DEVELOPING A CORPORATE INFORMATION SYSTEM ARCHITECTURE: THE CASE OF EUROSTAT

Georges Pongas, François Vernadat
EUROSTAT, European Commission, Bech A2/128, L-2920 Luxemburg

Keywords: Corporate Information Systems, Statistical Information Systems, Information Systems Integration, IT Architecture, Information System Interoperability

Abstract: The paper presents the vision being deployed at the Statistical Office of the European Communities (Eurostat) about a rationalised IT infrastructure for integrated operations of its various statistical production systems. The new architecture being implemented isolates physical data from applications and users, uses database federation mechanisms, strongly relies on the use of meta-data about storage systems, application systems and data life cycles, emphasises the use of thematic and support servers and will use a message-oriented middleware as its backbone for data exchange. Portal technology will provide the unique gateway both for internal and external users to have public or restricted access to information produced by over 130 statistical production systems working in the back-office. Architectural principles and solutions are discussed.

1 INTRODUCTION

Eurostat is the Statistical Office of the European Communities. Its mission is to provide the European Union with a high-quality statistical information service. It also co-ordinates the European Statistical System (ESS), a network of National Statistical Institutes (NSI) of all EU member or candidate states.

Eurostat currently runs and manages about 130 software systems and applications. These systems act both on data, meta-data and nomenclature items originating from the Member States, the candidate Member States and some other organisations (e.g. OECD or IMF). The ultimate goal of Eurostat is to collect, assemble, manipulate and disseminate these data so that they get added-value for a wide range of end-users (e.g. other DG, European Parliament, ECB, government agencies, banks, enterprises, press agencies, politicians, researchers and citizens).

A major problem is the independent and unstructured evolution of these systems and applications, and the many ways data are dealt with. This results in high maintenance costs and on-going ad hoc developments and practices with a risk of system "fossilisation", i.e. rigid, isolated legacy systems. Another problem is the lack of synergy among these systems due to poor interoperability, no real integration and weak data mutualisation.

The paper presents the corporate IT solution adopted to solve these problems at Eurostat in the form of a new information system architecture (Pongas and Vernadat, 2002). Ideas presented can easily be adapted to other similar intensive data processing environments.

2 EUROSTAT BACKGROUND

The existing architecture of Eurostat, similar to that of any other large statistical institute, consists of four environments as summarised by Fig. 1:

- Data collection environment: dealing with receiving data from Eurostat's suppliers. Data are collected by means of individual, asynchronous (e.g. mail, e-mail) or preferably automatic (e.g. STATEL/STADIUM tools) data transfer mechanisms. Furthermore, data are received in a variety of formats and are stored on disk after a first input validation step.
- Production environment: made of a large set of heterogeneous tools, statistical applications, data stores, procedures, meta-data storage systems, etc. used by different people in

Pongas G. and Vernadat F. (2004).
DEVELOPING A CORPORATE INFORMATION SYSTEM ARCHITECTURE: THE CASE OF EUROSTAT.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 21-29
DOI: 10.5220/0002670900210029

different units of Eurostat. In this environment, validated data coming from the data collection environment are manipulated in various ways (data validation, seasonal adjustment, imputation, masking, analysis…) to produce EU aggregates (e.g. EU-12, EU-15, Euro-indicators…) for the reference and dissemination environments.

- Reference environment: containing the data and meta-data that can be browsed by external users of Eurostat. Currently, it mainly consists of two distinct systems, i.e. NewCronos (containing many multi-dimensional data tables on various domains) and Comext (for external commerce).
- Dissemination environment: dealing with the data collections and products to be made available (free of charge or with a fee) to external Eurostat users or to the public (via the web site, the EC Publication Office, etc.). Both NewCronos and Comext have dedicated dissemination facility.

Information systems in the production environment are mostly developed on software systems such as Oracle DBMS, Oracle Express, Fame, SAS and in some cases Access and Excel. They are programmed in C, C++, PL/SQL, Java, the languages imbedded in Fame, SAS, Oracle Express or Visual Basic, but code in APL, Pascal, Fortan or Perl can also be found.

