

Grid Technologies for High Throughput Data Acquisition

M. Sutter¹, T. Jejkal¹, R. Stotzka¹, V. Hartmann¹, M. Hardt² and H. Gemmeke¹

¹*Forschungszentrum Karlsruhe Institute for Data Processing and Electronics
76344 Eggenstein–Leopoldshafen, Germany
e-mail: Michael.Sutter@ipe.fzk.de*

²*Forschungszentrum Karlsruhe Steinbuch Centre for Computing
76344 Eggenstein–Leopoldshafen, Germany*

(Received: 1 December 2008; published online: 25 March 2009)

Abstract: The Grid is an upcoming technology, providing access to an unimaginable amount of storage and computation resources. In the final stage of the Grid nearly every computer worldwide offers its free capacities to the Grid, which gives access to the resources as simple as getting electricity from a power outlet. As process data processing has very high resource demands, growing up rapidly from project to project, the Grid offers the capabilities to fulfil the requirements for upcoming process data processing projects. This paper evaluates the already existing remote control software packages to be used in process data processing projects, explains the demands for developing a high throughput data acquisition system with the introduced requirements, shows the advantages of using Grid technologies and proposes an architecture for developing such a system.

Key words: Data Acquisition, Instrumentation, Grid, Computation

I. INTRODUCTION

Process data processing has always had very high demands from computing and storage resources, increasing rapidly with several new projects. Nevertheless, in the near future the requirements of many process data processing projects will even grow further up to a point where data processing and data storage on local workstations will not be feasible any more. Good examples for demanding such extreme resources are the Karlsruher Tritium Neutrino Experiment (KATRIN) [1], the International Thermonuclear Experimental Reactor (ITER) [2] and Ultrasound Computer Tomography (USCT) [3] at Forschungszentrum Karlsruhe, as they all use data acquisition systems with high data rates.

Another problem for process data processing projects is that in some of them, e.g. the Auger project [4], the sensors are deployed in a distributed system in an area of several thousand square km. To handle such distributed systems it is strictly required that all involved sensors can be controlled from one central campus and that the measured data will be stored in a data sink. The storage in a data sink is necessary with regard to the data analysis purposes, allowing access to the data in a unique manner.

Monitoring data about the health of the sensors increases the amount of data rapidly, if used additionally.

The usage of monitoring data is required in regard to automatic control technologies and for the evaluation of the sensors during development and runtime.

To develop a high throughput data acquisition system, the mentioned boundary conditions are extremely important, especially if it is a requirement that the data acquisition and data analysis should be possible within near real-time conditions. To solve the near real-time requirements, it is necessary for the warranties about the possible data throughput to be established and maintained.

One big advantage of the oncoming Grid technologies is that they can fulfil the requirements for process data processing, as the Grid will allow access to an unimaginable amount of computing and storage resources. For example, the EGEE Grid infrastructure offers about 10 PByte of storage resources, 80,000 CPU's and GBit networks for the communication at the moment. Access to all the resources is not the only advantage of using Grid technologies; the integrated security features are another one, prohibiting the unauthorized access to the resources and instruments located within the Grid. Using Grid technologies offers not only advantages – the biggest disadvantage is that the Grid is still under development and scientists willing to use it are forced to learn a lot about the underlying technologies and how to use them.

II. FUNDAMENTALS

This section introduces the existing technologies for data acquisition, as well as the requirements to develop a high throughput data acquisition system for process data processing sensors and already existing remote control software packages to communicate with various types of sensors.

II.1. Data Acquisition Techniques

A data acquisition system is a combination of several technologies to measure data about physical properties of the environment. The physical properties are observed to analyse the data, to increase the knowledge of the properties and how they interact with other properties [5]. For observing a data acquisition system consists of hard- and software components. The hardware components are sensors for measuring the data, e.g. the temperature of a room for a very simple data acquisition system and are assembled with other electronic components and cables. The software components of data acquisition systems are for the communication and controlling of the hardware, like starting and stopping and for retrieving and storing the measured data, as well as the analysis of the data – if it is a requirement of the data acquisition system. The communication with the hardware components normally includes the implementation of a special protocol to access the hardware.

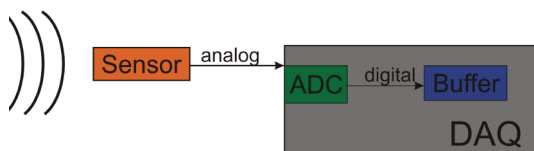


Fig. 1. The schematic design of a data acquisition system. The sensor is measuring the physical properties. The input data is digitalized with an analogue digital converter (ADC) and buffered within the data acquisition system

There are several methods to measure the data with data acquisition systems:

- Continuous measurement systems directly start to measure data after startup. Typical for them is also that they normally need no interaction from an operator, as they run autonomously – even if the measurement takes several weeks up to several months. To prevent the hardware of a continuous measuring system from damage, control technologies are used. As such systems produce data permanently, it might be required to stream the data directly to an appropriate infrastructure.

