Fast Engineering Archives a new future for mission intelligence

Steven James [(1)]

(1) Logica, Darmstadt 64295, Germany
Email : Steven.james@logica.com

Mission analysts and principal investigators thrive on data. Typically many agencies extract and reprocess telemetry packets. This requires costly reprocessing associated with long delays depending on the volume of data. Having processed engineering data generated by the spacecraft or ground station at your finger tips, can open up new possibilities for mission intelligence. Logica has developed for ESA/ESOC a high performance engineering archive (DARC) that could pave the way for a new breed of mission intelligence applications. The archives ability to handle large volumes of data including statistics opens up the possibility for analysts to perform complex queries quickly at any time.

INTRODUCTION

Flight, Ground station, Simulation and system test engineering data provides a wealth of information on the behaviour of a given device regardless of whether this is space or earth bound asset. ESA's mission control and EGSE system SCOS-2000 currently does not provide a specific means to store calibrated parameter data. While the current packet archive contains this data within stored telemetry packets, obtaining and calibrating the parameters requires intensive reprocessing. This approach is highly inefficient and slow thereby greatly limiting the ability of a mission to undertake speedy analysis or research. For example, MUST (ref MUST) permits retrieval and analysis of data within 5-10 minutes, the equivalent extracting the packet data from SCOS 2000 via TDRS (ref TDRS) would be 2-3 hours. Furthermore, quick access to parameter data can benefit mission operations greatly as was realised by SMART1[2] during its early phases.

To bridge this gap and develop upon the knowledge gained during the development of MUST ESA/ESOC began development of a high performance engineering archive code named DARC (Data Archive). The objective to develop an archive that could support the flight control team and provide the basis to support speedy fast analysis. To ensure the development of a system that could support user needs the system was developed using Agile development techniques in collaboration with a flight control team user group. The core features provided are:-

- Population and retrieval of the static characteristics of TM parameters;
- Population and retrieval of the TM parameter samples;
- Population and retrieval of statistical automatically generated on the samples;
- Replication of the database containing the parameters data (across firewalls)
- Administration of the archive data and configuration.
- Generic system independent of the Mission Control infrastructure

Given the archives capabilities to supply samples at speeds around 20K per second and where readily available statistics data can facilitate trend analysis over decades at performance speeds due to the low number of records, potentially paves the way for new high performance analysis tools. In flight dynamics they sometimes need to regenerate the products using different parameters for analysis purposes. This can talk several hours to reprocess the needed data to perform this. Furthermore, some analysis is performed but using less accurate techniques. Whist this works the decision to perform in this manner is simple down to the time taken to correctly perform compared with known risks. When processed data is quickly and readily accessible, new processes tools will be developed that support working. The result of this is greater value is gained to the organization.

This paper provides a detailed overview of the archive and then examines current developments that would improve near real time data provision and integration.

DATA ARCHIVE OVERVIEW

From the beginning of the project the Data Archive (DARC) requirements and design were focused on general use i.e. to provide a generic storage and management capability for engineering data. To achieve this the system architecture is logically split into two core concepts namely Data provider and Archive engine. These core logical divides are illustrated below.

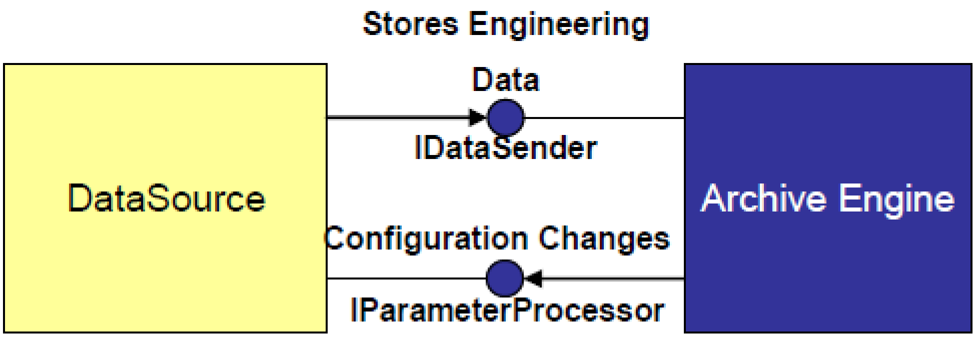

Figure 1 – Archive core concepts

The Archive Engine is generic and provides the following responsibilities:

- Storage of data from data sources
- Provision of data to client applications
- Data administration

The Data Source is a domain aware components which is responsible for:

- Acquisition of data from a given system (EGSE, SCOS-2000) etc
- Provides Processing required to generate a parameter sample
- Initiating storage data into the Archive Engine

As illustrated above the logical areas communicate through standardized interfaces which are language and platform independent. The IDataSender interface is used by the Datasource to store samples of engineering data into the archive. The IParameterProcessor interface is used by the archive engine to inform dynamically the Datasource about changes to configurations that effect the sampling for example which parameters should be sampled or not. In addition, this interface is used to make requests to the Datasource to repopulate the archive, due to gaps, database changes or other refill needs etc should that be required to facilitate the operational concept.

The generic data archive is being used to provide the backbone archive capabilities currently in two distinct areas:-

Parameter Archive: The generic archive DARC concept defined has been validated through its integration with ESA's satellite Mission Control System SCOS-2000[4]. In this context a SCOS addon component was developed for the SCOS-2000 framework which processed reception and generation time telemetry and generated calibrated parameter data for storage.

Ground Segment Archive: In context of the ground stations the data archive was modified slightly to contain a set of new capabilities which would facilitate its use to manage a engineering data collected by the ESTRACK G/S devices. Additionally, these modifications have provided the stepping stone to enable the multi mission capability of the archive i.e. it provided the DARC to manage cleanly manage in a single archive data generated by distinct sources. During the development of this variant the The SystemElement concept was introduced. This makes possible the correlation of data across different Ground Segment Domains (for example, Ground Stations and Mission Control Systems). The SystemElement is a path like string that provides a flexible mechanism to describe a Ground Segment's organization. Typically, the SystemElement will follow a pattern like "Mission/Context/Domain/System/Subsystem". However, this pattern is not mandatory in order to be able to adapt to differing needs.

The generic data archive is composed of a number of subordinate components, which work together as a cohort of Inter-communicating processes to provide the complete capabilities of the archive. In the EGSE environment the archive could be used in the post processing arena. The overall logical model is illustrated below in figure 2 and described later.