The main stream of processing (or CVD process, for Data Life Cycle) in statistical production systems is depicted by Fig. 2. The flow is essentially linear (iterations may happen with NSI's in the first two phases). Figure 2 outlines the growing importance of statistical meta-data production.

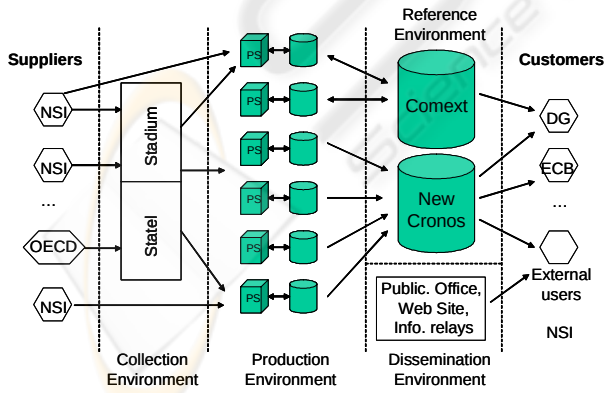


Figure 1: Current situation at Eurostat

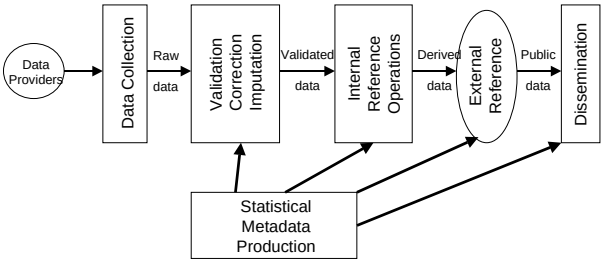


Figure 2. Main steps of data processing at Eurostat

3 ARCHITECTURAL PRINCIPLES

Before detailing components of the architecture, key principles underlying its design are stated to indicate requirements addressed.

- Service orientation: It is the idea of the architecture to commonly offer predefined core business functionalities as remote services, when appropriate, rather than hard coding them in several application systems. These services will be implemented on dedicated functional servers and invoked in a client-server mode, possibly over the web (as Web Services) (Freemantle *et al.*, 2002; Stal, 2002).
- These core functionalities offer a business framework to Eurostat applications in the production environment, i.e. applications call the core functionalities on request. Provided that these functionalities are properly designed, reuse can be deployed to other applications.
- To implement the core functionalities, the optimal software solution is used. For instance, if it is found out that, let us say, SAS is the best tool to implement a particular functionality, it would make no sense to use another solution. Implementing these functionalities with the most appropriate tools and using a generic accessing interface will enhance reuse of these basic building blocks and will avoid duplicated implementations.
- Transparency of Data Storage: This can be achieved by means of a Meta-Data Server containing IT meta-data (ISO/IEC, 1994), not to be confused with statistical meta-data (GESMES, 2002; UN/UNECE, 2000), before data are physically accessed where they are stored using the appropriate data connectivity drivers (i.e. data language and connectivity method).
- All navigation and the construction of data queries are based on the data objects available in

a meta-data database. By relating the data objects to keywords, a keyword-based search mechanism is also made possible. As a consequence, the user does not query the physical data directly and does not see where and how the data are physically stored, but s/he works on a virtual data space tailored to his/her needs (virtual datasets).

- The meta-data database must contain information on the available objects, their physical location, the dimensions, nomenclatures, keywords and other "statistics typed meta-data", the mapping between virtual objects and their components, etc.
- Data storage must not only be transparent to the users but also to the business applications. The implementation of an operation must be independent on the way and the place the data is stored. The fact that the data is stored in a particular way (e.g. a SAS dataset) may not imply that all related analyses or manipulations have to be done that way (e.g. through SAS). This necessitates a generic, tool-independent data exchange format among applications, which are written in a way that enables them to read the meta-database information concerning the methods and possibly location of data, to request execution of remote services and to receive the associated results.