- Externally triggered systems are typically in a kind of idle mode until an event, mainly initialized by an operator, starts the data acquisition. After the measurement is performed, the system usually switches back to the idle mode and it is possible to retrieve the measured data. The measurement with externally triggered systems are normally very short – only a few seconds up to some minutes, whereas the analysis usually needs much longer, corresponding to the amount and complexity of the data.
- Event-based systems are continuously measuring and analysing the data to observe special kinds of events after startup. As such systems normally have very high data rates, specialized trigger algorithms are used to obtain the events and reduce the data to the interesting parts. The trigger algorithms have a hierarchical structure, whereas the first algorithm (first level trigger) must be very fast. The subsequent algorithms (second level trigger, third level trigger, etc.) do not have such strict time requirements and can evaluate the data more specific. The time for evaluation increases according to the position in the trigger hierarchy, whereas the amount of data decreases with every trigger level. An example of a trigger system can be the following: The first level trigger is used to digitize and analyse the complete data stream. If a possible event candidate is detected, the data is written to a buffer. The buffer is needed, as the first level trigger normally has to analyse the data within only a few milliseconds, which is a time critical process. Another reason for the buffer is that the run should not be stopped for a deeper analysis, done by the following trigger algorithms. To fulfil the time requirements, the first level trigger is normally implemented in hardware, whereas the second and following trigger algorithms can be implemented in hardware or software, depending on the requirements of the system.

II.2. Requirements

The section in front describes the general technologies for measuring data with data acquisition hardware. Nevertheless, for developing a data acquisition system especially for process data processing purposes, several requirements must be fulfilled:

- The analysis of the measured data must be possible with near real-time properties;
- Secure access to the data acquisition hardware, measured data and analysis infrastructure must be

possible to ensure that the measured and analysed data is authentic;

- The controlling of the sensors must be possible from one graphical user interface, even if the hardware is at a remote location like a distributed system. The graphical user interface is called virtual control room;
- The handling of data of any amount and complexity is needed as several projects use more or less similar data acquisition hardware, but have different amounts of measured data and also different data formats for the storage;
- Guaranties of the data throughput capacities must be given, in order to solve the near real-time properties;
- The monitoring of the controlled hardware must be possible including, on the one hand, the usage of control technologies to prevent the system(s) from damage and, on the other hand, the storage of the monitoring data for evaluation purposes.

In the presented list the security and near real-time entries are the primary requirements for building a data acquisition system for process data processing purposes as it is not advisable to operate a distributed remote system with high data rates without both features.

II.3. Remote Control Software

In this section a selection of a huge amount of the already existing remote control software packages is described. The packages are chosen because they are well known in the community and are expandable with drivers for new hardware components needed to integrate the communication with self-developed hardware. The described packages are only an exemplary summary and do not seek to be complete.

- LabVIEW is a development environment for a visual programming language from National Instruments and mainly used for data acquisition, controlling of instruments and industrial automation tasks. The programming language of LabVIEW is called G and is a data flow programming language. Every component (e.g. a sensor) is visualized as a block and has a well-defined functionality. Different blocks are connected via wires in the graphical user interface, the reason why LabVIEW is a graphical workflow programming environment. LabVIEW is a standard and very popular product. Thus most hardware manufacturers deliver LabVIEW drivers for their commercial products.
- The Grid Enabled Remote Instrumentation with Distributed Control and Computation (GRIDCC)

framework [6] is a project funded by the European Commission. The aim of GRIDCC is to provide access to and control of distributed instrumentation components using Grid technologies, as well as access to Grid storage resources and to access Grid computing resources directly from the sensors. To allow the access to Grid resources, every physical sensor is embedded in a so called Instrument Element (IE) [7]. Within the DORII project a new version of the IE is available, making the integration and installation of the various components of the framework very easy.

- TACO [8] is an object-oriented distributed control system originally developed at the European Synchrotron Radiation Facility (ESRF). The aim of TACO is to control accelerators, beamlines and data acquisition systems. All sensors to control are treated as objects in a network space by TACO and implemented in separate classes. The whole network communication from the client to the servers and vice versa is handled by the system libraries of TACO and based on SUN's remote procedure calls.
- TANGO [9] is based on TACO, but CORBA is used for the network communication. The provided object model in TANGO supports methods, attributes and properties. All objects are representations of devices and can be located on a single computer or distributed over the network. To simplify the usage, TANGO provides a kernel API (Application Programming Interface) hiding all the details of network access and also providing object browsing, discovery and security features. Bindings for commonly used commercial software are available, allowing communication from this software with TANGO [9].
- EPICS (Experimental Physics and Industrial Control System) is a set of Open Source software tools, libraries and applications developed in a collaborative manner. It is used to create distributed soft real-time control systems for scientific instruments, e.g. particle accelerators and telescopes [10]. EPICS uses a client/server architecture and publish subscribe methods for the communication between all used components. One kind of computers in EPICS are the Input/Output controllers (IOCs). They collect measurement and control data from the connected instruments in real-time and send it over a network protocol called Channel Access (CA) to the clients. CA supports soft real-time conditions for which Linux or Windows can be used. Hard real-

time conditions are normally provided by RTEMS [11] or VxWorks [12].

