking partition will need to receive detector specific triggers directly or indirectly from the relevant TTC partition. In the main partition, the DFM receives its input from the L2SVs. The maximum foreseen number of partitions in ATLAS is 35. In practise, the number of active concurrent TDAQ partitions which require a DFM and in use at any time is likely to be less than this. A back-pressure mechanism will throttle the detector specific trigger rate via the ROD-busy tree if it becomes unmanageable.

5.5.7 HLT processors

LVL2 and EF processors will be normal rack-mounted PCs running Linux. Dual-CPU systems are the most appropriate solution at the moment, although that may change in the future. We estimate that there will be ~500 LVL2 processors and ~2000 EF processors in the final (100 kHz) system (assuming 8 GHz clock speed), but initial setups during commissioning, and for the initial data-taking when the luminosity will be below the LHC design luminosity, will be much smaller. LVL2 and EF processors are COTS components and can be replaced or added at any time. Computing performance is more important than I/O capacity in the EF nodes.

Although 1U servers currently offer an attractive implementation for the farms, blade servers, which house hundreds of processors in a rack together with local switches, have a number of potential advantages, e.g. lower power requirements, higher density, and may offer a more attractive solution of sufficient maturity by the time the farms are purchased.

Studies are currently underway (see [5-14]) to investigate the possibility and implications of siting some EF sub-farms at locations outside CERN (e.g. at ATLAS institutes or computing centres). This would have the advantage of being able to benefit from existing computing equipment, both during the initial phases of the experiment, when only a fraction of the EF equipment at CERN will have been purchased and throughout the lifetime of the experiment for non-critical applications such as data monitoring.

5.5.8 Event-Building processors

The SFI's are again rack-mounted PCs running Linux. The event building process requires a large CPU capacity to handle the I/O load and the event assembly. Again dual-CPU systems are the most cost-effective solution at the moment. The SFI components require no special hardware apart from a second Gigabit Ethernet interface that connects them to the EF network. Roughly 90 units are envisaged for the final system.

5.5.9 Mass storage

The SFOs write events to the disks in a series of files, and provide buffering if the network connection to the CERN mass storage system is down. Assuming that the SFOs have to buffer ~1 day of event data, with an average event size of 1.5 Mbyte per event at a rate of ~200 Hz, they will need a total of ~35 Tbyte of disk storage. Today, relatively cheap PC servers can be bought with > 3 Tbyte of IDE disk storage. The SFOs will therefore consist of normal PCs, with a housing that allows the addition of large disk arrays — approximately 30 units are assumed for the final system.

5.5.10 Online Farm

The online farm provides diverse functions related to control, monitoring, supervision and information/data services. It will be organized hierarchically in terms of controllers and database servers local to the TDAQ functional blocks (for example the ROS or an EF sub-farm) and clusters of processors providing global functions (experiment control, general information services, central database servers).

The processors for experiment control and general information services will be standard PCs organized in small farms of $O(20)$ PCs each. Local file servers are also standard PCs, with the addition of a large local disk storage (~1 Tbyte) while the central data base servers are large file servers with several TByte of RAID disk storage.

At the time of system initialisation, quasi-simultaneous access to the configuration and conditions databases will be required by many hundreds of processes, in some cases requiring $O(100)$ Mbyte data each. This calls for a high-performance in-memory database system and a structured organisation of the database servers so as to spread the I/O load sufficiently widely that the system can be initialized and configured in a reasonable time. It is proposed to organize the database infrastructure as follows:

- Local data servers: database servers that hold a copy of the conditions and configurations databases to be accessed locally by an homogeneous sub-set of the system (e.g. an EF sub-farm). A copy of specific software and miscellaneous data that is to be downloaded into the components in the sub-set are also held on the local server.
- Central server: a fault tolerant, redundant database server which holds the master copy of the TDAQ software and configuration as well as a copy of the conditions data. Local servers are updated from the central server at appropriate intervals. The central server may both update and be updated by the central offline conditions database.

The control of database write access and synchronization – in particular in the case of conditions data where both online and offline clients may wish to write to the database at the same time – is a complex problem which has not yet been addressed in detail. This issue will be studied in common with the offline community as well as with other experiments with the aim of arriving at a common solution.

