Financé par ANR

Specification of the NeuroLOG architecture components

Deliverable L3

Authors: **Responsible: INRIA Rennes**
Partners: I3S
LARIA
Business Objects
Visiocopie
INSERM/GIN
INSERM/IFR49
INRIA Sophia

Summary:
The purpose of this technical report is to specify the different interfaces between the different NeuroLOG architecture components, including:

- Interface Specifications with « Data Federator »
- Specifications of the methodology for building the NeuroLOG ontology
- Specification of the Interface for image processing workflow composition and computing resources management
- Specification of the Security Constraints
- Specification of the interfaces to access the computing GRID engines
- Specification of the image processing workflow engine
- Specification of the test-bed applications

Document layout

Item	Value	Remark
1H MR	Proton Magnetic Resonance	
ACL	Access Control List	Technology for controlling individually user accesses to sensitive resources.
AES	Advanced Encryption Standard	
AnaCOM	Anatomo-Clinical Overlapping Maps	A method developed in Pitié Salpétrière to study anatomo-clinical correlations from imaging data
API	Application Programming Interface	
BET/FSL	Brain Extraction Tool	Skull Stripping image processing tools from FSL software library (Oxford)
BFO	Basic Formal Ontology	A foundational ontology proposed by B. Smith et coll. from IFOMIS (Leipzig)
BrainVISA	Brain Visualisation and Statistical Analysis	Brain image analysis and visualisation tool.
CA	Certificate Authority	A recognized authority delivering electronically signed certificates.
CE	Computing Element	EGEE interface to computing resources
CNIL	Commission Nationale de l'Informatique et des Libertés	
COPS	Core Ontology of Programs and Softwares	A core ontology based on DOLCE to address the field of computer programs and software
CORESE	COnceptual REsource Search Engine	Semantic data search engine.
CPS	Carte Professionnel de Santé	
DAWG	RDF Data Access Working Group	A working Group of the W3C
DB	Data Base	
DBMS	Data Base Management System	
DICOM	Digital Imaging and COmmunications in Medicine	Standard for image communication and archiving in medicine
DF	Data Federator	
DN	Distinguished Name	
DOLCE	Descriptive Ontology for Linguistic and Cognitive Engineering	Foundational ontology developed in the context of the WonderWeb EU project: DOLCE has "a cognitive bais"
DSA	Digital Signature Algorithm	
DTI	Diffusion Tensor Imaging	MR acquisition modality.
EGEE	Enabling Grids for E-sciencE	European project. **http://www.eu-egee.org**
EM	Expectation Minimization	Statistical parametric estimation
EU-GridPMA	European Policy Management Authority for Grid Authentication	
FLAIR	Fluid-attenuated inversion recovery	
gLite	Lightweight middleware for grid computing	gLite is operating on the EGEE grid infrastructure. **http://www.glite.org**
GIN	Grenoble Insitut des Neurosciences	
GRID	Computing and Data Grid.	Shared IT infrastructure composed of standard computing units spread over the internet and operating a middleware which aims at hiding the system distributed nature to the users.
GRID-FR	French Certification Authority for grids operated by the CNRS	
GUID	Grid-wide Unique IDentifier	
I&DA	Information and Discourse Acts	A core ontology based on DOLCE, initially

		developed to classify documents based on their contents
IFR	Institut Fédératif de Recherche	
INR	INRIA image file format	
JDBC	Java Database Connectivity	
JDL	Job Description Language	
LFN	Logical File Name	
LOCUS	Local Cooperative Unified Segmentation	An image segmentation tool developed by the GIN (Inserm, Grenoble)
MCI	Mild Cognitive Impairment	a pathology of the brain
METAmorphoses	A software tool to transform relational data into RDF triplets	
MOTEUR	home-Made OpTimisEd scUfl enactoR	Grid-enabled workflow engine. **http://egee1.unice.fr/MOTEUR**
MRI	Magnetic Resonance Imaging	
MS	Multiple sclerosis	A pathology of the brain
MySQL	Standard open-source SQL implementation	**http://www.mysql.org**
NFS	Network File System	
OAR	Grid Resource Allocation System	
OGF	Open Grid Forum	**http://www.ogf.org**
ONTOSPEC	A methodology to specify an ontology using a semi-informal representation	
OS	Operating System	
OWL	Ontology Web Language	Knowledge representation language
RB	Resource Broker	EGEE component
RDF	Resource Description Framework	Knowledge resources representation
RBAC	Role-Based Access Control	
RDFS	Resource Description Framework Schema	
ROI	Region of Interest	
RSA	Rivest-Shamir-Adleman encryption algorithm	
SE	Storage Element	EGEE storage resource interface
SPARQL	Simple Protocol and RDF Query Language	Semantic data query language
SPM	Statistical Parametric Mapping software (FIL, London)	
SQL	Sequential Query Language	Standard DB query language
SSL	Secure Socket Layer	Encrypted communications software layer.
STREM	SpatioTemporal Robust EM	Brain Tissue and Lesion detection tools
UID	Unique IDentifier	
URL	Uniform Resource Locator	
UUID	Universal Unique IDentiifier	
Virage	An MR clinical protocol used in the GIN in Grenoble, to explore patients after an acute stroke using MR	
VO	Virtual Organization	
VOMS	Virtual Organization Management System	
WSDL	Web Service Description Language	
WMS	Workload Management System	
WP	Work Package	

XML	eXtensible Markup Language	

Table 1: Table for definition of Acronyms used in the following document

Objectives of the Deliverable

The purpose of this technical report is to picture the overall software architecture and to specify the interfaces between the different NeuroLOG architecture components. The overall software architecture is first presented and the interface between the different modules is then discussed.

NeuroLOG architecture overview

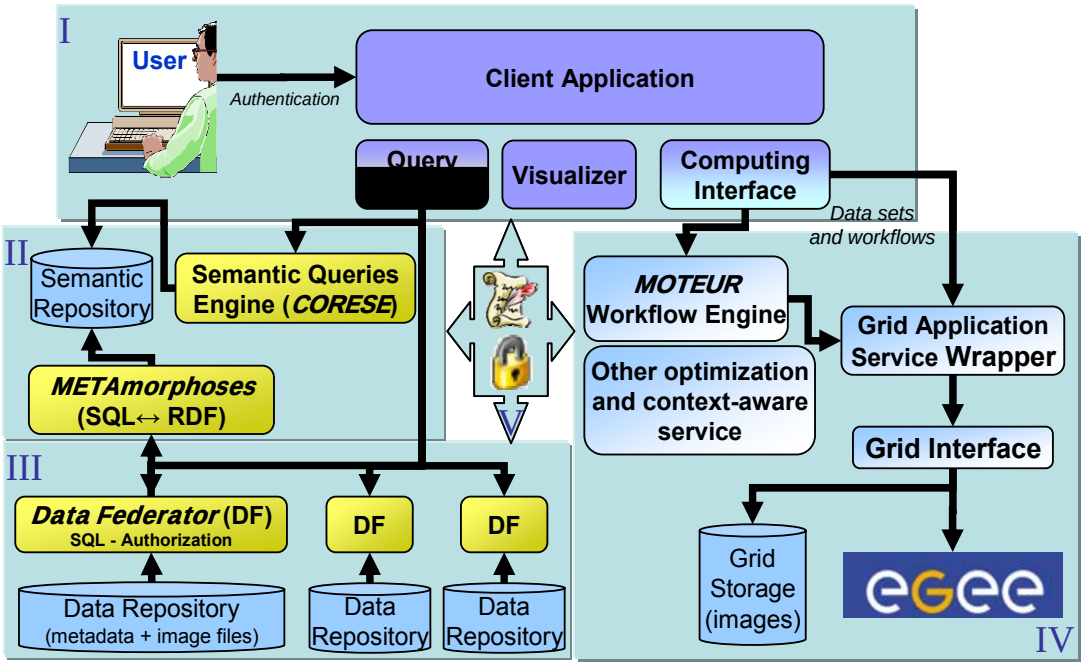

Figure 1: Summary of the NeuroLOG Architecture

As shown in Figure 1, the NeuroLOG architecture is composed of five major blocks of modules. A typical use case of the architecture exploitation is as follows. First, a user concerned by one of the three test-bed neuroimaging applications interacts with the *Application Module* (I) in order to query the NeuroLOG data management system. Secondly, the query is built and processed through a mediation engine performed by the *Semantic Module* (II). This module defines the shared semantic referential used by all NeuroLOG partners to map the different local views to the same NeuroLOG semantic referential. Once the query has been adapted to the shared representation, the *Data Module* (III) performs the transversal search of information through a set of local repositories by using specific adapters to local information. Once the data are retrieved from the semantic query (II) and the transverse search through the wrappers (III), the *Computing Module* (IV) executes image processing workflows using GRID infrastructures. Since we deal with medical data, security constraints apply to all links and components dealing with these modules. The *Security Module* (V) takes this part in charge.

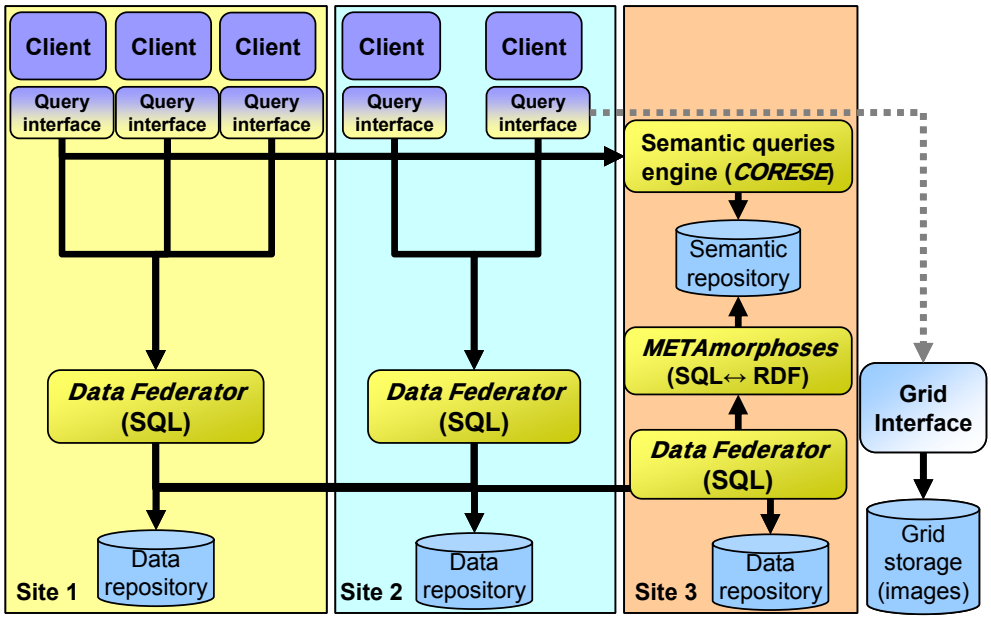

Figure 2: Example of platform **deployment scenario with respects to 3 participating sites**

Document organization

As illustrated in this use case, the infrastructure manipulates both shared medical data and data analysis tools. Different kinds of data distributed over the participating sites will therefore be considered:

1. Files, mostly images, containing the core medical data;
2. Metadata with different origins:
    a. Medical metadata associated to image files,
    b. Image processing tools metadata,
    c. Administrative metadata *e.g.* for security needs;
3. Semantic data, enabling rich queries and retrieval capabilities.

The Semantic Module (II) will extract and structure information using an ontology designed in the context of the project. A semantics repository populated from the sites metadata through this ontology will be deployed. The ontology design methodology is detailed in §1.