- ORCA (Object-oriented Real time Control and Acquisition) is a framework designed for the development of modular data acquisition and analysis systems [13]. Applications are distributed via a client/server model, whereas servers provide the data acquisition and analysis and clients a user interface for access to the servers. Physical hardware and data processing elements are represented via objects in servers and can be dynamically configured by the clients. The communication between clients and servers is implemented using CORBA. The complex details of the communication are hidden in an interface library that exposes only a simple programming interface.

As the security and near real-time properties are important for building a data acquisition system for process data processing purposes Table 1 lists both for the introduced remote control software packages in this section.

III. USE CASES

This section describes the use cases needed to build a data acquisition system especially for process data proc-

essing purposes with different data rates and compares the use cases for suitability with the remote control software packages described in section II.3. Common for all use cases is that it must be possible to control different kinds of instruments remotely from a virtual control room. The instruments can be deployed in two different kinds of locations:

- Deployed on a local site means that the access is only available from inside the site. Access from the outside over the Internet is not possible. The security requirements for accessing the instruments inside a local site are not really hard, as access to the site requires security features.
- Deployed on a remote location means that access to the instruments is possible from the Internet, including strict security requirements to deny unauthorized access and to ensure that the measured data is authentic.

As the first entry has nearly no security requirements and the focus is on world wide distributed systems, the first case is not covered in this paper. The reasons therefore are, on the one hand, that security is a strict requirement for world wide distributed systems and, on the other hand, that a system developed with security is also usable within a site without security.

To develop a remote data acquisition system, not only the control of the instruments is needed, but also another

Table 1. The table list security and real-time properties of the introduced remote control software packages from section II.3

	Security	Real-time properties
LabVIEW	The DSC Module provides security features	LabVIEW offers own real-time modules to handle real-time requirements. Hard real-time is only possible via a real-time operating system
GRIDCC	Grid Security based on X.509 certificates	Communication over the Java Message Service, no evaluation about real-time behavior available for the IE 2 at the moment
TACO	Implemented at device command level	LynxOS and VxWorks ports are available, which are real-time operating systems For the Linux version it is possible to replace the standard scheduler with a Posix real-time scheduler offering some real-time features The Windows/NT version has only limited real-time behavior, so it is not really useable
TANGO	CORBA security model with SSL for the communication	Some real-time device servers exists and are in production
EPICS	Build in security model for accessing hosts and user management	Soft real-time via the operating system Hard real-time via RTEMS or VxWorks
ORCA	Only locking of ORCA dialogs, not really a security model	Offers no real-time behavior

architectural detail related to the storage of the measured data is extremely important. If the instrument does not provide sufficient memory or storage space, or if the data has to be analysed immediately, the recorded data has to be streamed to a second location providing the necessary capabilities. The streaming is not a common feature for the remote control software packages mentioned in section II.3, as they normally use a more common way. The measured data is buffered locally on the instrument and replicated after the measurement.

Especially for streaming of the measured data a streaming provider was developed at IPE. The provider is based on Globus Toolkit 4 (GT 4) security and allows the streaming of data with two security levels. GT 4 is a Grid middleware, allowing the communication with Grid resources via services, making the development of Service Oriented Architectures based on GT 4 installations possible. The built-in security features are based on X.509 certificates, allowing to sign or encrypt the messages sent to the GT 4 servers. The amount of possible data throughput between two computers with the streaming provider was measurements and evaluated in Table 2.

Table 2. The table shows the possible data throughput of the streaming provider between two computers with different security levels and the theoretical amount for 100 MBit and 1 GBit Ethernet

	100 MBit	1 GBit
Theoretical	12.5 MByte/s	125 MByte/s
Insecure	6-8 MByte/s	100 MByte/s
Signed	3-4 MByte/s	40 MByte/s
Encrypted	1 MByte/s	n/a

III.1. Low Data Rates

These kinds of data acquisition systems have no hard requirements. They consist of one or several instruments. Every instrument produces a low data rate of only a few KByte/s. The required network bandwidth for such systems is no problem, as already a 100 MBit Ethernet offers a theoretical data rate of 12.5 MByte/s. A normal desktop computer already offers the capabilities to store data of a few KByte/s. The analysis of such an amount of data is no problem because of the low data rate and the corresponding minimal computing resources. It can be a solution to directly analyse the data within the instrument, as introduced in Figure 2.