- Transparency of Functionality: Transparency of application functionality can be achieved in a similar way than transparency of data storage, i.e. by means of a Meta-Data Server supporting an application connectivity driver that gives access to implemented functionality. The meta-data database must also store information about the available operations and the way to use them (e.g. parameters, data structure constraints, outputs...).
- No physical but a logical separation of the four environments: a separate data store for each of the four environments is not necessarily mandatory. The logical separation between the data collection and the production environments, the internal reference (REFIN)[1], the external reference (REFEX)[2] and the dissemination environment can be achieved through IT meta-data information. This implies that the meta-data model must be rich enough to incorporate this information.

  Although we do not see any strong need for this, a physical separation of the data belonging to the different environments is however feasible. In this case, the internal design of the physical storage system may be slightly different (data warehousing approach optimised in terms of data security, data access, search efficiency…).
- The Data Life Cycle stages (defining the so-called CVD macro-process) are part of a complex workflow procedure (Pongas and Vernadat, 2003). At any moment, the authorised user must be able to see the Data Life Cycle stage of a particular data set (e.g. state, process steps already applied, etc.). These CVD stages have also important functional side-effects:
- They are used to decide whether a particular operation is allowed on the data set (for instance, no manipulation allowed on data that is not yet validated).
- They relate to the location of the data in the four environments.
- Data Confidentiality: This is also considered as a meta-data aspect because it is indeed information about data. There are many ways to design this in the meta-data model (Pongas and Vernadat, 2003). This could range from a "flag" approach towards a more sophisticated approach in which several levels of confidentiality are defined.
- e-Access will be provided by the use of an enterprise portal (Wilkinson *et al*., 2000). Enterprise portals allow both casual users and profile-based web access to a dissemination environment. It is then possible to provide the user with the right content, possibly aggregated from different sources, both from the user's and/or website owner's perspectives. Portal features can be used to offer services other than simply dissemination services. For instance, if Eurostat functional services are implemented as web services, they could be invoked by external parties (possibly with a fee). Conversely, Eurostat applications could make use of services provided on the web by an external service provider. This is a promising solution to enable future operations of the ESS (European Statistical System).