The Data Module (III) concerns files and metadata which will be distributed over the different sites. Figure 2 illustrates the envisaged platform deployment scenario (considering 3 participating sites in this example). Each site produces image files and associated metadata that is stored locally into a site data repository. A cross-sites file identification schema is provided through the use of File Unique IDentifiers (UIDs) and association between file location and UIDs. This mechanism extends to files stored externally on grid storage. The metadata is federated between sites through the Data Federator engine. Each site deploys one Data Federator service for the needs of its clients. The interface of Data Federator to the rest of the architecture is described in §2.

The Computing Module (IV) is interfaced to the Data Module as described in section §3 to fetch data to analyze and store results. It is also interface to

grid middlewares to ensure high performance computing (EGEE or Grid'5000 for our concern) as detailed in §5.

The security module (V) specified in §4 is transverse to the others. It implements the security policy designed to match the application-area specific requirements.

Finally, the *Application Module*, specifying the three test-bed applications (Multiple Sclerosis, Strokes and Brain Tumors) is specified in chapter §6.

1. Specifications of the methodology for building the NeuroLOG ontology

Schedule: M6 (task L2.1)

Responsible: INRIA Rennes

Partners: LARIA, INSERM/GIN

1.1. Objectives

The purpose of this chapter is to describe and specify the methodology to build and maintain the application ontology.

The following of this section is composed of four parts:

1. the first part summarizes the motivations for developing an application ontology
2. the second part focuses on the retained basic principles
3. the third part describes the methodology to create the ontology, which addresses 3 major aspects:
    a. definition of the scope (domain of discourse), based on the analysis of the needs arising from the applications
    b. method to express, in a semi-informal way, the entities and relationships involved in the domain of discourse
    c. method to express the ontology in a formal (machine-processable) way
4. the fourth part addresses the quality assurance issue as well as the suitability of the semantic tools (in terms of performance)

1.2. Motivations

The primary goal is to specify a common language throughout the NeuroLOG system, in order to overcome the heterogeneity of data and programs through annotations based on this common language.

This includes:

- Expression of queries concerning neuroimaging data
- Reasoning (in the context of the evaluation of the queries)
- Expression of queries concerning processing tools
- Interoperability of processing tools, especially in the context of their re-use in new data processing pipelines

1.3. Basic principles

1.3.1. Articulate the ontology on a common foundational ontology

To comply to common principles such as the one provided in the DOLCE ontology is essential in order to develop a large and multi-domain ontology and to guarantee that reasoning can be achieved in a uniform, sound and predictable way. This is really essential in order

to achieve a globally consistent and easy to maintain ontology. Besides, we make the assumption that many existing terminology systems in the field of biomedical informatics will evolve toward full-fledged ontologies in the next years – based on one of the existing foundational ontologies, i.e. DOLCE or BFO [Grenon 2003]. So, relying on a foundational ontology such as DOLCE should facilitate the integration of multi-disciplinary data and knowledge in the future (medical imaging, image processing, clinical medicine, anatomy, biology, genomics, etc).

1.3.2. Re-use of existing ontologies, called "*Core Ontologies*"

Re-using existing ontologies is a way to "not re-invent the wheel", and to provide a modular architecture that facilitates maintenance [Temal 2007].

We can provide a preliminary list of the *Core Ontologies* [Gangemi & Borgo, 2004] that we plan to use :

- I&DA, which is a *Core Ontology* in the domain of semiotics, and was initially built to classify documents by their contents. I&DA extend DOLCE by introducing three main concepts: *Inscriptions*, *Expressions* and *Conceptualizations*.
- Participant Roles and Knowledge roles, a core ontology that is relevant to representing how image content is processed.

1.3.3. Development of the new ontologies

The new ontologies to be developed may be either *Core ontologies* or *Domain ontologies*. This modularity is essential to facilitate management and maintenance.

For example, COPS, a *Core Ontology* of the domain of programs and software, encompasses main concepts and relations for this domain. This core ontology will be used in a second step to conceptualize a sub-domain of computer programs, namely that of image processing tools.

We also plan to develop a *Domain Ontology* for the brain anatomical aspects related to the specific needs of our applications, based on the Foundational Model of Anatomy and taking into account articulation with DOLCE.

1.4. Methodology for the creation of the ontology

1.4.1. Definition of the scope

Entities and relationships: The goal is first to determine what entities and relationships should be included in our application ontology.

Two major aspects have to be considered, based on the analysis of the needs concerning our three applications (multiple sclerosis, strokes, tumours):

- Neuroimaging data and related metadata
- Processing tools

The work to be done consists in listing the entities to be managed:

- Acquired neuroimaging data:
  - MRI
    - T1-weighted images (without and with injection of Gadolinium)
    - T2-weighted images
    - T2*-weigted images
    - FLAIR
    - DTI
    - MR spectroscopy
  - Other imaging modalities (if any)
- Related metadata concerning the acquisition protocol
- Related metadata concerning the patient
  - Demographic data (sex, age, marital status, occupation, etc)
  - Pathology
    - Strongly depends on application
      - Phases : e.g. early phase of stroke, follow-up
  - medication (drug)
  - radiotherapy
  - results of biological tests
  - results of clinical tests
  - results or neuro-psychological and behavioural tests
- Data processing
  - format conversions
  - image processing (denoising, bias correction, inter-modality registration, segmentation, normalisation, tissues and lesions classification, fiber tracking …)
  - statistical analysis (e.g. AnaCOM)
- Processing tools
  - Elementary processing tools:
    - Inputs and outputs
    - Nature of data processing
    - Pre and Post-conditions
    - Constraints on platforms (Linux, Windows, Mac)
  - Composite processing tools (pipelines):
    - Connections between outputs and inputs
    - Specific constraints
  - Execution jobs (instances of execution of a processing tool, applied to a particular set of inputs)
- Processed neuroimaging data
  - segmentation dataset
  - registration datasets
  - templates datasets
  - etc.

- Imaging biomarkers
  - Volume of anatomical structure
  - Volume of lesion
  - cortical atrophy
  - lesion load
  - etc.
- Related ROIs and annotations
- Brain anatomy
  - Relevant anatomical structures:
    - Hemispheres, lobes, cortical regions, gyri, pars
    - Brain stems
    - Brodmann areas
    - Vascular territories
  - Relevant relationships (e.g. “part-of”, “topology”, etc)
  - Degree of granularity expected by the applications
- Atlases
  - Determine which atlases are needed: Talairach, Tzourio parcellation, Colin?

Queries: The goal is then to analyze detailed examples of queries that are needed by the applications. Particular attention has to be paid to “semantic queries”, that will use ontological knowledge about related entities, e.g.:

- queries specifying a general entity class (e.g. ROI delimiting an “Hippocampus”), leading to the retrieval of instances that are specializations of this entity class (e.g. ROI’s delimiting a “Right-hippocampus” or a “Left-hippocampus”)
- Queries mentioning classes of Datasets at various levels of the taxonomy, e.g. MRI Dataset, T1weighted MRI Dataset.

Other types of reasoning: Any kinds of reasoning that may arise from the application have to be documented. For example,

- Spatial reasoning based on ROIs, e.g. related to the use of the “part-of” and “located-at” relationships.
- Reasoning concerning the creation of new processing pipelines, which will use both general knowledge about processing tools and specific characteristics of the processing tools to be piped.
- Reasoning concerning the creation of a specific processing job, which will use both the specific characteristics of the concerned processing tools and the characteristics of data to be processed.

1.4.2. Collection of detailed descriptions of relevant entities

Detailed information will be collected from the application experts in various ways.

Concerning Datasets and related data, detailed information will be collected, e.g. through existing databases schemas and other documentation of existing data. For example, concerning the Tumour application, the relational schema used in the Grenoble *eTumor* project will provide a valuable input.

Concerning the Processing tools, a detailed semantic description of all the tools that are supposed to be shared in the NeuroLOG system must be done, in terms of inputs, outputs, data types, pre and post-conditions, platforms, etc. Such list of items will be established, then tested and refined on a limited number of processing tools (e.g. FSL/BET and LOCUS). These examples will then be communicated to the application experts as a model to fill in the forms concerning all processing tools to be shared.

It is expected that the detailed analysis of the Processing tools will raise new questions about the Data and vice versa.

1.4.3. Modelling of the entities and relationships

The modelling involves several steps:

1. Sort general entities (common to the 3 applications) versus application-specific ones. Actually, modelling should not be carried out application by application, but rather globally, in order to detect and take into account commonalities. In principle, it should reinforce consistency and facilitate extension to additional applications (e.g. to other pathologies).
2. Consider properties that are either involved in queries, or necessary to do the reasoning expected by the applications. A sound balance must be reached concerning this aspect, in order to limit modelling to what is really useful in the applications, while nevertheless capturing the essential properties of each entity, even if they are not used in the application in the very short term.
3. Articulate our application-specific entities and relationships to more general entities, obtained from relevant *Core Ontologies* and from the DOLCE Foundational ontology.
4. Write ONTOSPEC documents, that model entities and relationships in a semi-informal way; this will provide a documentation of the ontology. This semi-informal version of the ontology is very important because it puts few restrictions on semantics, which is not the case with a RDFS or OWL ontology. It means that this semi-informal ontology will provide a semantic reference, from which several versions of formal ontology may be derived, in order to address different kinds of needs (e.g. with priority put on high expressivity versus performance).

In addition, ontologies are also defining rules that relate different concepts:

- Rules may be useful to represent existing knowledge about our application entities, e.g.

constraints that cannot be represented in OWL-Lite.

- CORESE supports rules expressed in SPARQL, a query language for RDF, undergoing standardization by the RDF Data Access Working Group (DAWG) of the World Wide Web Consortium (http://www.w3.org/TR/rdf-sparql-protocol/).
- The CORESE rule language is based on the triple model of RDF and SPARQL. The syntax of a rule is the following, where `cos` is the predefined prefix for the CORESE namespace (http://www.inria.fr/acacia/corese#) and where the triples correspond to RDF statements whose conjunction is translated into a conceptual graph:

```
<cos:rule>
    <cos:if>
        RDF Query
    </cos:if>
    <cos:then>
        RDF Pattern
    </cos:then>
</cos:rule>
```

Rules may be very useful to represent in a formal way knowledge that could not be represented in Description Logic (DL) formalism.

1.4.4. Expression of the ontology in a formal language

We must make a choice for a representation language. Therefore, we must analyze the possibilities, advantages and limitations of several candidate languages with respect to our needs, namely RDF(S) / OWL-lite / OWL-DL. A major constraint concerns the capabilities and performances of CORESE. Currently, CORESE supports OWL-Lite only.

Besides, we must also express the ontology entities and relationships in a SQL schema, so that relational queries can be implemented using Data Federator.

So, we will have to (manually) translate ONTOSPEC definitions into a formal ontology using an ontology editor (namely Protégé), e.g. using the OWL-Lite language.

We will then translate the OWL-Lite version of the ontology into a SQL schema, and document the mapping between the two (because we will need this mapping to transform relational data into RDF triplets using METAmorphoses).

Preliminary tests will be needed to specify precisely how to create consistent representations on the ontology (OWL-Lite and relational schema). The term “consistent” should be understood here as, “*that guarantees that RDF triplets obtained from the relational data, via the*

METAmorphoses tool based on the OWL-Lite – SQL schema mapping, represent OWL-lite instances of the ontology".

1.5. Quality assurance of the ontology

1.5.1. Quality assurance

Validation has to be done in close cooperation with the applications experts. We must make sure that the proposed ontology is relevant in order to:

- annotate the neuroimaging data and the processing tools;
- meet our expectations in terms of querying and reasoning.

We must also make sure that such annotations can be produced from existing information (e.g. images in native formats, i.e. DICOM).

1.5.2. Verification of the suitability of the semantic tools

Beyond the preliminary tests that have already been achieved, we have to assess the performances of METAmorphoses and CORESE in realistic situations.