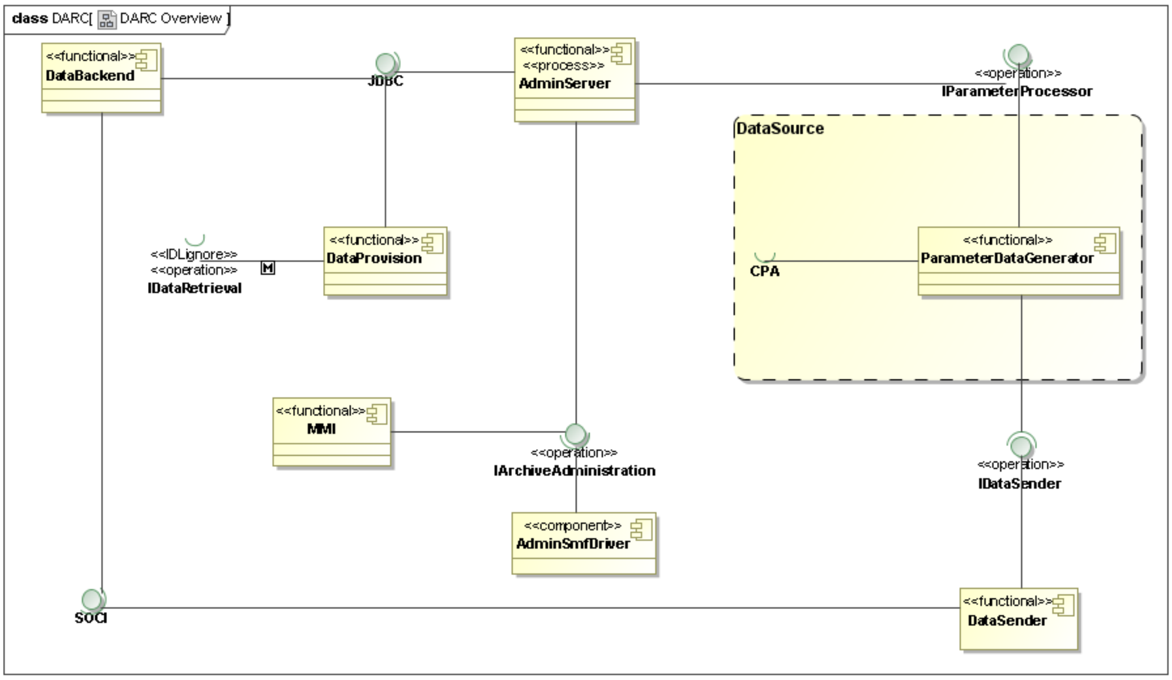

Figure 2 – Logical model

DATA SOURCE (PROVIDER)

The datasource is the provider of sample data information that will be stored into the archive. In the case of the SCOS-2000 the Datasource enables the storage of parameter data received from the SCOS-2000 Mission Control System telemetry sources. It receives telemetry as packets processes and stores the data using the data storage component which is part of the generic archive (Archive Engine). As in SCOS-2000 the data archive could be used equally in the any EGSE system based on SCOS-2000. In the case of a SCOS-2000 based MCS or EGSE (HLCS) system the following sub elements are required.

Data Receiver: the purpose of the Data Receiver is to provide the required telemetry data to the Parameter Data Generator (PDG) for it to perform parameter extraction and calibration of parameter values. This component is a virtual component in the sense that the capabilities required are provided by a number of existing processes available in SCOS-2000. The actual interface to the parameter archive is the Client Packet Access (CPA), which is the S2K API providing access to packet data. This allows a number of data sources to be used to supply the Parameter Archive with telemetry data.

- Packet Data Files (RAPID files): Spacecraft TM data stored in files used to move data between short term and long term Packet Archives.
- Live Packets: Telemetry packets received in near real time during continuous or time duration contact passes.
- Packet Archive: Telemetry requested from the packet archive on demand from the parameter archive, to fill gaps or consolidate the content.

Parameter Data Generator: The parameter data generator is a standalone process which runs as a SCOS-2000 task therefore in SCOS-2000 based MCS or EGSE could be deployed. In a typical SCOS-2000 deployment of the Archive two instances will run as illustrated in figure 3 below.

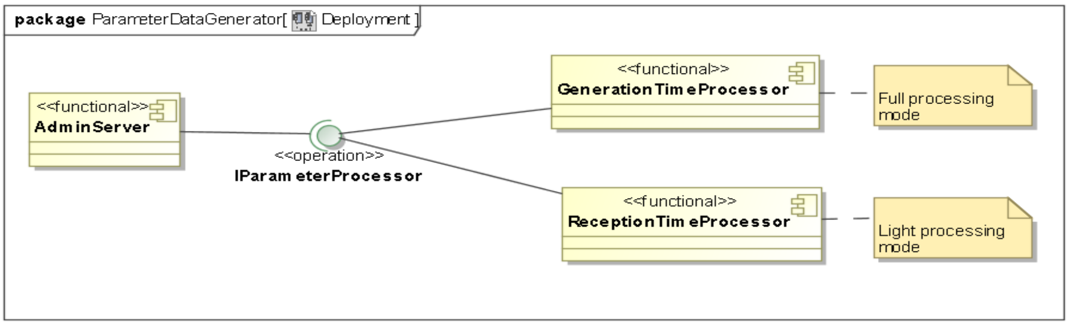

Figure 3 – PDG Component Model

PDG Instances

- Reception time processor which receives telemetry flow from CPD for processing of packets as they arrive.
- Generation time processing used to manage consolidation or repopulation processing which uses the Packet archive as the raw data source.

The core responsibility of this component is the extraction and calibration of parameters from SCOS-2000 packets. Therefore, this is a reuse of a SCOS-2000 specific subsystem, the telemetry model. The main purpose of the component is to process and subsequently store the parameter sample data. Data is being provided " live" (as soon as received) or in retrieval from the packet archive. Regardless of the data source the PDG uses the CPA interfaces of SCOS-2000 to request and receive data. The PDG manages a list of TM parameters, configured by the user, and extracts all samples of those parameters.

