Tender
5/10/2023

Procuring organization

Region Östergötland
Bernadett Brink

Procurement

RFI, Request for Information of openEHR platforms and related tools
RFI-2020-08:2
Version 2: published 4/18/2023 11:46 AM
Tender closing date: 5/10/2023 11:59 PM

Legend

- The text is included in the advert
- The text is included in the qualification
- The text will be part of the contract
- The text will be published in the contract catalogue
- The text/question contains requirements to be met
- ESPD The text/question contains ESPD requirements
- The question is weighted and included in the evaluation
- The question is weighted and included in the evaluation
- The question is asked for information only
- The question is answered by the buyer
- The question is marked for special follow-up
- The answer does not meet the requirement in the question
- Updated section or question

Tenderers

Supplier	Tender	Corporate ID	Qual.
CaboLabs	Sweden openEHR 2023 RFI	CaboLabs	

Contents

3.1.11 Newbie

1. Invitation to openEHR RFI and demo

1.1 Invitation to openEHR RFI and demo

Sydöstra sjukvårdsregionen (including Region Östergötland, Region Kalmar län and Region Jönköpings län), Västra Götalandsregionen, Region Uppsala, Region Stockholm, and Region Skåne hereby invites suppliers of openEHR platforms and related tools (in this document called “Solution”) to a request for information and a product demonstration.

1.1.1 RFI process

This RFI process is divided into two (2) parts:

- The first part is open for all suppliers of openEHR solutions and consists of questions to be answered in written format, plus an appendix for context.
- The second part consists of an online product demonstration and is subject to specific qualification criteria. See Part 2: Demonstration sessions for details.

1.1.2 Date and time for demonstration sessions

The following time slots are available:

Date	Time (CEST/UTC+2)			
May 31	8:00-10:00 AM	10:00-12:00 AM	1:00-3:00 PM	3:00-5:00 PM
June 1	8:00-10:00 AM	10:00-12:00 AM	1:00-3:00 PM	3:00-5:00 PM
June 2	8:00-10:00 AM	10:00-12:00 AM	1:00-3:00 PM	3:00-5:00 PM

June 5 is reserved as an extra date for back-up purposes.

State which is your company’s preferred demo time slot, and also state all other time slots being acceptable alternatives.

Text field

June 2 3:00 - 5:00 (preferred)

June 1 3:00 - 5:00 (alternative)
May 31 3:00 - 5:00 (alternative)

1.1.3 Terms and definitions

Solution	The openEHR platform, related tools, and supporting applications that the RFI respondent can offer
RFI respondent	The part responding to the RFI

RFI document	This document
Application	A CDR external application integrated with the CDR, as part of - or not part of - the Solution.
CDR	Clinical Data Repository implementing the openEHR specifications
We	The group of county counsils issuing the RFI document
Request Context	All request metadata on the incoming HTTP request such as methods, headers, access tokens etc
Personal Data	The term ”personal data” is used throughout this document to describe every piece of information related to a specific patient kept by a healthcare organization.

1.1.4 No procurement

This is not a procurement. Please note that this does not constitute an RFP. Response to this invi

However, this is not bound to accept any of such information and/or expression of interest or to consider it further in any associated documents such as a RFP.

1.1.5 Confidentiality

During the RFI process, confidentiality prevails according to Chapter 19, Section 3 of the Public and Confidentiality Act (2009: 400).

Upon completion of the RFI, continued confidentiality may apply if there is reason to fear that a disclosure of information concerning the individual's business and operating conditions could cause harm to the individual. Furthermore, continued confidentiality may apply for the protection of the public interest.

When appealing decisions on confidentiality of information, RFI respondent shall assist the county councils and be responsible for their own costs arising from this.

In the event that the RFI respondent requests confidentiality, the RFI respondent must enclose documents describing the scope of the confidentiality and describe what damage the RFI respondent may suffer in the event of a publication. If the RFI respondent requests confidentiality, the RFI respondent must enclose a document specifying the parts of the RFI document for which the RFI respondent requests confidentiality and describe the damage the RFI respondent may suffer in the event of a publication.

a. Is privacy requested?

Yes/No

Answer

No

b. In those cases that the RFI respondent requests confidentiality, the RFI respondent must here attach what the privacy includes and describe which damage the bidder will suffer upon publication.

Attachment

0 attached documents

-

1.1.6 Questions about the request for information

All questions regarding the RFI must be asked via the VISMA TendSign RFI system, www.tendsign.com.

The wishes to receive questions in such a way that, together with the county counsils answer, they can be published without taking measures. The questions should therefore not contain information about the questioner's company, products or other information that can identify the questionnaire.

The county counsils want the RFI respondent to ask questions one at a time with reference to the point in the RFI document to which the question relates.

The county counsils answer the questions electronically in VISMA TendSign.

1.2 About this RFI

Region Östergötland, Västra Götalandsregionen, Region Uppsala, Region Stockholm, Region Skåne, and Region Kalmar (collectively referred to as “we” and “us” in this document) cover two thirds (⅔) of Sweden’s population. The majority of the county councils manage university hospitals with an extensive share of research and advanced healthcare. This RFI initiates the way forward, towards better healthcare and documentation solutions in Sweden.

This RFI aims at reaching all suppliers of openEHR solutions with an interest in the European market, in order to get an update on the latest news within the field. Doing this as a joint activity ensures higher quality results and is also timesaving for all parties.

The RFI may result in one or several procurements, either by each county council separately or by two or more county councils together. No decisions regarding possible joint procurements are taken yet and more county councils and organizations than these 5 may initiate procurements based on this RFI. Also note that all suppliers are welcome to take part in later coming procurements. There is no obligation to participate in the RFI and demo sessions, and participation does not affect later evaluation.

1.2.1 Facts about the County councils

The table shows some facts in figures about the county councils.

	Inhabitants (Total Swedish population is 10,5 million)	Hospitals	Health clinics	Dental care clinics	National specialized medical care assignments (46 different ones available)	Current main EHR system
Region Stockholm	2 440 027	5 (N/A)	Appr 600 (appr 1900)	Appr 80 (N/A)	36	CGM TakeCare

Region Uppsala	400 682	2 (3)	36 (58)	25 (80)	14	Cambio Cosmic
Region Östergötland	471 912	3 (3)	33 (47)	33 (109)	6	Cambio Cosmic
Region Skåne	1 414 324	9 (10)	100 (182)	69 (69)	25	Cerner Millenium
Västra Götalandsregionen	1 758 656	18 ()	117 ()	167 ()	29	Cerner Millenium
Region Kalmar län	247 711	3 (3)	26 (37)	18 (31)	N/A	Cambio Cosmic
Region Jönköpings län	369 184	3 (3)	28 (40)	26 (86)	N/A	Cambio Cosmic
Sum	7 102 496					

Population 2022 according to https://www.statistikdatabasen.scb.se/

National specialized medical care according to https://www.socialstyrelsen.se/en/clinical-practise-guidelines-and-regulations/regulations-and-guidelines/national-specialised-medical-care/.
Numbers within parenthesis () include collaborating private clinics etc.