4 ARCHITECTURAL SOLUTION

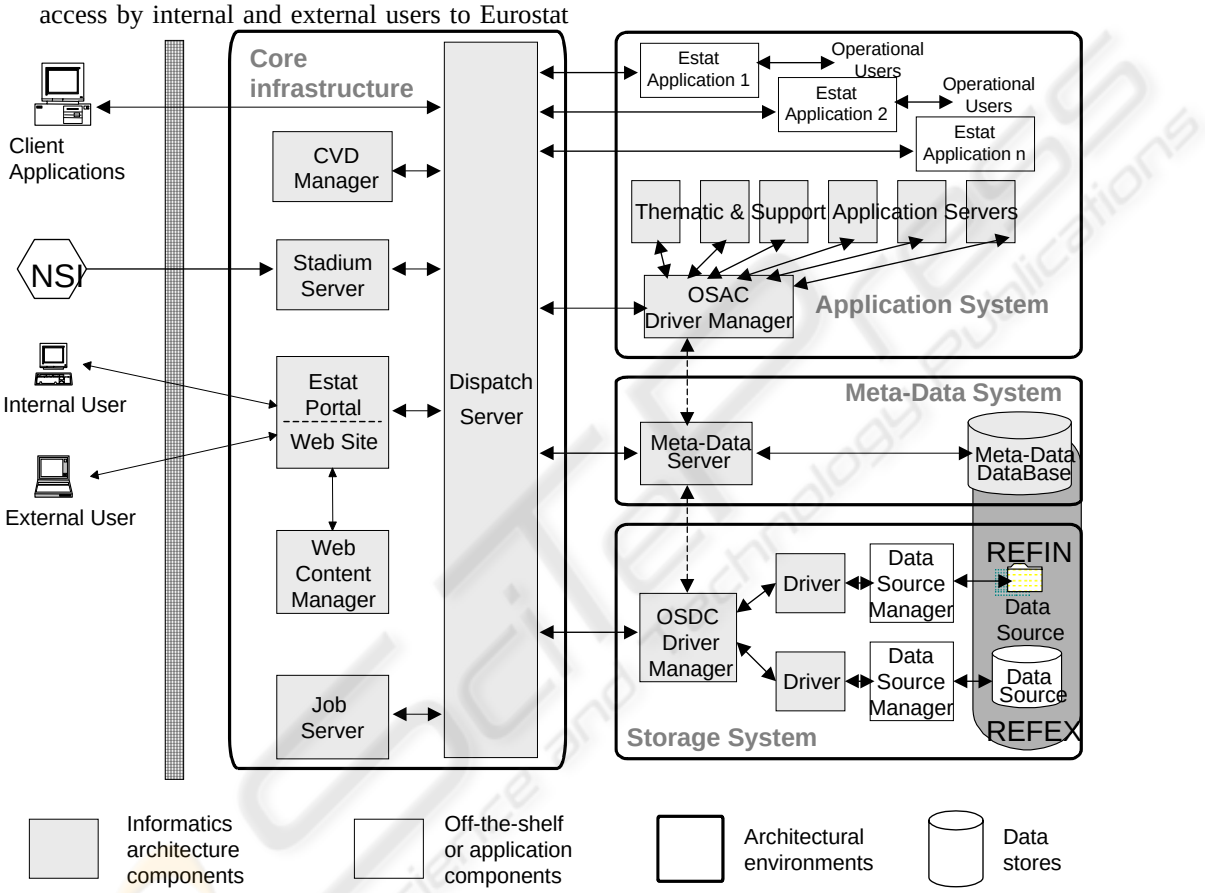
Four interoperable environments have been identified as key modules of the new IS architecture

[1] REFIN contains all valid micro- and macro-data (validated and/or confidential) placed under the responsibility of Eurostat.

[2] REFEX contains all data that can be made public (it is a subset of REFIN).

being implemented at Eurostat to cover the needs listed in the previous section (Fig. 3):

- Core Infrastructure, made of a Dispatch Server, or message broker middleware together with its Job Server, the STADIUM server for input data transfer from remote data providers, a CVD Manager, responsible for managing the life cycle of datasets and possibly of meta-data and nomenclatures, the Eurostat Portal, together with its Web Content Manager, allowing web-based access by internal and external users to Eurostat data and services.
- Application System, responsible for the applicative part of the architecture (i.e. statistical applications and functional/support servers).
- Meta-Data System, responsible for storage and management of meta-data about data stores, applications, datasets and data flows.
- Storage System, responsible for any kind of data storage of operational statistical data.



OSAC: Open Statistical Application Connectivity
OSDC: Open Statistical Data Connectivity

Figure 3: New architecture for Eurostat's information systems

Core Infrastructure

The Core Infrastructure provides all the services needed for communications, both in terms of internal data and message exchange among components of the architecture as well as data and message exchange with the outside world. It comprises:

- The Dispatch Server and its Job Server: The Dispatch Server provides the backbone of the architecture with 'broker' and message-oriented middleware functionality. All messages among components of the architecture (requests, replies or datasets) must transit through the Dispatch Server. It may be considered as an integrating infrastructure and a mediator between client and server components (e.g. thematic and support servers, data

and meta-data servers). It receives requests from any subsystem and knows how to delegate these requests to the appropriate subsystem for execution; after execution, the results might be sent back to the requestor. The Dispatch Server also offers capabilities of a workflow system to implement Eurostat Applications. It relies on the Meta-Data Server for its operations. The associated Job Server acts as an extension of the Dispatch Server for queuing, scheduling, handling, managing and monitoring batch, bulk data transfer or print requests. As opposed to on-line requests, batch or print requests may be executed when system resources are available (in order to use them efficiently) or at a user-defined moment in time (scheduled actions).