So we must define such "realistic conditions", and then set up the experiments allowing determining the response time that can be achieved using these tools.

1.5.3. Ontology version management

An adequate version management system needs to be used to manage dependencies between the software and the ontology.

Regarding the development process, we envisage a two-step implementation, with:

- a "Basic ontology", offering limited reasoning features;
- an "Enhanced ontology", offering more advanced possibilities.

2. Interface Specifications with « Data Federator »

Schedule: M6 (task L1.1)

Responsible: Business Objects

Partners: INRIA Rennes

2.1. Objectives

The purpose of this chapter is to describe the access procedures and the programming interface with «Data Federator» (DF), especially with respects to the security issues, the mediation and the collection of data for the purpose of image processing or reformulation of data queries

2.2. « Data Federator » interfaces

2.2.1. Accessing Data Federator through the JDBC interface

Data Federator is accessed by two NeuroLOG components: the Query Interface component integrated into the NeuroLOG client application and CORESE[1] to fill the Semantic Repository. CORESE uses *METAmorphoses*[2] to convert tables published by Data Federator into RDF and insert this information into the Semantic Repository, this is further detailed in §3.3.3.

The interface to access data published by Data Federator is the Java Database Connectivity (JDBC). The JDBC API is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases – SQL databases and other tabular data sources, such as spreadsheets or flat files. The JDBC API provides a call-level API for SQL-based database access.

Data Federator supports the Catalog / Schema / Table hierarchy of JDBC. It means that it is possible for example to have one catalog containing the virtual tables prepared for Query Interface component and another catalog with virtual tables prepared for METAmorphoses, Semantic Repository and CORESE. The data in both catalogs may overlap but it is organized in different manners.

This JDBC standard specifies the Java objects used to query data and manipulate the query results. The query language used is dependent of the database server considered. For Data Federator, the query language is SQL-92 with functions listed in Data Federator User Guide.

2.2.2. Data Federator security

Data Federator respects SQL-92 standard for SQL SELECT statements but also for user rights management. It is possible to

1 http://www-sop.inria.fr/edelweiss/wiki/wakka.php?wiki=Corese

2 METAmorphoses is a DB to RDF transformation software for the semantic web (http://metamorphoses.sourceforge.net/)

create *users* and *roles*. A *user* can be member of a *role* and a *role* can also be member of another *role*. Privileges are granted to a node of the hierarchy of *roles* and *users*.

For example, to specify that user `Bob` can access table "`patients`" in catalog / schema "`neurologV01.mysql`", the user has to execute the following SQL statement:

```
GRANT SELECT ON neurologV01.mysql.patients
```

Concerning encryption of data communications, Data Federator supports all cryptographic algorithms embedded in Java Runtime Environment 5.0. For example 256 bits AES key over SSL is supported.

2.2.3. Deployment

Each NeuroLOG site will have one running Data Federator Query Server integrating local database and remote data access of other sites. Local database will be configured and accessed using a JDBC connector (see section 2.2.4) and all remote data sites will be configured and accessed by a Remote Connector ("Remote Query Server" in DF User Interface).

The mapping from local tables to common unified tables is performed locally on the same site as the database. When a *Remote Connector* is used, it queries the tables already unified. Consequently, the unification is always done by the database administrators and a site is autonomous for the management of their local databases and associated mapping.

On each site, the mapping installed in Data Federator allows to deactivate a remote site. For example, if a site `S3` is not available, `S2` and `S1` can deactivate mappings using `S3` in order to avoid blocking accesses, and to be able to read metadata from `S1` and `S2`.

2.2.4. Interface between Data Federator and local databases

JDBC is used to communicate with Data Federator clients (see section 2.2.1) but also to communicate with database servers integrated to Data Federator by means of JDBC Connector of Data Federator. The JDBC Connector supports many database systems:

- Microsoft Access
- IBM DB2
- MySQL
- Oracle
- Microsoft SQL Server
- Teradata
- Sybase

2.2.5. Interface between Data Federator and computing resources

Data Federator and CORESE are read-only systems: they are used to query existing data but cannot insert new data. In order to

register new data produced by computations, a separate database with a specific schema will be added to each site. Therefore on each site is installed:

- One **site-specific database** with a schema customized for the local need of the site. Each site remains autonomous as its database can be used for other purposes than NeuroLOG. A Data Federator mapping unifies the local schema to the federated schema.
- One **results database** with the same schema on all sites. A simple common mapping is used to give access to this database.

The common schema of the results databases will simplify the registration of new results by the computing interface component. A good candidate for this schema is the relational schema defined by the ontology as it sill also used as the federated schema published by Data Federator. The new data produced is immediately available through Data Federator since it always pushes the queries to the source databases. CORESE will take into account new metadata after the next synchronization, which brings the new metadata into the Semantic Repository.

The steps involved when performing a computation with the NeuroLOG system can be summarized as follows:

- From CORESE or DF: the user selects an input data (result: `File UID`).
- The file controller converts the `File UID` to `file path` (result: `file path`, see section §3.3).
- The compute interface uses `file path` to execute program (result: `result file path`).
- After computation, the compute interface registers the new file(s) through the site file controller and registers the new metadata into the site results database.

3. Specification of the Interface for image processing workflow composition and computing resources management

Schedule: M6 (task L2.2)

Responsible: INRIA Rennes, LARIA

Partners: I3S, BO

3.1. Objectives

The NeuroLOG middleware involves different software components to manage data and processings on a distributed infrastructure. The aim of this section is to describe the interactions between the image data management system, the metadata manager and the processings execution interface. The aspects related to data security are postponed to the next section. The section is organized as follows. The representation of processings and the methods to invoke application codes is first discussed. The access to the various kinds of data manipulated is then described. The method to invoke processings on the data sets registered in NeuroLOG is finally described.

3.2. Representation and execution of image processing tools

An ontology of image processing tools will be developed in the context of the project to describe the different algorithms and processing chains that will be deployed. A registration mechanism for image processing tools is thus needed to describe and record new software components as well as a repository of algorithms. Code invocation can be envisaged both on the local system and remotely (*i.e.* on the grid infrastructure). The repository and the code invocation mechanisms will take into account both cases.

3.2.1. Image processing tools representation

Using an ontology makes it possible to share both general knowledge about processing tools (regarding e.g. the algorithms they implement and the datasets they accept as input and create as output), and specific knowledge about particular instances of processing tool. For example, one can state that an essential property of a registration tool is to have at least two inputs - a source image and a target image – having the same number of dimensions. For a specific registration tool instance, one may want to state that it acts on 3D images (X Y Z variables of space) ; furthermore, one may state additional pre-conditions to further specify the domain of use of this tool, such as to be applied only to images with resolution ≤256x256x256, and pixel value represented as unsigned char (<256). Other information may also be associated such as through which web service this processing tool may be invoked.

The descriptions of the processing tools may also be used to create pipelines by connecting together elementary processing tools. For

example, connecting the output of a tool #1 as input of a tool #2 should not violate existing constraints concerning either of them, e.g., if tool #1 generates images represented in float, this output cannot be used as input of tool #2 if the latter accepts only unsigned char images.

Finally, such descriptions may also be used to control whether a processing tool (elementary processing tool or pipeline) is suitable to process a particular Dataset, by verifying that existing constraints are not violated. This should be done by the Compute Interface at execution time.

It remains to be decided whether such descriptions will be managed in a centralized or a distributed way. The major arguments in favour of a distributed approach are: (1) that it would be elegant to share processing tools in a similar way as data, i.e. through locally-managed descriptions; (2) that such descriptions may evolve in time based on local decisions, e.g. concerning on which local platform the program may be executed; (3) that the description of new processing tools should not lead to any changes of the ontology, since it is one of our basic assumptions that the ontology is general enough to cover the needs of all sites (according to our “Local as View” approach of the mediation problem). Conversely, the arguments in favour of a centralized approach are: (1) that such tools descriptions may be relatively complex and so hard to create and maintain by local system administrators; (2) that such tools descriptions must be carefully defined otherwise the sharing and re-use of the processing tools will not be guaranteed; (3) that there is no added value of mediation through Data Federator, since the descriptions will be made according to the same database schema. Another question concerns the syntax of such tools descriptions: relational tables versus RDF, since the syntax that will be used for reasoning is certainly the RDF one.

3.2.2. Invocation of image processing tools

The case of local and remote image processing tools invocation have to be distinguished. To invoke image processing tools, the corresponding application programs first need to be registered into the system. The NeuroLOG metadata management system and the semantics data repository will enable the description and search of existing algorithms. A relational table for *image processing algorithm* will be set up in which each site, and will describe the local algorithms. The corresponding application programs also need to be physically registered into the system to enable invocation.

We propose to deploy a Web Service interface of application programs to standardize the application code calls. Beyond being a standard, this interface has the advantage that it can be used both for local and remote invocations, thus simplifying the management of image processing tools. In addition, Web Services are clearly described through WSDL documents that can be distributed through the NeuroLOG file management system. It should be noted that the application programs considered in the context of this project are not instrumented with Web Service interfaces. The encapsulation of application programs in standard service wrappers is part of the work program of NeuroLOG (task 4.1). It should also be noted that the

interface to grid computing resources requires remote login to a gateway (see section 5.2). It will be the responsibility of the web service to interface to the grid on behalf of the user and to log remotely.

The algorithms will be registered locally on the sites they are deployed through the metadata system and their Web Service interface. They also need to be registered on the grid infrastructure if remote execution is expected. There are two possibilities to execute programs on grid resources: either to transport the programs with each job or to pre-install the programs on the infrastructure. The first solution has the advantage that it does not require any specific pre-installation but it introduces a large overhead especially for programs that are executed frequently. This solution is limited to small programs. The installation procedure is heavier as it requires installing programs on every grid sites independently (up to 90 sites in our case).

3.3. Access to data

A mechanism to coherently access the NeuroLOG data, either locally or remotely is required. The case of image files, associated metadata and semantics data also have to be considered.

3.3.1. Local and remote data files

The access to image files stored on participating sites requires the deployment of a *site file controller*: a data access control software component that will receive user requests for data, perform access control, ensure encryption of any data transmitted and deliver the data. The access control and encryption mechanisms are discussed in section 4, which relates to security. Images may also be registered on and accessed from the grid infrastructure through the grid middleware. To ensure coherency, a unique file identification mechanism will be supported. To each file registered into the system will be associated a Unique IDentifier (UID). The UID will be registered in the metadata system and associated either to a local file path or to a grid file identifier. A UID is associated to a file as soon as it is registered in the NeuroLOG system. The users will register either input data files or files produced as the result of some computing. We expect that most input data files will be stored locally and most computational results will be registered on the grid but this is up to the user to decide were to store data. We propose that the UID of files either belonging or produced by users from a site are stored on this site. The site file controller will be able to retrieve files identified by their UID, either through local file access or by querying the grid data management system. At a larger scale, the federation of metadata through Data Federator will enable the identification of sites owning a specific UID.

The files can be delivered to any authorized client able to connect to the site file controller. In particular, files should be accessible directly from grid nodes given that outbound connection is possible on almost all computing sites participating to the infrastructure, provided

that the remote process can identify on behalf of a user (*i.e.* through her grid credential).

3.3.2. Access to metadata through Data Federator

Image descriptions (information about patient, study, acquisition of image, etc) will be stored on each site in Relational Databases like MySQL, SQL Server or Oracle and accessed through Data Federator. Using a database to store image metadata and file system for the image file allows quick search on metadata by Data Federator and efficient transfer of image files by the File Controller.

Metadata databases can have heterogeneous table schemas. For example, table names or column types are different, nomenclatures used are site-specific or some columns are filled only on one site. To unify metadata, a Data Federator mapping will be done on each site. The mappings adapt to the specificities of each site through a common target tables schema derived from the ontology.