1.2.2 Business impact goals

Three business impact goals of introducing openEHR-based healthcare systems are:

- Faster adaptation of IT systems to the constantly changing needs of the healthcare clinicians, including a more efficient system development process
- Increased control of stored health record data and increased reuse of information structures within and between applications, and between caregivers
- Increased freedom of action for the regions when the data is stored in a vendor neutral and open format

1.2.3 Purpose

The Swedish county councils are in the process of establishing an infrastructure for information management and information governance based on an information strategy and its target architecture. A key component of this infrastructure is to be able to store healthcare related information in a standardized and application neutral way.

The interoperability solution is an addition to existing healthcare information systems. A subset of the patients' medical records must be possible to handle in the CDR component both as master record as well as copies. We need a standardized reference model for how the information and data is structured and implemented in the CDR. Each application that renders information should have the ability to select, and customize its information stored in the CDR, in accordance with the reference model.

An example where this CDR capability would be relevant, is when an independent health app is used, but is not part of the main healthcare information system. In the long term, the CDR component will also be used for other applications of healthcare related information. Another early application will be remote/home monitoring.

Other secondary uses of interest are: patient created data, biobank data, healthcare business development, BI, AI, CDR, research, and quality registries.

2. Part 1: Questions

2.1 Questions

Answer the questions in this section in writing. Answer the questions that are relevant to your Solution. Not all questions in this RFI need to be answered, but the majority needs to be answered in order for you to be invited to the demonstration.

The supplier must enter all answers in the system.

The supplier may not attach documents.

2.1.1 General

a. What is the name and intended purpose of your Solution? Please name and (very briefly) describe the openEHR-related tools and platform components that you may be referring to in other parts of your RFI response.

Text field

The name of our solution is CaboLabs openEHR Platform

It is composed of:

1. Atomik: CDR and Demographic Data Repository compliant with the openEHR specs, single tenant, multi-API and GUI access, supports clustering for automated synchronization, DBMS agnostic (you can choose which database to use for storing information). https://atomik.app

2. EHRServer: open source openEHR-based CDR, multi-tenant, REST API + GUI access, supports clustering for automated synchronization, DBMS agnostic (you can choose which database to use for storing information). https://github.com/ppazos/cabolabs-ehrserver

3. openEHR Toolkit: SaaS application that provides tools to manage archetypes, templates, navigate their internal structures, and for generating and validating openEHR data. https://toolkit.cabolabs.com/

4. testEHR: is a load testing application that can be configured to send thousands of EHRs and compositions to an openEHR EHR, to allow measuring resource consumption and response times on the CDR. https://www.cabolabs.com/blog/article/i_tried_to_kill_the_ehrserver-5a9a340db6ebf.html

5. controlEHR: is a small app for health monitoring EHRServer and Atomik instances

6. openEHR SDK: reference model and template model implementation in Groovy, including parsers and serializers for XML and JSON, validation against openEHR XML and JSON Schemas, validation of data (locatables) against operational templates (OPTs), and more.

7. openEHR REST Client: an openEHR API client written in Groovy.

8. EHRCommitter: an application that can generate simple forms to input data that is mapped to an openEHR composition and committed to an openEHR CDR. It's used to demonstrate how openEHR works in practice and for training courses.

9. openEHR Conformance Framework: this is a set of specifications with a reference implementation focused on validating if any openEHR-based system complies or not with the openEHR specifications. The current implementation is focused on CDRs, though it can be extended for other types of systems (e.g. modeling tools, artifact managers, etc).

10. openEHR training: as part of our solution we offer different openEHR-related courses in three areas: foundation/introduction to the specs, clinical modeling and implementation (repositories, user interfaces, business logic, APIs and integration). We think training/education and, more generally, knowledge sharing is also a tool so we include this item in the answer.

11. Mirth Connect: is an open source integration engine designed for healthcare. We use Mirth to integrate different systems and apps, also as a broker to translate between different standards (openEHR, DICOM, HL7 v2.x, CDA, FHIR). We have experience on integrating with external components to add value to the whole solution, for instance we can implement imaginology test flows by integrating our CDRs with a radiology information system (RIS), a picture archiving and communication system (PACS) and a scheduling system. This kind of flow would require openEHR, HL7 v2.x and DICOM to work jointly. Similarly with laboratory test flows and medication prescription and dispatch.

---

b. In which country is your company located? Are there any sales partners or support partners in Sweden or Swedish speaking staff? Can your Solution or parts of it, e.g. additional services or license packs, be delivered via existing national Swedish framework agreements (see https://www.avropa.se/topplankar/In-English/).

Text field

We are based in Uruguay and have contacts in Sweden that can act as a local representative, so yes, we can get local partners to deliver services locally.

c. Describe the overall architecture of your Solution.

Text field

The CDR is the center of the architecture, it's responsible for managing archetypable information (mainly clinical and administrative) and its context. Information management includes organizing information in EHRs, recording contributions for those EHRs, audit logs and versioning openEHR objects (COMPOSITION, EHR_STATUS, FOLDER), in Atomik it also versions ACTORs (PERSON, ORGANISATION, GROUP and AGENT) and PARTY_RELATIONSHIPs (associations between ACTORs). Though it has its own template manager, it can use the openEHR Toolkit as a template storage and versioning solution. The CDR also uses an external SNOMED CT service to expand SNOMED expressions used as semantic filters in data queries. The CDR, specifically Atomik, provides different APIs:

EHR this is our implementation of the official openEHR EHR API.
Demographic this is our proposal for an official openEHR API that supports demographics, since at the moment of writing this documentation there is no official openEHR Demographic API.
Administrative this API contains extra operations that are needed, for instance, to run conformance tests on an Atomik instance and do physical deletions of certain data elements (useful for maintenance or to comply to local regulations like the "right to be forgotten")
Sync this API allows the implementation of Atomik server Clusters, allowing different instances of Atomik to share data between them (sync).
Monitoring this API allows external monitoring tools to connect to Atomik and get current status and health indicators, which is specially useful when managing Atomik Clusters with several instances.

For the SNOMED CT service to expand expressions from the CDR data queries we use Hermes (https://github.com/wardle/hermes).

The openEHR Toolkit, besides the internal management of archetypes, templates and automatically generated artifacts like COMPOSITIONs (generated from OPTs), offers an API to access all those items. The archetype and template access complies with the openEHR REST API endpoints (https://specifications.openehr.org/releases/ITS-REST/Release-1.0.2/definitions.html) and allow the CDR to use the Toolkit as an OPT repository. It also allows apps to access generated COMPOSITIONs to help integrating and testing with a CDR, also for rapid prototyping apps.

In the previous point we also mentioned some applications that run on top of the components mentioned here, the purpose was already mentioned, and the way they connect to the CDR is just via the openEHR API, in the case of apps that commit data, and using the Monitoring API in case of the controlEHR app. Then different instances of the CDR talk with each other via the Sync API when configured as a cluster, which is used for automated backups, high availability and for scalability (queries can be load balanced between the CDRs in the same cluster).

