Contents

[1 e-parliament Program - extract from the Program Charter 4](#_Toc466454509)

[1.1 Program overview 4](#_Toc466454510)

[1.2 Business objectives 4](#_Toc466454511)

[2 Scope and Architecture 6](#_Toc466454512)

[2.1 Functional scope 6](#_Toc466454513)

[2.2 Architecture 9](#_Toc466454514)

[2.2.1 AT4AM - Authoring Tool for AMendments 10](#_Toc466454515)

[2.2.2 AT4LEX - Authoring Tool for parliamentary texts 10](#_Toc466454516)

[2.2.3 DST - Drafting Support Tool 10](#_Toc466454517)

[2.2.4 CAT4TRAD - Computer-aided translation tool 10](#_Toc466454518)

[2.2.5 XML4EP - XML for EP contents 10](#_Toc466454519)

[2.2.6 DM-XML - XML Document Management 10](#_Toc466454520)

[2.2.7 PURE-XML - Parliament Unique REpository for XML text 11](#_Toc466454521)

[2.2.8 DiSP - Digital Signature 11](#_Toc466454522)

[2.2.9 EPS - e-Parliament Services 11](#_Toc466454523)

[ANNEX - e-Parliament components 12](#_Toc466454524)

[3 PURE-XML 12](#_Toc466454525)

[1.1. Basic information 12](#_Toc466454526)

[1.2. Network Architecture Schema 12](#_Toc466454527)

[1.3. Required Standard Runtimes 14](#_Toc466454528)

[1.4. Non-standard server-side softwares list 14](#_Toc466454529)

[1.5. Integration 15](#_Toc466454530)

[1.5.1. Applications integration schema 15](#_Toc466454531)

[1.5.2. Application integration criticality 15](#_Toc466454532)

[1.5.3. System 16](#_Toc466454533)

[1.5.3.1. List of required system services 16](#_Toc466454534)

[1.5.3.2. System services schema (optional) 16](#_Toc466454535)

[1.6. Security 16](#_Toc466454536)

[1.6.1. Authentication mechanism 16](#_Toc466454537)

[1.6.2. Authorization mechanism 16](#_Toc466454538)

[1.6.3. Specific confidentiality needs 17](#_Toc466454539)

[1.7. High availability 17](#_Toc466454540)

[1.8. Specific Development Tools 17](#_Toc466454541)

[4 PURE-XML Manager 18](#_Toc466454542)

[4.1 Basic information 18](#_Toc466454543)

[4.2 Network Architecture Schema 18](#_Toc466454544)

[4.3 Software Architecture 19](#_Toc466454545)

[4.4 Required Standard Runtimes 20](#_Toc466454546)

[4.5 Non-standard server-side softwares list 20](#_Toc466454547)

[4.6 Integration 20](#_Toc466454548)

[4.6.1 Applications integration schema 20](#_Toc466454549)

[4.6.2 Application integration criticality 21](#_Toc466454550)

[4.6.3 System 21](#_Toc466454551)

[4.6.4 Security 21](#_Toc466454552)

[4.6.5 Authentication mechanism 21](#_Toc466454553)

[4.6.6 Authorization mechanism 22](#_Toc466454554)

[4.6.7 Specific confidentiality needs 22](#_Toc466454555)

[4.7 High availability 22](#_Toc466454556)

[4.8 Specific Development Tools 22](#_Toc466454557)

[5 DM-XML 23](#_Toc466454558)

[5.1 Basic information 23](#_Toc466454559)

[5.2 Network Architecture Schema 23](#_Toc466454560)

[5.3 Software Architecture 24](#_Toc466454561)

[5.4 Required Standard Runtimes 24](#_Toc466454562)

[5.5 Non-standard server-side softwares list 25](#_Toc466454563)

[5.6 Integration 25](#_Toc466454564)

[5.6.1 Applications integration schema 25](#_Toc466454565)

[5.6.2 Application integration criticality 26](#_Toc466454566)

[5.6.3 System 26](#_Toc466454567)

[5.7 Security 27](#_Toc466454568)

[5.7.1 Authentication mechanism 27](#_Toc466454569)

[5.7.2 Authorization mechanism 27](#_Toc466454570)

[5.7.3 Specific confidentiality needs 27](#_Toc466454571)

[5.8 Specific Development Tools 27](#_Toc466454572)

[6 EPS 28](#_Toc466454573)

[6.1 Basic information 28](#_Toc466454574)

[6.2 Network Architecture Schema 28](#_Toc466454575)

[6.3 Software Architecture 29](#_Toc466454576)

[6.4 Required Standard Runtimes 30](#_Toc466454577)

[6.5 Non-standard server-side softwares list 31](#_Toc466454578)

[6.6 Integration 32](#_Toc466454579)

[6.6.1 Applications integration schema 32](#_Toc466454580)

[6.6.2 Application integration criticality 33](#_Toc466454581)

[6.6.3 System 33](#_Toc466454582)

[6.7 Security 34](#_Toc466454583)

[6.7.1 Authentication mechanism 34](#_Toc466454584)

[6.7.2 Authorization mechanism 34](#_Toc466454585)

[6.7.3 Specific confidentiality needs 34](#_Toc466454586)

[6.8 Specific Development Tools 34](#_Toc466454587)

[7 AT4LEX 35](#_Toc466454588)

[7.1 Basic information 35](#_Toc466454589)

[7.2 Network Architecture Schema 35](#_Toc466454590)

[7.3 Software Architecture 36](#_Toc466454591)

[7.4 Required Standard Runtimes 38](#_Toc466454592)

[1.1. Non-standard server-side softwares list 38](#_Toc466454593)

[7.5 Integration 39](#_Toc466454594)

[7.5.1 Applications integration schema 39](#_Toc466454595)

[7.5.2 Application integration criticality 40](#_Toc466454596)

[7.6 System 41](#_Toc466454597)

[7.6.1 List of required system services 41](#_Toc466454598)

[7.6.2 System services schema (optional) 41](#_Toc466454599)

[7.7 Security 41](#_Toc466454600)

[7.7.1 Authentication mechanism 41](#_Toc466454601)

[7.7.2 Authorization mechanism 41](#_Toc466454602)

[7.7.3 Specific confidentiality needs 41](#_Toc466454603)

[7.8 Specific Development Tools 42](#_Toc466454604)

# e-parliament Program - extract from the Program Charter

## Program overview

e‑Parliament is designed as a major multi-annual change program aimed at modernising the EP's core parliamentary information system. To control the changes and risks introduced by the program, e-Parliament progressively achieves its objectives by implementing the following two phases:

1. Phase 1 is focused on parliamentary text management and may introduce changes in related processes (Business/IT alignment). It deals with the nature of the product (text) and its control in terms of life cycle and versioning.
2. Phase 2 will be activity/process oriented and information provision. It deals with the optimization of text production processes and the workflows that support them. This phase is not described in this document.

Each phase is divided into two steps:

* Setup of foundation elements and first use; this step is managed under program framework PPO4EP version 1.0.
* Alignment of all relevant existing applications to foundation elements; this step will be managed as IT projects portfolio management. The alignments are out of the scope of e-Parliament program.

This Program Charter addresses the first phase and first step of e-Parliament vision, namely the setup of the foundation elements and their first use in the context of the parliamentary text management.

## Business objectives

DG Innovation and Technological Support is proposing to revamp existing information circuits by developing a new integrated architecture. The business objectives are derived from the *Administrative Work Programme*[[1]](#footnote-1):

* Business objective 1: facilitate document creation.
* Business objective 2: facilitate document verification.
* Business objective 3: minimise input errors.
* Business objective 4: make texts easier to reuse for translation.
* Business objective 5: make texts easier to reuse for publication on paper.
* Business objective 6: make texts easier to reuse for electronic dissemination Internet.
* Business objective 7: make texts easier to reuse for electronic dissemination Intranet.
* Business objective 8: improve the quality of texts produced.
* Business objective 9: reduce costs over the life-cycle of documents.
* Business objective 10: provide electronic signature.