To access metadata, the user is authenticated on the local or remote Data Federator servers. Once connected, each user has specific privileges. See section 2.2.2 for more details.

3.3.3. Semantics data

For reasoning on meta-data and performing advanced searches over images and images processing tools, the semantic search tool CORESE[3] will be used. It allows searching over resources previously described using statements based on ontology. The search is processed thanks to the processing of the knowledge contained in the ontology and in the meta-data. The CORESE tool allows us to exploit the ontologies of medical images and of medical images processing tools. CORESE has been developed within the ACACIA research project in Sophia Antipolis since 2002. Technically, it allows processing OWL-Lite and RDF files thanks to the conceptual graphs technology. CORESE is also endowed with a rule engine based on conceptual graphs rules. So, our aim is to give to that tool an OWL-Lite ontology. Moreover, to fill our ontology, we must extract meta-data from Data Federator and propose them to CORESE in a RDF form. To carry out this extraction, we propose to use the *METAmorphoses* tool currently developed by the Department of Computer Science and Engineering at FEE CTU in Prague[4]. This tool needs an XML mapping between the RDF ontology and the relational schema in Data Federator. This needs to have two representations of the meta-data raises a problem of synchronization between the up to date data in Data Federator and the current data in CORESE. The first tests we made proved that *METAmorphoses* was robust enough for the NeuroLOG needs and allows us to envisage a daily populating of the ontology from the Data Federator database.

[3] http://www-sop.inria.fr/edelweiss/wiki/wakka.php?wiki=Corese

[4] http://metamorphoses.sourceforge.net/

3.4. Accessing data for image processing tools

The program execution interface needs to access data files, and possibly other metadata, for code invocation. Data files will be identified through UIDs and most image analysis programs will be executed by reading local data files with file names and parameters provided from the command line. Therefore, the code invocation mechanism needs to transform UIDs into local files.

3.4.1. Unitary invocation (local or remote)

Calls to application programs will be performed through a Web Service wrapper. This service will be able to 1) interpret files UID transmitted as input parameters, 2) to transfer the data (if needed), and 3) to build the command line invocation that encompasses local file names only. The submission service mostly needs to access the files that are not locally available on behalf of the user and prior to the code invocation.

In case of local invocation: files may be accessible locally, on a remote site or through the GRID. Local files names are resolved by converting their UID to a local file name through a simple query on the local site metadata. For remote site files, a query to the remote site file controller is first performed to cache the file locally and an access to the encryption key is done on data owner site for decryption. For GRID files, an access to the GRID data manager (see section 5.3) is first performed to cache the file locally and the owner site is queried for the encryption key. After processing, the decrypted cached files are cleared from the execution site and the produced results are returned to the caller who may decide to register some of the resulting files on his site.

In case of GRID execution, the same process applies but the necessary files need to be transferred to the GRID data management system. Indeed, only a limited amount of data can be transferred during the execution of a GRID task (in the order of 2 MB on EGEE) and large data files such as images need first to be stored in the GRID data management system prior to the code invocation. GRID files are immediately accessible: only a `UID` to `GRID_file_ID` translation is needed. Files stored in NeuroLOG sites need to be temporarily registered onto the GRID infrastructure. Ideally, file transfers should be performed directly from the local site to the GRID infrastructure. The feasibility (*i.e.* accessing the site file controller from GRID nodes) needs to be further investigated. As a fall back solution, it will be possible to transfer the files on the caller site first and to pre-register the files on the GRID prior to the execution. In any cases, the file decryption key should be accessible through the metadata interface in order to enable decryption before execution.

3.4.2. Invocation of processing pipelines on grid

To complete processing pipelines, the invocation of successive services is similar but for optimization reasons, the data transferred

should be minimized by the workflow enactor: output data files should be cached and reused as much as possible. In case of successive execution of GRID tasks, the output files will be cached on the GRID file system. In case of successive execution of tasks on a specific site, the files will be cached on this site. In other cases, file transfers cannot be avoided. This optimization work is planed in the task 4.2 of the work program.

3.4.3. Use cases

Two use cases illustrate the data manipulation and transfer needed for invoking code manipulating this data. A user invokes a registration program that will estimate the rigid transformation between two brain MR images: I1 and I2. I1 is a source image file stored at the user's local site while I2 is a processing result that has been registered on the grid previously. Both I1 and I2 are identified by a similar UIDs (UID1 and UID2 respectively) so the user consistently manipulate both files despite their different location. In the first use case, the user executes the registration algorithm locally. In the second use case, the execution is remote (on the grid).

In case of local execution, a query to the local site metadata translates UID1 into the local file name of I1. A copy of image I2 is requested to the grid data management system and if the access is granted, the file is cached to the local site in its encrypted format. I2 encryption key is retrieved and the image is locally decrypted. The registration program can then be invoked using I1's file name and I2's decrypted cached file name.

In case of remote execution, the registration program is scheduled on a grid node by the grid job management system. Prior the execution of the registration program, the grid job needs to transfer I1 and I2 locally: I1 is registered to the grid data management system. During this process, I1 is anonymized and encrypted prior to exportation to avoid any sensitive data leak. I1 and I2 can then both be accessed by the grid data manager. The grid job will still need to access I1 and I2's decryption key to get local decrypted data and perform the registration computation.

4. Specification of the Security Constraints

Schedule: M3 (task L3.1)

Responsible: Visioscopie

Partners: I3S

4.1. Objectives

The purpose of this section is to identify and to propose technical solutions for the security constraints that arise when considering the manipulation of sensitive medical data in a distributed environment such as the one targeted in the NeuroLOG project. Three kinds of data are manipulated:

1. image files,
2. associated metadata and,
3. semantic data extracted from the former.

Their respective level of confidentiality and the capabilities of the current tools are considered. This data is manipulated at different levels: locally for each participating site, through a common interface to metadata (Data Federator) and at a larger scale when distributed over the grid data management system.

The various components included in the NeuroLOG architecture are handling security at different levels. The client will be the user access point and will be in charge of coordinating security by interfacing to the various security mechanisms implanted in the data-related components such as: the local and remote file systems, Data Federator, the semantic query engine (CORESE) and the grid data management system.

The reminder of this section is organized as follows: first we make an analysis of the security requirements for the applications considered. To complete this discussion, we introduce the CNIL recommendations related to medical data. Finally, we discuss the available technical solutions and we make a proposal for the NeuroLOG architecture.

4.1.1. Security Constraints analysis

Medical data is sensitive and its manipulation over a network infrastructure, especially in the context of a wide distributed grid infrastructure, requires ensuring that it is protected. The basic requirements to ensure data protection are:

1. all users manipulating data need to be individually authenticated;
2. access to data has to be controlled at an individual level; and
3. data should not be accessible in a readable form to any user or administrator of the distributed system, except if explicitly authorized.

These requirements are stringent but in addition, we consider that to control security risks additional controls should be enforced:

4. no nominative data should ever be transferred to remote sites; and
5. the system should ensure traceability of the data accesses.

It should be noted that in the context of the NeuroLOG project where all partners are participating to research, taking into account these five points is considered to guarantee a sufficient level of security. However, deployment in a clinical context would imply additional requirements. In the context of NeuroLOG, we are studying some of them with the long perspective of a clinical deployment although actually delivering a clinical data management system falls beyond the scope of the project. In particular, clinical deployment would at least require:

6. Data re-identification
7. Role-based access control.

Requirements 1 and 2 relating to access control can be addressed through regular user identification and fine grain control mechanisms. Requirement 3 on data protection implies key-based encryption of data and keys access control. Requirement 4 is met through data anonymization and requirement 5 through system logging. These points are discussed with more details in the following sub-sections.

4.1.2. On-disk files encryption

To protect image files from other users and administrators of the distributed system, it should be encrypted with a robust encryption technique. We recommend systematic encryption of the data files that are registered into the system. Encrypting data on disk will ensure that no unauthorized user, even if he or she gets access to the machine hosting the data, cannot read the data content. In addition, manipulation of encrypted data files will ensure that any data transferred over the network is protected against third party reading regardless of the transfer protocol and the network configuration.

We acknowledge the fact that on-disk encryption of local data may be considered as a constraint for the local users who are used to make direct access to local data through other interfaces than the NeuroLOG middleware. An acceptable, more relaxed policy in the context of the project is to let local files unencrypted on their owners' disk but to enforce encryption as soon as a file is transferred or replicated on a different site (which as to be done through the NeuroLOG middleware). This policy is acceptable provided that the participating sites are responsible for the security of their resources. The deployment in a clinical environment would require the adoption of the strict policy with systematic encryption though, given the scale of hospital radiology networks and the difficulty to completely secure large institution networks.

The NeuroLOG data management system will propose both file encryption policies and let to the site managers the choice of which one to implement of a particular site.

Contrarily to image files, the medical metadata is not intended to be replicated on external sites. Hence there is no need for systematic encryption of databases that will be confined on their site. Metadata can be accessed remotely through Data Federator though. Data Federator should ensure that the metadata is protected (encrypted) during transfers and only accessible to authorized users.

The semantic data will be accessible through CORESE which does not provide any access control. Hence, no sensitive semantic data can be manipulated and the data exported to the semantic repository has to be filtered.

4.1.3. Authentication and access control

Access control is fundamental to restrain the access to sensitive data. In NeuroLOG, all users will be individually identified to guarantee access control at the finer grain (individual level). The NeuroLOG middleware will involve various software components relying on different identification and access control technologies. The user should not be exposed to this internal complexity and the NeuroLOG client will be in charge to hide the system heterogeneity by providing single sign-on and coherent access control policies to all data manipulation services.

Access controls apply to image data files, associated metadata, semantic data and encryption keys. Encrypted data files could be relaxed from any control without exposing any patient identity but to remain coherent and to conform to most restrictive policies, data files and encryption keys (enabling data decryption) should be controlled identically. Metadata is very sensitive as it contains patient identifying information and it should be strictly controlled. One has to make a distinction between identifying (e.g. patient name or other image identifiers) and non-identifying (e.g. image modality, image dimensions) metadata. Identifying metadata should be restricted to local access only to ensure that no patient identity leak can ever happen in the system. Non-identifying metadata can be accessible remotely under control. Identifying and non-identifying metadata will be recorded in distinct relational tables to ensure separate access control policies. The access to semantic data cannot be controlled with the current technologies. Therefore, only non-sensitive metadata will be exported in semantic repositories.

Individual-grain access control can be enforced through the use of Access Control Lists (ACLs). It will also be useful to define group of users with identical access rights for simplifying access control settings (e.g. groups for local users or users participating to a same study and sharing data). Among the NeuroLOG partner, access control based on user groups is considered sufficient. In the context of clinical deployment, finer grain will become mandatory though. Access to data should be controlled at an individual level and since several individuals can play a similar role (e.g. two physicians participating to the healthcare of a same patient, etc), it should be possible to assign access right to roles and to independently assign roles to users to ensure a sufficiently flexible and extensible access

control mechanism. Role-based access control technologies are today well developed although they are not yet commonly available with standard tools.

It should also be noted that the access control to each site data will be under the responsibility of the local administrator. Each site access control policy has to be integrated so that the system is accepted by the users. In particular, there is not such notion as a super user with global access to all data in the distributed system. This means that the access to data will ultimately be controlled by the site delivering the data. In particular, the encryption key for a file will be stored on the site the files belong to and its access control will remain local.

4.1.4. Anonimization

Data will be processed prior to any transmission to guarantee data confidentiality. Nominative data and examination identifying information (such as place of examination, doctor names, etc) will not be exposed. In that purpose, any identifying information will be removed and replaced by unique numbers or UIDs (Unique IDentifiers). Two kinds of UIDs are considered:

- UUID (Universally Unique IDentifier) will replace personal patient data (name, first name…) to enable data re-association with initial patient record, after GRID treatments.