- The STADIUM Server: This module represents the encapsulation of an existing environment made by the Stadium and Statel tools used to receive data sets from data suppliers, record information about data issuers and recipients in the EDIFLOW database, put the datasets in the relevant directories and notify operational users about the availability of data.

- The CVD Manager: Its role is to deal with handling data related to the Data Life Cycle (CVD). It records information in the Meta-Data Database (via the Meta-Data Server) about the status of the Eurostat Applications, datasets or statistical meta-data. The CVD Manager mostly receives its information from the Stadium Server and the Dispatch Server assisted by the OSAC and OSDC Driver Managers.

- The Eurostat Portal: This is a mediating facility between external users (using a web interface and looking for Eurostat's public or payable data, products or services) and the REFEX via the Dispatch Server. It receives requests from a client (usually a web browser) and forwards these requests to the Dispatch Server for further processing by the relevant back-office service. It has to combine reliability, performance and security with the ability to deliver dynamic, personalised content, in an easy-to-manage environment enabling worry-free deployment of web-sites. Personalisation of services offered to customers (or profiling) can be envisaged. Internal users can use the portal as well to access Eurostat's data, services or applications.

Application System

The Application System environment is made of Eurostat Applications, i.e. statistical production systems, and a number of functional servers, called Thematic and Support Application Servers. Access to these servers is provided via the OSAC (Open Statistical Application Connectivity) Driver Manager.

- Eurostat Applications: These are the data processing and manipulation systems in use in the various production units of Eurostat to produce statistics (e.g. Acier, LFS, Eurofarm, FISH, Forest, PPP, TRAINS…). They are controlled by operational users in their day-to-day work. In most cases, they have been developed and installed by external contractors according to Eurostat's requirements. The trend is to develop these applications as workflow systems calling services of the thematic servers.

- Thematic and Support Servers: It is planned to divide the business logic of applications over a number of Thematic Servers, each one performing well-defined tasks. It is the idea to organise functionalities used at Eurostat into a set of Thematic Servers based on the Data Life Cycle activities (e.g. data reception, data validation, estimation, seasonal adjustment, reporting…). For instance, all operations/services dealing with the validation of received data will be put on a Validation Server. Similarly, all algorithms/methods for seasonal adjustment will be placed on a Seasonal Adjustment Server. It may be clear that each Thematic Server has to offer a clearly defined interface to other components of the architecture, and especially to the Dispatch Server which knows operations only by their name, logical name of the application server and list of parameters. In fact, it is the role of the OSAC Driver Manager to hide specific features of application servers to the rest of the architecture. The functional operations will be implemented as remote services accessed in a client-server mode and will be functionally grouped by Thematic & Support Application Servers. The servers used for their implementation may be based on specific platforms such as Oracle DBMS, Fame software, SAS software, etc. Support Servers are servers hosting commercial tools (e.g. Business Objects, Matlab…).

- OSAC Driver Manager: The aim of the OSAC Driver Manager is to offer a straightforward organisation of all predefined functional operations or support services available at Eurostat and to offer an easy-to-use and uniform navigation mechanism to locate these functional operations (by means of

meta-information accessible via the Meta-Data Server). The aim of the OSAC Driver Manager is to route requests for executing a given functional operation to the right application server with the appropriate format. This consists in:

- Receiving a request from the Dispatch Server to execute a given functional operation with relevant parameters (input data set, execution parameters).
- Locating the proper (Thematic or Support) Application Server and constructing business request(s) with the help of the Meta-Data Server to be sent to the services of the server via the appropriate Driver (this latter driver is not necessarily needed).
- Executing the relevant services.
- Routing the business results back to the Dispatch Server to be delivered to the right client.