When the flight control team (FCT) adds or removes parameters a parameter the PDG process is updated so that it can begin to sample or stop sampling on a given parameter. Users of the archive can dynamically update the parameters being sampled at runtime, this drag and drop capability allows the FCT to perform this quickly and safely (figure 4). The PDG is able to populate samples for on-board generated super-commutated parameters, extracting all parameters samples from a single packet. Within the SCOS-2000 Datasource, the PDG is responsible for consolidation or repopulation requests relayed from the admin component, but this could also be performed by any application that is interfaced with the admin server which is part of the Archive Engine.

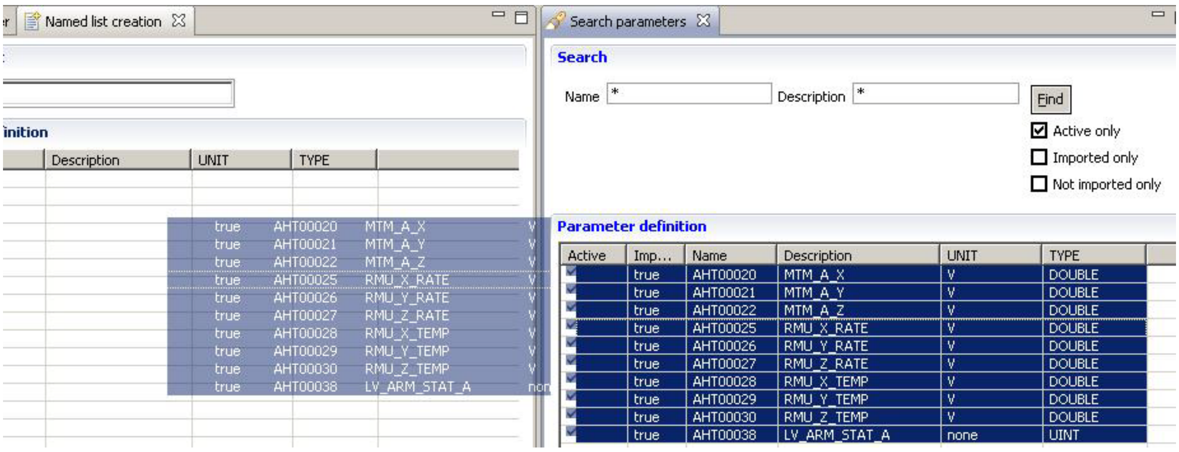

Figure 4 – Parameter drag and drop display

The PDG is able to correctly calculate synthetic parameters validity only when receiving packets ordered by generation time. This is performed by the PDG instance configured to access the packet archive. If it receives packets ordered only in reception time, validity determination is not reliable and therefore is not calculated. To store the parameter data the generator uses the data sender component.

Therefore, the PDG can be configured to run in two modes of operation:

- Full Processing Mode: This mode assumes reception of packets in generation time order. The PDG will always attempt to determine validity of the parameter. This is useful for missions (such as telecommunications) whose telemetry is well ordered and is not usually stored anywhere else within the GS (i.e. validity can be safely determined).
- Light Processing Mode: This mode assumes that the reception of packets is not in generation time order. This mode is faster and applicable when telemetry validity is not easily determined or unreliable at near real time.

ARCHIVE ENGINE

The Archive Engine represents the generic parts of the Parameter Archive that can be reused in multiple domains i.e. simulation, EGSE, flight dynamics. A number of integration points have been designed into the archive to allow for effective integration into existing infrastructure so that harmonization can be simplistically achieved. These design features were realized through effective use of design patterns. The figure 5 below illustrates the logical breakdown and data flows of the Archive Engine.