  A Universally Unique Identifier is an identifier standard used in software construction, standardized by the Open Software Foundation as part of the Distributed Computing Environment. The intent of UUIDs is to enable distributed systems to uniquely identify information without significant central coordination
- Anonymization functions will be applied on specific medical imaging modality data in DICOM format. Indeed, DICOM medical modalities generate their own UIDs. Non-identifying Specific DICOM UID will be generated to replace modality DICOM UID.

Completely anonymous data (even without UUID) will finally be generated for data exportation towards the grid. The association between anonymous identifiers and original nominative information will be kept at the data owner site.

The anonymization process will take place in the data importation mechanisms. It will be executed only once for each imported data, limiting the wastes of time.

4.1.5. Traceability of data accesses and transfers

To ensure a high level of security and recover on attacks or malignant use of the systems, it is important that all the data access activities are logged and traceable. This follow-up mechanism will trace actions of the users and data exchanges. More precisely, the following events will be logged:

- Sessions Time-stamping (date and hour of users connection, sessions duration…),
- Logging of data processing (import, export, inquiry and delete),
- Loggings of exchanges and works since and towards grid computing.

Mechanisms of non repudiation, based on digital certificates, will come to supplement these log files and ensure their veracity guarantee: non-repudiation of origin proves that data has been sent, and non-repudiation of delivery proves it has been received.

4.2. CNIL recommendations

The French « *Commission Nationale de l'Informatique et des Libertés* » (CNIL: National Commission for Data protection and the Liberties) emitted some recommendations about personal data health processing:

- to guarantee the patient anonymity before any data transmission towards a third party;
- to preserve information safety and integrity, related to patient health state.

The CNIL also published recommendations on the networks exchanges within data health processing systems5:

- Management of passwords: individual user password distinct from the user name; prohibition to re-use the last three passwords (system blocking).
- Methods of connection and disconnection:
  - Impossibility for several users to connect under same user name and password;
  - Systematic display of the last date and hour of user connection.
  - After several inputs (e.g. three) of incorrect password (associate to a correct user name), access is blocked and a message is displayed, asking user to call system administrator.
  - Automatic disconnection procedure in case of non-utilization of the system during a certain time (time out).
  - Use as far as possible of smart cards or similar devices (e.g. CPS card, health professional card).
- Logging of connections and exploitation of these data.
- Data confidentiality:
  - Use as far as possible of personal data coding.
  - Total or partial data encryption in compliance with French and European regulation.
- Data Integrity

[5] Source: http://www.cnil.fr/index.php?id=1367

  - Deployment of adapted transmission protocols allowing checking conformity between received and emitted data.
  - Use of standardized procedures guarantying data integrity while medical images digitalization and compression.
- In case of client-server architecture:
  - Manage access data and file transfer according to user's rights: minimizing complete file transfer and volume data, request logging on the server.
  - Separation of administrative and medical management networks.
- Internet connection:
  - Create separated physical network for each server that connects to the internet.
  - Provide a firewall or a software protection suite.
  - When health data are transferred via Internet, use of communication encryption (e.g.: SSL encryption with a 128 bits key)[6].

4.3. Technical implementation

In the former sections, we have reviewed the necessary security components that have to be integrated into the system. This section discusses the technical solutions that are available and that we plan to use.

4.3.1. Users identification

On grid infrastructures, users are identified through standardized X509 certificates [12] signed by a Certificate Authority (CA). The CA is a recognized institution in charge of controlling the identity of individuals to which it delivers certificates. In NeuroLOG we will rely on the GRID-FR CA hosted by CNRS (http://igc.services.cnrs.fr/GRID-FR) which is recognized by the international EU-GridPMA coordination organization (http://www.eugridpma.org, European Policy Management Authority for Grid Authentication). The user certificates are valid for a restricted period of time (usually one year) beyond which they have to be renewed by the CA. They are nominative and they contain a Distinguish Name (DN) that uniquely identifies the owner by its name and affiliation. Certificates are divided in a public and private part. The public part is world-readable and can be used by anyone to control a user identity. The private part is accessible to the owner only and can be used by its owner to identify himself securely. It is usually on the responsibility of the user to keep his certificate private part secured. To avoid exposition of the private part of the certificate it is password protected. Both for security reasons and convenience, users do not directly use their certificate for accessing grid services. They use proxies instead. Proxies are temporary certificate

[6] Source: http://www.cnil.fr/index.php?id=1321

generated and self-signed by the user. A proxy has a typical lifetime of 12 hours and thus it is a minor security risk if it becomes compromised. Proxies are self-signed but of course only valid provided that the signing owner is himself validated by a CA (the root of the chain of signatures belongs to a CA).

In the context of the NeuroLOG project we propose that the client keeps the certificate private part and ensures that it is read protected on the local file system. On local systems users are usually identified through a simple login and password. The NeuroLOG client will use this login-password identification and associate each login to the corresponding user's certificate. It will thus hide from the user the grid authentication mechanism.

In the clinical environment, CPS smart cards (*Carte de Professionnel de Santé*) are commonly used to control access to medical resources. We propose that the client is instrumented with a CPS interface and that it associates each CPS to the owner login and certificate for transparency.

4.3.2. Access control

Access to files and metadata is controlled through different technologies. Encryption keys can be stored in databases and will be considered as metadata. Two different cases have to be considered for files: locally stored files and grid files.

Files will be locally stored in UNIX file systems for which the access control is usually limited to the owner, the owner's group and all users (owner/group/other). This does not enable fine-grain access control as desired for NeuroLOG's data. To remain compatible with local access control policies, it is important that the files remain locally accessible. The access control will be enforced at a upper level for remote file accesses. The NeuroLOG middleware component responsible for delivering files outside will therefore check user’s identity based on their certificate and grant access only to authorized users. This implies an ACL-based control and therefore the association of ACLs to files in the local metadata database.

For grid files, access control will be delegated to the grid middleware. Certificates uniquely identify users but they are not sufficient to define access control policies. On the EGEE grid infrastructure, access to resources is controlled at a coarse grain by Virtual Organizations (VOs). A VO is a boundary-less group of users sharing a common interest for a scientific discipline. A VO Management Service (VOMS) [13] controls the affiliations of users for each VO. All users of the NeuroLOG project will be affiliated to the *biomed* VO (https://cclcgvomsli01.in2p3.fr:8443/voms/biomed). In addition, a VOMS enable the definition of groups and the assignment of users to groups. Access to files can be controlled through ACLs specifying access rights either for individuals or for VOMS groups. A *neurolog* group will be created in the *biomed* VO to isolate the NeuroLOG data from other users in the VO. Other sub-

groups will be created as needed to map the NeuroLOG system groups. This approach is valid but not very scalable as the VOMS administrator (external to the NeuroLOG project) will be solicited for new groups creation. In a longer term one could imagine a specific VOMS server administrated by the NeuroLOG members to ensure more direct control.

For use in clinical data, RBAC technologies should be investigated. It should be noted that the current UNIX file systems and the EGEE middleware do not include RBAC access control currently. It would be necessary to investigate external solutions such as the Shiboleth system [14].

The access to metadata will be achieved through Data Federator which relies on the SQL-92 access control policy. One DF instance will be deployed at each site and the access is thus controlled locally. SQL92 enables fine-grained access control through granted privileges assigned to table or column paths. Different set of privileges will be assigned to different columns depending on their sensibility (nominative data, non-nominative data or encryption key) as reported earlier.

4.3.3. Data encryption

As discussed above, local data will not be encrypted but data encryption will be enforced prior to any data transfer outside a site. Symmetric key-based encryption algorithms are today widely accepted and represent very reliable and fast encryption techniques which robustness can be tuned by setting the encryption keys length. The AES (Advanced Encryption Standard) algorithm [15], which is promoted by the US government for its administrations, is a widely available standard with open source implementations available. It is thoroughly analyzed and recognized robust by the security community. 128 bits encryption keys are usually used. It ensures a high level of security and low time and space overheads: according to its conception based on byte permutation, AES is one of the faster encrypting algorithm (2.7 time faster than 3DES for example), and does not increase the data size. For example, on actual computers, it takes less than 2 seconds to encrypt a 25 MB file with AES, and it increases in size by approximately 0.003% when encrypted.

We plan to store encryption keys locally on sites owning data. It should be noted that there exists a grid key store service called Hydra that is planed to be integrated in the future releases of the gLite middleware and that could become of interest [4]. Hydra provides ACL-based access control to the encryption keys and secured communication to the requester. In addition, Hydra exploits the Shamir secret sharing scheme [3] to improve security and reliability of this service. Shamir's scheme consists in splitting keys into n fragments stored in different places. Only m (with m<n) fragments are needed to reconstruct a complete key. However, owning less than m key fragments does not give any information on the complete key. Thus, the system is both resistant to attacks (at

least m key stores need to be compromised for an attacker to be able to reconstruct a key) and reliable (the disconnection or loss of a limited number of servers does not prevent the key reconstruction). The Hydra servers hosting the key shares are completely identical in terms of interface and functionality.

4.3.4. Traceability

Logging of actions and data exchanges associated to time-stamping and digital signature will ensure traceability. In addition, the use of individual certificates enables non-repudiation as the user cannot deny its actions.

For security reasons, all logs, generated by the software, will be stored in database. Indeed, SQL data bases mechanisms integrate user profile management functions which make it possible to restrict the access to these data to local administrators.

The documents recorded will be nesumerically signed by the software using the X.509 certificates. This technique guarantees the document authenticity in term of space/ location (User, machine…) but not in time. Thus, a third server will be used as Time-stamping Authority (TA), guarantor for each logs entry, which certifies hours and dates, without having interest to falsify them.

The time-stamping generic principle is as follows:

- The data to timestamp are passed through a hash function (SHA-1, MD5…) .
- The resulting hash value (a fixed value length, of 128 or 160 bits for example) is sent to the Time-stamping Authority (TA).
- The TA relates the GDH (Group Dates Hour, with format UTC: Coordinated Universal Time whose format is YYYYMMDDhhmmssZ) to the hash value.
- The TA signs the concatenation Hash + GDH (by means of RSA, DSA…): the result is a capsule CMS (Cryptographic Message Syntax, RFC 2630), called token (Time-Stamp token).
- The whole is returned to the applicant who has a proof then owing to the fact that the hash value existed before the given date in the GDH. The hash value being a “print” of the initial data, the proof applies also to these data.

The most used signature algorithms for the time-stamping server are RSA or DSA. The choice between these two algorithms will be done by considering the expected performances and the relative proportion of signatures and checks of signature in the whole infrastructure.

In NeuroLOG, the signature mechanism and time-stamping will be as follows:

1. The original document is written without encryption.

2. Its signature is made up of its numerical print quantified with the private key of its author.
3. A time-stamped token is computed and the unit is signed by the service of time-stamping.
4. The original document, its signature and its token are recorded.

4.3.5. Interface from the user point of view

The user has a single federative interface provided by the NeuroLOG project client software. This application has a local management of users, with for each one a record containing the necessary elements for databases access and grid connection.

At software start, the user can authenticate through several methods:

- By a login / password form.
- By the use of a CPS card (“Carte du Professionnel de Santé”, Professional of Health Card) which contains its personal information as well as a X.509 certificate,

Standard profiles (Administrator, researcher, doctors, and trainees) will be defined with a sufficient granularity for profile management.

Once authenticated, the user will access an interface adapted to her profile. For example, an administrator will have access to the user management functions (add, delete, modify), a researcher will be accredited to use the image importation functions, the data query functions and the job submission to the grid.

The software will relay the authentication through the different application layers in a transparent way. In background, it will be connected to the metadata thanks to Data Federator authentication mechanisms based on SQL 92 and to the grid services. Indeed, the SQL 92 standard makes it possible to define user rights thanks to the GRANT order. The SQL92 syntax for GRANT enables setting privileges for individual columns within a table, and allows setting a privilege to grant the same privileges to others.