Storage System

The Storage System environment provides the facilities for storing, managing and querying all kinds of data used in statistics (numeric data, textual data, geographical data, codes and nomenclatures, flags and footnotes, validation and confidentiality rules).

- Data Source Managers: They enable the communication between the architecture (OSDC Driver Manager) with their corresponding Data Sources (via ODBC/JDBC, APIs…). The Data Sources represent the native, physical data storage units. They are accessed and managed via their respective Data Source Manager. This could be done through a wide range of storage systems, such as Oracle DBMS, Oracle Express, Fame databases, SAS datasets and even file systems. All Data Sources have their own particularities (e.g. ways to access data). These particularities are handled by their Data Source Manager, so that new Data Sources or even new Data Source types may be added without having to change the OSDC Driver Manager. Each source is described in terms of meta-data in the IS Meta-Data Database (location, access methods, keys to be used...).

- Drivers: These are conversion mechanisms, one for each type of Data Source Managers officially approved at Eurostat (currently, Oracle DBMS, Oracle Express, SAS and Fame). They translate requests from a neutral format (used in the architecture) to the specific language used by the Data Source Manager and vice versa. Neutral format includes Gesmes messages (GESMES, 2002). The future format will have a SOAP/XML flavour (SOAP, 2002).

- OSDC Driver Manager: The Storage System interacts with the Dispatch Server with the help of an Open Statistical Data Connectivity (OSDC) Driver Manager. The aim of the OSCD Driver Manager is to provide a centralised and unified access point to deal with heterogeneous and distributed data servers. More specifically, its role is: (1) to route or break down queries from the Dispatch Server to the right Driver(s) for the relevant Data Source Manager(s) to be accessed, and (2), the other way round, to pass data sets from the target data sources to the Dispatch Server, which will then send them back to the requesting application.

Figure 4 provides a more detailed view of such a heterogeneous distributed database federation mechanism following principles advocated in (Parent and Spaccapietra, 2000; Sheth and Larson, 1990). This approach is scalable (new drivers can be added and duplicated to face addition of more data stores) and allows access to flat files with specific data formats such as XML or SAS files (in this latter case, a dedicated driver needs to be provided for each specific format).

The OSDC Driver Manager receives a generic data request from the Dispatch Server (global conceptual schema) and has to translate this into one or more physical data requests (depending on the complexity of the query) with the help of information from the Meta-Data Server. A data request could be anything going from a query to an update (this is possible if the mapping between the virtual view and the physical underlined component schemata can be inversed without ambiguity). Each generated physical data request has to be routed to the right Driver (local conceptual schema). It is then routed to the right Data Source Manager (local physical data schema) and executed. After this, the OSCD Driver Manager must convert the results into a consolidated response to be sent back to the requestor (i.e. the Dispatch Server in the name of any calling component of the architecture). A response could be anything varying from a dataset to an error message.

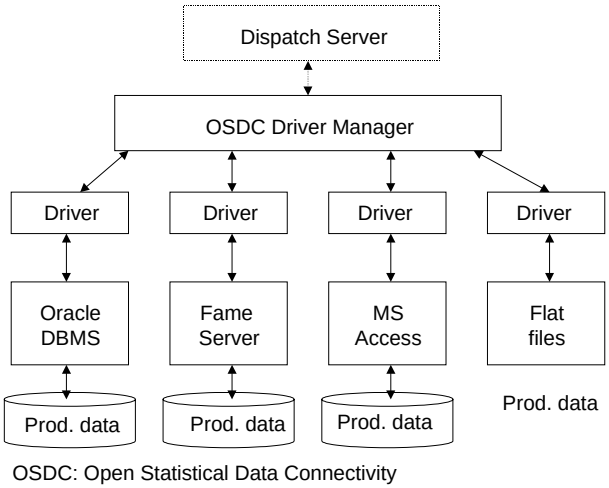


Figure 4: Database federation mechanism in the Storage System

Meta-Data System

The Meta-Data System comprises two elements: a Meta-Data Server and its Meta-Data Database. Its role is to maintain descriptive information about:
- Basic functionalities of Application Servers and to know how to access them (function name, list of parameters, functional server); logic and functionalities of application systems
- Structure, format, physical location, access methods and physical queries of business objects and data stored in the Data Sources; necessary information to form queries
- Data related to the Data Life Cycle (status of data, time stamps, data set versions and variants)
- Meta-data about REFIN/REFEX