A security component can be used as an identity provider, which also handles roles, permissions and API keys, is Keycloak. This is an optional component since our CDRs already have an identity provider, manage users, roles, and API keys internally.

So the core of the solution is CDR (EHRServer or Atomik) + SNOMED CT service + openEHR Toolkit. The rest of the components can be used optionally depending on the functional and integration requirements.

d. Describe if/how openEHR's Task Planning functionality (or other process support) is supported by your Solution now, and your future roadmap for such support.

Text field

We don't have process support in our architecture yet, we want to focus on information infrastructure and value added services on primary and secondary uses of data, data discovery and integration based on the CDR alone. If required, we will integrate with a process/flow execution from another company as a data provider. For instance we are in contact with MedicalFlows that have a solution exactly for this item (https://medicalflows.com/) and we already have a collaboration agreement.

e. Describe if/how the Solution supports development and use of clinical decision support (CDS), for example using openEHR's GDL or GDL2 specifications now, and your future roadmap for such support.

Text field

We did research work in the past and developed a prototype for a rule engine that consumed openEHR data to evaluate rules and provide different types of results like events, actions or just codes an external application could interpret. This tool is not currently in development, we are focused on the information infrastructure now, but it's in our future roadmap to continue developing that prototype into a product, then look for compliance with GDL2 (or maybe a future version of GDL). In the short term, we would integrate our CDRs into the CDS tools developed by Cambio, as data providers for rule evaluation.

2.1.2 Delivery models

a. List the delivery/deployment models you support, such as local installation (OnPrem) or cloud installation (for instance SaaS)?

Text field

Right now we have EHRServer open source, which the community installs and uses wherever they want, and we offer EHRServer in SaaS (https://cloudehrserver.com/), in which users pay a subscription.

Atomik can be installed on-premise, in a public cloud or in a private cloud. All the models are licensed.

b. Describe, in the case of SaaS deployments, your subcontractor structure used to deliver the service. List any hyperscalar public cloud services used and the jurisdiction they operate in with relation to the EU/GDPR and transfer of personal data.

Text field

Our solution is very agnostic in terms of infrastructure and is not tied to any PaaS or Cloud Service, so we can work with any cloud provider. For instance, if a SaaS deployment mode is required, we can use a local or EU PaaS provider, or even use the private clouds of the hospitals if they have those.

c. If you are dependent on third-party suppliers in your solution proposal, how do you package this with an overall responsibility regarding usability, licenses and support?

Text field

We are in contact with the developers of any component of our solution that wasn't developed by us, so we can receive their support during the deployment, support, adaptation, integration and maintenance processes. For this we can sign formal partnerships. We also tend to use free / open source products that are proven to be stable enough, like Hermes for SNOMED CT processing, and Mirth Connect for implementing integrations. With Mirth Connect we actually give support to other companies because we are experts on it, having been working with it since 2007 and providing online training in Spanish and English since 2014.

d. Can applications based on output from your products be published as open source? If so, are there any restrictions on usage? This implies e.g. that generated code, forms, configuration information etc. and exported runtime components should be perpetually allowed to be included in open source based systems and in associated, possibly public, versioning systems (like GitHub).

Text field

Yes. Our platform allows to develop any kind of application based on standardized information models, and our CDR is basically a vendor neutral archive (VNA), so any application developed on top of it is really independent from the technology stack itself, it just depends on the models and data managed by EHRServer or Atomik. So even if Atomik is not open source, apps developed as open source, using Atomik, are totally independent from it, and the API calls to Atomik are really openEHR API calls defined in the openEHR specifications, which are open and free. So there are no restrictions on usage.

e. Describe how your product can be installed using containers and container orchestration tools such as Kubernetes.

Text field

Focused on Atomik and EHRServer, though we use the same technology stack for other components, like the openEHR Toolkit, a container just needs:

1. JDK
2. Apache Tomcat (servlet container)
3. Apache HTTP Server (front to manage SSL certificates, provides HTTPS and load balancing)
4. MySQL (can be switched for another relational DBMS)

For simple deployments, all components (1-4) can be installed in the same box/container.

For scaling it's recommended to have a separated container for the database, it also allows clustering configurations at the database level, which is separated from the application level (1-3).

For a more distributed configuration, the Apache HTTP Server front can be in a separated container, and serve to the front to many Apache Tomcat containers running one instance of Atomik each. Apache HTTP Server can be configured to deliver requests to a set of Tomcats with load balancing, and the Atomik system deployed on each Tomat can connect to MySQL that can be configured as a single db or in a cluster, also with load balancing. This configuration is very flexible and scalable, and doesn't require anything specific from the container infrastructure to work, just the network to be properly configured.

In terms of volumes (persistent storage) each database instance requires one volume (for the physical database), and each Atomik instance also requires a volume (for storing logs and static resources).

f. Describe your approach to scaling your Solution. Describe known limitations, for instance regarding performance.

Text field

The components that can be scaled are described in the previous point, basically adding application servers (servlet container with load balancing) and adding database servers (clustering with load balancing). Atomik and EHRServer can also be synchronized at the API level to provide totally independent servers (application and database) that can serve exactly the same data and that can also be load balanced via a gateway.

Since scaling directly depends on usage, first we need to detect any bottlenecks caused by real usage to know on which area we need to scale. That will determine the usage profile like a write intensive environment or a read intensive environment, and the bottleneck level like API performance issues or I/O performance issues. Also resource usage levels that could affect performance should be analyzed, for instance if CPU or memory are constantly at 90% or more, scaling is needed to distribute that load. Then we add another server at the application level (Atomik), or another server at the servlet container level (Tomcat), or another server at the database level (new node for a MySQL cluster), depending on the factors aforementioned.

An extra item can be added for scaling and improving performance in general, that is Preemptive Caching, which can drastically improve performance of frontends, for instance, caching common EHR items that will be used in scheduled visits. This architectural component sits between the CDR and the application.

g. Briefly describe your three (3) largest or most interesting customer installations based on an openEHR CDR. Also describe how long it took to go from purchase to operational system with real patient data and actual use.

Text field

Atomik is just being released, so we don't have installations yet. We do have a partner in India that we are working with to have an integration and a deployment in the next few months. For EHRServer we don't have information on usage since most installations don't even contact us. CloudEHRServer is mainly used by researchers.

h. Describe what kind of infrastructure your Solution requires from a customer. Also describe your normal implementation/deployment process.

Text field

Depends on the deployment mode. For on premise deployments, a minimal installation would require a couple of servers to install the web server, servlet container and database, plus monitoring and other tools if required, like Mirth Connect to handle integrations.

Implementation without any integrations is pretty straight forward. We install the infrastructure software mentioned above, configure SSL certificates to provide HTTPS access, securely configure Apache, Tomcat and MySQL and execute a vast set of tests in the production environment, controlling resource usage (CPU, memory), response times and any errors caused by the application or the configuration. We go back and forth fixing and testing until we certify everything works as expected.

Then we go train the local teams that will work directly with our information infrastructure, and provide any reference materials they need to work with the platform. Finally we follow closely for the first couple of months of usage to see if there is any issue that we need to fix. After that we continue in support/maintenance mode.

i. Describe your software lifecycle strategy and release cadence.