Similarly, the software will connect to the grid services by using an authentication mechanism based on individual X.509 certificates. Each user owns a certificate delivered by a certification authority (the GRID-FR of IGC CNRS certification authority for the EGEE grid). The software recovers this certificate in the user computer and uses it to ensure the user authentication on the grid

An authenticated user can make all manipulations authorized by her profile, querying data, submitting jobs on the grid, from a unique federative graphic user interface.

5. Specification of the interfaces to access the computing GRID engine

Schedule: M6 (task L4.1)

Responsible: I3S

Partners: INRIA Rennes (Paris project)

5.1. Objectives

The purpose of this chapter is to describe and specify the different procedures to acces to the computing grids (EGEE, Grid'5000), to define the computing interface to the GRIDs and to define how the image processing workflows will be set up on the GRIDs.

Two grid infrastructures are envisaged in the context of the NeuroLOG project: the EGEE production grid (http://www.eu-egee.org) and the Grid'5000 (http://www.grid5000.fr) experimental grid. Both infrastructures have completely different access interfaces, data and workload management systems. In this document, the EGEE grid is mostly considered given that it provides security feature and high level data management that are mandatory for exploitation in production as expected in the context of NeuroLOG. The interface to Grid'5000 is described though as it will be used for testing and prototyping.

5.2. Grid interface and authentication

The use of both grid infrastructures requires logging on a GRID gateway on which the GRID client is installed. Some components of the EGEE grid infrastructures are planed to expose web services. This will avoid the use of the intermediate gateway but the deployment of these interfaces will be concurrent to the NeuroLOG project and an alternative solution has to be considered in a first time.

The EGEE grid gateway, known as the User Interface (UI), is a PC running the Scientific Linux v3 OS and the gLite middleware client [4]. Anybody can install a new gateway and it will make sense to deploy one of them for the needs of the NeuroLOG users. Grid users need an account on one of these UI. The users are authenticated and authorized through X509 certificates that were already discussed in section 4. The GRID-FR CA will deliver certificates to all NeuroLOG users. The project partners have been declared to the CA in this purpose. A user certificate needs to be registered on her UNIX account. The NeuroLOG client will need to make remote connection to this interface and to initialize there a proxy on the behalf of the user to access the EGEE data management and workload management services. A proxy is created with the command:

```
> voms-proxy-init --voms biomed
```

The properties and remaining lifetime of a proxy can be queried with:

```
> voms-proxy-info --all
```

The Grid'5000 gateway is a standard Linux PC. Grid'5000 is divided in a dozen of sites (clusters) all around France and there is a single

gateway to each site. To access to Grid'5000 a user need to obtain an account on a gateway from one of the sites administrator. The files inside a Grid'5000 cluster are shared through NFS. There is no file sharing among clusters and data has to be replicated if it is used on several clusters. The system workload management system is the OAR batch scheduler which client is installed on each gateway.

5.3. Data Management Systems

The EGEE Data Management System is composed of a file catalogue and a list of storage sites. There exists one file catalogue for each VO so the data manipulated in every VOs are compartmented. The gateway to each storage site is known as a Storage Element (SE). All SEs expose a common interface to the grid (an OGF standard known as *Storage Resource Manager*) to hide the heterogeneity of storage resources (disks, tapes, MSS…). On the EGEE grid, each file is identified by a Grid-wide Unique IDentifier known as GUID. The file catalogue is a centralized index establishing a relation between GUIDs and the corresponding file name and location. Several replicas may exist of a same physical file to ensure fault tolerance and speed up data accesses. Therefore, a single GUID may be associated to several physical instances. There is no problem with the coherency of replica in the sense that grid files are read-only. A user can create new files and delete old ones but not modify existing files. For the needs of users, human readable Logical File Names (LFNs) can also be associated to files. The file catalogue holds the association between LFNs and GUIDs.

The major commands to manage data on EGEE are summarized below. Details about each command can be found in [4]. The file catalogue can be manipulated through command whose name is prefixed with "`lfc-`" while the storage elements can be manipulated through command whose name is prefixed with "`lcg-`":

For the VO Biomed, the list of accessible storage elements is obtained with:

```
> lcg-infosites –vo biomed se
```

The root file catalogue server is defined through the LFC_HOST environment variable and a default path can be set with LFC_HOME:

```
> export LFC_HOST=cclcglfcli02.in2p3.fr
> export LFC_HOME=${LFC_ROOT}:/grid/biomed
```

Existing files can be listed with:

```
> lfc-ls
```

New directory (file collections) can be created and deleted with:

```
> lfc-mkdir
> lfc-rm –r dir/name
```

Files are registered to the grid (**c**opied and **r**egistered) through:

```
> lcg-cr --vo biomed –d se_name -l
lfn:logical/file/name
file:///local/file/absolute/path
```

Conversely, grid files can be retrieved through the command:

```
  > lcg-cp --vo biomed lfn:logical/file/name
file:///local/file/absolute/path
```

This commands returns a GUID for the new file that can be used to replicate the file to a different storage:

```
  > lcg-rep --vo biomed -d target_se guid:xxx
```

The list of replicas is known through:

```
  > lcg-lr --vo biomed lfn:logical/file/name
```

Finally, a file entry and all its replica can be deleted as:

```
  > lcg-del -a --vo biomed lfn:logical/file/name
```

As detailed in section 4, access control to files is expected at individual and group levels in NeuroLOG. The EGEE VOMS service defines a notion of groups of users known as *roles*. The access to files registered on EGEE can be controlled for each of these roles through the `lfc-setacl` command. Suppose that a `neurolog` role has been declared in the biomed VO and that the user with DN "`/O=GRID-FR/C=FR/O=CNRS/OU=I3S/CN=Johan Montagnat`" belongs to this role. A simple example is illustrated below:

Create a directory to protect named 'wdir' at the file system root:

```
  > lfc-mkdir wdir
```

Assign read-write-execute permission to this directory for this role (`g:biomed/Role=neurolog:rwx`) but no access rights to any other groups (`g::`) nor users (`o::`); apply a read-write-execute mask (`m:rwx`) and similar default rights (`d:g`, `d:o` and `d:m`):

```
  > lfc-setacl –m
g:biomed/Role=neurolog:rwx,g::,o::,m:rwx,d:g:biome
d/Role=neurolog:rwx,d:g::,d:o::,d:m:rwx wdir
```

Check the corresponding ACL:

```
  > lfc-getacl wdir
  # file: wdir
  # owner: /O=GRID-FR/C=FR/O=CNRS/OU=I3S/CN=Johan
Montagnat
  # group: biomed
  user::rwx
  group::---              #effective:---
  group:biomed/Role=neurolog:rwx    #effective:rwx
  mask::rwx
  other::---
  default:user::rwx
  default:group::---
  default:group:biomed/Role=Johan:rwx
  default:mask::rwx
  default:other::---
```

The Grid'5000 file management is performed through NFS inside each cluster. The standard UNIX file system commands apply. To transfer data between sites, the `scp` command should be used.

5.4. Workload Management System

EGEE computing tasks are handled by its Workload Management System (WMS). The entry point to the WMS is a Resource Broker (RB)

which knows a list of sites and their gateway known as Computing Elements (CEs) and hosting batch systems. A default RB is declared on each User Interface. Computing tasks are sent by the client to the RB. The RB searches for matching resources among the existing sites and pick a “best” CE. The computing task is delegated to this CE’s batch manager. A task is described through a small file with the Job Description Language (JDL) syntax. A concrete example of JDL creation and its execution on the grid infrastructure through the `edg-job-submit` command is illustrated below:

```
> cat hello.jdl
Executable    = "hello.sh";
Arguments     = "";
InputSandbox  = {"hello.sh"};
StdOutput     = "hello.out";
StdError      = "hello.err";
OutputSandbox = {"hello.out","hello.err"};
> edg-job-submit --vo biomed hello.jdl
```

It should be noted that the standard output and the standard error stream of the remote process are directed to files (`hello.out` and `hello.err` in this case). Those files need to be retrieved as part of the job results (the output sandbox) if the user wants to read them. Conversely, the input sandbox will transport local files to the computing node prior to the job execution (the `hello.sh` script in this example). The job submission command returns a unique job identifier formed as a URL (`https://xxxx`).

Additional requirements can be specified in the JDL file to constrain the target to which a job can be submitted. For instance, to enforce the execution to the `grid10.lal.in2p3.fr` CE, the following line should be added to the JDL:

```
Requirements  = other.GlueCEUniqueID ==
"grid10.lal.in2p3.fr:2119/jobmanager-pbs-sdj";
```

Once a job is delegated to a site batch system, it can be periodically queried to determine its progression:

```
> edg-job-status https://xxxx
```

The `edg-job-status` command returns a status such as “submitted” (job was submitted to the RB), “ready” (matching resources have been found), “queued” (job was delegated to a CE and is pending in a queue), “running” (job is currently executing), “done” or “failed”. Once a job is done and its output is ready it can be retrieved on the UI through:

```
> edg-job-get-output --vo biomed https://xxxx
```

Additional information on the job life cycle can be obtained through the execution trace:

```
> edg-job-get-logging-info --vo biomed
https://xxxx
```

A Web Service interface (known as WMProxy [5]) was recently added to the EGEE job management system. It is not available for testing

already but it should be deployed in the coming months. It will be interesting for the NeuroLOG project to follow on its development.

On Grid'5000, there are two very different ways of accessing computing resources: resources reservation and system images deployment. Resources at each site are controlled by a reservation system named OAR [6]. OAR can be used to allocate a given number of resources to the requesting user immediately or at a given time. The user can then log on the reserved resources and execute any task locally. OAR applies a first *come first served* policy. Schedules are published through the Grid'5000 web site to inform users of the resources availability through time. Typical examples of the OAR reservation mechanism are:

Immediate reservation of one node (the request is pending if no resource is available immediately):

```
> oarsub -I
```

Advanced reservation of 5 biprocessors nodes on the 31st of December this year at 2pm for a duration of 10 minutes:

```
> oarsub -r "2007-12-31 14:00:00" -l
walltime=0:10:00,nodes=5,weight=2
```

The state of a reservation can be queried through:

```
> oarstat –j <res_id>
```

By default, OAR only allocates resources on the submission site. An extension to OAR called GridOAR can be used for multi-sites reservations.

Nodes reserved through OAR run a standard Linux distribution and share the user directories through NFS. Grid'5000 is also extremely reconfigurable as a user can install its own operating system and middleware stack on the nodes she reserves. It is necessary to prepare a system image and to use the `kdeploy` tool. At the moment the reservation starts, the system image is deployed on the reserved computers which are rebooted under the new OS.

5.5. GRID limitations

Software running on the grid infrastructure needs to be pre-registered to grid resources or to be transferred prior to the execution. As a direct consequence, licensed software is usually not authorized to be executed (most licenses are either limited to one host or one user and cannot be transferred in an "open" grid environment). In the context of the NeuroLOG project, this may concern the Matlab software used by several users (or the SPM libraries depending on Matlab). The licensed software tasks will therefore be limited to local sites (unless evolution of licenses enables grid execution).

The Grid'5000 grid infrastructure is completely isolated from the Internet for security reasons: processes executing on this GRID are jailed into the infrastructure and cannot communicate with external processes. In particular, it will not be possible to access external services such as local sites data manager and encryption key stores: all data needs to be transferred to Grid'5000 prior to execution. On EGEE, the communication policy depends on the sites configuration. In a vast majority of cases, processes can get outbound connectivity but no inbound connectivity is

possible. Outbound connectivity enables access to out-of-grid NeuroLOG services for GRID processes. In the context of NeuroLOG, inbound connectivity might only be needed for interactive processes and past studies have shown the feasibility of interactive bridges set up to turn around this limitation if needed. A few sites may apply different policies though: they will have to be banned from the NeuroLOG job submission engine to avoid any problem.

