02. Technické informace a validace systému

1. Hardware

The CovIT software runs on computer cluster located on Institute of Molecular and Translational Medicine (IMTM), Faculty of Medicine and Dentistry, Palacky University in Olomouc. The facility is secured and under global surveillance.

Description of hardware

Servers

Blade chassis - BM Flex System Enterprise Chassis

14x Compute node IBM Flex System x240 with 10 GB virtual fabric

2x CPU Intel(R) Xeon(R) CPU E5-2650 0 @ 2.00GHz / 8C

6x DDRIII SDRAM – 8GB

Data Storages

HP 3PAR data storage 700TB.

HP EML tape library

Firewall

HP F1000-S-EI VPN Firewall

2. Software

The software requirements

The only requirement for using the CovIT software is Internet browser which supports HTML5 standard. The list of supported browsers:

- Chrome: (Current - 1) and Current
- Edge: (Current - 1) and Current
- Firefox: (Current - 1) and Current
- Internet Explorer: 11+
- Safari: (Current - 1) and Current
- Opera: Current

(Current means the last available version of given browser)

Programming language

The main programming language used for development of the CovIT application is Java 8. Other technologies used for development are:

- Spring Framework v5.
- HTML, CSS
- JavaScript
- jQuery
- jQuery UI
- Bootstrap
- MathJS
- Datatables
- SQL
- Oracle database

Operation system

Operation system installed on production servers is **RedHat Enterprise Linux 7.4.**

Proxy server

The **Apache HTTP Server** is used as gateway from outside world to internal application running in the production server.

Application server

The **ClinData** application runs on **Apache Tomcat,** which is an open-source Java Servlet Container developed by the Apache Software Foundation

3. Database

The CovIT Database

The database used for storing data from the **CovIT** software is **Oracle Database** (commonly referred to as **Oracle RDBMS**) which is produced by Oracle Corporation. Version of database is 12.1. Standard edition.

The Oracle database runs on separated Linux based server which si firewalled from external network (Internet) by hardware firewall. This database server is not accessible from outside of organization but only from enlisted inner servers (application and backup servers).

4. Backup

There are more levels of data archiving to ensure data safety and quick database recovery. Data are archived on **database level** and **operation system level**

1. Database level backups
    - **RMAN** utility is integral part of the Oracle database. It creates binary copy of whole database and stores it to filesystem. The RMAN utility is run **every week**. The files are stored internally on database server and are copied to two independent backup sites.
    - **EXPDP/IMPDP** is data pump exporting data into text base backups. The EXPDP utility is run **every 4 hours**. The backup target is the same as with RMAN. It is stored to two independent backup sites.
    - **Redo Logs** are archived **every day** to filesystem.
2. Operation system backups

- **IBM Tivoli Storage Manager** (TSM Admin) is enterprise solution from IBM for backups and recovery of physical or virtual servers. The backup created by TSM Admin includes redo logs, RMAN and EXPDP exports. It runs **every day** and the backup data is stored to disk array.

RMAN configuration file

```
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP ON;
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; # default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR LOAD TRUE ; # default
CONFIGURE RMAN OUTPUT TO KEEP FOR 7 DAYS; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/../../oracle/12c/dbs/snapcf_imtm.f'; # default
```

EXPDP configuration file

```
DIRECTORY=dtpump
DUMPFILE=registry.dmp
LOGFILE=registry.log
CONTENT=ALL
COMPRESSION=NONE
JOB_NAME=registry_migration
SCHEMAS=registry,registry_aud
```

05. Secure connection

Security

As the **CovIT** application is **web-based** application there is need to **secure communication** between **server** and **client's computer**. It is done by using **HTTPS** communication protocol which is encrypted using Transport Layer Security **(TLS).** This protocol is widely used for all secure transactions on the Internet (payment, emails etc.) and is considered as safe and unbreakable. It protects against man in-the middle attacks. Communication without the security layer (HTTP) is can be interfered by attackers, they can listen to it or change it.

Security redirection

All user requests coming via unsecured **HTTP** protocol are automatically **redirected** to secure **HTTPS** protocol. All communication between client and server is secured and there is no way how to connect to the CovIT software via unsecured connection.

Certificate

The secured communication requires a certificate stored on the web server. The certificate must be signed by **trusted certificate authority**. The CovIT server uses the certificate digitally signed by **TERENA** authority.

06. Authentication and authorization

Users administration

All user using the CovIT application must be registered before they can log in. There is no possibility to get unauthorized access to the server even for some demonstration purposes. There is a specialized application for user management - IMTM Admin tool.

The **Admin tool** is responsible for:

- Management of **institutions, companies, hospitals** and their departments. There can be unlimited number of organization levels, for example a university can have such structure university-faculty-department-laboratory. Each organization level can obtain different set of privileges and roles.
- Management of **users**. Every user is identified by email address as login and password. Users are assigned to their organizations. Users can work on more projects with different roles. It is allowed by **user profiles**. Number of profiles for a user is not limited. Each profile can have different set of **privileges and roles**.
- Management **roles and profiles**.

The Admin database with user's data is stored in the Oracle database as separated schema. Access to this schema is restricted only for admin users. The server with the Oracle database is firewalled out of public network and not accessible from Internet.

An account for new user can be created **only by administrator**. There is no way that n user could create his account on its own.