5.5.11 Deployment

Functionally homogeneous components, e.g. EF processors, are organized in terms of standard ATLAS sub-farm racks. Each rack contains, in addition to the computing components: a local

file server; a local online switch (to connect all the components to the online central switch); a rack controller; and the necessary power and cooling distribution. We are investigating, with other LHC experiments, cooling systems for racks with horizontal air-flow suitable for rack-mounted PCs. The number of components per rack depends on the size of the racks. Table 5-5 summarizes the organization of the different racks, the number of racks of a specified content per function and their physical location.

Table 5-5 Rack organisation and location

Unit Type	Number of units	Composition	Location
ROS rack	~15	~15 ROSs Local online switch Local file server Rack controller	USA15 – underground
HLT Rack	~20 for Level-2 ~80 for Event Filter	~30 Processing Units Local online switch Local file server Rack controller	SDX 15 – surface 50% of EF racks possibly located in the CERN computer center
SFI rack	3–4	~30 SFIs/rack Local online switch Local file server Rack controller	SDX 15 – surface
SFO rack	6–7	~5 SFO (+ Disk space) Local online switch Local file server Rack controller	SDX 15 – surface
Online rack	3	~20–30 Processors Local online switch Local file server Rack controller	SDX 15 – surface
Central switches	3	~256 ports per switch	SDX 15 – surface

5.6 Scalability and Staging

The performance profile anticipated for the ATLAS TDAQ system between the detector installation and the availability of the LHC nominal luminosity, is summarized in Table 5-6. It is expressed as the required LVL1 rate in kHz.

Table 5-6 TDAQ required performance profile

Phase	Date (TDAQ ready by)	Performance (LVL1 rate)
ATLAS Commissioning	2005	N/A ^a
Cosmic-ray run	2006	N/A ^a
ATLAS start-up	2007	37.5 kHz
Full performance	2008	75 kHz
Final performance	2009	100 kHz

- a. Provide full detector read-out but minimal HLT capability.

The scalability of the TDAQ system is discussed on the basis of the above performance profile. Prior to the ATLAS start-up, the TDAQ system will provide the full detector read-out, for the purpose of commissioning the experiment and the cosmic run, and a minimal HLT system appropriate to fulfil the requirements for this period (few kHz).

The detector readout system must be fully¹ available at the time of the detector commissioning (i.e. in 2005) irrespective of the rate performance, since all parts of the detector must be connected. In contrast, the processing power in the HLT depends on the maximum LVL1 rate that must be accepted. Since the LVL2 and EF farms, as well as SFIs, SFOs, etc, are implemented using standard computer equipment connected by Gigabit Ethernet networks, they can be extended as financial resources become available.

The strategy is therefore the staging of the farms and associated networks. In the latter case, additional central switches (if a topology based on multiple central switches will be chosen) or ports (for monolithic central switches) will be added to support the additional HLT nodes. The same argument, although on a smaller scale, applies to SFIs and SFOs.

The scaling of the TDAQ system size, as a function of the Level-1 trigger rate, is depicted in Figure 5-6 for the Event Builder (number of SFIs and number of EBN ports), Figure 5-7 for the Level-2 sub-system (number of L2Ps and number of L2N ports) and Figure 5-8 for the Event Filter (number of EFPs and ports of the Event Filter network).