6. Specification of the test-bed applications

Schedule: M6 (task L5.1)

Responsible: INSERM/GIN

Partners: INRIA Rennes, INSERM/IFR49, Visioscopie, INRIA Sophia

The purpose of this chapter is to describe and specify the different test-bed applications on which the NeuroLOG architecture will be experimented.

6.1. Test-bed applications

Three clinical applications have been identified in the context of brain pathologies: Multiple sclerosis, Strokes and Tumour will be considered.

6.1.1. Multiple sclerosis (MS)

Rationale: Magnetic Resonance Imaging (MRI) has a major impact on MS investigation. The technique is non-invasive and allows one to follow-up cerebral structures and to quantify the effects of medication. Central questions in the MS context are:

1. Can we detect the disease at an early stage?
2. Can we predict its evolution? And,
3. What is the impact of drugs on this evolution?

Answering these questions requires the computerized management of large amount of data. The central points are then:

- Which cerebral structures should be followed-up depending on the clinical context?
- Which parameters can be extracted from the images reflecting the evolution of the disease?
- Which data processing chain is optimal for the extraction of these parameters?

Data providers: Two centres are considered: the Pontchaillou Hospital in Rennes (INRIA Rennes partner) and the Pasteur Hospital in Nice (INRIA-Sophia partner).

In Rennes, two clinical protocols exist. The first one is currently applied for routine clinical studies since June 2006 and the second one is under development in collaboration with several other French centres. The goal of the latter is to set up a multi-centre data set consisting of MRI data from one or several time points of patients with MS or clinically isolated syndromes suggestive of Multiple

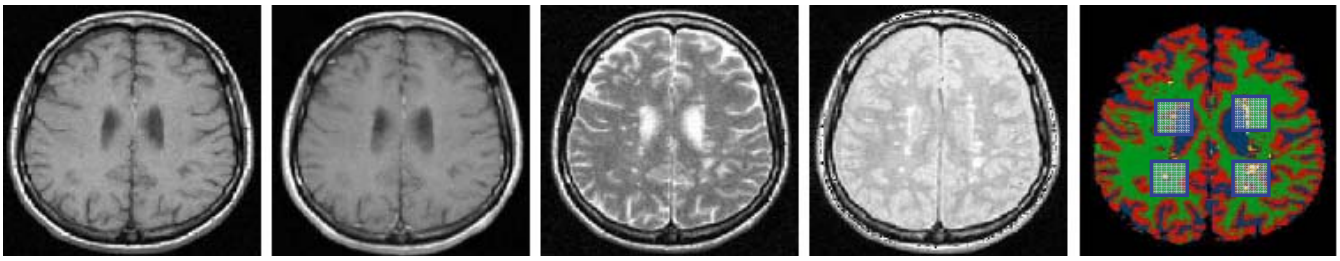

Figure 3: MR multisequence images with segmented brain tissues and lesions (right)

Sclerosis. For each patient's visit spaced from 1 to 3 months, a set of images is acquired.

In Nice, several clinical studies have been conducted. The first one, conducted in 2001-2002, aimed to propose an optimized acquisition protocol for the MR images [17]. This acquisition protocol has been used in the second study, named MCI, conducted between 2002 and 2005 [18]. This study led to built a database of 40 patients and 10 normal subjects that have been imaged at different time points. About half of this database does constitute a homogeneous subgroup with enough follow-up exams and will be available in a first step for the proof of concept. Since Rennes' hospital uses this acquisition protocol, the data provided by both centres will be comparable.

A multi-centre study, with the same acquisition protocol, and involving other hospitals, is currently in progress but the data cannot be shared before the publication of the study results (which is expected to happen before the end of the NeuroLOG project).

Data Description: In each centre the collected images are basically the same: 1) a structural image, 2) a double-echo T2-weighted image, 3) a FLAIR image and 4) a T1-weighted image after gadolinium injection. Images are initially stored in DICOM format then transformed in GIS format in Rennes and INR format in Sophia. Metadata about the patient, the medication, the pathology and the conditions of image acquisition are provided. The construction of a database is under study in interaction with partners of this project in Rennes. For Nice, a file organisation is defined. For Rennes, no local database is available yet. The construction of such a database is under study in interaction with partners of this project.

Data Processing: The data processing chain is similar for the two sites.

1. image conversion (from DICOM to GIS or INR);
2. inter-modality registration of the set of images, with a preprocessing step from INRIA-Rennes consisting in bias correction, denoising and intensity normalisation; a bias field model is also computed for INRIA-Nice after the first step of classification (EM Algorithm);

3. normalisation of the images (for the use of a priori spatial information);
4. skull striping (brain masking);
5. tissues and lesion classification.

6.1.2. Stroke

Rationale: MRI is currently the ideal imaging modality for the diagnosis of acute stroke. Presently, it allows for assessment of patients with acute stroke, detection of both cerebral ischemia and intracranial haemorrhage and discrimination of cerebrovascular causes from other causes. Several types of MR images are required to assess the diagnosis in the early phase and to follow-up the evolution of the lesion under a specific treatment.

Grenoble is more interested in a clinical application like the follow-up of patients after stroke. Paris (INSERM U610/IFR 49) collects clinical and anatomical data of patients presenting with focal brain damage in order to build and organize a digital database that crosses clinical, neuropsychological and radiological information for the benefits of clinical management and research projects. Benefits expected from the database are two-fold: (1) to help the diagnosis and follow-up (including rehabilitation) of patients by providing clinicians with precise information regarding the neuropsychological and behavioral impairments and their relationship with the brain damage; (2) to favor research projects aiming at improving knowledge about "structure-function mapping" within the brain for the cognitive functions.

The central scientific questions to tackle are:

1. Does an anatomo-functional relation exist between a specific anatomical localization of the lesion and functional deficits as assessed via specific neuro-psychological and behavioural tests?
2. Clinical application: does the volume of the lesion vary across time under treatment?

Data providers: Two centers are considered: the Michallon Hospital in Grenoble (GIN partner) and the Pitié-Salpêtrière Hospital in Paris (INSERM U610, IFR49 partner).

In Grenoble, the clinical objective is the follow-up of stroke patients using Magnetic Resonance Imaging (MRI). The current clinical protocol named *Virage* is routinely used: for each patient three sets of images are acquired respectively less than four hours after stroke episode, less than five days after stroke episode and more than one month after stroke episode.

In Paris, the clinical objective of IFR49 is to help diagnose and follow-up (including for rehabilitation) the patients. Homogeneous data are collected for each patient:

1. Neurological data: a neurological exam including the National Institute of Health Stroke Score.

2. Cognitive data: a neuropsychological assessment is carried out in order to evaluate the main cognitive functions. The neuropsychological assessment follows a precise and standardized protocol: on the one hand a basic neuropsychological assessment for the focal lesions; and in addition specific tests regarding stroke localization (language, hemi spatial neglect).
3. Behavioural data: a behavioral adaptation assessment
4. Radiological data: (3D MRI, Diffusion Tensor Imaging, metabolism …).

The research aspect concerns structure-function mapping. To bring new elements for answering the anatomo-functional question, IFR49 develops and uses an anatomo-clinical overlapping map (AnaCOM) [1] method to obtain functional maps from patients with lesions. AnaCOM is a new clinical-radiological correlation method that aims at establishing structure-function relationships. The technique is based on the anatomic MRI of patients with brain lesions who are administered neuropsychological tests.

However, various aspects of data and processing methods can be mutualised between the two centres.

Data Description: In each centre the common collected images are: 1) a T2-weighted image, 3) a FLAIR image, and 3) diffusion-weighted images (Figure 2). The modification of the protocol currently used in Grenoble is under study to be in accordance with the protocol routinely used by IFR 49, including a diffusion tensor imaging (DTI) acquisition for fiber tracking and a high resolution anatomical (T1-weighted) scan. The latter is essential for the use of the AnaCOM methodology [1].

The new protocol, including a neuro-psychological examination, will be applied in Grenoble to all young stroke patients (<60 years old) before possible return to work. At least 30 patients by year could be included. Images are available for GIN (acquired on a 1.5T Philips Intera) and IFR49 partners in the DICOM format, but Philips format (.par and .rec) files could be stored if needed for GIN. For

Figure 4: Stroke lesion on a diffusion-weighted image

data processing pipelines they are further transformed in Analyze (SPM2) format. Metadata about the patient, the medication, the pathology, neurological, cognitive and behavioural data, and the conditions of image acquisition are also provided.

At GIN, no local database is available yet. The construction of such a database is under study in interaction with partners of this project. Presently, neurological and cognitive data is not stored in a digital form.

For IFR49, data is organized in an anatomo-functional database, called CAC database, which is currently under re-structuration (MySQL database). Approximately sixty patients are already stored. At least 30 patients by year could be included.

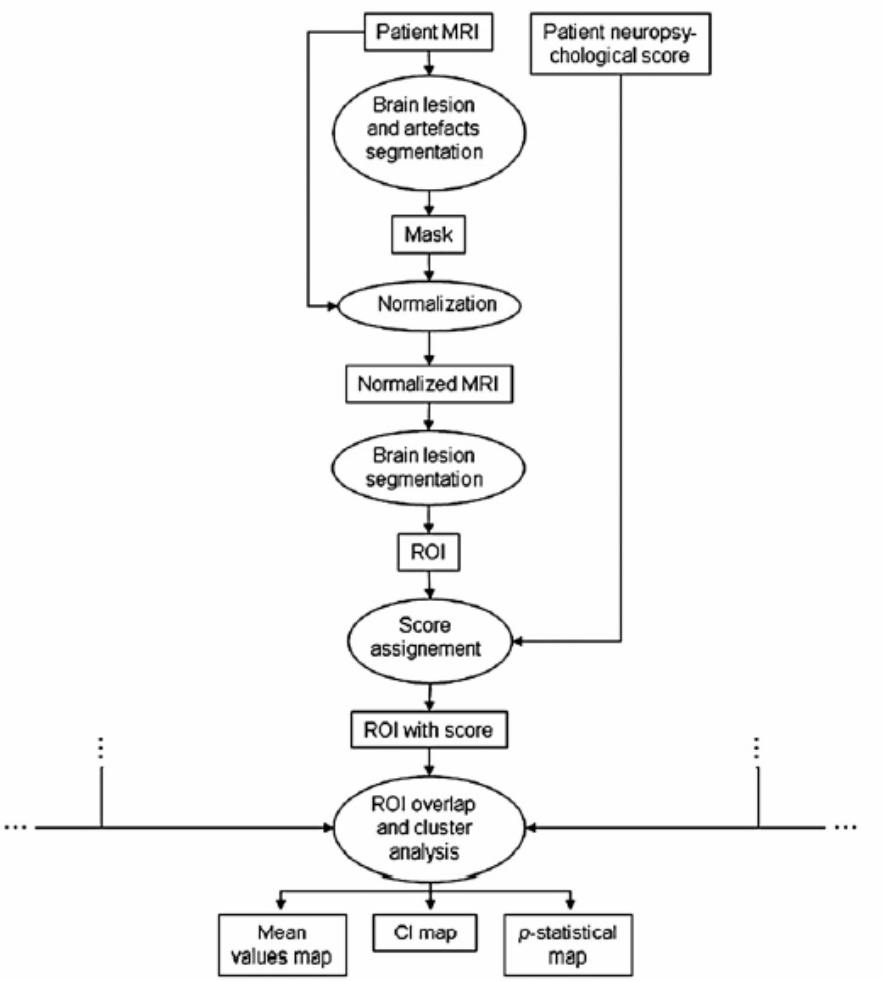

Figure 5: General flow chart diagram of the AnaCOM process

Data Processing: The two sites use the same tools, essentially coming from SPM (http://www.fil.ion.ucl.ac.uk.spm), BrainVISA (http://www.brainvisa.info) and R (http://www.r-project.org/).