# Scope and Architecture

## Functional scope

The functional scope of this phase is divided into text production chains. A chain is a set of processes, actors and tools across a range of DGs linked together to produce a collection of texts, the current identified text production chains are:

* Reports & Amendments.
* Committees Agenda.
* Plenary Verbatim.
* Parliamentary Questions.
* Written Declarations.
* Plenary Minutes.
* Committees Minutes.
* Plenary Agenda.

The first phase of the program deals with the definition of the e-Parliament architecture for the parliamentary text management: the foundation elements will be created by progressively building new components and adapting legacy applications to the new XML-based architecture.

The figure below illustrates the approach chosen to transform the EP’s core information system.

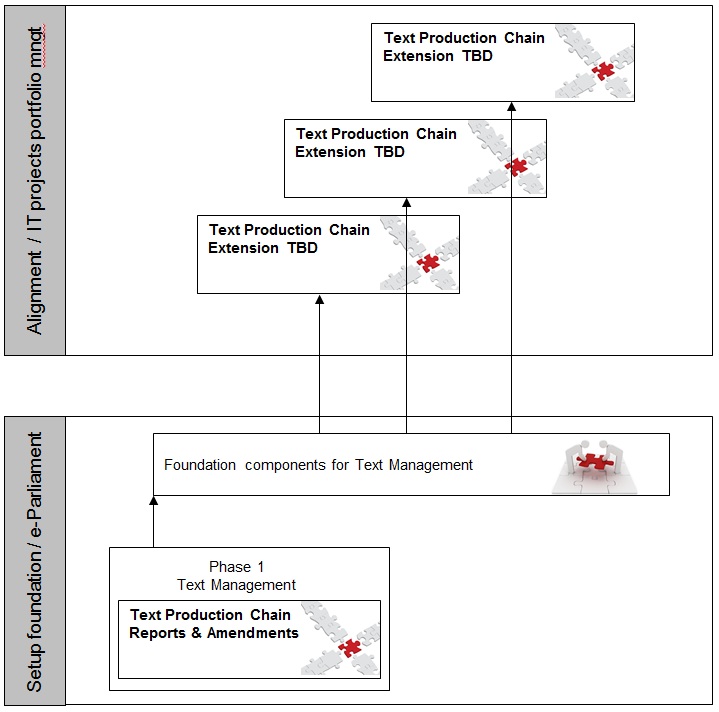


Figure 1 – Scope management approach

For this phase, the program treats the Reports & Amendments production chain as a pilot on which the foundation components of the e-Parliament architecture will be built.

Once the foundation components are operational for the pilot chain, further text production chains will be aligned to the e-Parliament architecture after completion of the first phase of the program.

The alignment of the chains to the e-Parliament architecture is out of scope of the program as such. The application e-Parliament principles and foundations to other text production chains or information systems are considered as an extension of e-Parliament. The knowledge gained and principles developed by the program in terms of technical foundations, business logic and program organisation, during phase 1 of the program will be made available to further parliamentary text production chains for which maintenance activities and/or new projects will be launched.

The text production management has been divided into nine functional areas as summarised below:

1. XMLisation: the transformation of MS Word content into an XML format suitable for processing with e-Parliament tools.
2. Digital Signature: the use of digital signatures as an alternative to manual signatures where the Parliamentary rules prescribe that a text must be signed by Members of Parliament (e.g. legislative amendments, written declaration, etc.) on internal EP documents.
3. Drafting: creation of parliamentary (legislative and non-legislative) texts in XML Akoma Ntoso format.
4. Verification: collaboration on drafting amendments and linguistic verification of documents that were produced using an e-Parliament authoring tool.
5. Internal Translation: translation of XML Akoma Ntoso content by DG TRAD staff.
6. External Translation: the possibility to outsource the translation of XML Akoma Ntoso content assuming that the translated content respects the Akoma Ntoso format.
7. Consolidated Texts: the possibility to produce XML Akoma Ntoso text amending a base text in the context of a 1st reading agreement. The amendments are directly consolidated in these texts ("pre-adoption finalisation" and "consolidated amendment" documents).
8. Consolidated Texts (TC):   
   a. Consolidation of Final Reports, on the basis of the Draft Report, Opinions and voted amendments adopted in Committee.

b. Production of the Consolidated Text (TC), on the basis of the Adopted Act Text and the Commission's legislative proposal.

1. Adopted Texts (TA): production with e-Parliament tools and in XML Akoma Ntoso format of the Parliament's Adopted Text (TA).

## Architecture

Phase 1 of the program will create specific business components (AT4AM), re-usable business components (AT4LEX, DST, CAT4TRAD) as well as technical support services (XML4EP, DM-XML, PURE-XML, DiSP, EPS).

The goals in terms of the management of parliamentary texts are to introduce in particular:

* Apply the XML Akoma Ntoso standard to parliamentary texts (foundations XML4EP and DM-XML).
* Use a single central repository for content created during the entire lifecycle of the parliamentary text production chain (PURE-XML).
* Allow parliamentary texts to be signed digitally.
* Create reusable components and services that can be used for further applications (centralisation of processes in DM-XML, PURE-XML, DiSP, EPS).

The figure below summarises the architecture of the foundation components of the phase 1 and how they apply to document processes.

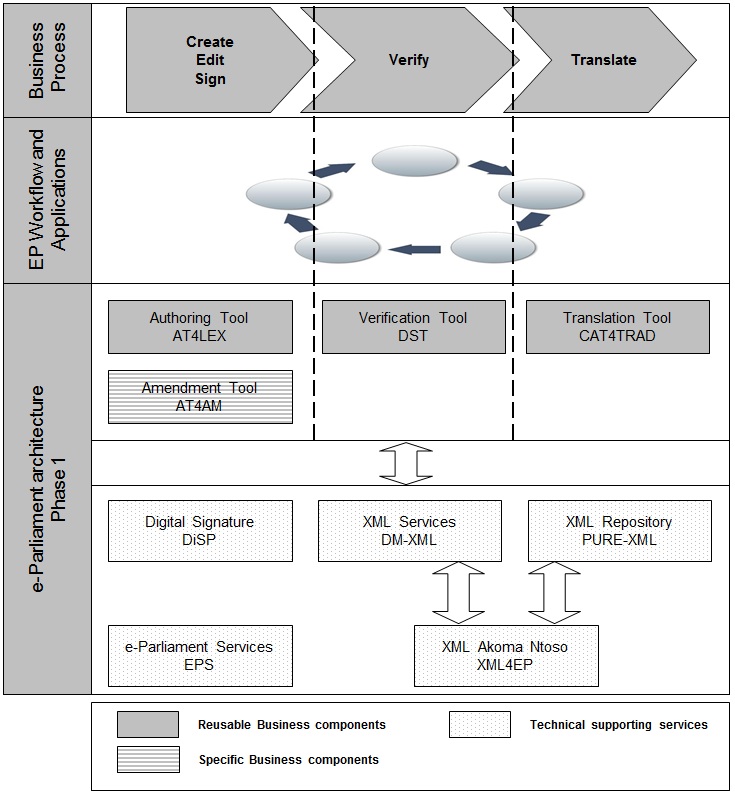


Figure 2 – Program architecture

Each component will produce a range of deliverables that are further detailed below.

### AT4AM - Authoring Tool for AMendments

AT4AM is the e-Parliament specific business component covering the authoring of amendments on parliamentary texts (legislative and non-legislative procedures) and the production of amendment's lists for downstream services.

This application relies on the e-Parliament technical supporting services.