Text field

We try to continuously publish updates, grouping them into milestones. Each milestone is a release that is planned for a certain date. That can be negotiated with the customer, though we like to do 2-3 month releases to have time to test any changes, but also document the updates and create any migrations needed, if the modification affects existing data.

For the updates we need a ticket formally created that describes the change requested, which could be about a correction, improvement or new feature. We organize them in milestones, trying to consider priority and balance workload to make this backlog manageable.

In terms of the infrastructure software, we have update cycles when new versions of Apache, Tomcat or MySQL are released. First we test the new versions in an isolated environment, then we schedule updates in production. Some of these updates might require some downtime, which we try to minimize, or data migration.

j. Describe your future roadmap. What major features are planned and when are they planned to be released?

Text field

In the future we plan to add context-specific functionalities from our information infrastructure platform, including:

a. Support for questionnaire definition and management, including services for rendering, storing and querying questionnaires.

b. Support for different types of informed consents for procedures, treatments and other areas of care, including definitions, management, rendering, storing and querying information about the consents.

c. Support subscription-notify model to get notifications about internal events, for instance when a COMPOSITION is stored and complies with a certain template, send a notification to another system that will use the information, for instance a lab order gets the order once it's stored in the CDR.

d. Create more data discovery services, for instance combining family trees with information about inherited genetic diseases, that can run automatically and notify clinicians of potential risks for patients, like risk of cancer on a person based on his/her ancestry, which is one of the domains we want to tackle.

2.1.3 Legal and regulatory aspects

Please refer to background information in appendix "OpenEHR – an Implementors Guideline related to Swedish laws and regulations in healthcare". It also reflects our level of ambition, and discusses some different possible openEHR-based solutions. Please feel free to be inspired by this document; we also look forward to receiving alternative solutions and discussions. We refer to COMPOSITIONs below to make the text more readable but we are actually interested in corresponding behavior regarding all relevant VERSIONED_OBJECTs (for example FOLDERs).

2.1.3.1 Multi-tenancy, Federation and Metadata

a. Describe how the Solution can be configured to support multi-tenancy where clinical data for hundreds of organizations (care providers/care units) can be managed efficiently.

Text field

EHRServer is natively multi-tenant, it works separating records by organization and associating users with those organizations.

Atomik was designed for single tenancy, having one logical instance per organization, and access to shared data between organizations will happen in a federated approach that could be implemented with an IHE XDS architecture or similar, by publishing documents, that accessible by the consumers in the federation, in a centralized directory/index, but persistence is not shared, just access is shared, so the directory points to the organization that is the custodian of the document, and a consumer needs to query the directory first, then access the document from the custodian by using its federated API to access the physical document. In this case, management is independent, only the index of the data and access to the data are shared.

b. Describe how the Solution can be configured in a fine-grained multi-tenant model (see Appendix A) so that a COMPOSITION and/or parts of a COMPOSITION within an EHR record can be attributed organizational ownership. Also describe how and where this metadata can be persisted.

Text field

In openEHR a COMPOSITION is not specifically designed to contain multi-organizational data, though at the ENTRY level, the provider data could be used to point to a specific provider at a specific organization. Even though that is used, the COMPOSITION itself should reside in a repository and be associated with an EHR, both which should have a custodian organization that is responsible for maintaining the EHR.

We believe this question has to do with how you envision to use data between organizations, but doesn't seem to fit in openEHR very well, at least with the current specifications. So we would recommend the IHE XDS federated approach first, but when learning more about your envisioned usage, we could add that specific requirement to the implementation, extending our solution to comply with your specific needs. Of course, this point would be better discussed with more context information, we are just assuming.

c. Describe how metadata about organizational ownership/multi-tenancy, and about source (e.g. originating/feeder-system), can be verified/validated against the Request Context and/or external attribute sources to make sure that the proposed metadata is valid and that the user has sufficient permissions to write/modify data for this unit.

Text field

For Atomik, each server instance is independent from other organizations and the way of sharing data between organizations is via an IHE XDS environment, for which Atomik would be a Document Repository actor (https://profiles.ihe.net/ITI/TF/Volume1/ch-10.html). In this context, only what is published in the XDS Document Registry will be shared. Atomik could be configured to share certain types of documents, based on openEHR templates, and we can add other types of filters at the patient or EHR level, including sharing based on consents.

For EHRServer, each user is associated with one organization and each user can access only records from that organization. If needed we can do an internal sharing of compositions between different organizations in the system, so the access will be based on the current user-organization relationship or in the sharing configuration. This sharing configuration can be for a type of document, based on a template ID, or specific for a patient or EHR.

2.1.3.2 Querying and Multi-tenancy

a. Describe how (see Appendix A) the Solution can be configured to filter a response from the EHR API resource endpoints based on metadata from the Request Context, external attribute source and/or metadata on the COMPOSITION itself (such as validated metadata for organizational ownership).

Text field

We don't filter responses directly in the CDR. Though in queries we return EHR IDs when querying multiple EHRs. If access to data is based on the access the user has over the EHR, then a small component could be added on top of the query service to check the EHRs accessible by the user and filter all results that are not in that set. Though if the user queries for data on a single EHR, the request could be filtered following the same criteria, because the request contains the EHR ID.

For finer grain filters, we can create filters by document type (template ID), by type of information (archetype ID), or by object (LOCATABLE UID). For COMPOSITIONs we can also use the data on EVENT_CONTEXT for filtering. These rules can be per individual user, per user role, or per user group. Note this component is not implemented. We would need to know exactly what you need to filter in order to provide the right solution. All this is technically possible, since we have the data and metadata to do it.

---

b. Describe how (see Appendix A) the Solution can be configured to block or filter out parts of a RESULT_SET from the Query Execute API resource endpoints based on metadata from the Request Context, external attribute source and/or metadata on the COMPOSITION itself, such as validated metadata for organizational ownership. (Example of possible solution: Incoming ad-hoc queries and/or stored queries may be temporarily modified to support the filtering.)

Text field

Mentioned above. Filtering a full response, or filtering part of it, would use the same mechanisms.

---

c. Describe if and how (a possibly extended set of) the openEHR Reference Model can be used to block or filter out parts of a RESULT_SET from the Query Execute API resource endpoints based on metadata from the Request Context, and/or external attribute sources. Describe at least support for using the following classes for blocking/filtering data

i. FOLDERs

ii. TAGsF

iii. EEDER_AUDIT

Text field

The filters could be by LOCATABLE instance, so in the case of FOLDER could be by it's UID, or by name (both inherited attributes from LOCATABLE). A more complex check would be to go through the items contained in the FOLDERs, and for the COMPOSITIONs, allow access to those folders to all the COMPOSITION composers and participations. Though that would affect performance if those data sets are not flatten and cached (e.g. in memory).