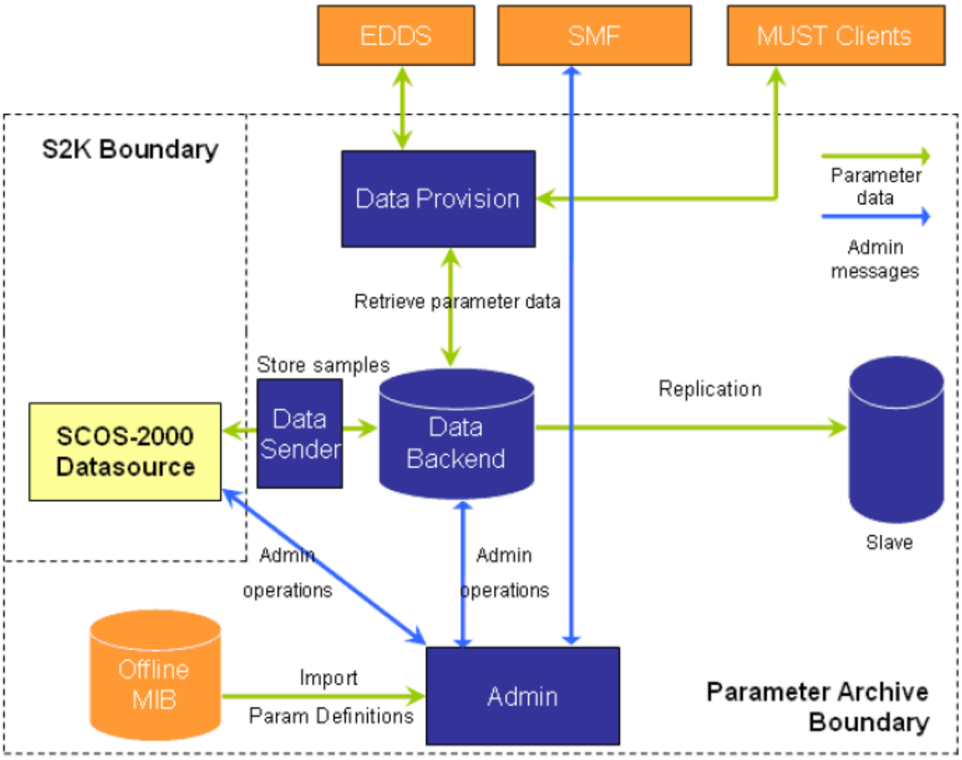

Figure 5 – Archive Engine System Overview

The following components are involved in the processing flow of the data Archive:

Data Backend: This component covers the functionality related to the storage and management of parameter data in a database. The component is based on use of a relational database system. Mainly, the data managed in the database consists of:

- Sample properties: Static information associated with the sample such as its mnemonic, description and source. Source can be considered the device/domain in which the sample value was generated.
- Sample values: Sample data associated with a parameter such as value, state, validity, generation time and storage time.
- Sample statistics: Generated hourly and daily statistical information about samples such as standard deviation, minimum and maximum values, allows extremely fast trend analysis to be conducted.
- Administration data: Information required for automatic operations such as gap detection and consolidation.

This component supports compression and database replication based on the native features of the selected Relational Database Management System. Additionally, data partitioning strategies are used to horizontally split the data in time. These partitions are managed automatically by the Admin component to gain maximum efficiency and disk usage. The benefits of this were considerable improvements in performance response times on large datasets.

Data Sender: This component is provided as a library in both Java and C++ provides data source applications with the capability to inject parameter samples into the Data Backend. This component is a reusable library, providing an abstraction layer from the access API of the database vendor.

Data Provision: This component provides the means for applications to access the stored parameter data (either static or dynamic data). This component is a reusable library available in C++ and Java. It provides an abstraction layer from the database vendor's access API. It represents the façade used by client applications to gain access and subsequently retrieve the archive data, or statistical data. Figure 6 below illustrates the API's being used to provide data in the EGOS User Desktop. Access to the system is restricted to authorized users via user name and password. Two different data delivery forms are supported, namely batch and stream. A number of algorithms were developed for this component to prevent overloading of the system, to achieve this, the time ranges are manipulated within the component to increase performance transparently to the user. To enable this capability database partitioning was used in the Data backend component. Algorithms could be developed to leverage the database data pruning facilities. These facilities minimize the data needed to be read to meet the needs of the query.