### AT4LEX - Authoring Tool for parliamentary texts

AT4LEX is the e-Parliament reusable business component covering the authoring of parliamentary texts (legislative and non-legislative procedures). Supporting content assembly and re-use, this business component will enable MEPs and EP officials to focus on content rather than on document layout, and will also make documents downstream treatments more efficient.

This application relies on the e-Parliament technical supporting services.

### DST - Drafting Support Tool

DST is the e-Parliament reusable business component covering the verification of parliamentary texts (legislative and non-legislative procedures) produced by authoring tools aligned to the e‑Parliament architecture. This component supports the collaboration between authors and lawyer linguists on drafting amendments.

This application relies on the e-Parliament technical supporting services.

### CAT4TRAD - Computer-aided translation tool

CAT4TRAD[[2]](#footnote-2) is the e-Parliament reusable business component covering the translation of parliamentary texts (legislative and non-legislative procedures) produced by authoring tools aligned to the e‑Parliament architecture.

This application relies on the e-Parliament technical supporting services.

### XML4EP - XML for EP contents

XML4EP is the e-Parliament activity aiming to select an XML standard (Akoma Ntoso) for EP parliamentary texts, to specify accordingly the XML representation of EP content, to provide support to the delivery teams for content and document representation and to identify management needs for the XML content representation.

### DM-XML - XML Document Management

DM-XML is the e-Parliament technical supporting service providing a unique services layer for XML text handling to the applications compliant to the e‑Parliament architecture.

### PURE-XML - Parliament Unique REpository for XML text

PURE-XML is the e-Parliament technical supporting service providing a unique content repository to the applications compliant to the e-Parliament architecture and the IRO models.

### DiSP - Digital Signature

DiSP is the e-Parliament technical supporting service providing a portal to manage the digital signature in different documents.

### EPS - e-Parliament Services

EPS is the e-Parliament technical supporting service providing common services and libraries for the e-Parliament reusable business components and the applications integrated with the e-Parliament architecture.

ANNEX - e-Parliament components

# PURE-XML

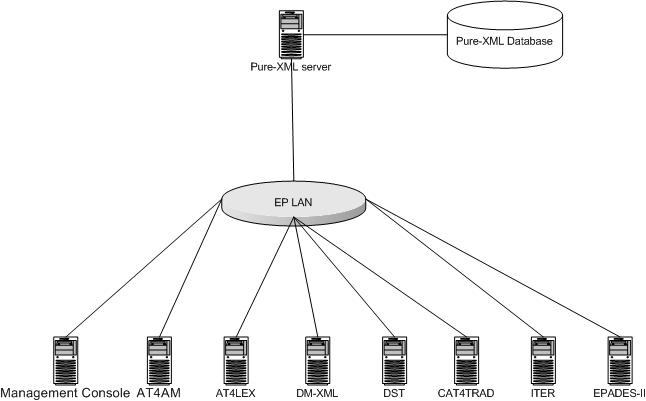
### Basic information

|  |  |
| --- | --- |
| Service | Required |
| Intranet | **Yes** |
| DMZ | **No** |
| VLAN dedicated | **No** |
| Internet access through the EP portal | **No** |
| Other | **No** |

### Network Architecture Schema

PURE-XML is the XML repository for the text production in the frame of the e-Parliament program. All the applications (including the management console) that will need to use the repository will have to access it via its API (Java binding or Web Services binding).





The PURE-XML application is composed of the following components:

* Repository connector
* Repository (Repository objects’ CRUD functions)
* Change log management
* Access Control management
* Relationship management
* Versioning management
* Content types management
* Tags management
* Locking management
* Notifications management

**List the technologies/libraries used.**

|  |  |
| --- | --- |
| Client User interface | - |
| Server Presentation layer | - |
| Server Business layer | Spring |
| Server Persistence Layer | JPA / HIBERNATE 4.1.1 Oracle 11gR2 with XML DB |
| Synchronous Remoting (WS, RMI) | Y (Spring HTTP Invoker, support for WS in the future) |
| Asynchronous Remoting (JMS) | HornetQ 2.2.14 |
| XML Handling | Standards Java APIs and Oracle 11gR2 with XML DB |

|  |  |
| --- | --- |
| Cache | EP Foundry Cache |
| Security | JAAS and Basic authentification |
| Framework | Spring 3.1.0.RELEASE |
| Software management tool | Apache Maven 3.0.4 |
| Application Server | Tomcat 6.0.x |
| Programming Language | Java 1.6.0\_31 |

### Required Standard Runtimes

List of required operational environment and associated 8 characters Code:

| **Existing**  **(Y/N)** | **Standard Runtime [[3]](#footnote-3)** | **Code**  **(8 char)** | **Environment(s) [[4]](#footnote-4)**  DEV, PPO, FOR, PROD, INT |
| --- | --- | --- | --- |
| N | Tomcat 6.0.x | PUREXML | DEV, INT, PPO, PROD |
| N | HornetQ 2.2.14 | PUREXML | DEV, INT, PPO, PROD |

Please note that one separate RHR document will be needed for each new runtime.

### Non-standard server-side softwares list

**Please note that this non-standard software are most likely not to be accepted quickly, though may trigger special attention and dedicated handling.**

**Fill in this chapter as accurately as possible.**

|  |  |
| --- | --- |
| **Non-Standard Server-side software** | **Explanation of the need** |
| - | - |

### Integration

### Applications integration schema



Client applications will be able to use the PURE-XML repository either via the Java binding or the Web services binding of its API. Asynchronous repository events will be published via JMS technology.

In the above diagram, the management console is a client application that uses the Java binding.

The following client applications have been identified in the frame of the e-Parliament program:

* Management Console
* AT4AM
* DST
* CAT4TRAD
* DM-XML
* AT4LEX
* DISP

All the above applications will access PURE-XML only via its API (Java bindings).

### Application integration criticality

| **Reference[[5]](#footnote-5)** | **Protocol** | **Synchronicity** | **Frequency[[6]](#footnote-6)** | **Business Criticality** |
| --- | --- | --- | --- | --- |
|  | **LDAP** | **Synchronous** | **High** | **High** |
|  | **HTTP** | **Synchronous** | **High** | **High** |
|  | **TCP** | **Asynchronous** | **High** | **High** |

The application is **mission critical** since it will serve content repository services for structured legislative texts created and managed in the context of the Parliament legislative processes.

### System

### List of required system services

|  |  |
| --- | --- |
| Database | Oracle 11gR2 with XML DB |
| File system (direct access, sharing, FTP, ...) | **Direct access** |
| Directory (Active Directory, other) | **Active Directory/LDAP** |
| Mails (Sending mails from an application) | **No** |
| Messaging Broker (JMS) | **HornetQ 2.2.14** |
| Application Server | **Tomcat 6.0.x** |
| JDBC | **Oracle OCI driver 11.2.0.3.0** |

### System services schema (optional)

For clarity (and mostly if you have asked for multiple runtimes), please provide a schematic vision of the interaction between system services and applications.

### Security

### Authentication mechanism

The client applications will provide username and user group that PURE-XML will accept as valid. Username and user group will not be validated against LDAP. Client applications will be authenticated by PURE-XML against LDAP.

### Authorization mechanism

The internal access control module is based on group membership of the authenticated user. PURE-XML will provide groups/permissions per authenticated user, which will allow a fine grained access control to specific parts of the application’s API and repository objects. User access will be limited on both functional level and repository object level.

### Specific confidentiality needs

None.

### High availability

The PURE-XML service is mission critical, therefore a high availability setup is recommended. In order to ensure high availability, the following additional services should be provided:

* Reverse proxy server
* RAID configuration, setup for high availability, for the HornetQ filesystem
* Oracle RAC. If in EP premises the database is set up to work on single servers, high availability on the database level is not guaranteed.

### Specific Development Tools

The ICTAS Unit provides standard tools for the development of applications.