We don't support tags yet, in fact it's a feature that is being discussed, I think there is an issue on the openEHR JIRA about that which is still open (https://openehr.atlassian.net/browse/SPECRM-87?search_id=00f25e3a-c95a-4d32-afeb-660e1ddfd7d1)

As mentioned above, having the access permission defined by user, role or group, the permission could be defined over any FEEDER_AUDIT or FEEDER_AUDIT_DETAILS attribute as needed, for instance associating a group with specific items by the remote ID (from FEEDER_AUDIT.originating_system_item_ids) or with a system_id (in FEEDER_AUDIT_DETAILS) to say "this group has access to everything that comes from this system_id (unless there is another rule that overrides that access).

So this is not trivial, and should be adapted for specific requirements considering local regulations. Any rule or set of rules can be written and checked over existing openEHR data points, we just need to know which subset of all the possible attributes will be used.

d. Describe how the Solution can be configured to block and/or allow requests to resource endpoints from the ITS-REST specification based on metadata from the Request Context and/or external attribute sources.

Text field

Since we use JWT as access tokens we can add metadata to the tokens. Some metadata can be about exactly this: which endpoints can a user invoke, and which HTTP methods are allowed for each endpoint (e.g. GET allowed but not POST).

2.1.3.3 Bulk Operations

a. Describe any tooling and/or APIs available for managing bulk operations on COMPOSITIONs. Describe how the target set of COMPOSITIONs (bundle/batch) can be defined from a result of an AQL query.

Text field

We have a sync API that is used to copy data (EHRs, templates, COMPOSITIONs, FOLDERs, stored queries, etc.) to another instance of Atomik/EHRServer. This API is not provided by openEHR, it's specific to our CDRs.

About batch selection of EHRs, we have "EHR queries" that use simpler queries as filters, so we can select a set of EHRs based on: age range, sex, chronic diseases and comorbidities, conditions, lab tests, etc. and get back the EHRs that contain at least one COMPOSITION with data that matches each of those simpler queries. This functionality is to be used for CDS and cohort building, for instance to get a set of patients to participate in a clinical trial or on a specific care plan/treatment.

The simple queries can be used to filter COMPOSITIONs from different EHRs and get the IDs of those for working with them in bulk by an external process. For instance if you want all COMPOSITIONs with Blood Pressure data from the last 3 years where the BP is high for all patients in the repository, then do something with those COMPOSITIONs, for instance: research, statistical analysis, epidemiology, etc.

b. Describe any tooling and/or APIs available for managing bulk import operations of COMPOSITIONs. Describe how metadata on COMPOSITIONs are validated/verified.

Text field

For importing we also have the sync API. So all Atomik/EHRServer have both a sync API server and a sync API client, so they can sync in any direction when configured. For importing data from other CDRs or third party repositories, we would do a normal ETL project and use an integration engine to manage the bulk import.

2.1.3.4 Audit Logging

a. Describe the set of triggers (instrumentation) the Solution can use for audit logging. What is logged and when?

Text field

We can log every request event, and we do, locally. For the commits we also log if it succeeded or not. At the openEHR level we use the usual CONTRIBUTION + AUDIT_DETAILS to log the openEHR operations. We can also integrate a syslog server to push events to an external server if analytics over the logs are needed for instance to put on a visualization framework like Grafana/Kibana.

b. Describe how the Solution can be configured to export audit logs and/or integrated to external SIEM systems. Also describe and/or list the supported technical interfaces.

Text field

We can integrate our logs with any external system, it’s not a configuration, it's an integration task. This highly depends on requirements. We do what the client needs.

2.1.3.5 Certification of products, tools and modules

a. Are any of your openEHR products, tools or modules certified (CE labeled) according to EU Medical Device Directive 93/42/EEC or the EU Medical Devices Regulation (MDR)? If yes, please state which product or module that fulfills which regulation.

Text field

No

b. Describe your experience of the process to CE label a software as a medical device?

Text field

-

2.1.3.6 Accessibility

Describe how the Solution supports (or helps creating) end user interfaces in accordance with the European accessibility directive European accessibility act - Employment, Social Affairs & Inclusion - European Commission (europa.eu).

Text field

We focus on backends and integration not on frontend.

2.1.4 Platform and development

a. What parts of the Solution are open source and what parts are proprietary? Describe what open source license you use.

Text field

EHRServer is completely open source. Apache 2.

Atomik is not open source. It uses a SNOMED CT service that is open source, Hermes.

The openEHR Toolkit is not open source, it's SaaS with subscriptions, with a free tier.

openEHR Toolkit is based on our openEHR Reference Implementation and CLI toolkit, which is open source. Apache 2.

We use the Mirth Connect integration engine for most integrations, it's open source. MPL 2.0

b. Describe any prebuilt products or EHR-modules based on the platform that you can provide, for instance end-user applications for surgery, emergency wards, medications, or primary care. Also describe any provided "portal" functionality or similar that can easily be configured to different use cases where e.g. clinical end users can browse, read and enter openEHR-based data. Also briefly describe the pricing model for these.

Text field

We have a small emergency application EMEhealth http://emehealth.com/

That is not yet integrated to our CDRs, but it's based on openEHR principles. It also generates some basic forms based on form definitions.

We also started working on a primary care application which is just a simplified version of EMEhealth.

The goal was to release both systems around 2020, but because of the pandemic we switched our focus to remote projects instead, because in-person showcase and selling was impossible at that time.

c. Describe your integration support, tooling and experience, including but not limited to the list items i-vii below. Clearly indicate which list item the answer refers to.

i) Software development kits (SDK:s) for developing and integrating towards your API:s etc.

ii) Publish/subscribe patterns

iii) HL7 FHIR

iv) API standards (such as HL7 v2, IHE, ODBC, OpenAPI) and other interoperability and connectivity standards

v) Integrations with medical imaging standards such as DICOM

vi) OMOP and other standards used for research

vii) Existing EHR systems in Sweden (if so, please state which)

Text field

We have our SDK which is the reference implementation of the specs plus CLI tools, which is integrated into Atomik and the openEHR Toolkit, and can be used by any client to create and manipulate openEHR data, serialize and parse, validate using schemas and templates and more. Then we have another component, the openEHR REST client, which also uses the SDK and allows simplified calls to the openEHR API implemented in Atomik.

We don't currently notify external systems of internal events, but we have a work line in that direction in our current backlog, so external systems can subscribe to certain events, and Atomik can automatically notify them, passing the context data, when those events happen. The use case is for test orders and medication prescriptions, so we can notify LIS, RIS and pharmacy systems about those events so they can process the orders and prescriptions. This might happen by using Mirth Connect (integration engine) for message transformation, for instance if the LIS or RIS accept HL7 v2.x and we need to generate that from an openEHR JSON COMPOSITION.

Note: HL7 v2.x is fully implemented in Mirth Connect so we use that component for any transformation to/from HL7 v2.x.

Atomik can work as a document repository component for the IHE XDSb. profile, it just needs a small SOAP component on top.