1. image conversion (from Dicom to Analyze);
2. inter-modality registration of the set of images;
3. normalization of the images, for the use of a priori atlases (GIN) or for group studies (AnaCOM methodology). Normalization can require to mask the lesion;
4. skull striping (brain masking);
5. tissues and lesions automatic classification (GIN) or manual lesion segmentation (IFR49)
6. AnaCOM maps calculation (Figure 5)

The AnaCOM methodology combines these different tools and steps into an enhanced script. Brain lesions of the MRI scans are first manually segmented. The MRI volumes are then normalized to a reference map, using the segmented area as a mask. After normalization, the brain lesions of the MRI are segmented again in order to redefine the border of the lesions in the context of the normalized brain. Once the MRI is segmented, the patient's score on the neuropsychological test is assigned to each voxel in the lesioned area, while the rest of the voxels of the image are set to 0.

Subsequently, the individual patient's MRI images are superimposed, and each voxel is reassigned the average score of the patients who have a lesion at that voxel. A threshold is applied to remove regions having less than three overlaps. This process leads to an anatomo-functional map that links brain areas to functional loss.

6.1.3. Brain tumours

Rationale: MR is a powerful tool for tumour diagnosis and tumour follow-up. MRI allows the tumour localization and its volume determination. 1H MR spectroscopy allows characterizing the composition of the tumour in order to refine diagnosis and to avoid biopsy. The main questions are:

1. Which type of tumour has a patient?
2. Is the treatment (drugs or radiotherapy) adapted? and
3. Does an anatomo-functional relation exist between a specific anatomical localization of the tumour and functional deficits as assessed via specific neuro-psychological and behavioural tests?

Answering these questions requires the computerized management of a large amount of data. The central points are then:

- Which parameters can be extracted from the images reflecting the evolution of the disease?
- Which data processing chain is optimal for the extraction of these parameters?

Data providers: Three centres are considered: the Michallon Hospital in Grenoble (GIN partner), the Pitié-Salpêtrière Hospital

(IFR49 partner) in Paris and the Lacassagne Hospital (INRIA-Nice partner) in Nice. The three centres have clearly different objectives:

- precise investigation of MRI and 1H MR spectra for tumour classification and the setting of a “virtual biopsy” of the brain lesion (GIN);
- treatment planification and follow-up of patients after radiotherapy (INRIA-Sophia);
- follow-up of patients after chemotherapy (INRIA-Sophia) and design of anatomo-clinical maps after brain tumour resection (IFR49).

Sharing data is then not the main objective; however, various aspects of image processing and visualization can be mutualised between the three centres.

At the Centre Antoine Lacassagne in Nice, the purpose of the image acquisition is two-fold: images are used in first intention to optimize the plan of the radiotherapy treatment. In second intention, images are used to evaluate the evolution of the tumour. The planning step for conformal radiotherapy requires the accurate localisation of the tumour, to maximise its irradiation, and of the critical structures where the irradiation has to be minimised.

Data Description: In each centre the common collected images are: 1) a T2-weighted image, 3) a T1-weighted image, and 3) a T1-weighted image after gadolinium injection. 1H MR spectra are also acquired at GIN. Metadata about the patient, the medication, the pathology and the conditions of image acquisition are also provided. Presently, a local database is available at GIN and IFR 49 (MySQL). Ten patients are presently available for GIN. Acquired data are in Dicom (all partners) or Philips (GIN) format.

At Sophia, a database of 29 patients with 2 to 6 time points is available with T2-weighted, T1-weighted and gadolinium injected T1-weighted images acquired on a 1.5 Genesis Signa MR scanner. About 70 more patients have only one time point. As most of the image processing system is already integrated in a commercial tool,

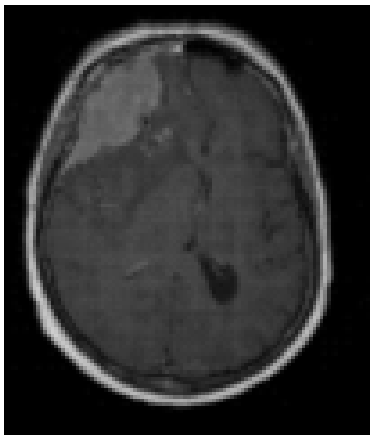

Figure 6: Tumour lesion on a T1-weighted image

only the data and part of the image processing algorithms are available for the NeuroLOG project.

Data Processing: The three sites use the following tools for image processing:

1. inter-modality registration of the set of images;
2. spatial normalisation of the images, for the use of a priori atlases for segmentation or for group studies (AnaCOM methodology), or conversely specialization of the generic atlas to the patient images (INRIA-Sophia). Normalization can require to mask the lesion;
3. skull striping (brain masking) ;
4. manual lesion segmentation (IFR49), or automatic segmentation of the structures at risk (INRIA-sophia). In Nice, segmentation of the tumour itself (gross tumour volume) and of associated clinical targets and planning target volumes is performed manually by the radiotherapist on multi-modal images as this step requires a deep expertise and carries out a huge responsibility. The critical structures are automatically segmented by registering a previously labelled atlas to the patient images. This segmentation can then be used directly, or as an initialisation for a more complex segmentation algorithm [7,8]. In such a system, the main difficulty is to obtain an inter-subject registration algorithm which is accurate enough and, more importantly, robust to the anatomical variability and to the pathologies (tumours may be quite important). For GIN, the characterization of lesions based on MR spectra, could be improved by the use of an automatic segmentation (not already available) of tissues and lesion;
5. AnaCOM maps calculation (IFR49).

8.2.4 Summary

Clearly, the needs for data processing are similar for each application. The ideal chain combines:

1. image conversion (from Dicom to GIS or INR or Analyze format) and data anonymisation;
2. inter-modality registration of the set of images (with different pre-processing steps: bias correction, denoising ...);
3. spatial normalisation of the images (for the use of spatial a priori);
4. skull striping (brain masking) ;
5. tissues and lesions classification.

For each step of this chain tools are available at each centre. They can be evaluated and compared to define the optimal chain for a given application. Note for instance that the segmentation tool LOCUS [2] developed at GIN for structural image can be also a good candidate for the bias field correction used in the MS

processing pipeline. It can also be tested on T2-weighted and FLAIR images coming from IFR49 and INRIA-Rennes. Once structural and additional data are available at GIN, AnaCOM methodology developed at IFR 49 can be used on GIN data. Segmentation and realignment tools developed at INRIA-Rennes and Nice can be in turn used on GIN and IFR49 data.

User queries are of the same nature for each application at the individual or group level:

1. searching for lesions (number, volume)
2. searching for tissue atrophy (% of variation)
3. information of localization (Talairach, Brodmann, vascular territories or anatomo-functional parcelling)

Visualization requirements are mainly:

1. visualisation of many slices in one modality (by default: 4)
2. visualisation of a slice in different modalities (by default: 4)
3. visualisation of a slice in one modality at different time points
4. visualisation of the segmented lesion superimposed on an image
5. 2D/3D visualisation
6. thresholding facilities for AnaCOM maps.

A discussion has been started with the Visioscopie partner to orient their developments towards these specifications. Example datasets have been shared for preliminary tests.

To conclude, the detailed study of each application shows that an optimal chain for each application can be designed based on the complementarities of the tools available at each site. Moreover, MS and Stroke data can also be shared to define groups of subjects larger than those initially available at the level of each site.

7. Bibliography

[1] S.R. Kinkingnéhun, E. Volle, M. Pélégrini-Issac, J.-L. Golmard, S. Lehéricy, F. du Boisguéheneuc, S. Zhang-Nunes, D. Sosson, H. Duffau, Y. Samson, R. Lévy, B. Dubois, *A novel approach to clinical-radiological correlations: Anatomo-Clinical Overlapping Maps (AnaCOM): method and validation* . **Neuroimage, 2007** (sous presse).

[2] Scherrer B, Dojat M, Forbes F and Garbay C. LOCUS: LOcal Cooperative Unified Segmentation of MRI Brain Scans. In: N. Ayache and N. Oursselin, eds., MICCAI 2007 (Proceedings of the MICCAI conference, Brisbane AU). Springer-Verlag, Berlin, 2007: to appear.

[3] Shamir, A. “How to share a secret”, Communications of the ACM 22:612-613, 1979.

[4] S. Burke, S. Campana, A. Delgado Peris, F. Donno, P. Méndez Lorenzo, R. Santinelli, A. Sciabà. “gLite 3 User Guide”. https://edms.cern.ch/file/722398/gLite-3-UserGuide.pdf.

[5] F. Pacini, “EGEE User’s Guide – WMProxy service”. https://edms.cern.ch/document/674643/ .

[6] N. Capit, G. Da Costa, Y. Georgiou, G. Huard, C. Martin, G. Mounié, P. Neyron, and O. Richard. “A batch scheduler with high level components”. In Cluster computing and Grid, 2005 (CCGrid'05). http://oar.imag.fr.

[7] Commowick O., Stefanescu R., Fillard P., Arsigny V., Ayache N., Pennec X., and Malandain G. Incorporating Statistical Measures of Anatomical Variability in Atlas-to-Subject Registration for Conformal Brain Radiotherapy. In J. Duncan and G. Gerig, editors, Proceedings of MICCAI 2005, volume 3750 of LNCS, pages 927-934, 2005. Springer Verlag

[8] Commowick O., Design and Use of Anatomical Atlases for Radiotherapy. PhD Thesis, Nice -- Sophia-Antipolis University, February 2007

[9] Temal L., Lando P., Dojat M., Fürst F., Gibaud B., Kassel G., Lapujade A. OntoNeuroLOG : une ontologie modulaire et multi-niveaux pour gérer l’hétérogénéité sémantique des métadonnées. Actes de la journée thématique « Ontologies et Gestion de l’hétérogénéité sémantique » du GDR I3, Grenoble, 3 juillet 2007.

[10] Grenon P. BFO in a nutshell: a bi-categorial axiomatization of BFO and comparison with DOLCE, IFOMIS Report, Universität Leipzig, ISSN 1611-4019, 37 pages; 2003.

[11] Gangemi A, Borgo S, editor. Proceedings of the EKAW*04 Workshop on Core Ontologies in Ontology Engineering. Northamptonshire (UK), October 8, 2004. http://ceur-ws.org(Vol-118).

[12] X.509 certificates. WikiPedia: http://en.wikipedia.org/wiki/X509.

[13] Alfieri R., Cecchini R., Ciaschini V., dell'Agnello L., Frohner A., Gianoli A., Lörentey K. and Spataro F. “VOMS, an Authorization System for Virtual Organizations”,in *European Across Grids Conference (EAGC)*, 2003.

[14] Shibboleth web site: http://shibboleth.internet2.edu/. Wikipedia: http://en.wikipedia.org/wiki/Shibboleth_(Internet2).

[15] Advanced Encryption Standard (AES). Wikipedia: http://en.wikipedia.org/wiki/Advanced_Encryption_Standard.

[16] Ferraiolo D. and Kuhn, D. “Role Based Access Control”, in *NIST-NCSC National Computer Security Conference*, pp 554-563, 1992.

[17] Rey D.. *Détection et quantification de processus évolutifs dans des images médicales tridimensionnelles : application à la sclérose en plaques*. Thèse de sciences, Nice Sophia-Antipolis university , October 2002

[18] Dugas-Phocion G. *Segmentation d'IRM Cérébrales Multi-Séquences et Application à la Sclérose en Plaques*. PhD Thesis, École des Mines de Paris, March 2006