An example for such a detector is the MBARI Ocean Observing System (MOOS), consisting of several kinds of detectors with different capabilities, e.g. limited bandwidth

[14]. One kind of such detectors are autonomous underwater vehicles (AUV), a kind of robot travelling underwater and measuring data. During measurement the data is stored on the AUV and replicated after appearing via e.g. radio telemetry or satellite communication, which can have a low bandwidth depending on the location of the AUV or when it travels on the borders of the communication range. Another example are the drifters of MOOS. For measurement the AUVs are more or less autonomous, the controlling includes e.g. the programming of new routes for the measurement.

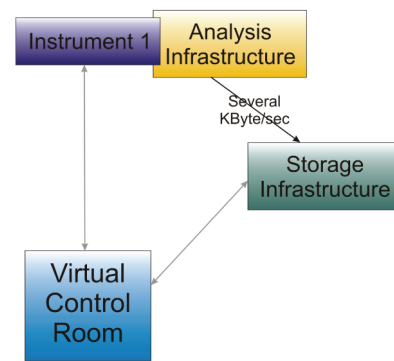


Fig. 2. The schematic design of a data acquisition system with a low data rate of several KByte/s. Visible is the whole infrastructure with virtual control room, instrument, data storage and analysis resource(s). The analysis resources are directly nearby the instruments as it is possible to analyse the data within the instruments

As it is no problem to solve the hardware requirements for such systems, it is also no problem to find suitable software. All the introduced software packages from section II.3 offer the capabilities to control a data acquisition system with such a data rate if it is a wired system. The requirements are low and all packages are capable of handling such an amount of data.

III.2. Medium Data Rates

A data acquisition system with medium data rates specifies sensors that produce data rates of less than 10 MByte/s (see Fig. 3). Modern network components can fulfil the required data rates, e.g. a 1 GBit Ethernet for the data transportation. Furthermore, it is not feasible to store the data on a simple desktop computer, especially if multiple instruments are controlled and must store the data on the same machine. To fulfil the storage requirements, storage resource managers must be used, e.g. dCache [15] or iRODS [16] or StoRM [17] – to name just a few of the

available systems. The analysis infrastructure is another component which must be increased for immediate data analysis to process the data as fast as possible. Service Oriented Architectures offers the needed capabilities for the analysis, especially the adding of resources to solve peak data rates.

The Pierre Auger Observatory in Argentina [4] and KATRIN are examples for such data acquisition systems. The Auger Observatory is studying the universe's highest energy particles which shower down on Earth in the form of cosmic rays. Special for those particles is that they have very high energies of over 10^{20} electron volts (eV) and an estimated arrival rate of just 1 particle per square km and century. The observatory consists of two independent measurement methods, one being the fluorescence detectors (FD). They use an event-based data acquisition technology with several trigger algorithms [18]. Every telescope consists of 440 photomultiplier tubes connected to a front-end-crate. The crate has 22 front-end-boards for the first level trigger included, whereas every board is responsible for 22 pixels of the camera. The digitization is done at 10 MHz with a 12 Bit analogue digital converter (ADC), which is in total a peak data-rate of 9,600 MByte/s for the first level trigger. The data rate is minimized by the subsequent trigger algorithms so that it is applicable to the medium data rates use case.

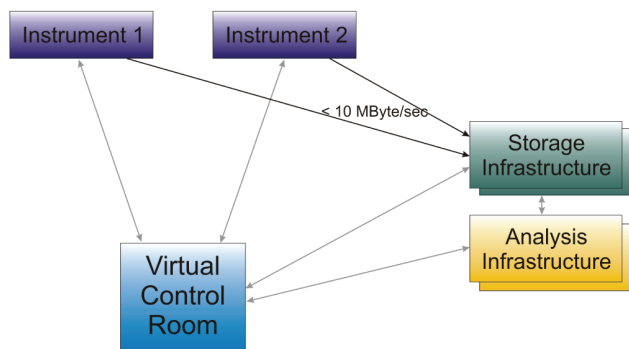


Fig. 3. The schematic design of a data acquisition system with medium data rates of less than 10 MByte/s. Shown is the whole infrastructure with virtual control room, instruments, data storage and analysis resources

KATRIN is designed to measure the mass of the electron neutrino with a sensitivity of 0.2 eV, which scales up the precision of previous experiments by an order of magnitude. The data acquisition, the hardware design and logic are similar to the data acquisition hardware used in the Auger project [1]. It is also event-based and uses a 2-level hardware trigger for the detection of events. The digitization is done at 10 MHz with a 12 Bit analogue

digital converter, whereas the ADC delivers the data directly to a 1,000 samples deep 16 Bit ring buffer. The experiment has 32 different ring buffers per pixel, which is enough for queuing all events until the second level trigger has evaluated the data.