We have a component that implements the Query/Retrieve and WADO DICOM services, used to query, get and display images from a PACS. We can also use Mirth Connect for DICOM exchange since it is fully implemented there (uses DCM4CHE Toolkit). In terms of integration with the CDR, we would store COMPOSITIONs representing DICOM Structured Reports (radiology reports) which would point to reference images from imaging studies (in openEHR those DICOM images can be referenced by DV_MULTIMEDIA which is fully supported). That way the report is part of the openEHR EHR.

We have experience with DICOM, connecting modalities, PACS and RIS with external systems, like HIS/EHR, generally using both DICOM and HL7 v2.x. At CaboLabs one of our main jobs is systems and data integration, so we are used to working with different standards, message formats and communication protocols.

d. Describe how an external terminology server can be connected to the Solution and used both for term selection in forms/GUI and for validation of incoming COMPOSITIONs via API.What terminology server standards or products have been successfully tested and used with the Solution?

Text field

The openEHR CDRs require an SNOMED CT service just for the queries, since we support embedded SNOMED CT expressions as an advanced semantic filter in our openEHR data queries.

We don't work in the GUI, for which a terminology server would be required for assisted coding in the frontend.

In the future we want to add a terminology server at the CDR level for optional coded text validation.

e. Describe if/how the openEHR demographic model specification is supported by your Solution now, and your future roadmap for such support.

Text field

It's fully supported in Atomik, including queries. We are designing some advanced use cases involving researching family trees for inherited chronic diseases.

In fact we proposed to the SEC a design for an initial demographic REST API based on our demographic API. We also submitted some design decisions for SEC consideration https://discourse.openehr.org/t/atomik-cdr-and-ddr-website-is-online-and-demographic-api-proposal/3877.

In the near future we want to add support for person/patient cross-referencing, identity merging and other Master Patient Index functionalities based on openEHR demographic data. We wil also focus on functionalities traversal to demographic and EHR information.

f. Describe query mechanisms in your Solution. Clearly indicate which list item the answer refers to.

i) Describe what version of the AQL specification the CDR supports and if something from the specification is not yet supported.

ii) What parts of the RM can be reached and used as selectors and filters in queries in addition to more "normal" COMPOSITION content? For example, how can FEEDER_AUDIT, LINK, FOLDER (including the FOLDER.details ITEM_STRUCTURE) and TAGs be used to select and filter content through AQL syntax (extensions) and/or via context information like API call parameters?

Text field

We implement a different formalism than AQL, since the AQL spec is heavily focused on the query syntax but doesn't specify how the queries should be executed internally and how the result sets are constructed. Our formalism relies on a model, not on a syntax, and the specification tries to define those areas AQL doesn't define. I personally reviewed the AQL spec in 2019-2020 and added a lot of fixed, trying to improve the specs, that is why we have deep understanding of the spec and its limitations, for instance portability between vendors without some refactoring, and even if you have the same query executed on different CDRs loaded with the same data, you might get slightly different results.

This is the SImple Archetype Query Model spec (WIP)
https://docs.google.com/document/d/1DDQTPqCZNjmPbmq2qZRiVFFxAumh9AMFV7guz4peOtc/edit?usp=sharing

SAQM could be seen as a subset of the general-purpose AQL syntax.

With SAQM you can query at the EHR level or at any LOCATABLE level, including demographics. Another type of query allows you to get individual data points. So it can return data values or full LOCATABLE objects, not intermediate structures alone, like a CLUSTER or an ITEM_TREE.

---

g. Describe if and how you support use of openEHR's TAG and FOLDER classes and mechanisms, including for what API endpoints (such as …/composition and …/query) they can be used to for example show/hide data based on if data belongs to certain FOLDERs (or it's subfolders) or not, or based on the presence or absence of certain TAG keys and TAG values.

Text field

It seems this question is related to 2.1.3.2 about blocking responses based on attributes or metadata, but this mentions showing/hiding data, which might be related to a GUI feature.

If that's the case, we don't work at the GUI level. If this relates to blocking responses based on attributes, please refer to the answer given in the aforementioned section.

2.1.5 Tools

a. Does the Solution provide integrated version control tool support (for example Git/Github integrations) for easy retrieval and storage of assets, such as archetypes, templates, forms, and queries? If yes, please describe it briefly.

Text field

We implement basic version control internally in openEHR Toolkit, following semver, with some tools that allow to check diffs between templates.

For queries we don't have a version control mechanism but it's something we sure need to add to manage queries during a long time.

We don't use the rest of the elements in the CDR (forms, archetypes, etc). Though in the Toolkit we do manage archetype versions. We want to improve that in the Toolkit so we can provide a similar version control like the CKM does.

We don't have tools yet, for instance, to detect dependency changes like archetypes changed after an OPT was published using those archetypes, and the archetype ID were not modified. For that a CKM like tool could be used, that could be Ocean's or Cambio's CKMs.

---

b. Describe how/if your products include tool support, and how well they comply with specifications, for openEHR archetype/template lifecycle management and related form lifecycle management.

Text field

Currently we don't manage template lifecycle. We can use any template uploaded by a user, even if it's on draft or published. Though it's another thing to add to the openEHR Toolkit to grant only published archetypes and templates are used in production.

c. Describe how your Solution supports multilingual openEHR models in data and end user interfaces. How do you provide workarounds for OPT 1.4 multilingual limitations? Describe if tool-interfaces are multilingual and can be translated and localized to Swedish.

Text field

Atomik and EHRServer are completely translatable to any language. Right now we have translations for English and Spanish. The openEHR Toolkit is English only and can be translated.

In terms of multiple languages in OPTs 1.4 you just need to generate one OPT for each language, and upload them using a different template_id to avoid conflicts at the REST API level.

d. To what extent do you support combining your Solution with components from other openEHR vendors? Describe successful tests you have done regarding this.

Text field

We don't have current integrations. Though we are in talks with Cambio GDL and form generator teams for integrating their solutions, and with Task Planning executable flows from MedicalFlows / NeoEHR.

e. Describe how/if your Solution includes tool support for (ad-hoc and stored) AQL management and use, and how well they comply with (and possibly extend) specifications, for instance the examples in the list items i-iv below. Clearly indicate which list item the answer refers to.

i) Nested and/or joined AQL queries

ii) Development and testing of variables in parametric queries

iii)AQL tools and environments for authoring queries, presentation, export and visualization of AQL responses

iv) Built in configurable/programmable pre- and/or post-processing of queries and results (server and/or client side)

Text field

Since we implemented a different query formalism, SAQM, these points don't apply directly. Though we have specific types of queries for using queries as filters for selecting EHRs that comply with complex sets of conditions, focused on the use cases mentioned on points 2.1.3.3.x

SAQM allows parametric queries, and all queries can be created and tested via our Query Builder that is integrated into the Web Console (our management GUI). All queries are built this way and stored, then executed via the REST API by providing the needed parameters and optional filters. Our top level filters are ehr_id, when querying COMPOSITIONs, FOLDERs and EHR_STATUS, and when no ehr_id is given, the query is treated as a population query. Then we have a date range for filtering by the time_committed.