Mention here licence-limited tools you may need ([restricted use product](http://www.ismsnet.ep.parl.union.eu/ispnet/cms/preconisations/ingenierie-logicielle/liste_produits)):

|  |  |
| --- | --- |
| Oxygen XML | Y |
| Modelling tool UML | Y |
| Maven | Y |
| Source management (SVN) | Y |
| Full Application server | Y |
|  |  |

**Non standard tools and justification:**

# PURE-XML Manager

## Basic information

This chapter describes the deliverables (applications, components, integration component) of the product in the EP's network using a synthetic schema.

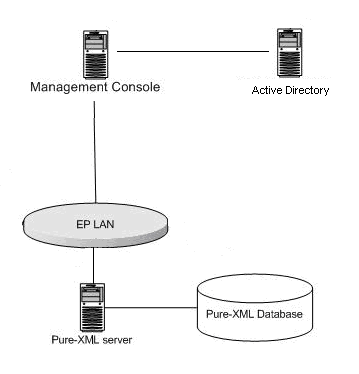
|  |  |
| --- | --- |
| **Service** | **Required** |
| Intranet | **Yes** |
| DMZ | **No** |
| VLAN dedicated | **No** |
| Internet access through the EP portal | **No** |
| Other | **No** |

## Network Architecture Schema

The PURE-XML management console is the web interface for the PURE-XML administrator. It connects to the PURE-XML server via the server's API (Java binding).

Insert a synthetic schema representing general software components in their respective network zones (application servers, applications, products...etc), as well as their interactions and the protocols used for these interactions.

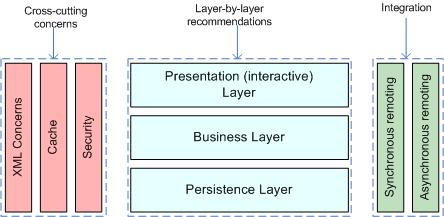




* The PURE-XML management console is a web interface for the administrator's PURE-XML API. The authentication of the administrator is done using the EP Active Directory.

## Software Architecture

This chapter describes the software architecture according to the following schema:



**List the technologies/libraries used.**

|  |  |
| --- | --- |
| **Client User interface** | **JQuery 1.8** |
| Server Presentation layer | Spring MVC |
| Server Business layer | - |
| Server Persistence Layer | - |
| Synchronous Remoting (WS, RMI) | Y (Spring HTTP Invoker, support for WS in the future) |
| Asynchronous Remoting (JMS) | - |
| XML Handling | - |
| Cache | Ehcache 2.5.1 |
| Security | JAAS and Basic authentification |
| Framework | Spring 3.1.0.RELEASE |
| Software management tool | Apache Maven 3.0.4 |
| Application Server | Tomcat 6.0.18 |
| Programming Language | Java 1.6.0\_31 |

## Required Standard Runtimes

This chapter lists the various required server-side softwares (for example, Tomcat, HornetQ...etc), and their runtime codes (allowing to map this document with multiple Runtime Hosting Request(RHR).

List of required operational environment and associated 8 characters Code:

| **Existing**  **(Y/N)** | **Standard Runtime [[7]](#footnote-7)** | **Code**  **(8 char)** | **Environment(s) [[8]](#footnote-8)**  DEV, PPO, FOR, PROD, INT |
| --- | --- | --- | --- |
| N | Tomcat 6.0.18 | PUREMGR | DEV, INT, PPO, PROD |

Please note that one separate RHR document will be needed for each new runtime.

## Non-standard server-side softwares list

This chapter lists the various required non-standard server-side softwares.

**Please note that this non-standard software are most likely not to be accepted quickly, though may trigger special attention and dedicated handling.**

**Fill in this chapter as accurately as possible.**

|  |  |
| --- | --- |
| **Non-Standard Server-side software** | **Explanation of the need** |
| - | - |

## Integration

This chapter describes integration of the product with other applications or services (Database, directory, etc.) and contracts and protocols used.

### Applications integration schema

This section has to contain a schema describing interactions between applications. You have to precise the protocols used, and information needed to describe the exchanges (synchronous (remoting, web services), asynchronous (JMS, etc.).

The PURE-XML Management Console will authenticate the user against the EP Active Directory and will then open a connection with PURE-XML communicating only the username and group.

The PURE-XML Management Console will communicate with the PURE-XML via its API.

### Application integration criticality

| **Reference[[9]](#footnote-9)** | **Protocol** | **Synchronicity** | **Frequency[[10]](#footnote-10)** | **Business Criticality** |
| --- | --- | --- | --- | --- |
|  | **LDAP** | **Synchronous** | **High** | **High** |
|  | **HTTP** | **Synchronous** | **High** | **High** |
|  | **TCP** | **Asynchronous** | **High** | **High** |

### System

#### List of required system services

|  |  |
| --- | --- |
| **Database** | **N/A** |
| File system (direct access, sharing, FTP, ...) | **Direct access** |
| Directory (Active Directory, other) | **Active Directory/LDAP** |
| Mails (Sending mails from an application) | **No** |
| Messaging Broker (JMS) | **N/A** |
| Application Server | **Tomcat 6.0.18** |
| JDBC | **N/A** |

#### System services schema (optional)

For clarity (and mostly if you have asked for multiple runtimes), please provide a schematic vision of the interaction between system services and applications.

### Security

This chapter describes the needs for security (secure client access, integration, confidentiality (signature) data exchanged or stored, business services implemented, etc.).

### Authentication mechanism

Standard EP authentication, using the EPActive Directory.

### Authorization mechanism

The authorization will be provided based on the stored information in PURE-XML.

### Specific confidentiality needs

None.

## High availability

No special requirements.

## Specific Development Tools

The purpose is to mention only components which need a special or specific attention

The ICTAS Unit provides standard tools for the development of applications.

Mention here licence-limited tools you may need ([restricted use product](http://www.ismsnet.ep.parl.union.eu/ispnet/cms/preconisations/ingenierie-logicielle/liste_produits)):

|  |  |
| --- | --- |
| **Oxygen XML** | **Y** |
| Modelling tool UML | Y |
| Maven | Y |
| Source management (SVN) | Y |
| Full Application server | Y |
|  |  |

**Non standard tools and justification:**

# DM-XML

## Basic information

|  |  |
| --- | --- |
| **Intranet** | **Y** |
| DMZ | **N** |
| VLAN dedicated | **N** |
| Internet access through the EP portal | **N** |
| Other | **N** |

## Network Architecture Schema

DM-XML is the application that performs all the transformations and pre-translations of XML data for the text production in the frame of the e-Parliament program. All the applications that will need to use the DM-XML services will have to access it via its API (Java binding or Web Services binding). All the services offered by DM-XML are stateless. The figure below depicts how the DM-XML services make use of other services (CODICT, PURE-XML, ITER) and provides its services to its clients (CAT4TRAD, DST, AT4AM).