With the increase of the data amount, also the requirements for the control software are higher as such systems need a kind of real-time behaviour. The real-time behaviour for Auger and KATRIN is solved within the custom built hardware to reduce the amount of data, but the data transportation and control technologies must also be applicable. In regard to this, the remote control software packages with real-time conditions (LabVIEW, TACO and EPICS) are usable. If a TANGO system with a real-time device server is used, it should also be no problem. A GRIDCC installation with a powerful server for the IE should also be enough for such kinds of systems, as GRIDCC is mainly for the controlling of the instrument and the notification of the user in error cases, whereas the complete data handling is managed by the IE. The usage of ORCA is not really clear for such systems. It must be strictly evaluated if ORCA is feasible, but it could also be an opportunity.

III.3. Distributed with High Data Rates

Distributed data acquisition systems with high data rates are the supreme discipline for data acquisition systems and at the moment a system for the future. It categorizes acquisition hardware producing data of 100 MByte/s and more and is fully distributed (see Fig. 4). The distribution means that the sensors are deployed “world wide”, e.g. in different countries and organizational domains, and it must be possible to control them from one virtual control room. The high data rates make a complete own infrastructure necessary to use such systems. For communication, extremely fast and dedicated network links have to be provided. They directly connect the instruments with the storage resource managers. To avoid problems with the used networks they have to be strictly evaluated if the behaviour corresponds with the desired one. Therefore [19] offers some possibilities, [20] and [21] introduces solutions for measurement over the Ethernet with real-time features. The used storage resource managers must also be able to handle such amounts of data, especially if several instruments are involved. Also the requirements must be guaranteed for the analysis infrastructure. All used components for such a system must be strictly evaluated if they can solve such requirements and what guarantees for data throughput can be made.

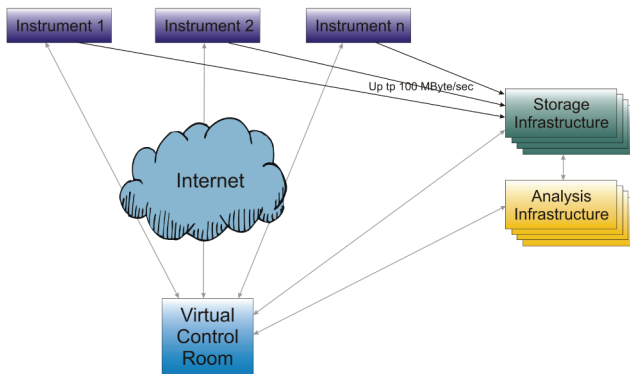


Fig. 4. The schematic design of a distributed data acquisition system with a high data rate of up to 100 MByte/s. Visible is the whole infrastructure with virtual control room, instruments, data storage and analysis resources

The low frequency array (LOFAR) is more or less an example for such a system. It is a new kind of telescope using an array of simple antennas instead of a radio telescope build as a dish antenna [22]. The radio frequencies observed are below 250 MHz, and only the deployment as an array can achieve the sensitivity needed to observe the secrets of the early universe. The measured analogue signals of the antennas for one station are digitized and transported to a central processing unit, where the signals are combined in software to emulate a conventional antenna. As the antennas are very cheap and simple to build, it is possible to deploy a huge software antenna of 350 km in diameter with about 25,000 antennas in the last stage of expansion. As the first step, an array of about 15,000 antennas and a diameter of about 100 km will be deployed. The data acquisition of LOFAR in the first phase consists of a compact core area and 45 remote stations. Every station and the core area is equipped with several independent measuring systems, e.g. 100 high band antennas, 100 low band antennas, 13 three-axis vibration sensors and other components [23]. The data rate of every antenna is about 1.8 GBit/s, for every station about 220 GBit/s as the data is reduced from every station. The bandwidth of the connection of the remote station with the central processing system will be about 10 GBit/s and based on optical fibre communication.

To adapt the remote control software packages, only the ones with hard real-time (LabVIEW, TACO and EPICS) conditions are feasible at first sight. Also, they must be evaluated if the requirements can be fulfilled. The problem with the other software packages is that they must be evaluated in more detail. As the first step, the real-time conditions and possible data throughput of those projects have to be measured. If the possible values are not enough, it must be evaluated if an increase is possible, if one of

those packages should be used. Also the analysis infrastructure for such data amounts must be increased by multiple, otherwise the near real-time processing is not possible.

IV. PROPOSED ARCHITECTURE

To build a remote data acquisition system fulfilling the introduced requirements in section II.2, a decision has to be made if the described software packages are feasible. Otherwise the only solution is to develop a new system from scratch. For the decision of a remote control software package from section II.3, the most important requirements are the security and the near real-time features for replication and analysis of the data and the software packages are evaluated to solve these requirements in the following.