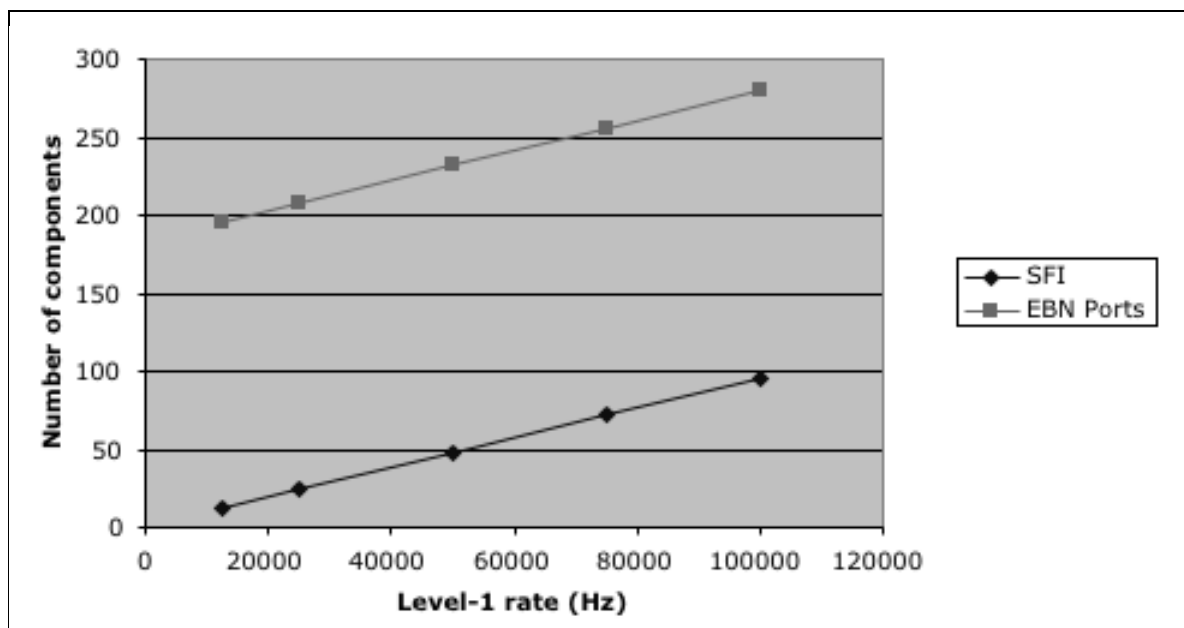


Figure 5-6 Scaling of the Event Builder sub-system

1. Some 25% of the detector ROLs, corresponding to the part of the ATLAS detector which is staged, will not be installed at the start-up of ATLAS but later: currently planned for 2008.