- Meta-Data Database: The development of the Meta-Data Database for the storage system at Eurostat has been identified as a complex and long-term task. The major difficulty relies in the necessity to align continuously its content to the reality. This means that adequate API's need to be implemented within existing Eurostat Applications to capture relevant meta-data. This database must contain all meta-data about Eurostat Application statuses and about the statistical data processed and stored at Eurostat.
More specifically, the IT meta-data concerning components of the Application System include:

- Eurostat Applications (triggering conditions, logic of processing, functional operations required)
- Preconditions before a service/operation of an Application Server can be executed (e.g. the required stage of data in the Data Life Cycle)
- Type of the required input data set(s)
- Parameters that are required or optional for the execution of the service/operation (e.g. algorithmic variables)
- Type of the output data set(s)

Concerning the Storage System, the IT meta-data must allow:

- To retrieve data (e.g. the physical location and storage mechanism, the dimensions, the relationships...)
- To find data (e.g. hierarchical or thematic searches, keywords, multi-dimensional searches, full text searches…)

Concerning the CVD, the IT meta-data must allow:

- To monitor the stage of the data in its process from reception towards dissemination, i.e. the status of the data in the Data Life Cycle (CVD)
- To assess the environment in which the data are situated
- To assess the confidentiality of the data: herein, a number of degrees of freedom are allowed: confidentiality may be seen as black-or-white or based on a number of intermediate levels.

- Meta-Data Server: To allow a flexible and simple use of Eurostat's data, all navigation and querying must be based on meta-data. Meta-data give information about the status of operational data and are therefore better suited for a straightforward organisation of all data available at Eurostat. The Meta-Data Server deals with the management of meta-data regarding the Application System, the Storage System, the CVD and the two reference environments (REFIN and REFEX). It encompasses:

- Support to easy and uniform navigation to and querying of data
- The construction of generic data or service requests to be sent to OSAC and OSDC Driver Managers
- Routing the query results in neutral data format back to the Dispatch Server

5 MIGRATION PATH

This project is a long term and strategic project for Eurostat. Its inception and functional design spanned over three years. The implementation phase, which of course did not start from scratch, has been going on for one year so far. It goes step by step and uses

an incremental approach. Completion of the architecture is targeted within the next four years.

Currently, part of the REFIN is in place and REFEX is materialised by a system called NewCronos. Nearly all Drivers for all types of Data Source Managers used at Eurostat (four in total) are available. The Stadium server is operational and is being extended with workflow capabilities to monitor input data flows. Work on the CVD Manager and the Meta-Data Database is on-going (Pongas and Vernadat, 2003). A portal strategy has been defined and re-building the Eurostat web site has started. Definition of the new neutral message format and underlying messaging system will soon be started (Gesmes and XML).

Figure 5 illustrates the planning currently proposed to carry out the project structured into 12 work-packages.

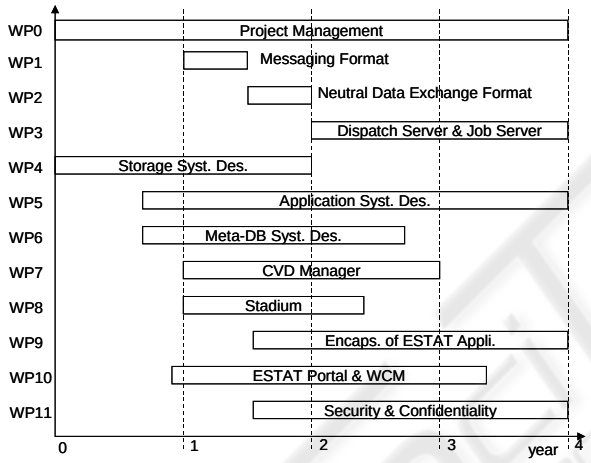


Figure 5: Project planning

6 CONCLUSION

Expected benefits of such a new architecture for an organisation like Eurostat include:

- Modularity: The architecture is made of modular components (services, data stores, messages, data formats…) implemented on servers. This provides a high degree of independence from technology and allows easy modification or replacement of any component with minimum impact on the rest of the architecture.
- Flexible integration: Due to its modularity principle, the architecture caters for high integration of the Application System and Storage System without creating a huge monolithic information system, i.e. a networked system in which production system autonomy is preserved and which does not fall apart when one of its components evolves or changes.
- Transparency of service and data location: Thanks to the use of meta information and normalised access to data in data stores as well as to functional operations on application servers, the users or the applications do not have to know where data and services are located and which access methods are required to use them.
- Scalability: Because of its modular structure, the architecture is scalable both in terms of new services that can be added than in terms of data stores that can be accessed.
- Traceability: Thanks to its CVD Manager and Meta-Data Manager, the architecture can ensure a high level of traceability both regarding the life cycle of data sets processed and flow of processing made in application environments.
- Independence to development languages and systems: The architecture must rely on state-of-the-art standards of the moment and is intended for long term solutions.
- Reduction of production and maintenance costs: Offering services on a shared basis to the many applications will lead to significant reduction of development and maintenance costs of statistical applications (i.e. the same update is made once).
- Major step forward to building a corporate data repository and management: By federating the various data stores used at Eurostat and managing the four environments (collection, production, reference and dissemination) under the same umbrella, the architecture gives Eurostat a way for better exploitation of the whole set of data put under its control and a means to increase its quality of services.

Key success factors identified include:

- realistic planning and design (in terms of timeframe, labour, budget and risks)
- strong and continuous support of top management
- incremental implementation and deployment
- continuity of service at the user level (i.e. current operations should not be stopped or disturbed)
- limited change of user operational habits, unless requested by the users themselves
- good communication and explanation of what's going on (at all levels of the organisation)

REFERENCES

Fremantle, P., Weerawarana, S., Khalaf, R. Enterprise services, *Communications of the ACM*, 45(10): 77-87. 2002.

GESMES Reference Guide, 2002. www.gesmes.org.

ISO/IEC, International Standard IS 11179-3, Information Technology – Specification and standardization of data elements – Part 3: Basic attributes of data elements, 1994.

Parent, C., Spaccapietra, S., Database integration: The key to data interoperability, In *Advances in Object-Oriented Data Modeling* (Papazoglou, M.P., S. Spaccapietra and Z. Tari, eds.), The MIT Press, 2000.

Pongas, G., Vernadat, F. A New Architecture for Eurostat Information Systems, Version 1.1, Unit A1, EC Eurostat, Luxemburg, 27 April 2002.

Pongas, G., Vernadat, F. Data Life Cycle object model for statistical information systems, Proc. Joint ECE/Eurostat/OECD meeting on the management of statistical information systems, Geneva, 17-19 February 2003.

Sheth, A., Larson, J., Federated database systems for managing distributed, heterogeneous, and autonomous databases, *ACM Computing Surveys*, 22(3): 183-236, 1990.

Stal, M. Web services: beyond component-based computing, *Communications of the ACM*, 45(10): 71-76. 2002.

SOAP, Simple Access Object Protocol, 2002, http://www.w3.org/TR/SOAP.

UN/UNECE, Terminology on Statistical Metadata, United Nations, Geneva, Switzerland, 2000, (www.unece.org/stats/publications/53metadatatermino logy.pdf).

Wilkinson, P., Jawa, N., Lange, P.B., Raimer, A. *Enterprise Information Portals – A Cookbook*, IBM RedBooks, 2000.