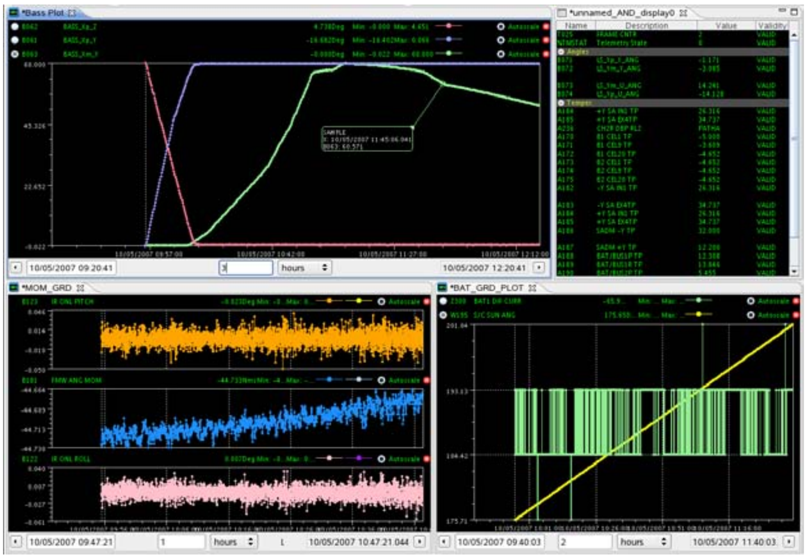

Figure 6 – Plotting using DARC in EGOS User Desktop

Admin: This component provides the means to manage the archive data. A user is able to set up a dataspace area for their campaign data. A dataspace is a virtual container that groups the dynamically generated sample data against a friendly name (e.g. "test data" or "operational data"). This keeps the data generated within different campaigns distinct and relatable. In the MCS environment simulation data is then clearly separated yet the API's can be used the to execute comparisons and other analysis. In the EGSE environment this could be used to store the results of a test run. The dataspaace can then be made available to external principal investigators for analysis.

Multi domain support is provided through use of SystemElements or Archive installations this facilities the archive to support . Each domain therefore has its own database schema instance. This component is split into two processing units and follows a client server architecture,

- Administration server: it is responsible for handling all of the archive administration processing: namely, Dataspace management, Parameter selection, Consolidation / repopulation management and statistics data generation.
- Administration MMI: it provides a user interface to access the administration server. It supports multi domain by identifying automatically (via configuration) archive servers in the naming context.

FUTURE

Analysis is underway looking at leveraging messaging technology to facilitate the supply of data to the archive and other interested parties in real time (In this context real time is defined as "*as they are processed*"). This would certainly in systems such as SCOS-2000 reduce the processing load and facilitate easier integration of additional analysis or automation tools that could act on the data. Currently, demonstrators are being prepared that will enable this in a platform and language independent means. The core architecture is depicted in figure x and is based around platform and language independence.

The solution is based on use of google protocol buffer and ActiveMQ, together these open source products provide the backbone of the solution. These products provide proven high levels of robustness, redundancy and performance. Google protocol buffers has an IDL language similar to CORBA although only supports data structures and not interface definitions i.e. it is not an RPC framework. This is highly beneficial as a clear language independent definition of the required structures can be modelled, reviewed and agreed.