In our backlog we have the addition of code post-processors for modifying the query result set structure before it's retrieved to the client. This is not currently implemented.

f. Describe how/if your Solution includes tool support for templates, and how well it complies with specifications for the examples in the list items i-iii below. Clearly indicate which list item the answer refers to.

i) Support for nested/embedded templates

ii) What template tools that have been tested and found compatible with your Solution

iii) Support for templates based on ADL 2

Text field

We use ADL 1.4 where there is no concept of nested/embedded templates. If that is needed, it will be done at the archetype level, then templates will be created over the archetypes, and then operational templates will be generated from the templates. Atomik and EHRServer work directly with operational templates 1.4

We noticed the artifacts generated by Ocean tooling are the most compliant with the specs. Veratech LinkEHR can generate some different results at the archetype level, and has very basic template support. The Archetype Designer generates archetypes like the Archetype Editor without problems, though it can export invalid/not compliant elements when generating operational templates, which we ignore but were an issue before when we had a more strict XML Schema for the OPTs. The point is: if the tool generates openEHR compliant content, we can use it.

We don't use ADL2 yet.

g. Describe how/if your Solution includes tool support for the examples in the list items i-v below. Clearly indicate which list item the answer refers to.

i) Developing GUI:s

ii) Data management

iii) Import, export, and migration of data, metadata and system configuration, in open well documented formats.

iv) SMART on FHIR integration

v) Mapping and conversion support other standards such as HL7/FHIR

Text field

i) We are not focused on GUIs though we can work with companies that have solutions on that area.

ii) Our solutions are focused on data management. We also have a tool in the openEHR Toolkit that allows us to generate test data sets that comply with a certain OPT, which we use to test EHRServer, Atomik, and we even used for testing compliance with openEHR for EHRBASE while working at HiGHmed.

iii) For import/export/migration we use the Sync API, which has access to all elements managed by the server. For doing integrations or ETL we use Mirth Connect on top of that to manage any data/message conversions needed for importing or exporting data. Formats are all openEHR compliant, except for the queries that don't have a formal openEHR format because it's not an openEHR spec, though we want to propose our query model to the SEC for standardization as an incubator spec.

iv) We don't have an implementation of SMART on FHIR.

v) All mappings are done on the Mirth Connect integration engine.

h. Describe how/if your Solution includes tool support for creation and use of entry forms based on openEHR templates. Clearly indicate which list item i-ii the answer refers to.

i) Which form rendering tools have been tested and found compatible with your CDR/platform?

ii) Do you supply a form builder and renderer? If yes, please briefly describe its features, for instance drag-n-drop, smart pictures (allowing annotations, term binding, graphs), low code/no code, conditional expressions.

Text field

We are in talks with Cambio to use their form generator. See 2.1.5.d.

i. Describe how/if your products include tool support, and how well they comply with any open specifications, for log management, such as alarms and access logs.

Text field

We offer services for integrating external solutions. Since we log a lot of events internally, there is no problem with integrating that information with an external log management tool, especially if their log format and communication specifications are open.

2.1.6 IT and Information Security

a. Describe what kind of IT security features are implemented in your Solution, for instance support for securing API, data at rest, data in transport, data in operation, data removal, and logging and audit.

Text field

Internal user authorization based on username/password, with optional 2FA and/or captcha for the Web Console.

Internal user authorization based on username/password for the REST API.

Access token authorization for REST API endpoints. Access token expiration time is configurable.

External user authorization via Keycloak.

It's not part of the system, but of the web server configuration: all communications should be HTTPS.

With internal user management, all passwords are encrypted at rest.

---

b. State if there are any relevant IT security certifications for your Solution, such as ISO27001, ISO27018.

Text field

We don't have ISO certifications. For instance, we think the policies, procedures and controls for the framework defined by ISO27001 should be designed with the organization managing and using the data, we don't think it has to do with just a couple of systems in a complex architecture.

---

c. Describe what kinds of authentication, authorization and access methods your Solution supports, for instance external IDP, role-based access control, privileged users control, just-in-time access.

Text field

Atomik and EHRServer have internal IDPs, though an external one can be integrated. We work with Keycloak. For Atomik, only the "admin" role can access the Web Console. There are "user" roles that can access the API. The "user" role is very generic, and attributes can be associated with it to define more complex access permissions. This depends on what the client needs, we can add the required attributes and the rules that check those attributes the allow/deny access to server resources.

---

d. Do you use supply chain risk management strategies/tools, such as SBOM? Describe how you mitigate risks associated with development, maintenance, acquisitions and, sunsetting of systems/components and/or services? How are risks and mitigating actions documented and what is your strategy for enforcing compliance?

Text field

We can generate an SBOM of internal and external components and can query existing platforms that report vulnerabilities in order to detect and update if anything appears there.

In terms of internal dependencies, we try to update when a. a vulnerability on a version we are using is found, b. the version we are using is no longer supported or the library we are using is not longer supported and we need to look for an alternative, c. there are new releases with meaningful improvements like performance or new features. We create a branch for testing the updated library, after it's completely validated against our systems, we roll out a new version with the update. Of course we document these processes and which versions of which libraries were introduced, when and why.

Something similar happens when updating underlying platform software like OS, database, JDK, tomcat, apache web server, etc., when sometimes also a data migration might be needed. In such cases, we do complete system backups (OS images, database dumps, web server configuration, database configuration, including the current compiled app). We also try to maintain servers with the current configurations just in case there is a problem in the migrations, so we can always rollback, fix the issues, and do the migration again.

For software updates managed by us, if breaking changes are introduced, we also release a migration path, in case data migration or alterations to the database schema are needed.

2.1.7 Training, documentation and consultant services

a. Describe the availability of course or on-line training for administrators, technicians, tool users, software developers, EHR end-users (if you provide modules/products for end-users).

Text field

Education is one of our main services. We provide an online education program on several foundations and advanced openEHR, integration and technology topics, relevant to this RFI. We also provide specific training on EHRServer and Atomik for managers, users and developers. See our education page https://www.cabolabs.com/education/

b. Describe which kind of product documentation you provide, for instance user manuals, installation guides, system administration guides.

Text field

We have basic online and public documentation. Manuals, installation guides and system administration guides are delivered to clients only. Yes, we have those.

c. Do you offer consultant services for implementation, configuration and/or development?

Text field

Yes we offer consultancy services for implementation, configuration, development, integration, support and coaching.

3. Part 2: Demonstration

3.1 Demonstration sessions

The second part of this RFI consists of a demonstration session where selected respondents, that meet the qualification criteria described below, are invited to demo their Solution.

3.1.1 Qualification and prioritization criteria

To be qualified for a demo time you will need to demonstrate a Solution that is helpful when creating applications, capturing or storing clinical data based on openEHR standards, that is, not just general integration or CDR products. If there is competition for available presentation/demo slots, the written responses to above listed questions will be used as prioritization criteria.

A maximum of six (6) suppliers will get an invite to a demo session.

3.1.2 Purpose

The purpose of the demo is to show how your Solution meets the needs of the stated target groups and the user stories described below.

3.1.3 Dates

The demo sessions are held on May 31, June 1 and June 2. June 5 is reserved as an extra date for back-up purposes. Each demo is limited to two (2) hours.