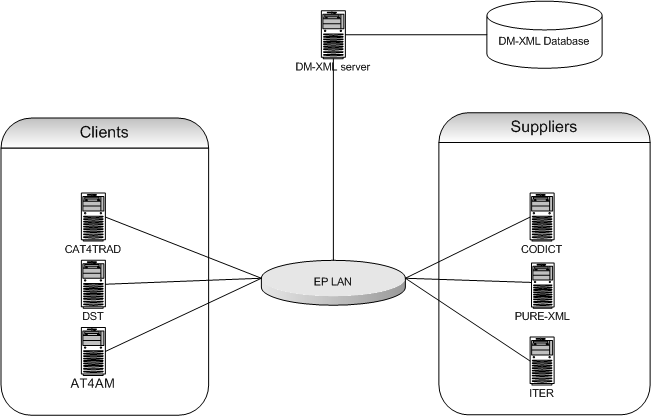


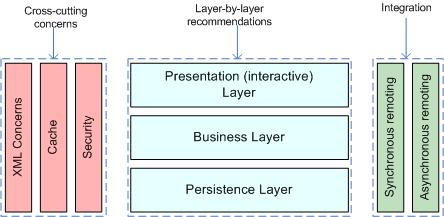
Figure 1: Network architecture diagram

The network topology complies to the regular J2EE Intranet network architecture:

* + DM-XML Web application is hosted on an in-house J2EE server (Tomcat) and accessed in the Intranet via HTTP protocol
  + Authentication will not be implemented
  + Role-based authorisation will be implemented, based on group membership
  + The persistence layer is assured by in-house Oracle databases

The DM-XML application is composed of the following components:

## Software Architecture



**List the technologies/libraries used.**

|  |  |
| --- | --- |
| **Client User interface** | **N** |
| Server Presentation layer | N |
| Server Business layer | Spring 3.1.3 (latest production release) |
| Server Persistence Layer | Spring Data 1.1/JPA / HIBERNATE 4.1.7 |
| RDB | Oracle 10gR2 |
| Synchronous Remoting (WS, RMI...) | Spring MVC 3.1.x for RESTful Web Services |
| Asynchronous Remoting (JMS...) | N |
| XML Handling | JAXP / StAX / TrAX |
| Cache | Ehcache 2.6.2 |
| Security | Authentication not implemented in the current scope. Role-based authorisation will be implemented, with custom Spring-based declarative authorization. |
| XSLT Processor | Saxon-HE 9.3 |
| XSLT 2.0[[11]](#footnote-11) | Saxon-HE 9.3 |
| Other | Jackson 1.9.11 (latest official release) |

Document generation: Aspose.Word for JAVA

## Required Standard Runtimes

List of required operational environment and associated 8 characters Code:

|  |  |  |  |
| --- | --- | --- | --- |
| **Existing**  **(Y/N)** | **Standard Runtime [[12]](#footnote-12)** | **Code**  **(8 char)** | **Environment(s) [[13]](#footnote-13)**  DEV, PPO, FOR, PROD, INT |
| N | Tomcat 6.0.18, Java 1.6.0\_31 | DMXMLDV | DEV |
| N | Tomcat 6.0.18, Java 1.6.0\_31 | DMXMLIT | INT |
| N | Tomcat 6.0.18, Java 1.6.0\_31 | DMXMLPP | PPO |
| N | Tomcat 6.0.18, Java 1.6.0\_31 | DMXMLPR | PROD |

Please note that one separate RHR document will be needed for each new runtime.

## Non-standard server-side softwares list

**Please note that these non-standard pieces of software are most likely not to be accepted quickly, though may trigger special attention and dedicated handling.**

**Fill in this chapter as accurately as possible.**

|  |  |
| --- | --- |
| **Non-Standard Server-side software** | **Explanation of the need** |
| Saxon-HE 9.3 | Saxon supports XSLT 2.0, unlike Xalan that only supports XSLT 1.0. Furthermore, the compiled version of Saxon with the TinyTree object model has very good results in performance benchmarks. |
| Jackson 1.9.11 | JSON string manipulation needed to generate the Aspose input. Jackson is the richest and most widely used JSON library in Java. |

## Integration

### Applications integration schema

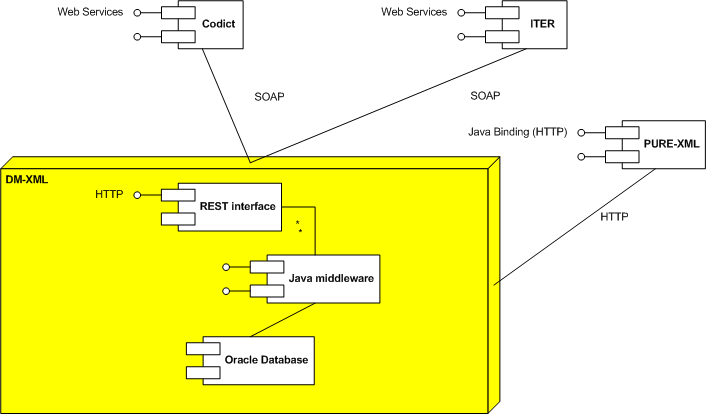


Figure 2: Applications integration schema

Several e-Parliament projects will call DM-XML stateless services:

- AT4AM

- DST

- CAT4TRAD

DM-XML will consult some applications using stateless services:

- CODICT (CSC)

- ITER-WS

### Application integration criticality

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Reference[[14]](#footnote-14)** | **Protocol** | **Synchronicity** | **Frequency[[15]](#footnote-15)** | **Business Criticality** |
| **PURE-XML** | **HTTP** | **Synchronous** | **>1000/min** | **High** |
| **CODICT** | **WS** | **Synchronous** | **>1000/min** | **High** |
| **ITER-WS** | **WS** | **Synchronous** | **< 10/min** | **Med** |

If the application cannot contact PURE-XML, some resources requested by the web service clients will not be accessible.

If CODICT services layer is not available, the results of some services will be poor.

Without ITER-WS, the init service of DM-XML will not be able to get the title of the document.

### System

#### List of required system services

|  |  |
| --- | --- |
| **Database** | **Oracle 10.2.0.5** |
| File system (direct access, sharing, FTP, ...) | Direct access |
| Directory (Active Directory, other) | N |
| Mails (Sending mails from an application) | N |
| Application Server | **EP Tomcat 6.0.18** |
| JDBC | **Oracle OCI driver 10.2.0.3.0** |
| Other | N |

#### System services schema (optional)

For clarity (and mostly if you have asked for multiple runtimes), please provide a schematic vision of the interaction between system services and applications.

## Security

### Authentication mechanism

DM-XML will be a service layer which will expose only WebServices. No authentication will be implemented. The services will trust the calling applications. Each webservice invocation will receive the user which was already authenticated by the calling application.

### Authorization mechanism

The internal access control module is based on group membership of the already authenticated user by the calling application.

DM-XML will keep a record of calling applications and the services they are permitted to access respectively.

### Specific confidentiality needs

None

## Specific Development Tools

The ICTAS Unit provides standard tools for the development of applications.

Mention here licence-limited tools you may need ([restricted use product](http://www.ismsnet.ep.parl.union.eu/ispnet/cms/preconisations/ingenierie-logicielle/liste_produits)):

|  |  |
| --- | --- |
| Oxygen XML | Y |
| Modelling tool UML | Y |

**Non standard tools and justification:**

# EPS

## Basic information

|  |  |
| --- | --- |
| **Service** | **Required** |
| Intranet | **Y** |
| DMZ | **N** |
| VLAN dedicated | **N** |
| Internet access through the EP portal | **N** |
| Other | **N** |

## Network Architecture Schema

e-Parliament Services will be a set of services or libraries that will be commonly used by different projects/products in the context of the e-Parliament Program.

- "PURE-XML Service Layer" (PSL): this module will provide products with a service layer to address the PURE-XML repository

- "TOPOLINO" (TPL): this module is a listener to EPADES 2 circuits that will trigger actions to consume a step and provide information to the applications plugged to the next step

- "XML4EP lib" is a library to work with XML4EP format

All the services offered by EPS are stateless.

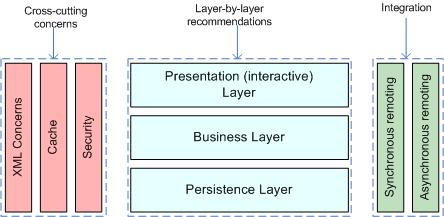
These services will be accessed it via its API (Java binding or Web Services binding) or using JMS.