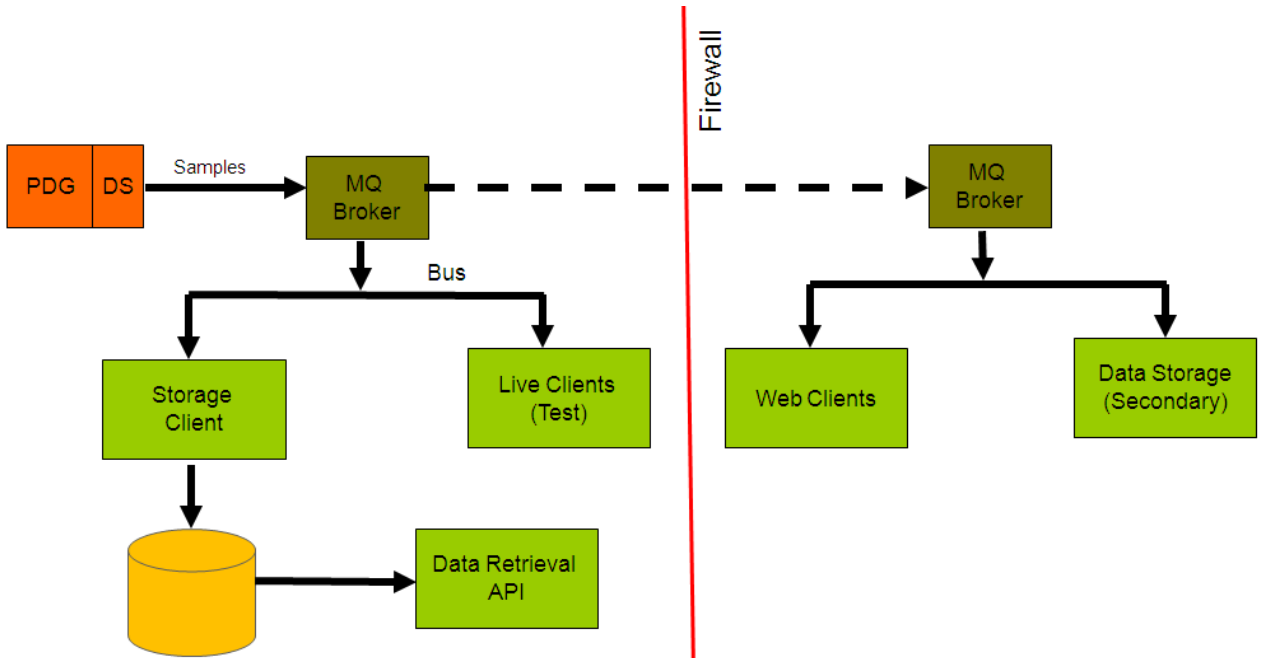

Figure 7 – ESB (Message Bus) based Data Archive

In the figure above the PDG (defined earlier) provides processed parameter sample values directly to the distributor chain (based on ActiveMQ broker). The distributor chain relays the sample values to active listing clients or on to additional brokers to provide a chaining effect. In numerous operational environments principal investigators do not have direct access to operational networks as such we analysis has been made on firewall crossing techniques based on ActiveMQ coupled with Camel. Camel is a framework that implements the Enterprise integration patterns developed by M Fowler et al. On standard PC hardware and transmission of each sample when generated by the PDG performance was in the order of 15k samples per second. To evaluate the source of potential bottlenecks we added an additional PDG process. The two instances were configured to sample different parameters from SCOS-2000 TM packets, the result was circa 30K samples per second. Whilst the analysis is not complete current indications isolate the PDG as the limitation.

This configuration allows new analysis, monitoring processes to be integrated. These processes only require access to the IDL that defines the data structure and have no dependencies on system interfaces or libraries. In the illustrated configuration external principal investigators could get live data feed directly to a web browser, application process and have historical data provided by a data storage system which can be local or remote. This system will help to facilitate new WEB 2.0, rich client, mobile and backend processes that can monitor and analyse mission data.

REFERENCES

[1] Ground systems and operations – Monitoring and control data definition, ECSS-E-70-31, Draft 24, June, 2006, ECSS
[2] A. Martínez-Heras, A. Baumgartner, A. Donati MUST "Mission Utility & Support Tools" DASIA 2005
[3] Ground systems and operations – Monitoring and control data definition, ECSS-E-70-31, Draft 24, June, 2006, ECSS
[4] M. Pecchioli1, E. Gomez1, G. Montroni1, S. Toribio1, C. Clark, "The Design of the SCOS-2000 Multi-Spacecraft Version" SpaceOps 2004
[5] V. Navarro, S. James, "Multi-purpose Engineering Mission Data Archive and Warehousing System", SpaceOps 2010

[6] M. Ferrante, A. Ortenzi, V. Del re, M. Bordin, F. Saccucci, “EGSE For Vega Launcher”, EGSE Symposium 2004