LabVIEW offers security and real-time features in separate modules. The biggest disadvantage of LabVIEW is that it is a commercial product and license fees must be bought for every installation. The advantage thereby is the available support, but making the usage of LabVIEW in scientific projects nearly impossible, especially for systems that consist of several distributed components. ORCA offers neither security nor real-time features in the standard package. The integration of both is too time-consuming and consequently not applicable. Furthermore, ORCA is only available for Macintosh computers. A port to Linux is under construction, but not yet available. The GRIDCC framework has the integrated Grid security, making the system very secure and offering the needed security requirements. The disadvantage is that no real-time requirements are offered, and at the moment no measurement of the available data throughput is available. TACO offers hard and soft real-time properties and also security features. The problem with TACO is that it seems that the project is not really active and tended. Therefore, it is not advisable to develop a system with this package. TANGO based on TACO offers some security and real-time features, but both are not strict enough. The security is based on CORBA with SSL for communication, and in this way not usable for a world wide distributed system, as the features are not strong enough for a remote data acquisition system. EPICS offer the needed real-time requirements as special real-time operating systems can be used. The problem are the provided security features. They are not strict enough for developing the introduced system.

The evaluation point out that only EPICS and GRIDCC offer some of the features with the required strength. EPICS offer hard real-time requirements via the usage of

special real-time operating systems, but missing the security features. GRIDCC offers the security features by using the Grid Security and X.509 certificates, but missing the real-time requirements. In regard to these, the development of a new system from scratch is not advisable, as the adaptation of one of the existing software packages is feasible.

The decision is to use the GRIDCC framework, as it is easier to integrate the real-time features into the extensible structure than to develop the whole security. As it is very difficult to fulfil the requirements for the data throughput, especially for very high data rates, those requirements are solved directly from the involved hardware by using different trigger algorithms, reducing the amount of data. Such specialized hardware is the state of art for sensors producing high data rates at the moment.

By using specialized hardware for the data acquisition and reduction of the data, the remaining problem is how the data is stored on an appropriate infrastructure, which will be solved via the integration of the communication with a storage resource manager in the IE. The communication with a storage resource manager allows the storage of data with medium data rates, on the one hand, and, on the other hand, also access to the data from Grid computing resources for analysing the data.

Another solution for the storage of data is the integration of the streaming provider introduced in section III and the communication with GT 4 based Service Oriented Architectures, allowing the immediate analysis with near real-time properties, if the bandwidth for the data replication and the capacities of the Service Oriented Architecture are high enough.

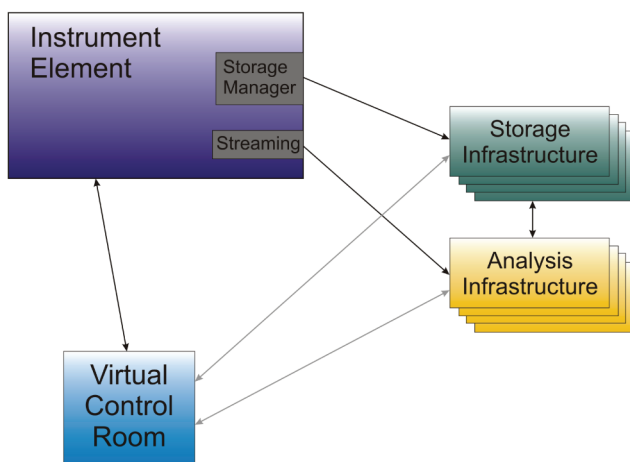


Fig. 5. The schematic design of the IE with integrated communication with a storage resource manager and data streaming in the proposed architecture for a data acquisition system

Figure 5 shows the proposed architecture with integrated communication with a storage resource manager and streaming provider. In a normal case only one of the storage solutions is integrated, as both would overstrain the resources of the instrument and network.

V. DISCUSSION AND FUTURE WORK

As seen, none of the introduced software packages offer all needed capabilities by default. Some offer real-time, but not security features. Providing the needed guaranties for the available data throughput over the network is very difficult and almost none of the provided packages offer such capabilities. Therefore, an out-of-the-box usage of any of the packages is not feasible. For every package different components must be developed or adapted to solve all requirements. One feature which has to be integrated is the data streaming, as no system offers it in the way needed for process data processing purposes.

What about the introduced use cases in section III? How relevant are they, especially for process data processing purposes? The low data rate use case has nearly no relevance for the development of a new system, as it describes an architecture for which nearly all problems, especially the amount of data, have been solved and almost every available product can fulfil the requirements. The medium data rate use case has actual relevance, as our institute is building such systems at the moment – some of them already in production. These systems offer a subset of the described requirements, as they are not needed by the especial projects. Nevertheless, the development of an all-in-one system offering all capabilities is needed to solve the requirements of our upcoming projects. The distributed use case with high data rates is more or less a system for future work, but becoming more and more important as we are building hardware that produces data in the described order of magnitude. At the moment such an amount of data is reduced by several trigger algorithms in the hardware, so that data with no information for the project can be discarded. Nevertheless, with the increasing storage, computational and transportation resources the projects are more and more interested in this data and in the new future they want to be able to use it, which increases the amount of data.