3.1.4 Format

The demo is an online two (2) hour session via Zoom. The sessions are recorded and made public on Youtube when all suppliers have held their sessions.The purpose of publication is to help other organizations interested in openEHR systems.

A demo session is on the following format:

- Short introduction of company and Solution and what is going to be presented in the demo (maximum 2 minutes)
- Demo based on target group descriptions and user stories
- Discussion with questions and answers (minimum 30 minutes)
- Optionally and on request, the recording can be stopped for the last 15 minutes of the discussion, if there are parts that should not be made publicly available.

Additional county councils may later join the RFI and attend the demonstrations as listeners.

3.1.5 Instructions

To reach business impact goals and purposes, it is essential that a procured solution meets the needs and expectations of the different target groups that will use the openEHR Solution. A number of essential target groups are identified – Platform administrator/technician, Application and content developer/administrator, Super user, External actor, Application end-user, and Newbie.

Each target group has a description and some of them have one or several user stories that highlight aspects of the target group that we think would be interesting for a demo. Use these descriptions and user stories as a basis for your demo. You are not expected to demo everything.

During the demo session, please refer to which target groups/user stories you are demonstrating.

3.1.6 Application and Content Developer/Administrator

This is an informatician, a software developer or a system/content manager. She develops applications, builds integrations, does information modeling and form building, and designs queries for information retrieval. She is also responsible for maintenance of applications, information structures and content. She gives technical support and help to other users of the openEHR tools. When functions that are more complicated are needed in an openEHR-based application, the application and content developer/administrator takes care of it. She is an advanced user with high demands on smart functions in the development tools.

User stories based on Application and content developer/administrator:

1. As an informatician I want to connect an external terminology service to make sure that the terms within the data are consistent with appropriate terminology standards and valuesets/subsets.
2. As a healthcare system developer I want to integrate software to be able to store and retrieve medical data in an openEHR EHR system alongside other healthcare system vendors.

3. As a healthcare developer working on a SmartOnFhir application I want to be able to access part of the openEHR information as standard FHIR API.

4. As an administrator or developer I want to configure or be able to create solutions for collecting IoT device measurements from patients. This includes

   a) data from medical devices that we as healthcare providers have provided, support and collect data from.

   b) data from patients' privately purchased devices (smartwatches, blood pressure meters etc) that they may have connected to apps in their Android and iOS devices - this transfer may be initiated by the patient without being actively requested by healthcare (e.g. before a visit). Such data should when stored be possible to identify as patient reported so that it can be logically separated from other data.

   c) where the data was created and by which person and device.

5. As an administrator or developer I want to configure or be able to create solutions for collecting data from patient-reported forms, photos, and videos.

6. As an administrator I want to be able to referens see Appendix A

   a) create/define metadata attributes to personal data so the Solution can be configured to meet our needs.

   b) add/update metadata for a specific piece of personal data.

   c) add/update metadata to personal data as a bulk update, e.g. for all compositions created at a certain organizational unit.

   d) use metadata to create functions managing what information a user has access to e.g. in an overview of an encounter of a patient who received specialist care.

3.1.7 Platform Administrator/Technician

This person works in the IT department, has a technical education and a few years working experience. It is his job to ensure that the platform and the development tools are sound and up and running. The platform administrator/technician is an advanced user that needs powerful tools for administration of the openEHR platform. He wants to have full control and overview, and efficient configuration and error handling and system diagnostics tools. The openEHR platform is not his only responsibility at work; there are many other systems as well, so he values extensive system documentation. Sometimes he needs support, and he is grateful that he gets it quickly.

User stories based on Platform administrator/technician:

1. As a server-admin, I want to use supporting functions so that I can carry out technical troubleshooting.

2. As a first line support tech, I want to view the system's operational status via web-UI so that I can at a glance check if there are any issues.

3. As an administrator I want to manage access-rights, e.g. configuring rules, roles and access control policies, so that I can restrict access to information based on user context and information attributes.

3.1.8 Super user

The super user is a nurse, a physician or a researcher at a healthcare unit and is interested in how new technical solutions can be used to improve the patient care, working processes, and gaining new medical knowledge. The super user maintains existing forms and templates in the openEHR-based applications that the department uses. The super user really prefers to be able to solve problems himself if possible. But in rare cases it gets a bit too complicated, for instance when programming skills are necessary or when a new template is needed, and then the super user contacts application and content developer/administrator for help and they cooperate on the solution. The super user also generates reports from the healthcare systems that the care department needs; often it is standard reports that are generated repeatedly, but sometimes a special report is needed.

The super user does not use the openEHR tools on a daily basis, but is more of a "burst" user where intense use is combined with periods of little use or no use at all. This pattern of use means that he might not ever be fluent in how to use the tools.

Since the super user does not have deep technical knowledge it is important that the tools he uses to update forms and templates are easy to use. It is also important for the super user that it is easy to get an overview of which templates and forms that the clinic is using, that version handling is easy and straightforward, and that efficient search and filtering tools are available. The super user also needs a comprehensible report generation tool.

User stories based on Super user:

1. As a clinician, I want to build and design a dynamic form, based on existing templates, with conditional form field display logic and automatic calculations, for structured documentation.

2. As a researcher, I want to create reusable methods to search, collect and present data, for example regarding a certain patient group/diagnosis and only for a specific gender at a certain age.

3. As a clinician, I want to design and generate ad hoc reports, from data collected through a form.

4. As a new employee (or occasional “burst” user) I need user friendly, and intuitive easy to use tools and graphical user interfaces.

3.1.9 Application End-User

Application end-user is a healthcare clinician or a citizen. He wants to enter and retrieve information from and to the health record system. The application end-user has no interest in the technical aspects of the applications they use; the important thing is that the applications support what they want to do in a smooth way. This may include that the applications are always available, or that only information that is relevant in the particular context is shown. In some situations, it may be of interest for the application end-user to switch language in an application. Since he could be any citizen, it might be the case that he has some kind of disability, for instance impaired vision, and is in need of things like enlarged text or textual descriptions of images. Thus, his needs concern the results of using the openEHR platform and development tools; as long as the resulting applications are stable

and good, he is happy.

User stories based on Application end-user:

1. As a clinician, I want to have a Clinical Decision Support and process support functionality, to improve the quality of care and reduce risks.

3.1.10 External Actor

External actor is a company, a student, another healthcare region, or a researcher. The external actor delivers applications or content. The external actor has no direct access to the internal systems and uses her own development tools. It is important for her that a full range of REST APIs is available, and she values extensive system documentation. It could be convenient for her to use openEHR tool licenses for a limited period when developing on behalf of a healthcare region.

3.1.11 Newbie

The Newbie is a nurse or a physician at a hospital, but may also be an informatician or a software developer. Newbie has a few years working experience but no or little knowledge of openEHR. Now is the first time Newbie takes part in maintaining existing forms and templates or in developing a new openEHR-based solution. It is important for the Newbie that the tools for developing forms are easy to learn and that the user documentation is pedagogical and covers all common use cases and functions. Some kind of introductory training to get started would help Newbie a lot.