The network topology complies with the regular J2EE Intranet network architecture:

* + EPS application is hosted on an in-house J2EE server (Tomcat) and accessed in the Intranet via HTTP protocol
  + Authentication will not be implemented.
  + No authorisation will be implemented.
  + The persistence layer is assured by in-house Oracle databases



## Software Architecture



**List the technologies/libraries used.**

|  |  |
| --- | --- |
| **Client User interface** | **N** |
| Server Presentation layer | N |
| Server Business layer | Spring 3.1.3 (latest production release) |
| Server Persistence Layer | Spring Data 1.1/JPA / HIBERNATE 4.1.7 |
| RDB | Oracle (latest official release) |
| Synchronous Remoting (WS, RMI...) | Spring MVC 3.1.x for RESTful Web Services |
| Asynchronous Remoting (JMS...) | JMS with HornetQ 2.2.X client |
| XML Handling | JAXP / StAX / TrAX |
| Cache | Ehcache 2.6.2 |
| Security | Authentication not implemented in the current scope. Authorisation not implemented in the current scope. |
| XSLT Processor | Saxon-HE 9.3 |
| XSLT 2.0 | Saxon-HE 9.3 |
| Other | Jackson 1.9.11 (latest official release) |

Document generation: Aspose.Word for JAVA

The deployments will be done using two war files.

Each war will have differect lifecycle.

First wars is for EPSPSL (EPS for Pure-XML Service Layer, including the XML4EP lib). This will be deployed on the EPSPSL Tomcat instance.

Second war is for EPSTPL.

This will be deployed on the EPSTPL Tomcat instance.

## Required Standard Runtimes

List of required operational environment and associated 8 characters Code:

|  |  |  |  |
| --- | --- | --- | --- |
| **Existing**  **(Y/N)** | **Standard Runtime [[16]](#footnote-16)** | **Code**  **(8 char)** | **Environment(s) [[17]](#footnote-17)**  DEV, PPO, FOR, PROD, INT |
| N | Tomcat 6.0.x | EPSPSL | DEV, INT, PPO, PROD |
| N | HornetQ 2.2.x | EPSPSLQ | DEV, INT, PPO, PROD |
| N | Tomcat 6.0.x | EPSTPL | DEV, INT, PPO, PROD |

EPS system needs two Tomcat instances for performance reasons: EPSPSL is an intensive module that needs its own instance.

First Tomcat Server (EPSPSL) will connect to the Oracle Database.

Second Tomcat Server (EPSTPL) doesn't require connection to database.

Please note that one separate RHR document will be needed for each new runtime.

## Non-standard server-side softwares list

**Please note that these non-standard pieces of software are most likely not to be accepted quickly, though may trigger special attention and dedicated handling.**

**Fill in this chapter as accurately as possible.**

|  |  |
| --- | --- |
| **Non-Standard Server-side software** | **Explanation of the need** |
| Saxon-HE 9.3 | Saxon supports XSLT 2.0, unlike Xalan that only supports XSLT 1.0. |
| Jackson 1.9.11 | Jackson is the richest and most widely used JSON library in Java. |

## Integration

### Applications integration schema

Several e-Parliament projects will call EPS stateless services:

- AT4AM

- DST

- CAT4TRAD

- DM-XML

EPS will call some applications using stateless services:

- PURE-XML

- DM-XML

- FILEBEANS

- UDI

EPS will call some applications using statefull services:

- EPADES2

EPS will listen to some applications JMS topics/queues:

- EPADES2



### Application integration criticality

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Reference[[18]](#footnote-18)** | **Protocol** | **Synchronicity** | **Frequency[[19]](#footnote-19)** | **Business Criticality** |
| **AT4AM** | **HTTP** | **Synchronous** | **<100/min** | **High** |
| **DST** | **HTTP** | **Synchronous** | **<100/min** | **High** |
| **CAT4TRAD** | **HTTP** | **Synchronous** | **<100/min** | **High** |
| **DM-XML** | **HTTP** | **Synchronous** | **<100/min** | **High** |
| **PURE-XML** | **HTTP** | **Synchronous** | **>1000/min** | **High** |
| **EPADES2** | **EJB** | **Synchronous** | **<100/min** | **High** |
| **EPADES2** | **JMS** | **Asynchronous** | **<100/min** | **High** |
| **UDI** | **SOAP** | **Synchronous** | **<100/min** | **High** |
| **FileBeans** | **SOAP** | **Synchronous** | **<100/min** | **High** |

AT4AM, DST and CAT4TRAD need EPS - PURE-XML Service Layer to store XML in PURE-XML.

EPS has no direct dependency to PURE-XML. If PURE-XML is down, EPS will handle gracefully the unavailability at the business layer and it will reconnect to PURE-XML once the external system is up and running (after a shut down, restart, update or error).

There is no dependency from PURE-XML towards EPS. If EPS is down PURE-XML will function without any impediment.

DM-XML is needed to generate PDF or Word versions of XML-Document, which is need in the context of e-Parliament, and triggered by EPS, consuming EPADES2 steps.

### System

#### List of required system services

|  |  |
| --- | --- |
| Database | Oracle 10.2.0.5 |
| File system (direct access, sharing, FTP, ...) | Direct access |
| Directory (Active Directory, other) | N |
| Mails (Sending mails from an application) | N |
| Application Server | EP Tomcat 6.0.18 |
| JDBC | Oracle OCI driver 10.2.0.3.0 |
| Other | N |

#### System services schema (optional)

For clarity (and mostly if you have asked for multiple runtimes), please provide a schematic vision of the interaction between system services and applications.

## Security

### Authentication mechanism

No authentication foreseen at this stage.

### Authorization mechanism

No authorisation foreseen at this stage.

### Specific confidentiality needs

None

## Specific Development Tools

The ICTAS Unit provides standard tools for the development of applications.

Mention here licence-limited tools you may need ([restricted use product](http://www.ismsnet.ep.parl.union.eu/ispnet/cms/preconisations/ingenierie-logicielle/liste_produits)):

|  |  |
| --- | --- |
| Oxygen XML | Y |
| Modelling tool UML | Y |
| Maven | Y |
| Source management (SVN) | Y |
| Full Application server | N |

**Non standard tools and justification:**

# AT4LEX

## Basic information

|  |  |
| --- | --- |
| Intranet | N |
| DMZ | N |
| VLAN dedicated | N |
| Internet access through the EP portal | N |
| Other | N |

## Network Architecture Schema

AT4LEX is the application that will allow the user the creation and the drafting of plenary documents. AT4LEX includes two main components: editor and dashboard.

The editor is based in AT4AMOS editor with some added services , these services are specific for EP:

**Style management.** Styles xml files will be stored in PURE-XML.

A layer to integrate with **EPS/PURE-XML** is also included. This will allow basically fetching and saving the documents.

The **validation services** will be able to detect TERMS[[20]](#footnote-20) (e.g. dates, names, references…) and perform spell checks by integration with a spell checker. (Hunspell)[[21]](#footnote-21).

Validation will be activate only by specific user actions (e.g. save) or automatic (e.g. backup) and performed at server side.

**Standard term caching.** User will be able to insert standard terms (country names, institutions names…). All these terms will be fetched from PURE-XML and cached to improve the performance.

The editor can be launch directly by other applications (e.g. AT4AM, DST, CAT4TRAD)

The AT4LEX dashboard will facilitate the creation (instantiation of documents), the management, security definitions and status changes. This dashboard it’s basically a set of datagrids showing the users different lists of plenary documents (new documents, recent documents, documents shared with me, documents managed by me, finalised documents….) and different possible actions (status change, edit, update from ITER, delete, share[[22]](#footnote-22) with other users…).

