EHE Infrastructure

Commercial Description

Version 5.1

1 Introduction

Electronic Healthcare Exchange (EHE) is a line of products fulfilling a variety of eHealth system needs, ranging from fundamental ones like infrastructure, security, and integration, over exchange and management of clinical documents and discrete medical information, to advanced functionalities like clinical decision support. Solutions made of different EHE products, alone or through integration with the existing infrastructure, support a wide range of processes in a healthcare system.

The operation of any information system, including information systems in healthcare, requires certain basic functionalities that enable efficient and reliable exchange of information among system components, simple installation and configuration of individual components and services, monitoring of the system and management of the system and its components.

EHE Infrastructure consists of components that support the listed functionalities for the complete EHE portfolio and that can also be used for solutions developed by other manufacturers or for solutions that are specially developed for a particular customer or a particular project.

This product consists of the following components:

- Centralized Configuration Management
- Identifier Registry
- Platform Monitoring
- Message Queue
- Service Mesh Infrastructure.

It is not necessary to use all the listed components in every project, but only those that are necessary for the realization of the goals of the project at issue.

2 Functionality Description

The following subsections describe the components that make up the EHE Infrastructure.

2.1 Centralized Configuration Management

This component enables the centralized definition and management of various configuration parameters necessary for the operation of other components of the EHE Infrastructure.

The component makes it possible to save all parameters in a common database. For the purposes of retrieving configuration parameters from the central repository, the Centralized Configuration Management implements a

service used by all EHE Infrastructure components to retrieve their configuration parameters.

The Centralized Configuration Management enables the definition of parameters common to all components of the EHE Infrastructure, as well as parameters specific to each component.

If it is necessary to change certain configuration parameters during operation, the Centralized Configuration Management makes it possible to send requests to solution components to update parameters. Each component of the solution that supports dynamic changing of component parameters when it receives such a request, retrieves the new configuration parameters from centralized configuration management, and applies them.

2.2 Identifier Registry

This component enables the central registration, management, and validation of unique identifiers. Each entity or event (patient data, clinical document, healthcare provider data, medication data, prescription, referral, etc.) needs to be assigned a unique identifier to identify it and to avoid multiple records for the same entity or event.

The component provides services for creating and checking identifiers in various forms (OID - Object Identifier, UUID - Universal Unique Identifier or CUID - Collision Resistant Unique Identifier) to other components of the solution, as well as to external systems connected to the solution. These services enable the registration of one or more identifiers at once.

2.3 Platform Monitoring

This component enables the collection of data from other components and applications as part of the solution for monitoring their operation.

The component allows viewing the status of individual components and applications via a graphical user interface.

Listed below are some of the module, component, application, and server parameters available via this component:

- processor load
- working memory load
- disk usage
- components and subsystems status
- database components and middleware status
- operating system status
- number of requests for a service (in total or to the external system)

- the number of requests per individual service that are in the temporary memory and are waiting to be executed
- number of rejected requests per individual service
- number of requests per individual component or module
- request tracing - monitoring the execution and performance of requests concerning system applications and system services on all modules and components that participate in request tracing
- application events on components and modules.

2.4 Message Queue

The Message Queue component enables the connection of solution modules via a message queue and the exchange of necessary data among them. The component implements a message queue that uses the publish/subscribe principle and thus enables a particular module to publish a specific message, i.e. data, which can then be received by all modules that are subscribed to that type of message.

2.5 Service Mesh Infrastructure

The Service Mesh Infrastructure allows all modules and applications of the platform to register and deregister integration interfaces (services) as well as to search and route them.

At the start of operation each instance of a module or application registers its integration interfaces (name, address, etc.) in the central Service Register. When they need to access these integration interfaces, other modules and applications search in the central Service Register to see if there is an instance of the required integration interface and where it is located. If there is one instance of the required integration interface or more, the module or application accesses that interface based on the received address.

For the central Service Register to have up-to-date information on which instances of a certain interface are working, the Service-mesh Infrastructure implements the so-called tests that check whether a certain instance is functional and if not, delete it from the central repository.

If there is more than one instance of a certain integration interface, this infrastructure enables balanced routing of traffic to these instances according to various criteria (regular, routing of traffic to the least used instance, etc.).

The Service-mesh Infrastructure also enables the implementation of security rules for access to integration interfaces based on access lists. In this way, it can be defined that only certain modules and applications can access certain integration interfaces.

3 Free and Open Source Software

This product uses the Free and Open Source Software (FOSS) components with the following licenses:

- Apache Software License 2.0 [1]
- MIT License [2]
- Eclipse Distribution License [3]
- Eclipse Public License [4]
- Creative Commons CC0 [5]
- BSD License (2 clause and 3 clause) [6]
- Bouncy Castle License [7]
- Common Development and Distribution License [8]
- GNU Library General Public License [9]
- Mozilla Public License (MPL) [10]
- Elastic license v2 [11].

4 Version

The current product version is 5.1.

5 References

[1] Apache Software License 2.0
https://www.apache.org/licenses/LICENSE-2.0.txt

[2] MIT License https://opensource.org/licenses/MIT

[3] Eclipse Distribution License
https://www.eclipse.org/org/documents/edl-v10.php

[4] Eclipse Public License https://www.eclipse.org/legal/epl-v10.html, also https://www.eclipse.org/legal/epl-2.0 /

[5] Creative Commons CC0
https://creativecommons.org/publicdomain/zero/1.0 /

[6] BSD License https://opensource.org/licenses/BSD-2-Clause, also https://opensource.org/licenses/BSD-3-Clause

[7] Bouncy Castle License https://www.bouncycastle.org/licence.html

[8] Common Development and Distribution License
https://opensource.org/licenses/CDDL-1.0

[9] GNU Library General Public License
https://www.gnu.org/licenses/old-licenses/lgpl-2.0.html

[10] Mozilla Public License (MPL)
https://www.mozilla.org/media/MPL/2.0/index.48a3fe23ed13.txt

[11] Elastic license https://www.elastic.co/licensing/elastic-license