These steps must be followed to **create new account**:

- New user asks a project owner to create new account
- The project owner asks administrator to create new account with specified privileges and roles
- The administrator creates new account and sets required privileges and roles
- The project owner checks account setting and approve it.
- New user receives his credentials and can log in.

Central authentication service (CAS)

The CovIT application must be connected with data from the IMTM Admin to control accounts, roles and privileges. It is done by integrating of the CAS technology into the ClinData software. The CAS technology consists of CAS Server and CAS Client.

The CAS server is responsible for authenticating users and granting accesses to applications. The CAS clients protect the CAS applications and retrieve the identity of the granted users from the CAS server.

07. Privileges and Roles

Access restrictions

A user access can be restricted in two different areas:

- restriction in **access to** ClinData **functionality**
- restriction in **access to data** stored in the ClinData software

All restriction is set in the **IMTM Admin tool**.

Functionality restrictions

Privileges

Access privileges determine which CovIT objects a user can browse or edit. Each functionality in the CovIT software is reflected in corresponding privilege so the access to everything is controlled. Any user or group of users can have access to any privilege granted or restricted.

The picture shows schema of privileges in the CovIT software.

Roles

Roles are virtual entities which serve as container for more privileges.

There are predefined roles and users, or groups of users can be assigned to them. The most frequently used roles are:

- ClinData system admin - full access to all functions in ClinData, no restrictions, creating new project
- ClinData project admin - full access to all function in selected project including study designer
- ClinData project data manager - access to all functions needed to insert new/update patient's data.
- ClinData project data monitor - access to all functions needed for study monitoring, validation and finishing CRFs.
- ClinData project data browser - read only access to selected data.

Data restrictions

Default setting for accessing of data in the CovIT software is maximally restricted. A user can see only data he inserted himself. By default, he doesn't see any data inserted by any other user. Access to any other data must be explicitly permitted.

These options can be set:

- user can see only his data
- user can see data inserted by other user or group of users
- user can see data linked with an organization
- user can see all data in a study

Personal data

There can be studies or registers which contain personal data. Access to this data can be restricted by special privilege.

These options can be set:

- user can see personal data
- user can't see personal data

8. Logging

The ClinData software **records everything** happening in the system. Admin user can browse these records in user friendly way and analyze potential problems, watch user activities etc.

There are three different types of logging mechanisms:

- **Software logging** is done on programming language level and is very detailed. The log files contain data about internal state of the whole system in time of logging event. This approach is designed for detailed analyses of problems which happened in past.
- **Access logging** is designed for controlling of user's activities. The access record contains data about who did an action and when. It logs all actions done on all objects in the system. Object can be study, patient, CRF form, file. These actions are logged:
    - create
    - open
    - change
    - add
    - remove
    - delete
    - export
- **Auditing** is focused to changes done in CRF forms. It records complete history of what was changed by users. One record contains data about:
    - when the change was done
    - who changed the data
    - what was changed
    - what is the new value

The important information is that the ClinData software **doesn't delete any record**. Every record in the database has system **flag ACTIVE**. Deleting of the row just sets this **ACTIVE** flag to **false**. The inactive rows are not displayed in the ClinData software but are still stored in the database.

9. Software development

Issue tracking

Any problem found in the CovIT software is documented and created as a **new issue in JIRA software**. JIRA software is developed by Atlassian and is an issue tracking tool. The new issue is analyzed, and priority is assigned. The list of issues is sorted by priorities and processed by developers. When a serious problem is fixed then it is published in new version of the CovIT software. The issue is also closed as done in JIRA.

Changes management

All requests for changes planned in the CovIT software are stored in JIRA. When a new request is coming then it is analyzed, time estimation is done, and priority assigned. The list of issues is sorted by priorities and processed by developers.

Versioning

The source code of the CovIT software is stored in **GIT repository** which allows tracking of changes in files. There is possibility to browse history of any source code file in the repository. Every change is also documented so it is easy to understand the development cycle.

Code review

Any change done in source code of the CovIT software must be **reviewed** by another developer. This process is called **code review**. This process minimizes number of bugs in source code because everything is double checked. **Bitbucket software** (developed by Atlassian) is used for code reviews. It prevents developers from using not proven code in public versions of the CovIT software.

10. Quality assurance

Testing environment

All new versions of the CovIT software must be tested and proven as functional and correct before they are published. There is a special environment which is used form testing of the new version before it is published. The testing environment must be similar to production environment to avoid configuration issues.

Unit testing

Unit testing is a software testing method by which individual units of source code are tested to determine whether they are fit for use. There are actually more than one thousand-unit tests in the source code of the CovIT software. All critical parts of the source code are covered by unit test close to 100%. Overall source code is covered by unit test by more than 85%. Any problem in unit testing is blocker for publishing of the version of the software.

Application testing

The whole application is tested by application exploratory testing before it is published. The application testing is done in testing environment. Any problem in application testing is blocker for publishing of the version.

Publishing

Publishing process means that a new version of the CovIT software is being released and made accessible for users. The Bamboo software (developed by Atlassian) is used for building and publishing new versions. Unit testing is also involved in publishing of the new version. In case of any problem in any unit test the whole publishing, process is interrupted, and an notification email is sent to responsible persons.