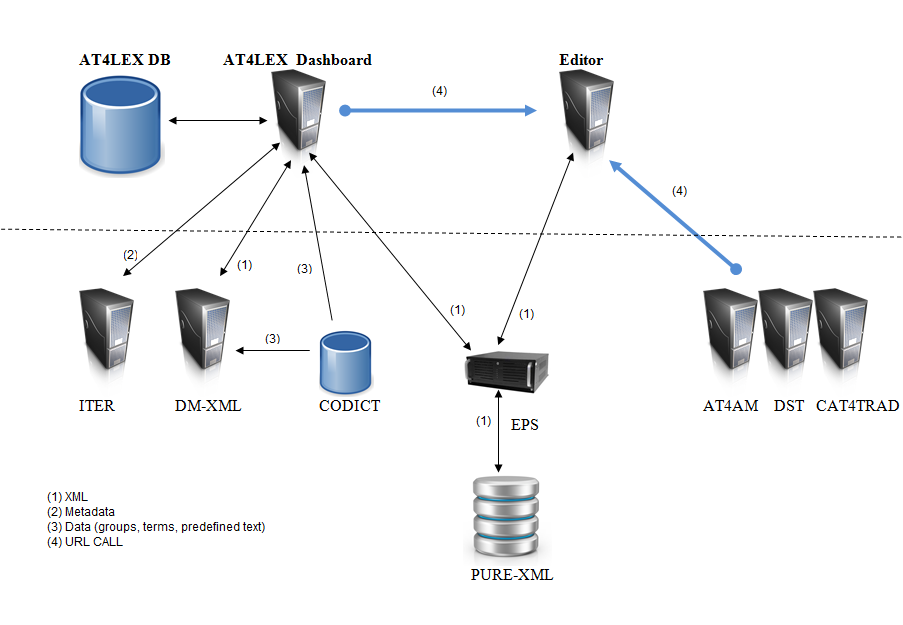


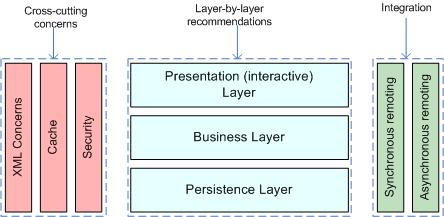
Figure 1: Network architecture diagram (EP Lan)

The network topology complies with the regular J2EE Intranet network architecture:

* + AT4LEX Dashboard Web application is hosted on an in-house J2EE server (Tomcat) and accessed in the Intranet via HTTP protocol
  + Authentication will be implemented using the standard JAAS (EPAUTH).
  + Role-based authorisation will be implemented, based on group membership (CODICT definition and AD)
  + The persistence layer is assured by in-house Oracle databases for data and PURE-XML repository for XML.
  + AT4LEX might send mails through a smtp mail server (intranet)
  + The Editor component is hosted in a separated Tomcat server.

The AT4LEX application is composed of the following components:

## Software Architecture



**List the technologies/libraries used:**

|  |  |
| --- | --- |
| **Client User interface (dashboard)** | JQuery 1.8.0 or ExtJs |
| **Client User interface (editor)** | AT4AMOS[[23]](#footnote-23) |
| Server Presentation layer | JSP |
| Server Business layer | Spring 3.1.x (latest production release) |
| Server Persistence Layer | Spring Data 1.1/JPA / HIBERNATE 4.1.7 |
| Db | Oracle 11 |
| Synchronous Remoting (WS, RMI, etc.) | Spring MVC 3.1.x for Restful Web Services |
| Asynchronous Remoting (JMS, etc.) | HornetQ 2.2.x (client and hosting) |
| XML Handling | JAXP / StAX / TrAX  Spring OXM (Castor, JAXB)  XStream (for basic serialisation) |
| XSLT Processor | Saxon-HE 9.3 (for XSLT2 capabilities) |
| Cache | Ehcache 2.6.2 |
| Security | Authentication EPAUTH over JAAS (ActiveDirectory) |
| Other | Jackson 1.9.11 |

## Required Standard Runtimes

List of required operational environment and associated 8 characters Code:

|  |  |  |  |
| --- | --- | --- | --- |
| **Existing (Y/N)** | **Standard Runtime [[24]](#footnote-24)** | **Code (8 char)** | **Environment(s) [[25]](#footnote-25) DEV, ATE, PPO, FOR, PROD, INT** |
| N | Tomcat 6.0.x, Java 1.6.0\_45 | AT4LEXD | DEV,INT,PPO,PROD |
| N | Tomcat 6.0.x, Java 1.6.0\_45 | AT4LEXE | DEV,INT,PPO,PROD |

Please note that one separate RHR document will be needed for each new runtime.

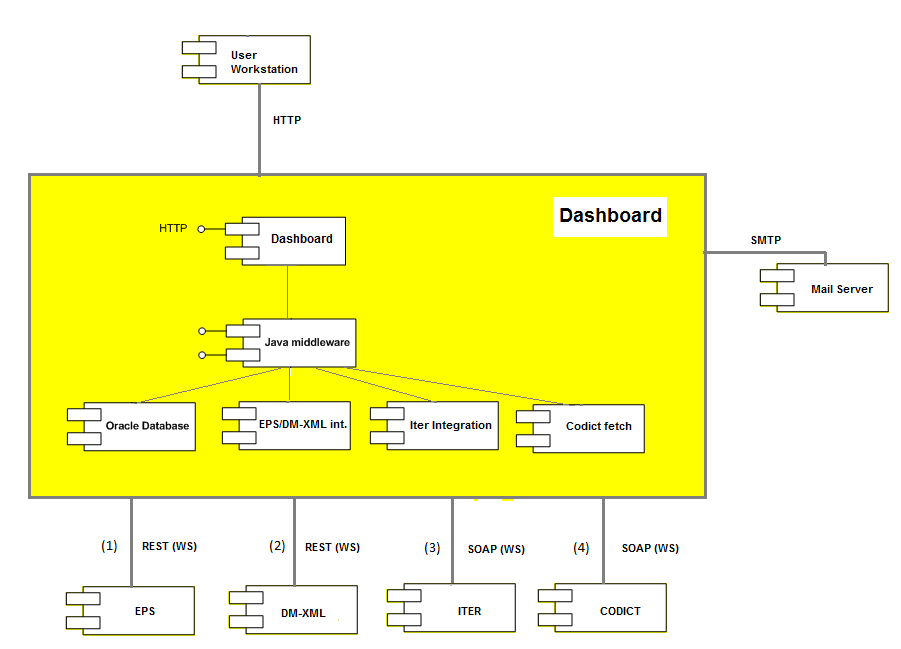
## Non-standard server-side softwares list

**Please note that these non-standard softwares are most likely not to be accepted quickly, though may trigger special attention and dedicated handling. Fill in this chapter as accurately as possible.**

|  |  |
| --- | --- |
| **Non-Standard Server-side software** | **Explanation of the need** |
| GWT is used by AT4AMOS. | Inside the EP programme we will converge to a single GWT editor which will be used by At4AM, DST and AT4LEX. Today 2 different GWT applications are in PROD (AT4AM, DST)  AT4AMOS is the base of this single editor but some features will be included. (e.g. spell checker. Print, copy/paste)  That’s why in this DTA two components are defined: Editor (GWT implementation) and AT4LEX dashboard (not GWT). |
|  |  |