As a summary, only GRIDCC and EPICS of the introduced software packages remain after the evaluation of the requirements, whereas both lack one requirement. Our decision is to use the GRIDCC framework, as we consider it to be the better way to integrate the real-time

requirements than developing the security features. Therefore the GRIDCC installation must be strictly evaluated. What is the amount of data the framework can handle per second? Is it enough or can it be increased by adapting the framework?

To solve the data transportation requirements in the first step, the integration of two different storage possibilities are introduced. The first one is the communication with a storage resource manager, allowing the storage of data with high data rates, also if several distributed instruments are involved. The second solution is the integration of the streaming provider and offers the throughput with not such high data rates as the communication with the streaming provider. The advantage of the streaming is that the data can be analysed immediately and the results are available soon after measurement.

One question still pending belongs to the data throughput requirements. How can they be guaranteed, especially if the sensors are deployed in a remote location where the infrastructure is not as good as e.g. in central Europe?

References

- [1] KATRIN Collaboration, *KATRIN Design Report 2004*. Tech. Rep. FZKA-7090, February 2005.
- [2] *ITER Technical Basis*, ITER EDA Documentation Series, vol. 24, IAEA Vienna 2002.
- [3] N. Ruiter, M. Zapf, R. Stotzka et al., *First Images with a 3D-Prototype for Ultrasound Computer Tomography*. IEEE International Ultrasonics Symposium, 2005.
- [4] J. Abraham, M. Aglietta, I. C. Aguirre et al., *Properties and performance of the prototype instrument for the Pierre Auger Observatory*. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **523 (1-2)**, 50-95, 2004.
- [5] *Wikipedia – The Free Encyclopedia*. (2008, November) Welcome to Wikipedia. [Online]. Available: <http://en.wikipedia.org/>
- [6] R. Pugliese, F. Asnicar, L. D. Cano, L. Chittaro, R. Ranon, L. D. Marco and A. Senerchia, *Collaborative Environments For The GRID: The GRIDCC Multipurpose Collaborative Environment*. Springer US, 2006, ch. IV.
- [7] F. Lelli, E. Frizziero, M. Gulmini, G. Maron, S. Orlando, A. Petrucci and S. Squizzato, *The many faces of the integration of instruments and the grid*. International Journal of Web and Grid Services 2007, **3 (3)**, 239-266 (2007).
- [8] A. Götz, W.-D. Klotz, P. Mäkijärvi, J. Meyer, E. Taurel, and J. Quick, *TACO: An object oriented system for PC's running Linux, Windows/NT, OS-9, LynxOS or VxWorks*. The second workshop on PCs and Particle Accelerator Control, Tsukuba, Japan, 1999.
- [9] J.-M. Chaize, A. Götz, W.-D. Klotz, J. Meyer, M. Perez, and E. Taurel, *TANGO an Object Oriented Control System based on CORBA*. ICALEPCS, 1999.
- [10] R. Lange, A. Johnson, M. R. Kraimer, W. E. Norum and J. Hill, *Epics: Recent developments and future perspectives*. 9th International Conference on Accelerator and Large Experimental Physics Control Systems, p. 278, 2003.
- [11] RTEMS Steering Committee. (2008, September) Welcome to the RTEMS home page. [Online]. Available: <http://www.rtems.com/>
- [12] Wind River Systems. (2008, September) Wind River. [Online]. Available: <http://www.windriver.com/>
- [13] M. A. Howe, G. A. Cox, P. J. Harvey, F. McGirt, K. Rielage, J. F. Wilkerson and J. M. Wouters, *Sudbury Neutrino Observatory Neutral Current Detector Acquisition Software Overview*. IEEE Transactions on Nuclear Science **51 (3)**, 878-883 (2004).
- [14] T. O'Reilly, D. Edgington, D. Davis, R. Henthorn, M. McCann, T. Meese, W. Radochonski, M. Risi, B. Roman and R. Schramm, *Smart network. Infrastructure for the MBARI ocean observing system*. OCEANS, 2001. MTS/IEEE Conference and Exhibition **2**, 1276-1282 (2001).
- [15] M. Ernst, P. Fuhrmann, M. Gasthuber, T. Mkrtychyan and C. Waldman, *dCache, a Distributed Storage Data Caching System*. Proceedings of the International CHEP 2001, Beijing, China, September 2001.
- [16] A. Rajasekar, M. Wan, R. Moore, and W. Schroeder, *A Prototype Rule-based Distributed Data Management System*. HPDC workshop on "Next Generation Distributed Data Management", May 2006.
- [17] F. Donno, A. Ghiselli, L. Magnoni and R. Zappi, *STORM: GRID middleware for disk resource management*. Proceedings of the International CHEP 2004, Interlaken, Switzerland, October 2004.
- [18] Auger Collaboration, *Properties and performance of the prototype instrument for the Pierre Auger Observatory*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **523 (1-2)**, 50-95 (2004).
- [19] G. Giorgi and C. Narduzzi, *Detection of Anomalous Behaviors in Networks From Traffic Measurements*. IEEE Transactions on Instrumentation and Measurement **57 (12)**, 2782-2791 (2008).
- [20] A. Depari, P. Ferrari, A. Flammini, D. Marioli and A. Taroni, *A New Instrument for Real-Time Ethernet Performance Measurement*. IEEE Transactions on Instrumentation and Measurement **57 (1)**, 121-127 (2008).
- [21] P. Daponte, D. Grimaldi and M. Marinov, *Real-Time Measurement and Control of an Industrial System Over a Standard Network: Implementation of a Prototype for Educational Purposes*. IEEE Transactions on Instrumentation and Measurement **51 (5)**, 962-969 (2002).
- [22] J. D. Bregman, *Towards a Concept Design for a LOFAR*. The Universe at Low Radio Frequencies, Proceedings of IAU Symposium 199, 484 (2002).
- [23] S. Achterop, M. Devos, K. V. D. Schaaf and L. Spaanenburg, *Architectural Requirements for a LOFAR Generic Node*. Proceedings of ProRise 2001.