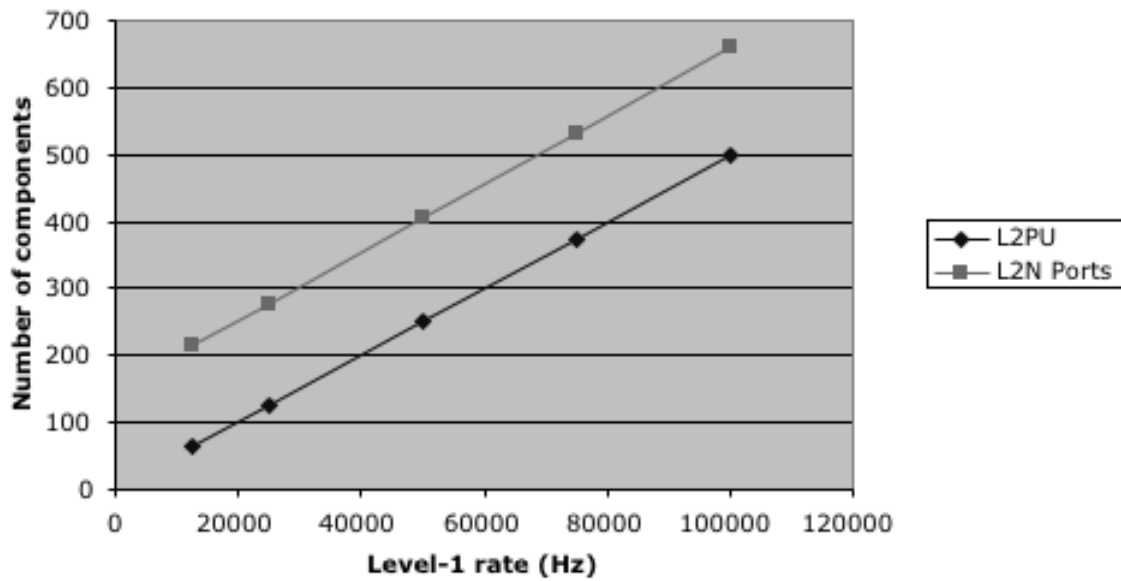


Figure 5-7 Level-2 system scaling

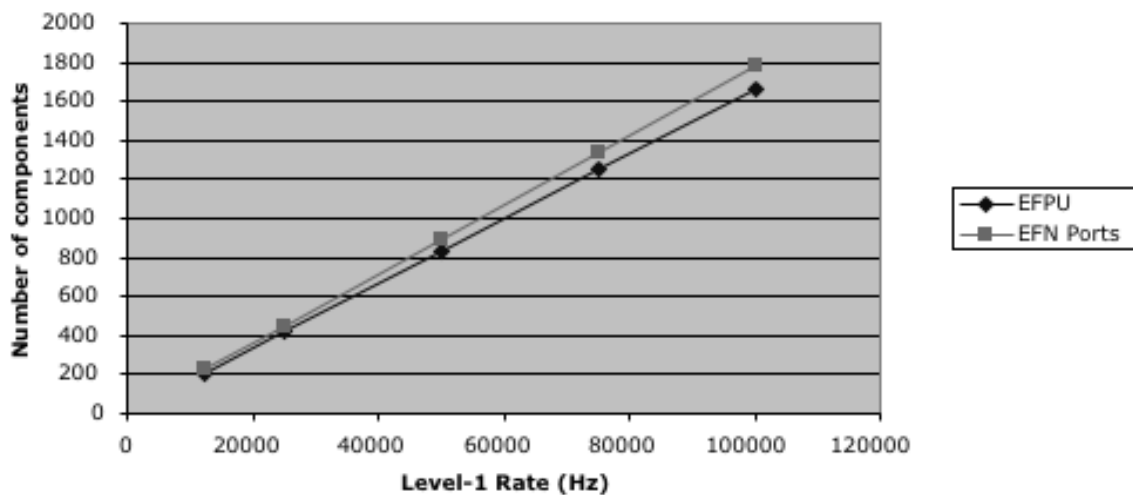


Figure 5-8 Event Filter system scaling

5.7 References

- 5-1 *ATLAS First-Level Trigger Technical Design Report*, CERN/LHCC/98-14 (1998)
- 5-2 *Specification of the LVL1 / LVL2 trigger interface*, ATLAS EDMS Note, ATL-D-ES-0003, <http://edms.cern.ch/document/107485/1/1>
- 5-3 *Proposal for a Local Trigger Processor*, ATLAS EDMS Note, ATL-DA-ES-0033, <http://edms.cern.ch/document/374560/1>
- 5-4 *Recommendations of the Detector Interface Group - ROD Working Group*, ATLAS EDMS Note, ATL-D-ES-0003, <http://edms.cern.ch/document/332389/1>

- 5-5 *The raw event format in the ATLAS Trigger & DAQ*, ATLAS Internal Note, ATL-DAQ-98-129 (1998)
- 5-6 *Event Storage Requirements*, ATLAS EDMS Note, ATL-DQ-ES-0041,
<https://edms.cern.ch/document/391572/0.4>
- 5-7 ATLAS Collaboration, *ATLAS: Technical proposal for a general-purpose pp experiment at the Large Hadron Collider at CERN*, CERN/LHCC/94-43 (1994)
- 5-8 *ATLAS DAQ, EF, LVL2 and DCS Technical Progress Report*, CERN/LHCC/98-16 (1998)
- 5-9 ATLAS Collaboration, *ATLAS High-Level Triggers, DAQ and DCS: Technical Proposal*, CERN/LHCC/2000-017 (2000)
- 5-10 *Summary of Prototype ROBins*, ATLAS EDMS Note, ATL-DQ-ER-0001,
<http://edms.cern.ch/document/382933/1>
- 5-11 *A Prototype RoI Builder for the Second Level Trigger of ATLAS Implemented in FPGA's*, ATLAS Internal Note, ATL-DAQ-99-016 (1999)
- 5-12 B. Gorini et. al., *ROS Test Report, in preparation*
- 5-13 *ATLAS TDAQ: A Network-based Architecture*, ATLAS EDMS Note, ATL-DQ-EN-0014,
<https://edms.cern.ch/document/391572/0.4>
- 5-14 Remote farms ref.