## Integration

### Applications integration schema



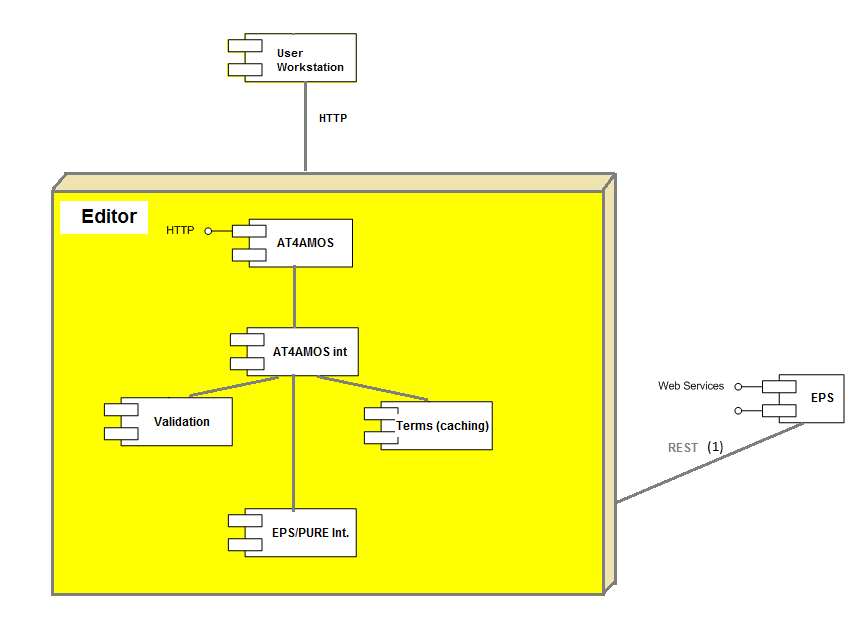


Figure 2: Applications integration schema

Several e-Parliament projects can open via URL-call the editor. In this URL the needed parameters to identify the documents (UBI), users (UUI), templates (UTI) will be defined.

Editor can be called by but is not limited to:

* AT4AM
* DST
* CAT4TRAD

### Application integration criticality

| **Reference[[26]](#footnote-26)** | **Protocol** | **Synchronicity** | **Frequency[[27]](#footnote-27)** | **Business Criticality** |
| --- | --- | --- | --- | --- |
| **PURE-XML/EPS (1)** | **WS** | **Synchronous** | **< 10/min** | **High** |
| **DM-XML (2)** | **WS** | **Synchronous** | **< 10/min** | **High** |
| **ITER-WS (3)** | **WS** | **Synchronous** | **< 10/min** | **High** |
| **CODICT (4)** | **WS** | **Synchronous** | **< 10/min** | **High** |

If the application cannot contact PURE-XML via EPS, some resources requested by the web service clients will not be accessible.

If CODICT services layer is not accessible, AT4LEX Dashboard will not work. The dashboard will not be able to check all the groups the user belongs (committee, political group, rapporteur, staff, assistant, Member…), but the Editor will continue to work.

Without ITER-WS and CODICT, the initiation of documents through DM-XML will not work.

Without DM-XML the Document instantiation/rendering will not work.

## System

### List of required system services

|  |  |
| --- | --- |
| Database | Oracle 11 |
| File system (direct access, sharing, FTP, etc.) | Direct Access |
| Directory (Active Directory, other) | N |
| Mails (sending mails from an application) | Y |
| Application Server | **EP Tomcat 6.0.x** |
| JDBC | Oracle THIN client |
| Other | N |

### System services schema (optional)

For clarity (and mostly if you have asked for multiple runtimes), please provide a schematic vision of the interaction between system services and applications.

## Security

### Authentication mechanism

The application’s users will login into application with EPAUTH mechanism, being validated against EP Active Directory.

### Authorization mechanism

Basic role management through AD (….)

Three systems will be checked for authorization depending if the user is accessing to data (Oracle) or XML file (e.g. Editor component):

1- Main Authorization for coarse grained access is based on Active Directory roles (defining the user type eg: reviewers, drafter, initiators, General Administrator, Normal User, IPOL/EXPO admin, PRES admin)

2-Authorization based on PURE-XML rights (ACL definitions per XML document)[[28]](#footnote-28).

3-Authorisation for more fine grained roles which can not be maintained in Active Directory are with CODICT organisational groups (e.g. : AGRI, AFCO…)

1. Used to authorize the operations at object level (change the status, publish, share…)
2. Used to authorize if the user can edit, check-in, update the document (XML).
3. Used to decide the objects accessible by the users . e.g. AGRI cannot access to AFCO documents.

### Specific confidentiality needs

NONE

## Specific Development Tools

The STANDARDS Unit provides standard tools for the development of applications.

Mention here licence-limited tools you may need ([restricted use product](http://www.ismsnet.ep.parl.union.eu/ispnet/cms/preconisations/ingenierie-logicielle/liste_produits)):

|  |  |
| --- | --- |
| Oxygen XML | Y/N |
| Modelling tool UML | Y/N |
|  | Y/N |

1. Program Mandate, p. 6-7. [↑](#footnote-ref-1)
2. This component is managed under the e-Parliament program framework, the budget is managed by DG TRAD and the MOE of the project is DG TRAD [↑](#footnote-ref-2)
3. Runtime: for example: Tomcat, HornetQ...etc [↑](#footnote-ref-3)
4. Environment (DEV, PPO, FOR, PROD, INT...etc) [↑](#footnote-ref-4)
5. An identifier helping to point this relation in the Application Integration Schema [↑](#footnote-ref-5)
6. Frequency of the interaction (e.g. once every night for a batch, thousands of calls per minute if directly related to end user input or interaction) [↑](#footnote-ref-6)
7. Runtime: for example: Tomcat, HornetQ...etc [↑](#footnote-ref-7)
8. Environment (DEV, PPO, FOR, PROD, INT...etc) [↑](#footnote-ref-8)
9. An identifier helping to point this relation in the Application Integration Schema [↑](#footnote-ref-9)
10. Frequency of the interaction (e.g. once every night for a batch, thousands of calls per minute if directly related to end user input or interaction) [↑](#footnote-ref-10)
11. DM-XML requires XSLT 2.0 to facilitate the implementation of different functionalities. See some examples in annexe I. [↑](#footnote-ref-11)
12. Runtime: for example: Tomcat, HornetQ.etc [↑](#footnote-ref-12)
13. Environment (DEV, PPO, FOR, PROD, INT...etc) [↑](#footnote-ref-13)
14. An identifier helping to point this relation in the Application Integration Schema [↑](#footnote-ref-14)
15. Frequency of the interaction (e.g. once every night for a batch, thousands of calls per minute if directly related to end user input or interaction) [↑](#footnote-ref-15)
16. Runtime: for example: Tomcat, HornetQ...etc [↑](#footnote-ref-16)
17. Environment (DEV, PPO, FOR, PROD, INT...etc) [↑](#footnote-ref-17)
18. An identifier helping to point this relation in the Application Integration Schema [↑](#footnote-ref-18)
19. Frequency of the interaction (e.g. once every night for a batch, thousands of calls per minute if directly related to end user input or interaction) [↑](#footnote-ref-19)
20. Terms are used in XML4EP(akoma) to define metadata that can be reused in the verification, consistency check process across languages and translation process (automatic translation). [↑](#footnote-ref-20)
21. Used by DG TRAD in CAT4TRAD. [↑](#footnote-ref-21)
22. Sharing operation will only change the PURE-XML ACL and include a new set of users or groups. [↑](#footnote-ref-22)
23. AT4AMOS is based on GWT technology. [↑](#footnote-ref-23)
24. Runtime: for example: Tomcat, HornetQ, etc. [↑](#footnote-ref-24)
25. Environment (DEV, ATE (Applicative Test Environment), PPO, FOR, PROD, INT, etc.) [↑](#footnote-ref-25)
26. An identifier helping to point this relation in the Application Integration Schema [↑](#footnote-ref-26)
27. Frequency of the interaction (e.g. once every night for a batch, thousands of calls per minute if directly related to end user input or interaction) [↑](#footnote-ref-27)
28. PURE-ACL: When a user try to access a PURE-XML document (via EPS), PURE-XML checks the associated PURE-ACL and perform or not the action. This is the same for check-in/out operations, save..... In this case PURE-XML is performing the action, not AT4LEX.  
    Concerning the Codict groups; these are filtering the information that the user can access. At the end AT4LEX will include these filters in the accesses to DB (queries) [↑](#footnote-ref-28)