MICHAEL SUTTER was born in Bad-Kreuznach, Germany, on April 20th, 1979. He received the Master of Science in Computer Science from the University of Applied Science in Darmstadt in 2005. Since 2006 he has been working as a scientist at the Institute for Data Processing and Electronics at Forschungszentrum Karlsruhe. His main topic is the development of a remote control software package for the Pierre Auger Observatory in Argentina. His research interests concern the development of remote control technologies, Grid computing and Client/Server systems.



THOMAS JEJKAL was born in Spremberg, Germany, on March 13th, 1980. He studied information technology at the University of Cooperative Education, Karlsruhe from 2001 to 2004. Currently he is employed at the Institute for Data Processing and Electronics of Forschungszentrum Karlsruhe. His interests are medical image processing, Grid computing and database programming.



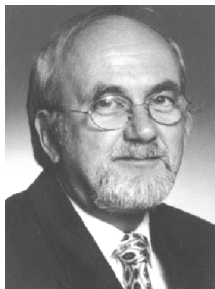
DR. RAINER STOTZKA was born on April 28th, 1962. He studied technical cybernetics in Stuttgart and received his Ph.D. degree from the University of Mannheim, Germany in 1995. After that he joined 1996 the Forschungszentrum Karlsruhe, where he has been the Head of the Department for Software Methods of the Institute for Data Processing and Electronics since 2005. His research interests are computer assisted diagnosis, image understanding systems, pattern recognition, image processing, ultrasound systems, classification, neural networks, 3-D reconstruction, complex data visualization and Client/Server systems.



VOLKER HARTMANN was born in Karlsruhe, Germany, on May 9th, 1967. He received the graduate engineer in electrical engineering from the University of Karlsruhe in 1993. From 1993 to 1995 he made a postgraduate study of computer science at the University of Karlsruhe. Since 1995 he has been working as a scientific engineer at the Institute for Data Processing and Electronics at Forschungszentrum Karlsruhe. His main topics are Grid computing and signal and image processing.



MARCUS HARDT was born in Bonn, Germany, on November 26th, 1974. He graduated as physicist at RWTH in Aachen in 2001 with his thesis on computational analysis of Xray refraction signals. He was working as an IT freelancer between 1999 and 2002. Since 2002 he has been working as a staff member at the department for Integration and Virtualisation of the Steinbuch Centre for Computing (SCC). He participated in several EC-funded Grid computing projects, where he he was managing the software deployment and configuration on the Europe-wide distributed computing infrastructure and was responsible for the projects automated release building infrastructure. Currently he is working on his Ph.D. in the FZK project "Ultrasound Computer Tomography" (USCT) where he is developing diffraction tomography algorithms to compute 3D mammograms from ultrasound signals.



PROF. DR. HARTMUT GEMMEKE was born in Northeim, Germany, on March 9th, 1944. He studied exp. Physics in Goettingen and Heidelberg and received his Ph.D. degree from the University of Heidelberg, Germany in 1971. After two years at the University of Marburg, Germany, he joined 1982 the Forschungszentrum Karlsruhe (FZK), where he was working on the KARMEN neutrino experiment at Rutherford Appleton Laboratory, England. Since 1990, he has been a supernumerary professor at the University of Heidelberg and 1996 at the University of Karlsruhe. Since 1993, he has been head of the Central Department of Electronics and Data Processing at FZK and in 2001, appointed as director to the Institute for Data Processing and Electronics at FZK. He has about 150 publications including 20 in medicine techniques. His present research activities are focused on the investigation of Embedded systems and system on chip, Data-Acquisition-Systems with very high data rates and massive parallel processing, pattern recognition, Neuronal Networks in Hardand Software, Ultra Sound Computer-Tomography for early breast cancer recognition, new functional layers in printed circuit boards, instrumentation of KATRIN neutrino experiment, and detection of air showers by their fluorescence (Auger) and their radio signal (LOPES experiment).